

ANÁLISIS DEL FUNCIONAMIENTO Y DESEMPEÑO DEL CÓDIGO DE ALAMOUTI EN SISTEMAS MIMO



ANEXO A

“LIBRERÍA DE LabVIEW PARA LA SIMULACIÓN DE ENTORNOS DE COMUNICACIONES CON CODIFICACIÓN ALAMOUTI”

UNIVERSIDAD DEL CAUCA

FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES

DEPARTAMENTO DE TELECOMUNICACIONES

POPAYÁN

2008

TABLA DE CONTENIDO		Pág.
1.	INTRODUCCIÓN	4
2.	GENERALIDADES DE LA HERRAMIENTA LabVIEW	5
2.1	Definición de LabVIEW	5
2.2	Aplicaciones de LabVIEW	5
2.3	Programación gráfica con LabVIEW	5
2.4	Diseño de una interfaz de usuario a partir del código.....	6
2.5	El entorno LabVIEW	7
2.6	Flujo de datos	10
3.	DESCRIPCIÓN DE LOS MÓDULOS GENERADOS PARA LA SIMULACIÓN DE LA CODIFICACIÓN ALAMOUTI.....	11
3.1	Modos de acceso a los módulos	11
3.1.1	Arrastre de ícono	11
3.1.2	Librería	12
3.2	Descripción de los subVI's diseñados.....	18
3.2.1	Modulador M-PSK	18
3.2.2	Codificador Alamouti.....	19
3.2.3	AWGN MIMO 2x2 y AWGN MIMO 2x1	21
3.2.4	Canal Rayleigh MIMO 2x1 y Canal Rayleigh MIMO 2x2	23
3.2.5	Decodificador Alamouti.....	24
4.	IMPLEMENTACIÓN DE LOS subVI's EN UN ENTORNO DE TELECOMUNICACIONES	30
	BIBLIOGRAFÍA.....	35

LISTA DE FIGURAS	Pág.
FIGURA 1. Vista del panel frontal de LabVIEW	7
FIGURA 2. Vista del panel de programación de LabVIEW	7
FIGURA 3. Vista de la paleta de controles en el panel frontal	9
FIGURA 4. Vista de la paleta de funciones en el panel de programación	9
FIGURA 5. Ejemplo de un diagrama de bloques.....	10
FIGURA 6. Ejemplo de acceso a un subVI por medio del arrastre de su ícono	11
FIGURA 7. Vista de la carpeta que contiene todos los módulos generados.....	12
FIGURA 8. Ubicación de la nueva librería de usuario	13
FIGURA 9. Despliegue de la paleta de funciones sobre el panel de programación	14
FIGURA 10. Ubicación de la nueva librería de usuario sobre la paleta de funciones	15
FIGURA 11. Acceso a los subVI's desde la paleta de funciones.....	16
FIGURA 12. Ubicación de un subVI sobre el panel de programación	17
FIGURA 13. Interfaz de usuario del modulador M-PSK	18
FIGURA 14. Representación del Modulador M-PSK en el panel de programación	19
FIGURA 15. Interfaz de usuario del codificador Alamouti	20
FIGURA 16. Representación del codificador Alamouti en el panel de programación	21
FIGURA 17. Interfaz de usuario del AWGN MIMO 2x2	21
FIGURA 18. Representación del AWGN MIMO 2x1 y AWGN MIMO 2x2 en el panel de programación	22
FIGURA 19. Interfaz de usuario de Canal Rayleigh MIMO 2x2	23
FIGURA 20. Representación del Canal Rayleigh MIMO 2x1 y Canal Rayleigh MIMO 2x2 en el panel de programación.....	24
FIGURA 21. Interfaz de usuario del decodificador Alamouti.....	25
FIGURA 22. Representación del decodificador Alamouti en el panel de programación ...	26
FIGURA 23. Representación del combinador en el panel de programación	26
FIGURA 24. Interfaz de usuario del combinador.....	27
FIGURA 25. Interfaz de usuario del detector de máxima verosimilitud.....	29
FIGURA 26. Representación del detector de máxima verosimilitud en el panel de programación	29
FIGURA 27. Interfaz de los controles de la simulación del Entorno 1.....	31
FIGURA 28. Interfaz con las señales procesadas durante la simulación.....	32
FIGURA 29. Interfaz para observar las gráficas de BER.....	33
FIGURA 30. Gráfica de BER del entorno 1 y modulación BPSK.....	34

1. INTRODUCCIÓN

En este anexo se presentan las generalidades acerca de la herramienta LabVIEW en la que se ha desarrollado la simulación utilizada para concluir acerca del desempeño de la codificación Alamouti, cuyos componentes o módulos pueden ser utilizados para diseñar diferentes entornos de telecomunicaciones dependiendo de sus fines.

Con el fin de lograr la familiarización de los usuarios con el entorno de programación, en la primera parte de este documento se hace una breve reseña acerca de qué es LabVIEW y cómo funciona su interfaz.

En la segunda parte se plasma el objetivo principal de este anexo, que es mostrar detalladamente la funcionalidad de cada módulo diseñado, junto a una breve explicación del tipo de dato de entrada y salida.

2. GENERALIDADES DE LA HERRAMIENTA LABVIEW

2.1 Definición de LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) es un lenguaje de programación gráfico para el diseño de sistemas de adquisición de datos, instrumentación y control. LabVIEW permite diseñar interfaces de usuario mediante una consola interactiva basada en software, especificando el sistema funcional, el diagrama de bloques o una notación de diseño de ingeniería de cada usuario. Labview es a la vez compatible con herramientas de desarrollo similares y puede trabajar con programas de otra área de aplicación, como por ejemplo Matlab. Tiene la ventaja de permitir una fácil integración con hardware, específicamente con tarjetas de medición, adquisición y procesamiento de datos (incluyendo adquisición de imágenes) [2].

2.2 Aplicaciones de LabVIEW

LabVIEW tiene su mayor aplicación en sistemas de medición, como monitoreo de procesos y aplicaciones de control, un ejemplo de esto pueden ser sistemas de monitoreo en transportación, laboratorios para clases en universidades, procesos de control industrial. LabVIEW es muy utilizado en procesamiento digital de señales, procesamiento en tiempo real de aplicaciones biomédicas, manipulación de imágenes y audio, automatización, diseño de filtros digitales, generación de señales, entre otras [2].

2.3 Programación gráfica con LabVIEW

Cuando se diseñan programas con LabVIEW se trabaja bajo algo denominado VI (Virtual Instrument, Instrumento Virtual), que se puede crear a partir de especificaciones funcionales que cada usuario diseñe. Este VI puede utilizarse en cualquier otra aplicación como una subfunción dentro de un programa general. Los VI's se caracterizan por: ser un icono cuadrado con su respectivo símbolo relacionado con su funcionalidad, tener una interfaz con el usuario, tener entradas con su color de identificación de dato, tener una o varias salidas y por su puesto ser reutilizables [2].

2.4 Diseño de una interfaz de usuario a partir del código

En el ambiente de trabajo de LabVIEW existen dos paneles, el panel frontal y el panel de programación ó diagrama de bloques; en el panel frontal se diseña la interfaz de usuario y en el panel de programación se relacionan los elementos utilizados en la interfaz mediante operaciones que determinan en sí cómo funciona el programa o el sistema, exactamente es la parte donde se realizan las especificaciones funcionales.

En el panel de programación se puede diseñar el funcionamiento del sistema de manera gráfica, como un diagrama de bloques. La programación gráfica se basa en la realización de operaciones mediante la asignación de iconos que representen los datos numéricos e iconos que representan los procedimientos que deben realizar los VI's. Con estos iconos y mediante una conexión simple como lo es una línea recta, se enlazan para determinar una operación y/o una función.

Al diseñar el programa de forma gráfica, se hace visible una programación orientada al flujo de datos, donde se tiene una interpretación de los datos también de forma gráfica. Por ejemplo un dato booleano se caracteriza por ser una conexión verde, cada tipo de dato se identifica con un color diferente dentro de LabVIEW; también es necesario tener en cuenta que cuando se realiza una conexión a un VI esta conexión se identifica por un tipo de dato específico, que debe coincidir con el tipo de dato de la entrada del VI permitiendo una concordancia en el flujo de datos, aunque no siempre el tipo de dato de la entrada del VI es el mismo que el de la salida, pero sin embargo para la mayoría de los casos si se cumple.

El flujo de datos va de izquierda a derecha en el panel de programación y está determinado por las operaciones o funciones que procesan los datos. Es fácil observar en el panel de programación cómo se computan los datos en cada parte del programa cuando se realiza una ejecución del programa paso a paso. En LabVIEW las variables se representan mediante una figura tanto en el panel frontal como en el panel de programación, de esta forma se puede observar su respuesta en la interfaz del usuario y en el flujo de datos del código del programa. Otros objetos como gráficas y accesos directos a páginas web cumplen estas mismas condiciones [2].

Ejemplos del panel frontal y de programación se presentan en las figuras 1 y 2 respectivamente.

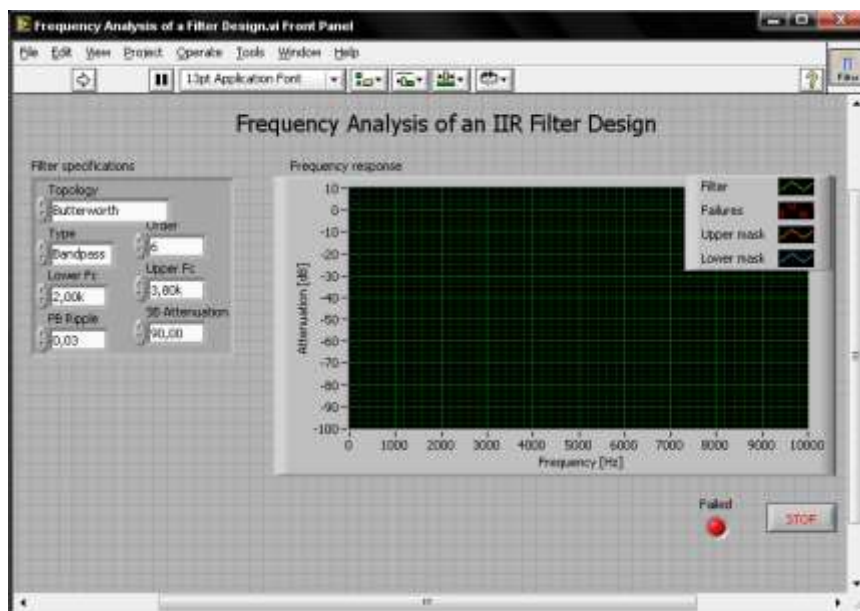


FIGURA 1. Vista del panel frontal de LabVIEW

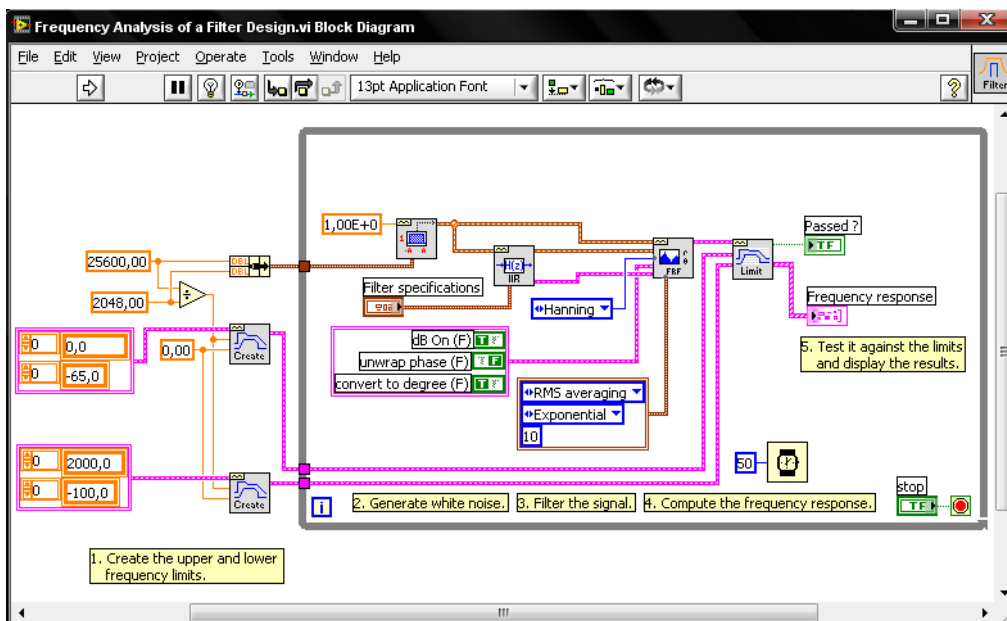


FIGURA 2. Vista del panel de programación de LabVIEW

2.5 El entorno LabVIEW

La programación G (gráfica) de Labview consta de un panel frontal y un panel de programación como se mencionó antes. En el panel frontal se diseña la interfaz de usuario y se ubica la paleta de controles e indicadores a la que se accede por medio de

