

**ANALISIS DE MODULACION Y DEMODULACION EN
TELECOMUNICACIONES POR MEDIO DEL SOFTWARE VISSIM/COMM,
DIRIGIDO HACIA METODOS DE DETECCION DE ERRORES**

JUAN DAVID GARCIA ROJAS

FRANCISCO ALBERTO PABA SUAREZ



**Universidad
Tecnológica de Bolívar**
CARTAGENA DE INDIAS

**UNIVERSAD TECNOLÓGICA DE BOLIVAR
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERIA ELECTRONICA
CARTAGENA D.T Y C.**

2009

**ANALISIS DE MODULACION Y DEMODULACION EN
TELECOMUNICACIONES POR MEDIO DEL SOFTWARE VISSIM/COMM,
DIRIGIDO HACIA METODOS DE DETECCION DE ERRORES**

**JUAN DAVID GARCIA ROJAS
FRANCISCO ALBERTO PABA SUAREZ**

**MONOGRAFIA PRESENTADA COMO REQUISITO PARA OPTAR EL
TITULO DE INGENIERO ELECTRÓNICO**



**Universidad
Tecnológica de Bolívar**
CARTAGENA DE INDIAS

**ASESOR:
ING. EDUARDO GOMEZ VASQUEZ**

**UNIVERSAD TECNOLÓGICA DE BOLIVAR
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERIA ELECTRONICA
CARTAGENA D.T Y C.**

2009

Nota de aceptación.



**Universidad
Tecnológica de Bolívar**

CARTAGENA

Presidente del Jurado

Jurado

Jurado

Cartagena, Junio de 2009

A Dios creador del universo y dueño de mi vida que me permite construir otros mundos mentales posibles.

A mis padres, por el apoyo incondicional que me dieron a lo largo de la carrera.

A mis abuelos que me dieron el soporte, las bases y la fortaleza de mi formación personal.

A mis hermanos por enseñarme a apreciar la vida.

A todas las directivas de la Universidad Tecnológica de Bolívar, por su apoyo y colaboración para la realización de esta investigación.

A la Facultad de Ingeniería, por el soporte institucional dado para la realización de este trabajo.

Al Ingeniero Gonzalo López V por su asesoría y dirección en el trabajo de investigación.

A Juan David García Rojas compañero y amigo en estos años de formación.

MUCHAS GRACIAS

FRANCISCO PABA SUAREZ

Cartagena, Junio de 2009

A Dios, por todas las oportunidades que me ha brindado y por guiar mi vida en todo momento.

A mis padres, por el apoyo que me dieron a lo largo de todos los años de estudios universitarios.

A mi familia, que fueron de gran ayuda en todo mi proceso formativo.

A mis amigos, que estuvieron en las buenas y en las malas.

A la Facultad de Ingeniería, por toda la colaboración en lo concerniente a la realización de este documento.

Al Ingeniero Eduardo Gómez Vásquez por su asesoría y dirección en el trabajo de investigación.

A Francisco Paba Suárez Rojas compañero y amigo en estos años de formación.

MUCHAS GRACIAS

Juan David García Rojas

CONTENIDO

GLOSARIO	
RESUMEN	
INTRODUCCION	14
1. ENLACES DE TELECOMUNICACIONES	16
2. CODIFICACION	20
2.1 DETECCIÓN Y CORRECCIÓN DE ERRORES DE TRANSMISIÓN	20
2.1.1 Tipos de Errores	21
2.1.1.1. <i>Error de Bit</i>	21
2.1.1.2 <i>Error de Ráfaga</i>	22
2.2 CONCEPTO DE REDUNDANCIA	23
2.3 EL CRITERIO DE DETECCIÓN DE ERRORES	24
2.4 CODIGOS DE DETECCION Y CORRECCION DE ERRORES	27
2.4.1 Códigos De Bloque	27
2.4.1.1 <i>Distancia Hamming</i>	29
2.4.1.2 <i>Distancia Hamming mínima</i>	29
2.4.2 <i>Detección y Corrección De Errores Mediante Códigos De Bloque</i>	30
2.4.2.1 <i>Detección De Errores Con Códigos De Bloque</i>	30
2.4.2.2 <i>Corrección De Errores Con Códigos De Bloque</i>	31
2.4.3 Codificación Convolutacional	33
2.4.4 Codificacion Trellis	38
3. DECODIFICACIÓN	39
3.1 DECODIFICACIÓN DE VITERBI	39
4. GUIA VISSIM/COMM	41

4.1 SIMULACIÓN NIVEL I	41
4.1.1 Barra principal	42
4.1.2 Annotation Blocks	42
4.1.2.1 “Label”	43
4.1.2.2 “Date”	43
4.1.3 Arithmetics blocks	44
4.1.3.1 “Unit Conversion”	44
4.1.3.2 “Negate”	45
4.1.3.3 “Boolean Blocks”	46
4.1.4 “Consumer Blocks”	46
4.1.4.1 “Plot”	47
4.1.4.2 “Display”	48
4.1.5 “Producer blocks”	49
4.1.6 “Dynamic Blocks”	50
4.1.7 Sim Control	52
4.1.8 Main	53
4.1.9 Simulation Properties	54
4.1.10 Actividades Nivel I	59
4.2 SIMULACIÓN NIVEL II	60
4.2.1 Modulación Analógica	60
4.2.1.1 Modulación en Amplitud	62
4.2.1.2 Modulación en Frecuencia	64
4.2.2 Modulación digital	67
4.2.1 Modulación FSK y MSK	67
4.2.2 Modulación PSK	71
4.2.3 Actividades Nivel II	82



4.2.3.1 Modulación en amplitud	82
4.2.3.2 Modulación en Frecuencia	83
4.2.3.3 Modulación FSK	85
4.2.3.4 Modulación PSK y QPSK	86
4.3 SIMULACIÓN NIVEL III	90
4.3.1 Viterbi	90
4.3.2 Trellis	95
4.3.3 Código de Bloques	99
4.3.3.1 Interleaving de datos	99
4.3.4 Códigos Convolutivos	99
4.3.5 Actividad Nivel 3	104
BIBLIOGRAFIA	108
ANEXOS	



GLOSARIO

Canal AWGN

Un canal AWGN (additive white gaussian noise) es un canal que introduce un ruido en la señal con las siguientes propiedades: 1. Aditivo: el proceso de llegada al receptor es la suma de la entrada al canal y un proceso llamado ruido. 2. Blanco: el ruido tiene un espectro constante (afecta igual a todas las frecuencias). 3. Gaussiano: la función de densidad de probabilidad del ruido es una Normal.

Canal Simétrico Binario (BSC)

Este modelo asume que todos los errores en los bits son igualmente probables y que los errores ocurren con una probabilidad fija que se especifica por el usuario.

CUANTIZACION

La cuantización permite decidir cual código digital es el más adecuado para la representación del valor continuo de la señal muestreada.

DISTANCIA HAMMING

Se denomina así a la efectividad de los códigos de bloque y depende de la diferencia entre una palabra de código válida y otra. Cuanto mayor sea esta diferencia, menor es la posibilidad de que un código válido se transforme en otro código válido por una serie de errores.

Entrelazado (interleaving)

Es una técnica consistente en organizar la información digital de forma no contigua para mejorar las prestaciones de un sistema.

MUESTREO

Proceso mediante el cual la información de la fuente es convertida a muestras discretas tomadas a intervalos regulares. La mínima frecuencia a la cual estas muestras deben ser tomadas es una frecuencia al menos igual a 2 veces la frecuencia máxima de la información.

ONDA PORTADORA

Una onda portadora es una forma de onda, generalmente senoidal, que es modulada por una señal que se quiere transmitir. Esta onda portadora es de una frecuencia mucho más alta que la de la señal moduladora.

PNSEQUENCE: (PSEUDO NOISE SEQUENCE)

También conocido como pseudo random binary sequence (PRBS), es un generador de secuencias binarias aleatorias.

SEÑALES

Una señal es una cantidad eléctrica que está definida por tres características principales; Amplitud, fase y frecuencia.

SEÑAL EN BANDA BASE

Se denomina banda base al conjunto de señales que no sufren ningún proceso de modulación a la salida de la fuente que las origina, es decir son señales que son transmitidas en su frecuencia original.

RESUMEN

Este documento presenta al lector una posibilidad de obtener un rápido aprendizaje sobre el manejo del software de modelado y simulación VisSimComm, partiendo de una base teórica en la cual se trabajan temas específicos de las Telecomunicaciones. Centrándose en los enlaces de telecomunicaciones como una trayectoria que debe recorrer la información para transmitirse de un lugar de origen a un lugar destino, haciendo énfasis en los siguientes elementos: fuentes de información, Codificador, Modulador, Canal de Transmisión, Demodulador, Decodificador y Recepción de la Información

Teniendo en cuenta que VisSim es un interfaz visual que ofrece al usuario un método muy simple para desarrollar, construir, modificar y mantener modelos complejos, que además posee un motor de simulación que proporciona rápidas y precisas soluciones para diseños de sistemas lineales, no lineales, sistemas continuos y discretos e híbridos, se procede a desarrollar un proceso explicativo en el cual por medio de una detallada clasificación se logra proponer un método de aprendizaje en el cual los usuarios además de conocer todas las características que este potente software de simulación presenta, pondrá en práctica todas sus habilidades y conocimientos en cuanto al ámbito de las telecomunicaciones.

De esta forma se diseñan tres niveles de aprendizajes descritos así:

Nivel 1: Inicio de VisSimComm, Explicación detallada Barras Principales, Barras de Trabajo y Actividades Nivel 1.

Nivel 2: Modulación Análoga (AM, FM y PM), Modulación Digital (FSK, MSK, PSK, QAM y BPSK) y Actividades Nivel 2.

Nivel 3: Codificación y decodificación por medio del código Viterbi, Codificación y decodificación por medio del código de Trellis, Codificación y decodificación por medio del código de Gray, Codificación convolucional y Actividades Nivel 3.

INTRODUCCION

Este documento contiene un acercamiento al software de simulación VisSim, enfocado a los sistemas de comunicación, de esta manera se trabaja paralelamente un constante aprendizaje de todas las posibilidades que nos presenta VisSim como software de modelado de sistemas dinámicos complejos combinado con elementos netamente de las telecomunicaciones.

El propósito de este documento es introducir al lector los conocimientos básicos necesarios para comprender el ambiente de VisSim y poder utilizar sus bloques para realizar configuraciones concernientes al ámbito de las telecomunicaciones, convirtiendo así a VisSim en un engranaje con la parte teórica principalmente en cuanto a la modulación, demodulación y métodos de detección y corrección de errores.

VisSim se presenta como una opción para mejorar el aprendizaje de los estudiantes de telecomunicaciones ya que brinda un soporte para integrar la teoría con la práctica, situación que es indispensable para una buena comprensión, teniendo en cuenta que VisSimCOMM es una de las herramientas de modelado más potentes que nos presenta el mercado hoy en día, de allí la importancia de un documento que explore las herramientas principales del software ahorrando así tiempo valioso para cualquier tipo de usuario que tenga necesidad de hacer uso del programa.

Utilizando VisSim/Comm, los ingenieros pueden moverse fácilmente entre las distintas fases de construcción, simulación, optimización y validación del modelo. Esta plataforma de desarrollo altamente integrada permite a los ingenieros simular y visualizar las señales en cualquiera de las fases de la

cadena del sistema de comunicaciones. Además, los ingenieros pueden completar todas las tareas de modelado

Este documento trabaja varias secciones así:

-Sección de introducción a las telecomunicaciones en la cual se tratan los aspectos teóricos principales de modulación, demodulación y métodos de detección y corrección de errores, para familiarizarnos con términos y características de cada tema que se trate más adelante.

-Sección de la guía de trabajo de Vissim/Comm aplicada a través de 3 niveles de complejidad progresiva así:

-Nivel 1, en este nivel se tratan los aspectos básicos de VisSim, como introducción a los bloques y simulación de los mismo, así como un breve repaso por las barras principales del simulador.

-Nivel 2, en este nivel se trabajaran aspectos del software relacionados con la modulación y demodulación análogas y digitales, basándose en ejemplos y actividades propuestas para profundizar los conocimientos adquiridos.

-Nivel 3, en este ultimo nivel, se tratan los temas de detección y corrección de errores por medio de ejemplos simulados en VisSimCOMM y una serie de actividades propuestas que profundizan los conocimientos adquiridos durante todo el proceso.

1. ENLACES DE TELECOMUNICACIONES

Un enlace de comunicación, es todo trayecto que debe recorrer la información para transmitirse de un lugar origen a un lugar destino, incluye como mínimo, tres elementos claves:

- Un transmisor.
- Un medio de comunicación (canal)
- Un receptor.

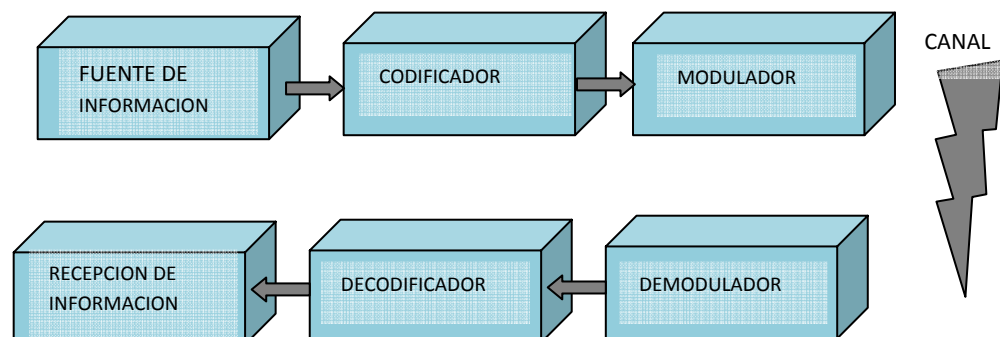
La posibilidad de simular estos tres elementos es necesaria para lograr diseñar modelos o sistemas de comunicación exitosos.

Los elementos transmisores y receptores pueden ser subdivididos en otros elementos. En el transmisor, se incluye la fuente de información (análoga o digital), un codificador opcional, un modulador.

En el receptor, se incluye un demodulador, un decodificador (opcional) y un enlace de comunicación.

La estructura anteriormente descrita, puede ser representada mediante la figura 1.1:

FIGURA 1.1 Estructura De Un Enlace De Comunicación.



El enlace de comunicación se desarrolla de la siguiente manera:

La **fente de Información** genera una señal de información que requiere ser enviada a un receptor en particular. Esta señal puede ser analógica como la voz, o digital como una sucesión de datos binarios. Esta señal es típicamente de banda base representada por un voltaje.

En el caso de una señal analógica, a menudo es deseable transformarla a un formato digital antes de proceder a la transmisión. Ello se logra mediante la conversión análoga-digital (A/D); lo cual se realiza con un chip A/D, el cual posee sub-bloques de muestreo, retención, cuantización y conversión. Este formateo convierte la señal analógica en una digital. Durante este proceso de muestreo y cuantización algo de información es perdida (error de cuantificación); pero, la señal digital resultante es menos susceptible a los efectos de ruido al viajar por el canal de transmisión.

Por otro lado, toda información proveniente de una fuente puede poseer redundancia. Ello puede aprovecharse para disminuir el número de bits para representar un nivel determinado de datos. Lo anterior lleva a que es posible aumentar la tasa de transmisión sin grandes cambios. El bloque que logra tal milagro se denomina **codificador de fuente**.

Un problema diferente que se debe prever es la protección de los datos antes de entrar al canal. Una estrategia simple es agregar redundancia al flujo de datos digitales, ello se logra agregando datos adicionales, que en cierto modo proporciona una capacidad de corrección de errores en el receptor. Este proceso se denomina **codificación de canal**. Existen muchos tipos de codificadores, uno de ellos es el denominado corrección de error hacia adelante (FEC) FORWARD ERROR CORRECTION. Entre los esquemas FEC más populares se tienen la codificación convolucional, codificación en bloques y la Trellis. Es importante destacar que normalmente la tasa de salida de tal codificador no es igual a la tasa de entrada. Para distinguir apropiadamente entre las dos tasas, la tasa de entrada del transmisor se denomina tasa de

datos de la información, mientras la tasa de salida del transmisor, se designa por tasa de datos del canal.

El bloque siguiente es el **modulador**, su elección depende del tipo de señal de información y el medio de transmisión empleado. La modulación se refiere a la técnica empleada para representar la señal de información y es la que físicamente se transmite al receptor. Por ejemplo, en la Modulación de Amplitud (AM), la información es representada por variaciones de amplitud de la señal portadora.

Una vez que la señal se modula, se envía a través del medio de transmisión, también conocido como **canal**, para alcanzar en el otro extremo al receptor. Éste puede ser un par de alambre de cobre, cable coaxial, o la atmósfera en el caso de una transmisión de radio. En general, todos los canales introducen alguna forma de distorsión a la señal original. Se han desarrollado muchos modelos de canal para representar tales distorsiones en forma matemática. Un modelo de canal normalmente usado es el Canal con Ruido Gaussiano Blanco Aditivo (AWGN). En este canal, un ruido con densidad espectral de potencia uniforme (de allí el término “blanco”) es adicionado a la señal de modulada. Otros tipos de canales incluyen el desvanecimiento y multitrayectoria.

Cuando la señal transmitida alcanza el receptor, sufre un proceso de **demodulación**. Este paso es inverso al de la modulación y se refiere al proceso requerido para extraer la señal de información original desde la señal modulada. La demodulación también incluye cualquier paso asociado con sincronización de la señal, como el uso de lazo enclavados en fase que logran la coherencia de la fase entre la señal entrante y el oscilador local del receptor. Este tipo de demodulación se denomina coherente, se le coloca nuevamente la señal portadora y luego se filtra para retener la señal en banda base.

Cuando se utiliza codificación de canal en el transmisor, debe incluirse un bloque de **decodificación** que debe situarse antes de recuperar la señal de datos original. Tal proceso de decodificación, normalmente es más complicado que el proceso de codificación y puede ser computacionalmente muy intensivo.

Se han desarrollado los últimos años esquemas de decodificación muy eficaces - un ejemplo es el algoritmo de Viterbi que se usa convolucionalmente para decodificar los datos.

Finalmente, una estimación de la señal original se produce a la salida del receptor.

Para la realización de esta guía, se tratarán los bloques centrales del enlace, a saber, la codificación, la modulación, demodulación y decodificación.



Universidad
Tecnológica de Bolívar
CARTAGENA DE INDIAS

2. CODIFICACION

En ocasiones se producen diferencias entre las secuencias de datos enviadas a través de un canal y las secuencias de datos recibidas debidas a la existencia de ruido en el canal. A estas diferencias se les denomina errores¹. Por ello es necesario realizar una codificación a la entrada del canal, cuyo objetivo es que el receptor sea capaz de detectar y corregir los errores producidos en los datos durante su transmisión por el canal.

La codificación del canal consiste en introducir redundancia, de forma que sea posible reconstruir la secuencia de datos original de la forma más fiable posible.

Hay dos técnicas de corrección de errores:

- Detección de errores o corrección hacia atrás o ARQ (Automatic Repeat Request): Cuando el receptor detecta un error solicita al emisor la repetición del bloque de datos transmitido. El emisor retransmitirá los datos tantas veces como sea necesario hasta que los datos se reciban sin errores.
- Corrección de errores o corrección hacia delante o FEC(Forward Error Correction): Se basa en el uso de códigos autocorrectores que permiten la corrección de errores en el receptor.

2.1 DETECCIÓN Y CORRECCIÓN DE ERRORES DE TRANSMISIÓN

Las redes deben ser capaces de transmitir datos de un dispositivo a otro con cierto nivel de precisión. Para muchas aplicaciones, el sistema debe garantizar que los datos recibidos son iguales a los transmitidos. Sin embargo, siempre que una señal electromagnética fluye de un punto a otro, está sujeta a interferencias impredecibles debido al calor, el magnetismo y diversas formas de electricidad. Esta interferencia puede cambiar la forma o la temporización

¹ <http://www.isa.cie.uva.es/proyectos/codec/marco1.html>

de la señal. Si la señal transporta datos binarios codificados, tales cambios pueden alterar su significado.

Las aplicaciones requieren entonces un mecanismo que permita detectar y corregir los posibles errores ocurridos durante la transmisión. Algunas aplicaciones tienen cierta tolerancia de errores (ej. transmisión de audio/video), mientras que para otras aplicaciones se espera un alto nivel de precisión (ej. transmisión de archivos).

2.1.1 Tipos de Errores

Antes de estudiar los mecanismos que permiten la detección y corrección de errores, es importante entender cuáles son esos posibles errores.

2.1.1.1. Error de Bit

Este término significa que únicamente un bit de una unidad de datos determinada (byte, carácter, paquete, etc.) cambia de 0 a 1 o de 1 a 0. Para comprender el impacto de este cambio, podemos imaginar que cada grupo de 8 bits es un carácter ASCII con un 0 añadido a la izquierda.

Un error de bit podría alterar completamente el carácter ASCII enviado (ej. 'A': ASCII 65) y en el receptor se obtendría un carácter completamente diferente (ej. 'l': ASCII 73).

Los errores en un único bit son el tipo de error menos probable en la transmisión de datos en serie.

Imagine que un emisor envía datos a 1Mbps. Esto nos dice que cada bit dura únicamente 1/1000000 seg. Para que ocurra un error de bit, el ruido debe tener una duración de sólo 1µseg, lo que es muy raro. Sin embargo, puede ocurrir un error de bit si se están enviando los datos usando transmisión paralela. Por



ejemplo, si se usan 8 cables para enviar los 8 bits de un byte al mismo tiempo, y uno de los cables es ruidoso, se puede corromper un bit de cada byte.²

2.1.1.2 Error de Ráfaga

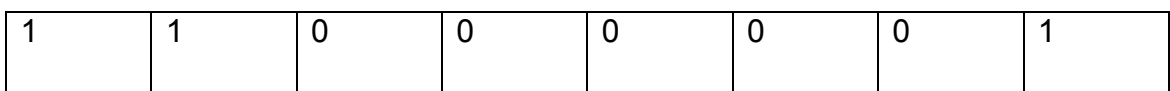
Significa que dos o más bits de la unidad de datos han sido alterados. Es importante notar que los errores de ráfaga (Figura 2.1) no implican que se afecten bits consecutivos. La longitud de la ráfaga se mide desde el primer hasta el último bit incorrecto. Algunos bits intermedios pueden no estar afectados.

Byte enviado:

Figura 2.1. Representación de error de ráfaga (5 bits)



Byte recibido:



En este caso, la longitud de la ráfaga sería 5, porque es la distancia en bits desde el primer bit erróneo hasta el último. Dentro de la ráfaga puede haber bits correctos y/o erróneos.

La presencia de errores de ráfaga es más probable en las transmisiones en serie. La duración del ruido es normalmente mayor que la duración del bit, lo que significa que cuando el ruido afecta los datos, afecta un conjunto de bits. El

² Karima Velásquez C, 7ª ed. 2004, Control de Errores, Stallings, W. Comunicaciones y Redes de Computadores. Prentice Hall.

número de bits afectados dependerá de la tasa de datos y de la duración del ruido.

2.2 Concepto de Redundancia

Una vez que se conocen los tipos de errores que pueden existir, es necesario identificarlos. En un entorno de comunicación de datos no se tendrá una copia de los datos originales que permita comparar los datos recibidos para detectar si hubo errores en la transmisión. En este caso, no habrá forma de detectar si ocurrió un error hasta que se haya decodificado la transmisión y se vea que no tienen sentido los datos recibidos. Si se comprobaran errores de esta forma, sería un proceso muy lento y costoso. Es necesario un mecanismo que sea sencillo y completamente efectivo.

El concepto clave para detectar o corregir errores es la *redundancia*. Para esto es necesario enviar bits extra junto con los datos. Estos bits son añadidos por el emisor y eliminados por el receptor, permitiendo detectar y posiblemente corregir los bits afectados.

Un mecanismo de detección de errores que podría satisfacer los requisitos antes expuestos sería enviar dos veces cada unidad de datos. El dispositivo receptor podría entonces comparar ambas copias bit a bit. Cualquier discrepancia indicaría un error y se podría corregir mediante un mecanismo apropiado. Este sistema sería extremadamente lento. No solamente se doblaría el tiempo de transmisión, sino que además habría que añadir el tiempo necesario para comparar cada unidad bit a bit.

El concepto de incluir información extra en la transmisión con el único propósito de comparar es bueno. Pero en lugar de repetir todo el flujo de datos, se puede añadir un grupo más pequeño de bits al final de cada unidad. Esta técnica se denomina *redundancia* porque los bits extra son redundantes a la información,

descartándose tan pronto como se ha comprobado la exactitud de la transmisión.

2.3 El criterio de detección de errores

Un error es detectado de acuerdo a las siguientes características que lo determinan como tal:

Tasa de error de bit: BER

Probabilidad de una trama sin errores: PF

Probabilidad de una trama con errores no detectables: Po

Probabilidad de una trama con errores detectables o sin errores indetectables: Pe

Primero se considera el caso en el que no se ponen medidas para detectar errores. En ese caso la probabilidad de errores detectables es cero.

Suponiendo que los sistemas son estacionarios (las distribuciones de probabilidad no modifican sus parámetros en el tiempo), tenemos que:

$$PF=(1 - BER)L$$

$$Po= 1 - Pe$$

Donde L es la longitud de una secuencia dada. De ahí se concluye que la probabilidad que una secuencia llegue sin errores disminuye al aumentar la longitud de la trama.

Si el valor BER no es aceptable en un sistema, entonces se puede hacer lo siguiente:

- **Incrementar la potencia en el transmisor:** Aumentando la relación señal-ruido (SNR) se reducirá el BER.
- **Diversidad:** Al sumar 2 señales que proceden de 2 canales independientes, la relación señal-ruido se duplica (mejoramiento de 3dB) en un canal Gaussiano y un mejor incremento se obtiene en canales con desvanecimiento (fading channels). De esta forma el BER es reducido. Esto se puede lograr a través de diversidad espacial (antenas), diversidad en frecuencia o diversidad en tiempo. En la diversidad espacial, la señal es recibida por 2 antenas que se encuentran separadas por una distancia lo suficientemente grande para que las señales recibidas puedan ser consideradas independientes. En el caso de la diversidad en frecuencia, la señal es transmitida a través de 2 canales separados en frecuencia. Finalmente, la diversidad en tiempo corresponde a la transmisión de la señal 2 veces.
- **Transmisión Duplex:** En una sistema de transmisión dúplex, el receptor regresa el mensaje al transmisor con el objetivo de que el transmisor pueda verificar que la información recibida es correcta. Este tipo de comunicación requiere el doble del ancho de banda requerido en un sistema simplex.
- **Solicitud de Repetición Automática:** En esta técnica, se utiliza un código para la detección de error. Cuando un error se detecta, el receptor solicita al transmisor re-transmitir el bloque donde se encontró el error. Esta técnica tiene desventajas similares a la técnica de transmisión dúplex aunque no requiere el doble del ancho de banda.
- **Codificación para la Corrección de Errores (FEC):** En esta técnica la señal de información es codificada de tal manera que los errores puedan ser corregidos agregando extra bits a la señal de información. Si se desea mantener la tasa de transmisión de datos, estos bits adicionales

representan aumentar el ancho de banda requerido. Sin embargo, si se desean lograr bajas tasas de error entonces este requerimiento es aceptable. FEC es utilizado en CD, almacenamiento de datos, enlaces por radio frecuencia, modems, comunicaciones espaciales (estudio del universo).

La detección de errores se basa en el siguiente principio: Dada una secuencia, se añaden bits adicionales por parte del transmisor para formar un código que tenga capacidad de detectar errores³.

Este código se calculará en función de los otros símbolos que se vayan a transmitir. El receptor realizará el mismo cálculo y comparará los dos resultados. Se detecta un error si y sólo si, los dos resultados no coinciden. Por lo tanto se incluye el valor de P_e como la probabilidad que si una trama contiene errores, el sistema de detección de errores los detecte. Aquí P_o sería la tasa de error residual (RER), y es la probabilidad que un error no se detecte aunque se esté usando un esquema de detección de errores. Los esquemas más comúnmente usados para detección de errores son: Paridad y CRC.

Las limitaciones teóricas de la codificación están dadas por los resultados de la teoría de la información. Este resultado es frustrante ya que ofrece una pequeña pista acerca de cómo el código debería realizarse. El código de detección de errores se diseña para permitir la detección de errores. Una vez detectado, el receptor puede pedir una re-transmisión de los bits erróneos, o simplemente informar al destinatario que la transmisión fue corrupta. En un canal binario, los códigos de verificación de errores se llaman Códigos de Detección de Paridad. El receptor asignaría a la frase recibida otra frase codificada pre asignada tal que minimice la distancia Hamming entre las dos.

