



**UNIVERSIDAD CATÓLICA DE
SANTIAGO DE GUAYAQUIL**

**TESIS DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO(A) EN TELECOMUNICACIONES CON MENCIÓN EN
GESTIÓN EMPRESARIAL**

TEMA:

**APLICACION DEL SOFTWARE MATLAB/SIMULINK
EN LA ASIGNATURA DE FUNDAMENTOS DE
COMUNICACIÓN DE LA CARRERA DE
INGENIERIA EN TELECOMUNICACIONES**

ALUMNOS:

**Miguel Angel Andrade Moreira
Alejandro Antonio Cortez Bohórquez**

DIRECTOR:

ING. LUZMILA RUILOVA



TESIS DE GRADO

“APLICACION DEL SOFTWARE MATLAB/SIMULINK EN LA ASIGNATURA DE FUNDAMENTOS DE COMUNICACIÓN DE LA CARRERA DE INGENIERIA EN TELECOMUNICACIONES”

Presentada a la Facultad de Educación Técnica para el Desarrollo, Carrera de Ingeniería en Telecomunicaciones de la Universidad Católica de Santiago de Guayaquil

REALIZADO POR:

Miguel Angel Andrade Moreira

Alejandro Antonio Cortez Bohorquez

Para dar cumplimiento con uno de los requisitos para optar por el título de:

Ingeniero en Telecomunicaciones con Mención en Gestión Empresarial

Ing. Luzmila Ruilova

Directora de Tesis

MsC. Edwin Palacios Meléndez

Vocal

Msc. Manuel Romero Paz

Vocal

MsC. Manuel Romero Paz

Decano de la Facultad

MsC. Luis Córdova Rivadeneira

Director de Carrera

CERTIFICACIÓN

Certifico que el proyecto de grado titulado “APLICACION DEL **SOFTWARE MATLAB/SIMULINK** EN LA ASIGNATURA DE FUNDAMENTOS DE COMUNICACIÓN DE LA CARRERA DE INGENIERIA EN TELECOMUNICACIONES” elaborado por Miguel Angel Andrade Moreira y Alejandro Antonio Cortez Bohórquez fue realizado, corregido y terminado, razón por la cual está apto para su presentación y sustentación.

Ing. Luzmila Ruilova

DIRECTOR DE TESIS

AGRADECIMIENTO

Agradezco a Dios ante todo por salud y bienestar que me ha dado. A mis padres principalmente por sus esfuerzos de día a día, su amor y confianza incondicional, el haberme brindado esta oportunidad de crecer bajo sus sabios consejos y bajo sus sombras del buen camino a pesar del inconveniente de la distancia entre ciudades, siempre supieron estar ahí cuando más los necesitaba. A mi familia y a mis amigos q han sido de gran ayuda en todo momento. A Edith por estar ahí en cada momento de mi vida brindándome ayuda, amor y cariño. Todo esto es lo que me ha hecho llegar a lograr mi último paso para convertirme en un Ingeniero de la Universidad Católica de Guayaquil.

A las autoridades, al personal administrativo de la católica, a los profesores quienes fueron nuestra fuente de sabiduría, gracias a todos ellos pudimos realizar y culminar nuestro proyecto.

Miguel Andrade

AGRADECIMIENTO

En primer lugar quiero agradecer a Dios por bendecirme hasta donde he llegado sobre todo por la salud y bienestar que me ha brindado. A mis padres por saber guiarme por el camino del bien e inculcarme buenos valores para ser una persona de bien tanto en lo profesional como lo humano, por su amor y confianza incondicional, por sus consejos tan sabios para no rendirme jamás por los obstáculos que se presentan en la vida. Gracias a todo esto he podido culminar unos de mis grandes éxitos para seguir avanzando en lo profesional y ser un excelente profesional que la Universidad Católica de Santiago de Guayaquil “Facultad Técnica para el Desarrollo”, brinda al país como Ingeniero en Telecomunicaciones con Gestión en Mención Empresarial. También a mi compañero de tesis por brindarme su apoyo y compartir está excelente experiencia juntos.

A las autoridades, al personal administrativo de la católica, a los profesores quienes fueron nuestra fuente de sabiduría, gracias a todos ellos pudimos realizar y culminar nuestro proyecto.

Alejandro Cortez

DEDICATORIA

Dedico mi tesis a mis padres quienes fueron los que estuvieron ahí siempre en cada momento brindándome su amor, cariño, ayuda para la culminación de esta etapa de mi vida. A mi familia la cual siempre ha sido un brazo de gran ayuda. A mis leales amigos que se los cuentan con una sola mano. A Edith L. por ser una persona muy especial en mi vida por todo su amor sincero y confianza que a deposito en mí.

A mi tutor por toda la ayuda que me brindo para realizar con éxito este proyecto y la enseñanza que vamos a dejar para los futuros estudiantes de Ingeniera en Telecomunicaciones.

Miguel Andrade.

DEDICATORIA

Mi tesis se la dedico a Dios Todopoderoso por darme la oportunidad de vivir y darme las fuerzas para luchar día a día.

Con mucho cariño principalmente a mis padres quienes fueron los que me inculcaron los excelentes valores de personalidad, humildad, sabiduría, confianza, amor y fe para llegar hacer un Ingeniero en telecomunicaciones. A mi hermano Yordano porque siempre me brindaba su apoyo y a una persona muy especial mi abuelito Celso Bohórquez que desde el cielo me quería ver cumplir una de mis tantas metas. A mi leal amigo Fernando que también estuvo allí para brindarme su apoyo y sus consejos.

A mi tutor por toda la ayuda que me brindo para realizar con éxito este proyecto y la enseñanza que vamos a dejar para los futuros estudiantes de Ingeniera en Telecomunicaciones.

Alejandro Cortez

RESUMEN

El proyecto del cual se va a analizar a continuación trata acerca de los principales aspectos relacionados con el análisis de la cátedra de Fundamentos de Comunicaciones que se imparte en la carrera de Ingeniería en Telecomunicaciones de la Facultad Técnica para el Desarrollo de la Universidad Católica de Santiago de Guayaquil, para permitir el estudio e introducción de un software en la materia que se mencionó anteriormente y de tal manera esta se convierta en una herramienta práctica fundamental para el aprendizaje y afianzamiento del estudiante.