un click derecho en cualquier espacio. En el panel de programación se encuentra la paleta de funciones a la que también se accede por medio de un click derecho sobre cualquier espacio. Cada control que se utiliza en la interfaz tiene una representación en el panel de programación, igualmente los indicadores necesarios para entregar la información procesada al usuario tienen un ícono que los identifica en el panel de código o de programación.

Los controles pueden ser booleanos, numéricos, strings, un arreglo matricial de estos o una combinación de los anteriores; y los indicadores pueden ser como los controles pero visualizados como tablas, gráficos en 2D o 3D, browser, entre otros. Las funciones pueden ser VIs prediseñados que pueden ser reutilizados en cualquier aplicación para procesar datos de entrada y entregar de salida. Estos bloques pueden también estar conformados por otros subVI's y así sucesivamente, formando un árbol genealógico donde un VI se relaciona o depende de varios SubVIs.

LabVIEW tiene VIs de adquisición de datos e imágenes, de comunicaciones, de procesamiento digital de señales y desde funciones matemáticas simples, hasta otras que utilizan programas como Matlab o HiQ para resolver problemas. Además tiene funciones más complejas como "nodos de fórmula" que se utilizan para la resolución de ecuaciones editando directamente estas como en lenguajes de programación tradicionales, definiendo las entradas y las salidas.

LabVIEW también se puede utilizar para graficar en tres dimensiones, en coordenadas polares y cartesianas, tiene disponibles herramientas para análisis de circuitos RF como la Carta de Smith, tiene aplicaciones en manejo de audio y se puede comunicar con la tarjeta de sonido del computador para trabajar conjuntamente. Entre sus muchas funciones especiales se encuentran las de procesamiento de imágenes, como capturar una imagen a través de una tarjeta de adquisición como la PCI-1408 (monocromática), analizarla y entregar respuestas que difícilmente otros sistemas realizarían [2].

Ejemplos de las paletas de controles y funciones se presentan en las figuras 3 y 4 respectivamente.

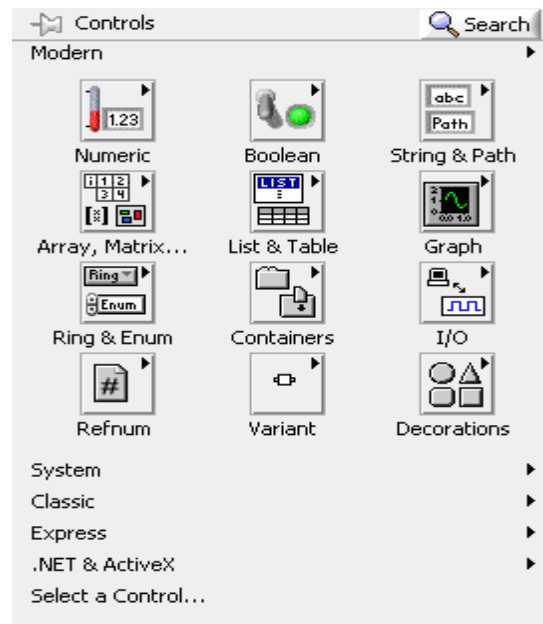


FIGURA 3. Vista de la paleta de controles en el panel frontal

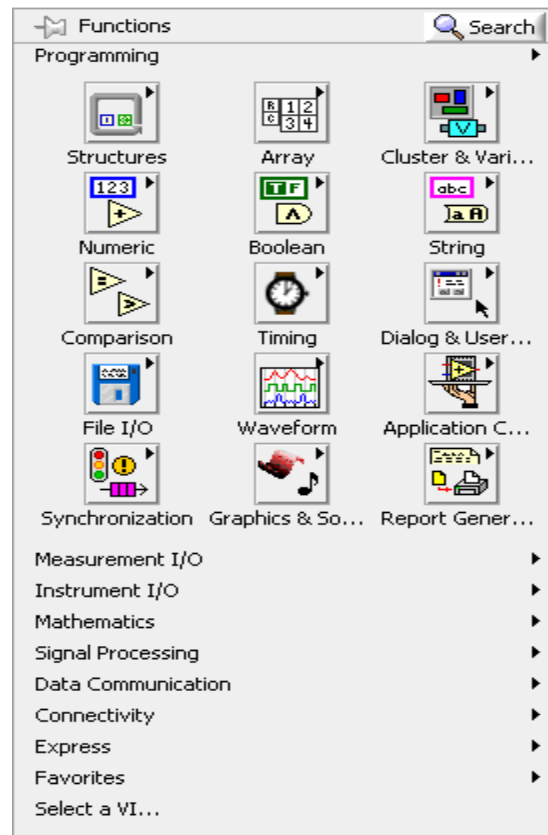


FIGURA 4. Vista de la paleta de funciones en el panel de programación

2.6 Flujo de datos

El flujo de datos muestra la ejecución secuencial del programa y una tarea no se inicia hasta no tener en todos sus variables de entrada información o que las tareas predecesoras hayan terminado de ejecutarse. Debido al lenguaje gráfico el compilador con que cuenta LabVIEW es más versátil ya que sobre el mismo código de programación se puede ver fácilmente el flujo de datos, así como su contenido [2]. En la figura 5 se presenta un ejemplo de diagrama de bloques en el panel de programación.

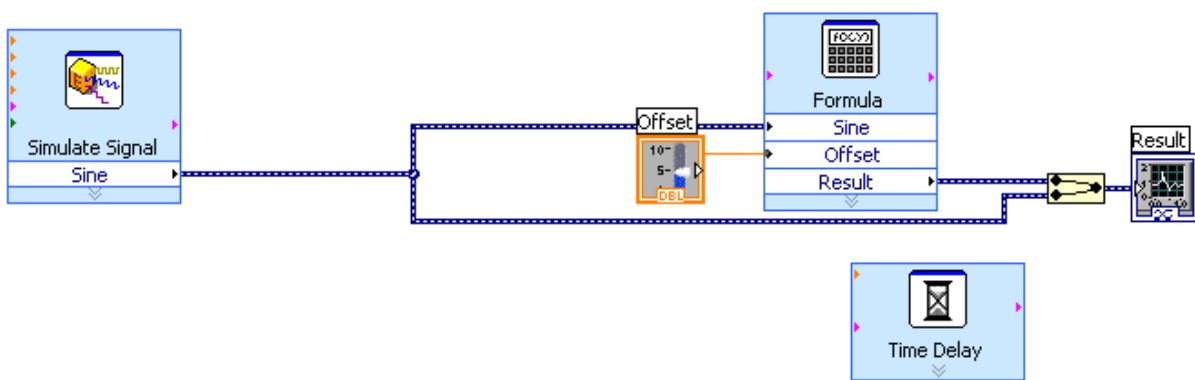


FIGURA 5. Ejemplo de un diagrama de bloques

3. DESCRIPCIÓN DE LOS MÓDULOS GENERADOS PARA LA SIMULACIÓN DE LA CODIFICACIÓN ALAMOUTI

En este capítulo se describen las formas de acceso y las características de cada uno de los módulos generados para la simulación de la codificación Alamouti en sistemas MIMO (Multiple Input Multiple Output, Múltiples Entradas-Múltiples Salidas).

3.1 Modos de acceso a los módulos

3.1.1 Arrastre de ícono

Es el modo más sencillo de acceder a cualquier módulo generado en LabVIEW, desde ahora llamado subVI, y consiste en arrastrar el ícono correspondiente desde su carpeta origen hacia el panel de programación. La figura 6 presenta un ejemplo de este modo.

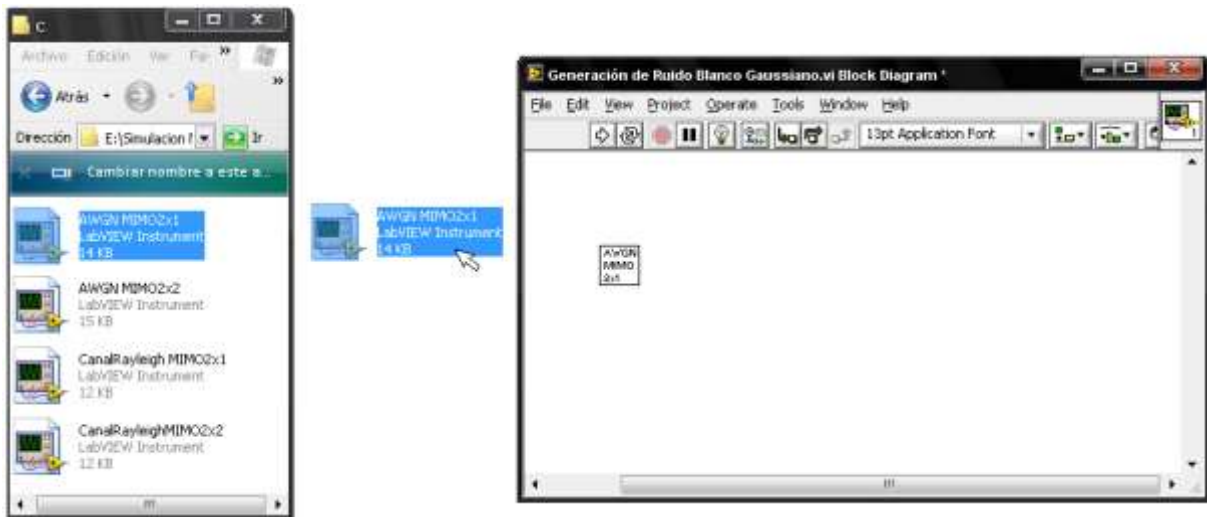


FIGURA 6. Ejemplo de acceso a un subVI por medio del arrastre de su ícono

3.1.2 Librería

Este modo es muy útil si un subVI debe utilizarse en varias ocasiones, pues permite su rápida ubicación.

