

ANEXO 1. IMPLEMENTACIÓN DEL MODELO DE SIMULACIÓN EN LA HERRAMIENTA SPD DE COWARE®

INTRODUCCIÓN

Este anexo se evalúan los aspectos más relevantes de la implementación del modelo de simulación del nivel físico del estándar IEEE 802.11g en la herramienta de simulación Signal Processig Designer (SPD) proporcionada por la empresa CoWare®, de la cual se hará énfasis en aspectos importantes como la técnica de transmisión OFDM, la tasa de error de bits (BER), la tasa de error de bloques (BLER) y la relación Eb/No.

Como se mencionó en el capítulo 2, el sistema a simular se dividió en tres subsistemas que son: transmisor, canal y receptor tal como lo muestra la figura 1.

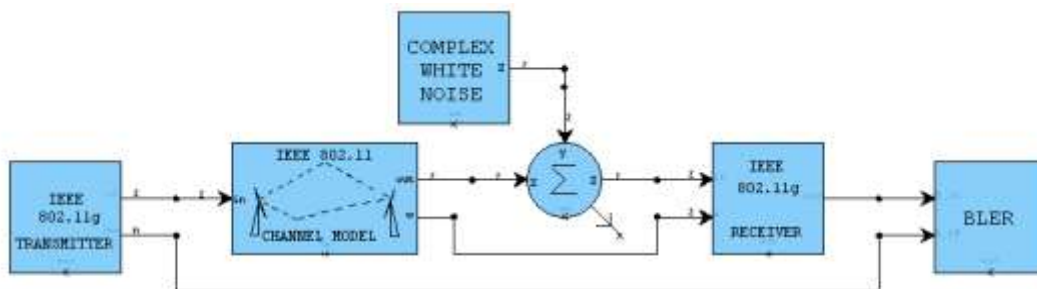


Figura 1. Modelo de Simulación del Sistema.

Cada uno de los componentes del sistema será descrito a continuación.

1. TRANSMISOR

El subsistema transmisor está conformado por los bloques campo de datos (*Data field*), codificador del campo de datos (*Data field scrambler*), polaridad del piloto (*Pilot polarity*), transmisor OFDM (*OFDM Transmit*), preámbulo PLCP (*Preamble PLCP*), codificación del campo señal (*Signal field encode*), concatenador (*Append complex*), ventana (*Window*), vector a escalar (*Complex vector to scalar*) y escalar a vector (*Complex scalar to vector*). Ver figura 2.

Parámetros:

- Velocidad de datos Mbps (*Data rate*): Tipo entero. 18, 24, 36 o 54. Indica la velocidad de transmisión de datos.
- Longitud del PSDU (*PSDU length*): Tipo entero. Indica el número de bytes que contiene el PSDU en la trama.
- Bits de servicio (*Service bits*): Tipo entero. 16. Indica el número de bits de servicio definidos para la trama PPDU.
- Bits de cola (*Tail bits*): Tipo entero. 6. Indica el número de bits de cola.
- Bits de relleno (*Pad bits*): Tipo entero. Indica el número de bits de relleno y su valor depende del tipo de modulación y codificación que se utilice.
- Bits de datos (*Data bits*): Tipo entero. Indica el número de bits calculados para el campo datos.
- Modulación (*Modulation*): Tipo cadena. "BPSK", "QPSK", "16QAM" ó "64QAM". Indica la modulación utilizada en el campo datos, calculada según el campo de velocidad de transmisión de datos (*Rate*).
- Tasa de codificación (*Coding rate*): Tipo cadena. "1/2", "2/3" ó "3/4". Indica la tasa de codificación del código convolucional.

- N_{PPSC} : Tipo entero. 1, 2, 4 o 6. Indica el número de bits por subportadora, calculada a partir del tipo de modulación.
- N_{CBPS} : Tipo entero. Indica el número de bits codificados por símbolo OFDM, calculado a partir de la velocidad de transmisión.
- N_{DBPS} : Tipo entero. Indica el número de bits de datos por símbolo OFDM, calculado a partir de la velocidad de transmisión de datos.
- Número de símbolos OFDM (*Number of OFDM symbols*): Tipo entero. Indica el número de símbolos OFDM que transportan los bits de datos en el PDU.

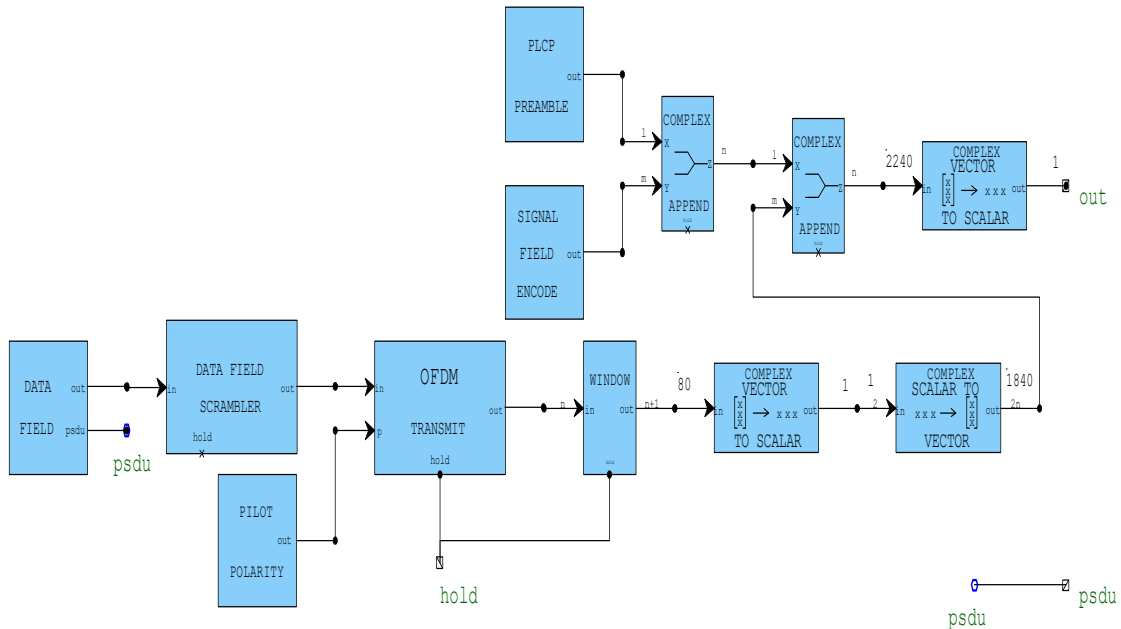


Figura 2. Subsistema transmisor.

1.1. Bloque Campo de datos (Data field).

El bloque campo de datos es el que contiene la información o datos de usuario en la simulación y es ésta información la que se transmite a través del sistema. Este bloque a su vez se encuentra dividido en los bloques: Datos aleatorios (*Data random*), Fuente de señal (*Signal source*), Control (*Control*) y valor Constante (*Constant*) los cuales conforman el campo PSDU de la trama; Escalar a vector (*Scalar to vector*), tres bloques Vector constante (*Constant vector*) conformando respectivamente los campos Servicio (*Service*), Cola (*Tail*) y Relleno (*Pad*) y tres bloques Concatenadores de vectores (*Vector join Vector*) que concatenan los campos mencionados. Ver figura 3.

Parámetros:

- Longitud del PSDU (*PSDU length*): Tipo entero. Indica el número de bytes que contiene el campo datos en el PSDU.
- Bits de servicio (*Service bits*): Tipo entero. 16. Indica el número de bits de servicio en el campo datos.
- Bits de cola (*Tail bits*): Tipo entero. 6. Indica el número de bits de cola en el campo datos.
- Bits de relleno (*Pad bits*): Tipo entero. Indica el número de bits de relleno en el campo datos.

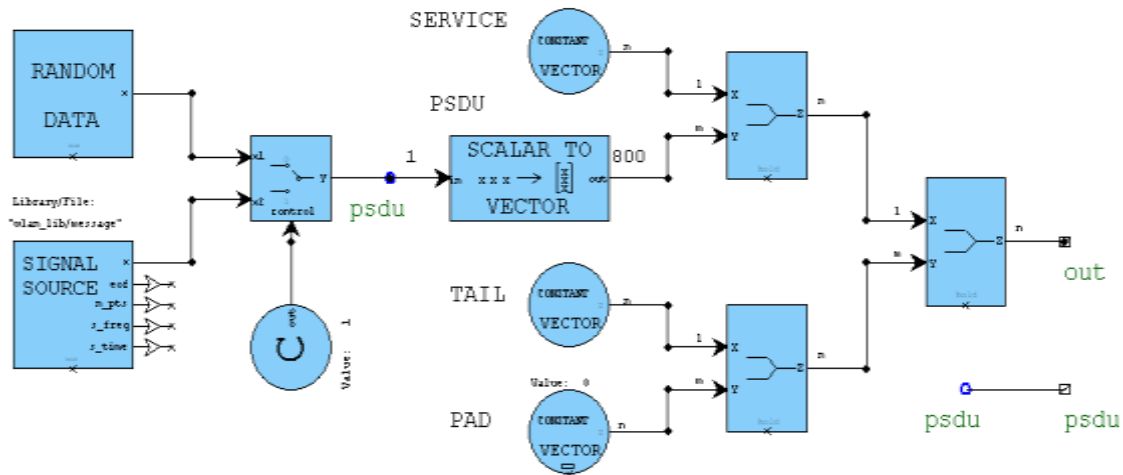


Figura 3. Bloque Campo de datos.

1.1.1. Datos aleatorios (Random data)

Este bloque se encarga de generar un flujo aleatorio de datos binarios. Este bloque cuenta con los siguientes parámetros:

- Velocidad de bit (*Bit rate*): Tipo doble. Indica la velocidad de los bits la cual está dada en Hz.
- Probabilidad de cero (*Probability of zero*): Tipo doble. 0.0 a 1.0. Indica la probabilidad de que la salida sea 0 (falsa) para cada bit.

1.1.2. Fuente de Señal (Signal Source)

Este bloque entrega una señal de tipo escalar binaria, la cual es generada a partir de un archivo llamado message.ascsig, el cual se encuentra dentro de la librería wlan_lib. Además este archivo define el tamaño de la señal, la frecuencia de muestreo, el tiempo de inicio y el tipo de la señal. Se definen dos parámetros editables para este bloque y el resto se dejan por defecto:

- Ruta (*Path*): Indica la ruta donde se encuentra almacenado el archivo generador de la señal, para el caso en cuestión el archivo se encuentra en la ruta "c:\ProgramFiles\SPD2007.1\libraries\wlan_lib".
- Tipo de archivo (*File format*): Tipo cadena. "binary" ó "ascii". Indica el tipo de archivo que se utiliza el cual puede ser binario o ASCII que es el que se trabaja en este caso.