Este análisis se realiza por motivo que en la actualidad la materia funcionalmente se dicta basada únicamente en teoría aislada de la parte práctica que es donde el estudiante absorbe mucho más conocimientos, de tal manera es necesario para la consolidación de los conocimientos, que dentro de las clases teóricas el docente pueda contar con una herramienta que le permita realizar pruebas o demostraciones prácticas hacia el estudiante mediante un software que permita realizar simulaciones de como son los fundamentos de comunicación y que sea amigable para el uso del alumnado.

ÍNDICE

CARÁTULA	1
CERTIFICACIÓN	3
AGRADECIMIENTO	4
DEDICATORIA	6
RESUMEN	8
ÍNDICE GENERAL	9
ÍNDICE DE DE FIGURAS	14
CAPÍTULO 1 GENERALIDADES	16
1.1 INTRODUCCIÓN	16
1.2 ANTECEDENTES	18
1.3 JUSTIFICACIÓN	19
1.4 PLANTEAMIENTO DEL PROBLEMA	20
1.5 HIPÓTESIS	20
1.6 OBJETIVOS	20
1.6.1 OBJETIVO GENERAL	20
1.6.2 OBJETIVOS ESPECÍFICOS	21
1.7 METODOLOGÍA	
CAPITULO 2 SISTEMAS DE COMUNICACIONES ELECTRÓNICAS	22
2.1 DIVISIÓN DE SISTEMAS DE COMUNICACIONES	23
2.1.1 TRANSMISOR	23
2.1.2 RECEPTOR	23
2.1.3 MEDIO DE TRANSMISIÓN	23
2.1.4 INFORMACIÓN	25
2.2 MODULACIÓN	25
2.2.1 MODULACIÓN POR MANIPULACIÓN	26
2.2.2 MODULACIÓN DE AMPLITUD (AM)	26
2.2.3 MODULACIÓN ANGULAR	27

2.2.3.1	Modulación de frecuencia (FM)	27
2.2.3.2	Modulación de fase (PM)	28
2.2.3.3	Modulación de fase PSK	29
2.2.4	MODULACIÓN DE PULSOS (TRANSMISIÓN DIGITAL)	29
2.2.4.1	Modulación por anchura de pulsos PWM	30
2.2.4.2	Modulación por posición de pulsos PPM	30
2.2.4.3	Modulación de pulsos en amplitud PAM	31
2.2.4.4	Modulación de pulsos codificados PCM	32
2.3	¿PORQUE SE MODULA?	32
2.4	DEMODULACIÓN	33
2.5	RUIDO ELÉCTRICO	33
2.5.1	RUIDO CORRELACIONADO	33
2.5.2	RUIDO NO CORRELACIONADO	33
2.5.2.1	Ruido Interno	34
2.5.2.1.1	Ruido térmico	34
2.5.2.1.2	Ruido de disparo	34
2.5.2.1.3	Ruido de tiempo de contacto	35
2.5.2.2	Ruido externo	35
2.5.2.2.1	Ruido atmosférico	35
2.5.2.2.2	Ruido extraterrestre	36
2.5.2.2.3	Ruido hecho por el hombre	36
2.6	TRASLACIÓN DE FRECUENCIA	36
2.7	CANAL	37
2.8	ANCHO DE BANDA Y CAPACIDAD DE INFORMACIÓN	37
2.9	MODO DE TRANSMISIÓN	38
2.9.1	SIMPLEX	38
2.9.2	HALF DÚPLEX	39
2.9.3	FULL DÚPLEX	39

2.9.4 FULL/FULL DÚPLEX

CAPITULO 3	PROGRAMA DE SIMULACIÓN MATLAB / SIMULINK	41
3.1	ORIGEN	42
3.2	INICIACIÓN DE MATLAB	42
3.3	REQUISITOS DEL SISTEMA PARA PODER INSTALAR MATLAB EN CONSOLA WINDOWS	43
3.4	PASOS A SEGUIR PARA LA INSTALACIÓN DE MATLAB VERSIÓN ESTUDIANTIL.	43
3.5	INGRESO AL PROGRAMA	44
	3.5.1 ACERCA DE MATLAB	45
	3.5.2 USO DEL HELP	47
	3.5.3 FULL PRODUCT FAMILY HELP	47
	3.5.4 MATLAB HELP	47
	3.5.5 USING THE DESKTOP	48
	3.5.6 USING THE COMMAND WINDOWS	48
	3.5.7 DEMOS	48
3.6	EL ENTORNO DE TRABAJO DE MATLAB	49
3.6.1	ESCRITORIO DE MATLAB (MATLAB Desktop)	50
3.7	COMMAND WINDOW	52
3.8	LAUNCH PAD	52
3.9	COMMAND HISTORY BROWSER	52
3.10	CURRENT DIRECTORY BROWSER	53
3.11	PATH DE MATLAB: Establecer el camino de búsqueda (Search Path)	53
3.12	WORKSPACE BROWSER	53
3.13	ARRAY EDITOR	54
3.14	EL EDITOR/DEBUGGER	54
3.15	PREFERENCIAS: FORMATOS DE SALIDA Y DE OTRAS OPCIONES DE MATLAB	54
3.16	GUARDAR VARIABLES Y ESTADOS DE UNA SESIÓN:	

COMANDOS SAVE Y LOAD	56
3.17 LÍNEAS DE COMENTARIOS	56
3.18 FUNCIONES DE LIBRERÍAS	56
3.19 FUNCIONES MATEMÁTICAS ELEMENTALES	57
3.20 TIPOS DE DATOS DE MATLAB	57
3.21 PROGRAMACIÓN DE MATLAB	58
3.22 FICHEROS *.M	58
3.23 FICHEROS DE COMANDOS (SCRIPTS)	59
3.24 HELP PARA LAS FUNCIONES DE USUARIO	60
3.25 HELP DE DIRECTORIOS	60
3.26 INTEGRACIÓN NUMÉRICA DE FUNCIONES	60
3.27 GRÁFICOS BIDIMENSIONALES	61
3.28 FUNCIONES GRÁFICAS 2D ELEMENTALES	61
3.29 FUNCIÓN <i>PLOT</i>	62
3.30 FUNCIÓN <i>TITLE</i>	62
3.31 FUNCIÓN <i>XLABEL</i> y <i>YLABEL</i>	63
3.32 ESTILOS DE LÍNEA Y MARCADORES EN LA FUNCIÓN <i>PLOT</i>	64
3.33 CONTROL DE LOS EJES	65
3.34 FUNCIÓN <i>LINE()</i>	65
3.35 INTRODUCCIÓN A SIMULINK	66
3.36 ACCESO Y DESCRIPCIÓN	66
3.37 CONSTRUCCIÓN DE UN MODELO EN SIMULINK	70
3.38 REFERENCIAS DE LIBRERÍAS DE SIMULINK	72
3.38.1 SINKS	72
3.38.1.1 Scope	72
3.38.1.2 To work space	73
3.38.2 SOURCES	73
3.38.2.1 SINE WAVE	73
3.38.2.1.1 Uso del bloque de onda Sinusoidal en modo continúo	74
3.38.2.1.2 Uso del bloque de onda	