Para poner en práctica este modo de acceso a los subVI's presentados en este trabajo, se deben seguir los siguientes pasos:

1. Transportar todos los íconos que representan los subVI's generados, desde cada una de las carpetas de origen a una con un nombre en particular, por ejemplo, **sistema MIMO**, como se presenta en la figura 7.

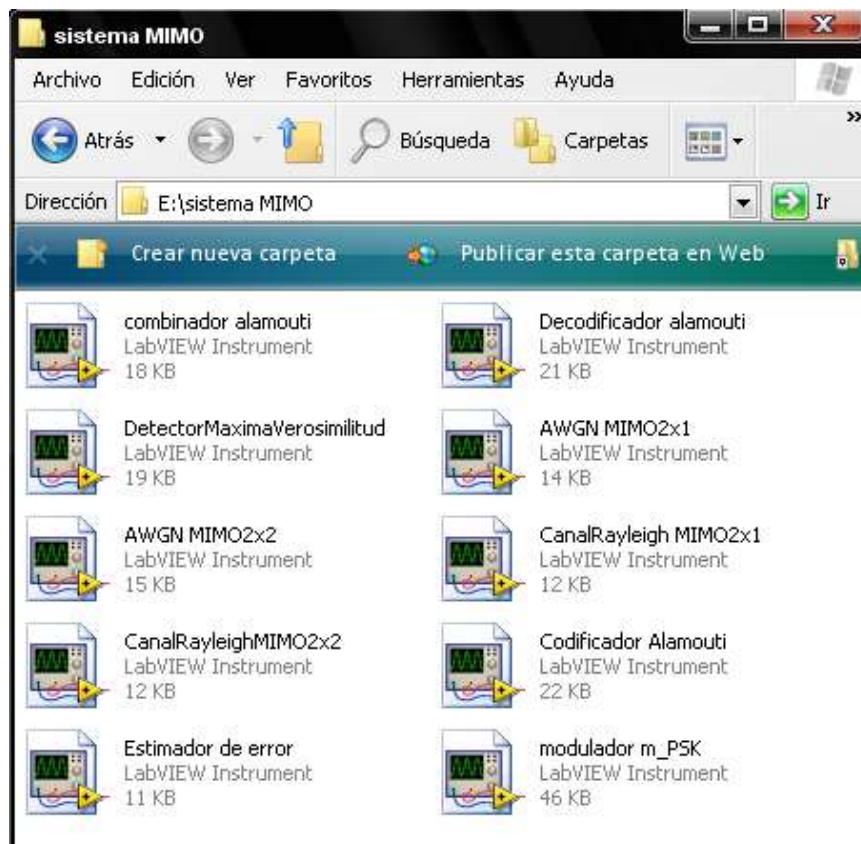


FIGURA 7. Vista de la carpeta que contiene todos los módulos generados

2. La nueva carpeta debe copiarse en la carpeta **user.lib** ubicada en la dirección:
C:\Archivos de programa\National Instruments\LabVIEW 8.0\user.lib

Ahora la carpeta **sistema MIMO** funciona como una nueva librería de usuario.

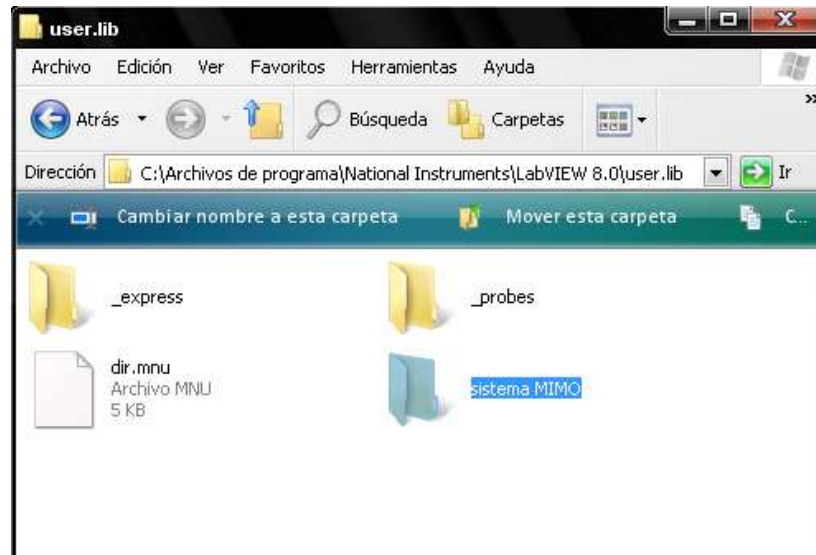


FIGURA 8. Ubicación de la nueva librería de usuario

3. Abrir LabVIEW y hacer click derecho sobre cualquier espacio del panel de programación para desplegar la paleta de funciones como se presenta en la figura 9.

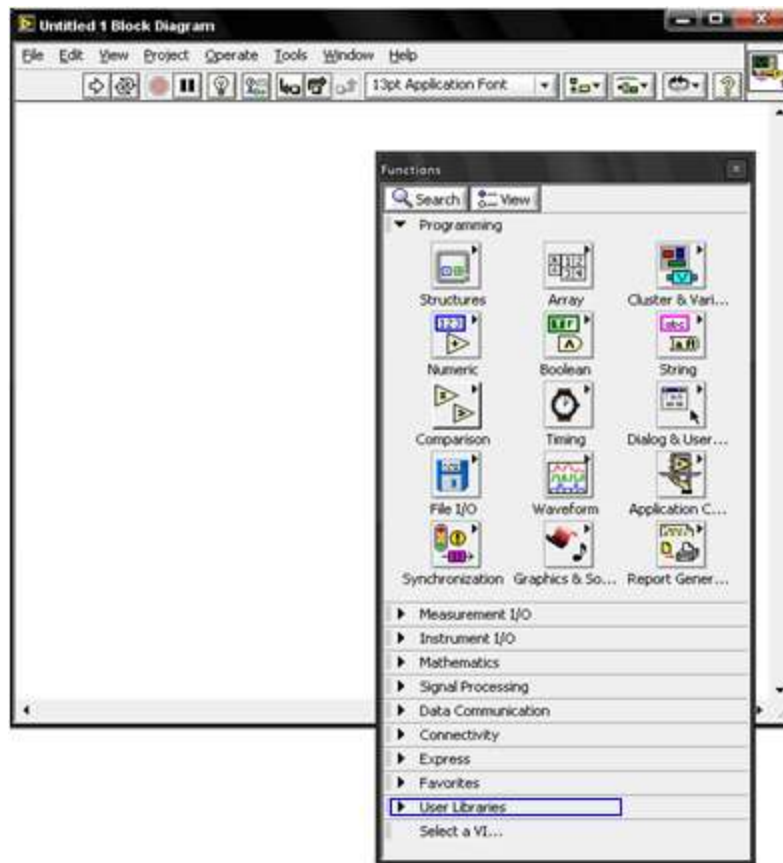


FIGURA 9. Despliegue de la paleta de funciones sobre el panel de programación

4. Seleccionar con un click la opción **User Libraries** ubicada en la parte inferior de la paleta de funciones para finalmente encontrar la nueva librería llamada **sistema MIMO** como se presenta en la figura 10.

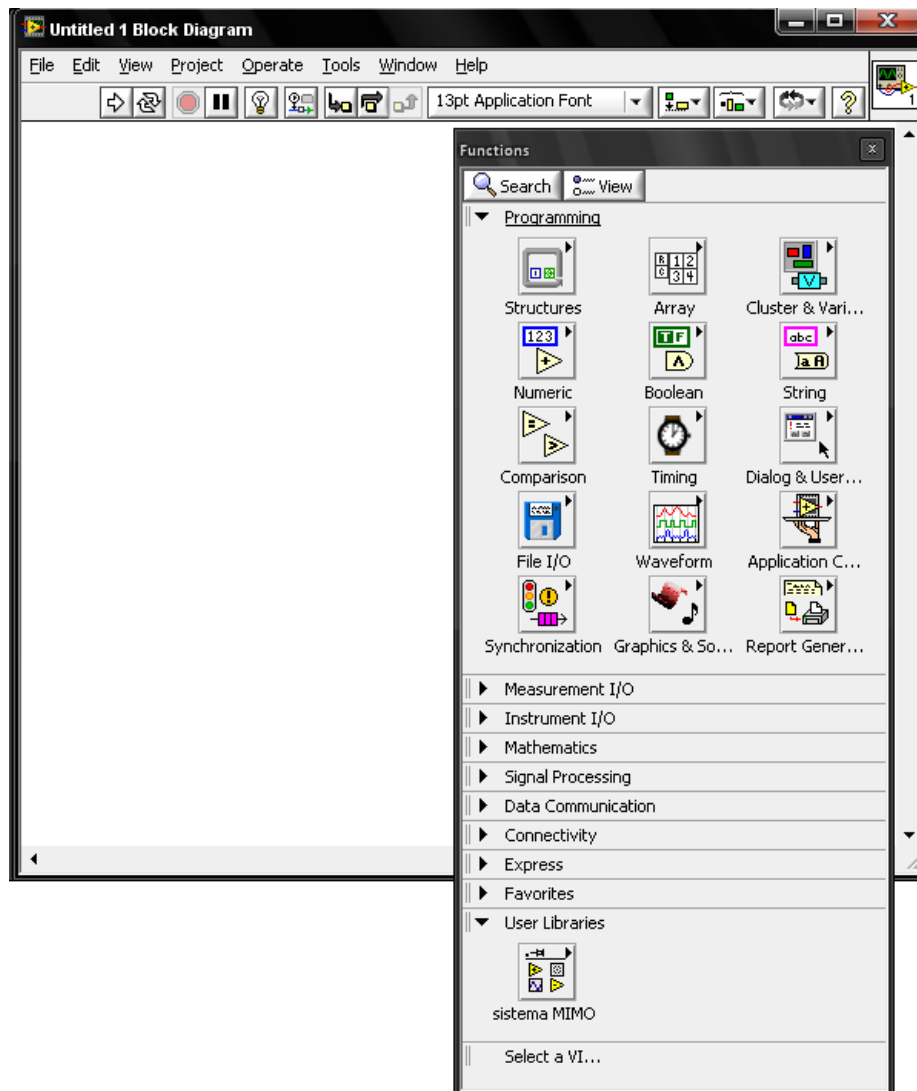


FIGURA 10. Ubicación de la nueva librería de usuario sobre la paleta de funciones

5. A continuación se hace click sobre la librería **sistema MIMO** para acceder a todos los subVI's generados como se presenta en la figura 11.