1.1.3. Conmutador de entrada (switch in)

Este bloque conecta una de sus dos entradas (x1 o x2) a su salida "y", según el estado de la lógica en la entrada de "control". Cuando la entrada de control es 0, se utiliza x1 y cuando la entrada de control es 1, se utiliza x2.

1.1.4. Escalar a vector (scalar to vector)

Este bloque convierte una secuencia de datos de N valores a un vector de N componentes. El valor de N es un parámetro situado en la salida del bloque.

Parámetros:

- Tamaño vector de salida (*out_IOVEC_LEN*): Tipo entero mayor que 1. Especifica el tamaño o número de componentes del vector de salida y el número de datos agrupados a la entrada para generar el vector de salida.
- Primer elemento cargado (*first element loaded*): Tipo cadena. "first" ó "Last". Determina el orden en el cual los componentes del vector deben ser puestos en la salida. Si se ha seleccionado "first", el primer dato recibido es el componente menos significativo del vector de salida. Si se ha seleccionado "Last", el último dato recibido es el componente menos significativo del vector de salida. En este caso se eligió "first".

1.1.5. Constante (constant)

Este bloque genera un vector de valor constante y es utilizado para generar cada uno de los siguientes campos: servicio, cola y relleno. El valor del número de componentes de cada vector se puede leer desde un archivo o puede ser fijado por parámetros.

Parámetros:

- Tamaño vector (*Size of vector*): Tipo entero. 1,2... ó 16. Indica el número de componentes en el vector de salida.
- Longitud del PSDU (*PSDU length*): Tipo entero. Indica el número de bytes que contiene el PSDU en la trama.
- Leer desde archivo (*Read from file*): Tipo cadena. "yes" ó "no". Indica si el valor del número de componentes es leído desde un archivo o desde un parámetro.
- Nombre del archivo de señal (*Signal file name*): Tipo cadena. "nombre_de_archivo". Indica el nombre del archivo si se ha seleccionado la opción "yes" en el parámetro anterior.

1.1.6. Vector concatenador (*Vector join vector*)

Este bloque une o concatena los vectores de entrada X y Y, formando el vector de salida Z. Los componentes menos significativos del vector Z están formados por los componentes del vector X, seguidos por los componentes del vector Y. Las longitudes de los vectores se especifican en los parámetros.

Parámetros:

- Longitud del vector X (*X vector length*): Tipo entero mayor que 0. Indica el número de componentes del vector X.
- Longitud del vector Y (*Y vector length*): Tipo entero mayor que 0. Indica el número de componentes del vector Y.
- Longitud del vector Z (*Z vector length*): Tipo entero mayor que 0. Indica el número de componentes del vector Z.

1.2. Codificador del Campo de Datos (*Data Field Scrambled*)

El bloque codificador del campo de datos reordena los bits a transmitir del campo de datos de tal forma que se evite largas cadenas de ceros o unos. Este bloque está conformado internamente con los bloques: Concatenador (*Vector join vector*), Vector a escalar (*Vector to Scalar*), Vector constante (*Constant vector*), Inicializador de aleatorización (*Scrambler initialize*), Control (*Control*) y valor Constante (*Constant*), Aleatorizador desaleatorizador (*Scrambler Descrambler*), Sincronización (*Timing*), Escalar a vector (*Scalar to vector*), Desconcatenador (*Vector split vector*) y Sustitución (*Replace*) tal como se indica en la figura 4.

Parámetro:

- Bits de datos (*Data bits*): Tipo entero. Indica el número de bits calculados para el campo datos.

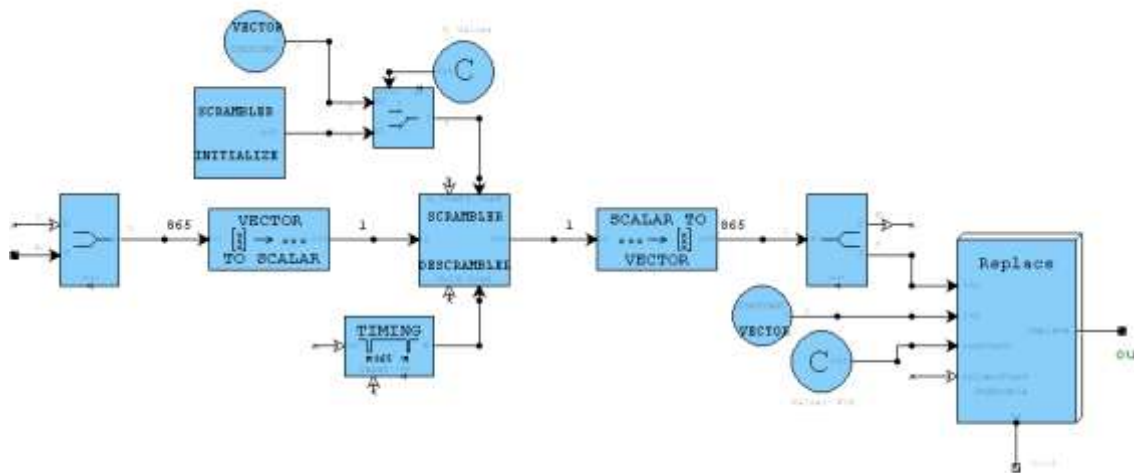


Figura 4. Bloque Codificador del Campo de Datos.

1.2.1. Inicialización del codificador (Scrambler Initialize)

Este bloque genera un vector de estado inicial para el bloque Codificador/Decodificador (Scrambler/Descrambler). El estado inicial del codificador se fija en un estado no cero pseudo-aleatorio.

1.2.2. Codificador – Decodificador (Scrambler descrambler)

Este bloque implementa el polinomio generador para la codificación y decodificación del campo de datos PLCP.

1.2.3. Sincronización (Timing)

Este bloque genera pulsos periódicos los cuales son utilizados normalmente para la sincronización del sistema. En los parámetros del bloque se fijan el valor del periodo, ancho del pulso y polaridad del pulso.

Parámetros:

- Periodo (*Period*): Tipo entero mayor que cero. Indica el periodo del pulso ($1/f$) expresado como un número de muestras.
- Inicio de conteo (*Start count*): Tipo entero mayor que cero. Indica el número de muestras con las cuales se produce el primer pulso.
- Restablecer conteo (*Restart count*): Tipo entero mayor que cero. Indica el número de muestras que deben transcurrir antes que el próximo pulso sea generado. El restablecimiento finaliza con una transición de la señal de alto a bajo en la entrada "reset".
- Ancho del pulso (*Pulse width*): Tipo entero mayor que cero y menor que el periodo. Indica la duración del pulso expresado como un número de muestras.
- Salida invertida (*Invert output*): Tipo cadena. "yes" ó "no". Sí el valor está en "yes", la salida durante el pulso es 1 y el resto del tiempo es 0, sí el valor está en "no", la salida durante el pulso es 0 y el resto del tiempo es 1.

1.2.4. Vector a escalar (Vector to scalar)

Este bloque convierte un vector de N componentes en una secuencia de N muestras de datos. El valor N es un parámetro situado en la entrada del bloque.

Parámetros:

- Parámetros de entrada en el bloque (*N parameter at input of block symbol*): Tipo entero mayor que 1. Especifica el número de componentes del vector de entrada.
- Primer elemento de salida (*First element out*): Tipo cadena. 'first' ó 'last'. Determina el orden en el cual los componentes del vector son ubicados en la salida. Si se selecciona "first", el componente menos significativo del vector de entrada es el primero en la salida, por el contrario, si se selecciona "last", el componente más significativo del vector de entrada es el primero en la salida. Para esta simulación se fijó en "first".

1.2.5. Divisor de vector (Vector split vector)

Este bloque divide el vector de entrada X, en dos vectores de salida Y y Z, el vector Y formado por la parte inferior del vector X y el vector Z formado por la parte superior del vector X. La longitud de los vectores se especifica como parámetro.

Parámetros:

- Longitud del vector X (*X vector length*): Tipo entero mayor que 0. Indica el número de componentes del vector X.
- Longitud del vector Y (*Y vector length*): Tipo entero mayor que 0. Indica el número de componentes del vector Y.
- Longitud del vector Z (*Z vector length*): Tipo entero mayor que 0. Indica el número de componentes del vector Z.

1.2.6. Sustitución (Replace)

El bloque toma un primer vector y sobrescribe una parte del mismo con un segundo vector.

Parámetros:

- Entrada 1 (In 1): Indica el vector original.
- Entrada 2 (In 2): Indica los nuevos datos utilizados los cuales se van a sustituir en el vector original.
- Fila de inicio (*rowstart*): Tipo doble. Indica la fila donde se inicia la sustitución. Este valor se trunca a un entero internamente definido.
- Columna de inicio (*colstart*): Tipo doble. Indica la columna donde se inicia la sustitución. El valor se ignora cuando la entrada es un vector.

1.3. Polaridad del piloto (Polarity pilot)

Este bloque genera la polaridad de las subportadoras piloto insertadas en cada símbolo OFDM. La polaridad de las subportadoras piloto está determinada por una secuencia pseudo aleatoria de longitud 127, generada por el Aleatorizador (*Scrambler*). La estructura de este bloque está compuesta por otros bloques tales como: bloque de Secuencia (*Timing*), bloque Aleatorizador/Desaleatorizador (*Scrambler/Descrambler*), bloque Vector constante (*Constant vector*), bloque Inversor (*Inverter*), dos bloques Escalar constante (*Constant scalar*), bloque multiplicador escalar – escalar (*Scalar – scalar*) y el bloque Restador de escalar (*Subtractor scalar*) tal como se indica en la figura 5.

Parámetros:

- Número de símbolos OFDM (*Number of OFDM symbols*): Tipo entero. Indica el número de símbolos OFDM que contiene el campo Datos.