	sinusoidal	
	En modo discreto	74
3.38.2.2	Pulse generator	75
3.38.3	OPERACIONES MATEMÁTICAS	76
3.38.3.1	Producto	76
3.38.3.2	Suma	77
3.38.4	DISEÑO DE FILTRO ANALÓGICO	77
3.38.4.1	Design method	78
3.38.4.2	Filter type	78
3.38.4.3	Filter order	78
3.38.5	FM MODULATOR PASSBAND	79
3.38.6	POWER SPECTRAL DENSITY	80
3.38.7	TO WORKSPACE	80
CAPITULO 4	APLICACIÓN DEL SOFTWARE MATLAB	
SIMULINK EN LA ASIGNATURA DE FUNDAMENTOS DE		
COMUNICACIÓN.		82
4.1	PROGRAMA ACTUAL DE LA ASIGNATURA	85
4.2	SYLLABUS DE LA ASIGNATURA	83
4.3	PRÁCTICAS RECOMENDADAS	87
4.3.1	PRÁCTICA # 1: GENERACIÓN DE UNA SEÑAL SINUSOIDAL.	87
4.3.2	PRÁCTICA # 2: MODULACIÓN DE AMPLITUD (AM) EN SIMULINK.	90
4.3.3	PRÁCTICA # 3: SEÑAL AM CON MODULACIÓN DE TONO.	93
4.4	PRACTICAS ADHERIDAS AL SYLLABUS DE LA ASIGNATURA.	95
CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES.		
5.1	CONCLUSIONES	98
5.2	RECOMENDACIONES	99
5.3	GLOSARIO	

5.4 BIBLIOGRAFÍA

ÍNDICE DE FIGURAS

CAPÍTULO 2: SISTEMAS DE COMUNICACIONES ELECTRÓNICAS

Figura 2.1. Modelo de un medio de transmisión	24
Figura 2.2. Gráfico de modulación de amplitud (AM)	27
Figura 2.3. Gráfico de modulación de frecuencia (FM)	28
Figura 2.4. Gráfico de modulación de fase PSK	29
Figura 2.5. Gráfico de modulación PWM	30
Figura 2.6. Gráfico de modulación PPM	31
Figura 2.7. Gráfico de modulación PAM	31
Figura 2.8. Gráfico de modulaciones PAM, PDM, PCM	32
Figura 2.9. Gráfico de modelo de Tx SIMPLEX	38
Figura 2.10. Gráfico de modelo de Tx HALF DUPLEX	39
Figura 2.11. Gráfico de modelo de Tx FULL DUPLEX	39

CAPÍTULO 3: PROGRAMA DE SIMULACION MATLAB SIMULINK

Figura 3.1. Modelo de Ventana inicial de MATLAB	44
Figura 3.2. Gráfico de la Función Seno(X)	46
Figura 3.3. Gráfico de ubicación del HELP	47
Figura 3.4. Ventana inicial de help windows.	48
Figura 3.5. Ventana inicial de Help Desk.	48
Figura 3.6. Menú para configurar el MATLAB Desktop.	51
Figura 3.7. Configuración por defecto del MATLAB Desktop.	51
Figura 3.8. Cuadro de dialogo PREFERENCES/GENERAL.	55
Figura 3.9. Cuadro de dialogo PREFERENCES/FONTS&COLOR.	55
Figura 3.10. Grafico aplicando función TITLE.	63
Figura 3.11. Grafico aplicando función XLABEL Y YLABEL.	63
Figura 3.12. Grafico de icono para acceso a SIMULINK.	66
Figura 3.13. Ventana de librería del entorno SIMULINK	66
Figura 3.14. Apertura de un nuevo fichero de modelo	67

Figura 3.15. Diferentes toolboxes del Simulink. toolbox básica.	68
Figura 3.16. Diferentes toolboxes del simulink. toolbox básico.	69
Figura 3.17. Diferentes toolboxes del simulink. toolbox básica.	69
Figura 3.18. Ejemplo de generación de una señal sinusoidal.	70
Figura 3.19. Parámetros o configuración del Sine Wave (librería).	71
Figura 3.20. Muestra grafica de un Scope de SIMULINK.	72
Figura 3.21. Grafico del sine wave.	73
Figura 3.22. Grafico del generador de pulso.	75
Figura 3.23. Grafico del generador de pulso.	76
Figura 3.24. Grafico del PRODUCTO.	77
Figura 3.25. Grafico del SUMA.	77
Figura 3.26. Grafico del diseño de filtro analógico.	78
Figura 3.27. Grafico de los parámetros del filtro analógico.	79
Figura 3.28. Grafico del modulador pasa banda	79
Figura 3.29. Grafico de los parámetros del filtro pasa banda	80
Figura 3.30. Grafico del power spectral density	80
Figura 3.31. Grafico del to Workspace	81
Figura 3.32. Grafico de los parámetros del to workspace.	81

CAPITULO 4: APLICACIÓN DEL SOFTWARE MATLAB SIMULINK EN LA ASIGNATURA DE FUNDAMENTOS DE COMUNICACIÓN.

Figura 4.1: Señal Sinusoidal	88
Figura 4.2: Señal Sinusoidal	89
Figura 4.3: Visualización de señal sinusoidal	90
Figura 4.4: Modulación AM	91
Figura 4.5: Visualización de señal modulada por amplitud	92
Figura 4.6 Tono modulador	93
Figura 4.7: Grafico de la señal AM con modulación de tono	94
Figura 4.8: Grafico de la señal de DBLPS con un tono	97

ÍNDICE DE FIGURAS

CAPÍTULO 3: PROGRAMA DE SIMULACION MATLAB SIMULINK

Tabla 3.1: Estilos de Línea y Marcadores en la Función Plot. 60

CAPITULO 4: APLICACIÓN DEL SOFTWARE MATLAB SIMULINK EN LA ASIGNATURA DE FUNDAMENTOS DE COMUNICACIÓN.