³ Forouzan, b. data communications and networking. mc graw hill. 4thed. 2007, pag 217.

2.4 CODIGOS DE DETECCION Y CORRECCION DE ERRORES

La codificación para corregir errores es más sofisticada que la codificación para detectar errores.

Su objetivo es descubrir y localizar los errores en la transmisión. Una vez localizado, la corrección es trivial: Se invierte el valor del bit. La codificación de corrección de errores requiere una menor proporción de codificación que la detección del error. Se puede demostrar que para corregir hasta n bits erróneos, un esquema de codificación requiere el uso de frases codificadas pre asignadas con una distancia Hamming de por lo menos $2n + 1$. Esta codificación no es común en comunicaciones terrestres donde normalmente se obtiene un mejor desempeño con detección y retransmisión de errores. Sin embargo, en las comunicaciones satelitales, a menudo el retardo por propagación significa que muchos comandos o tramas pueden ser transmitidos antes de recibir una indicación de retransmisión. Esto puede hacer la tarea de proceso de datos algo muy complejo. La

Transmisión en tiempo real evita a menudo la retransmisión. Es necesario entonces realizar una corrección desde la primera vez. En estas circunstancias especiales, el ancho de banda adicional requerido para incluir bits de verificación redundantes es un precio aceptable. Existen dos tipos de técnicas: Códigos de Bloque y Códigos Convolucionales.

2.4.1 CODIGOS DE BLOQUE

Los códigos de bloque son técnicas utilizadas para transformar un conjunto de datos binarios " N " en otro un poco más largo " K " donde se agrega unos bits de mas para dar redundancia al código saliente K donde $(K > N)$. el numero de

dígitos de comprobación o redundancia será $M=K-N$; donde M son la cantidad dígitos adicionados.

El principio que se utiliza en los códigos de bloque consiste en estructurar los datos en bloques de longitud fija y añadir a cada bloque un cierto número de bits llamados bits de redundancia.

Sólo ciertas combinaciones de bits son aceptables y forman una colección de palabras de código válidas.

Los códigos de bloque es una técnica de codificación que agrupa la señal de información en bloques de bits y a estos se le agregan bits de código (información redundante-bits de paridad). Esto da como resultado un bloque con longitud de n bits, donde $n > k$. La **tasa o razón de codificación** es, por lo tanto k/n . El número de bits de paridad es entonces $n-k$.

Un código de bloque se genera cuando la secuencia de información de longitud K se procesa dentro de un registro de longitud fija L , donde $L < K$ (ver Figura 2.2).

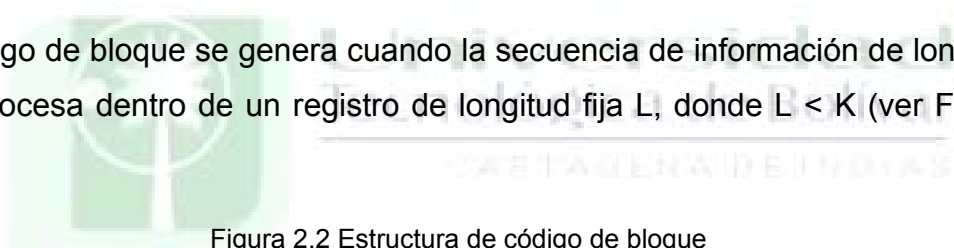
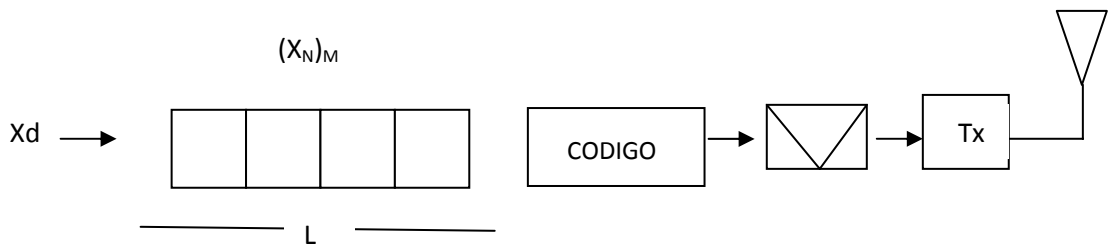


Figura 2.2 Estructura de código de bloque



Este bloque de tamaño L se presenta como el codificador del canal, el cual genera uno de los M posibles símbolos del alfabeto $\{X_k\}_1, \{X_k\}_2, \dots, \{X_k\}_M$, $k = 1, \dots, L$, donde $M = D^L$ y D es la longitud del alfabeto básico de la secuencias (el cual se considera binario ($D = 2$), si no se expresa otra cosa). El código de bloques queda entonces compuesto por un vector de Dimensiones $X(K,L) \times L$, donde L es el numero de columnas y $X(K,L)$, el cual es una función dependiente de la técnica utilizada, es el numero de filas. En la mayoría de los casos $X(K,L) = K$.

La tasa de codificación es la relación K/L , y la redundancia del código es $1-K/L$. La habilidad de detectar errores depende de la tasa. Una baja tasa tiene una probabilidad de detección alta, pero una redundancia alta.

Para entender la manera en que pueden manejarse los errores, es necesario estudiar de cerca cómo se codifican los datos. Por lo general, una unidad de datos (generalmente llamada en este ambiente *trama*) consiste de m bits de datos y r bits redundantes usados para la verificación, siendo la longitud total de una trama n ($n = m + r$). A la unidad de n bits que contiene datos y bits de redundancia se le conoce como *palabra codificada*. La cantidad de bits de redundancia y la robustez del proceso son factores importantes del esquema de codificación.

2.4.1.1 Distancia Hamming

Para empezar se define un concepto de utilidad. Se define la *distancia Hamming* $d(v_1, v_2)$ entre dos palabras codificadas de n bits v_1 y v_2 , como el número de bits en el que v_1 y v_2 difieren. Por ejemplo:

$v_1 = 10001001$; $v_2 = 10110001$ | entonces, $d(v_1, v_2) = 3$

2.4.1.2 Distancia Hamming mínima

Se llama *distancia Hamming mínima* a la distancia Hamming más pequeña entre todos los posibles pares de palabras codificadas de un esquema de codificación. Se usa el término *dmin* para definir la distancia Hamming mínima en un esquema de codificación. Para hallar este valor, se deben encontrar las distancias Hamming entre todas las palabras codificadas del esquema, y se selecciona la más pequeña.

2.4.2 Detección y corrección de errores mediante códigos de bloque

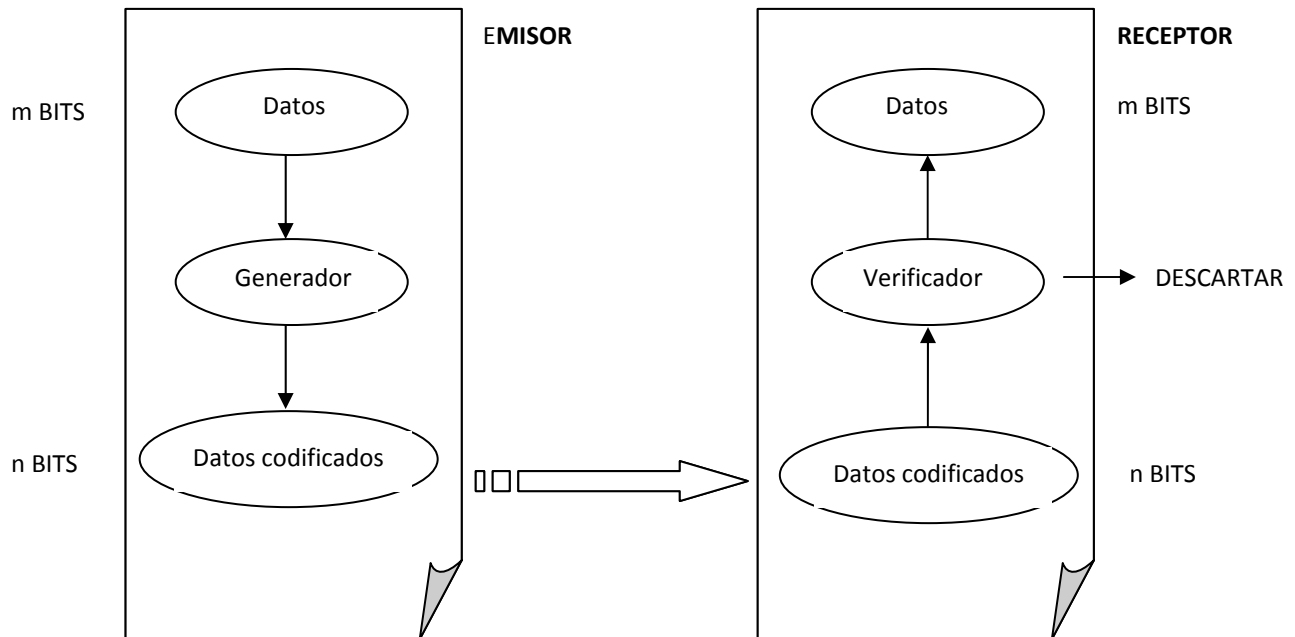
Las *palabras de datos* de longitud m bits no se transmiten directamente, sino que son previamente transformadas en *palabras codificadas* de n bits. Con m bits se pueden crear hasta 2^m palabras de datos, y con n bits se pueden crear hasta 2^n palabras codificadas. Como $n > m$, el número de palabras codificadas es mayor al número de palabras de datos. El proceso de codificación en bloques es uno a uno: la misma palabra de datos es transformada siempre en la misma palabra codificada. Las palabras codificadas obtenidas a partir de una palabra de datos son llamadas *válidas*. Esto significa que se tendrán $2^n - 2^m$ palabras codificadas que no serán utilizadas. Estas palabras codificadas son llamadas *inválidas*.

2.4.2.1 Detección de errores con códigos de bloque

Ahora, ¿cómo puede usarse la codificación por bloques para detectar errores? Si se cumplen las siguientes dos condiciones, el receptor será capaz de detectar variaciones en la palabra codificada original:

1. El receptor tiene la lista de las palabras codificadas válidas.
2. La palabra codificada válida transmitida ha sido modificada a una inválida.

Figura 2.3. Esquema de detección de errores

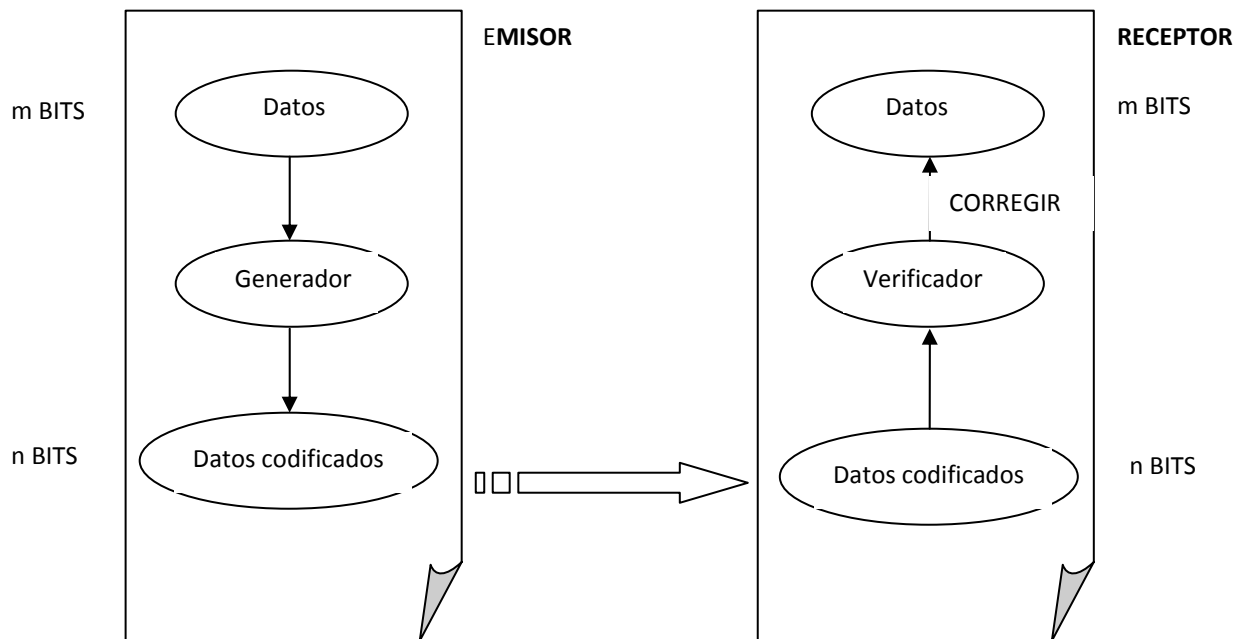


El emisor crea palabras codificadas a partir de palabras de datos usando un generador que aplica reglas y procedimientos de codificación específicos del esquema empleado. Cada palabra codificada que es enviada al receptor puede variar durante la transmisión. Si la palabra codificada recibida no es válida, es descartada. Sin embargo, si la palabra codificada es modificada como otra palabra codificada válida durante la transmisión, el error no será detectado.

2.4.2.2 Corrección de errores con códigos de bloque

En el caso discutido previamente (detección de errores, ver imagen 2.3), el receptor sólo necesita saber que la palabra codificada es inválida para detectar un error. En la corrección de errores, el receptor deberá descubrir la palabra codificada originalmente enviada. La idea principal es la misma que la empleada en la detección de errores, pero el verificador es mucho más complejo.

Figura 2.4. Esquema de corrección de errores



Una vez que se recibe una palabra inválida, el receptor calcula la distancia Hamming entre la palabra recibida y las palabras válidas. La menor de las distancias calculadas indica cual es la palabra codificada válida que el emisor originalmente transmitió. Si dos o más palabras válidas generan el mismo valor, que resulta ser el mínimo, entonces el error no puede ser corregido y la palabra recibida se descarta (ver imagen 2.4).

En la mayoría de las aplicaciones de transmisión 2^m palabras son válidas, pero como se ha visto, debido a la manera en que se codifican no se usan las 2^n palabras codificadas posibles. Es viable entonces hacer una lista de las palabras codificadas válidas y encontrar las dos cuya distancia Hamming sea mínima. Esta será la distancia Hamming de todo el código.

Las propiedades de detección y corrección de errores de un código dependen de su distancia Hamming. Si dos palabras codificadas están separadas una

distancia Hamming d , se requerirán d errores de un bit para convertir una en otra.

Para detectar d errores se necesita un código de distancia $d + 1$, pues con tal código no habrá manera de que d errores de bit puedan cambiar una palabra codificada válida a otra. Cuando el receptor encuentra una palabra codificada no válida, sabe que ha ocurrido un error de transmisión.

De manera similar, para corregir d errores se necesita un código de distancia $2d + 1$, pues así las palabras codificadas válidas estarán tan separadas que, aún con d cambios, la palabra codificada original sigue estando más cercana que cualquier otra palabra codificada, por lo que puede determinarse de manera única.

2.4.3 Codificación Convolutiva

En las transmisiones digitales actuales la codificación es un proceso muy elaborado, en la figura 2.5 se observa un diagrama de bloques general.

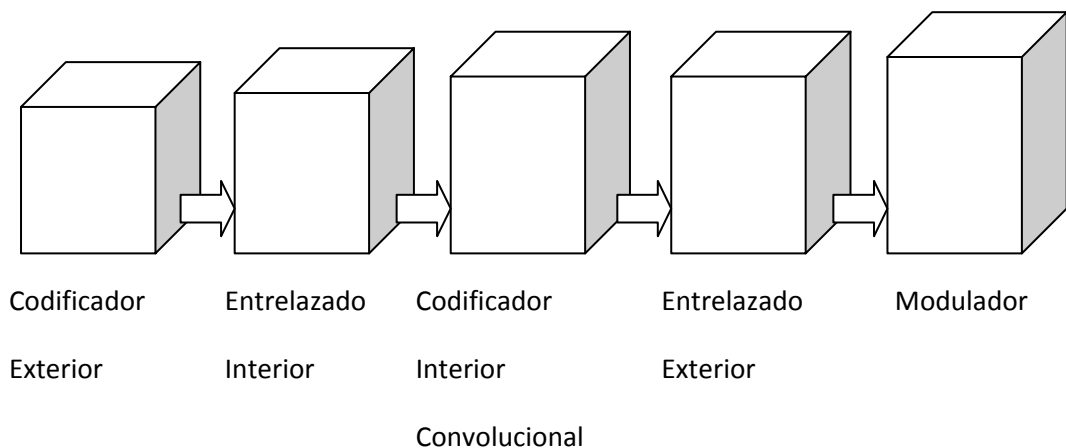


Figura 2.5. Esquema codificación convolutiva con interleaving

Este sistema contiene dos codificadores propiamente dichos: uno exterior de bloques, y uno interior convolucional. Estos dos codificadores introducen redundancia en los datos con objeto de poder detectar o corregir errores de transmisión. El sistema contiene además dos entrelazados (interleaving) cuyo objetivo es hacer que bloques correlativos de bits no se transmitan juntos, es decir realiza una Multiplexación de la información. Este es el esquema de codificación empleado en la televisión digital DVB (Digital Video Broadcasting)⁴. El esquema de codificación utilizado en la telefonía móvil GSM es similar, pues usa un codificador de bloques, un codificador convolucional y un entrelazado. Los códigos convolucionales son adecuados para usar sobre canales con mucho ruido (alta probabilidad de error).⁵

El sistema tiene memoria: la codificación actual depende de los datos que se envían ahora y que se enviaron en el pasado.

Un código convolucional queda especificado por tres parámetros (n,k,m):

n es el número de bits de la palabra codificada

k es el número de bits de la palabra de datos

m es la memoria del código o longitud restringida

La codificación convolucional es una codificación continua en la que la secuencia de bits codificada depende de los bits previos. El codificador consta de un registro de desplazamiento de K segmentos de longitud k (en total kK) que se desplaza k posiciones por ciclo y genera n funciones EXOR también por ciclo. La tasa de codificación es, entonces, $R=k/n$.

⁴ RODEN MARTIN. 1998. Digital Communication System Design, Ed. Prentice Hall.

⁵ LABTEL TM. 2002. LABORATORIO DE TEORÍA DE INFORMACIÓN EXPERIENCIA N° 1- N° 7. [libro en línea] <http://www.labtel.cl/ing-civil/>.

Proceso de codificación

El proceso de codificación de estos códigos se realiza utilizando un dispositivo lógico en el codificador.

Ejemplo: Codificador convolucional

Figura 2.6. Registro de desplazamiento codificador convolucional

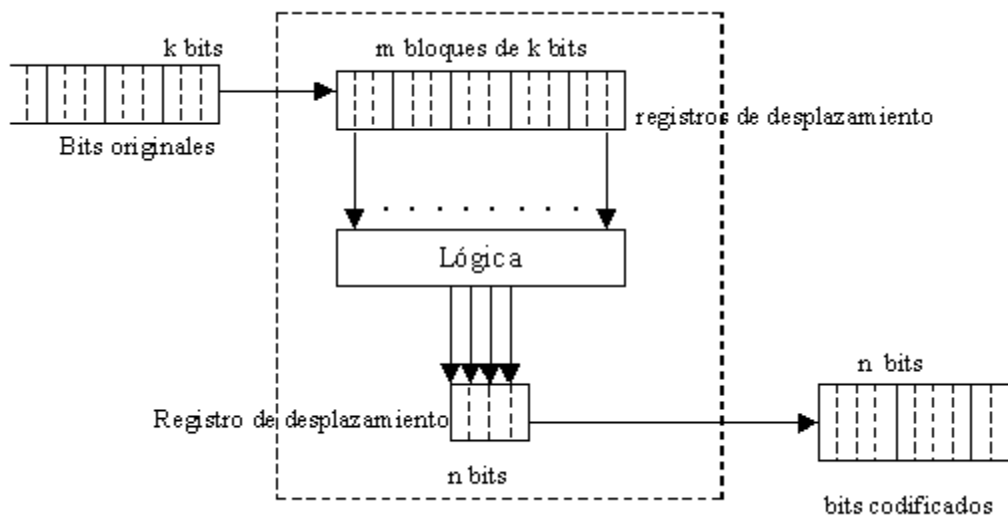


Figura Tomada de <http://web.usc.es/~elusive/simula3.html>

La palabra codificada se obtendría como el resultado de realizar una serie de operaciones lógicas entre determinados bits que están almacenados en los registros intermedios (ver figura 2.6).

Ejemplo: Codificador convolucional

Figura 2.7. Conmutador Código convolucional.

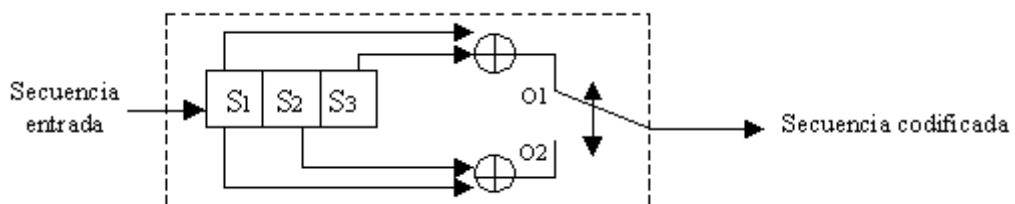


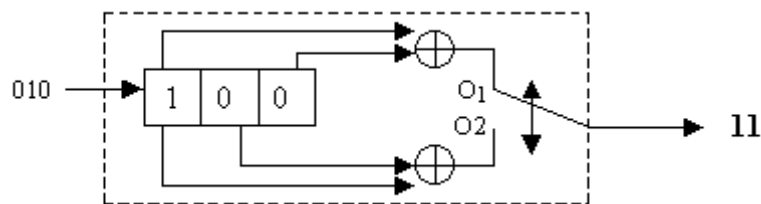
Figura Tomada de <http://web.usc.es/~elusive/simula3.html>

- El conmutador con las dos entradas hace el papel de un registro de desplazamiento de dos estados.
- El código convolucional es generado introduciendo un bit de datos y dando una revolución completa al conmutador.
- Inicialmente se supone que los registros intermedios contienen ceros.

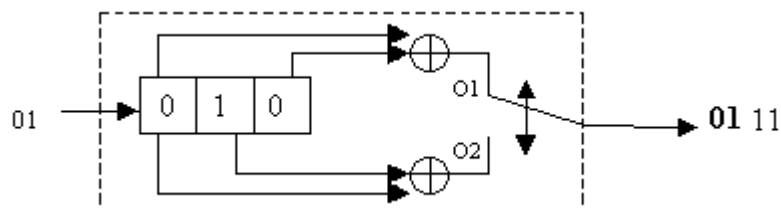
En este ejemplo la palabra codificada se obtiene como resultado de sumas módulo-2 entre los bits indicados que están almacenados en los registros intermedios (ver figura 2.8-2.11).

Supongamos que se quiere enviar la secuencia de bits 0101 (donde los bits más a la derecha son los más antiguos). El proceso de codificación es el siguiente:

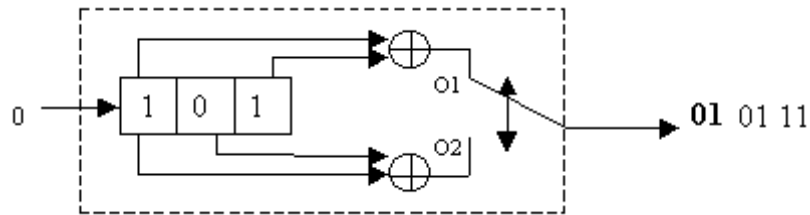
2.8 . Se introduce el primer bit de la secuencia en el codificador:



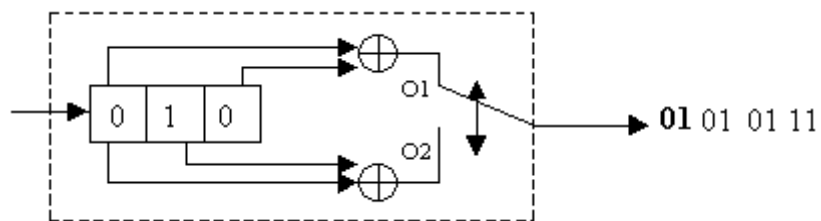
2.9 . Se introduce el segundo bit de la secuencia en el codificador:



2.10. Se introduce el tercer bit de la secuencia en el codificador:



2.11 Se introduce el cuarto bit de la secuencia en el codificador:



Figuras 2.8 – 2.11 Tomadas de <http://web.usc.es/~elusive/simula3.html>

Al final del proceso de codificación obtenemos que la secuencia codificada es 01 01 01 11.

Sigamos con la exposición del proceso de codificación.

Debido a la memoria del código es necesario de disponer de medios adecuados para determinar la salida asociada a una determinada entrada.

Hay tres métodos gráficos:

- **Diagrama árbol o árbol del código:** representación mediante un árbol binario de las distintas posibilidades.
- **Diagrama de estados:** es la forma menos utilizada.
- **Diagrama de Trellis o enrejado:** es la forma más utilizada porque es la que permite realizar la decodificación de la forma más sencilla.

DIAGRAMA DE TRELIS.

El diagrama de Trellis es un diagrama en forma de red. Cada línea horizontal se corresponde con uno de los estados del codificador. Cada línea vertical se correspondería con uno de los niveles del árbol del código.

Partimos del estado inicial del codificador en el primer nivel del árbol. A partir de aquí se trazan dos líneas desde este estado. Una para el caso de que la siguiente entrada fuera un 0 y otra para el caso de que fuera un 1. Estas líneas irán hasta el siguiente nivel del árbol al estado en el que queda el codificador después de haber codificado las correspondientes entradas. Encima de cada una de estas líneas escribiremos la salida del codificador para esa codificación.

2.4.4 CODIFICACION TRELIS

“La codificación Trellis difiere de otros métodos de codificación porque combina en una sola operación la codificación de datos y el mapeo de la constelación. Por esta razón, se usa a menudo el término Trellis Coded Modulation(TCM). La salida de un Encoder Trellis es una señal de banda base (par (I,Q) que es un punto en la constelación del modulador). A diferencia de otros códigos, la tasa de símbolos de canal transmitida es a menudo igual a la tasa de símbolos de la información”⁶. Esta propiedad es altamente deseable cuando el ancho de banda del canal disponible es limitado. El overhead no existe, ya que se tiene un aumento en el número de puntos de la constelación disponible en el transmisor. Por ejemplo, la codificación Trellis usada en el módem V.32 mapea una palabra de entrada de 16 estados (4 bits) a una constelación de 32 puntos (5 bits). En esta norma, se logra un 25% de incremento en la tasa de bits de codificación sin un aumento en el ancho de banda del canal. Como puede mostrarse fácilmente, el ancho de banda se relaciona con la tasa de símbolo transmitida, la cual es proporcional a la tasa de bits del canal.

⁶ Camilo A. Londoño. 31 de agosto de 2004. Tutorial: Introducción a la Modulación Codificada de Trellis. Universidad de los Andes Bogotá, Colombia.

3. DECODIFICACIÓN

Este segmento de codificación de canal se hace necesario debido, a que todo canal de comunicación introduce pérdidas o basura a nuestra información ya sea en menor o mayor grado, pero siempre se presenta este inconveniente. Es por esto que nuestra información debe protegerse de alguna forma.

Si en el transmisor fue usada algún tipo de codificación de señal, se requiere necesariamente en el receptor realizar el proceso de decodificación después de demodular. Tal proceso requiere de un conocimiento de la codificación utilizada en el transmisor. La decodificación de una señal es normalmente más complicado que el proceso de codificación y puede ser computacionalmente muy intensa. En una experiencia anterior, se estudiaron los codificadores del tipo convolucional y los del tipo Trellis, así como los interleaving.

Además de los diferentes esquemas de decodificación, debe hacerse una distinción adicional en lo que se conoce como “hard decisión” y “soft decisión”. En una hard, el codificador proporciona bits de decisión del tipo “hard” al demodulador. Esto significa que el decodificador consigue un “0” a la entrada o un “1” por cada bit recibido desde el canal. Por otro lado, en el caso soft, el decodificador pasa un valor multi-escalado del bit recibido desde el canal (e.g. un nivel de 0-7). Esta información adicional provee al decodificador de un nivel de confianza con una métrica asociada por cada bit recibido. Una buena regla es que una implementación con soft decisión tiene una ventaja de aproximadamente 2 dB sobre la aproximación hard.

Se describen a continuación un tipo de decodificador.