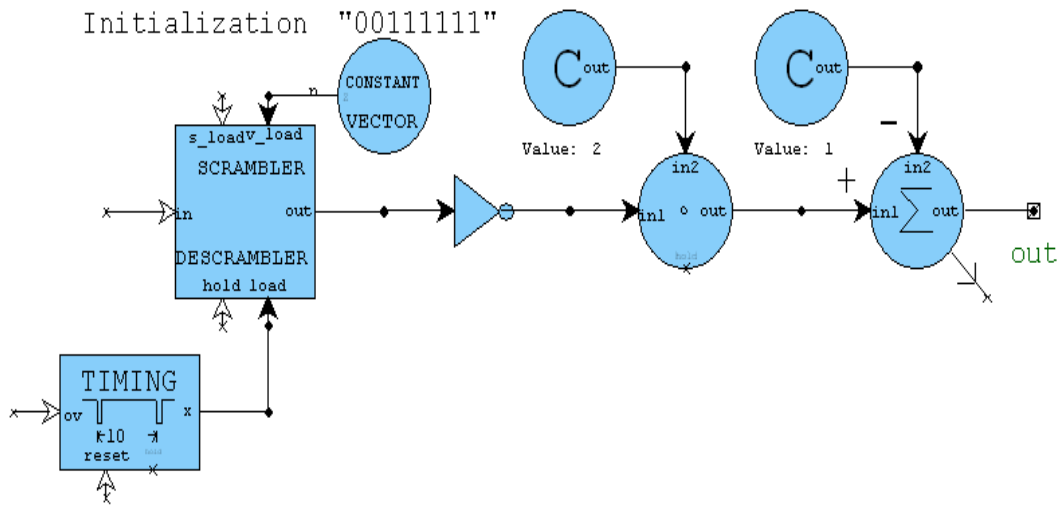


Figura 5. Bloque Polaridad del Piloto.

1.3.1. Multiplicador escalar – escalar (*Scalar – scalar*)

Este bloque multiplica dos señales de entrada tipo escalar doble y entrega a la salida el resultado de este producto. Ver figura 6.

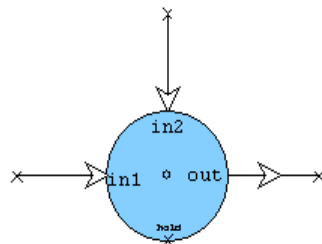


Figura 6. Bloque multiplicador escalar – escalar.

1.3.2. Restador escalar (*Subtractor scalar*)

Este bloque resta la entrada 2 de la entrada 1, luego una salida indica el resultado de la resta y la otra indica cualquier desbordamiento. Ver figura 7.

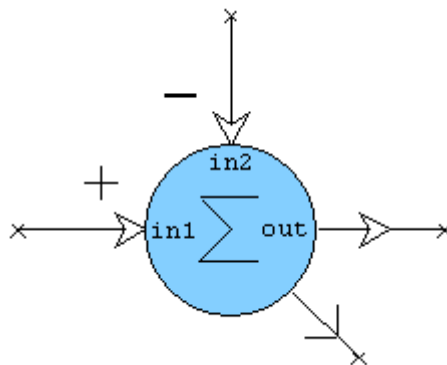


Figura 7. Bloque restador escalar.

1.4. Transmisor OFDM (*OFDM Transmit*)

Este bloque es el principal componente para la implementación del transmisor 802.11g, el cual realiza la codificación del canal, entrelazado de datos, la modulación de subportadoras, inserción de las subportadoras piloto y el mapeo de las subportadoras. Está compuesto por los siguientes bloques: Codificador OFDM (*OFDM Encode*), Inserción del piloto (*Insert pilot*), Vector de mapeo FFT (*Map FFT*), Modulador OFDM (*OFDM Mod*) y Prefijo cíclico (*Cyclic ext*) tal como lo indica la figura 8.

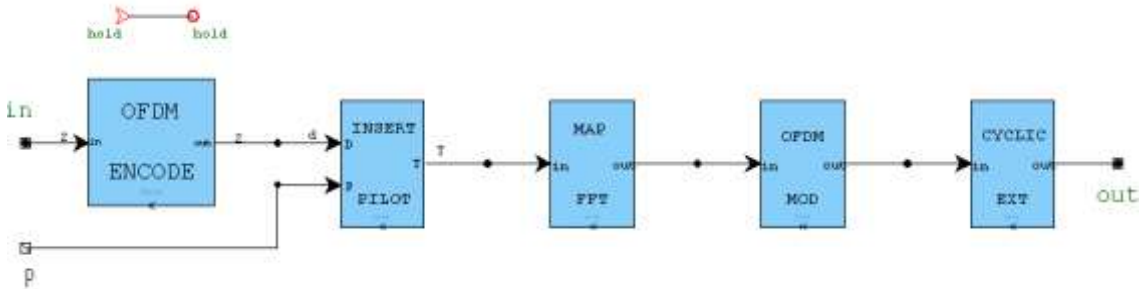


Figura 8. Bloque Transmisor OFDM.

1.4.1. Codificador OFDM (*OFDM Encode*)

Este bloque realiza la codificación del canal, perforación, entrelazado de datos y modulación de subportadoras. La estructura de este bloque se muestra en la figura 9.

Parámetros:

- Modulación (*Modulation*): Tipo cadena. Indica el tipo de modulación utilizada para el campo de Datos.
- Tasa de codificación (*Coding rate*): Tipo cadena. "1/2", "2/3" o "3/4". Indica la tasa de codificación para codificar el canal.
- N_{BPSK} : Tipo entero. Indica el número de bits por subportadora.
- N_{CBPS} : Tipo entero. Indica el número de bits codificados por símbolo OFDM.
- Campo de datos (*Data field*): Tipo entero. Indica el número total de bits en el campo de datos.
- Bits perforados (*Punctured bits*): Tipo entero. Indica el número total de bits codificados después del perforado.

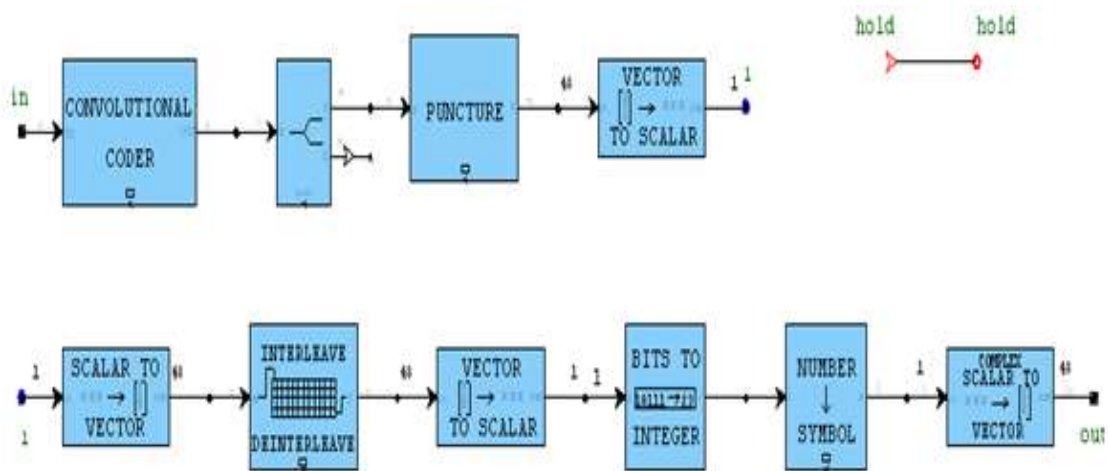


Figura 9. Bloque Codificador OFDM¹.

¹ El punto marcado con 1 del gráfico superior indica conexión con el punto 1 del gráfico inferior (conectores).

1.4.1.1. Codificador convolucional (*Convolutional coder*)

Este bloque codifica el vector de bits en la entrada y produce a la salida un vector de bits codificado mediante una codificación convolucional. K bits de entrada están codificados en N bits de salida lo cual da una tasa de K/N.

Parámetros:

- Tamaño del vector de entrada (*Input vector size*): Tipo entero mayor que 0. Indica el número de componentes en el vector de entrada.
- Longitud de trama de información – K_0 (*Information frame length*): Tipo entero mayor que 0. Indica los bits de entrada por símbolo OFDM.
- Longitud de trama transmitida – N_0 (*Transmit frame length*): Tipo entero mayor que 0. Indica los bits de salida por símbolo OFDM.
- Longitud restringida – I (*Constraint length*): Tipo entero mayor que 0. Indica el orden del polinomio usado en el codificador convolucional.
- Archivo generador de entrada (*Generator input file*): Tipo cadena. "nombre_de_archivo". Indica el nombre del archivo de texto que contiene los parámetros y los polinomios generadores del codificador.
- Modo (*Mode*): Tipo cadena. Indica uno de los tres modos de codificación: "continuous", "tailbits" y "tailbiting". Para el modo "continuous" el vector de datos de entrada es considerado como un flujo de datos continuo, para el modo "tailbits" el estado del codificador se restablece a un estado específico conocido por el valor de los bits de cola antes de la codificación de cada uno de los vectores y para el modo "tailbiting" el estado del codificador es inicializado con el primer elemento del vector de entrada.
- Valores de bits de cola (*Tail bits values*): Tipo cadena. Indica los bits que se utilizarán al final de la trama para rellenar el codificador. Este parámetro debe ser una cadena de valores de 0's y 1's. El bit menos significativo de la cola es el primero en ingresar al codificador.

1.4.1.2. Perforado (*Puncture*)

Este bloque realiza el perforado de los bits codificados, este procedimiento omite algunos de los bits codificados en el transmisor y por tanto reduce el número de bits transmitidos incrementando de esta forma la tasa de codificación. Los patrones de perforación para diferentes tasas de codificación son mostrados en la tabla No. 1.

Tasa de codificación	Vector de perforado
1/2	1
2/3	1110
3/4	111001111001111001

Tabla 1. Patrones de perforación.

Parámetros:

- Patrón de perforado (*Puncturing pattern*): Tipo cadena. Indica el patrón de unos y ceros, los ceros representan los bits omitidos de la salida del codificador y los unos representan los bits que se mantienen.

1.4.1.3. Entrelazador – desentrelazador (*Interleave - deinterleave*)

Este bloque realiza el entrelazado de los bits de datos codificados, el parámetro utilizado por este bloque es:

- Modo (*Mode*): Tipo cadena. "interleave" ó "deinterleave". Indica el proceso que va a realizar el bloque ya sea como entrelazador o como desentrelazador.

1.4.1.4. Bits a entero (*Bits to integer*)

Este bloque convierte un flujo de bits a un flujo de enteros, los parámetros de este bloque permiten fijar el número de bits por entero.

Parámetros:

- Parámetro a la entrada del símbolo del bloque (*Parameter at input of block symbol*): Tipo entero. 1,2,... ó 32. Indica el número de bits con los cuales se obtiene el número entero.
- Primer bit a enviar (*First bit sent*): Tipo cadena. "MSB" ó "LSB". Indica el orden en que son recibidos los bits. "MSB" para recibir primero el bit más significativo y "LSB" para recibir primero el bit menos significativo.

