

ANEXOS

Factibilidad Tecnológica de la Implementación de un Sistema de Acceso Múltiple al Satélite basado en OFDMA utilizando TDM



**Luis Alberto Aguirre Romo
Luis Alberto Rodríguez Rodríguez**

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Popayán
2010**

ANEXOS

Factibilidad Tecnológica de la Implementación de un Sistema de Acceso Múltiple al Satélite basado en OFDMA utilizando TDM



**Luis Alberto Aguirre Romo
Luis Alberto Rodríguez Rodríguez**

**Documento final de trabajo grado presentado como requisito
para optar al título de Ingeniero en Electrónica y Telecomunicaciones**

**Director
ING. MÁG. HAROLD ROMO ROMERO**

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Popayán
2010**

ANEXO A

Tabla de contenido

LISTA DE FIGURAS	1
ANEXO A. DESCRIPCIÓN DE LOS BLOQUES DE SIMULINK UTILIZADOS EN LA SIMULACION DE UN SISTEMA DE ACCESO MÚLTIPLE AL SATÉLITE BASADO EN OFDMA UTILIZANDO TDM.....	2
1. INTRODUCCIÓN	2
2. DESCRIPCIÓN DE LOS BLOQUES DE SIMULINK.....	2
2.1. Fuente Binaria de Bernoulli	2
2.2. Conversor de Bit a Entero	3
2.3. Scrambler o aleatorizador	4
2.4. Modulador Digital	6
2.5. Herramienta FDATool para diseño de Filtros	7
2.6. Transformada de Fourier.....	9
2.6.1. PAD.....	9
2.6.2. IFFT.....	10
2.6.3. FFT	11
2.6.4. Bloque Selector de datos	11
2.7. CANAL AWGN	12

LISTA DE FIGURAS

Figura 1 Generador de Bernoulli	3
Figura 2 Conversor de bit a entero	4
Figura 3 Proceso de aleatorización.....	5
Figura 4 Aleatorizador (Scrambler).....	5
Figura 5 Modulador digital	6
Figura 6 Configuración del modulador digital	7
Figura 7 FDATool para diseño de filtros.....	8
Figura 8: Tipos de respuesta de filtros	8
Figura 9: Métodos de diseño de filtros	9
Figura 10: Bloque de relleno (PAD)	9
Figura 11: Bloque IFFT	10
Figura 12: Bloque FFT	11
Figura 13: Selector de datos	12
Figura 14: Canal AWGN	13

ANEXO A. DESCRIPCIÓN DE LOS BLOQUES DE SIMULINK UTILIZADOS EN LA SIMULACIÓN DE UN SISTEMA DE ACCESO MÚLTIPLE AL SATÉLITE BASADO EN OFDMA UTILIZANDO TDM

1. INTRODUCCIÓN

En el presente documento se expondrá el funcionamiento de los bloques más importantes utilizados en la simulación de los sistemas de acceso múltiple FDMA y OFDMA, la cual es el resultado del trabajo de grado titulado “**Factibilidad Tecnológica de la Implementación de un Sistema de Acceso Múltiple al Satélite basado en OFDMA utilizando TDM**” desarrollado por los estudiantes Luis Alberto Aguirre Romo y Luis Alberto Rodríguez Rodríguez. Este manual ayuda a comprender la forma de operación de dichos bloques aplicados a la simulación de un sistema de acceso múltiple por división de frecuencia siguiendo los parámetros de configuración propios de la herramienta Simulink® de MATLAB® versión r2008a.

2. DESCRIPCIÓN DE LOS BLOQUES DE SIMULINK

2.1. Fuente Binaria de Bernoulli.

Es la fuente básica generadora de los datos de los clientes. Se configura con los siguientes valores:

- *Probability of a zero*: sirve para controlar la aparición de un cero o un uno para lo cual se escoge una probabilidad entre 0 y 1 para la generación de un cero. Se configura con un valor de 0.5 como valor equiprobable por defecto.
- *Initial seed*: en este campo se introduce el primer valor binario que genera la fuente en su equivalente decimal.
- *Sample Time*: se introduce el tiempo de generación de cada muestra en segundos.
- *Frame-based outputs*: los datos se pueden generar en forma de muestras o tramas, para este último caso se debe marcar esta casilla.
- *Samples per frame*: en caso de haber marcado la casilla *Frame-based outputs* se habilita este campo para escoger cuantas muestras contiene cada trama generada.
- *Output data type*: se selecciona el tipo de dato de salida entre los que se puede observar en la figura 1.