3.1 Decodificación de Viterbi

La decodificación de un dato que fue codificada convolucionalmente es comúnmente realizada por medio del algoritmo de Viterbi. “Este algoritmo construye una Trellis finita de todas las posibles secuencias de bits recibidas y

mantiene una métrica acumulativa para cada camino en la Trellis⁷. Cuando cada nuevo bits del canal es detectado, la métrica acumulada se actualiza en cada camino por medio de una tabla determinada por los coeficientes del generador de código. Después de un “largo de truncamiento” el decodificador selecciona el camino con la mejor métrica como la ganadora y entrega una decisión de bits. Las longitudes de truncamiento típicas son aprox 5 veces la longitud de restricción del código Para una longitud de restricción de 7 para un código, una longitud del truncamiento aceptable es por consiguiente de 35 bits. Esto significa que el decodificador opera con un retraso de 35 períodos de bits.

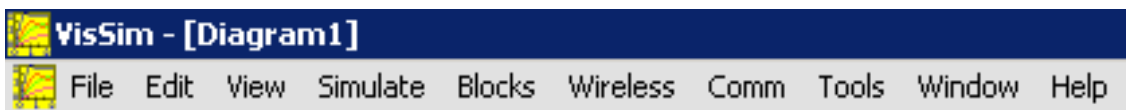
Existen los decodificadores de Viterbi con hard y soft decisión. La diferencia entre los dos, es que el bloque de hard decisión espera un flujo de bits digital como entrada de la etapa anterior, mientras que el soft espera una señal de entrada analógica. El paso de cuantización de la señal se lleva a cabo dentro del bloque de soft decisión que incluye el número de bits de cuantización del conversor A/D como un parámetro.

⁷ Leslei Murray. 2000; 20 de mayo de 2000. Introducción a los Códigos Convolucionales. Escuela de Ingeniería Electrónica, Facultad de Ciencias Exactas, Ingeniería y Agrimensura Universidad Nacional de Rosario. [libro en línea].

4.1.1 Barra principal

La denominada barra principal es la que contiene todas las opciones con las cuales se puede trabajar en VisSim, la cual se puede observar en la figura 4.1.2.

Figura 4.1.2. Barra Principal



Se puede observar que dicha barra posee pestañas que empiezan desde "File" y termina en "help, cada una será explorada a su debido tiempo mostrando las principales características.

Para que el usuario se familiarice con el programa se recomienda hacer una previa exploración de estas pestañas haciendo énfasis en las denominadas: "Blocks" y "Comm" que contienen la mayoría de bloques útiles en el ámbito de las telecomunicaciones.

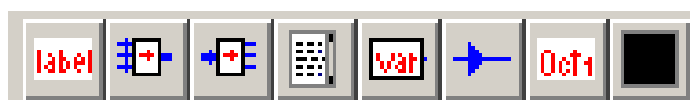
4.1.2 Barras de trabajo

Se realiza un paso por las barras que se encuentran en el VisSim, donde se puede observar las características más importantes de estas.

4.1.2 Annotation Blocks

En esta barra (ver figura 4.1.3) se encuentran ítems de gran ayuda para futuros trabajos, a continuación se nombra alguna de ellas:

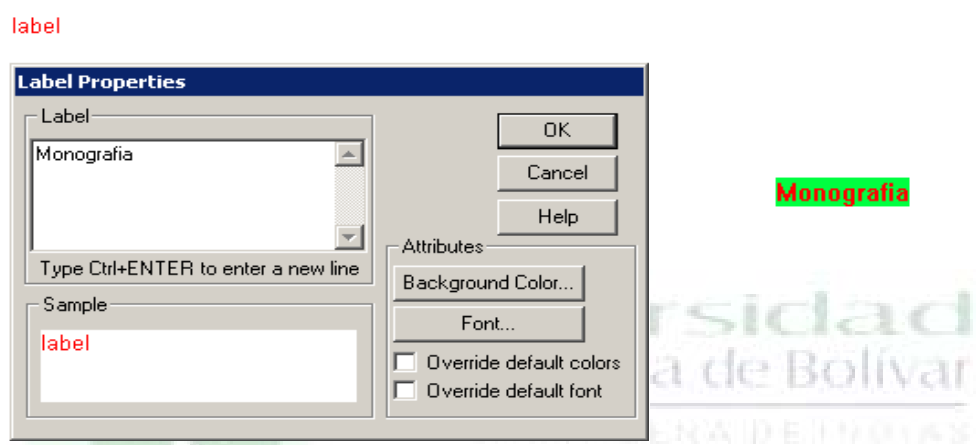
Figura 4.1.3. Barra "Annotation Block"



4.1.2.1 “Label”

Permite hacer anotaciones de carácter importante, las cuales pueden ser ubicadas en cualquier sector de la pantalla, pudiendo seleccionar un color de fondo para dicha anotación, así como el tipo de letra para la misma, tal cual se muestra en la siguiente imagen. Son usualmente utilizadas para nombrar señales e identificar cualquier bloque en especial.

Figura 4.1.4. Ventana de propiedades de “Labels”

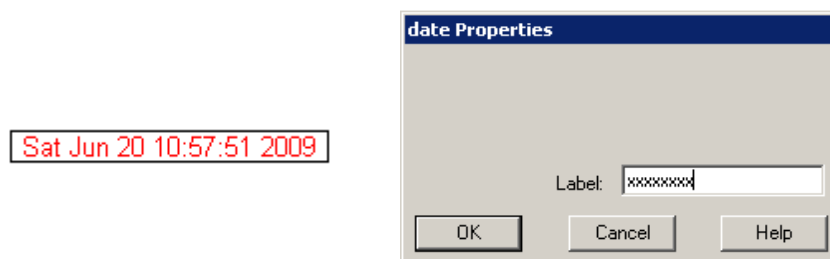


Se observa a la izquierda de la figura 4.1.4 el cuadro de configuración de dicho bloque y a la derecha el resultado obtenido de la configuración asignada.

4.1.2.2 “Date”

Como su nombre lo indica este bloque mostrara la fecha y la hora actual. La fecha y la hora se actualizaran cuando se presione sobre el bloque o cuando se mueva el mismo. Se puede observar un ejemplo en la figura 4.1.5.

Figura 4.1.5. Ventana de propiedades de “Date”



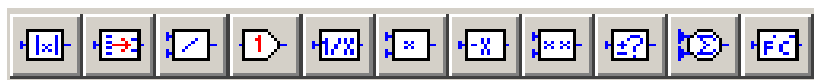
Se observa a la izquierda como queda digitalizada la fecha y hora sobre la pantalla de VisSim y a la izquierda se puede observar el cuadro de propiedades.

Annotation blocks posee ocho diferentes bloques los cuales pueden ser explorados por el lector para familiarizarse con el contenido completo de la barra, y para mayor información respecto a cada uno, se puede ir a las propiedades del mismo, dando doble clic sobre el bloque y presionar la pestaña denominada "Help", la cual realizará una reseña respecto a dicho bloque.

4.1.3 Arithmetics blocks

Esta barra (Ver figura 4.1.6) presenta una serie de bloques enfocados a operaciones matemáticas sencillas como la multiplicación, división entre otros.

Figura 4.1.6. Barra "Arithmetics Block"

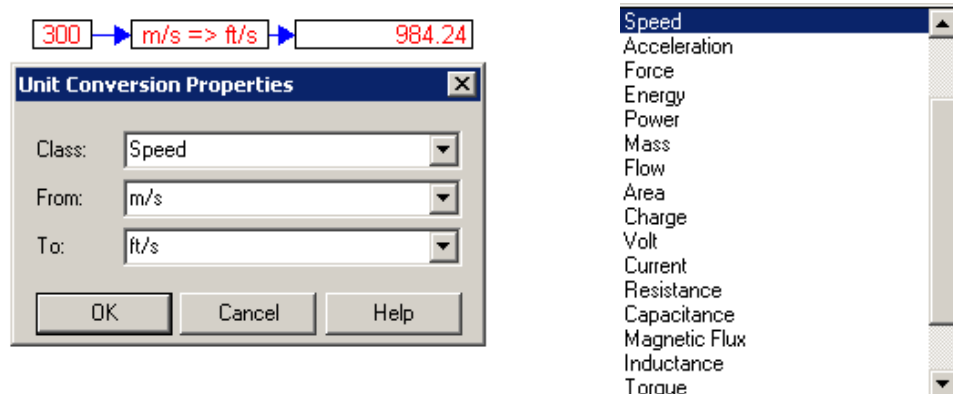


A continuación se presentaran los más llamativos

4.1.3.1 "Unit Conversion"

Este bloque trae consigo una amplia gama de conversión de unidades de medida, como se logra observar en la figura 4.1.7.

Figura 4.1.7. Ventana de bloque conversión de unidades

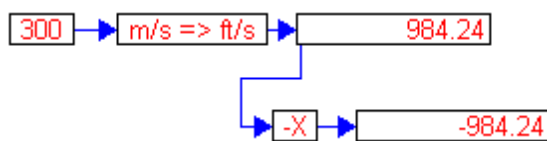


Se observa en la imagen de la izquierda un ejemplo de conversión y el recuadro de propiedades y a la derecha la lista de clases de unidades que nos presenta VisSim/Comm.

4.1.3.2 “Negate”

Este bloque permite hacer la negación de alguna variable, tal y como se muestra en el siguiente ejemplo (ver figura 4.1.8).

Figura 4.1.8. Ejemplo de diagrama de bloque donde se realiza conversión de unidades y se aplica el bloque “Negate”.



Arithmetics blocks posee once diferentes bloques los cuales pueden ser explorados por el lector para familiarizarse con el contenido completo de la barra, y para mayor información respecto a cada uno, se puede dirigir a las propiedades del mismo, dando doble clic sobre el bloque y presionando la pestaña denominada “Help”, la cual realizará una reseña respecto a dicho bloque.

4.1.3.3 “Boolean Blocks”

Esta barra (ver figura 4.1.9) en particular posee 9 bloques de operaciones booleanas entre las cuales se tiene: mayor que, menor que, mayor o igual que, entre otros.

Figura 4.1.9. Barra de “Boolean Blocks”

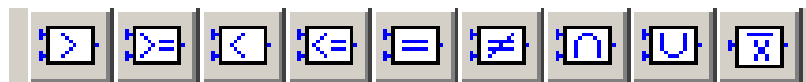
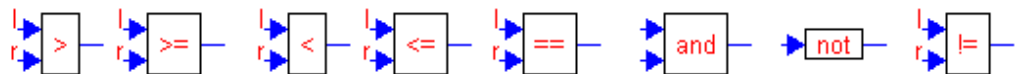


Figura 4.1.10. Algunos bloques de operaciones Booleanas.

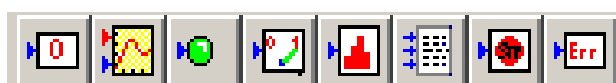


Boolean blocks posee nueve diferentes bloques los cuales pueden ser explorados por el lector para familiarizarse con el contenido completo de la barra.

4.1. 4 “Consumer Blocks”

Es una de las barras (ver figura 4.1.11) más importantes de VisSim/Comm, ya que con esta se logra observar las salidas o resultados las simulaciones, de cualquier tipo ya sean señales análogas, bits, constantes, entre otros.

Figura 4.1.11. Barra de “Consumer Blocks”

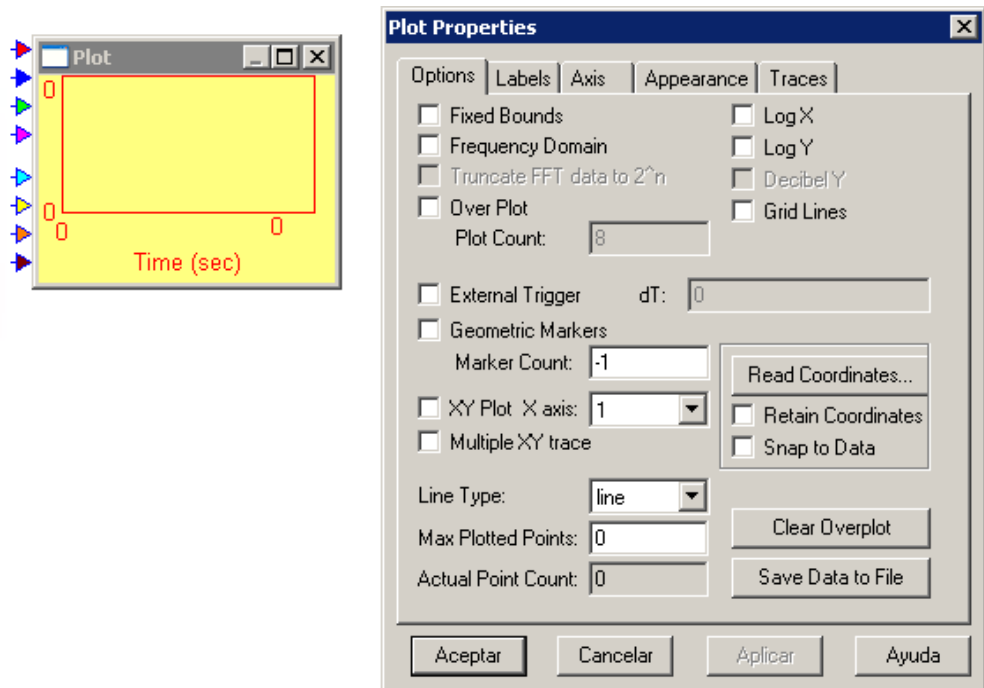


A continuación se presentaran los más llamativos:

4.1.4.1 “Plot”

El bloque “Plot” es un de los más utilizados para observar la salida de señales análogas y digitales, tiene la cualidad de poder recibir un máximo de ocho señales en una misma venta, la cual puede ser configurada de la forma deseada, cambiando la escala de sus ejes, colocando etiquetas a cada posible señal de acuerdo con el color otorgado y hasta cambiando su apariencia modificando su color de fondo entre otros.

Figura 4.1.12. Bloque de Plot con su respectiva ventana de configuración.

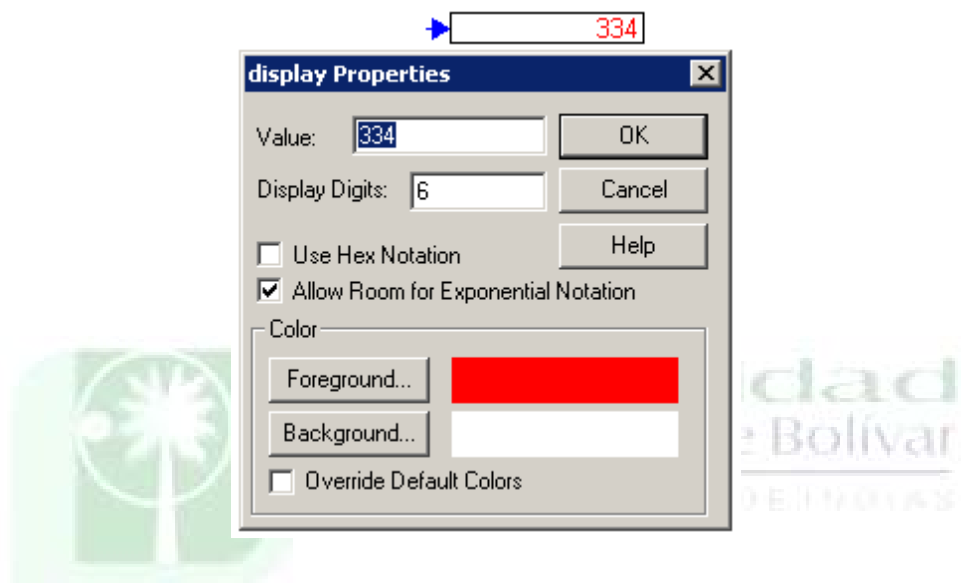


Se puede observar en la figura 4.1.12 a la izquierda de la imagen el recuadro de Plot en el cual se muestra la imagen/señal que se obtiene de la simulación y a la derecha el recuadro de propiedades que se puede modificar en a medida que sea necesario para apreciar mejor la simulación.

4.1.4.2 “Display”

Como su nombre lo indica este bloque brinda la opción de tener una pantalla en la cual visualizar la salida de cualquier variable, también se le puede dar un valor previo, cambiar la estructura física como el color y usar notación hexadecimal, entre otros.

Figura 4.1.13. Bloque display y ventana de configuración.



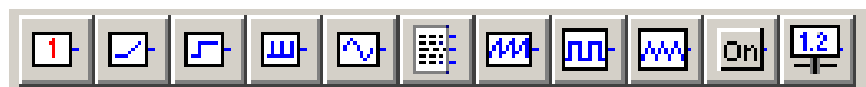
Se puede observar en la figura 4.1.13 parte superior de la imagen el Display y en la parte inferior las propiedades del mismo, las cuales pueden ser editadas en cualquier momento.

Arithmetics blocks posee ocho diferentes bloques los cuales pueden ser explorados por el lector para familiarizarse con el contenido completo de la barra, y para mayor información respecto a cada uno, se puede ir a las propiedades del mismo, dando doble clic sobre el bloque y presionar la pestaña denominada “Help”, la cual realizará una reseña respecto a dicho bloque.

4.1.5 “Producer blocks”

Estos bloques dan la opción de crear diferentes señales, permite visualizar desde una constante, pasando por un escalón unitario, hasta llegar a parámetros deslizantes, es decir generar señales que varían fácilmente mediante un botón deslizante.

Figura 4.1.14. Barra de “Producer Blocks”.



Puede verse cómo crear una señal escalón en la figura 4.1.15, en este caso unitario porque tiene de amplitud la unidad y con un tiempo de retardo de 2 segundos.

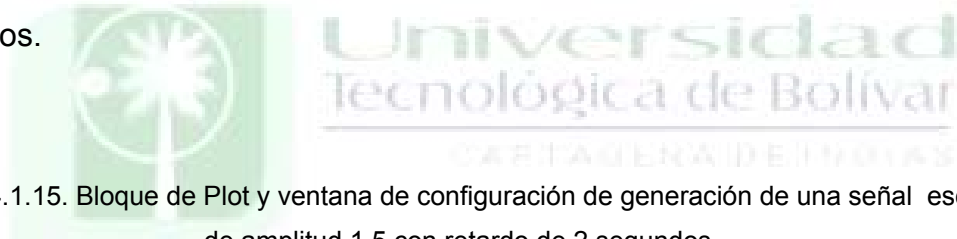
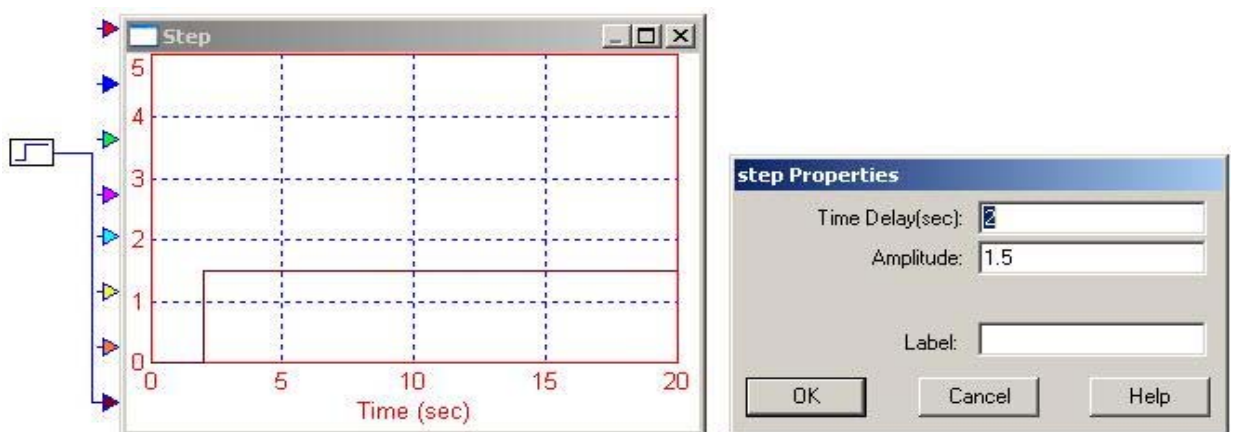
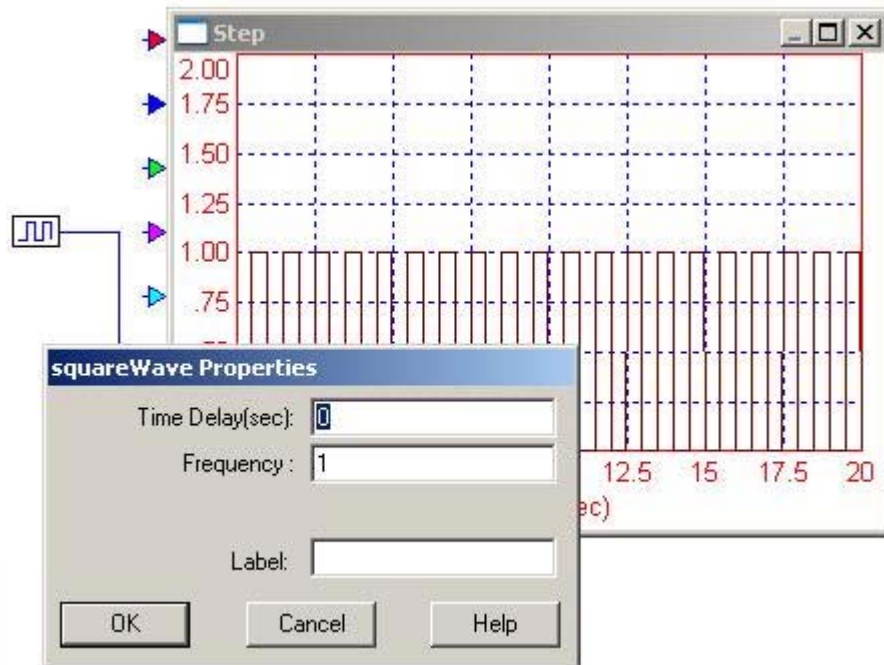


Figura 4.1.15. Bloque de Plot y ventana de configuración de generación de una señal escalón de amplitud 1.5 con retardo de 2 segundos.



Se pueden crear señales “tren de pulsos”:

Figura 4.1.16. Bloque de Plot y ventana de configuración de generación de una señal tren de pulsos con amplitud 1 con retardo de 0 segundos.

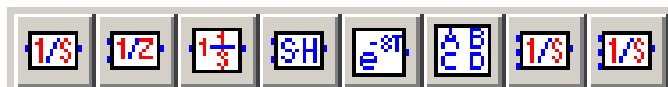


El bloque “onda cuadrada” (SquareWave) permite crear un tren de pulsos por defecto de amplitud 1, frecuencia de 1 en Hz, y tiempo de retardo de 0 segundos. Configurables de acuerdo a las necesidades de operación, ver figura 4.1.16.

4.1. 6 “Dynamic Blocks”

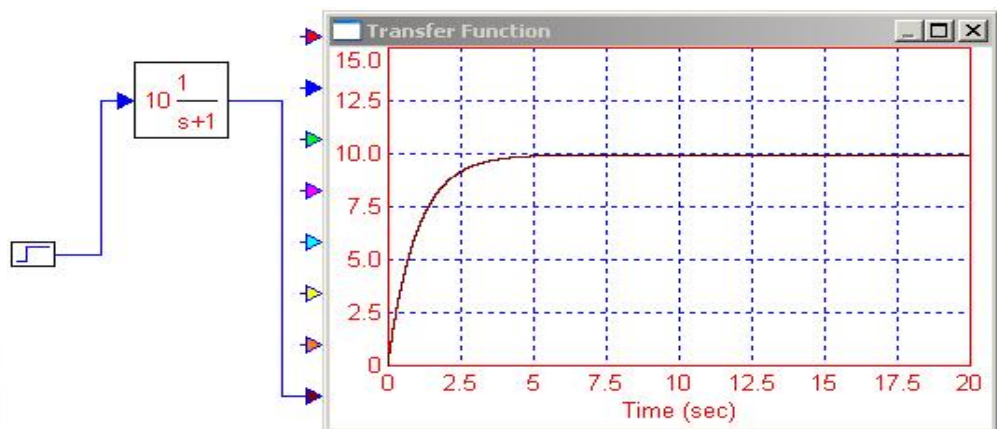
Posee ocho bloques diferentes que permiten trabajar en frecuencia a través de funciones de transferencia, integradores a través de transformadas de Laplace, Transformadas discretas Z, ecuaciones de Espacio Estado.

Figura 4.1.17. Barra de “Dynamics Blocks”



A manera de ejemplo se realiza una muestra del bloque “función de transferencia”, la cual es configurada con ganancia 10, un polo en -1 y de primer orden en presencia de una señal escalón unitario.

Figura 4.1.18. Bloque de Plot que representa la respuesta de un bloque de Función de transferencia.

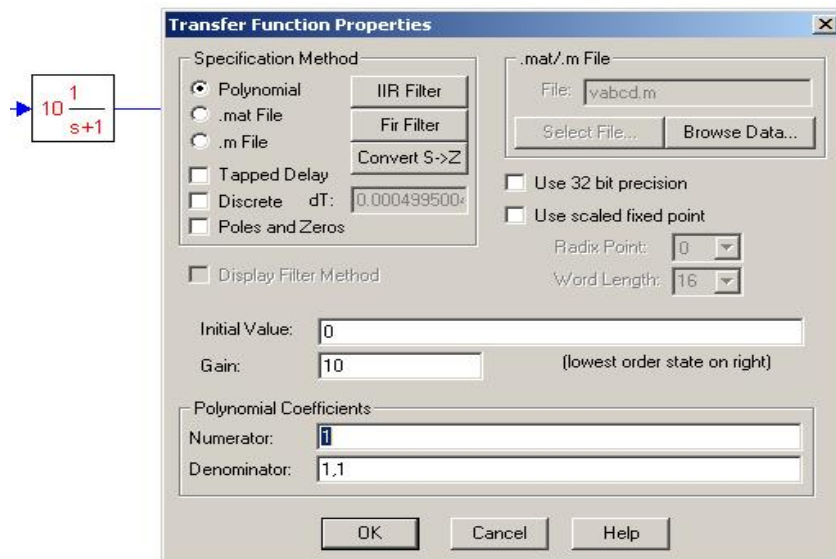


En la figura 4.1.18 se puede observar que es usado un bloque de “Producer Block”, uno de “Consumer Blocks” y el bloque función de transferencia de “Dynamics Blocks”.

En la parte izquierda se encuentra una señal escalón unitario la cual es a una función de transferencia que da una respuesta de acuerdo a la grafica que se muestra en Plot.

El bloque función de transferencia permite configurar sus propiedades para realizar la función de transferencia adecuada, se puede observar una ventana de dicho bloque en la figura 4.1.19.

Figura 4.1.19. Ventana de configuración de función de transferencia.



Se especifica el método a utilizar para expresar la función de transferencia. En este caso Polinómica. Las funciones pueden ser traídas de un archivo de Matlab a través de un archivo con formato .mat File o a través de el formato .m File. En la parte de Gain: Se configura la ganancia de la función de transferencia. En la parte de numerador y denominador que se encuentra activos se colocan los coeficientes de los polinomios que conforman el numerador y el denominador.

4.1.7 Sim Control

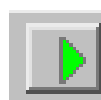
Los bloques “Sim Control” (ver figura 4.1.20) permiten manejar las simulaciones realizadas mediante cuatro bloques que admiten realizar la experiencia paso a paso, simulaciones en un solo paso o simulaciones continuas.

Figura 4.1.20. Barra de bloques de control de simulación.



El bloque más utilizado es el bloque de simulación en un solo paso:

Figura 4.1.21. Bloque de simulación en un solo paso.



Este bloque es utilizado para realizar simulaciones en un solo paso, permite obtener una respuesta de manera inmediata pero omite algunos pasos que sería bueno conocer para un mejor aprendizaje. Cada vez que se realice cambios al diseño es preciso utilizar este bloque para que se puedan visualizar los en las señales de salida.

4.1.8 Main

En estos bloques de uso común y ampliamente difundidos en la mayoría de software actuales, se encuentran funciones básicas como Abrir, copiar, pegar, cortar y algunas de aplicación especial para este software como lo son los bloques de añadir conector o eliminar conector (ver imagen 4.1.22).

Figura 4.1.22. Barra de bloques principales



Figura 4.1.23. Bloque de añadir que permite como ampliar la cantidad de conectores por ejemplo de una sumatoria.



El anterior bloque permite añadir conectores a operaciones básicas de otros bloques. Para este caso a una sumatoria de dos términos fue posible añadir un término más por medio de este bloque.