1.4.1.5. Número a símbolo (*Number to symbol*)

Este bloque mapea los datos de entrada a un punto de la señal en la constelación de salida. Un archivo asigna los puntos en la constelación y su respectiva secuencia, este archivo puede ser creado o editado por el usuario y consta de una lista de números complejos donde el primer punto es el número 0, el segundo punto es el número 1 y así sucesivamente. Este bloque cuenta con los parámetros descritos a continuación:

- Archivo de constelación (*Constellation file*): Tipo cadena. "nombre_de_archivo". Indica el nombre del archivo que contiene la señal compleja de los puntos de la constelación en orden numérico.
- Rotación de la constelación (*Constellation rotation*): Tipo doble. -360 a 360. Indica el ángulo de rotación, en grados, de la constelación.
- Potencia de normalización (*Normalize power*): Tipo cadena. "yes" ó "no". Indica el tipo de señal de constelación utilizando normalización. Si es "yes" la potencia de la señal de constelación es normalizada a un nivel específico, si es "no" se aplica un factor de escala a la señal de constelación.

1.4.1.6. Escalar complejo a vector (*complex scalar to vector*)

Este bloque convierte una secuencia de datos complejos de N valores a un vector complejo de N componentes. El valor de N es un parámetro situado en la salida del bloque.

Parámetros:

- Parámetro N de salida del bloque símbolo (*N parameter at output of block symbol*): Tipo entero. Especifica el tamaño o número de componentes del vector de salida y el número de datos agrupados a la entrada para producir el vector de salida.
- Primer elemento cargado (*first element loaded*): Tipo cadena. "first" ó "last". Determina el orden en el cual los componentes del vector deben ser puestos en la salida. Si se ha seleccionado "first", el primer dato recibido es el componente menos significativo del vector de salida. Si se ha seleccionado "last", el último dato recibido es el componente menos significativo del vector de salida. En este caso se eligió "first".

1.4.2. Inserción del piloto (*Insert pilot*)

Este bloque adiciona cuatro subportadoras piloto a las 48 subportadoras de la señal OFDM además de adicionar una subportadora ubicada en cero (0) con lo cual el número de

subportadoras se incrementa a 53. Estas subportadoras piloto son insertadas en las posiciones -21, -7, 7, 21 de la señal OFDM.

1.4.3. Mapeo transformada rápida de Fourier (*MAP FFT*)

Este bloque realiza el mapeo de las 53 subportadoras en un vector de longitud 64 de acuerdo a la figura 10.

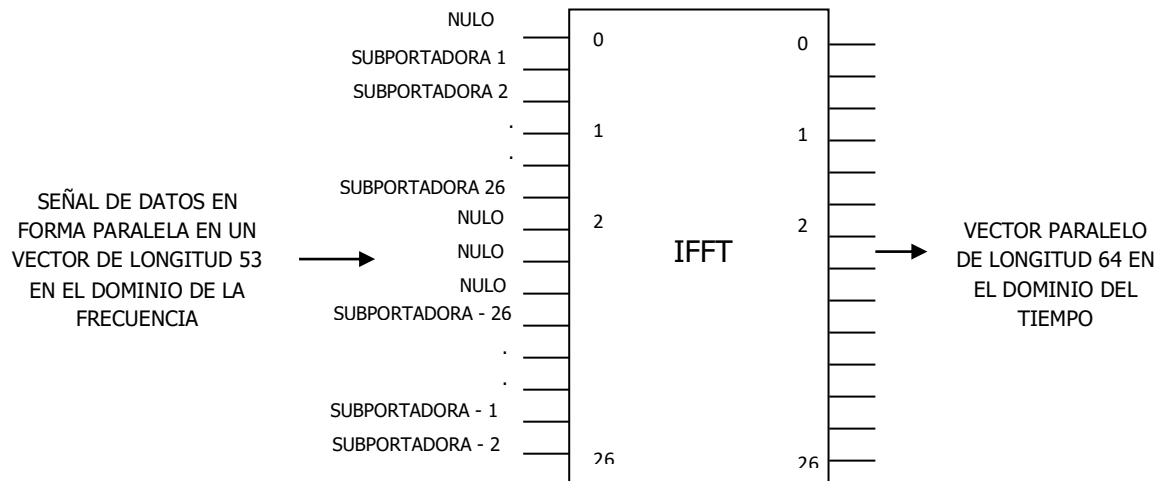


Figura 10. Estructura interna del bloque transformada inversa rápida de Fourier.

1.4.4. Modulador OFDM (*OFDM MOD*)

Este bloque realiza la Transformada Inversa Rápida de Fourier del vector de longitud 64 de la señal OFDM y lo convierte al dominio del tiempo de acuerdo con la ecuación 1.

$$x(n) = \frac{1}{64} \sum_{K=0}^{63} X(k) e^{\frac{j2\pi kn}{64}} \quad \text{Ecuación (1)}$$

Este bloque está conformado por los bloques Complejo FFT, Complejo escalar y Escalar contante. Ver figura 11.

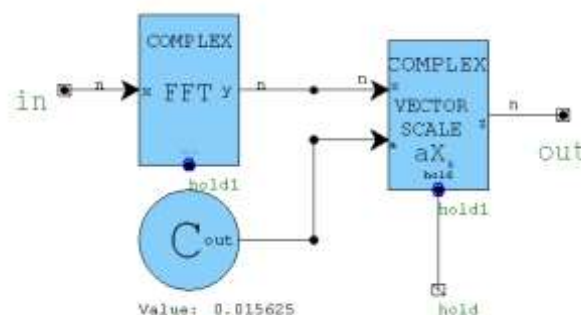


Figura 11. Bloque Modulador OFDM.

1.4.4.1. Complejo FFT (*Complex FFT*)

Este bloque realiza la transformada inversa rápida de Fourier, en la entrada se ingresa un vector complejo de longitud N y produce a la salida un vector complejo. Para este cálculo se tiene en cuenta la ecuación 1. descrita anteriormente.

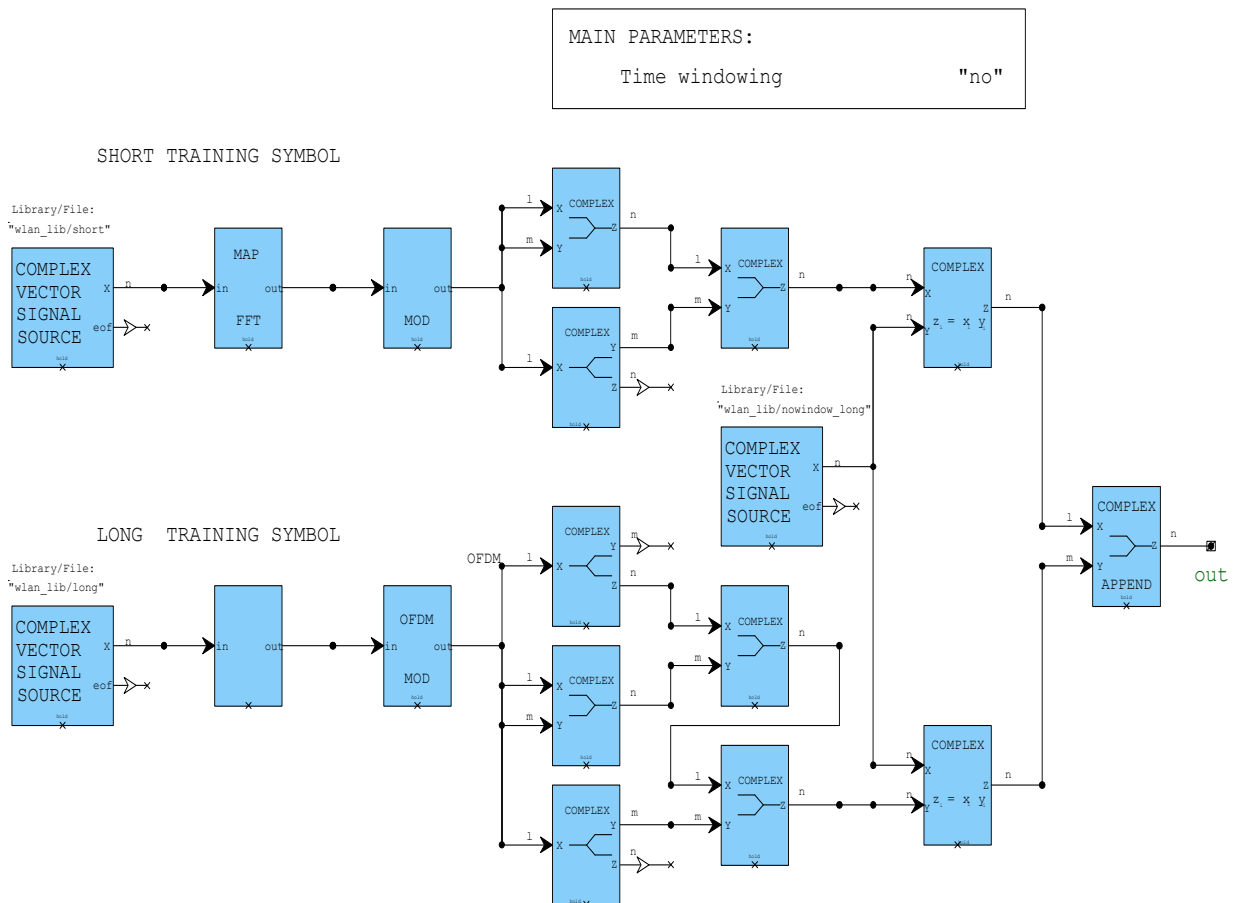


Figura 12. Bloque Preámbulo PLCP.

1.5.1. Vector complejo (Complex vector signal source)

Este bloque lee los datos de la señal compleja almacenada en un archivo y coloca los valores en un formato vectorial complejo. En cada iteración, el vector de salida es llenado con puntos consecutivos de la señal almacenada en el archivo.

Parámetros:

- Librería/Archivo (*Library/File*): Tipo cadena. "nombre_de_archivo". Indica el nombre del archivo en el cual están almacenados los datos de la señal compleja.
- Tamaño del vector (*Size vector*): Tipo entero mayor que 0. Indica el número de componentes del vector complejo de salida. Este también indica el número de valores leídos de la señal compleja almacenada en el archivo en cada iteración.
- Formato de archivo (*File format*): Tipo cadena. "binary" ó "ascii". Especifica si los datos se almacenan en formato binario o en ASCII.
- Mostrar el orden de la señal *GET_SIM* (*GET_SIM Signal Display Order*): Tipo entero mayor que 0. Indica el orden en que aparece la señal cuando está es desplegada en el calculador de señal cuando se despliegan los resultados de la simulación.

1.5.2. Multiplicador complejo (Mult complex)

En este bloque se obtiene un vector complejo de salida denotado por Z resultado de la multiplicación de los vectores complejos de entrada X y Y. Cada componente de los vectores de entrada es multiplicado individualmente para generar el componente del vector de salida. El tamaño de los vectores se especifica como parámetro.

Parámetro:

- Tamaño de los vectores (*size of vectors*): Tipo entero mayor que 0. Indica el número de componentes de los vectores de entrada y salida.