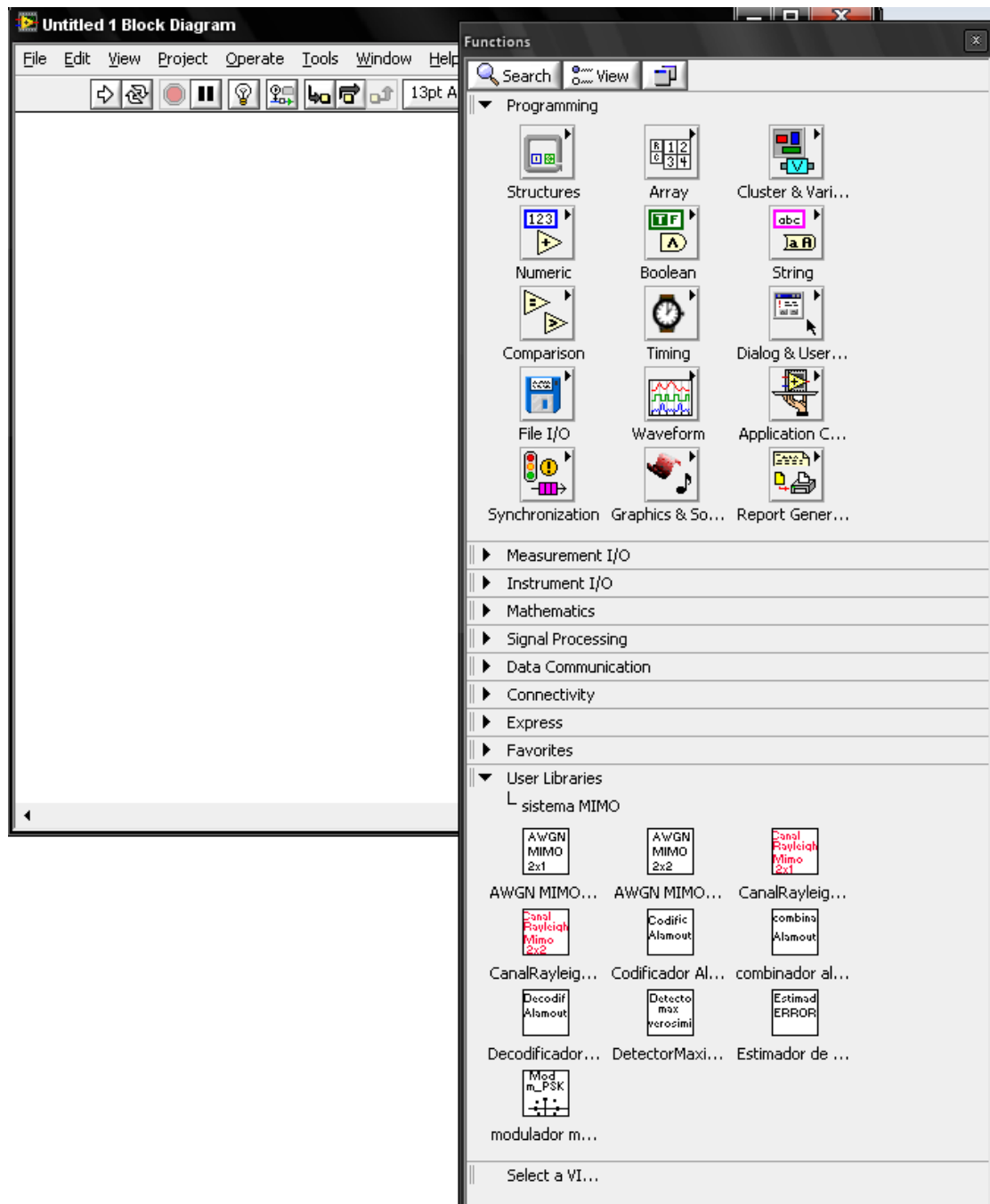


FIGURA 11. Acceso a los subVI's desde la paleta de funciones

6. Se hace click sobre el subVI necesario y finalmente se ubica sobre el panel de programación como se presenta en la figura 12.

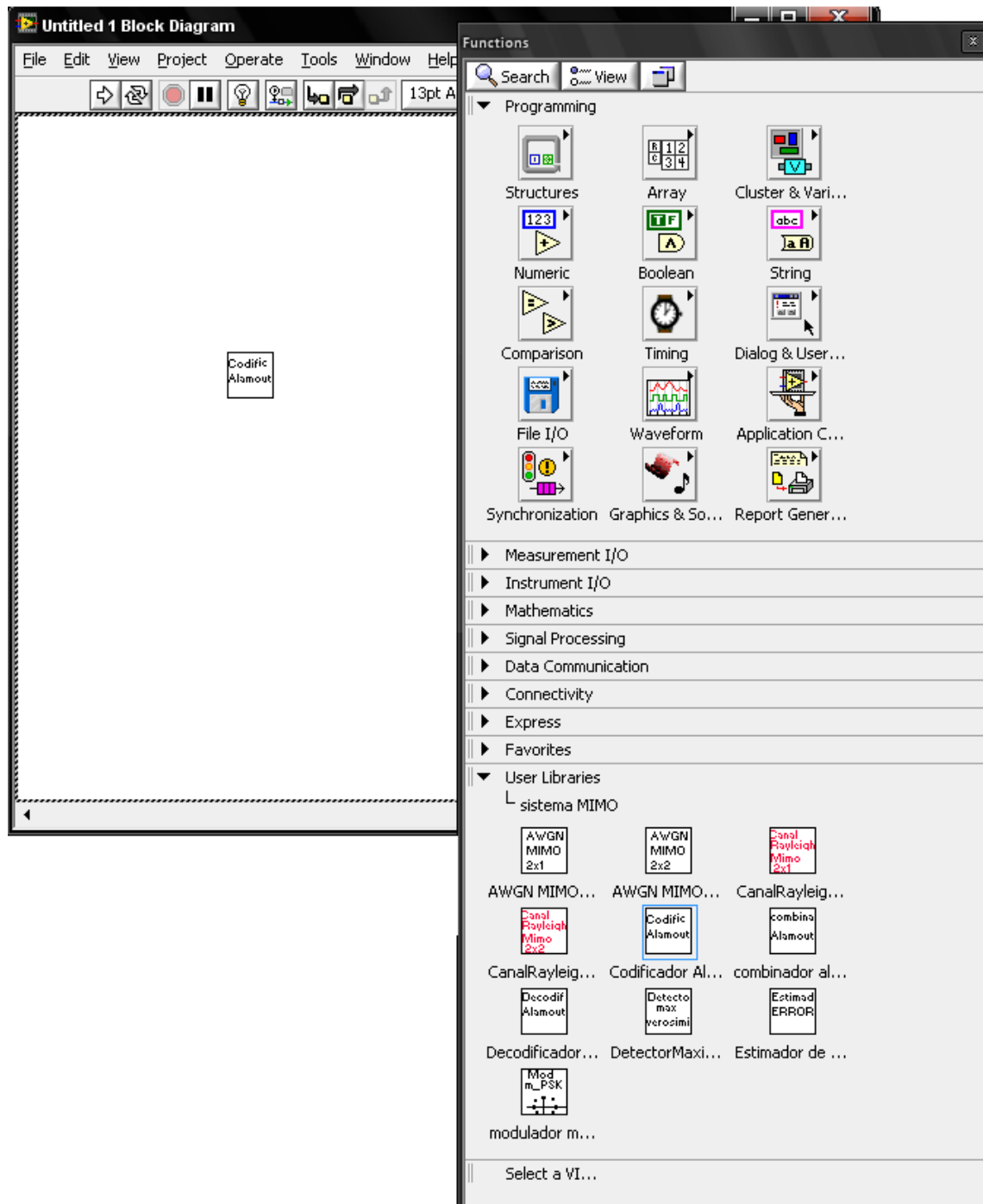


FIGURA 12. Ubicación de un subVI sobre el panel de programación

3.2 Descripción de los subVI's diseñados

En la carpeta **Simulación de entornos de telecomunicaciones** se encuentran los subVI's diseñados ubicados en diferentes carpetas de acuerdo a su función dentro de los entornos de telecomunicaciones. Cada uno de los subVI's presentados a continuación funcionan como un programa con una interfaz de usuario en el panel frontal, al que se puede tener acceso por medio de un doble click al ícono que lo representa.

3.2.1 Modulador M-PSK

Se encuentra ubicado en la carpeta **/Simulación de entornos de telecomunicaciones/modulador M-PSK**. Sirve para generar secuencias aleatorias de símbolos M-PSK y la figura 13 presenta su interfaz correspondiente.

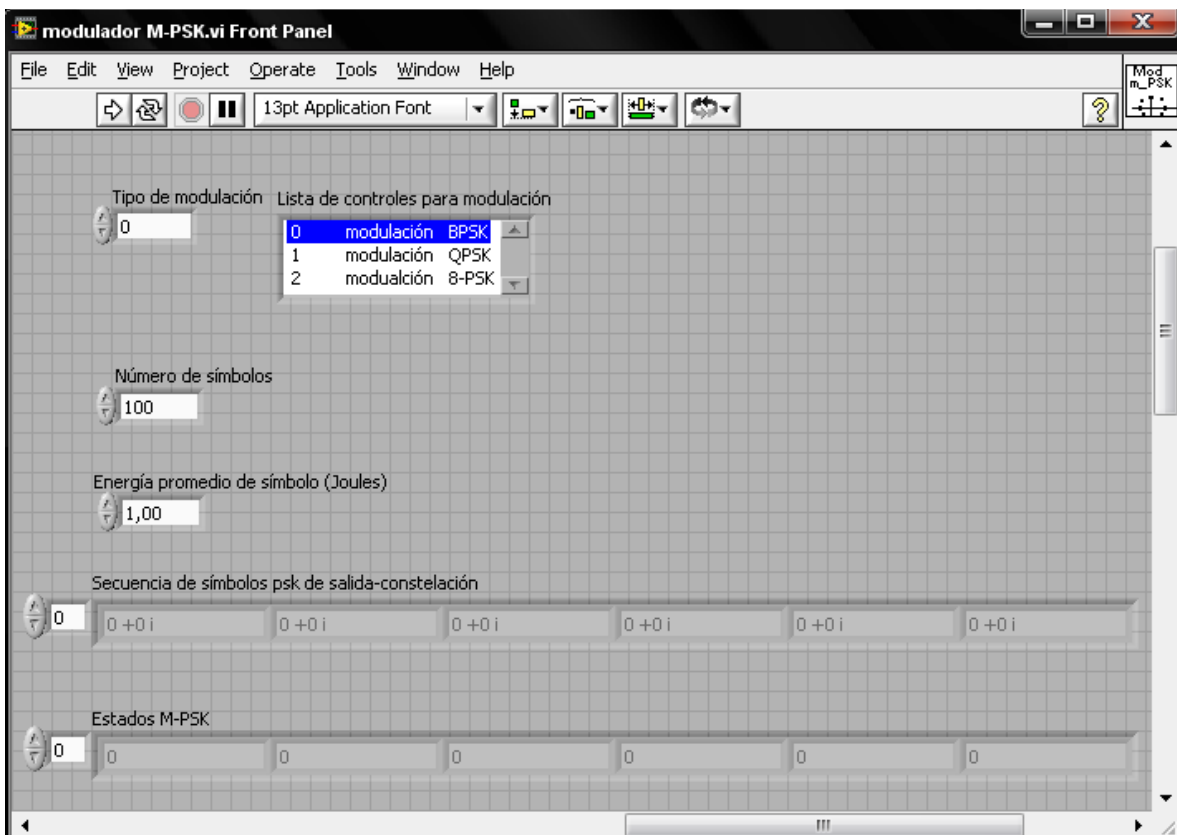


FIGURA 13. Interfaz de usuario del modulador M-PSK

En el campo **Tipo de modulación** el usuario puede escoger el tipo de secuencias a generar de acuerdo a la **lista de controles para modulación** que pueden ser BPSK, QPSK y 8-PSK.

En el campo **Número de símbolos** se especifica cuántos símbolos se requiere generar.

En el campo **energía promedio de símbolo** se especifica la energía promedio de los símbolos con que deben generarse las constelaciones M-PSK.