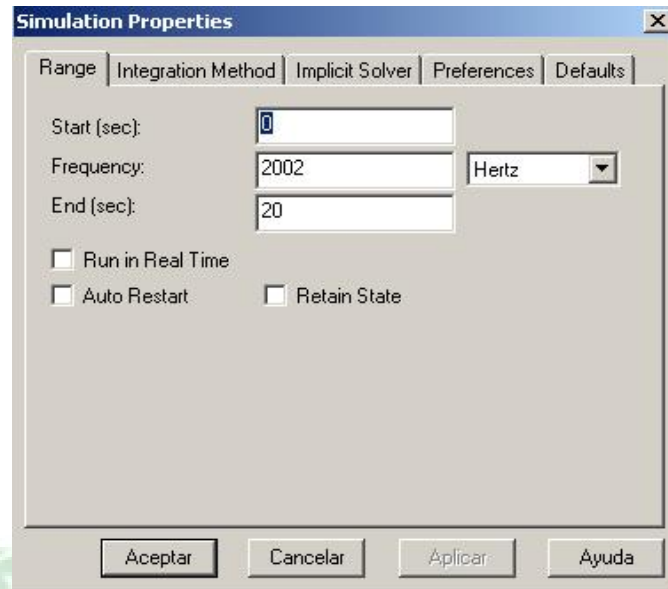
4.1.9 Simulation Properties

Figura 1.4.24. Bloque de propiedades de simulación.



Este bloque permite realizar la configuración de los parámetros iniciales del motor de simulación del software.

Figura 1.4.25. Ventana de propiedades de simulación.

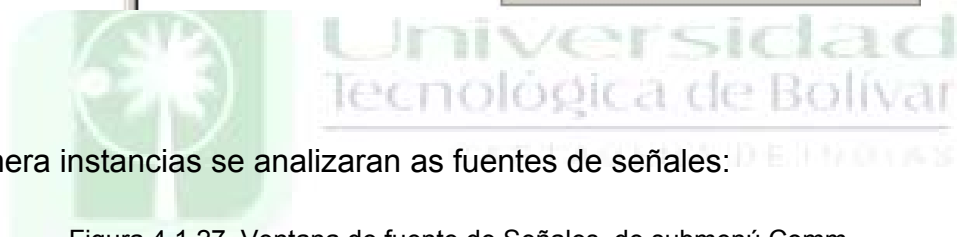
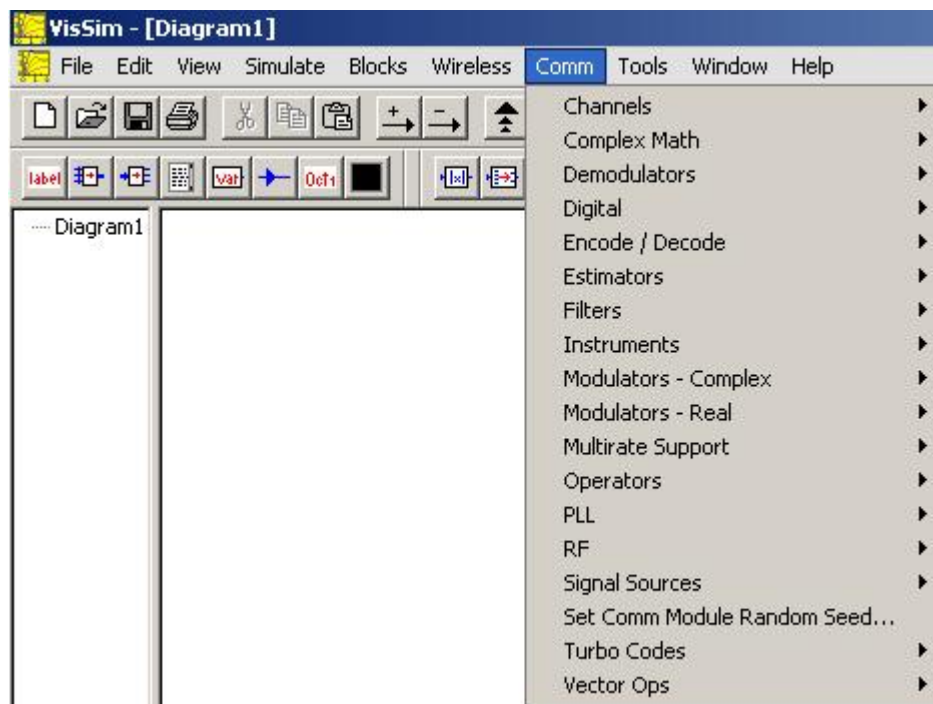


Se puede configurar los rangos de frecuencia, las unidades, los métodos de integración utilizados para mejorar el programa, los métodos que utiliza para resolver operaciones internas y algunas preferencias más (ver figura 1.4.25).

De la barra principal se analizara detalladamente la lista desplegable de "**Comm**". Este submenú se caracteriza por poseer una cantidad considerable de aplicaciones a Telecomunicaciones (ver figura 4.1.26).

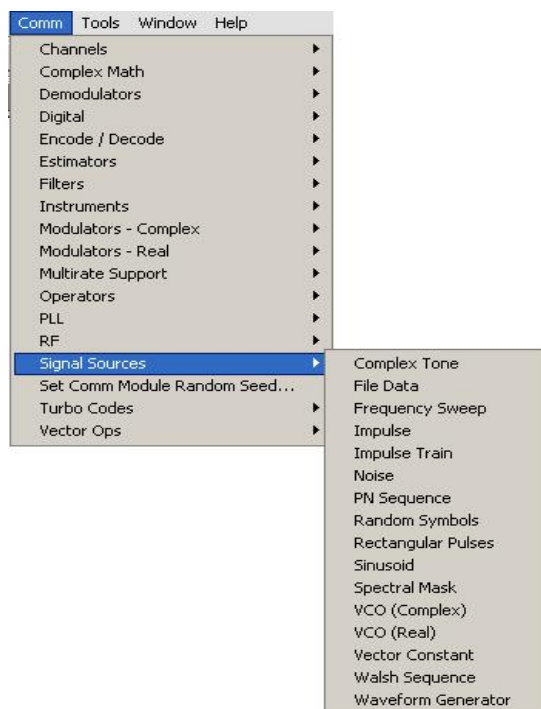
De este submenú Comm, se detallaran algunos bloques importantes para realizar una buena simulación de sistemas de telecomunicación, se encuentran fuentes de señales, codificación, decodificación, filtros demoduladores, moduladores etc.

Figura 4.1.26. Submenú Comm de barra de bloques principal.



En primera instancia se analizarán las fuentes de señales:

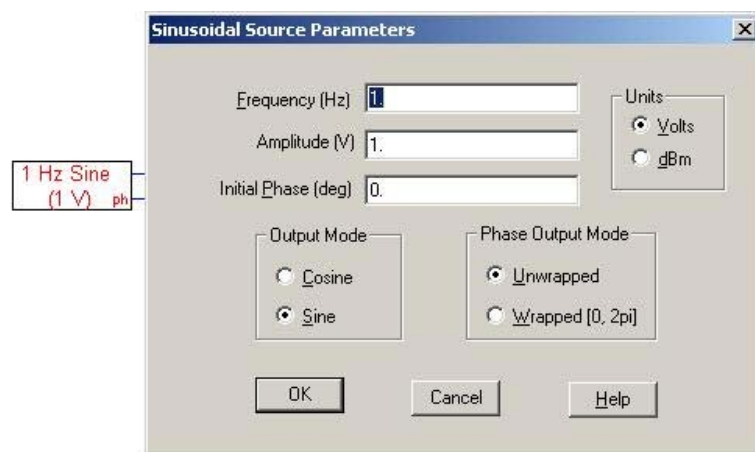
Figura 4.1.27. Ventana de fuente de Señales de submenú Comm.



Las fuentes de señales pueden ser senoidales, señales de ruido, pulsos rectangulares, tren de impulsos o simplemente un archivo de datos que se encuentra compuesto por una señal.

Para efectos de visualizar un ejemplo (ver figuras 4.1.28 y 4.1.29) se utilizara una señal senoidal representada en el tiempo mediante el bloque Plot.

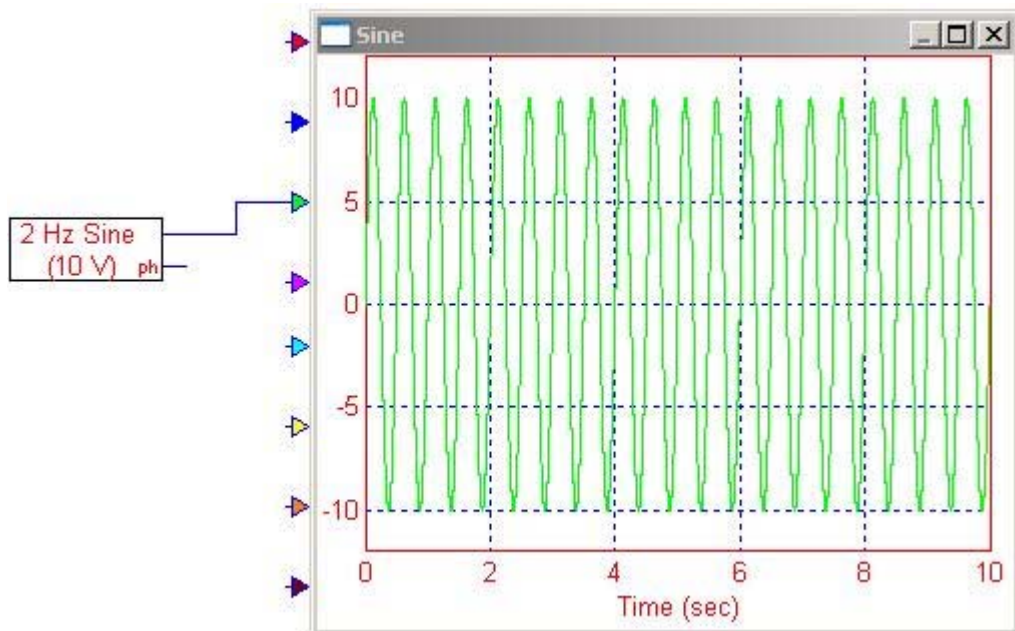
Figura 4.1.28. Ventana de configuración de fuente de señal senoidal.



Del bloque generador de señal senoidal se pueden cambiar los parámetros de frecuencia, amplitud, y fase, además de determinar si es una señal seno o una señal coseno, entre otros parámetros. Como en todos los bloques se encuentra la opción “help” que nos explica brevemente las utilidades del bloque.

Se configura la frecuencia a 2 Hz (2 ciclos por segundo), la amplitud a 10 V de una señal seno y se puede visualizar en la figura 4.1.29.

Figura 4.1.29. Grafica de onda senoidal configurada a 2 Hz y 10 V de amplitud.

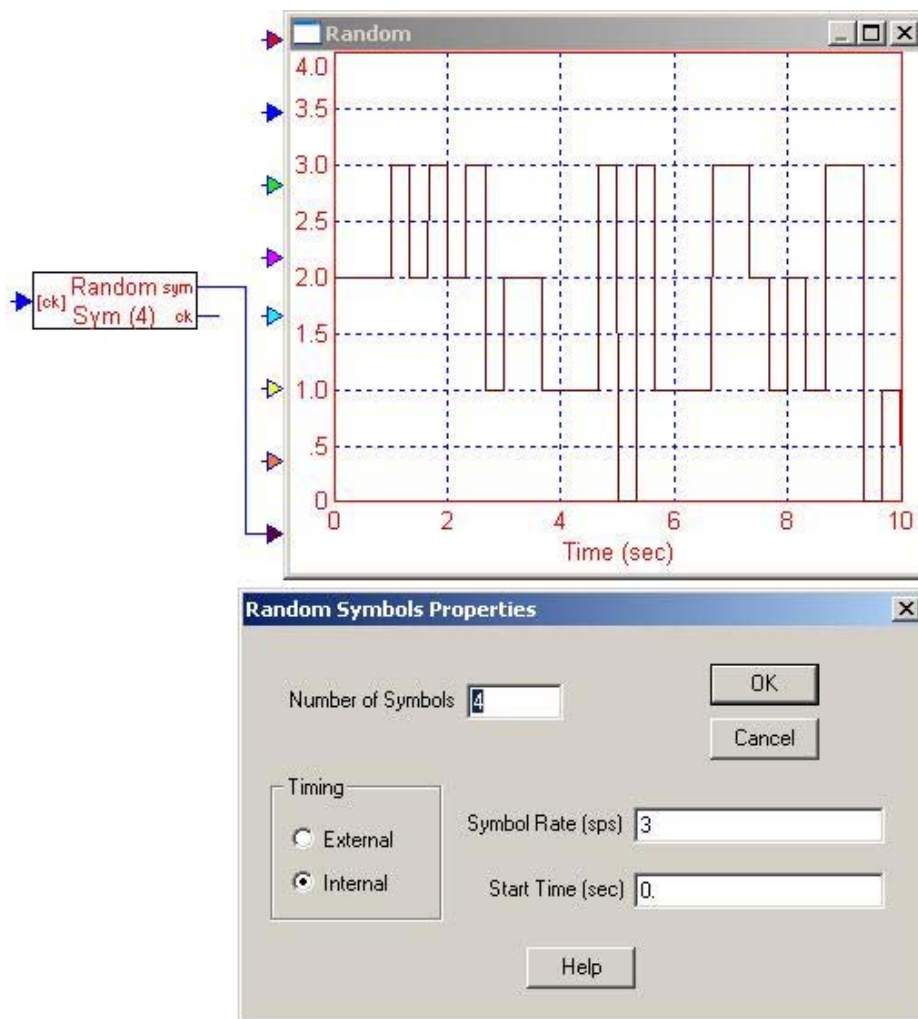


Otro tipo de señal que se puede manejar son las señales “Random symbols” o de símbolos aleatorios.

Este bloque genera de manera uniforme símbolos aleatorios entre 0 y N-1, donde N es el número total de símbolos. El valor de N, la rata de símbolos y el retardo inicial pueden ser especificados.

Se configura para un número de símbolos de 4 y una rata de símbolos de 3 (símbolos por segundo), se configura para señal de reloj interna, y el resultado es mostrado en la figura 4.1.30.

Figura 4.1.30. Grafica de señal Random, configurada a 4 símbolos y 3 símbolos por segundo.



4.1.10 ACTIVIDADES NIVEL I

El Programa VisSim/Comm proporciona bloques para fuentes analógicas y digitales. Combinando bloques básicos, el usuario puede generar cualquier forma de onda en el dominio del tiempo del tipo analógica. Los bloques básicos incluyen sinusoides, constante, rampa, función escalón y de impulso por nombrar algunos. Tales bloques se pueden obtener directamente desde su icono o desde los conjuntos BLOCK o COMM.

- De manera sencilla y con ayuda de los bloques descritos anteriormente generar cada señal individualmente acomodar las propiedades e identificar parámetros como frecuencia y tiempo de retardo y una señal que finalmente sea el resultado de la suma de 4 señales diferentes a saber:

➤ **1 señal cuadrada**

Delay 3 seg

Frecuencia : 15 Hz

➤ **2 señales senoidales.**

Delay 0 seg

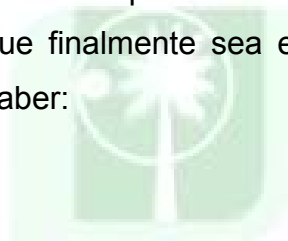
Frecuencia 1: 1Hz

Frecuencia 2: 3 Hz,

Amplitud 1: 5V

Amplitud 2: 3V

➤ **1 señal constante de 5.**



4.2 SIMULACIÓN NIVEL II

- **Entrenamiento:** duración estimada 200 min.

- **Actividad:** duración estimada 80 min.

En este nivel se aplicarán todas las funciones básicas de creación de señales y de configuración para realizar simulaciones antes aprendidas, dirigidas a los diferentes tipos de modulación analógica y digital.

4.2.1 Modulación Analógica

En este tipo de modulación, la señal transmitida puede variarse continuamente sobre un rango especificado. Ejemplos de modulación analógica incluyen a la Modulación de Amplitud (AM), Modulación de Fase (PM), y Modulación de Frecuencia (FM). Como su nombre implica, un transmisor de AM opera variando la amplitud de la portadora según el voltaje de la señal en banda base de entrada. En un transmisor de PM, la señal de entrada se usa para controlar la fase instantánea de la portadora. En la FM, por otro lado, la señal de entrada se usa para variar la frecuencia instantánea de la portadora. En los tres casos, la señal de entrada es una señal banda base analógica, como un señal de voz.

A continuación se realizará un ejemplo sencillo en el cual aplicaremos una modulación AM con los bloques principales de *VisSIMCOM*.

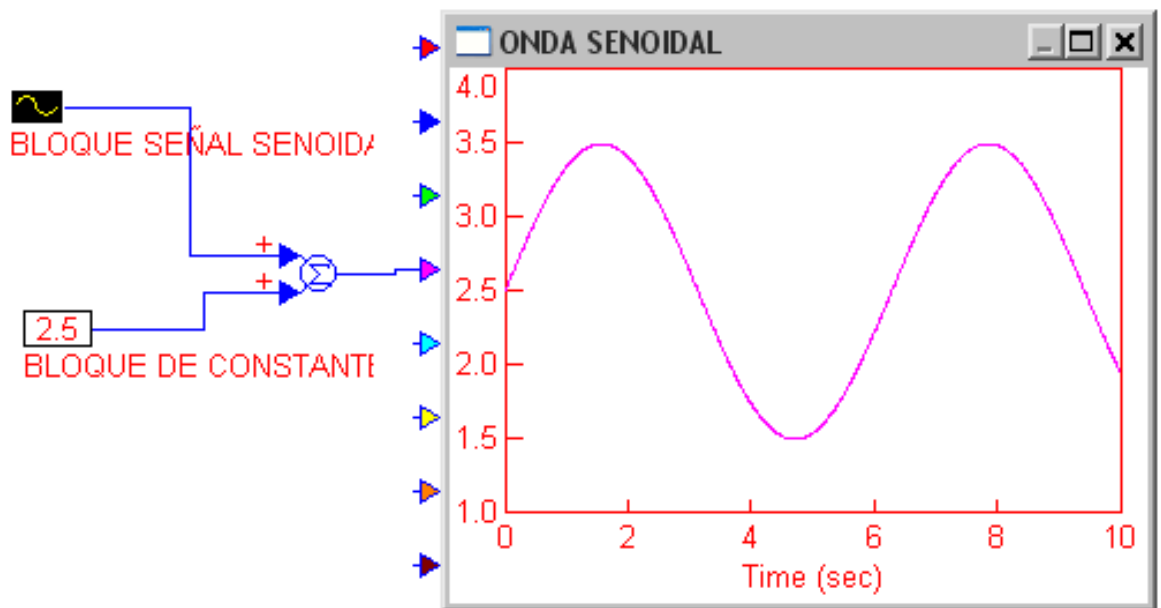
Luego de haber iniciado nuestro programa y tenerlo en la pantalla principal procedemos a crear una señal analógica, que tendrá la función de representar una señal en banda base que será transmitida mediante el método de modulación en amplitud. Se crea una señal de baja frecuencia tal como se desarrollo en el nivel numero 1.

Para crear la señal en banda base:

- Creamos la señal por medio de tres bloques principales:
 - *Summing Junction*
 - *Const*: se configura un valor constante
 - *Sinusoid*: se configura frecuencia y amplitud.

Al final se obtendrá una grafica en Plot, la cual se puede observar en al figura 4.2.1.

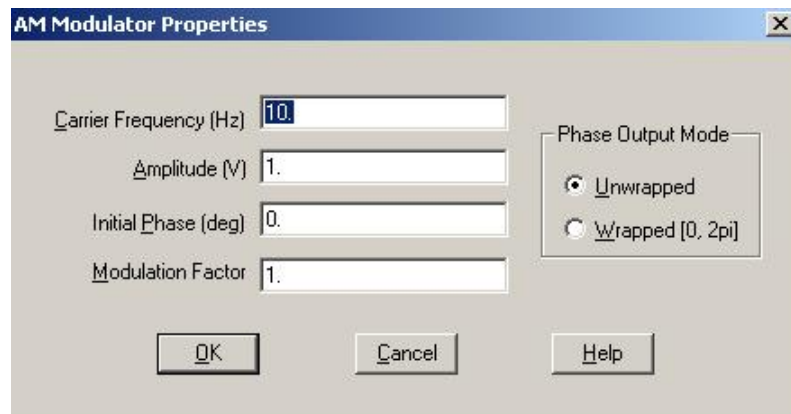
Figura 4.2.1 .Grafica de señal senoidal modificada, por medio de la unión de un bloque de senoidal, un bloque constante y el bloque suma de funciones.



4.2.1.1 Modulación en Amplitud

Luego de obtener la señal moduladora buscamos en VisSIMCOM en el submenú “Comm” el bloque de modulación AM, el cual se encuentra en la siguiente ruta: *Blocks/Modulators-Real/AM(Re)*

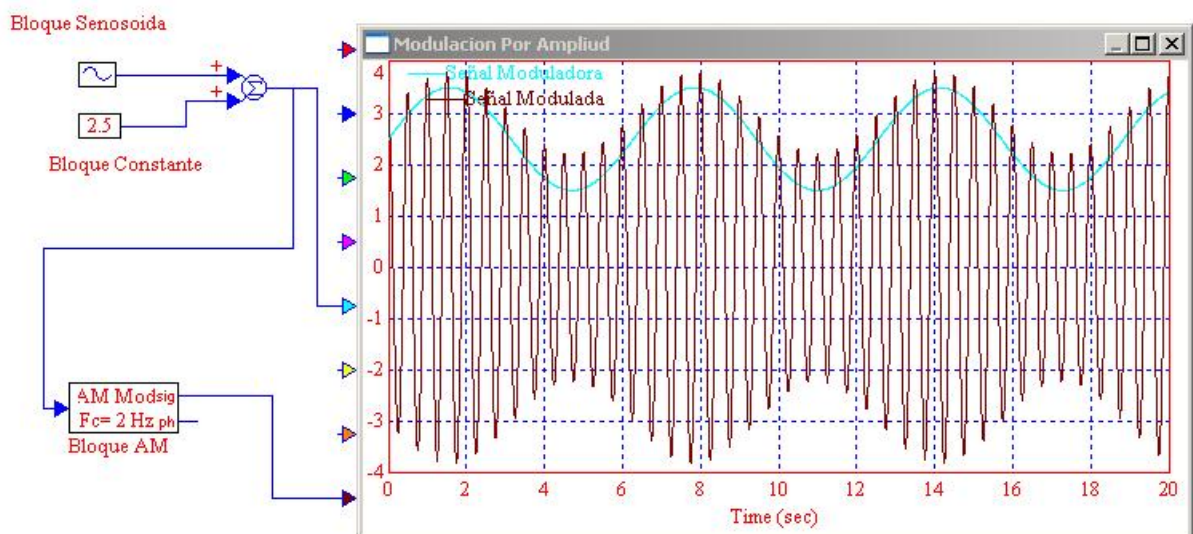
Figura 4.2.2. Ventana de configuración de modulación en amplitud.



Se configura una señal portadora o Carrier de frecuencia mucho mayor que la señal de información, se configura la amplitud de la portadora y el factor de modulación del sistema como se observa en la figura 4.2.2.

Para finalizar se realizan las conexiones correspondientes para obtener algo similar a lo que se muestra en la siguiente imagen.

Figura 4.2.3. Grafica de señal de banda base salida del bloque *junction* (color azul) y la señal modulada salida del bloque AM MOD (color café).



A manera de ejemplo se configura una señal en banda base con los siguientes parámetros:

Frecuencia: 1 Hz

Amplitud: 1 V

La configuración del bloque AM es la siguiente:

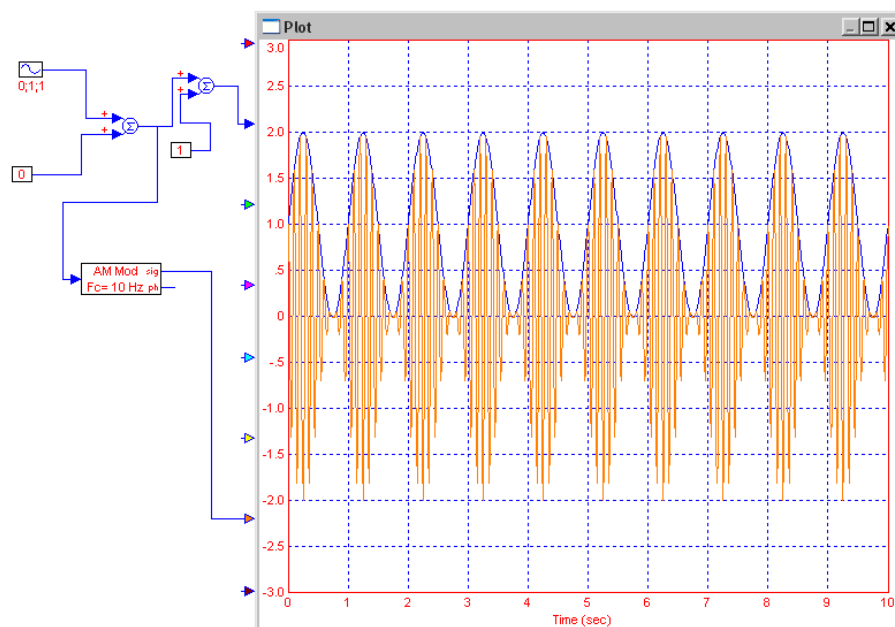
Frecuencia portadora (Carrier): 10 Hz

Amplitud: 1V.

Factor de modulación: 1

Las dos señales se conectan directamente a un *Plot*. Debido a que estamos realizando la grafica en el mismo *Plot*, la señal de de banda base estará por debajo de la señal modulada. El usuario deberá sumar una constante a la señal de banda base para que esta suba sobre el eje (ésta señal debe sumarse después de la entrada al modulador AM para que no lo afecte) y se pueda comprobar que la modulación va de acuerdo a la señal de información. La señal de salida deberá coincidir entre la señal modulada en amplitud y la señal de información (ver figura 4.2.4).

Figura 4.2.4. Grafica de señal de banda base (color azul) y la señal modulada (color café),



Como se puede observar en el *PLOT* tenemos dos señales superpuestas, la primera, que es la señal de “información” y la segunda es la señal modulada (señal café).

Se denomina índice de modulación (m), donde V_m corresponde a la amplitud de señal moduladora y V_p corresponde a la amplitud de la señal portadora.

$$m = \frac{V_m}{V_p}$$

Este valor puede ser cambiado en el bloque de modulación AM, pero se deja al usuario para que lo manipule y observe los resultados cuando las condiciones ocasionan que dicho índice sea mayor que 1.

4.2.1.2 Modulación en Frecuencia

La modulación de frecuencia consiste en variar la frecuencia de la onda portadora de acuerdo con la intensidad de la onda de información. La amplitud de la onda modulada es constante e igual que la de la onda portadora. La frecuencia de la portadora oscila más o menos rápidamente, según la onda moduladora, esto es, si aplicamos una moduladora de 100 Hz, la onda modulada se desplaza arriba y abajo cien veces en un segundo respecto de su frecuencia central, que es la portadora; además el grado de esta variación dependerá del volumen con que modulemos la portadora, a lo que denominamos “índice de modulación”.

A continuación se representa un ejemplo de la misma:

Para que la simulación fuera más ilustrativa se realizó una variación en la señal en banda base que funcionaría como señal moduladora, el usuario puede realizar cualquier cambio ya sea en la frecuencia del modulador, como en la señal a modular y observar las variaciones obtenidas en el recuadro de salida “Plot”.