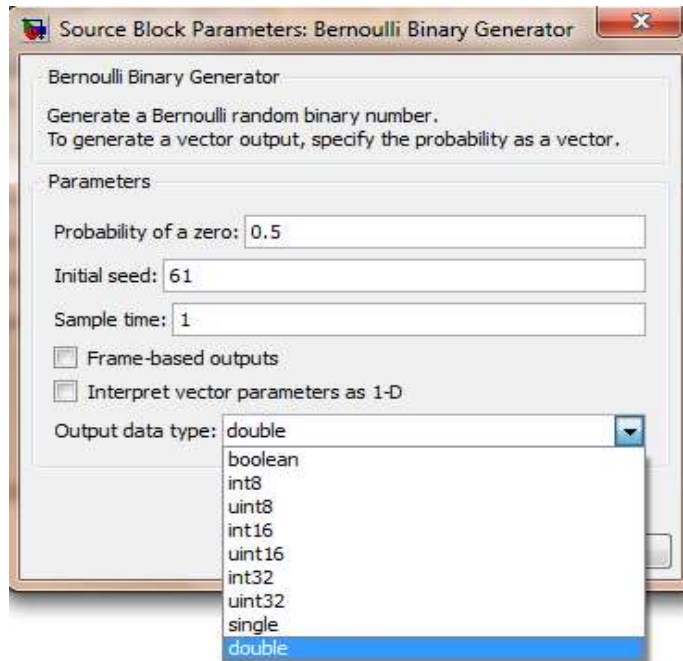


Figura 1. Generador de Bernoulli.

2.2. Conversor de Bit a Entero.

Este bloque toma los datos binarios generados en serie por la fuente de Bernoulli y los convierte en valores enteros en paralelo por lo cual se le denomina conversor serie-paralelo. La ventana de configuración del bloque se muestra en la figura 2. Los campos de configuración son los siguientes:

- *Number of bits per integer:* Especifica el número de bits que conforman cada bloque entero de salida. Cada bloque tiene un valor entero equivalente al valor binario que contiene. El numero de bits que se puede introducir para cada bloque se configura de acuerdo con el esquema de modulación a utilizar, por ejemplo 2 para QPSK o 4 para 16-QAM y el rango de valores validos está entre 1 y 32.
- *Input bit order:* El orden de entrada de los bits se puede escoger de tal manera que el primero de ellos sea el más significativo (MSB first) o el menos significativo (LSB first). Por convención general se ha configurado la opción MSB first.
- *After bit packing, treat resulting integers values as:* en este campo se selecciona la opción sin signo (Unsigned) o con signo (Signed) para el dato de salida.
- *Output data type:* se selecciona el tipo de dato de salida considerando las opciones que se observa en la figura donde la primera (Inherit via internal rule) indica que se entrega el mismo tipo de dato de la entrada.

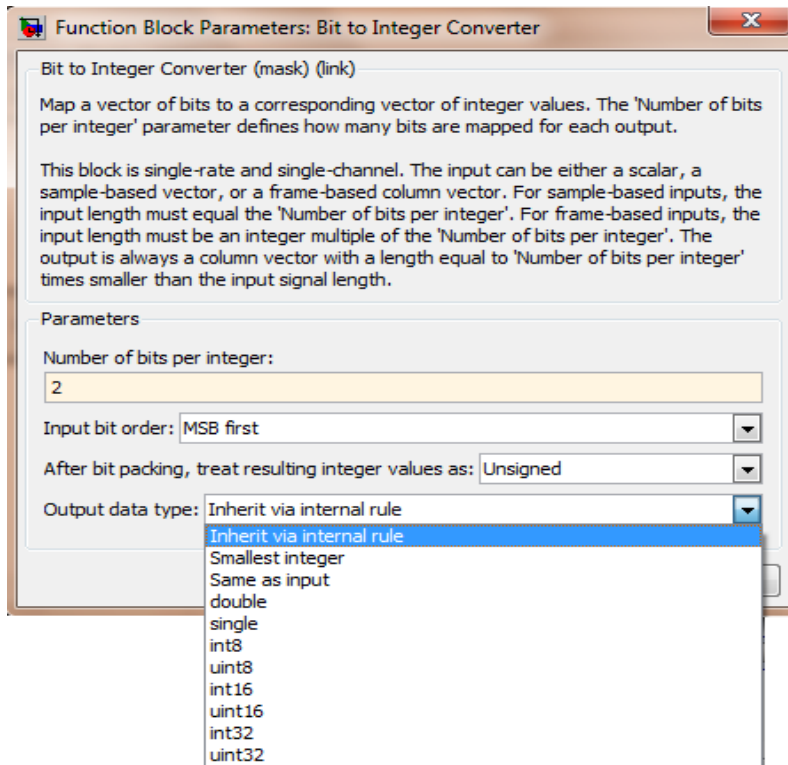


Figura 2: Conversor de bit a entero

2.3. Scrambler o aleatorizador.

Este bloque codifica los datos de entrada mediante un polinomio de aleatorización representado por la expresión 1.