1.5.3. Concatenador de complejos (**Complex append**)

En este bloque, dos vectores de entrada X y Y son concatenados (unidos) el uno con el otro, mientras los últimos n elementos del vector X son reemplazados por los primeros n elementos del vector Y, donde n es definida por el parámetro "*Overlap length*".

Parámetros:

- Longitud del vector complejo X (*X vector length*): Tipo entero mayor que 0. Indica la longitud del vector de entrada X el cual genera la parte superior del vector de salida.
- Longitud del vector complejo Y (*Y vector length*): Tipo entero mayor que 0. Indica la longitud del vector de entrada Y el cual genera la parte inferior del vector de salida.
- Longitud del vector complejo Z (*Z vector length*): Tipo entero mayor que 0. Indica la longitud del vector de salida.
- Superposición de la longitud (*overlap length*): Tipo entero mayor que 0. Indica la longitud de la superposición de elementos.

1.6. Codificador del campo señal (**Signal field encode**)

Este bloque genera el campo señal del PPDU, este campo consiste de un símbolo OFDM y le sigue al preámbulo PLCP. Este campo transmite información acerca del tipo de modulación y de la tasa de codificación utilizada en el resto de la trama transmitida.

Ver figura 13.

Parámetros:

- Longitud del PSDU (*PSDU Length*): Tipo entero. Indica el número de bytes que contiene el PSDU de la trama.
- Número de bits por subportadora (N_{BPSC}): Tipo entero. Indica el número de bits que son asignados a cada subportadora.
- Tasa de codificación (*Coding rate*): Tipo cadena. Indica la tasa de codificación convolucional.
- Velocidad de transmisión de datos (*Data rate - Mbps*): Tipo entero. Indica la velocidad de transmisión de los datos.
- Tiempo de ventanas (Time windowing): Tipo cadena. Indica el tiempo de activación de ventanas.

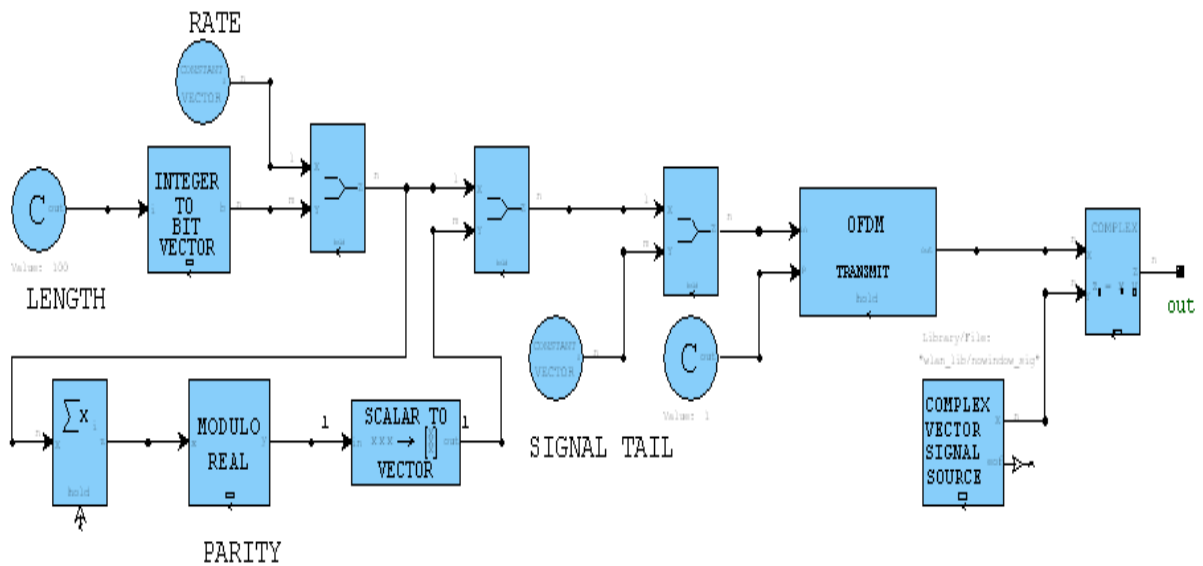


Figura 13. Bloque Codificador Campo Señal.

1.6.1. Entero a vector de bits (*Integer to bit vector*)

Este bloque convierte la señal de entrada a un vector de bits.

Parámetros:

- Tamaño del vector de salida (*Output vector size*): Tipo entero. 1,2,...ó 32. Indica el número de bits utilizados para representar el valor de entrada.
- Ordenamiento de los bits (*Bit ordering (MSB ó LSB) first*): Tipo cadena. "MSB" ó "LSB". Indica el orden en que los bits son ubicados en el vector de salida. "MSB" para ubicar el bit más significativo de primero y "LSB" para ubicar el bit menos significativo de primero.

1.6.2. Componente sumador racional (*Sum component-wise*)

Este bloque suma cada uno de los componentes del vector de entrada X y produce un escalar de salida Z el cual es el resultado de esta suma.

Parámetros:

- Tamaño del vector (*Size of vectors*): Tipo entero mayor que 0. Indica el número de componentes en el vector de entrada.

1.6.3. Modulo real (*Modulo real*)

Este bloque divide la entrada X por el "Mod constant" el cual se especifica en los parámetros del bloque. El residuo de ésta división es colocado en la salida Y del bloque. Este módulo se ejecuta con el fin de establecer el bit de paridad del campo señal que en este caso debe ser de paridad positiva³.

³ El bit de paridad es de paridad positiva (1) si el número de bits en la cadena con valor 1 es un número par, y es de paridad negativa (0) si el número de bits en la cadena con valor 1 es un número impar.

Parámetros:

- Módulo constante (*Modulo constant*): Tipo doble. Indica la base alrededor de la cual tiene lugar el módulo de la división.

1.7. Vector complejo a escalar (Complex vector to scalar)

Este bloque convierte un vector complejo de N componentes en una secuencia de N muestras de datos complejos. El valor N es un parámetro ubicado en la entrada del bloque.

Parámetros:

- Parámetro N a la entrada en el bloque (*N parameter at input of block symbol*): Tipo entero mayor que 1. Indica el número de componentes en el vector de entrada; cada uno de los datos muestreados ubicados en la salida pertenecen a cada una de los componentes del vector de entrada. Las N salidas operan N veces más rápido que el vector de entrada.
- Primer elemento de salida (*First element out 'first' or 'last'*): Tipo cadena. "first" ó "last". Determina el orden en el cual los componentes del vector deben ser puestos en la salida. Si se ha seleccionado "first", el componente menos significativo del vector de entrada es primero en la salida. Si se ha seleccionado "last", el componente más significativo del vector de entrada es primero en la salida.

1.8. Escalar a vector complejo (Scalar to vector complex)

Este bloque convierte una secuencia de N muestras de datos complejos a un vector complejo de longitud N, donde el valor de N es un parámetro ubicado en la salida del bloque.

Parámetros:

- Parámetro N a la salida en el bloque (*N parameter at output of block symbol*): Tipo entero mayor que 1. Indica el número de componentes en el vector de salida, el número de muestras de datos recogidas en la entrada producen el vector de salida. Las N entradas operan N veces más rápido que la salida.
- Primer elemento de salida (*First element out 'first' or 'last'*): Tipo cadena. "first" ó "last". Determina el orden en el cual las muestras de datos son acumuladas en el vector de salida. Si se elige "first", el primer dato tomado del escalar de entrada es el componente menos significativo del vector de salida. Si se ha seleccionado "last", el último dato del escalar de entrada es el componente menos significativo del vector de salida.

2. CANAL

El canal está compuesto por el bloque generador de ruido blanco gaussiano aditivo (AWGN) complejo, un modelo de canal IEEE 802.11 con el cual se simula la multitrayectoria y un bloque sumador de señales complejas, el cual adiciona la señal de ruido a la señal de información transmitida. Ver figura 14.

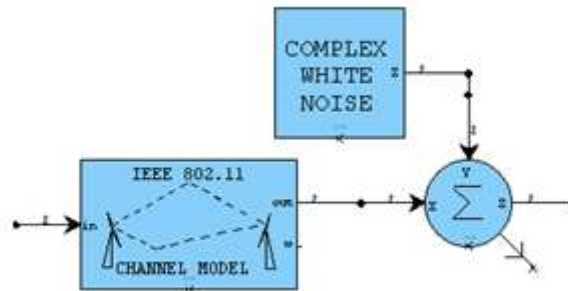


Figura 14. Bloque Canal.

2.1. Ruido Blanco Complejo (Complex White noise)

Este bloque genera ruido blanco gaussiano complejo, a partir del valor medio y la varianza de ruido, que especifican las características de amplitud del ruido en la salida Z.

Se debe tener en cuenta la fórmula de la varianza de ruido que está en función de E_b/N_0 y se describe a continuación:

$$\text{Varianza de ruido} = \frac{\text{Potencia de la señal} * \text{Periodo de símbolo}}{\text{Tiempo de muestreo} * 10^{\frac{E_b}{N_0}}} \quad \text{Ecuación (2)}$$

Donde:

$$\frac{E_b}{N_0} = \frac{E_b}{N_0} + 10 * \log_{10}(k) [dB] \quad \text{Ecuación (3)}$$

Donde: k es el número de bits de información por símbolo OFDM de entrada.

Parámetros:

- Valor Medio (*Mean*): Especifica el valor medio de la distribución gaussiana. En este caso este valor es igual a 0.
- Varianza (*Variance*): Especifica el valor de la varianza de la distribución gaussiana. En este caso se fijó con una ecuación que permite definir la varianza en función de la relación E_b/N_0 . La sentencia es:

$$((\text{pow}(10.0,((-0.1*(\text{SNR_db}))) * 1.0)))/20)$$

Esta sentencia representa la ecuación 2 anteriormente mencionada.

2.2. Modelo de canal IEEE 802.11 (IEEE 802.11 Model channel)

Este bloque implementa un modelo de canal con una línea de retardo, que simula un canal con multitrayecto y con desvanecimiento selectivo en frecuencia. Se basa en el modelo de canal adoptado por el grupo de trabajo IEEE 802.11. [18]

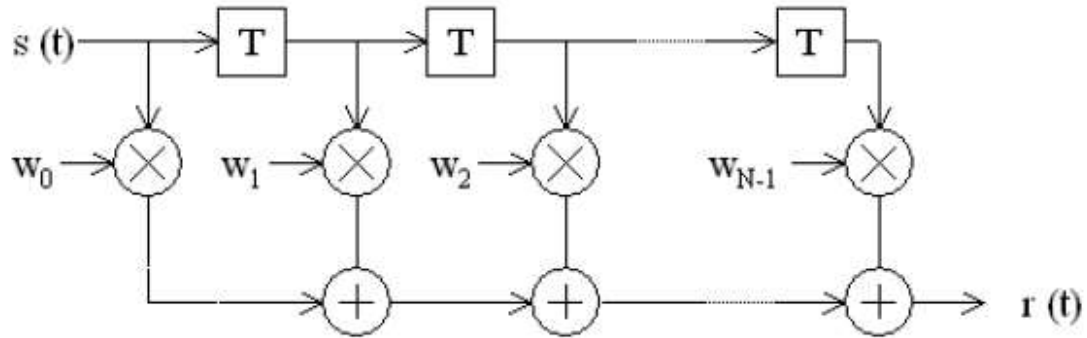


Figura 15. Modelo de canal.