Al ejecutar el programa, automáticamente se generan dos tipos de salida de acuerdo a los parámetros especificados por el usuario. La **secuencia de símbolos psk de salida** contiene las coordenadas de cada símbolo en la constelación M-PSK generada y los **Estados M-PSK** representan el estado al que corresponde cada símbolo. En el ejemplo que se presenta en la figura 13, se generó una secuencia BPSK de 2 símbolos con una energía promedio de símbolo de 0,8. La primera coordenada de símbolo generada es $-0,89+0i$ y su estado correspondiente es el 1(uno), mientras que para la coordenada $0,89+0i$ es el 0 (cero).

Puede ser utilizado en otro programa por medio de los modos presentados en la sección 3.1 y su ícono representativo en el panel de programación es el presentado en la figura 14 con todas sus entradas y salidas.

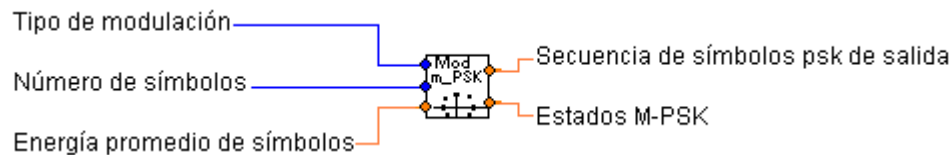


FIGURA 14. Representación del Modulador M-PSK en el panel de programación

3.2.2 Codificador Alamouti

Se encuentra ubicado en la carpeta **/Simulación de entornos de telecomunicaciones/codificador Alamouti**. Este subVI toma bloques de 2 símbolos y aplica la matriz de codificación propuesta por Alamouti [1]. La figura 14 presenta la interfaz correspondiente.

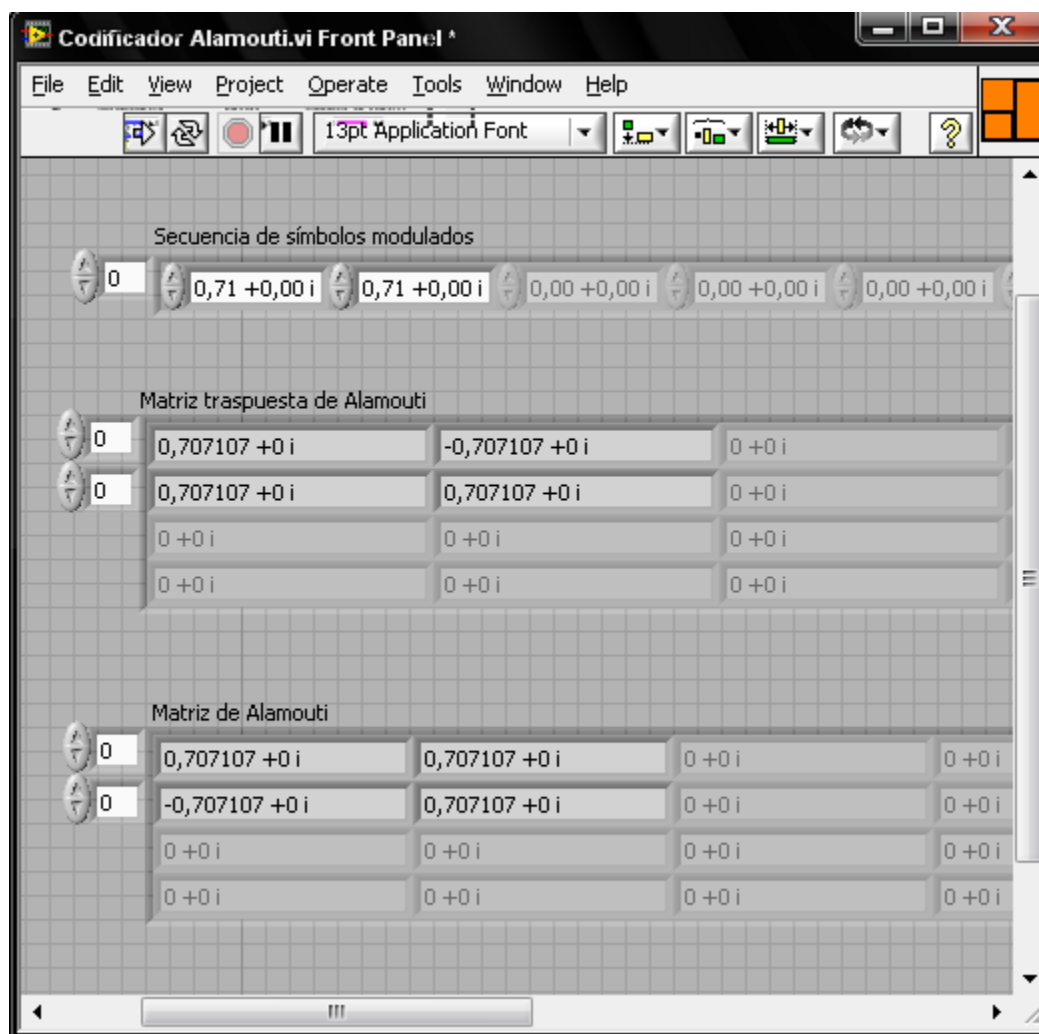


FIGURA 15. Interfaz de usuario del codificador Alamouti

En la figura 15 se muestra cómo el programa organiza el bloque de dos símbolos (S_1 y S_2) en dos matrices de salida. La **Matriz de Alamouti** obedece a la matriz:

$$\begin{bmatrix} S_1 & S_2 \\ -S_2^* & S_1^* \end{bmatrix}$$

La otra salida es la **matriz traspuesta de Alamouti**:

$$\begin{bmatrix} S_1 & -S_2^* \\ S_2 & S_1^* \end{bmatrix}$$

La matriz traspuesta suele ser utilizada en simulaciones por efectos de organización, además que facilita las operaciones matriciales.

Puede utilizarse en otro programa por medio de los modos presentados en la sección 3.1 y su ícono representativo en el panel de programación es el presentado en la figura 16 con sus entradas y salidas.

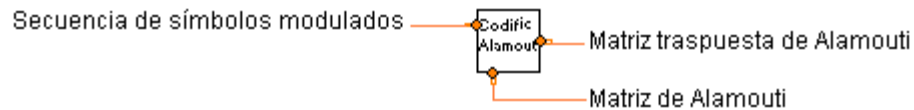


FIGURA 16. Representación del codificador Alamouti en el panel de programación

3.2.3 AWGN MIMO 2x2 y AWGN MIMO 2x1

Se encuentran ubicados en la carpeta **/Simulación de entornos de telecomunicaciones/Canales Rayleigh y AWGN**.

Generan aleatoriamente muestras que obedecen a una distribución gaussiana (ideal para simular el ruido añadido a la señal por el receptor) para sistemas MIMO 2x1 y 2x2. La figura 17 presenta la interfaz correspondiente al subVI **AWGN MIMO 2x2**.

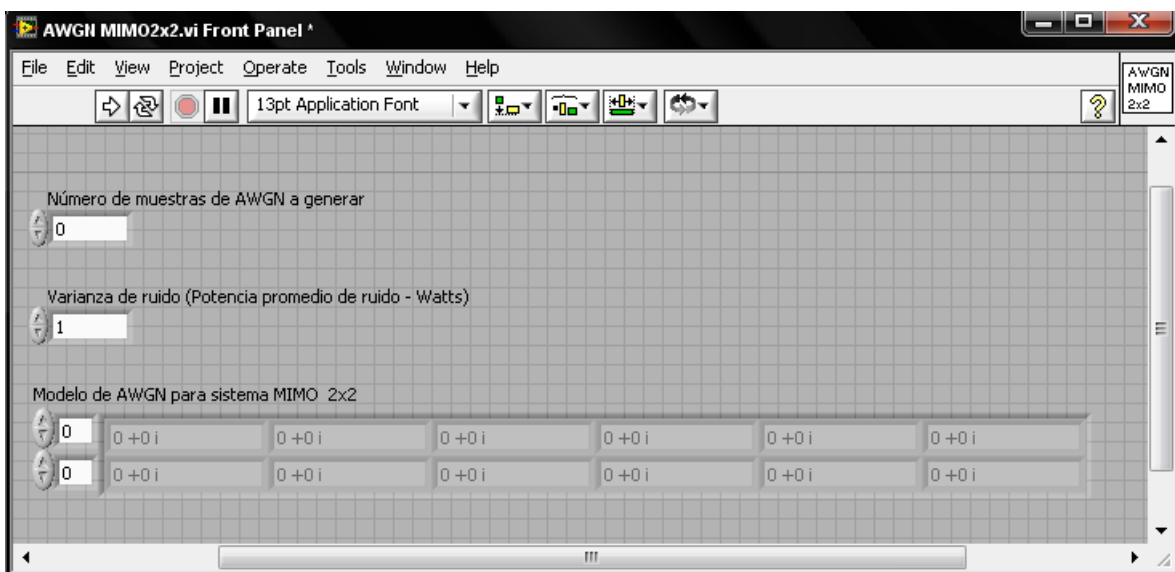


FIGURA 17. Interfaz de usuario del AWGN MIMO 2x2

El usuario debe especificar el número de muestras AWGN a generar y la varianza de ruido de la que depende la distribución gaussiana. Finalmente se produce una matriz con las muestras de AWGN generadas. El subVI AWGN MIMO 2x1 genera una matriz de una sola fila debido al único receptor siguiendo el modelo:

$$[n_{11} \quad n_{21}]$$

Donde:

η_{11} = muestra de ruido añadido por Rx_1 en el instante t

η_{21} = muestra de ruido añadido por Rx_1 en el instante t+T

Como se observa en la figura 17, **AWGN MIMO 2x2** genera una matriz de 2 filas debido a los 2 receptores del sistema que tiene el siguiente modelo:

$$\begin{bmatrix} \eta_{11} & \eta_{21} \\ \eta_{12} & \eta_{22} \end{bmatrix}$$

Donde:

η_{11} = Muestra de ruido añadido por el receptor 1 en el instante t.

η_{12} = Muestra de ruido añadido por el receptor 2 en el instante t.

η_{21} = Muestra de ruido añadido por el receptor 1 en el instante t+T.

η_{22} = Muestra de ruido añadido por el receptor 2 en el instante t+T.

Pueden utilizarse en otro programa por medio de los modos presentados en la sección 3.1 y sus íconos representativos en el panel de programación es el presentado en la figura 18 con todas sus entradas y salidas.

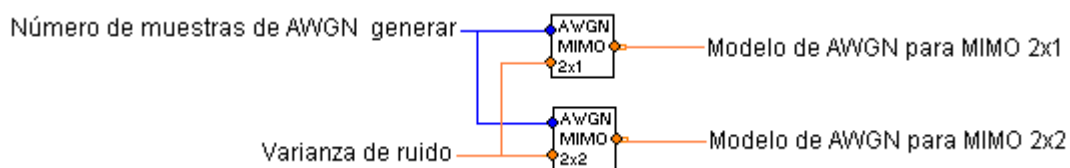


FIGURA 18. Representación del AWGN MIMO 2x1 y AWGN MIMO 2x2 en el panel de programación

3.2.4 Canal Rayleigh MIMO 2x1 y Canal Rayleigh MIMO 2x2

Se encuentran ubicados en la carpeta **/Simulación de entornos de telecomunicaciones/Canales Rayleigh y AWGN**. Generan aleatoriamente muestras que obedecen a una distribución Rayleigh (ideal para simular la respuesta del canal entre transmisor y receptor) para sistemas MIMO 2x1 y 2x2. La figura 19 presenta la interfaz del subVI **AWGN MIMO 2x2**.