$$1 + P_1Z^{-1} + P_2Z^{-2} + \dots + P_{m-1}Z^{-(m-1)} + P_mZ^{-m} \quad (1)$$

P toma los valores 0 o 1 representando una conexión abierta o cerrada respectivamente en el esquema de la figura 3, de tal forma que el polinomio se puede escribir utilizando tales valores por ejemplo [1 0 1 1], o mediante potencias descendentes de Z es decir [0 -2 -3]. Los dos ejemplos representan el polinomio $1 + Z^{-2} + Z^{-3}$.

Cada dato de entrada es un numero entero que es procesa de acuerdo con el polinomio (1) dando como resultado un corrimiento secuencial de los registros (1, 2, M-1, M en la figura 3) en cada operación.

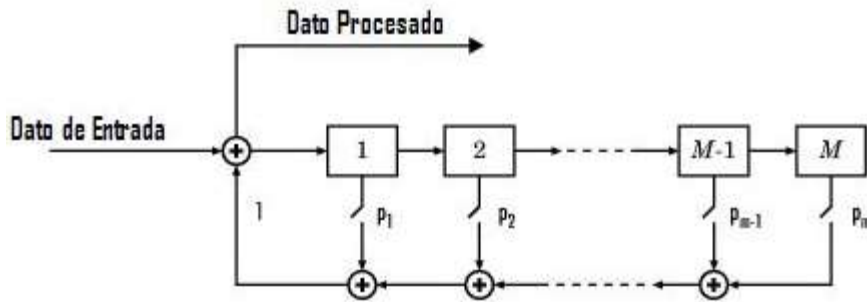


Figura 3: Proceso de aleatorización

La configuración de este bloque depende de la modulación utilizada ya que debe coincidir con el número de bits que ella utilice. El polinomio de aleatorización contiene, además de los valores (P) que determinan las conexiones del circuito de la figura 3, los estados iniciales del aleatorizador que corresponden a los valores que tienen sus registros al comenzar la simulación.

La ventana de configuración del bloque se puede observar en la figura 4 y tiene los siguientes campos de configuración:

- *Calculation base:* en este campo se introduce el número total de datos que entran al bloque, para QPSK el valor es 4 y para 16-QAM es 16.
- *Scramble Polynomial:* aquí se introducen los coeficientes (P_i) del polinomio de aleatorización descrito anteriormente y representado por la expresión 1 y la figura 3.

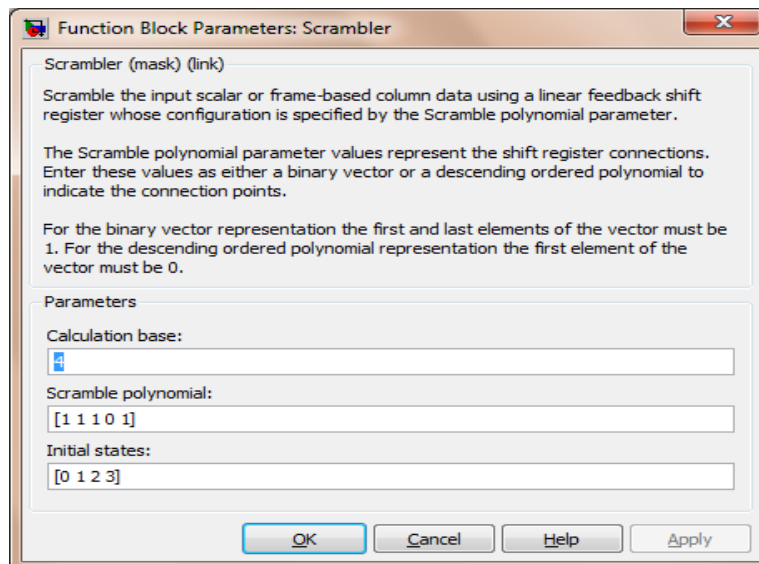


Figura 4: Aleatorizador (Scrambler)

2.4. Modulador Digital.

Se utilizan dos tipos de moduladores uno para QPSK y otro para 16-QAM que sirve para cualquier modulación tipo QAM, cuyas ventanas de configuración se muestran en las figuras 5 y 6 respectivamente.