Los pesos de cada derivación son números aleatorios complejos con magnitud que varía con distribución Rayleigh y fase que varía con distribución uniforme. El desvanecimiento cambia de una trama a otra y permanece constante durante cada trama. La potencia media de los pesos decae exponencialmente como función de la dispersión del retardo en el canal. A continuación se presenta la ecuación que define la respuesta impulsiva del modelo de canal.

$$h_k = N\left(0, \frac{\sigma_k^2}{2}\right) + jNN\left(0, \frac{\sigma_k^2}{2}\right) \quad \text{Ecuación (4)}$$

$$\sigma_k^2 = \sigma_0^2 \cdot e^{-k \cdot T_s / T_{rms}} \quad \text{Ecuación (5)}$$

$$\sigma_k^2 = 1 - e^{-T_s / T_{rms}} \quad \text{Ecuación (6)}$$

Donde:

- **T_{rms}**: retardo rms del canal.
- **T_s**: tiempo de muestreo.

Este modelo fue propuesto por Naftali Chayat en el año de 1997 en una publicación denominada "IEEE P802.11-97-96".

En el modelo se propone que el tiempo de muestreo sea por lo menos cuatro veces menor al tiempo de duración de un símbolo, para determinar esta cantidad se utiliza la ecuación 7.

$$N_{trayectos} = 10 \cdot \frac{T_{rms}}{T_s} \quad \text{Ecuación (7)}$$

Donde:

- **T_{rms}**: retardo rms del canal.
- **N_{trayectos}**: número de trayectos (multitrayectos).

Debido a las cuantiosas diferencias entre el estándar 802.11b y el 802.11a, se hizo necesario realizar modificaciones al modelo de canal originalmente propuesto por Naftali. Las modificaciones que se llevaron a cabo tuvieron que ver con el tiempo de muestreo de la respuesta impulsiva del canal, en cuanto al número de muestras requeridas se mantuvo la ecuación para su determinación.

En el modelo actualizado se calcula el tiempo de muestreo como $T_{rms}/2$ si el retardo rms es menor a 100 ns, y un tiempo fijo de 50 ns en el caso de que el retardo sea mayor a 100 ns.

Parámetros:

- Canal (*channel*): Tipo cadena. "awgn" o "fading". Indica el tipo de canal que se va a simular.
- Dispersión de retardo (ns) (*Delay spread ns*): Tipo doble. Indica la dispersión del retardo del canal RMS.
- Número de caminos (*Number of paths*): Tipo entero. Indica el número de derivaciones en la línea de retardo tapped.
- Longitud total de la trama (*Total packet length*): Tipo entero. Indica la longitud de la trama que viajará a través del canal.

2.3. Sumador de complejos (Complex adder)

Este bloque suma dos señales complejas X y Y, las cuales están ubicadas en la entrada del bloque y el resultado de esta operación es colocado en la salida Z.

3. RECEPTOR

El receptor está compuesto por un conjunto de bloques que realizan la función ideal de un receptor, donde se asume perfecta sincronización con la trama recibida y conocimiento del canal, además recupera la trama PPDU y extrae de esta los siguientes campos: PLCP preámbulo, campo señal y campo de datos.

Parámetros:

- Velocidad de transmisión de datos Mbps (*Data rate*): Tipo entero. 18, 24, 36 ó 54. Indica la velocidad de transmisión de datos.
- Longitud del PSDU (*PSDU length*): Tipo entero. Indica el número de bits que contiene el PSDU en la trama.
- Retardo entre derivaciones (ns) (*Delay between taps (ns)*): Tipo doble. Indica el retardo entre las derivaciones de la línea de retardo en el modelo del canal.
- Decodificación (*Decoding*): Tipo cadena. "soft" ó "hard". Indica el modo de demodulación de las subportadoras.
- Número de caminos (*Number of paths*): Tipo entero. Indica el número de caminos que simula el canal.
- Bits de servicio (*Service bits*): Tipo entero. 16. Indica el número de bits de servicio definidos para la trama PPDU.
- Bits de cola (*Tail bits*): Tipo entero. 6. Indica el número de bits de cola.
- Bits de relleno (*Pad bits*): Tipo entero. Indica el número de bits de relleno y su valor depende del tipo de modulación y codificación que se utilice.
- Bits de datos (*Data bits*): Tipo entero. Indica el número de bits calculados para el campo datos.
- Modulación (*Modulation*): Tipo cadena. "BPSK", "QPSK", "16QAM" ó "64QAM". Indica la modulación utilizada en el campo datos, calculada según la velocidad de datos.
- Tasa de codificación (*Coding rate*): Tipo cadena. "1/2", "2/3" ó "3/4". Indica la tasa de codificación del código convolucional.
- N_{BPSK}: Tipo entero. 1, 2, 4 o 6. Indica el número de bits por subportadora, calculada a partir del tipo de modulación.
- N_{CBPS}: Tipo entero. Indica el número de bits codificados por símbolo OFDM, calculado a partir de la velocidad de transmisión de datos.
- N_{DBPS}: Tipo entero. Indica el número de bits de datos por símbolo OFDM, calculado a partir de la velocidad de transmisión de datos.
- Número de símbolos OFDM (*Number of OFDM symbols*): Tipo entero. Indica el número de símbolos OFDM que transportan los bits de datos en el PPDU.

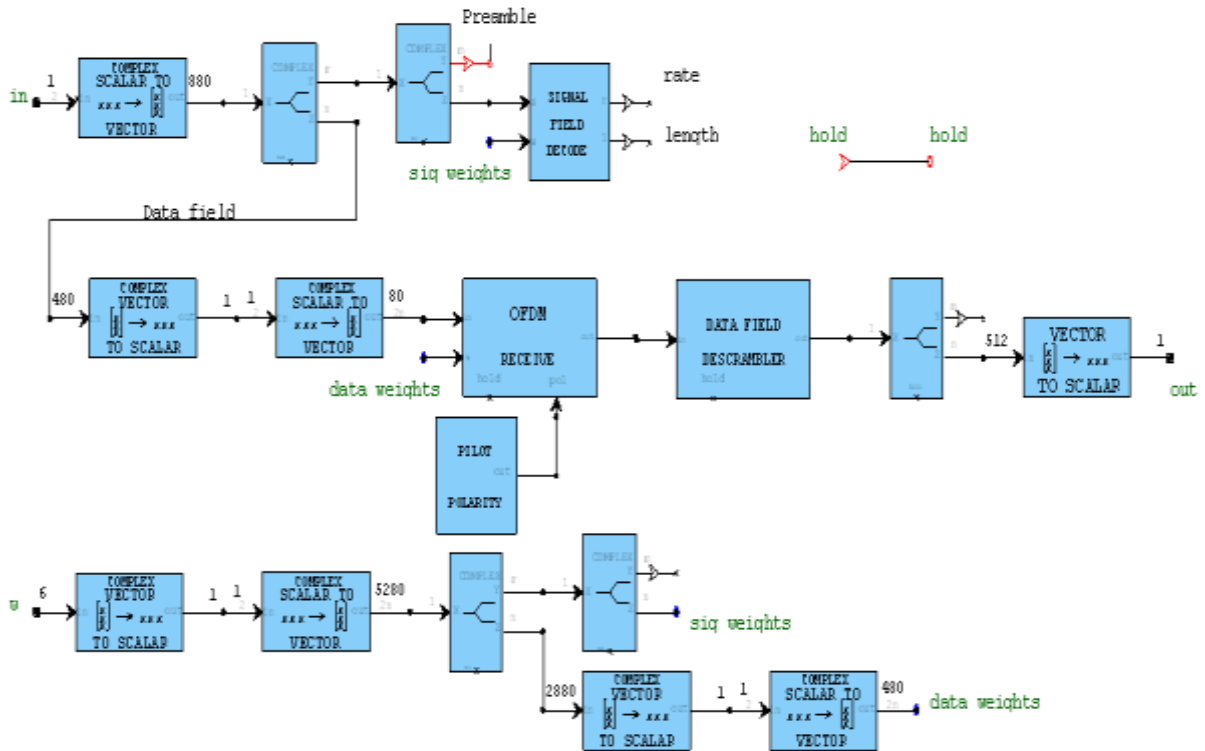


Figura 16. Subsistema receptor⁴.

3.1. Decodificador del campo señal (Signal field decode)

Este bloque está formado por otros bloques los cuales se encargan de realizar las siguientes tareas: decodificar el campo señal de la PPDU, demodulación OFDM, desmapeo y extracción de las señales piloto, demodulación de subportadoras desentrelazado.

Parámetros:

- Retardo entre derivaciones (ns) (*Delay between taps (ns)*): Tipo doble. Indica el retardo entre derivaciones en la línea de retardo tapped del modelo de canal.
- Número de trayectos (*Number of paths*): Tipo entero. Indica el número de caminos en la línea de retardo en el modelo de canal con desvanecimiento.
- Decodificación (*Decoding*): Tipo cadena. "soft" ó "hard". Indica el modo de demodulación para las subportadoras.
- N_{BPSK}: Tipo entero. 1, 2, 4 o 6. Indica el número de bits por subportadora.
- N_{CBPS}: Tipo entero. Indica el número de bits codificados por símbolo OFDM.

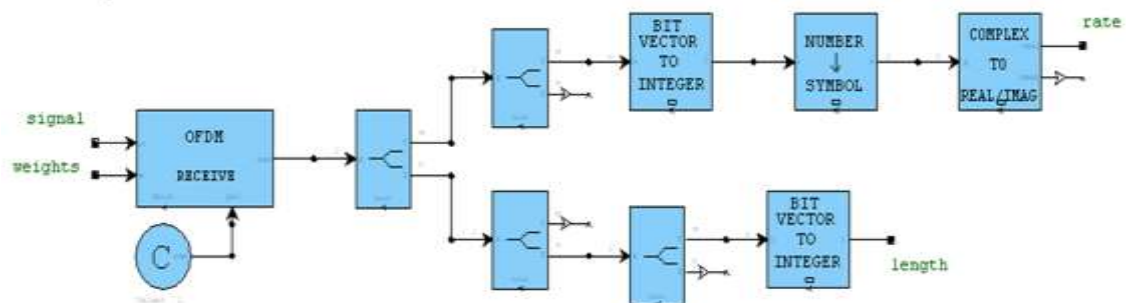


Figura 17. Bloque decodificador del campo señal⁵.