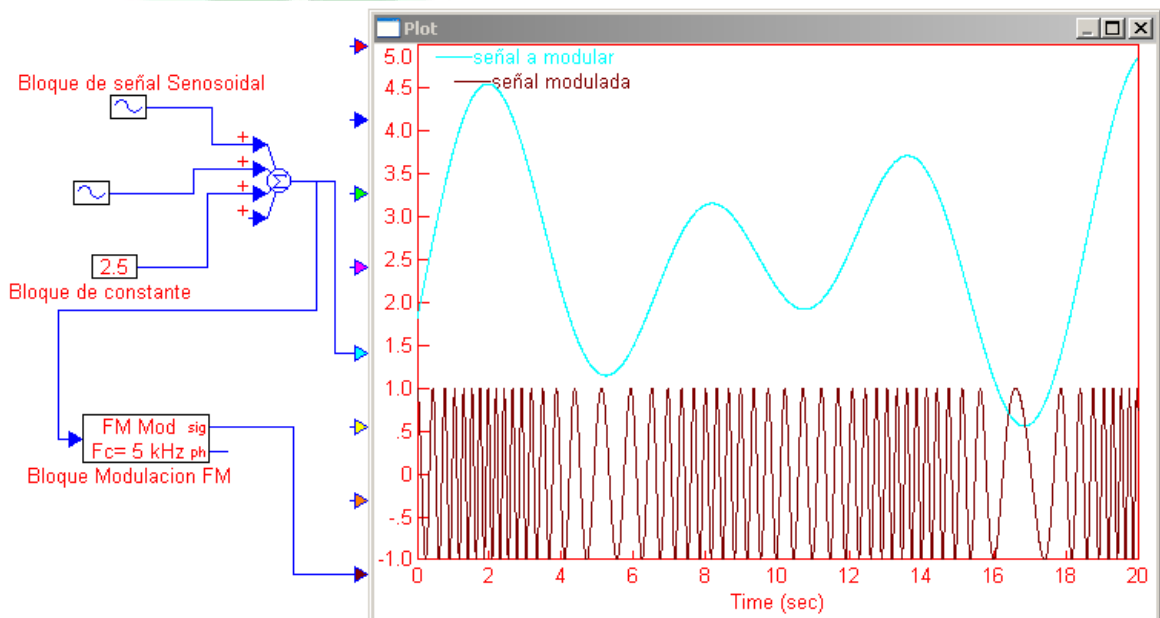
El procedimiento es el mismo, se crea la señal en banda base a partir de la suma de varias señales, en el submenú Comm se localiza la modulación en frecuencia, se configuran los parámetros para la modulación, a saber, Frecuencia portadora Amplitud y Desviación en frecuencia.

La señal moduladora debe ser creada a partir de la suma de tres señales de diferentes, en este caso se sumaron dos señales senoidales y una constante.

Esa señal es modulada en frecuencia a través del modulador FM, por medio de una portadora de frecuencia 5KHz.

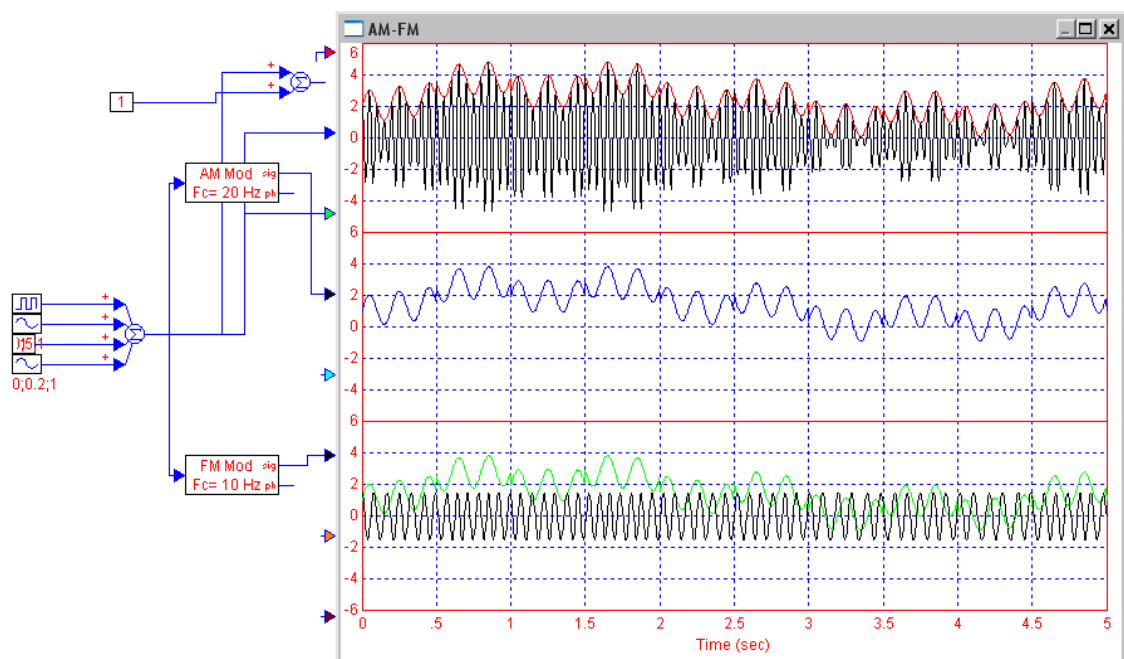
Se observa en la grafica 4.2.5 que a medida que cambia la forma de onda de la modulante creada se cambia de manera instantánea la frecuencia de la señal modulada.

Figura 4.2.5. Grafica de señal de banda base (color azul) y la señal modulada (color café), la señal en banda base modula la frecuencia de la portadora.



El ejemplo de la figura 4.2.6 que sigue a continuación compara las técnicas de modulación AM y FM que usan una forma de onda de entrada arbitraria. Se utilizaron diferentes frecuencias portadoras para AM y FM.

Figura 4.2.6. Grafica de señal de banda base (color azul) y la señal modulada en amplitud (superior), y la señal modulada en frecuencia (inferior).



La señal en banda base fue generada por la suma de 4 señales análogas de diferentes amplitudes y frecuencias. La señal portadora para AM fue utilizada a 20 Hz y para FM 10 Hz. Para la señal de AM se utilizó un índice de modulación de 1, mientras que para FM se utilizó un índice de 5.

4.2.2 Modulación digital

En la modulación digital, la señal transmitida puede asumir un número finito de estados, normalmente llamado “tamaño del alfabeto” o “tamaño de la constelación” de la señal modulada. Éstos incluyen niveles de amplitud discreta, fase discreta, frecuencias discretas, o combinaciones de las anteriores. Los ejemplos de técnicas de modulación digital incluyen desplazamiento de Fase (PSK), Modulación de Amplitud en Cuadratura (QAM), desplazamiento de Frecuencia (FSK) y Modulación por Posición de Pulso (PPM). Cada una de las técnicas anteriores puede llevarse a cabo con distintos niveles de complejidad, dependiendo principalmente del número total de estados (la constelación) que se permite en el modulador.

La modulación digital tiene inherentes beneficios en relación a la modulación analógica, ya que sus distintos estados de la transmisión son más fáciles de detectar en la presencia de ruido que una señal analógica la cual puede asumir un número infinito de valores. Cuando la señal transmitida en forma digital se origina como un forma de onda analógica, un intercambio ocurre subsecuentemente en la fase de codificación; y cierta información se pierde en el proceso de cuantización.

4.2.1 Modulación FSK y MSK

En la modulación por desplazamiento de Frecuencia (FSK), la información digital se transmite asignando frecuencias de salida discretas a cada uno de los posibles símbolos de entrada. La amplitud de la portadora permanece constante. Existen dos clases de moduladores FSK: aquellos que mantienen una fase continua en la portadora entre los estados, y aquellos que no hacen. El primer caso se aplica cuando un sintetizador de frecuencia se usa para generar la salida modulada. El segundo caso ocurre cuando se usan múltiples osciladores independientes para generar los distintos tonos FSK. El ancho de banda utilizado por una señal FSK es directamente proporcional a la tasa de señalización. El Programa *VisSim/Comm* provee un bloque modulador FSK

básico y también un modulador MSK (Minimum Shift Keying) que representa un caso especial de FSK con fase continua, donde los dos tonos de salida son espaciadas exactamente en la tasa de símbolo R .

A continuación se realizara un ejemplo de modulación MSK, el procedimiento se realiza de la siguiente manera:

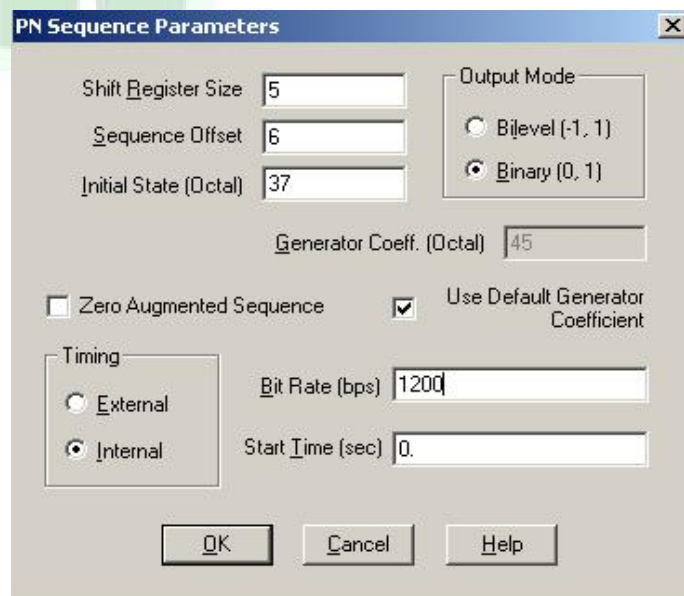
Se ubican los bloques importantes (*PN sequence, MSK Mod y Plot*), se simula, y se configura el bloque de modulación MSK para obtener un resultado que sea visible de dicha simulación, tal cual se muestra en la figura 4.2.9.

La configuración de los bloques utilizados es la que sigue:

Bloque *PN sequence* de creación de la señal de entrada:



Figura 4.2.7. Ventana de parámetros para el bloque de *PN Sequence*



PN Sequence Parameters

Shift Register Size: 5

Sequence Offset: 6

Initial State (Octal): 37

Generator Coeff. (Octal): 45

Output Mode:

- Bilevel (-1, 1)
- Binary (0, 1)

Zero Augmented Sequence

Use Default Generator Coefficient

Timing:

- External
- Internal

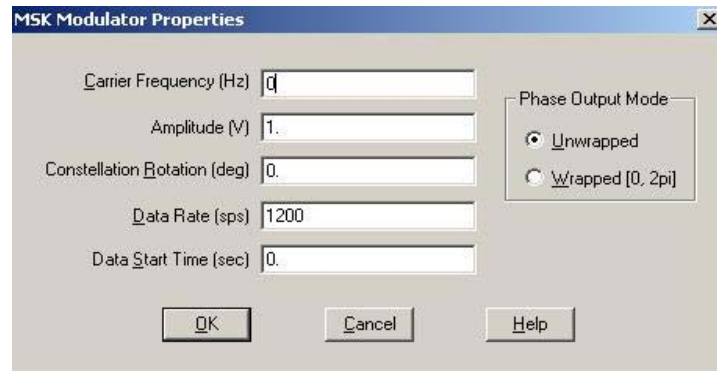
Bit Rate (bps): 1200

Start Time (sec): 0

OK Cancel Help

Ventana de configuración del modulador MSK, es posible modificar la frecuencia de la portadora, la amplitud, la tasa de datos y la rotación de la constelación.

Figura 4.2.8. Ventana de parámetros para el bloque de modulación MSK.



De acuerdo a la configuración anterior el resultado de la onda modulada será el siguiente:

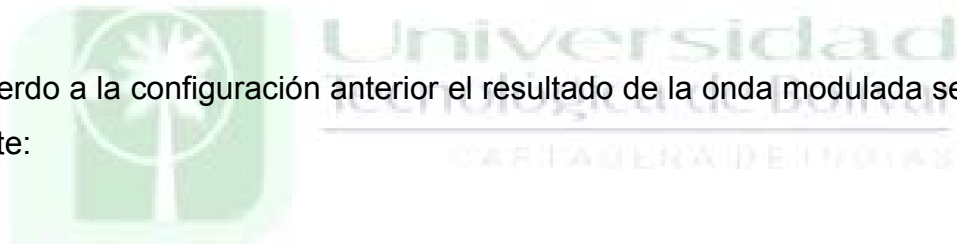
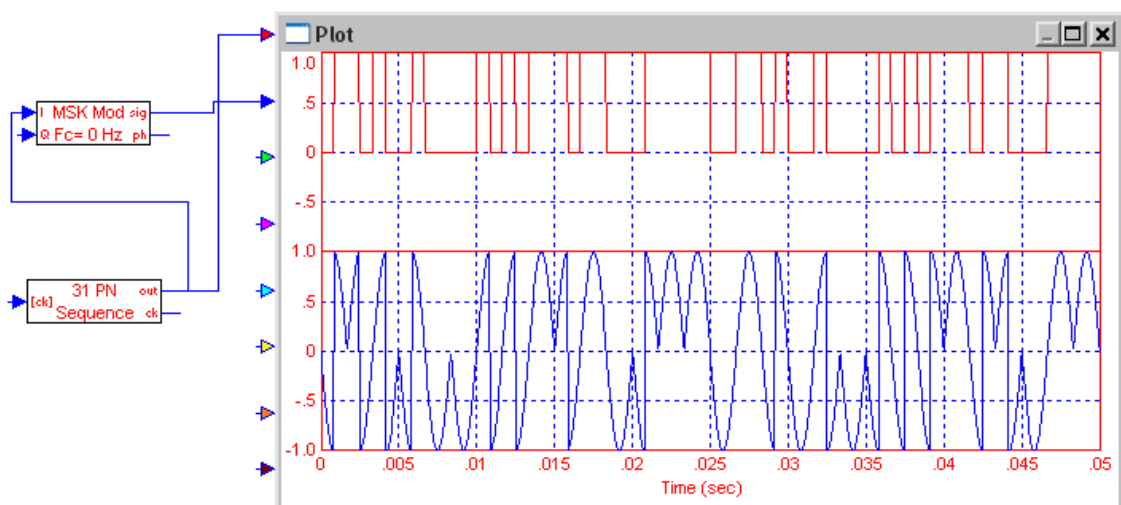


Figura 4.2.9. Señal digital de *PN sequence* (en la parte superior) y señal modulada en MSK (en la parte inferior)



Se realizara un ejemplo de cómo simular una modulación FSK (ver figuras 4.2.10-4.2.12), utilizando dos tipos de entradas, ya sea *PN Sequence* o *Random Sym*. Como se explico en el nivel 1, se puede encontrar las dichas señales en la siguiente ruta: *COMM/SIGNAL SOURCE/ PN Sequence-Random Sym*. Se ubican los bloques importantes (Random Sym, FSK Mod y Plot), se simula y se configuran los bloques de modulación para obtener un resultado que sea visible de dicha simulación, la cual se puede observar en al figura 4.2.12.

Figura 4.2.10. Ventana de configuración de *Random Symbols*

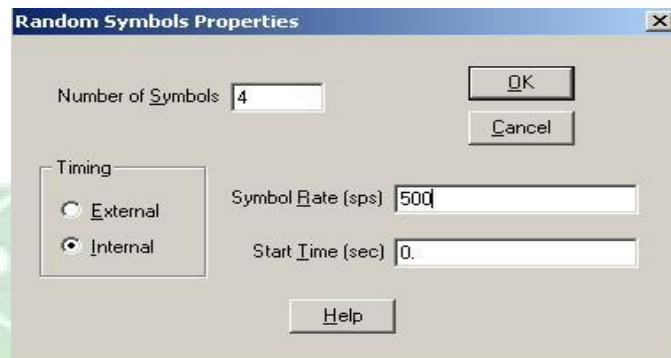


Figura 4.2.11. Ventana de configuración del modulador FSK, configuración de frecuencias superior, inferior, amplitud y fase.

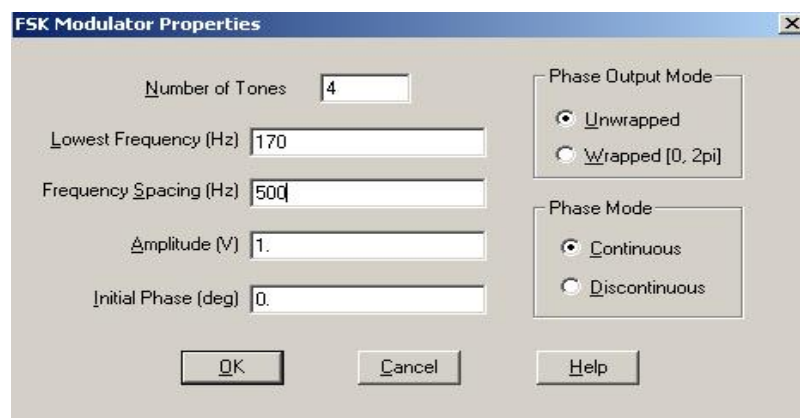
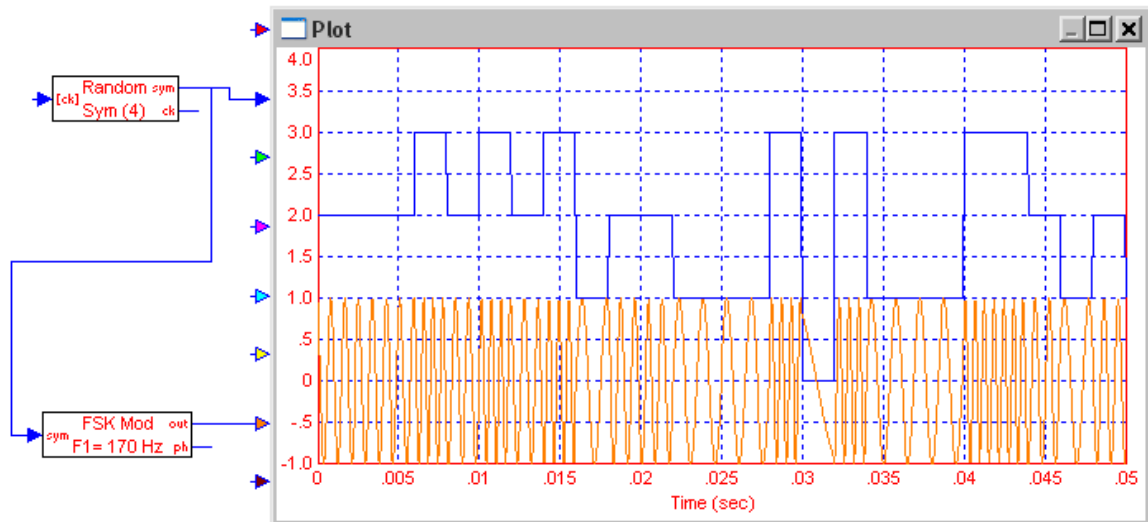


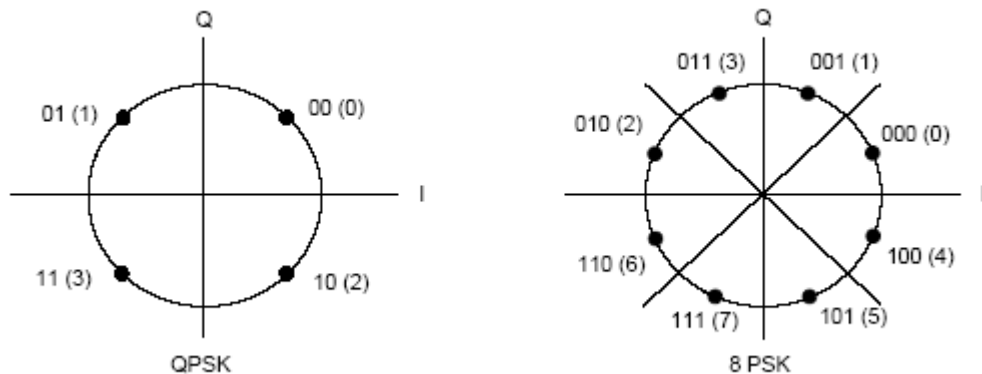
Figura 4.2.12. Ventana de Plot para la modulación en FSK, señal digital en la parte superior y señal modulada en la parte inferior.



4.2.2 Modulación PSK

En la modulación de PSK, la información digital se transmite variando la fase del portador entre los estados de la fase conocidos con la amplitud de la portadora de valor constante. Esta es una propiedad altamente deseable y se conoce como envolvente constante, dado que todos los puntos de la constelación tienen igual potencia. El ancho de banda ocupada por una señal PSK es directamente proporcional a la tasa de símbolo. Así cuando aumenta el tamaño de la constelación, no aumenta el BW. Por otro lado, como los puntos de la constelación están más cerca, se exige una potencia más alta para mantener una BER dada. *VisSim/Comm* ofrece los formatos PSK siguientes: BPSK, QPSK, SQPSK, 8-PSK, y 16-PSK. El usuario posee control completo sobre la asignación del valor de la entrada versus puntos de constelación de salida. Los ejemplos de la figura de mas abajo, ilustran las constelaciones en el plano (I,Q) para QPSK y 8-PSK con una asignación del tipo Gray. Tal mapeo asegura que los puntos de constelación vecinos difieren por sólo un bit. IQ.

Figura 4.2.13. Diagrama de constelaciones de Modulación PSK



Un ejemplo de una modulación BPSK en el dominio temporal y el espectro de una señal QPSK se muestran en las figuras siguientes. En relación al ancho de banda (distancia entre nulos) de una señal PSK es $2R$, donde R representa la tasa de señalización de símbolo. En el caso de QPSK, la tasa de símbolo es $\frac{1}{2}$ de la tasa de bits, y cada punto de la constelación representa dos bits. El espectro de salida está centrado alrededor de la frecuencia de la portadora f_c del modulador.

Este bloque realiza Phase Shift Keying (PSK), la modulación de la señal de entrada sobre los parámetros de modulación seleccionados.

Este bloque pertenece a la familia de moduladores digitales. Acepta su entrada como una señal binaria (BPSK solamente) o un símbolo y número de mapas a la constelación de punto especificado en el archivo de mapa PSK.

x_1 = Entrada símbolo número (entero $[0 \dots N - 1]$, donde N es el tamaño de la constelación)

x_2 = Entrada de reloj (tren de impulsos)

y_1 = señal modulada ($[Re, Im]$ complejas)

y_2 = fase de portadora no modulada (rad)

Frecuencia portadora

Indica la salida de frecuencia portadora f_c en hercios. Se puede establecer en 0 cuando se trabaja en el complejo sobre la representación.

Desequilibrio de ganancia

Especifica la ganancia de desequilibrio (Q relativa a I) del modulador en unidades de dBs. Un valor positivo corresponde a una mayor potencia en el eje de cuadratura que en el eje de fase.

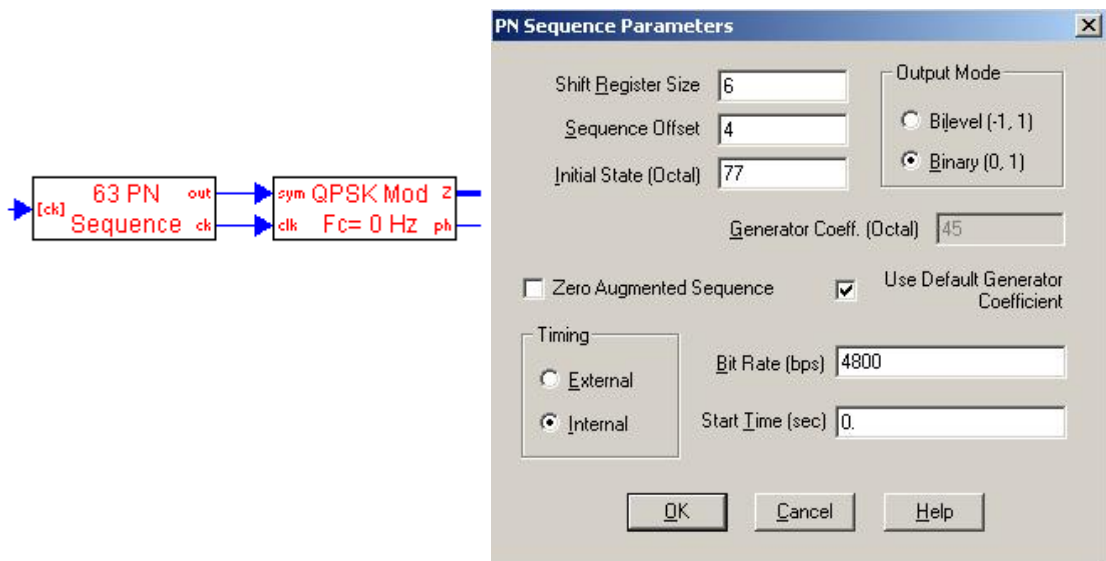
Desequilibrio de fase

Especifica la fase de desequilibrio del modulador en grados como una desviación de la ideal. Un valor positivo corresponde a una rotación en el sentido horario del eje Q relativo al eje I. Por ejemplo, 10 grados de desequilibrio supone un ángulo de 80 grados entre los ejes I y Q, en lugar de los 90 grados ideal.

A manera de ejemplo se configuró de la siguiente manera:

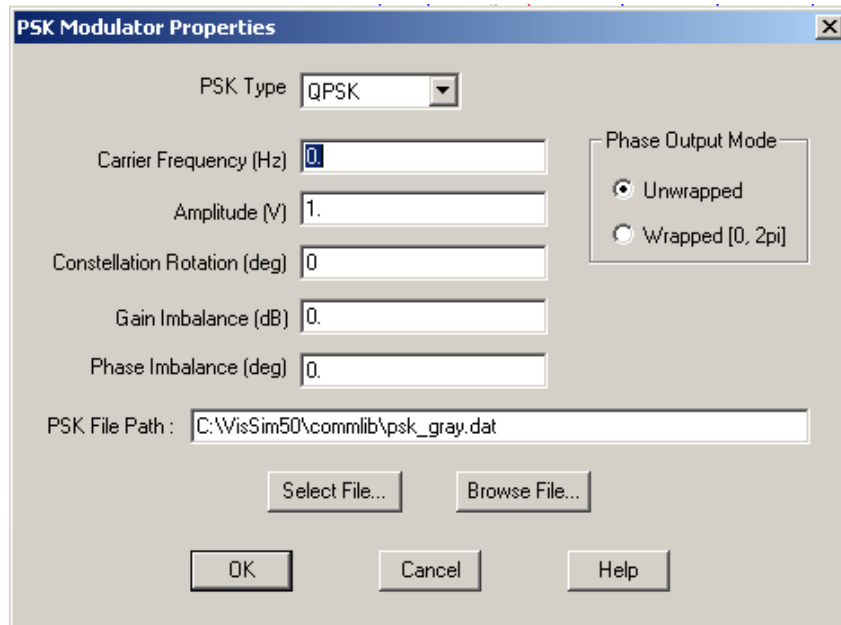
La señal de entrada que se aplicó corresponde a *PN SEQUENCE*, y su configuración fué la siguiente:

Figura 4.2.14. Ventana de configuración *PN sequence* para modulación en PSK



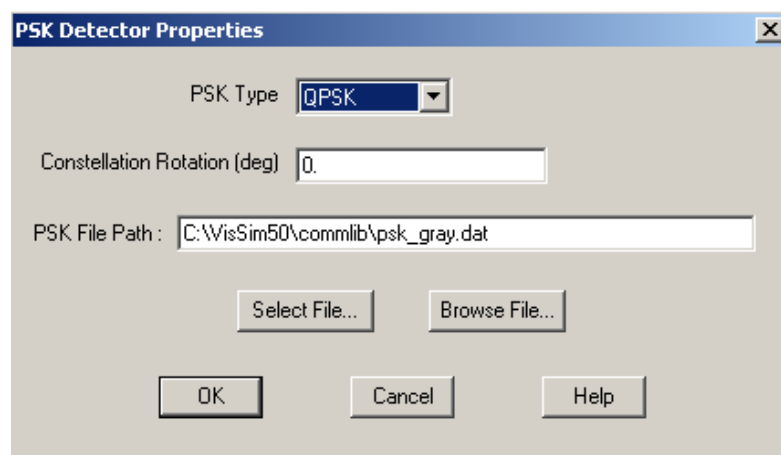
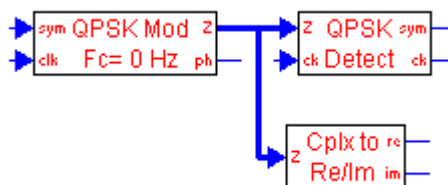
Este modulo se conecta directamente con el modulador PSK que se configura:

Figura 4.2.15. Ventana de configuración de Modulador PSK



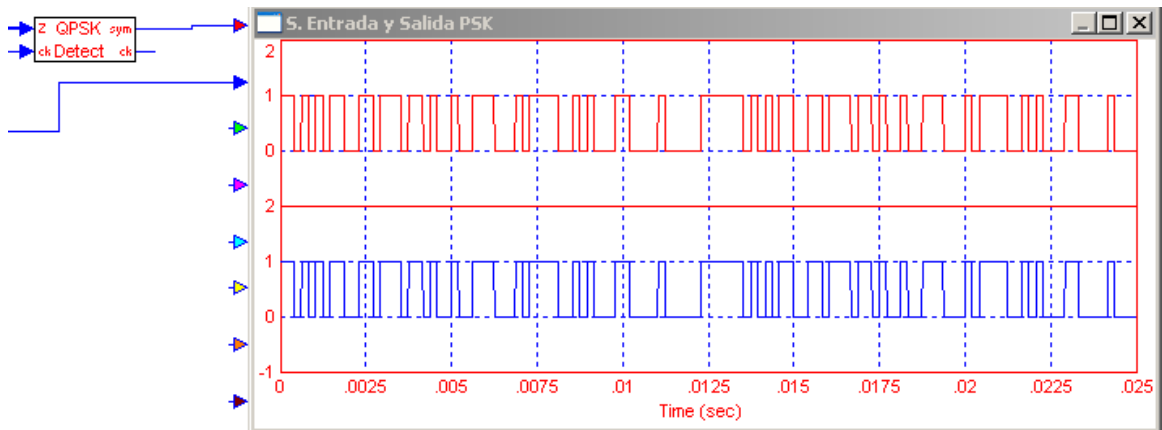
La salida del modulador se envía al *detector de PSK*, que configura de la siguiente manera:

Figura 4.2.16. Ventana de configuración del *Detector de PSK*



De la salida del demodulador (detector QPSK) y comparando con la señal de entrada las señales de información obtenidas serán las de la figura 4.2.17

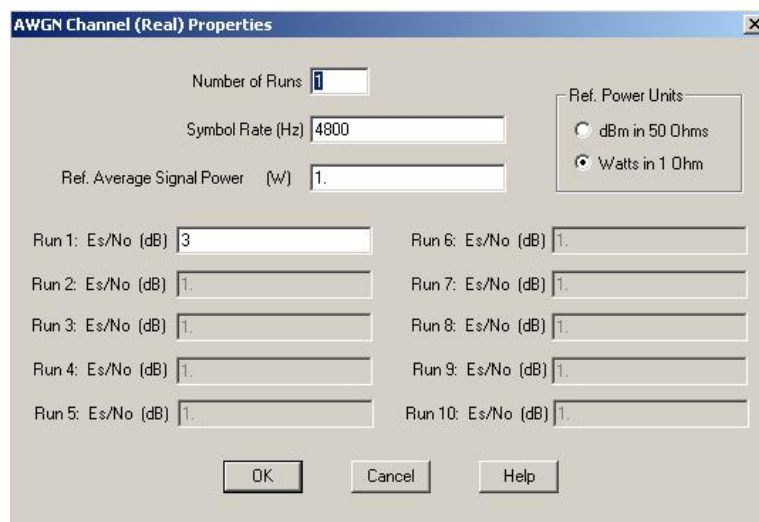
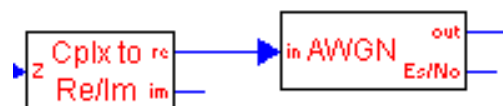
Figura 4.2.17 Señal de información digital en la parte inferior del *Plot* y señal de salida después de la modulación y demodulación QPSK sin añadir error por canal.