- *Input type (QPSK)*: en este campo se selecciona el tipo de dato de entrada entre binario o entero.
- *Constellation ordering (QPSK)*: se escoge, entre codificación Gray o Binaria, el ordenamiento de los datos o tipo de mapeo que se aplicara a los datos según se necesite. Este ordenamiento se puede visualizar en un diagrama de constelación.
- *Phase offset (rad) (QPSK)*: se introduce la fase de los puntos de la constelación, en otras palabras, el ángulo en radianes en el cual se ubicaran los puntos en los cuadrantes del plano I-Q.
- *Output data type (QPSK y QAM)*: se selecciona el tipo de dato de salida entre las opciones que se pueden observar en la figura 5 y 6, donde la opción *Inherit via back propagation* indica que el tipo dato de salida conserva sus características de entrada.

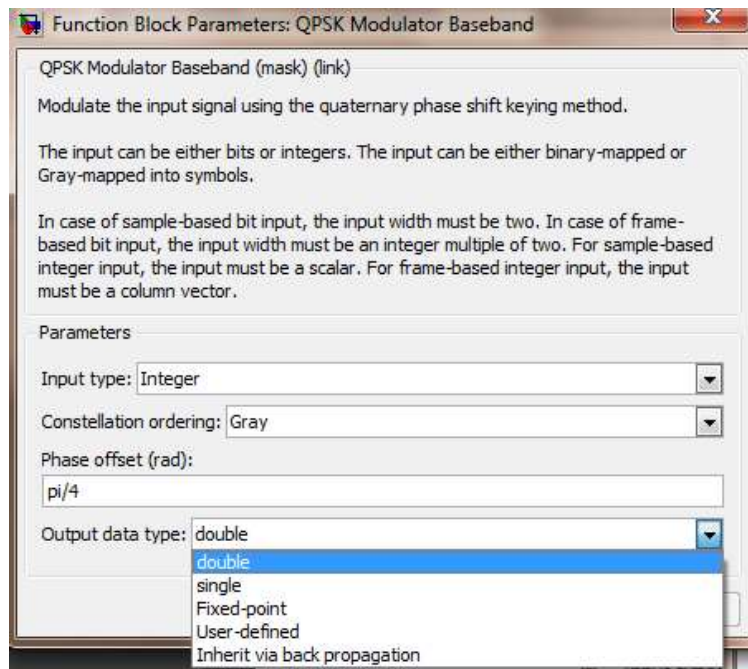


Figura 5: Modulador digital

- *Signal constellation (QAM)*: en este campo se ingresa la función o variable que representa la constelación M-QAM que se desee utilizar, para este caso se utiliza

una variable que representa una constelación 16-QAM normalizada definida en el script de Matlab que complementa la simulación.

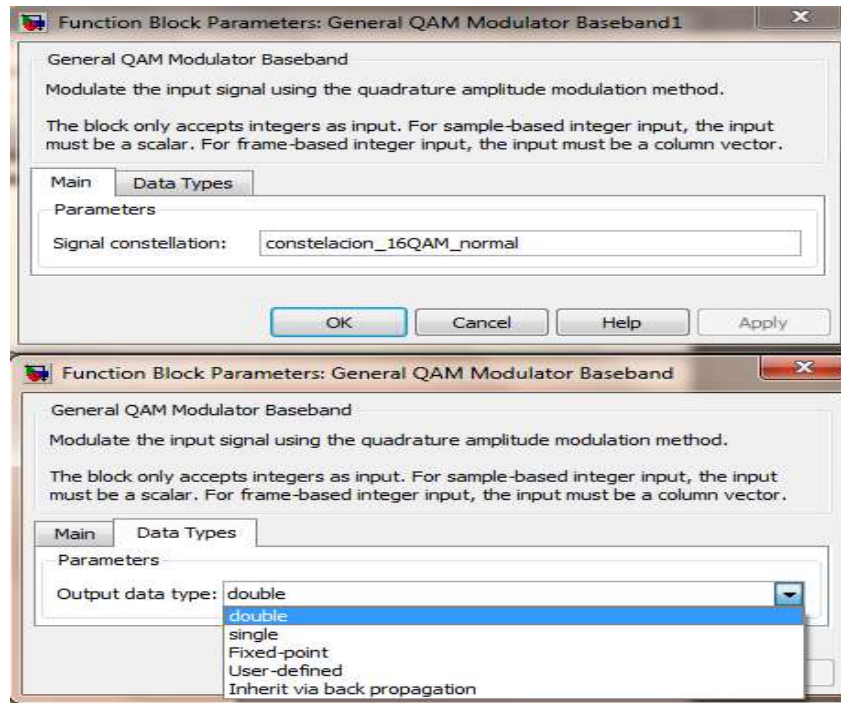


Figura 6: Configuración del modulador digital

2.5. Herramienta FDATool para diseño de Filtros.

Para el diseño de todos los filtros utilizados en la simulación se empleó la herramienta FDATool de matlab, en la cual se introduce los parámetros de diseño básicos para construir cualquier tipo de filtro. La herramienta se habilita ejecutando la orden *fdatool* en la línea de comandos de la ventana principal de matlab. En la figura 7 se observa la ventana principal de la aplicación y con sus siguientes secciones de configuración.