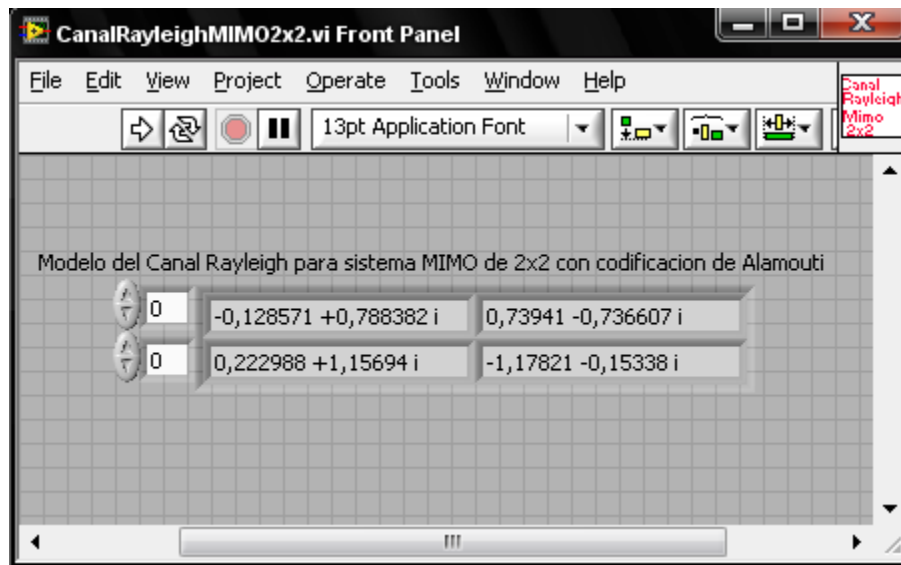


FIGURA 19. Interfaz de usuario de Canal Rayleigh MIMO 2x2

El subVI **Canal Rayleigh MIMO 2x1** genera una matriz de dimensión 2x1 que presenta el siguiente modelo:

$$[h_{11} \quad h_{21}]$$

Donde: h_{11} = coeficiente de la respuesta del canal entre Tx_1 y Rx_1

h_{21} = coeficiente de la respuesta del canal entre Tx_2 y Rx_1

En la figura 19 se observa que el **Canal Rayleigh MIMO 2x2** genera una matriz 2x2 que sigue el siguiente modelo:

$$\begin{bmatrix} h_{11} & h_{21} \\ h_{12} & h_{22} \end{bmatrix}$$

Donde:

h_{11} = Coeficiente de la respuesta del canal entre la antena transmisora 1 y el receptor 1.

h_{12} = Coeficiente de la respuesta del canal entre la antena transmisora 1 y el receptor 2.

h_{21} = Coeficiente de la respuesta del canal entre la antena transmisora 2 y el receptor 1.

h_{22} = Coeficiente de la respuesta del canal entre la antena transmisora 2 y el receptor 2.

Pueden utilizarse en otro programa por medio de los modos presentados en la sección 3.1 y sus íconos representativos en el panel de programación son los presentados en la figura 20 con todas sus entradas y salidas.

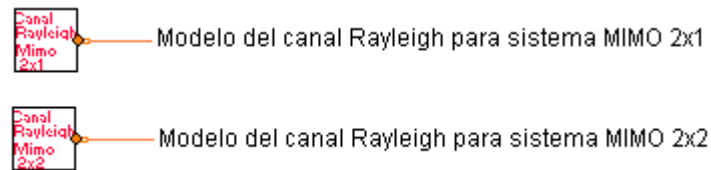


FIGURA 20. Representación del Canal Rayleigh MIMO 2x1 y Canal Rayleigh MIMO 2x2 en el panel de programación

3.2.5 Decodificador Alamouti

Se encuentra ubicado en la carpeta **/Simulación de entornos de telecomunicaciones/Decodificador Alamouti** y toma los símbolos que han llegado a recepción y aplica el proceso de combinación y detección, propuestos en [1].

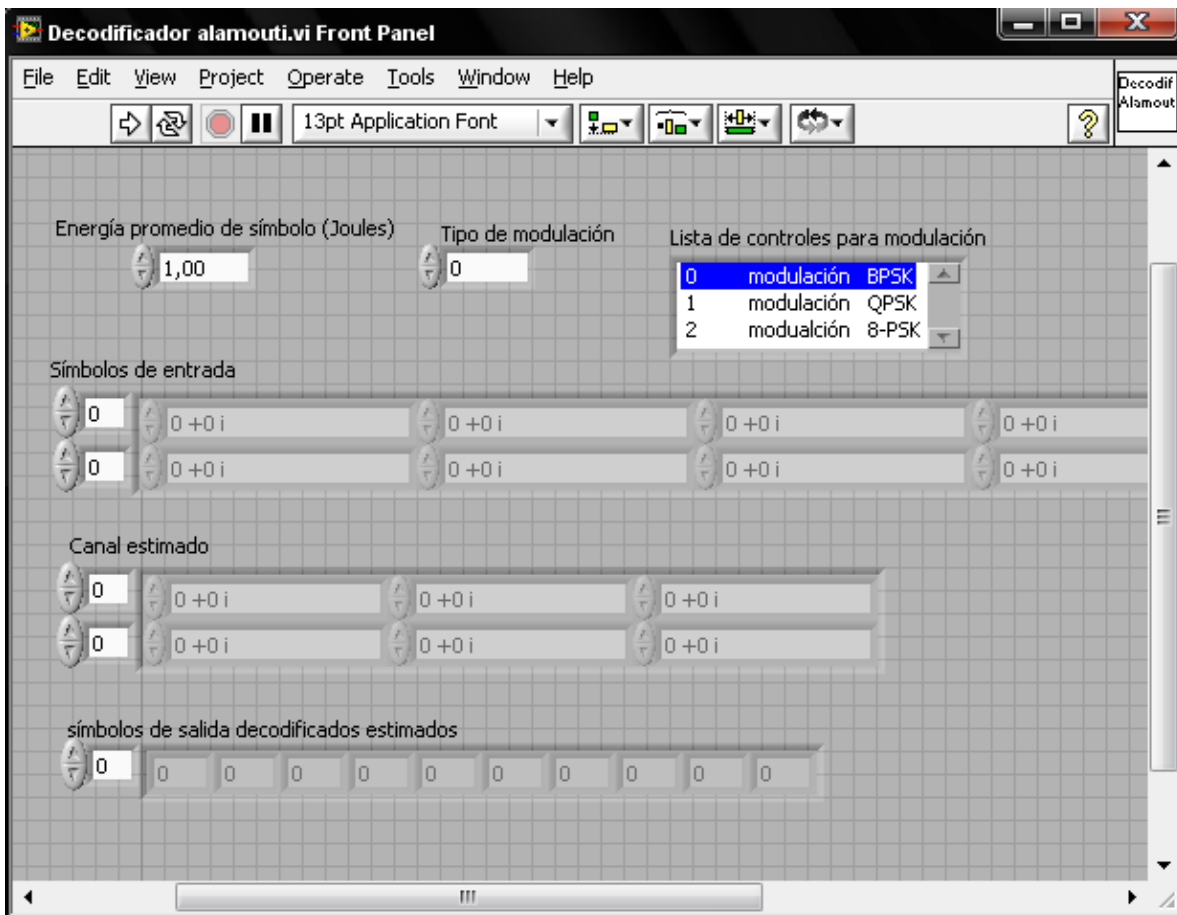


FIGURA 21. Interfaz de usuario del decodificador Alamouti

Permite especificar la **energía promedio de símbolo** y el **tipo de modulación** de los símbolos transmitidos originalmente. Además, como entradas se requieren los **símbolos de entrada** que son a su vez los que llegan a recepción y el **canal estimado**, que en condiciones ideales, es el mismo modelo del canal generado por el **canal Rayleigh MIMO 2x1 ó 2x2** según se requiera.

El programa finalmente, por el criterio de máxima verosimilitud, muestra los símbolos decodificados estimados.

Puede utilizarse en otro programa por medio de los modos presentados en la sección 3.1 y su ícono representativo en el panel de programación es el presentado en la figura 22 con todas sus entradas y salidas.

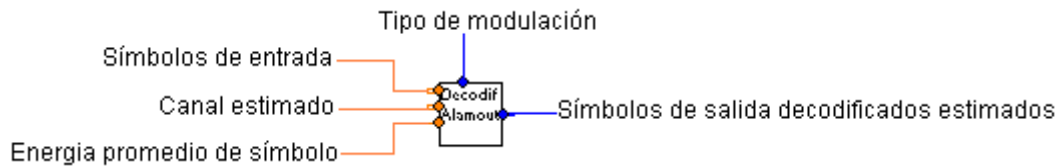


FIGURA 22. Representación del decodificador Alamouti en el panel de programación

Gracias a que los estudios actuales sobre la codificación espacio-tiempo apuntan a modificaciones en el proceso de combinación propuesto por Alamouti, además de la utilización del criterio de máxima verosimilitud, el decodificador Alamouti está modularizado en dos subVI's que pueden también ser utilizados como programas independientes, reutilizados o modificados en estudios posteriores.

Combinador Alamouti

Se encuentra ubicado en la carpeta **/Simulación de entornos de telecomunicaciones/Decodificador Alamouti** y se encarga de realizar el proceso de combinación propuesto en [1]. En la figura 23 se encuentra su ícono representativo en el panel de programación con todas sus entradas y salidas.



FIGURA 23. Representación del combinador en el panel de programación

Los símbolos de entrada son los que llegan al receptor y deben operarse con el canal estimado para llevar a cabo el proceso de combinación. Debido a que los símbolos de entrada están dispuestos en una matriz, ésta debe descomponerse en bloques de dos columnas por lo que debe especificarse a qué **columna n** y **n+1** se quiere realizar la combinación. Debe tenerse en cuenta que para LabVIEW, la primera columna es la número cero.