Tabla 4.1 : Syllabus actual de la asignatura Fundamentos de comunicación 79

CAPÍTULO 1: GENERALIDADES

En este primer capítulo trataremos los índices del porque la realización de este trabajo en el cual presentamos antecedentes del porque se da el problema, lo planteamos sacamos su hipótesis para tener base sobre la solución del problema, establecemos objetivos los cuales son metas para que el proyecto tenga fiabilidad.

1.1. INTRODUCCIÓN

A mediados del siglo XIX la teoría sobre las comunicaciones electrónicas comenzó, gracias al físico inglés Jaime Cler Maxwell. Las investigaciones matemáticas de este científico indicaron que la luz y la electricidad viajan en forma de una onda electromagnética, y por tanto, las dos están relacionada una con la otra. Maxwell con los estudios de su investigación llegaría a la conclusión de que era posible que se propaguen ondas electromagnéticas por el espacio libre utilizando descargas eléctricas. Sin embargo la propagación de ondas no fue lograda sino hasta el año 1888, donde Heinrich Hertz pudo radiar energía electromagnética desde un equipo que se lo conoce como oscilador.

En trayecto a esta investigación el científico Maxwell no fue el único involucrado, a lo largo del tiempo fueron apareciendo nuevos científicos los cuales permitieron avanzar hacia las comunicaciones, y como el indicio del primer sistema de comunicación en ese entonces fue la clave Morse, que a través de la inducción electromagnética pudo transmitir información por medio de puntos, guiones y espacios por un medio de conducción metálico. Al pasar de los años transmitieron exitosamente ya una conversación humana a través de un sistema telefónico funcional por medio de conductores metálicos como medio de transmisión.

A medida que los años avanzaban esta tecnología, fueron apareciendo lo que son las comunicaciones inalámbricas, en la cual aparecieron los tubos de vacío, el mismo que permitió la primera amplificación práctica de las señales electrónicas, la emisión regular de radio comenzó con las estaciones de radio AM, y de ahí vino la invención de la frecuencia modulada (FM).

Pues a todo esto la electrónica ha revolucionado sorprendentemente en las vidas de las personas, por tal motivo el presente trabajo de investigación trata de introducir principalmente a docentes y alumnos de la asignatura de Fundamentos de Comunicaciones en la Carrera de Ingeniería en Telecomunicaciones con la finalidad de motivarlos hacia el desarrollo de prácticas mediante simulación a través de un software, con el fin de que el estudiante no sea tan teórico, sino que pueda aplicar la teoría mediante un modelo de simulación que pueda generar señales típicas en las Telecomunicaciones.

El programa **MATLAB** (“*Matrix Laboratory*”) es un software que permite realizar cálculos numéricos con vectores y matrices, como caso particular puede también trabajar con números escalares (tanto reales como complejos), con cadenas de caracteres y con otras estructuras de información más complejas. Una de las capacidades más atractivas es la de realizar una amplia variedad de gráficos en dos y tres dimensiones. **MATLAB** tiene también un lenguaje de programación propio. En nuestro caso nos va a servir para el entendimiento y estudio de las señales típicas de las telecomunicaciones.

Simulink es una aplicación que permite construir y simular modelos de sistemas físicos y sistemas de control mediante diagramas de bloques. El comportamiento de dichos sistemas se define mediante funciones de transferencia, operaciones matemáticas, elementos de **Matlab** y señales predefinidas de todo tipo. **Simulink** dispone de una serie de utilidades que facilitan la visualización, análisis y guardado de los resultados de simulación.

Con **MATLAB / SIMULINK** se pretende afianzar la teoría con la práctica, introduciendo en el programa de estudios de la materia, Fundamentos de Comunicaciones. Las simulaciones que se aplicarán serán ejercicios teóricos resueltos en clases, en este caso vendrían a ser ejercicios teóricos que se introducirán en el software, para así visualizar el resultado obtenido a través de un esquema gráfico de la señal al momento de la transmisión o de los problemas que se pueden generar en dicha transmisión.

1.2. ANTECEDENTES

MATLAB es un gran programa de cálculo técnico y científico para ciertas operaciones es muy rápido, cuando puede ejecutar sus funciones en código nativo con los tamaños más adecuados para aprovechar sus capacidades de vectorización. En otras aplicaciones resulta bastante más lento que el código equivalente desarrollado en C/C++ o Fortran. Sin embargo, siempre es una magnífica herramienta de alto nivel para desarrollar aplicaciones técnicas, fácil de utilizar y que, como ya se ha dicho, aumenta significativamente la productividad de los programadores respecto a otros entornos de desarrollo. **MATLAB** dispone de un código básico y de varias librerías especializadas (**toolboxes**). En estos apuntes se hará referencia exclusiva al código básico.

MATLAB se puede arrancar como cualquier otra aplicación de **Windows**, dando click dos veces en el icono correspondiente en el escritorio o por medio del menú (Inicio). Es un **software** bastante conocido en Universidades e institutos para el aprendizaje en cursos básicos y avanzados de matemáticas, ciencias y especialmente en ingeniería. En el área de la industria este **software** se utiliza en lo que es investigaciones, desarrollo y diseño de prototipos.

MATLAB siendo un software bastante extenso, es imposible tratar todos los temas aquí, por tanto solo nos enfocaremos en señales típicas en el ámbito de las Telecomunicaciones, por tanto nos centraremos en los aspectos básicos de **MATLAB**. En consecuencia, el estudiante o docente podrán enfrentarse a otros aspectos más avanzados utilizando el menú ayuda (**help**) del **software**.

El entorno de trabajo de **MATLAB** ha mejorado mucho a partir de la versión 6.0, asiéndose mucho más gráfico e intuitivo, similar al de otras aplicaciones profesionales de **Windows**. Todos estos componentes son los que caracterizan a **MATLAB** como una excelente herramienta y en nuestra carrera Ingeniera en Telecomunicaciones en la materia de Fundamentos de comunicaciones es donde este **software** va a ser un excelente complemento a las clases teóricas y teniendo presente que las prácticas son de gran importancia para el desarrollo profesional del estudiante.

1.3. JUSTIFICACIÓN

En la actualidad, la mayoría de las universidades utilizan metodologías aplicativas con visiones y objetivos de que el alumnado pueda entender la teórica y llevarlo a la forma práctica en materias que se están dictando, con la finalidad de alcanzar la excelencia académica, disponiendo de herramientas actualizadas para incluirla en el programa de estudios dentro de la malla curricular de la carrera de Ingeniería en Telecomunicaciones.