- *Response Type*: aquí se puede seleccionar varios tipos de filtros, estos se pueden observar en la figura 8. De esta sección, para la simulación se utilizó solamente diseños con filtros pasabanda y pasabajas.
- *Design Method*: se escoge la clase de filtro a utilizar dentro de las opciones de respuesta infinita al impulso (IIR) o de respuesta finita al impulso (FIR) que aparecen en la figura 9.
- *Filter Order*: se especifica el orden del filtro a utilizar. En todos los diseños se escogió la opción *minimum order*, la cual logra el mínimo orden posible para el diseño propuesto.

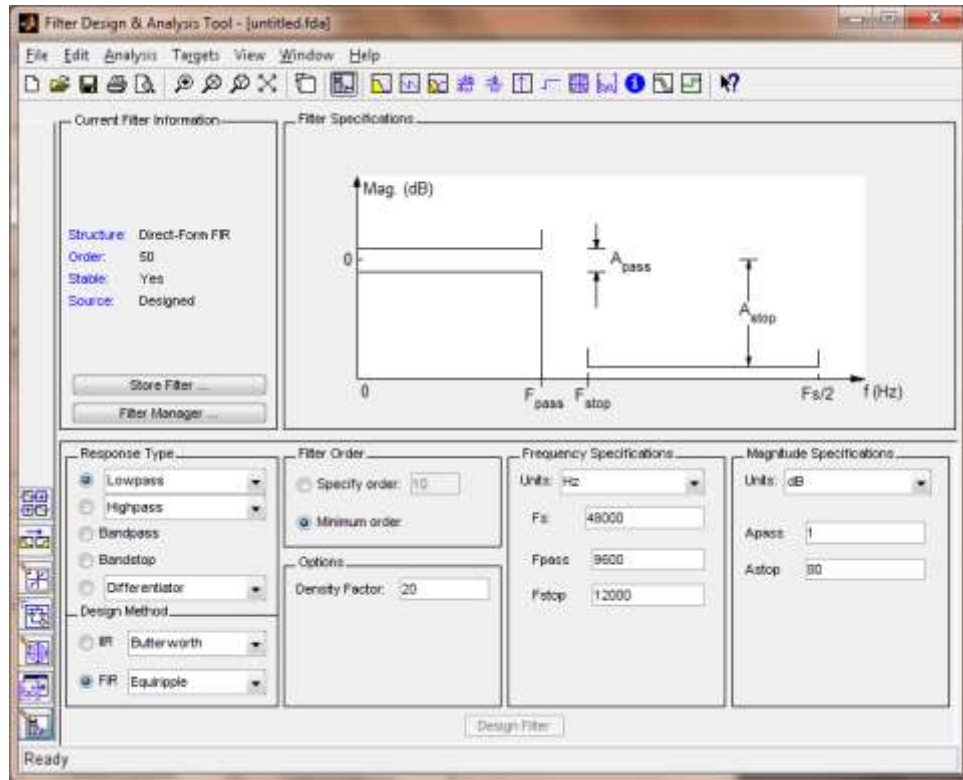


Figura 7: FDATool para diseño de filtros

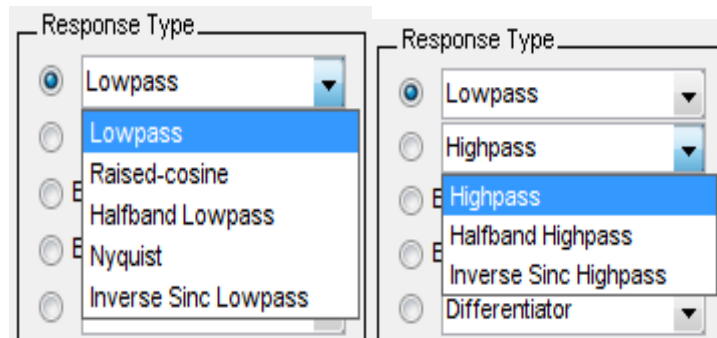


Figura 8: Tipos de respuesta de filtros

- *Frequency Specifications*: esta configuración clave ya que ella se introduce los valores de frecuencias de paso o de corte para el diseño del filtro, especificando las unidades adecuadas para cada caso.
- *Magnitude Specifications*: este campo también es importante dado que se introduce el nivel de atenuación deseado para la respuesta del filtro especificando unidades lineales (*linear*) o logarítmicas (dB).