A continuación se agregará un canal de transmisión que le sumara un error a la transmisión:

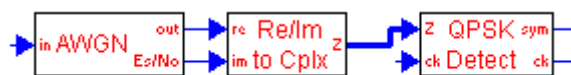
Para este objetivo se le agrega un bloque *Complex to real/imag* a la salida del modulador QPSK. El bloque *complex/real imag* no se configura y se envía al bloque que simula un canal de transmisión *AWGN CHANNEL*.

Figura 4.2.17. Ventana de configuración del canal AWGN (REAL).



El bloque *AWGN CHANNEL* se envía al bloque *real/imag to complex* y de allí se envía a otro demodulador (detector de PSK) que se configura de la misma manera que el utilizado anteriormente.

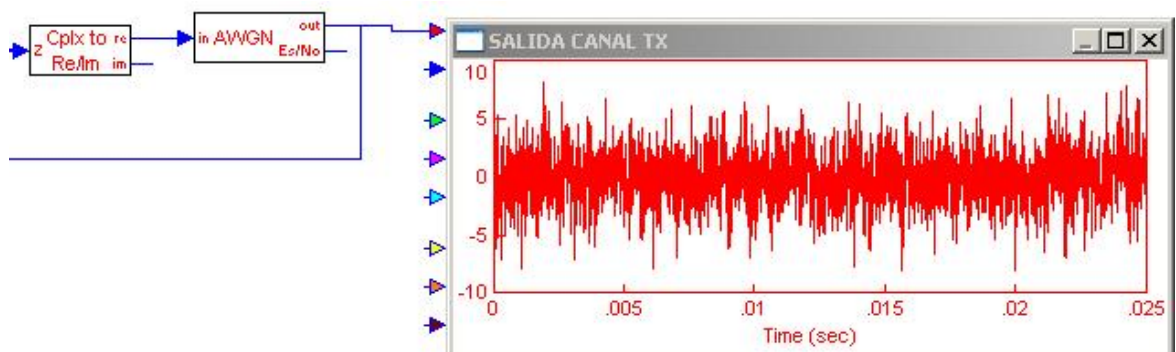
Todo este procedimiento a partir de la introducción del canal *AWGN channel* se realizará con el fin de visualizar la salida de la señal luego de la introducción de errores por parte del canal de transmisión. Y además de esto observar como el demodulador corrige errores y regenera la señal.



Conectamos bloques de *Plot* de la salida de los bloques detectores de QPSK, de la señal de salida del bloque *PN Sequence*, de la salida del bloque *AWGN*.

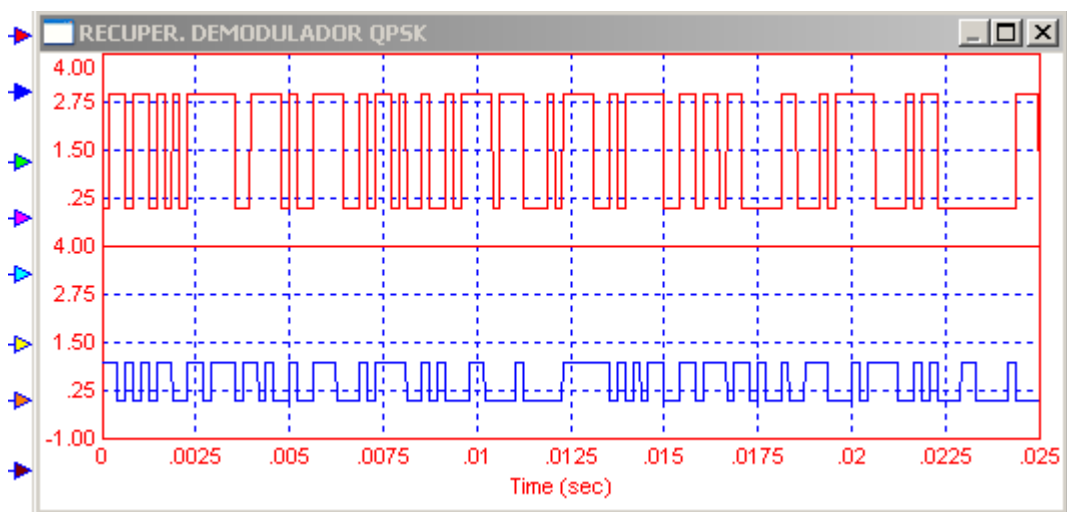
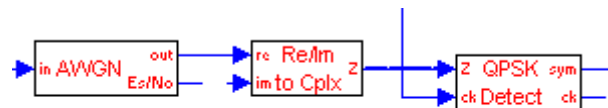
A la salida del bloque (*AWGN*) de simulación del canal de transmisión la señal fue la siguiente:

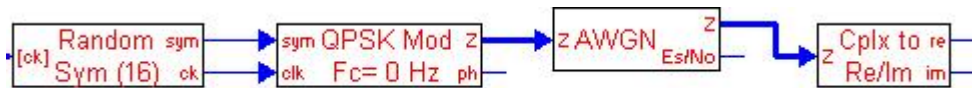
Figura 4.2.18. Señal de salida luego de simularse pasar por un canal de transmisión.



Después del canal de transmisión se realiza la recuperación de la señal por medio del segundo detector de QPSK y de acuerdo a la organización de bloques explicada anteriormente se recupera la señal original.

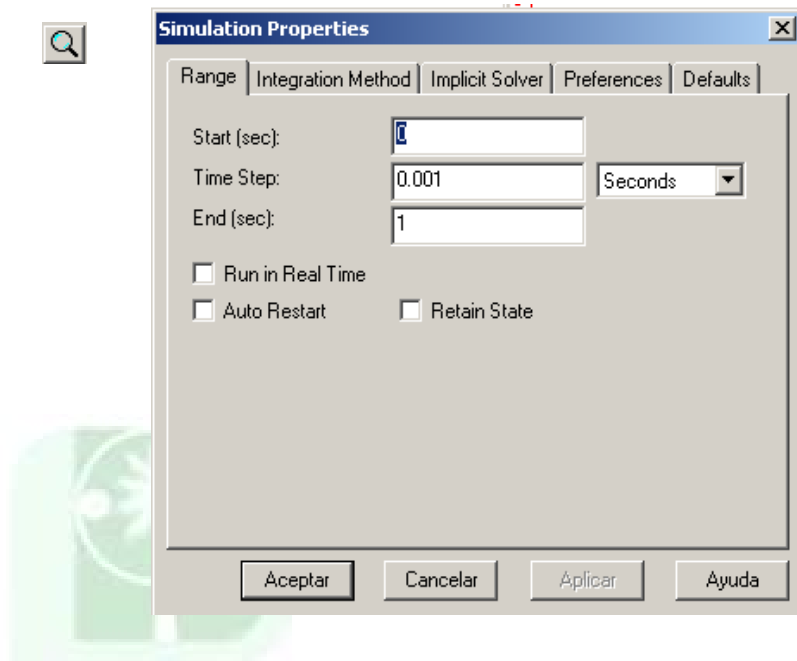
Figura 4.2.19 En la parte superior, Señal recuperada a partir de la modulación pasada por el canal de transmisión, en la parte inferior señal original.





La ventana de configuración global del software se configura así para simular:

Figura 4.2.21 Botón y ventana de configuración de propiedades de simulación globales



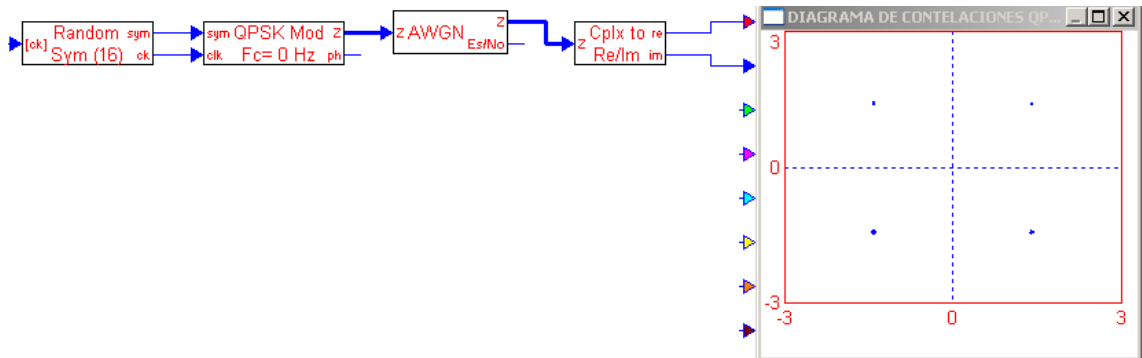
Todo este procedimiento permite simular el diagrama de constelaciones de 4 puntos correspondiente a QPSK.

Figura 4.2.22 Diagrama de Constelaciones modulación BPSK.



El bloque completo que permite la visualización del diagrama de constelaciones es el siguiente:

Figura 4.2.23 Diagrama de Constelaciones modulación BPSK.



El siguiente ejemplo (ver figura 4.2.24) *VisSim/Comm* compara dos técnicas de modulación digital que usan una señal de entrada digital arbitraria. La siguiente figura muestra la banda base de la señal de entrada que puede asumir uno de cuatro niveles [0-3]. Las dos señales adicionales muestran ondas moduladas en FSK y QPSK.

Figura 4.2.24. Muestra de dos tipos de Modulación FSK (superior) y QPSK (inferior), además de la señal moduladora (parte central)

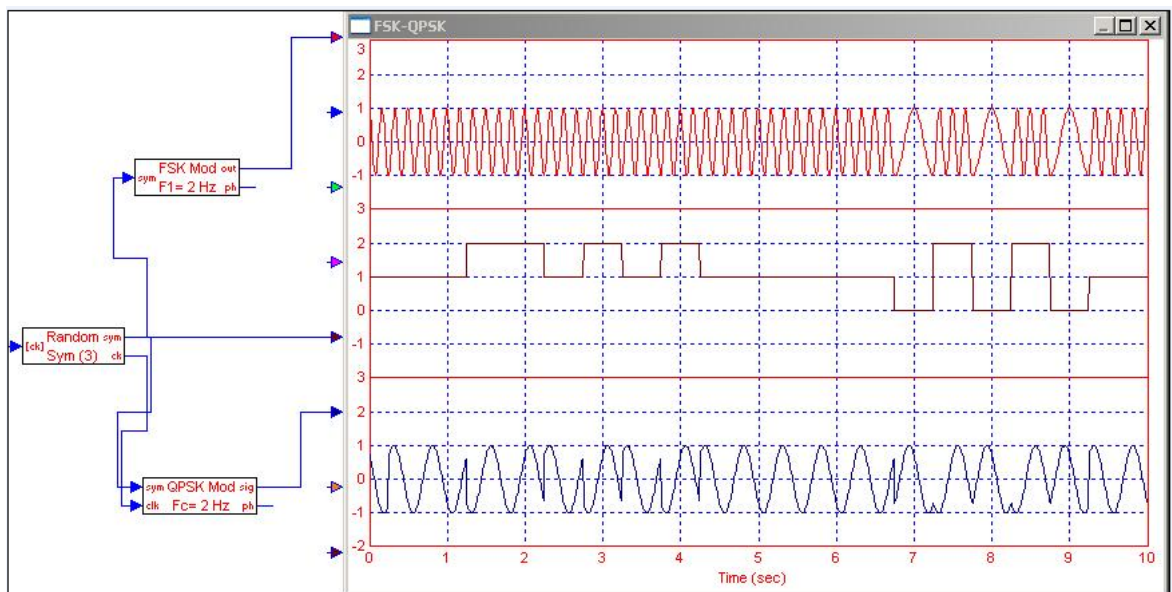
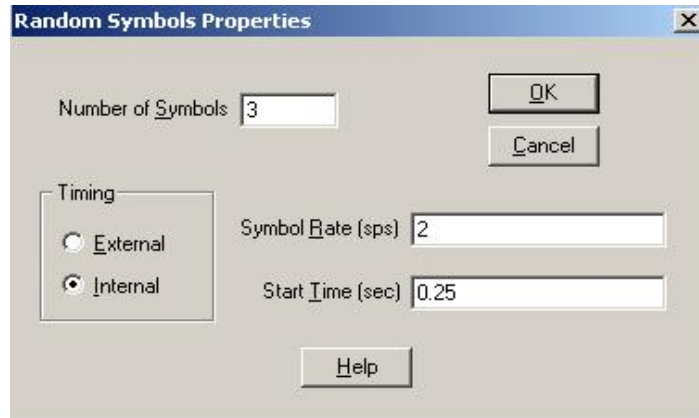
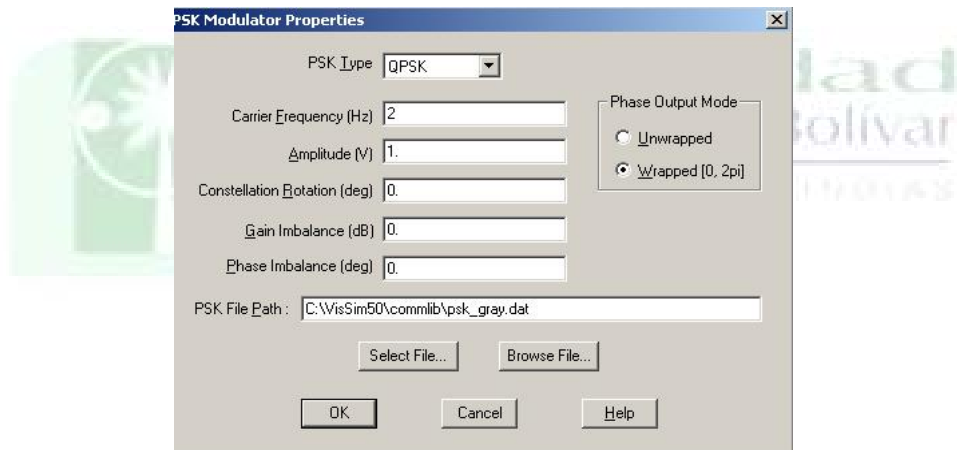


Figura 4.2.25. . Ventana de configuración de la fuente de señales digitales



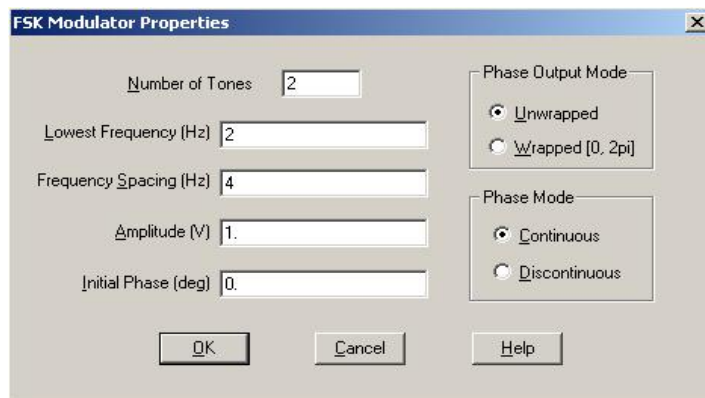
El modulador QPSK fue configurado de la siguiente manera:

Figura 4.2.26. Ventana de configuración de modulación QPSK para este ejemplo.



El modulador FSK fue configurado de la siguiente manera:

Figura 4.2.27. Ventana de configuración de modulación FSK para este ejemplo.



4.2.3 ACTIVIDADES NIVEL II

4.2.3.1 Modulación en amplitud

Procedimiento

1. Sume dos señales sinusoidales de 3Hz y 1Hz con 1 y 1.5 respectivamente, al resultado de la salida de este sumador increméntelo con un nivel DC de 2.5 V y grafique el resultado. A la señal de banda base obtenida en el primer sumador multiplíquela con una señal que actuara como portadora de 40 Hz, 1V, al resultado del producto súmele la portadora amplificada con un factor de 2.5, para obtener la expresión matemática de una señal AM Convencional. Grafique esta señal modulada con otro color en la primera grafica.

Ahora vamos a recuperar la señal de banda base , por los dos métodos clásicos:



Universidad
Tecnológica de Bolívar
CARRIACANA DE BOLÍVAR

Demodulación coherente

Detección de envolvente

Demodulación coherente

Este método no fue aplicado durante la guía pero su aplicación solo consiste en aplicar una señal de igual frecuencia a la señal de la portadora y luego pasarla por un filtro pasa bajos.

2. La señal modulada la vamos a multiplicar por la portadora generada en el receptor (40 Hz, 1 Volt), la salida del multiplicador le colocamos un filtro butterworth de orden 6 pasa bajos de 5 Hz y la visualizamos en otra grafica.

Detección de envolvente

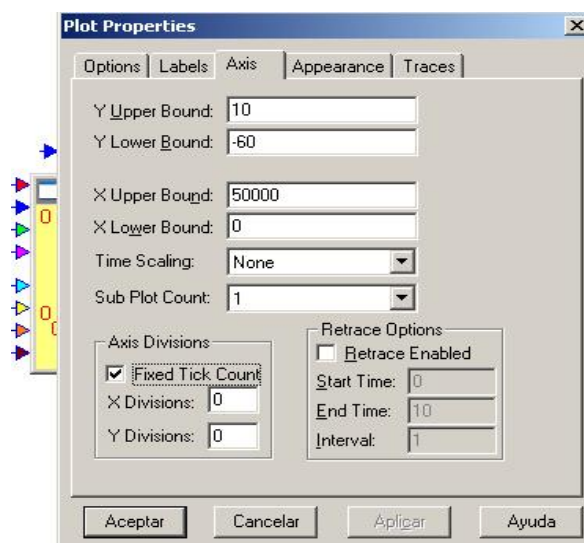
3. A la señal modulada la entramos a un Detector de Envolvente (*Peak Rectifier*) y en la otra entrada le aplicamos una constante de 0.2 y a la salida le conectamos otro *filtro butterworth* de orden 6 pasabajos de 5 Hz y la visualizamos en otra grafica, y la salida del Detector de Envolvente la graficamos para compararla con la salida del filtro.

4.2.3.2 Modulación en Frecuencia

Procedimiento

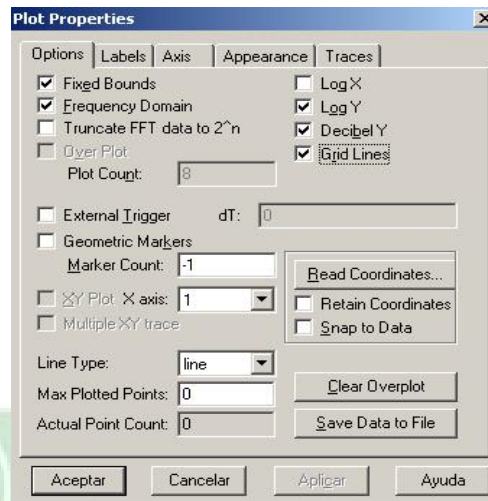
1. Generamos una señal senoidal de 2 Khz, la graficamos en el Dominio del tiempo y la conectamos al Modulador de FM, el cual bajamos de la barra de menú Comm/Modulador Real, y le programamos una portadora de 25 Khz, amplitud de 1 V y una Desviación de frecuencia de 5000 Hz/volt. La graficamos en el cuadro anterior el cual denominaremos Dominio del tiempo. Y esta misma salida la conectamos a un grafico que rotularemos dominio de la frecuencia y el cual programaremos de la siguiente forma:

2. Figura 4.2.28. . Ventana de configuración del Plot, se configuran los ejes para una buena visualización.



A continuación se configura el bloque *Plot* para que grafique en el dominio de la frecuencia.

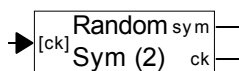
Figura 4.2.29. . Ventana de configuración de Plot para colocarlo en el dominio de la frecuencia.



3. Simule y busque la manera de buscar parámetros que se puedan variar y ver el comportamiento del espectro.

4.2.3.3 Modulación FSK

1. Aplicar una señal de dos niveles, a una velocidad de 1 sps(símbolo por segundo) generada con el bloque:

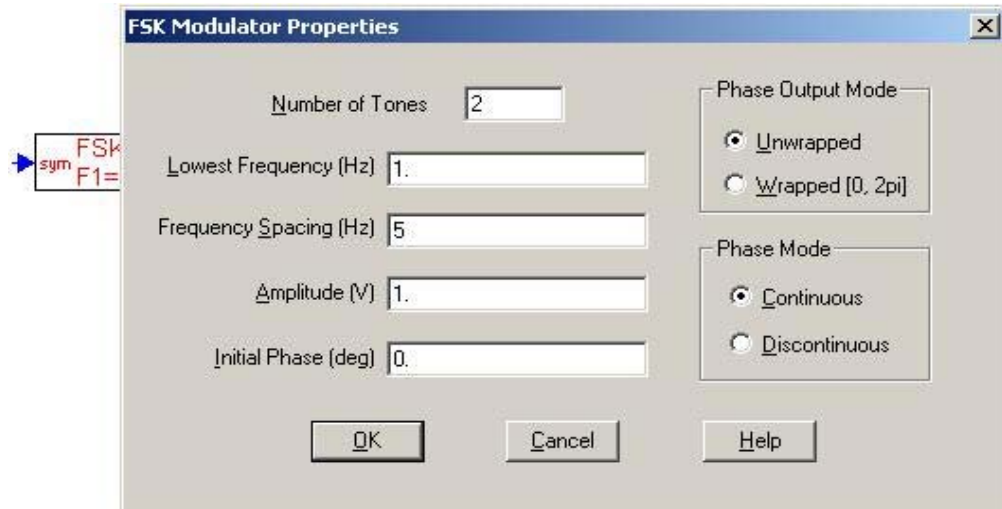


2. Conecte la salida a un filtro con estas características (¿Para qué?)



3. La salida del filtro llévela al *Plot* y a la entrada del Modulador FSK, y prográmelo de acuerdo al siguiente cuadro de dialogo

Figura 4.2.30. Ventana de configuración del modulador FSK para la práctica.

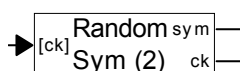


4. Conecte la salida del modulador a Plot. No se le olvide conectar la señal rampa a los Plot.

5. Con el operador FFT, analice el espectro de la señal moduladora antes del filtro y después del filtro. Corra la simulación y guarde sus datos.

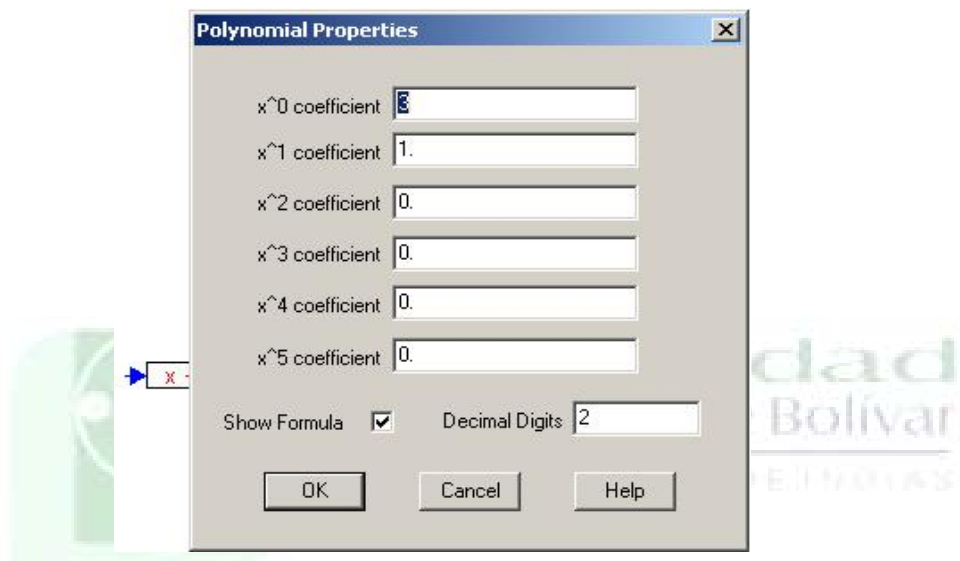
4.2.3.4 MODULACION PSK Y QPSK

1. Aplicar una señal de dos niveles, a una velocidad de 10 sps(símbolo por segundo) generada con el bloque:



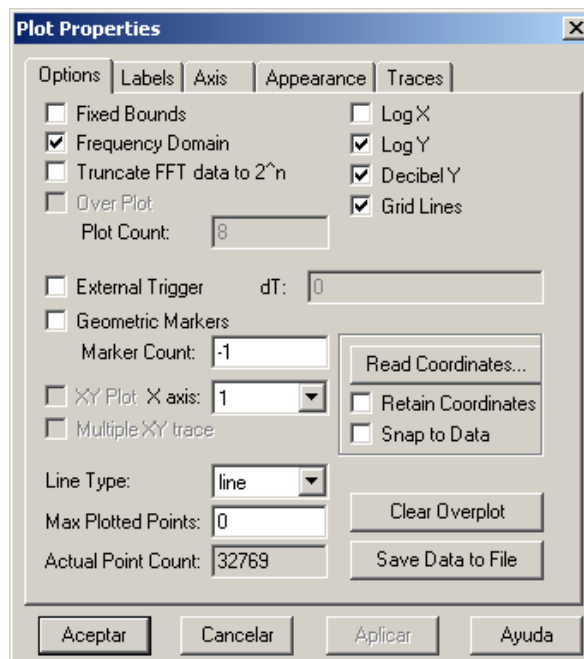
2. Las salidas del generador llévela al Plot. Y a las entradas del Modulador PSK (Menu, COMM/MODULATORS-REAL/ PSK(REAL) y prográmelo con una frecuencia portadora de 50 Hz
3. Conecte la salida del modulador al *Plot*. Con el fin de que no se superponga la señal, antes del PLOT, subala de nivel con *OPERATOR/POLINOMIAL* y programe el cuadro de dialogos así:

Figura 4.2.31. Ventana de configuración del operador Polinomial



- 4 Conecte la salida del modulador a otro *Plot*, prográmelo en el Dominio de la frecuencia, de esta manera.

Figura 4.2.33 Ventana de configuración del *Plot* para respuesta en frecuencia.



5. Simule y por supuesto guarde el archivo.

Figura 4.2.33 Ventana de configuración de la fuente de señales digitales.