⁴ El conector marcado con w proviene de la conexión con el subsistema canal y es utilizado por el receptor para la estimación ideal del canal.

3.1.1. Receptor OFDM (Receive OFDM)

Este bloque es uno de los componentes principales para la implementación del receptor 802.11g, el cual realiza la decodificación del canal, desentrelazado de datos, la demodulación de subportadoras, extracción de las subportadoras piloto y el desmapeo de las subportadoras. Está compuesto por los siguientes bloques: Decodificador OFDM (*OFDM Decode*), Extracción de la señal piloto (*Remove pilot*), Desmapeo (*Demap*), Estimación de canal ideal (*Ideal Channel estimation*), Demodulador OFDM (*OFDM Demod*), Vector conjugado (*Vector Conj*) y Complejo (*Complex*) tal como lo indica la figura 18.

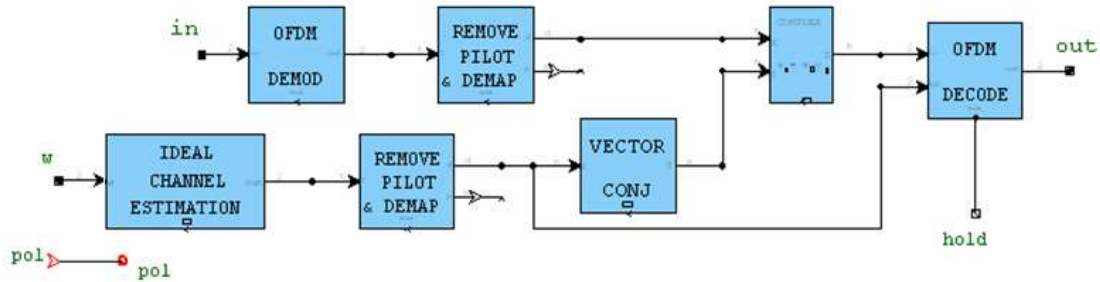


Figura 18. Bloque receptor OFDM.

3.1.1.1. Demodulador OFDM (OFDM Demod)

Este bloque remueve el intervalo de guarda de la forma de onda OFDM y realiza la Transformada Directa Rápida de Fourier. La forma de onda OFDM en el dominio del tiempo es transformada al dominio de la frecuencia en un vector de longitud 64 de acuerdo con la ecuación 10.

$$X(k) = \frac{1}{64} \sum_{n=0}^{63} x(n) e^{-\frac{j2\pi kn}{64}} \quad \text{Ecuación (10)}$$

Este bloque está conformado por los bloques: Complejo, Complejo Vector Escalar, Complejo FFT y Constante. Ver figura 19.

Parámetros:

- Longitud del prefijo cíclico (*Cyclic Prefix Length (ns)*): Tipo entero. 800 o 400. Indica la longitud del prefijo cíclico (ó intervalo de guarda) en ns.

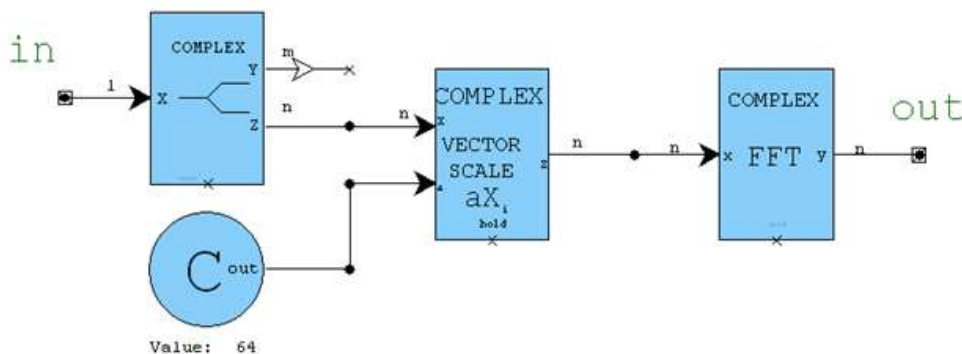


Figura 19. Bloque demodulador OFDM.

⁵ El conector signal recibe la señal de datos proveniente del canal y el conector weights hace referencia a la w anteriormente explicada en el pie de página 3.

3.1.1.2. Estimación ideal del canal (Ideal channel estimation)

Este bloque hace una estimación de los coeficientes del canal para un vector de longitud 64 de las subportadoras OFDM. Por cada símbolo OFDM, el bloque recibe 80 muestras en el tiempo, para cada uno de los pesos en la línea de retardo del canal (Las 80 muestras de diferentes pesos son concatenadas formando un vector). Por cada peso una muestra sale de las 80 y es usada para obtener la respuesta en frecuencia del canal sobre el ancho de banda del símbolo OFDM, utilizando una FFT de longitud 64. Los elementos de la respuesta en frecuencia del vector son las estimaciones de los coeficientes para el canal de 64 subportadoras en el símbolo OFDM.

Las derivaciones en el modelo de línea de retardo están igualmente separadas y el retardo entre dos derivaciones debe ser un entero múltiplo del inverso de la frecuencia de muestreo tal como lo indica la ecuación 11.

$$Td = \frac{k}{f_s} \quad \text{Ecuación (11)}$$

Donde:

- **Td:** tiempo de retardo entre dos derivaciones.
- **fs:** frecuencia de muestreo.
- **K:** número de derivaciones.

Si k es mayor que 1, entonces el muestreo se realiza para generar una respuesta correcta en frecuencia. Para que este bloque funcione correctamente debe tener los siguientes parámetros:

$$Td * fs * Ntrayectos = k * Ntrayectos < 64 \quad \text{Ecuación (12)}$$

Donde:

- **Ntrayectos:** número de trayectos.

Parámetros:

- Retardo entre derivaciones (*Delay taps*): Tipo doble. Indica los retardos entre derivaciones en la línea de retardo tapped en el modelo del canal.
- Longitud del prefijo cíclico (*Cyclic Prefix Length*): Tipo entero. 800 ns ó 400 ns. Indica longitud de prefijo cíclico (intervalo de guarda) en ns.
- Frecuencia de muestreo (Hz) (*Sampling frequency*): Tipo doble. Indica la frecuencia de muestreo para la señal.
- Número de caminos (*Number of paths*): Tipo entero. Indica el número de caminos en el modelo del canal.

3.1.1.3. Decodificador OFDM (OFDM Decode)

Este bloque realiza la demodulación de subportadoras, desentrelazado de de datos, deperforado y decodificación del canal. La estructura de este bloque se muestra en la figura 20.

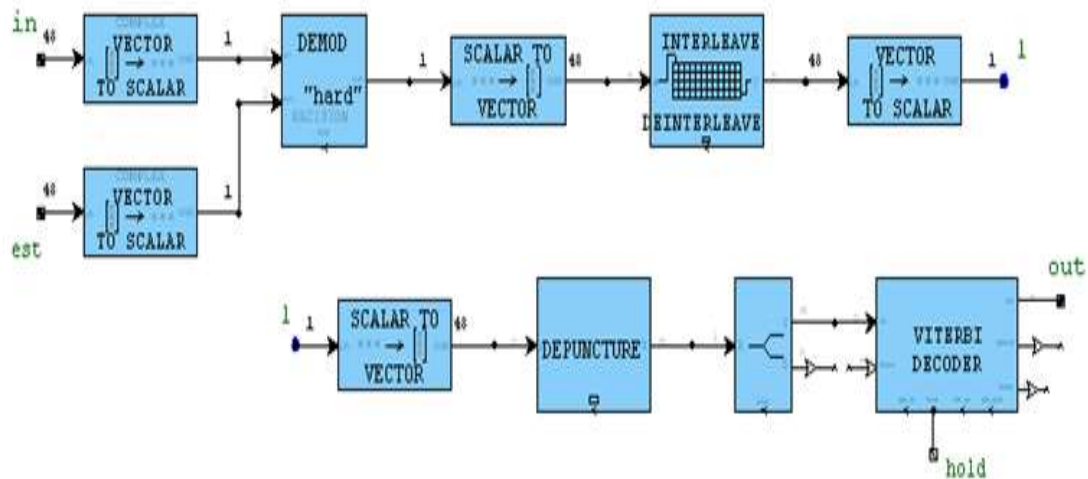


Figura 20. Bloque decodificador OFDM⁶.

Parámetros:

- Modulación (*Modulation*): Tipo cadena. Indica el tipo de modulación utilizada para el campo de Datos.
- Tasa de codificación (*Coding rate*): Tipo cadena. Indica la tasa de codificación.
- N_{BPSK}: Tipo entero. Indica el número de bits por subportadora.
- N_{CBPS}: Tipo entero. Indica el número de bits codificados por símbolo OFDM.
- Bits de relleno (*Pad bits*): Tipo entero. Indica número de bits insertados a los campos servicio, PSDU, y bits de cola para que el total de número de bits sea un múltiplo entero de N_{CBPS}.
- Campo de datos (*Data field*): Tipo entero. Indica el número total de bits en el campo de datos.
- Bits perforados (*Punctured bits*): Tipo entero. Indica el número total de bits codificados después del proceso de perforación.
- Campo decodificado (*Decoded field*): Tipo cadena. "data" ó "signal". Indica el campo decodificado por el bloque.
- Decodificación (Decoding): Tipo cadena. "soft" ó "hard". Indica el modo de demodulación de las subportadoras en el demodulador.

3.1.1.3.1. Demodulación dura (Demod "hard")

En este bloque jerárquico, cada subportadora se demodula sobre. Siendo **r** el símbolo recibido en una determinada subportadora.

$$\mathbf{r} = \mathbf{a}\mathbf{s} + \mathbf{n} \quad \text{Ecuación (13)}$$

Donde:

- **a** es el canal estimado.
- **s** el símbolo transmitido.
- **n** es el ruido térmico.
- **r** símbolo recibido.

Parámetros:

- Modulación (*Modulation*): Tipo cadena. Indica el tipo de modulación utilizada para el campo de Datos.

⁶ El conector est proviene del bloque estimación ideal del canal y representa las estimaciones de los coeficientes para el canal de 64 subportadoras en el símbolo OFDM.

- N_{BPSC} : Tipo entero. Indica el número de bits por subportadora.
- Decodificación (*Decoding*): Tipo cadena. "soft" ó "hard". Indica el modo de demodulación de las subportadoras en el demodulador.