- *Filter Specifications*: en esta sección se muestra la grafica de la respuesta en magnitud o fase del filtro diseñado.

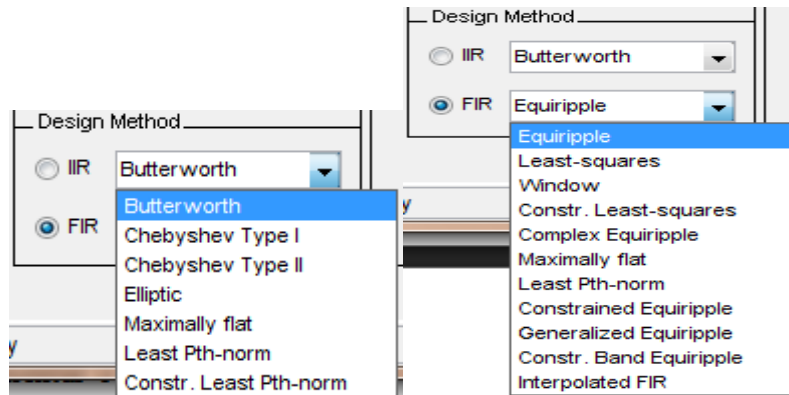


Figura 9: Métodos de diseño de filtros

2.6. Transformada de Fourier.

2.6.1. Bloque PAD.

Se encarga de aumentar o reducir el tamaño en bits de la señal de entrada desde el modulador digital, rellenando o suprimiendo bits de las filas, las columnas o de ambas dimensiones simultáneamente cuando se hace necesario tener en la salida un valor en potencias de 2 predeterminado.

Este bloque se configura básicamente introduciendo un valor entero de relleno y el valor binario hasta el cual es necesario rellenar sea las filas o las columnas. Su ventana de configuración se puede observar en la figura 10.

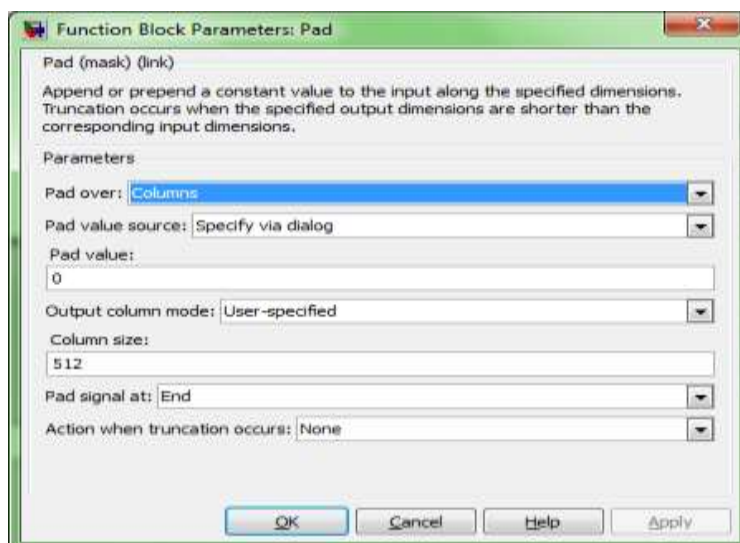


Figura 10: Bloque de relleno (PAD)

2.6.2. Bloque IFFT.

Este bloque calcula la transformada inversa rápida de Fourier (IFFT) de una entrada u conformada por varios canales de longitud P . Se puede seleccionar si se desarrolla la transformada tomando como longitud directamente el número de entradas o escogiendo la potencia de 2 requerida. El cálculo se realiza mediante la transformada inversa discreta de Fourier (IDFT) utilizando básicamente los algoritmos: Mariposa, Radix 2 y Bit inverso.

Las opciones para configurar este bloque se muestran en la figura 11.