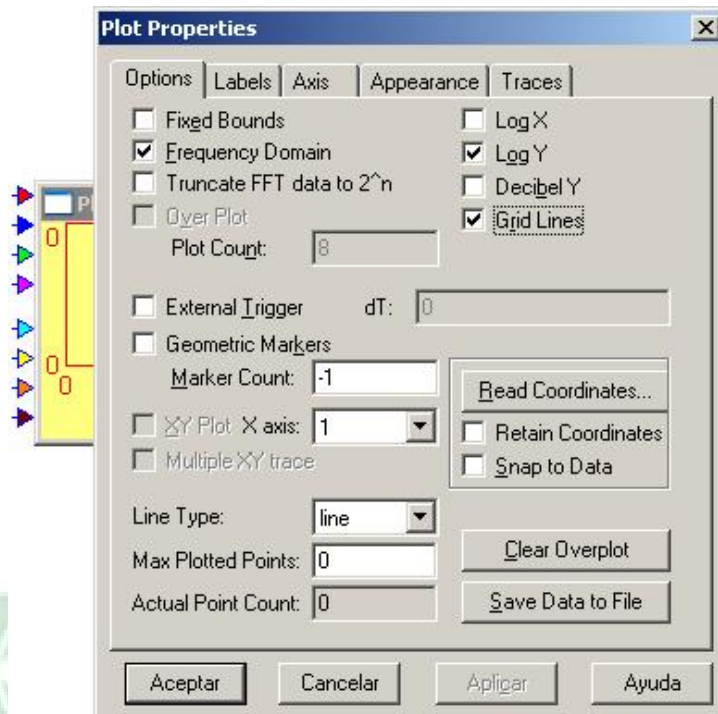


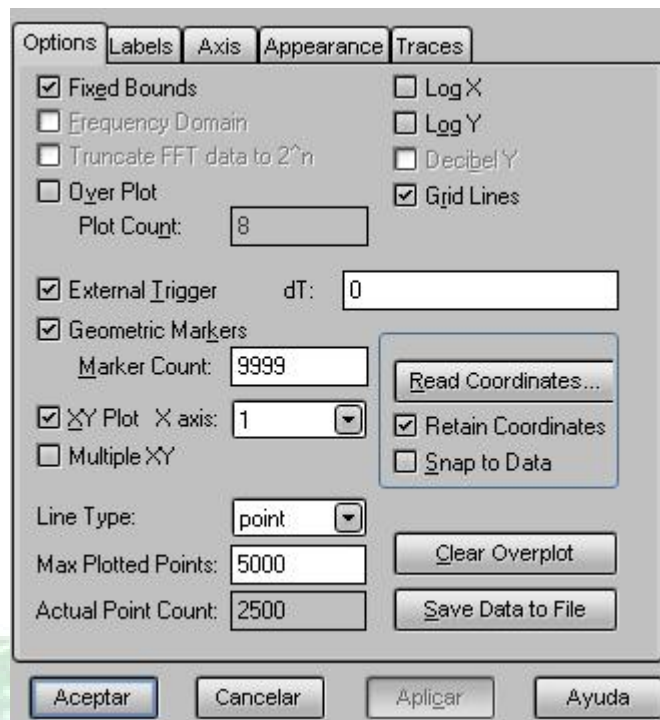
DIAGRAMA DE CONSTELACIONES

Obtener el diagrama de constelaciones de los diferentes tipos de modulaciones.

Utilice e interconecte los siguientes bloques:

- *Comm/Signal source/Random Symbol*
- *Comm/Modulators Complex/PSK*
- *Comm/Channel/Awgn(Complex)*
- *Comm/channel/Complex Math/Complex to real/Imag.*
- *Plot* con la siguiente programación

Figura 4.2.33 Ventana de configuración del *Plot*.



4.3 SIMULACIÓN NIVEL III

- **Entrenamiento:** duración estimada 160 min.

- **Actividad:** duración estimada 45 min.

En este nivel se aplicaran los conocimientos de codificación y decodificación y se utilizan los mismos para la detección errores.

4.3.1 VITERBI

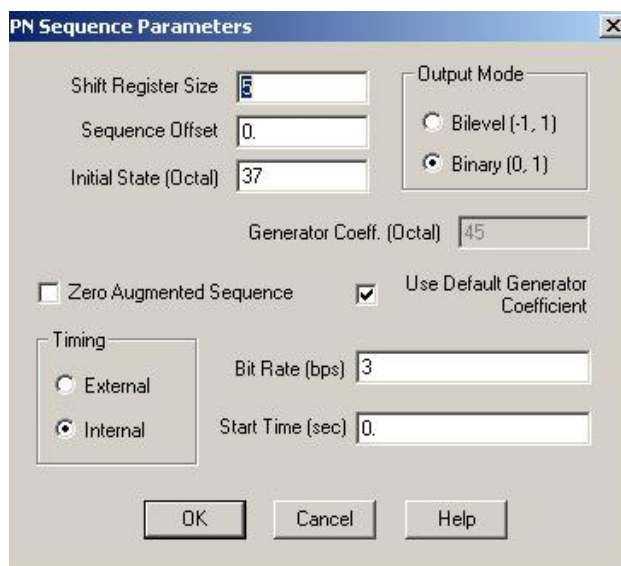
A continuación se presenta un ejemplo que muestra como la codificación/decodificación puede ser utilizada para la detección de errores introducidos por el canal.

El proceso es el siguiente:

Se tiene que ubicar los bloques en la pantalla de acuerdo con la configuración deseada, todo esto siguiendo los parámetros de los sistemas de comunicaciones tratados en el marco teórico, iniciando con la fuente de información (*PN Sequence*), pasando por el codificador (*Convolutional Encoder*), seguido del canal de transmisión (*BSC*) y finalizando con el decodificador de Viterbi (ver las figuras 4.3.1 – 4.3.4).

Comenzamos con la señal que se va a enviar o input, denominada “*PN SEQUENCE*” de la siguiente manera:

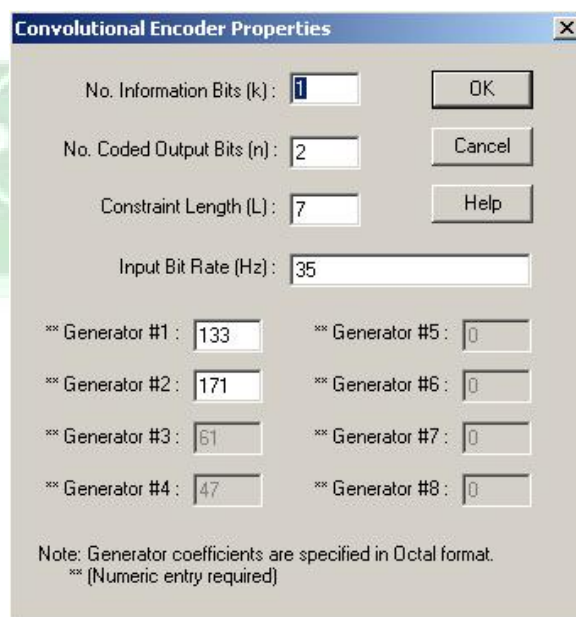
Figura 4.3.1. Ventana de configuración de la fuente de señales digitales *PN sequence*.



En la figura 4.3.1 se observa las posibles modificaciones que se le pueden realizar a este bloque en especial, de acuerdo con las características que se necesitan representar.

De la misma forma como se ubicó el bloque anterior se selecciona el bloque “Convolutional Encoder” el cual será el encargado de codificar a la señal generada por el *PN Sequence* en la siguiente ruta: *Comm/Encode-Decode/Convolutional Encoder*. En la siguiente imagen observamos las propiedades que podemos modificar de este bloque.

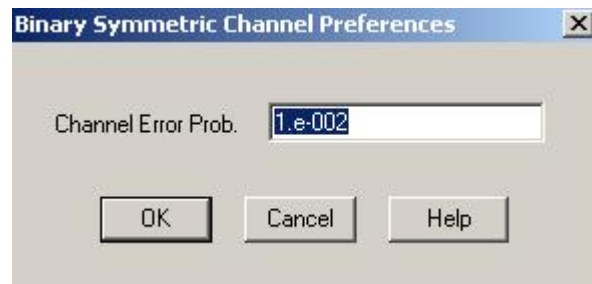
Figura 4.3.2. Ventana de configuración Del codificador convolucional.



Ahora de selecciona el bloque del canal, denominado *BSC* en la siguiente ruta: *Comm/Channels/Binary Symetric Channel*.

A diferencia de los bloques anteriores, el bloque del canal de comunicación solo presenta una posible opción a modificar denominada nivel de probabilidad de error. Se recomienda probar con diferentes valores hasta un valor máximo de uno (1) y analizar los cambios obtenidos.

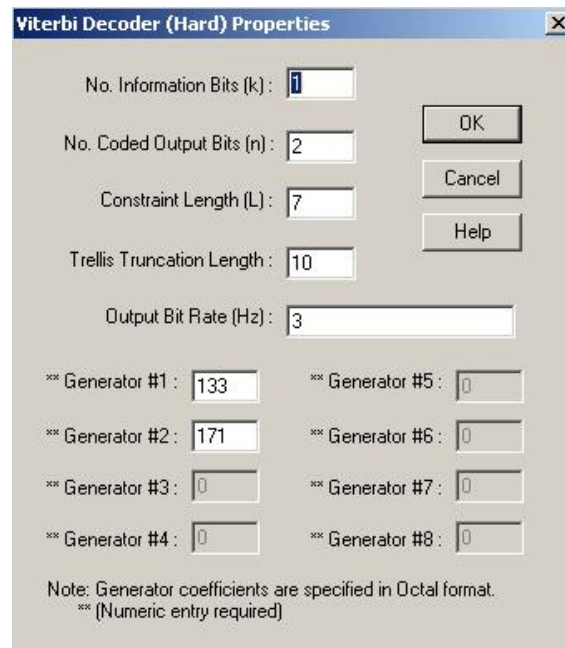
Figura 4.3.3. Ventana de configuración BSC para diferentes probabilidades de error.



Finalmente se selecciona el bloque de decodificación de Viterbi el cual se encuentra en la siguiente ruta: *Comm/Encode-Decode/Viterbi Decoder*. Este bloque como los dos primeros presenta una variedad de propiedades a editar, las cuales se podrán modificar dentro de un rango razonable de valores. A continuación se muestra la configuración utilizada en este ejemplo.

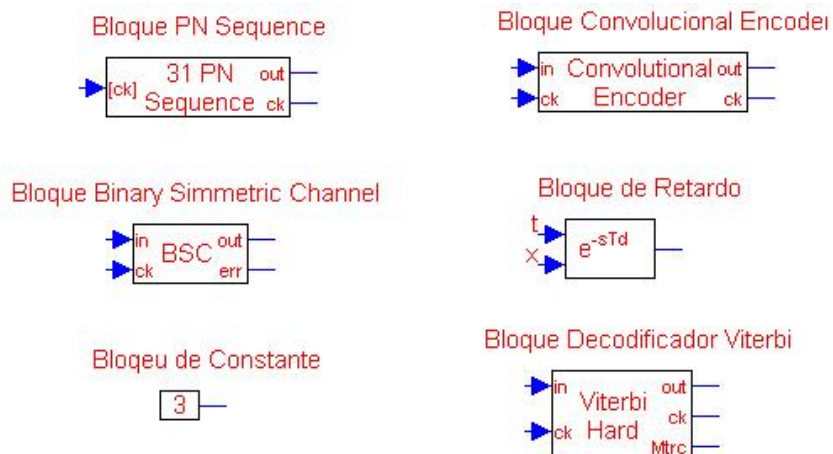
Se recomienda realizar este ejemplo con ambos tipos de bloques *Viterbi (Hard y Soft)* y analizar los resultados obtenidos.

Figura 4.3.4. Ventana de Configuración del Bloque Viterbi Hard.



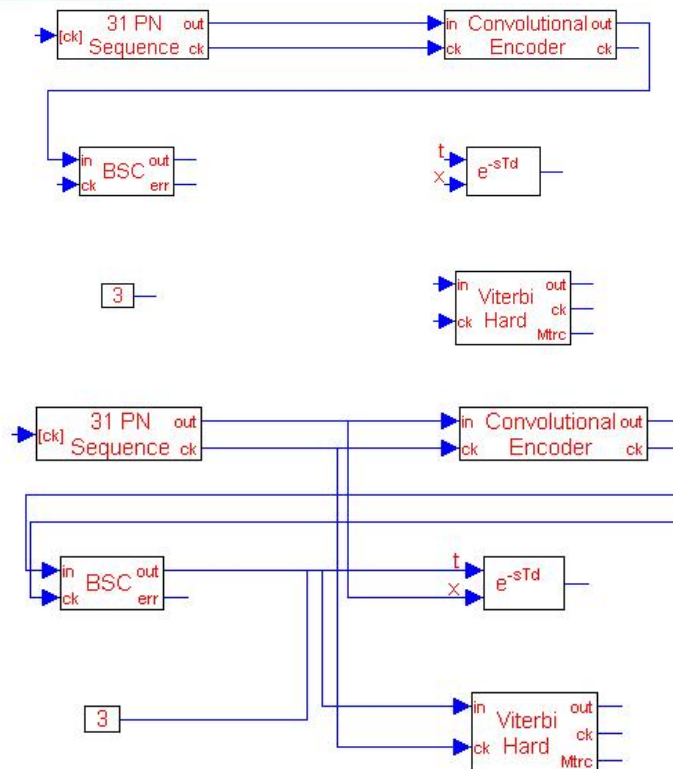
Para terminar se realiza las conexiones entre bloques y se ubican tres ventanas de *PLOT* en las cuales visualizaremos los resultados obtenidos o realizamos una serie de *SUBPLOTS*. Primero ubicamos todos los bloques a utilizar de la siguiente manera:

Figura 4.3.5. Ubicación de bloques para la simulación de Codificación/Decodificación Viterbi.



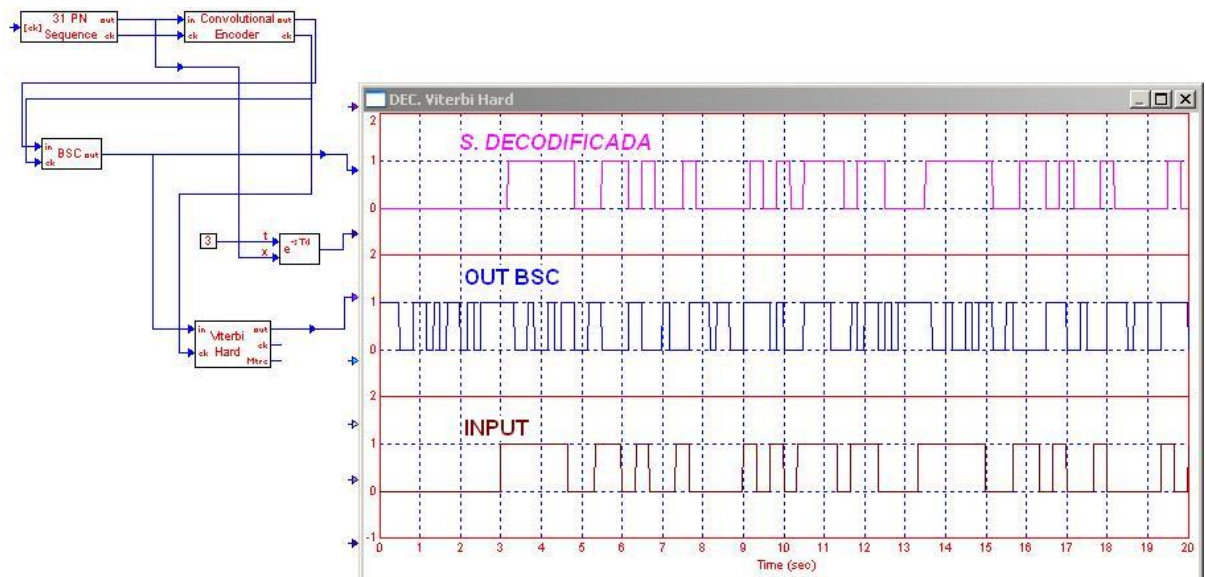
Realizamos la conexión de los bloques como sigue:

Figura 4.3.6. Conexión de bloques para la simulación de Codificación/Decodificación Viterbi.



Finalmente se obtiene la conexión de la figura 4.3.6, y se procede a conectar las salidas deseadas al plot. Para este caso particular se conecta las salidas de los siguientes bloques: *TimeDelay*, *Convolutional Encoder*, *BSC* y *Viterbi Hard*.

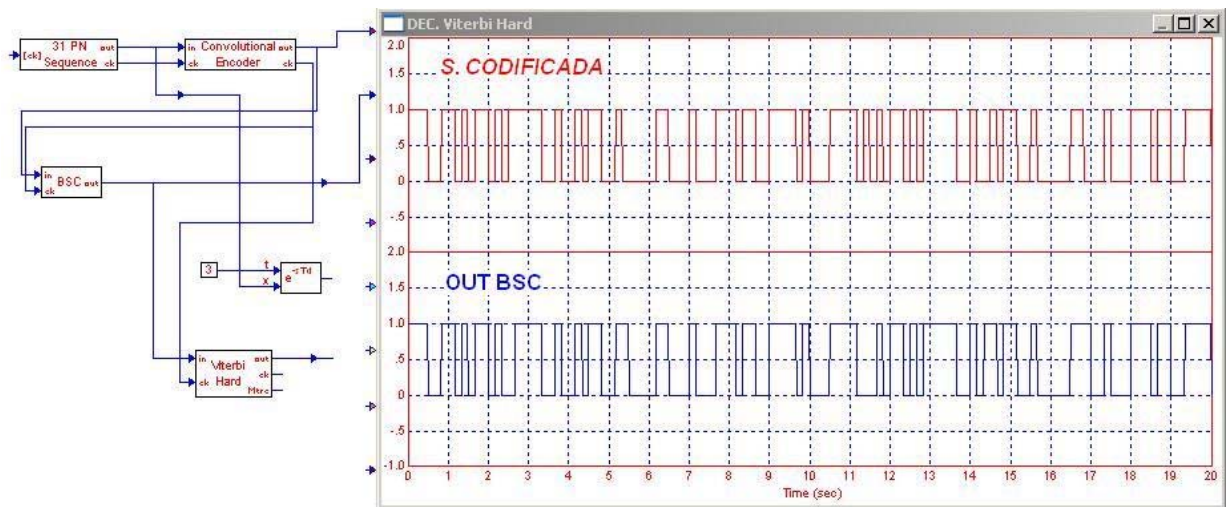
Figura 4.3.7. Simulación de decodificación Viterbi, codificación convolucional



En la figura 4.3.7 se observa todo el proceso creado con la configuración explicada donde la señal transmitida es codificada en el bloque Convolutional Encoder, luego transmitida por el canal BSC (bloque "Binary Symmetric Channel") el cual introduce un cierto porcentaje de error (el deseado por el usuario) y finalmente pasa por el decodificador de Viterbi el cual muestra la señal obtenida al final de dicho proceso, por medio de la detección de errores explicada en la sección de Viterbi. Se pueden realizar pruebas con diferentes niveles de probabilidad de error para observar las diferencias en la salida del bloque Viterbi.

A continuación en la imagen 4.3.7 se realiza una comparación entre la señal de salida del *Convolutional Encoder* con la señal del bloque *BSC*, con el fin de analizar el error introducido por el canal a dicha señal.

Figura 4.3.8. Simulación de decodificación Viterbi, S Codificada Vs. S BSC.



Para mayor comprensión de la simulación realizada se puede hacer una serie de variaciones a los parámetros de cada bloque, variando la señal a enviar, los parámetros de codificación, el nivel de probabilidad de error del canal de transmisión y la tasa de bit de salida, finalmente hacer un análisis de los cambios que se obtiene, en las señales de entrada, de codificación y decodificación respectivamente.

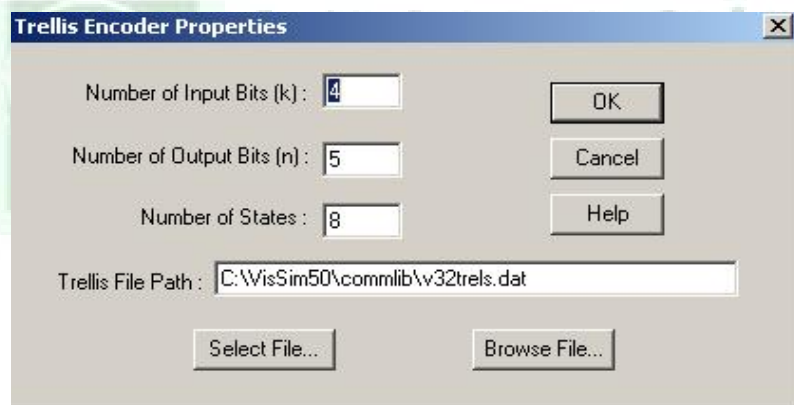
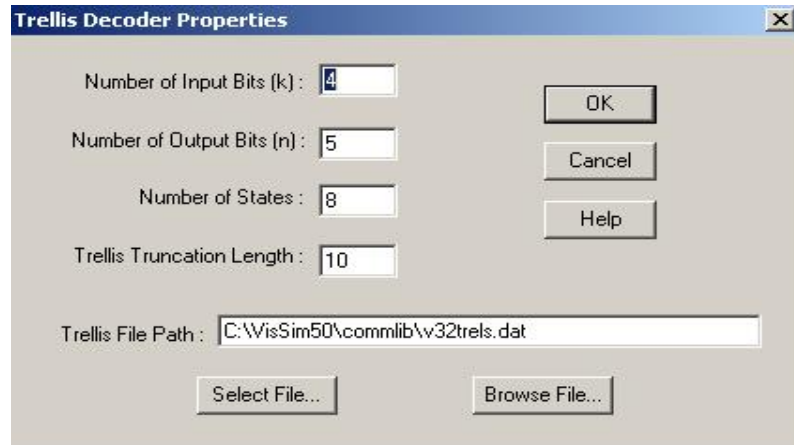
4.3.2 Trellis

A continuación se realiza un proceso de simulación de un esquema Trellis sencillo, en el cual se utilizaran bloques de señal aleatoria, codificación y decodificación Trellis.

Primero se debe elegir una señal a procesar, la cual será en este caso una señal Random de 15 símbolos.

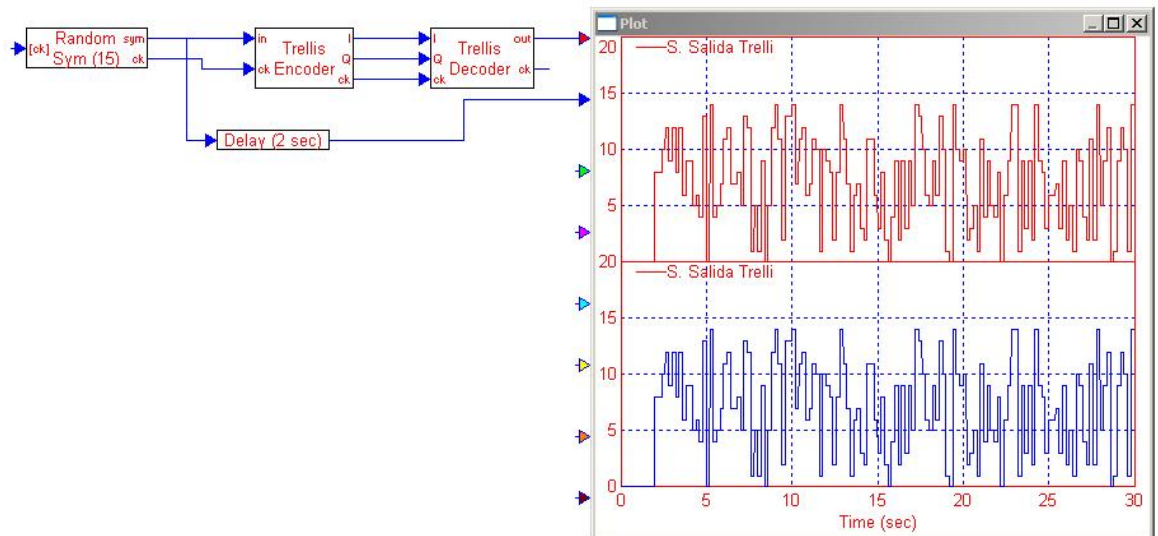
Ahora se selecciona los dos bloques de codificación y decodificación respectivamente, los cuales se encuentran en el submenú COMM, y se configuran teniendo en cuenta la tasa de símbolos que se colocó en la señal de inicio (Random), a continuación se muestra los recuadros de propiedades de cada uno de los dos bloques (figuras 4.3.9).

Figura 4.3.9. Ventanas de configuración para la Codificación y Decodificación de Trellis.



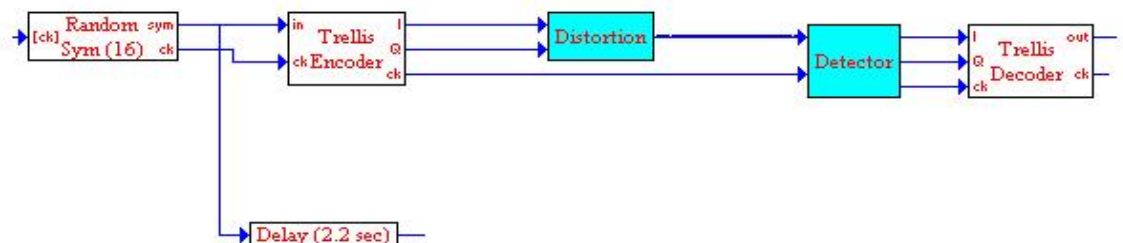
Luego de ubicar en la pantalla de VisSim todos los elementos y conectarlos de la forma que sigue se procede a realizar la simulación, como se muestra en la siguiente imagen:

Figura 4.3.10. Plot de señal de salida de decodificador Trellis (superior color rojo), señal de entrada al codificador (inferior color azul).



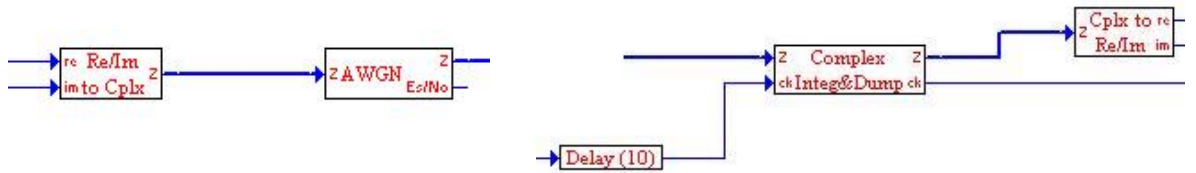
distorsión y un detector a la misma configuración, obteniendo la figura que sigue:

Figura 4.3.11. Configuración con distorsión y detector



En la figura 4.3.11 se cambia a un *Random* de 16 símbolos y a un *Delay* de 2.2 segundos. Los bloques de *Distorsión* y *Detección* se configuran con un bloque *Compound*. Solo se tiene que poner la configuración de bloques deseada, seleccionarlos todos y dirigirse a *Create Compound Block*, en la ruta: *Edit/Create Compound Block*. La configuración de dichos bloques es como sigue:

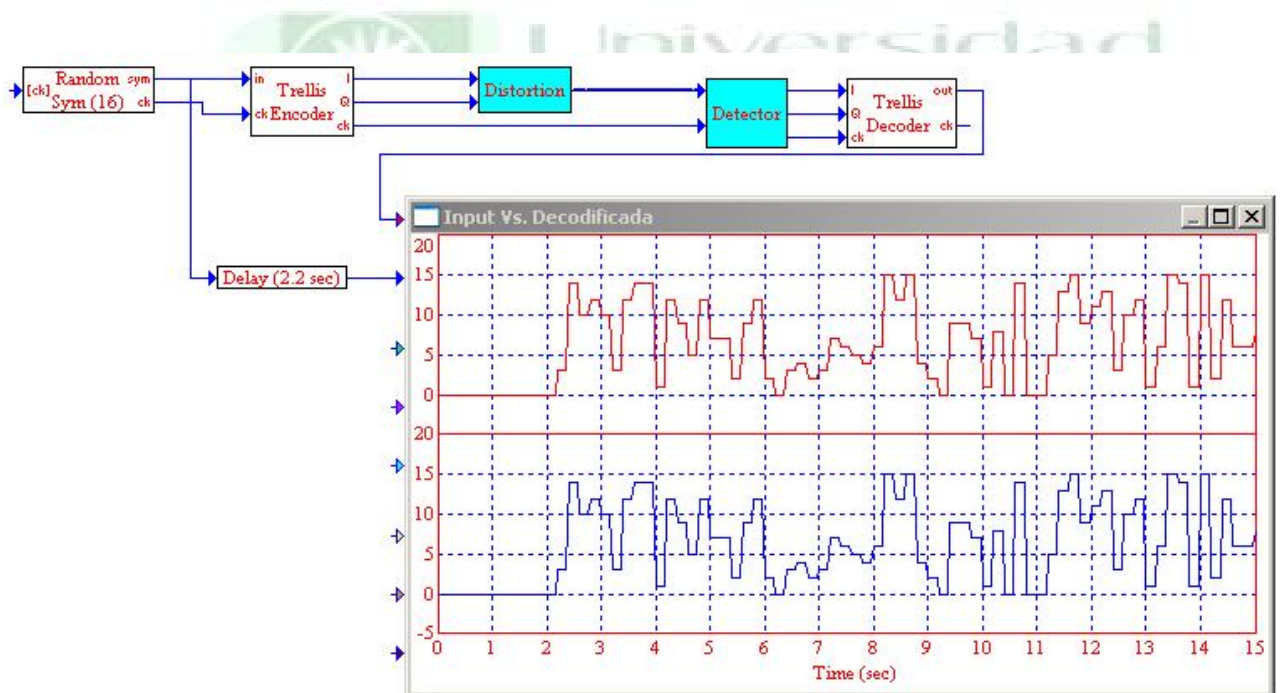
Figura 4.3.12. Izquierda: Bloque Distorsión. Derecha: Bloque Detector.