Este tipo de metodologías lleva a enfocarnos a la investigación, planificación y elaboración de la tesis, puesto que con esto mejoramos el estudio de la materia, tomando la misma planificación del pensum, haciéndolo más aplicativo dicha herramienta, con la finalidad de obtener un mejor aprendizaje de la materia. La justificación de la tesis plantea reformar el desarrollo y entendimiento de la materia de Fundamentos de Comunicaciones, basada en la introducción de un **software** profesional amigable con el estudiante con herramientas nada complejas en su empleo, como lo es **MATLAB**, el cual reúne cualidades para poder simular la teoría a través de señales típicas en sistemas de telecomunicaciones.

El punto de vista metodológico de esta tesis, estará basado en el uso del método deductivo, por motivos que este método permite la integración de la parte teórica de la materia con la aplicativa; lo cual proporcionara un juicio técnico respecto a las diversas aplicaciones de las señales típicas de las Telecomunicaciones.

Esta investigación logrará abrir nuevas investigaciones en el campo de las comunicaciones eléctricas; con este sistema los alumnos podrán tener la capacidad de dar soluciones a los inconvenientes que se pueden presentar a diario, para lograr experticia en el ámbito profesional de los estudiantes como futuros ingenieros, al haber tenido una instrucción académica en hechos reales y observables.

1.4. PLANTEAMIENTO DEL PROBLEMA

En la asignatura de Fundamentos de Comunicaciones de Sexto Ciclo de la Carrera de Ingeniería en Telecomunicaciones surge la necesidad de adherir la parte práctica para poder tener mejor comprensión de la parte teórica empleando herramientas tecnológicas disponibles en el ámbito universitario tanto nacional como internacional.

1.5. HIPÓTESIS

La aplicación de la herramienta de simulación **MATLAB** como ayuda didáctica de la asignatura Fundamentos de Comunicación permitirá mejorar en proceso de aprendizaje en los alumnos y de enseñanza de los docentes; para así comprobar o corroborar la teoría.

1.6. OBJETIVOS

A continuación se va a detallar el objetivo general del proyecto y los objetivos específicos que se tienen planteados:

1.6.1. OBJETIVO GENERAL

Evaluar la aplicabilidad de la herramienta de programación **MATLAB/SIMULINK** en la asignatura de Fundamentos de Comunicación del VI ciclo de la Carrera de Ingeniería en Telecomunicaciones para que el estudiante fortalezca de mejor manera la parte práctica previo aprendizaje de la teoría de la mencionada asignatura.

1.6.2. OBJETIVOS ESPECÍFICOS

- Determinar el estado del arte en cuanto a los sistemas de comunicaciones electrónicas.
- Diseñar aplicaciones de comunicaciones e incluirlos en el programa de estudio de Fundamentos de Comunicaciones.
- Proponer un manual de prácticas para simulación de las aplicaciones en comunicaciones.

1.7. METODOLOGIA

MÉTODOS:

- Método Inductivo-Deductivo
- Método Activo
- Método Comparativo
- Método lógico
- Método Socrático
- Método Analítico

TÉCNICAS:

- Lluvia de ideas
- Exposiciones Grupales e individuales
- Trabajo en equipo.
- Mesas redonda
- Simposio
- Acróstico
- Charlas virtuales

CAPÍTULO 2: SISTEMAS DE COMUNICACIONES ELECTRÓNICAS

La idea principal para las comunicaciones electrónicas, es la transmisión de paquetes de un lugar a otro, con referencia a esto se puede concluir que los sistemas de comunicaciones electrónicos son la transmisión, la recepción y procesamiento de información entre dos o más lugares mediante los circuitos electrónicos.

Existen dos tipos básicos de sistemas de comunicaciones electrónicas: analógico y digital. Un sistema de comunicaciones analógico es un sistema en el cual la energía electromagnética se transmite y recibe de forma en forma analógica (una señal variando continuamente tal como una onda senoidal). Los sistemas de radios comerciales emiten señales analógicas. Un sistema de comunicaciones digital es un sistema en el cual la energía electromagnética se transmite y recibe en forma digital (niveles discretos tal como +5 V y tierra).

Los sistemas binarios utilizan señales digitales que sólo tienen dos niveles discretos. Frecuentemente la información de la fuente original está en una forma que no es adecuada para la transmisión y debe convertirse en una forma más adecuada antes de la transmisión. Por ejemplo, con los sistemas de comunicaciones digitales, la información analógica se convierte en una forma digital antes de la transmisión, y con los sistemas de comunicaciones analógicas, la información digital se convierte a la forma analógica antes de la transmisión.

El origen de la información puede estar de forma analógica como por ejemplo la música o la voz humana, o también puede estar de forma digital discreta como por ejemplo los números codificados binariamente o los códigos alfanuméricos, sin embargo todas las formas de información se deben convertir en energía electromagnética antes de ser propagadas a través de un sistema electrónico de comunicaciones.

2.1. DIVISIÓN DE SISTEMAS DE COMUNICACIONES

Los sistemas de comunicaciones electrónicos se dividen en cuatro puntos característicos:

1. Transmisor
2. Receptor
3. Medio de Transmisión
4. Información

2.1.1. TRANSMISOR

Éste se encarga de convertir la información original de la fuente a una forma más adecuada para la transmisión. El transmisor es un equipo que emite una señal a través de un medio. El transmisor transforma y codifica la información produciendo señales electromagnéticas susceptibles de ser transmitidas a través de algún sistema de transmisión. Por ejemplo, un modem convierte las cadenas de bits generadas por un computador personal y los transforma en señales analógicas que pueden ser transmitidas a través de la red telefónica.

2.1.2. RECEPTOR

El receptor es aquel que convierte o decodifica la información recibida de un medio de transmisión a su forma original y la transfiere a su destino. Por ejemplo, un modem aceptará la señal analógica de la red o línea de transmisión y la convertiría en una cadena de bits.

2.1.3. MEDIO DE TRANSMISIÓN

El medio de transmisión constituye el canal que permite la transmisión de información entre dos terminales en un sistema de comunicación. Es un medio de conexión entre transmisor y receptor, las transmisiones habitualmente se realizan empleando ondas electromagnéticas que se propagan a través del canal. El medio de transmisión es el camino físico entre el transmisor y el receptor. Los medios de transmisión se clasifican en guiados y no guiados. En ambos casos, la comunicación se lleva a cabo con ondas electromagnéticas.