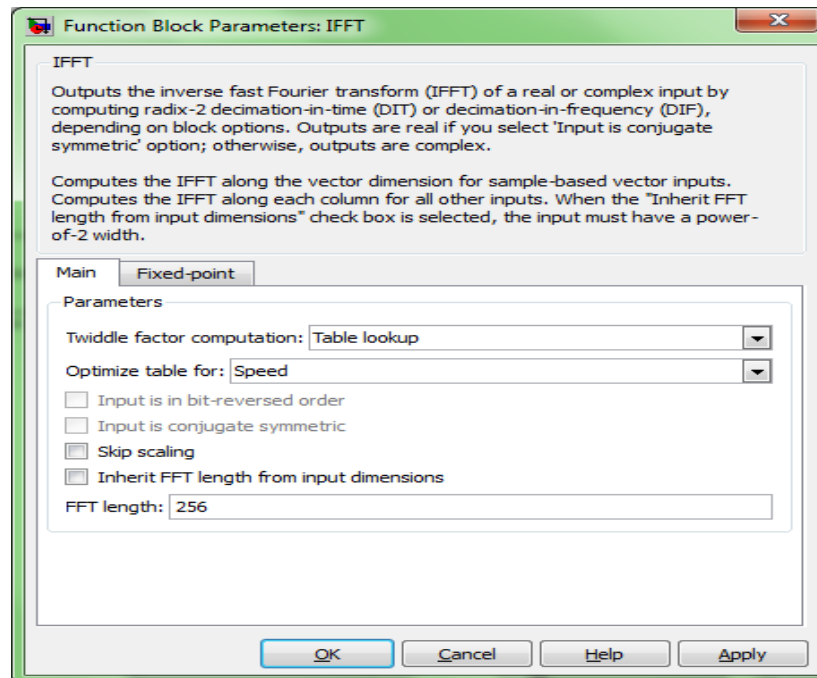


Figura 11: Bloque IFFT

Las opciones que generalmente se utilizan son:

- *Input is in bit-reversed order*: esta opción permite escoger si se aplica el algoritmo de bit inverso de acuerdo a la organización de los bits de entrada.
- *Input is conjugate symmetric*: con esta opción se elige el tipo de dato de salida, si es real, se marca la casilla, en caso contrario se deja vacía para obtener una salida compleja.
- *Inherit FFT length from input dimensions*: al seleccionarla, la longitud de la transformada corresponde al número de entradas.
- *FFT length*: este campo aparece solamente si no se selecciona la opción “*Inherit FFT length from input dimensions*” y se utiliza para escoger manualmente la longitud de la transformada introduciendo un valor en potencia de 2.

2.6.3. Bloque FFT

Este bloque calcula la transformada rápida de Fourier (FFT) utilizando los mismos mecanismos que los de su bloque inverso IFFT. Las opciones utilizadas en este bloque: “*Inherit FFT length from input dimensions*” y “*FFT length*” cumplen las mismas funciones que las de su bloque inverso.

Su ventana de configuración es básicamente la misma que la del bloque IFFT a excepción de que no requiere configurar el tipo de dato de entrada y salida pues procesa automáticamente entradas de tipo real o complejo. A diferencia de su bloque inverso IFFT, este presenta la siguiente opción:

- *Output is in bit-reversed order*: se selecciona la aplicación del algoritmo de bit inverso para organizar los datos de salida de acuerdo con dicho algoritmo.

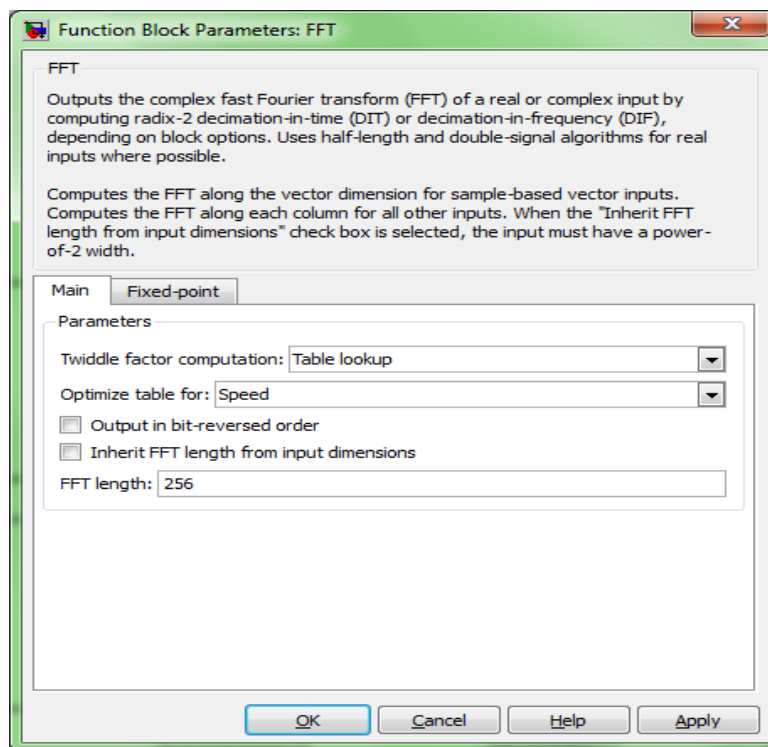


Figura 12: Bloque FFT

2.6.4. Bloque Selector de datos.

Este bloque permite seleccionar una fracción de los datos de entrada (vector o matriz) cuando no es necesario trabajar con todos ellos. Se escoge una longitud menor o igual al número de datos de entrada introduciendo en el bloque una sentencia de la forma:

$$[A:B:C] \quad (3)$$

Donde A representa el punto inicial del vector a seleccionar, B representa con qué intervalo se hace la selección y C representa hasta que punto del vector se desea tomar. En caso de que B no aparezca es porque se toman intervalos de 1.