Solo se hace necesario modificar la configuración del bloque de retardo *Delay*.

Para finalizar se corre al configuración y se obtiene algo similar a lo mostrado en la figura 4.3.13.

Figura 4.3.13. Configuración Trellis.



4.3.3 Código de Bloques

4.3.3.1 Interleaving de Datos

La mayoría de los códigos que se han desarrollado operan bien en un canal con errores uniforme, como aquellos producidos por un canal Gaussiano. Cuando los errores tienden a ocurrir en ráfagas, deben realizarse pasos adicionales para aumentar el performance del proceso de codificación. Una técnica común es scramblear los bits antes de transmitirlos usando un Interleaver.

El entrelazado de datos altera el flujo de bits codificados para minimizar los efectos de las degradaciones de canal tipo ráfaga. Usando esta técnica, los errores de bits adyacentes causados por desvanecimientos de la señal o ráfagas de interferencias son esparcidas en el tiempo en un intervalo de tiempo más largo, dando al decodificador una probabilidad mayor de recuperar la ráfaga de datos original. La figura siguiente ilustra el funcionamiento de un interleaver simple.

4.3.4 Convolutional

Por medio de esta práctica se aprenderá a codificación y decodificación Convolutional, los bloques que utilizaremos en este ejemplo serán los siguientes:

-Ramp

-Int

-SquareWave

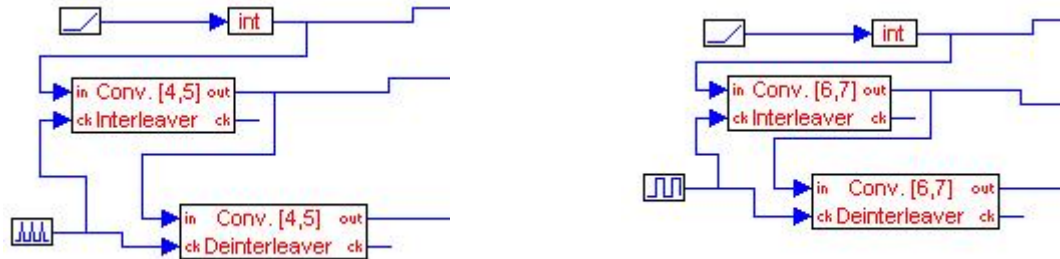
-PulseTrain

-Conv. Interleaver, Deinterleaver

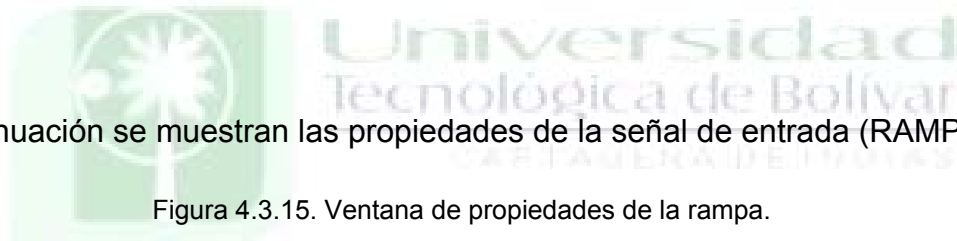
Los pasos a seguir para desarrollar este ejemplo son descritos a continuación.

Ubicar los bloques nombrados anteriormente en la siguiente disposición:

Figura 4.3.14. Configuración para simulación dependiendo de la señal de reloj.



Nota: cada imagen representa una configuración independiente, que pueden desarrollarse en distintos plots.



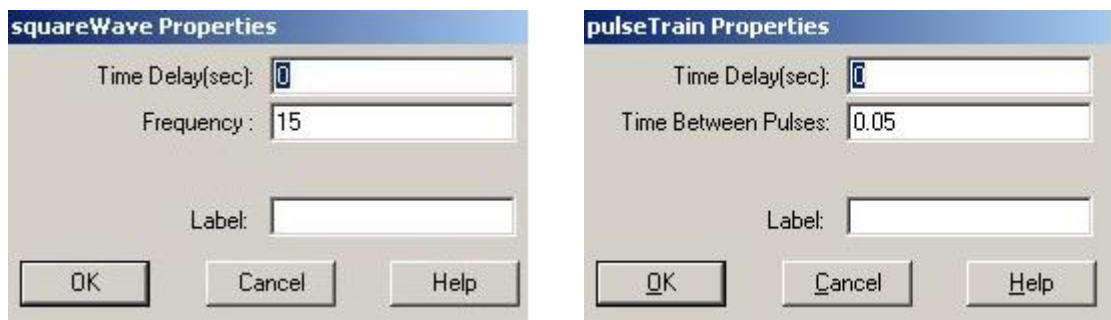
A continuación se muestran las propiedades de la señal de entrada (RAMP).

Figura 4.3.15. Ventana de propiedades de la rampa.

The image shows a dialog box titled 'ramp Properties'. It has three input fields: 'Time Delay(sec):' with a value of 0, 'Slope:' with a value of 7, and 'Label:' which is empty. At the bottom, there are three buttons: 'OK', 'Cancel', and 'Help'.

Las señales de reloj empleadas para los bloques convolucionales (*PULSETRAIN* - *SQUAREWAVE*) deben ser configuradas de la siguiente manera:

Figura 4.3.16. Ventana de configuración de onda cuadrada para señal de reloj (1). Ventana de configuración de tren de pulsos para señal de reloj (2).



Los siguientes dos cuadros de configuración hacen referencia a la descodificación y codificación convolucional del juego de bloques que tiene como referencia de reloj la señal SquareWave.

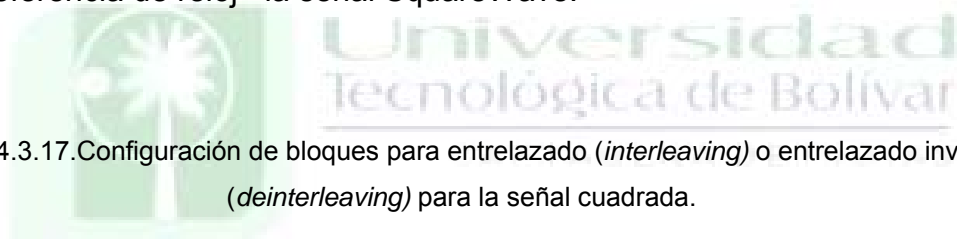
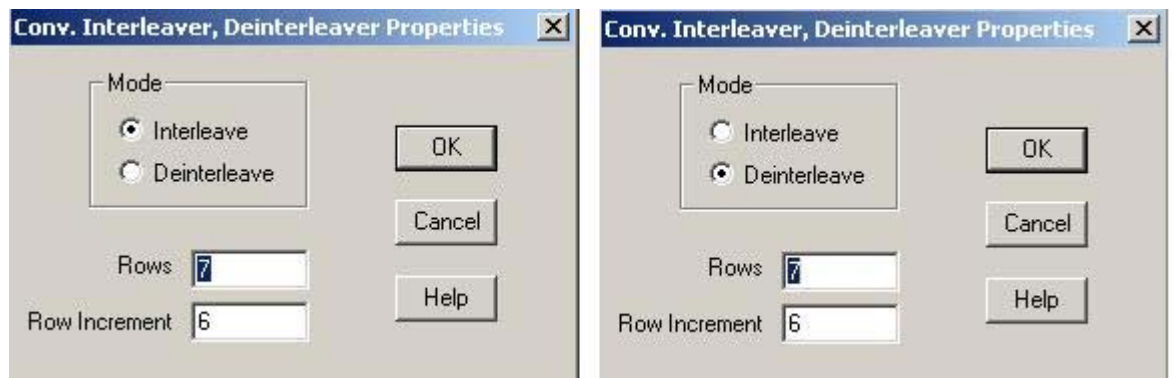
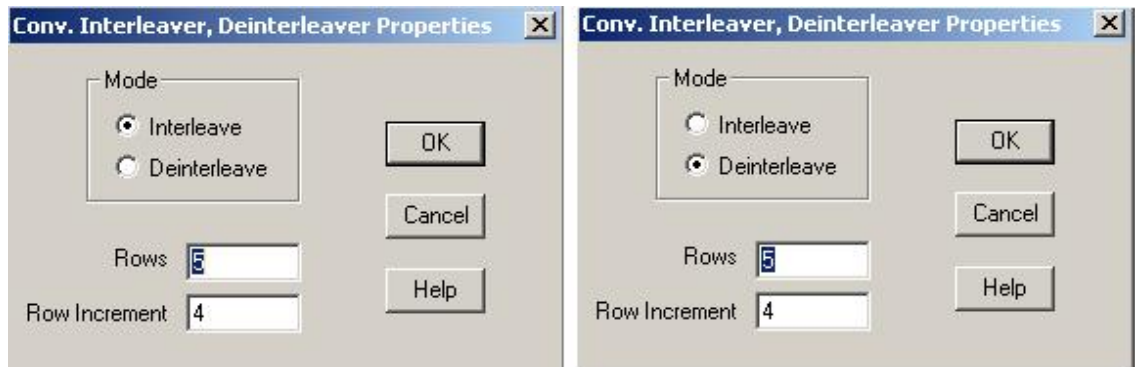


Figura 4.3.17. Configuración de bloques para entrelazado (*interleaving*) o entrelazado inverso (*deinterleaving*) para la señal cuadrada.



Los siguientes dos cuadros de configuración hacen referencia a codificación y descodificación convolucional del juego de bloques que tiene como referencia de reloj la señal PulseTrain (Tren de Pulsos).

Figura 4.3.18. Configuración de bloques para entrelazado (*interleaving*) o entrelazado inverso (*deinterleaving*) para la señal Tren de Pulsos.



Finalmente se ubica un bloque *PLOT* para cada configuración Codificación/Decodificación y se procede a realizar la simulación y con lo cual se debe obtener la imagen mostrada en la figura 4.3.19 y 4.3.20.

Figura 4.3.19. Simulación de señal por medio de código convolucional, señal de entrada (superior), señal de salida (inferior) y señal codificada (intermedio color azul), para señal de reloj de onda cuadrada.

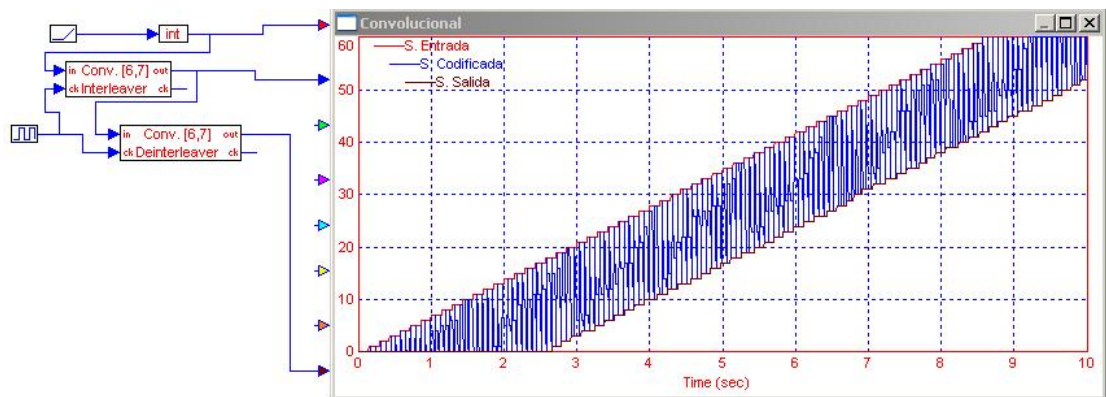
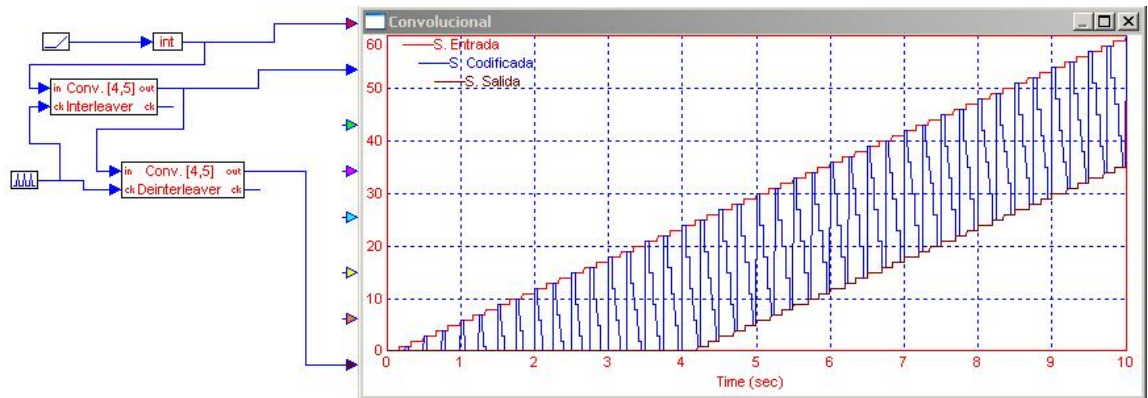
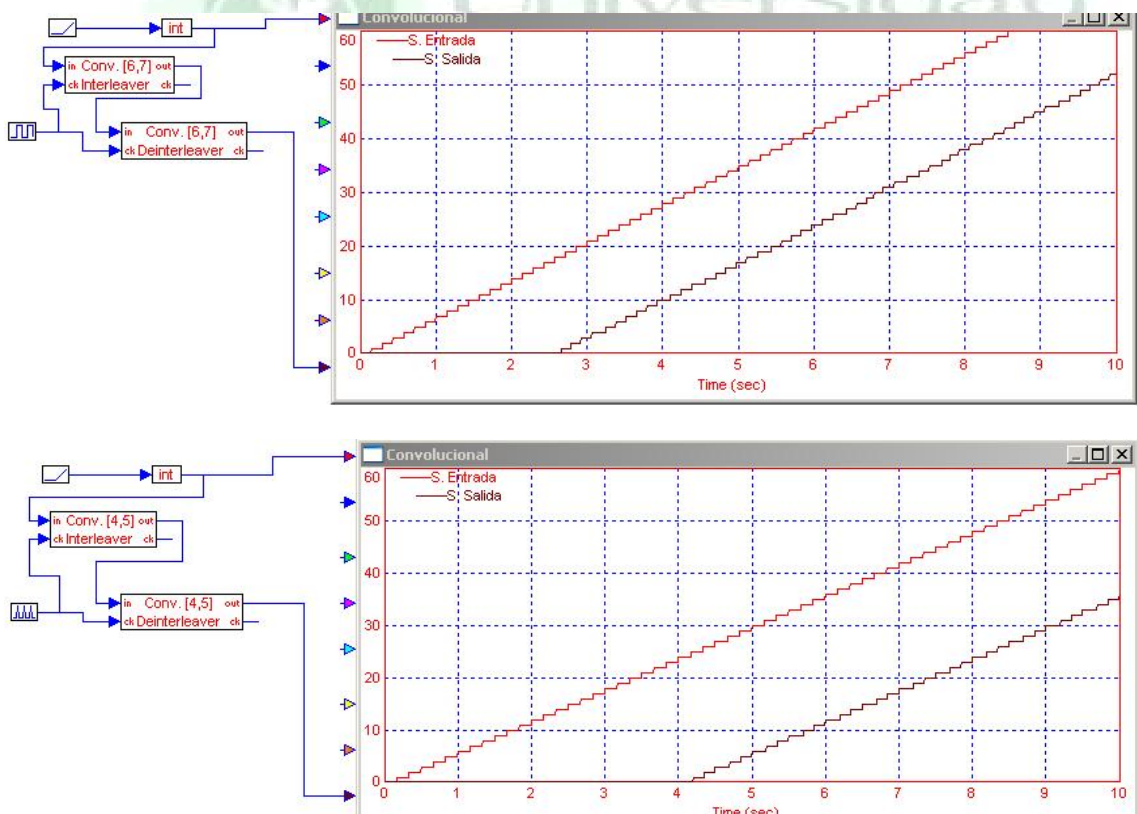


Figura 4.3.20. Simulación de señal por medio de código convolucional, señal de entrada (superior), señal de salida (inferior) y señal codificada (intermedio color azul) para señal de reloj Tren de Pulsos.



Para una mejor percepción del resultado de las anteriores dos simulaciones, se omitirá la señal codificada, para que poder comparar la señal de entrada contra la señal de salida (decodificada), ambas simulaciones son mostradas a continuación en las figura 4.3.21.

Figura 4.3.21. Simulación Convencional sin señal codificada.



Como se pudo observar se completó la simulación exitosamente, ahora analizando lo obtenido en las dos gráficas podemos evidenciar que la misma señal de entrada es la que tenemos en la salida, comprobando que este tipo de

decodificación es eficiente, además se observa la diferencia de las señales codificadas, generadas por una señal cuadrada y un tren de pulsos.

Para una mejor comprensión el usuario puede variar todos los parámetros de simulación, pero es muy importante que lo realice uno por uno, es decir, que varíe solo un parámetro de un bloque específico y observe los posibles cambios que este puede traer al resultado de la simulación y así poder sacar conclusiones respecto al trabajo realizado.

4.3.5 Actividad Nivel 3

Hasta el momento se ha observado el potencial de *VisSim/Comm*, toda la cantidad de bloques que nos permiten realizar innumerables tareas, en el ámbito de las comunicaciones. *VisSim/Comm* es un elemento imprescindible para modelar diferentes tipos de proceso.

De acuerdo con lo aprendido, en este y en los niveles anteriores el usuario debe estar en la capacidad de desarrollar los siguientes ítems:

1. Simular un canal auditivo gaussiano (*AWGN*), en donde se utilice una señal senoidal de 1Hz y 1V, realizando tres simulaciones diferentes variando los valores S/N compare dichas configuraciones en las ventanas *plot*.

Configurar la ventana *plot* de 5 a 10 segundos máximo dependiendo del nivel S/N.

2. El objetivo de esta práctica consiste en variar progresivamente el porcentaje de error de un canal *BSC* para encontrar el punto en el cual el bloque *Viterbi Hard* no sea capaz de decodificar la señal de entrada correctamente.

Realizar paso a paso las configuraciones que siguen en VisSim/Comm. (VITERBI HARD). Configure un bloque *PN Sequence* y un bloque *Convolutional* con los datos que sigue:

Figura 4.3.22. Configuración *PN Sequence*.

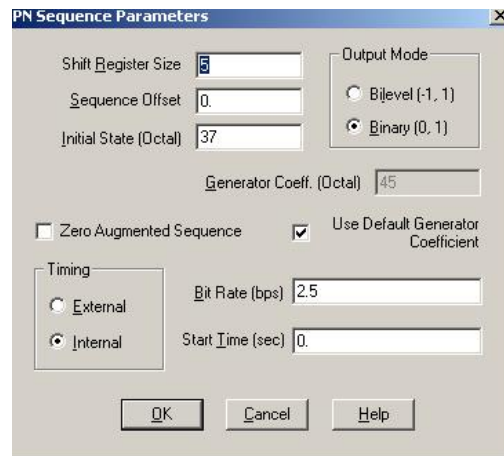
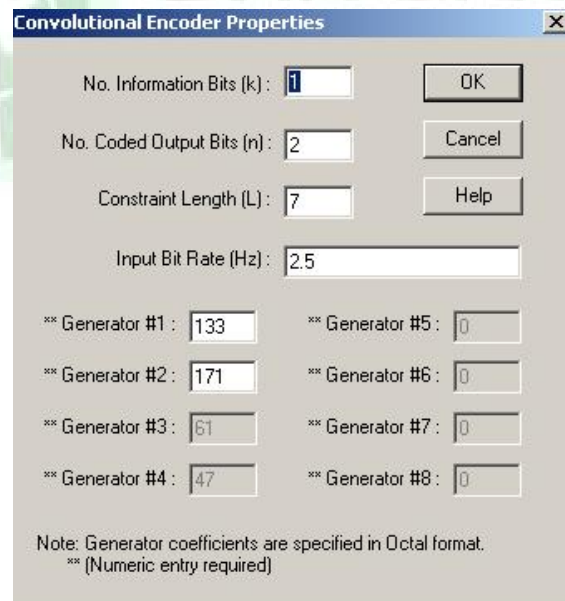


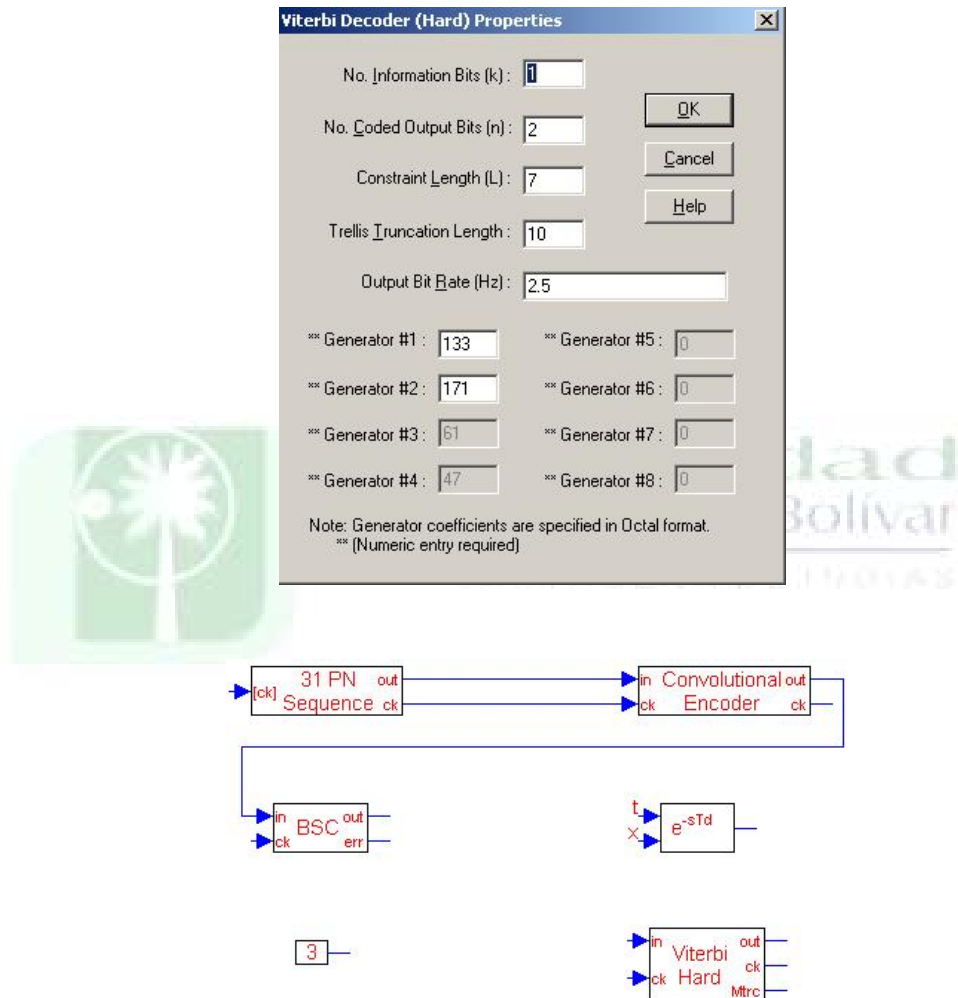
Figura 4.3.23. Configuración *Convolutional Encoder*.



- Configure un bloque *BSC* y un bloque *Viterbi Hard* como sigue:

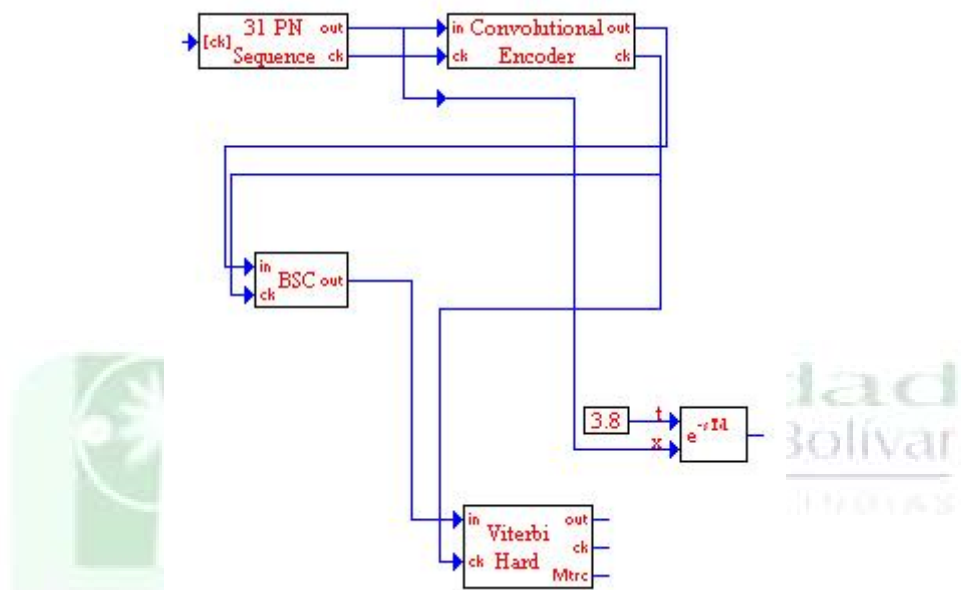
Nota: dejar el bloque BSC con al configuración predeterminada y el tiempo de retardo modificarlo a 3.8 segundos en vez de 3 segundos.

Figura 4.3.24. Configuración *Viterbi Decoder* y bloques restantes



- Finalmente terminar la configuración como sigue y ubicar las salidas necesitas para realizar una buena comparación y encontrar el porcentaje en el cual el bloque *Viterbi Hard* pierde efectividad.

Figura 4.3.25. Configuración *final de bloques*.



BIBLIOGRAFIA

VisSim/CommTM. 1989. versión 5.0. Westford, Massachusetts. Pagina Web de VisSim/Comm disponible en: <http://www.vissim.com/products/comm/index.htm>

Leslei Murray. 2000; 20 de mayo de 2000. Introducción a los Códigos Convolutivos. Escuela de Ingeniería Electrónica, Facultad de Ciencias Exactas, Ingeniería y Agrimensura Universidad Nacional de Rosario. [libro en línea].

Camilo A. Londoño. 31 de agosto de 2004. Tutorial: Introducción a la Modulación Codificada de Trellis. Universidad de los Andes Bogotá, Colombia.

RODEN MARTIN. 1998. Digital Communication System Design, Ed. Prentice Hall.

VISSIM/COMM. 1998. User`s Guide, Versión 3, Sección III del Manual (pág.1-9) Eritek.

VISSIM. 1998. User`s Guide, Version 3, Eritek, Inc, 1998

LABTEL TM. 2002. LABORATORIO DE TEORÍA DE INFORMACIÓN EXPERIENCIA N° 1-N° 7. [libro en línea] <http://www.labtel.cl/ing-civil/>.

STALLINGS, W. COMUNICACIONES Y REDES DE COMPUTADORES.

PRENTICE HALL. 7ª ED. 2004.

FOROUZAN, B. DATA COMMUNICATIONS AND NETWORKING. MC GRAW
HILL. 4THED. 2007.



Universidad
Tecnológica de Bolívar

CARTAGENA DE INDIAS