En los medios guiados las ondas se confinan en un medio sólido, como por ejemplo: un par trenzado, un cable coaxial o una fibra óptica. La atmósfera o el espacio exterior son ejemplos de medios no guiados, que proporcionan un medio de transmitir las señales pero sin confinarlas; este tipo de transmisión se denomina inalámbrica.

Las características (ver figura 2.1) y calidad de transmisión están determinadas tanto por el tipo de señal, como por las características del medio. En el caso de los medios guiados, el medio en sí mismo es lo más importante en la determinación de las limitaciones de transmisión.



Figura 2.1: Modelo de un medio de transmisión

Fuente: <http://www.monografias.com/trabajos17/medios-de-transmision/medios-de-transmision.shtml>

En medios no guiados, el ancho de banda de la señal emitida por la antena es más importante que el propio medio a la hora de determinar las características de la transmisión. Una propiedad fundamental de las señales transmitidas mediante antenas es la "directivita". En general, a frecuencias bajas las señales son omnidireccionales; es decir, la señal desde la antena se emite y propaga en todas direcciones. A frecuencia más alta, es posible concentrar la señal en un haz direccional.

En el diseño de los sistemas de transmisión es deseable que tanto la distancia como la velocidad de transmisión sean lo más grandes posibles. Hay una serie de factores relacionados con el medio de transmisión y con la señal que determinan tanto la distancia como la velocidad de transmisión, algunos de ellos son:

- El ancho de banda.
- Dificultades en la transmisión.

- Interferencias.
- Número de receptores.

2.1.4. INFORMACIÓN

La información es un conjunto organizado de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sistema que recibe dicho mensaje. Es el conocimiento, la sabiduría o la realidad. La información puede ser:

Analógica: Proporcional o continúa.

Digital: Etapas discretas.

Ondas: Electromagnéticas para propagarse.

2.2. MODULACIÓN

Es el proceso de cambiar alguna característica de la portadora de acuerdo a la información original de la fuente. Estas técnicas permiten el aprovechamiento del canal de comunicación permitiendo así mayor transferencia de información en forma simultánea además de mejoramiento de la resistencia contra posibles ruidos e interferencias.

1. Modulación por manipulación (telégrafo).
2. Modulación de amplitud (AM).
3. Modulación angular.
 - Modulación de radio frecuencia (FM).
 - Modulación de Fase (PM).
4. Modulación de pulsos (TRANSMISIÓN DIGITAL).
 - De amplitud (PAM).
 - De anchura (PWM).
 - De posición (PPM).
 - De codificación de pulsos (PCM).

2.2.1. MODULACIÓN POR MANIPULACIÓN

Básicamente este fue el uso de lo que fue el sistema de clave Morse que utilizaba un hilo de cobre para su transmisión y trabajaba por medio de puntos, rayas y espacios entre elementos, espacios entre letras y espacios entre palabras.

2.2.2. MODULACIÓN DE AMPLITUD (AM).

Es el proceso de cambiar la amplitud de una portadora de frecuencia relativamente alta (RF) de acuerdo con la amplitud de la señal modulante. En AM la amplitud se imprime sobre la portadora en forma de cambios de amplitud (ver en figura 2.2).

La modulación de amplitud es una forma de modulación relativamente barata y de baja calidad de modulación que se utiliza en la radiodifusión de señales de audio y video. La modulación de amplitud también se usa para las comunicaciones de radio móvil de dos sentidos tal como una radio de banda civil (CB) (26.965 a 27.405 MHz).

Un modulador AM es un aparato no lineal con dos señales de entrada de información: una señal portadora de amplitud constante y de frecuencia sencilla, y la señal de información. La información actúa sobre o modula la portadora y puede ser una forma de onda de frecuencia simple o compleja compuesta de muchas frecuencias que fueron originadas de una o más fuentes. Debido a que la información actúa sobre la portadora, se llama señal modulante. La resultante se llama onda modulada o señal modulada.

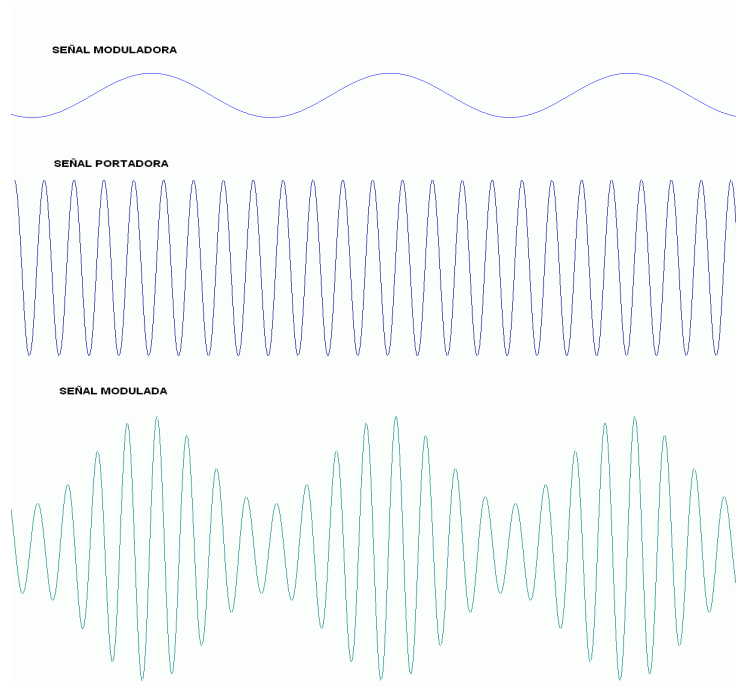


Figura 2.2. Gráfico de modulación de amplitud (AM)

Fuente: <http://upload.wikimedia.org/wikipedia/commons/f/f1/Modulacionenamplitud.gif>

2.2.3. MODULACIÓN ANGULAR

A continuación mencionamos los tipos de modulación angular:

- Modulación de frecuencia
- Modulación de fase (PM)
- Modulación de fase (PSK)

2.2.3.1. MODULACIÓN DE FRECUENCIA (FM).

En esta forma de modulación es la frecuencia la característica de la portadora que se va a alterar para insertar información. La modulación de frecuencia es una forma de la modulación angular. La modulación angular resulta cuando el ángulo de fase de una onda sinusoidal, varía con respecto al tiempo. La frecuencia modulada FM es usada comúnmente en las radiofrecuencias de muy altas frecuencia por la alta fidelidad de la radiodifusión de la música y el habla. El sonido de la televisión analógica también es difundido por medio de FM.