En la figura 13 se observa su ventana de configuración con un ejemplo donde se toman desde 1 hasta 512 puntos en intervalos de 1 con un tamaño total de 512.

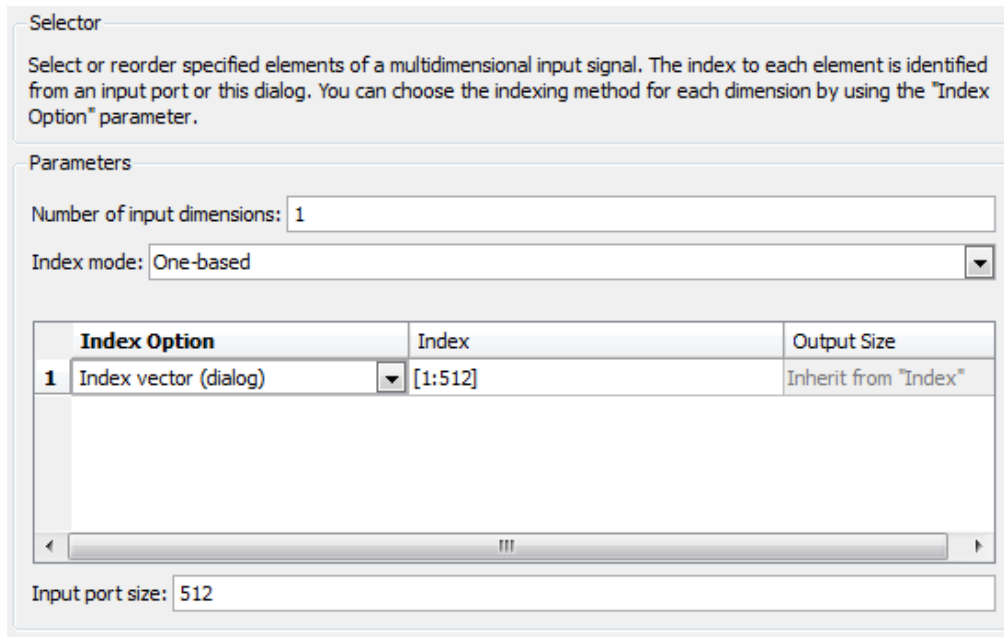


Figura 13: Selector de datos.

2.7. CANAL AWGN.

Este bloque representa el canal de transmisión que introduce ruido blanco gaussiano aditivo. Utiliza los siguientes parámetros para su configuración, los cuales son mostrados en la figura 14:

- *Relación de energía de bit a densidad espectral de potencia de ruido E_b/N_0 .*
- *Número de bits por símbolo.*
- *Potencia de la señal de entrada para símbolos o muestras.*
- *Período de símbolo T_{sym} .*
- *Relación de energía de la señal a densidad espectral de potencia de ruido E_s/N_0 .*
- *Relación de potencia de la señal a potencia de ruido SNR.*

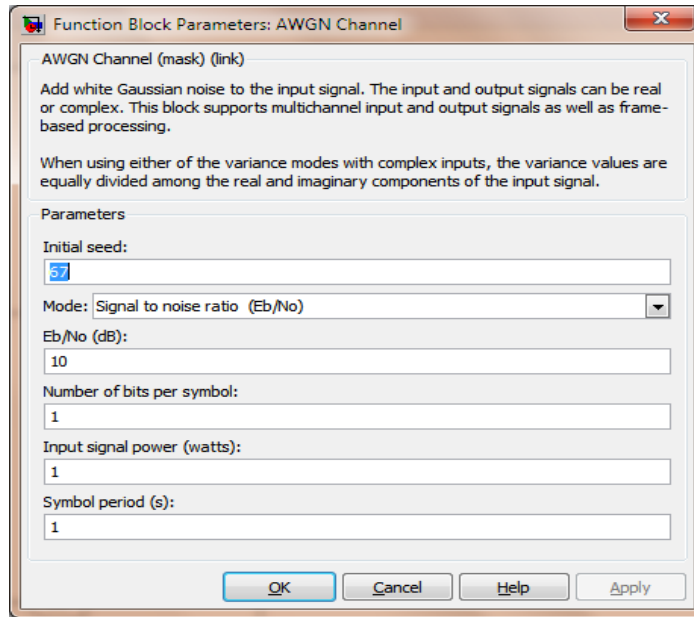


Figura 14: Canal AWGN.