3.1.1.3.2. Desperforado (*Depuncture*)

Este bloque realiza el proceso inverso al perforado (*puncturing*). Este bloque inserta algunos bits cero en el lugar de los bits que fueron omitidos en el proceso de perforado en la transmisión. El patrón de perforado debe ser idéntico al utilizado en el bloque de perforado

Parámetros:

- Patrón de perforado (*Puncturing pattern*): Tipo cadena. Indica un patrón de ceros y unos, los ceros representan los bits omitidos en la salida del codificador y los unos representan los bits que son retenidos.

3.1.1.3.3. Decodificador Viterbi (*Viterbi decoder*)

Este bloque implementa un decodificador general Viterbi con una tasa K/N y un codificador convolucional de longitud constante L con polinomios generadores arbitrarios e interrupción arbitraria de la longitud del camino.

Parámetros:

- Tamaño del vector de entrada (*input vector size*): Tipo entero mayor que 0. Indica el número de componentes en el vector de entrada.
- Longitud de la trama de información K_0 (*Information frame length (K_0)*): Tipo entero mayor que 0. Indica el número de bits de entrada por símbolo.
- Longitud de la trama a transmitir N_0 (*Transmit frame length (N_0)*): Tipo entero mayor que 0. Indica el número de bits de salida por símbolo.
- Restricción de longitud (*constraint length L*): Tipo entero mayor que 0. Indica el orden del polinomio utilizado en el código convolucional.
- Longitud del camino truncado (*Truncation path length*): Tipo entero mayor que 1. Indica la longitud del camino truncado utilizada para el decodificador de Viterbi.
- Generador del archivo de entrada (*Generator input file*): Tipo cadena. "nombre_de_archivo". Especifica el nombre del archivo de texto que contiene los parámetros y el generador de polinomios para el codificador, la primera línea del archivo contiene los valores de N , K y L . Cada una de las líneas restantes especifican un polinomio generador en forma octal.
- Modo (*Mode*): Tipo cadena. "Continuous", "TailBits" ó "TailBiting". Especifica uno de los tres modos de codificación los cuales pueden ser "Continuous", "TailBits" ó "TailBiting", para el modo "Continuous" los datos de los vectores se consideran como un flujo continuo de datos, para el modo "TailBits" el codificador es restablecido a un estado conocido especificado por los bits de cola utilizados en los parámetros del codificador antes de la codificación de cada uno de los vectores. Para el modo "TailBiting" el estado del codificador es inicializado con los primeros $(L-1)$ elementos del vector de entrada.
- Trama de sincronización de retardo para el modo continuo (*Frame sync delay for continuous mode*): Tipo entero. Indica el número de símbolos de retardo necesarios para la sincronización de la trama.

3.1.1.4. Desmapeo y remoción del piloto (Remove pilot y demap)

En este bloque el vector de 64 subportadoras OFDM es desmapeado a un vector de 52 subportadoras (48 de información y 4 subportadoras piloto).

Parámetros:
Ninguno.

3.1.1.5. Vector conjugado (vector conj)

Este bloque realiza la conjugación del vector complejo de la señal. Es decir, la parte imaginaria de los vectores complejos de entrada se niega con el fin de generar el producto vectorial.

Parámetros:

- Tamaño del vector (*vector size*): Tipo entero mayor que 0. Indica el número de componentes a la entrada del vector.

3.1.2. Vector bit a entero (Bit vector to integer)

Este bloque convierte un flujo de vectores bit en un flujo de enteros. Convierte el entero a doble y ubica el resultado en la salida del bloque. Los parámetros del bloque permiten fijar el número de bits por entero y ordenarlo (el bit más significativo en primer lugar)

Por ejemplo considere la siguiente secuencia de valores de vectores de entrada:

[010] [110] [001] [011] [111]...

Con el número de bits fijado en tres y el orden de MSB (el bit más significativo en primer lugar), la salida de la secuencia es:

2 6 1 3 7...

Parámetros:

- Tamaño del vector (*vector size*): Tipo entero. 1, 2,... ó 32. Indica el número de bits utilizado para representar el valor de salida.
- Orden de bit (*Bit ordering MSB o LSB*): Tipo cadena. "MSB" ó "LSB". Indica el orden en el cual son colocados los bits en la entrada del vector.

3.1.3. Complejo a real imaginario (Complex to real_imag)

Este bloque separa la parte real de la imaginaria de un vector complejo de entrada.

Parámetros:

- Ninguno.

3.2. Codificador del campo datos (Data fiel descrambler)

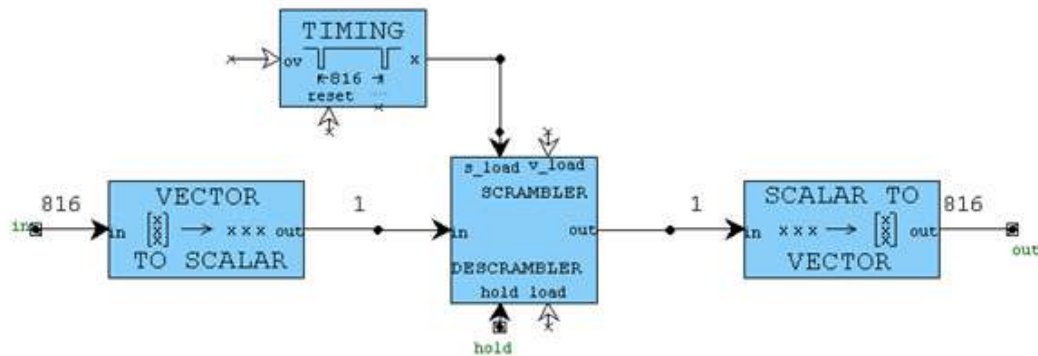


Figura 21. Bloque codificador del campo de datos.

Este bloque se encarga de decodificar los bits recibidos.

Parámetros:

- Bits de servicio (service bits): Tipo entero. Indica el número de bits en el campo servicio.
- Longitud del PSDU (*PSDU length*): Tipo entero. Indica el número de bytes que contiene el PSDU que hace parte de la trama.

4. BLER/BER

Este bloque calcula la tasa de error de bit (BER) y la tasa de error de bloque (BLER, Block Error Rate) entre la señal transmitida y la señal recibida. Además este bloque puede reportar un intervalo de confianza y opcionalmente detiene la simulación una vez que la evaluación alcance cierta calidad.

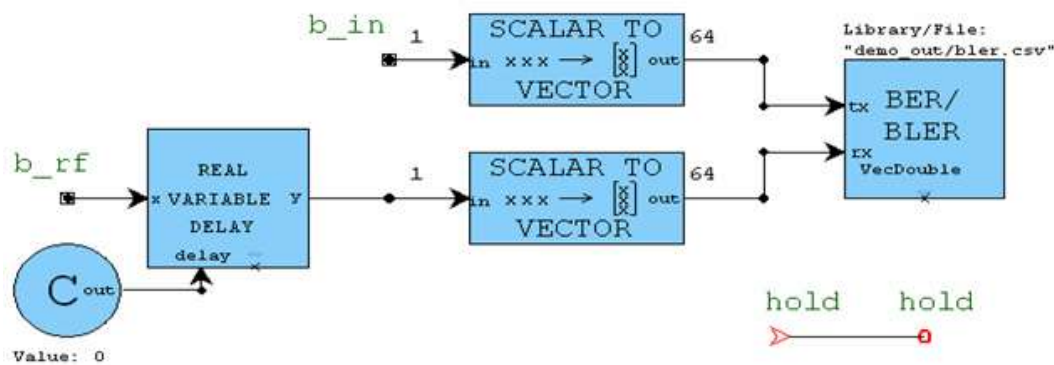


Figura 22. Bloque BER/BLER.

El BER es calculado como el número total de elementos diferentes en el tiempo, dividido por el número total de elementos observados todo el tiempo. El BLER es calculado como el número de bloques de información con alguna diferencia durante todo el tiempo dividido sobre el número de bloques de información observados todo el tiempo.

Parámetros:

- Tipo de señal (*Typename*): Especifica si el tipo de señales a comparar son vectores o valores escalares. En este caso se fijó en VecDouble, es decir que las señales son de tipo vector para calcular el BLER.
- Ruta (Path): Tipo cadena. Especifica la ruta del archivo donde serán almacenados los resultados del BER/BLER. En este caso "demo_out/bler.csv"
- Número de entradas a ignorar (*Number of inputs to ignore*): Tipo entero. Indica el número de entradas a ignorar antes de que empiece el conteo. En este caso se deja en 1 que es el valor por defecto del bloque.
- Texto para agregar a los títulos de los resultados (*Text to add to results titles*): Tipo cadena. Especifica el texto que se quiere agregar a las etiquetas de los resultados.
- Almacenar resultados de BER (*Store BER results*): Tipo cadena. "yes" ó "no". Indica si se quieren almacenar los resultados del BER.
- Almacenar resultados de BLER (*Store BLER results*): Tipo cadena. "yes" ó "no". Indica si se quieren almacenar los resultados del BLER.
- Ignorar la desigualdad de tamaño (*Ignore BER/BLER on size mismatch*): Tipo cadena. "yes" ó "no". Si se fija en "yes", los bloques de datos que superan una tasa de error de decisión, serán ignorados en los cálculos del BER y el BLER.
- Prueba de confianza (*Confidence test*): Tipo cadena. Indica la prueba de confianza a realizar. Los tipos de pruebas son "BER bound", "BLER bound", "BER interval" ó "BLER interval". Las pruebas de intervalo definen el intervalo de confianza de la tasa de error asociada y las pruebas de límites definen el límite de confianza de la tasa de error asociada.
- Porcentaje de confianza (*Confidence percentage*): Tipo entero. Indica el porcentaje del nivel de confianza el cual puede tomar valores entre 90, 95 ó 99.
- Informe de resultados de confianza (*Report confidence results*): Tipo cadena. "yes" ó "no". Indica si se desea conocer el límite o el intervalo de confianza medido.
- Modo de parada automático (*Auto stop mode*): Tipo cadena. "disabled", "confidence test", "block errors" ó "bit errors". Especifica el tipo de prueba que se utilizará para detener la simulación.
- Máximo error contado (*Error count max*): Tipo entero. Indica el número de errores que hayan sido observados antes de parar la simulación si en el parámetro "Auto stop mode" esta seleccionada la opción "block errors" ó "bit errors".
- Objetivo para la prueba de límite (*Target for bound test*): Tipo doble. Si en el parámetro "Auto stop mode" el modo está fijado como "confidence test" y este a su vez está fijado como límite para la BER o el BLER. La simulación se detendrá una vez que el intervalo de confianza se encuentre en el rango del valor estimado.