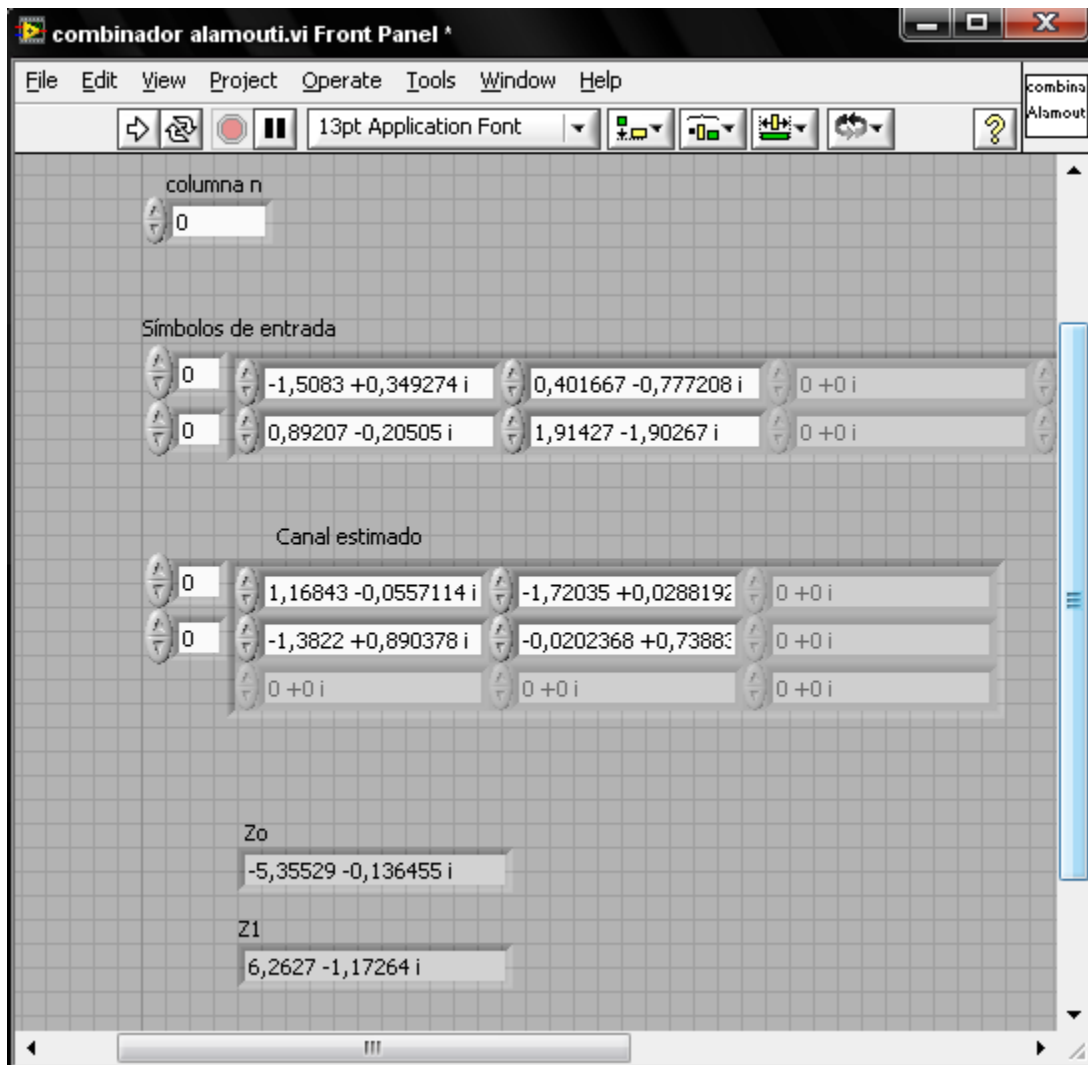


FIGURA 24. Interfaz de usuario del combinador

En la figura 24 se muestra un ejemplo de la aplicación del combinador. La matriz correspondiente a **símbolos de entrada** tiene la siguiente configuración:

$$\begin{bmatrix} r_{11} & r_{21} \\ r_{12} & r_{22} \end{bmatrix}$$

Donde:

r_{11} = Señal recibida en la antena 1 en el instante t.

r_{12} = Señal recibida en la antena 2 en el instante t.

r_{21} = Señal recibida en la antena 1 en el instante t+T.

r_{22} = Señal recibida en la antena 2 en el instante $t+T$.

El valor de la **columna n** es cero, por lo que la combinación se hace entre la columna cero y uno, con la columna cero y uno de la matriz del canal de la siguiente manera:

Z_1 es igual a $h_{11}^*r_{11} + h_{21}r_{21}^*$ para un sistema 2x1 y es igual a $h_{11}^*r_{11} + h_{21}r_{21}^* + h_{12}^*r_{12} + h_{22}r_{22}^*$ para un sistema 2x2. Esta señal es la construida por el combinador en el instante t .

Z_2 es igual a $h_{21}^*r_{11} - h_{11}r_{21}^*$ para un sistema 2x1 y es igual a $\widetilde{S}_2 = h_{21}^*r_{11} - h_{11}r_{21}^* + h_{22}^*r_{12} - h_{12}r_{22}^*$ para un sistema 2x2. Esta señal es la construida por el combinador en el instante $t+T$.

Detector de máxima verosimilitud

Se encarga de aplicar el criterio de máxima verosimilitud a Z_1 y Z_2 para estimar qué símbolos se transmitieron originalmente.

En la figura 25 se muestra un ejemplo del funcionamiento del detector de máxima verosimilitud. En este, el subVI estima a partir de Z_1 y Z_2 que el primer y segundo símbolo transmitido son el uno y el cero respectivamente.

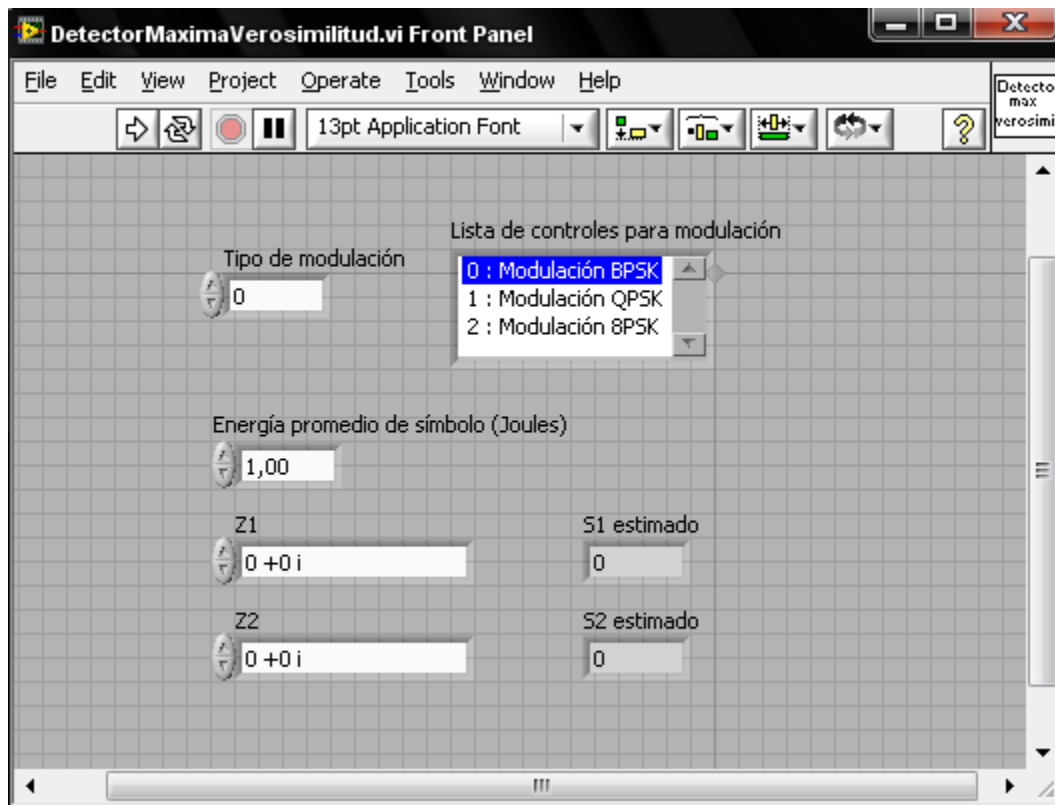


FIGURA 25. Interfaz de usuario del detector de máxima verosimilitud

En la figura 26 se encuentra el ícono representativo de éste subVI en el panel de programación con todas sus entradas y salidas.

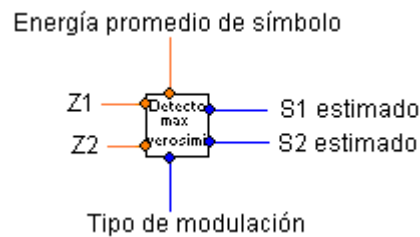


FIGURA 26. Representación del detector de máxima verosimilitud en el panel de programación

4. IMPLEMENTACIÓN DE LOS SUBVI'S EN UN ENTORNO DE TELECOMUNICACIONES

Con el presente trabajo se entrega una simulación en la que se hace uso de todos los subVI's diseñados. La implementación permite obtener gráficas de BER (Bit Error Rate, Probabilidad de Error en Bit) de sistemas sin diversidad, MRRC (Maximum Ratio Receive Combining, Combinación de Relación Máxima en Recepción) y sistemas con codificación Alamouti, a partir de la especificación de sus variables por parte del usuario.

En total se entregan 2 implementaciones correspondientes a:

Simulación de entorno 1: - Sistema sin diversidad

- Sistema MRRC con 2 receptores
- Sistema MISO (Multiple Input-Single Output, Múltiples Entradas- Única Salida) 2x1 con codificación Alamouti

Simulación de entorno 2: - Sistema sin diversidad

- Sistema MRRC con 4 receptores
- Sistema MIMO 2x2 con codificación Alamouti

A continuación se explica la manera en que se opera la interfaz de la simulación del entorno 2 que es similar a la del entorno 1.

Para abrir la simulación se debe hacer doble click en el ícono **Entorno 2** que se encuentra ubicado en la dirección **\Simulación de entornos de telecomunicaciones\Simulaciones**.

Se desplegará una interfaz como la presentada en la figura 27.

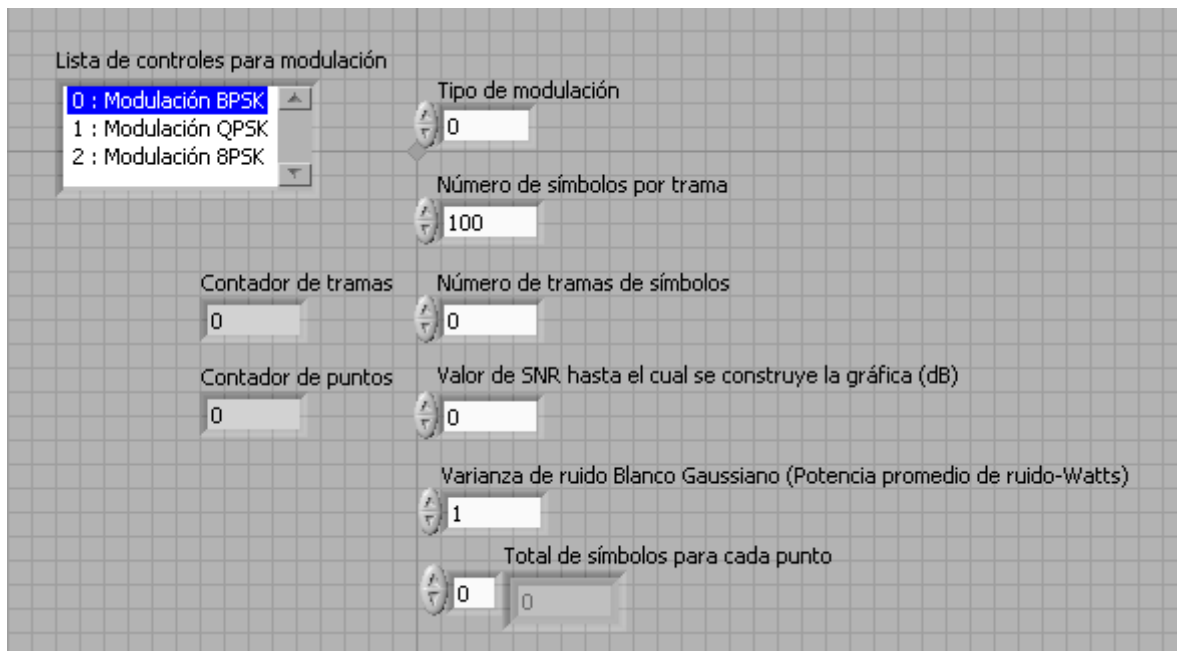


FIGURA 27. Interfaz de los controles de la simulación del Entorno 2

En esta parte de la interfaz se deben especificar las características que controlan la simulación. Entre ellas se tiene:

Tipo de modulación: 0 para modulación BPSK
 1 para modulación QPSK
 2 para modulación 8-PSK

Número de símbolos por trama: Cuantos símbolos modulados se requiere generar por cada trama.