La modulación de frecuencia (FM), varía la frecuencia de la portadora de amplitud (Ver figura 2.3) constante directamente proporcional, a la amplitud de la señal modulante, con una relación igual a la frecuencia de la señal modulante.

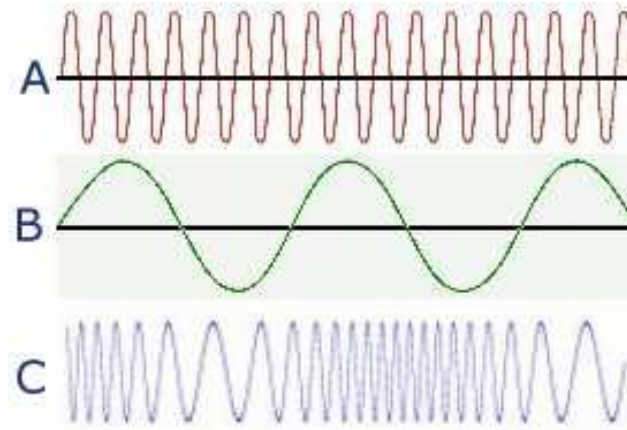


Figura 2.3. Gráfico de modulación de frecuencia (FM)
Fuente: <http://www.natureduca.com/radioblog/?p=96>

2.2.3.2. MODULACIÓN DE FASE (PM).

Este también es un caso de modulación donde las señales de transmisión como las señales de datos son analógicas y es de tipo exponencial al igual que la FM. En este caso el parámetro de la señal portadora que variará de acuerdo a señal moduladora es la fase. La modulación de fase (PM) no es muy utilizada principalmente por que se requiere de equipos de recepción más complejos que en FM.

La modulación de fase (pm) se caracteriza por que la fase de la onda portadora varía directamente con la señal modulante resultando una señal de modulación en fase.

2.2.3.3. MODULACIÓN DE FASE PSK.

Una de las principales características de la técnica de modulación PSK es que la fase de la señal portadora es la representante de cada símbolo de información de la señal moduladora, el cual el modulador elije un valor angular entre un conjunto discreto de n valores posibles. (Ver figura 2.4)

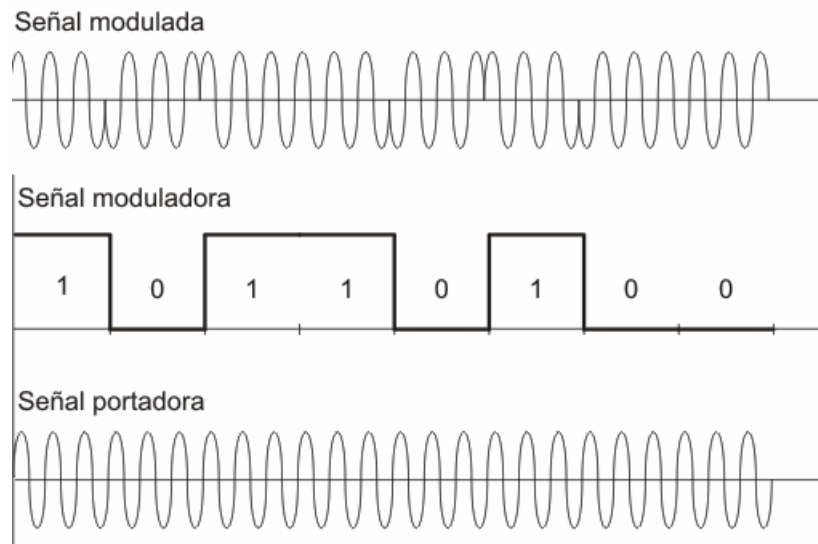


Figura 2.4: Gráfico de modulación de fase PSK

Fuente: <http://upload.wikimedia.org/wikipedia/commons/8/83/Modulacion-PSK.gif>

2.2.4. MODULACIÓN DE PULSOS (TRANSMISIÓN DIGITAL).

La modulación de pulsos incluye muchos métodos diferentes para convertir información a forma de pulsos para transferirlos de una fuente a un destino. Los cuatro métodos predominantes se describen a continuación:

2.2.4.1. MODULACIÓN POR ANCHURA DE PULSOS PWM.

Este método a veces se llama modulación de duración del pulso (PDM) o modulación de longitud del pulso (PLM). El ancho del pulso (porción activa del ciclo de trabajo) es proporcional a la amplitud de la señal analógica. En este caso para variar la anchura o duración de los pulsos las muestras de la señal tienen que ser empleadas (Ver figura 2.5). Aunque este método ya no es muy común, ahora en día se emplea en transmisores modulados en amplitud, en que la modulación se realiza primero en esta forma. Con este método se vuelve más eficiente el transmisor.

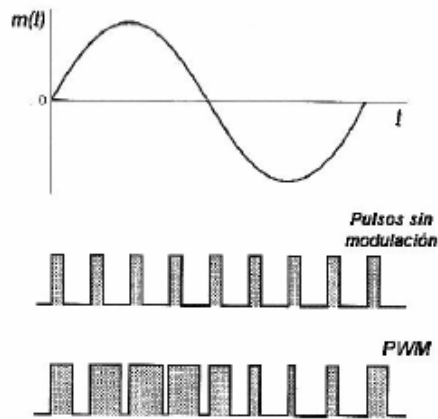


Figura 2.5. Gráfico de modulación PWM

Fuente: Print Screen de manual de fundamentos de comunicaciones

2.2.4.2. MODULACIÓN POR POSICIÓN DE PULSOS PPM.

La posición de un pulso de ancho constante, dentro de una ranura de tiempo prescrita, varía de acuerdo a la amplitud de la señal analógica. Aquí en este caso se produce un desplazamiento de los pulsos por medio de la señal moduladora respecto a la posición de estos en ausencia de modulación (Ver figura 2.6).