Número de tramas de símbolos: Especifica el número de tramas que se requiere generar.

Hasta qué valor de SNR (Signal Noise Ratio, Relación Señal a Ruido) **se quiere construir la gráfica.**

Varianza de ruido blanco gaussiano: especifica la varianza de la distribución gaussiana que modela el ruido inducido por los receptores.

Los campos de **contador de tramas**, **contador de puntos** y **total de símbolos para cada punto** no son entradas para la simulación y sólo sirven para llevar un control de su evolución.

La interfaz permite ver también las señales procesadas durante la simulación, que influyen en las gráficas de BER como se presenta en la figura 28.

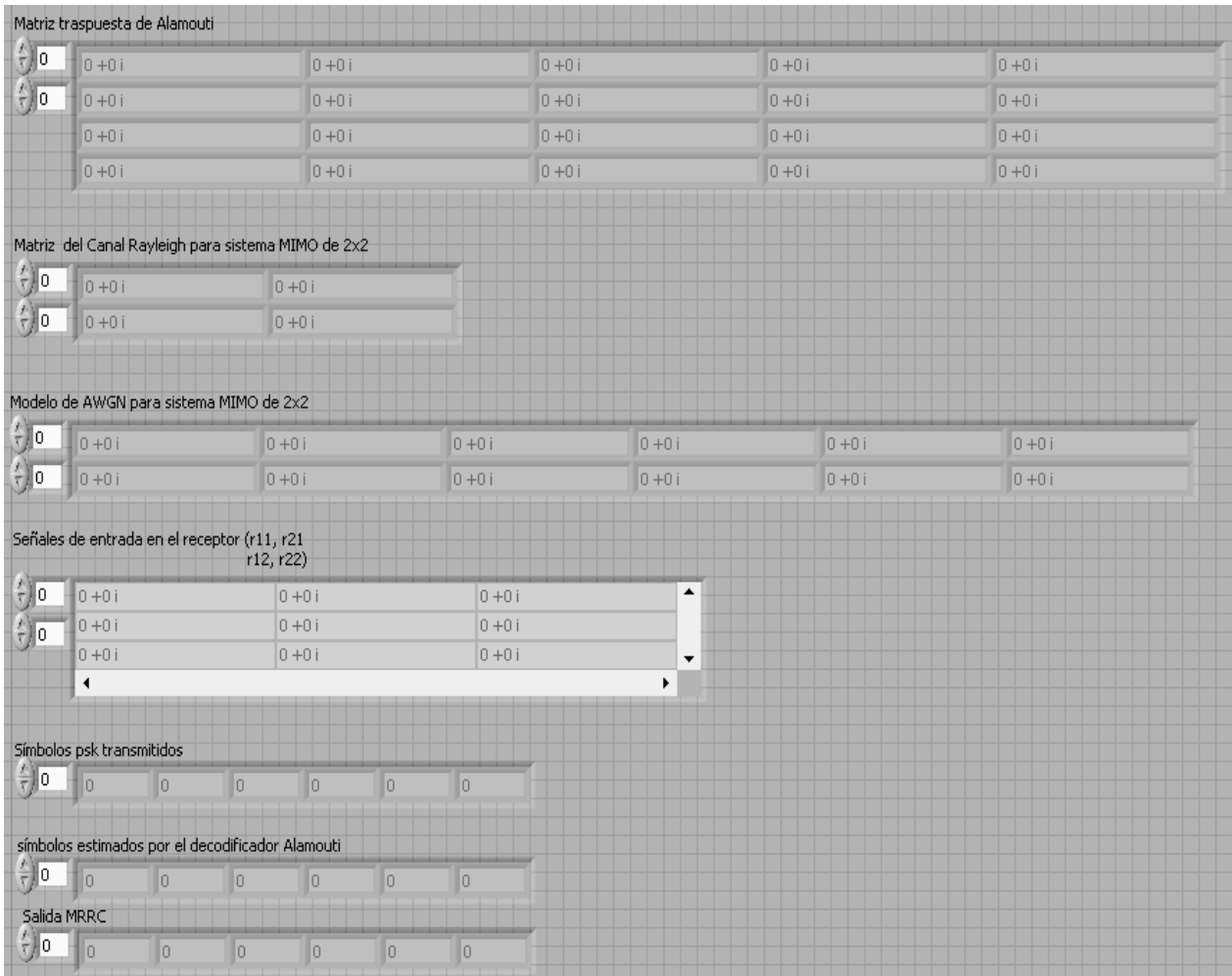


FIGURA 28. Interfaz con las señales procesadas durante la simulación

Otra parte de la interfaz permite ver las gráficas de BER obtenidas al finalizar la simulación.

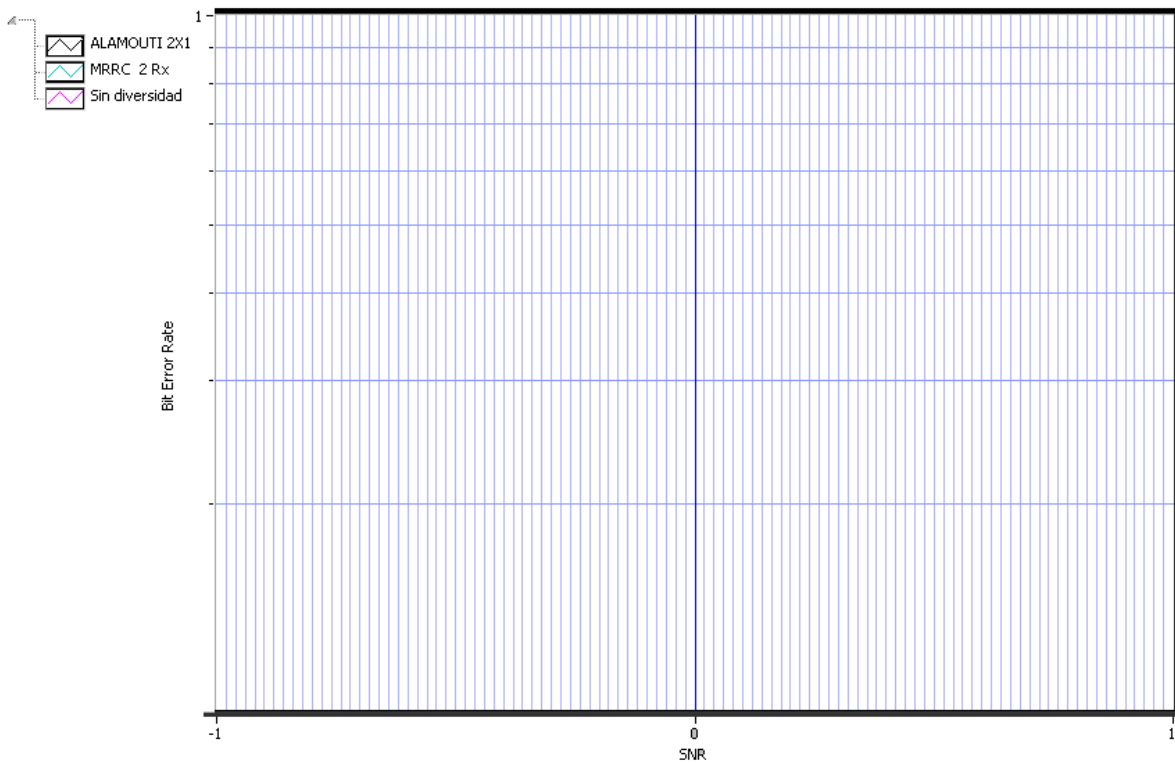


FIGURA 29. Interfaz para observar las gráficas de BER

Al momento de realizar simulaciones debe tenerse en cuenta que para alcanzar la mayor precisión posible en las gráficas, para lo que es recomendable fijar el número de símbolos por trama en 130 y el número de tramas de símbolos en 260.000. Con estos datos, modulación BPSK varianza de ruido 0,8, se obtiene una gráfica como la mostrada en la figura 30.

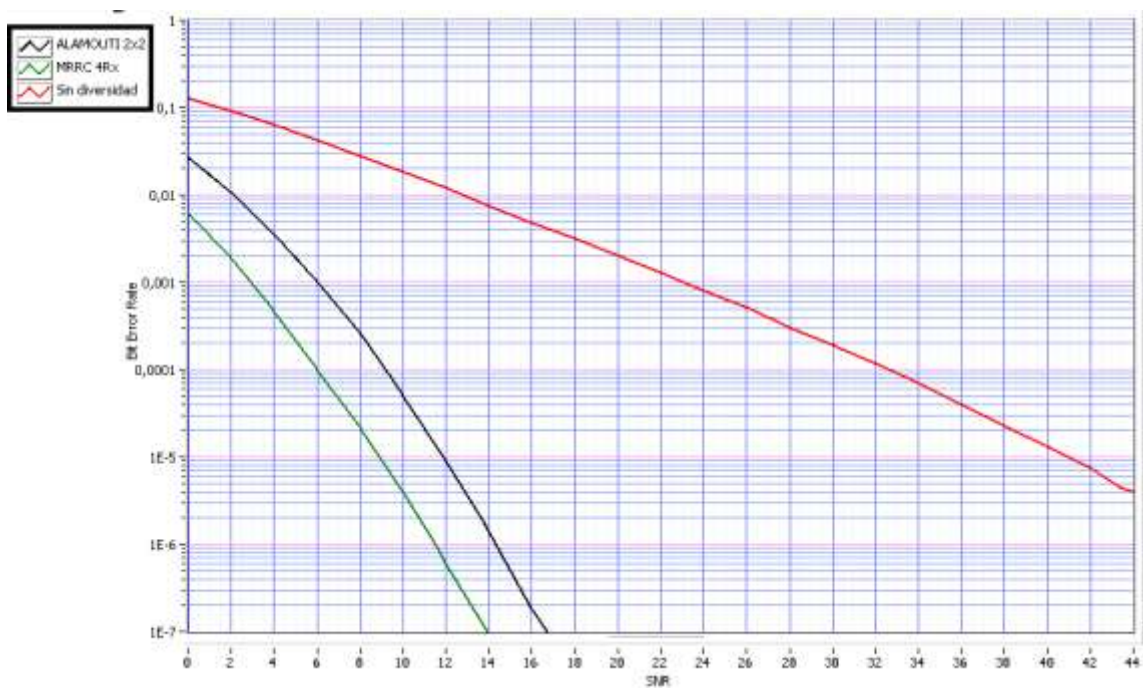


FIGURA 30. Gráfica de BER del entorno 2 y modulación BPSK

BIBLIOGRAFÍA

- [1] S. M. Alamouti, “*A simple transmit diversity technique for wireless communications*”, IEEE J. Select. Areas Commun., vol. 16, no. 8, pp. 1451–1458, Oct. 1998.
- [2] H. A. Roncancio, H. Cifuentes, “Tutorial de LabVIEW”, Universidad Distrital Francisco José de Caldas, 2001