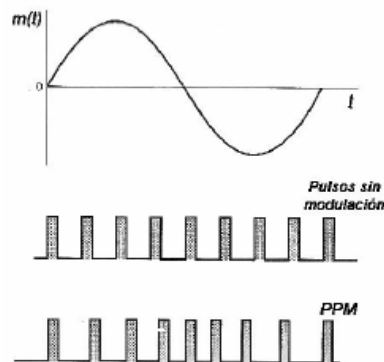


Figura 2.6. Gráfico de modulación PPM

Fuente: **Print Screen** de manual de fundamentos de comunicaciones

2.2.4.3. MODULACIÓN DE PULSOS EN AMPLITUD PAM.

La amplitud de un pulso de longitud constante y de ancho constante varía de acuerdo a la amplitud de la señal analógica. Este tipo de modulación es el resultado previo del muestreo de una señal analógica (Ver figura 2.7).

Si una señal analógica ya podría ser de voz se muestra a intervalos regulares en tener una serie de valores continuos, se tendrán valores discretos a intervalos específicos, determinados por la que debe ser como mínimo del doble de la frecuencia máxima de la señal muestreada. Esto se ilustra esquemáticamente en la (figura 2.7).

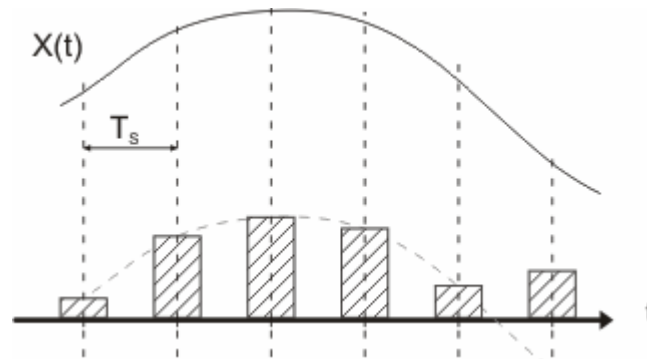


Figura 2.7. Gráfico de modulación PAM

Fuente: <http://www.textoscientificos.com/redes/modulacion/pulsos-analogica>

2.2.4.4. MODULACIÓN DE PULSOS CODIFICADOS PCM.

La señal analógica se muestrea y se convierte a una longitud fija, un número binario serial para su transmisión. El número binario varía de acuerdo a la amplitud de la señal analógica. Siendo esta la modulación de pulsos más utilizada de todas (Ver figura 2.8). PCM siempre lleva en sí modulación previa de amplitud de pulsos. Una de las características principales de una señal analógica es que su amplitud puede tomar cualquier valor entre un mínimo y un máximo, de forma continua. Una señal analógica se puede convertir en digital mediante un proceso de muestreo y codificación. El muestreo la convierte en una señal PAM, la cuantificación redondea el valor de la amplitud al número permisible más cercano, generalmente en el intervalo $(0, 2^n)$ y lo codifica en un cierto número de bits.

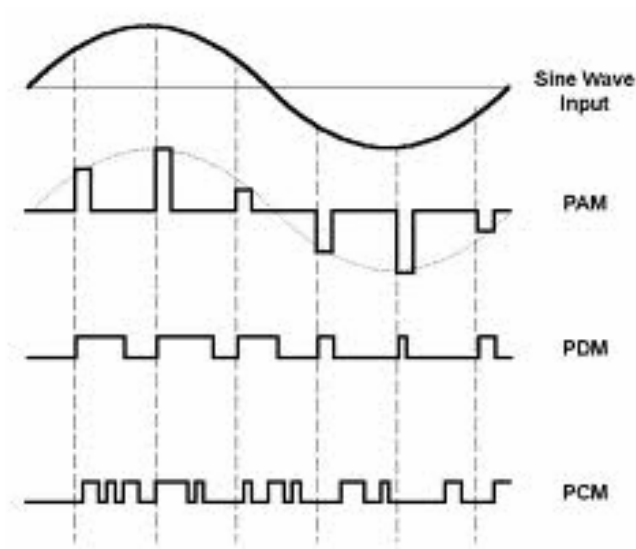


Figura 2.8. Gráfico de modulaciones PAM, PDM, PCM
Fuente: Print Screen de manual de fundamentos de comunicación

2.3. ¿PORQUE SE MODULA?

Es una pregunta básica al analizar los métodos de transmisión de un sistema de comunicaciones, a continuación se presentan algunas de las razones que demuestran que la modulación es un proceso imprescindible al momento de transmitir información.

1. Porque facilita la propagación de la señal de información por cable o por el aire.
2. Ordena el espectro radioeléctrico, disminuyendo canales a cada fuente de información.
3. Disminuye las dimensiones de las antenas.
4. Optimiza el ancho de banda de cada canal.
5. Evita interferencia entre canales.
6. Protege a la información de las degradaciones del ruido.
7. Define la calidad de la información transmitida.

2.4. DEMODULACIÓN

Es el proceso de convertir los cambios de la portadora analógica a la información original de la fuente. La demodulación se realiza en el receptor, en

un circuito llamado demodulador. Detector es el nombre que también se le da a demodulación.

2.5. RUIDO ELÉCTRICO

Es cualquier energía eléctrica no deseada presente en la banda de paso útil de un circuito de comunicación. Por ejemplo en una grabación de audio cualquier señal no deseada que cae en la banda de frecuencias, entre 0 y 15 KHz, es perceptible e interferirá con la información de audio. Para los circuitos de audio, cualquier energía eléctrica no deseada en la banda de frecuencias entre 0 y 15 KHz se considera ruido.

El ruido eléctrico se clasifica de la siguiente manera:

2.5.1. Ruido correlacionado.

Es cuando existe una relación entre señal y ruido. Sin señal no hay ruido.

2.5.2 Ruido no correlacionado.

Está presente sin importar si existe una señal o no.

- **Ruido interno.** El cual puede ser:
 - Ruido térmico.
 - Ruido de disparo.
 - Ruido de tiempo de contacto.
- **Ruido externo.** Que se clasifica a su vez en:
 - Ruido atmosférico.
 - Ruido extraterrestre.
 - Ruido hecho por el hombre.

2.5.2.1 RUIDO INTERNO

Es interferencia eléctrica generada dentro de un dispositivo.

2.5.2.1.1 RUIDO TÉRMICO

Está asociado con el movimiento de electrones libres en un conductor causado por la agitación térmica, este depende de la temperatura. Por tanto el ruido

interno es eléctrico y es producido por la energía interna de la materia. El ruido térmico es aleatorio dado que el movimiento de los electrones está agitado por la energía browniana el cual esta energía se encuentra en movimiento aleatorio. El ruido térmico también toma como nombre ruido plano, toma es nombre dado que su respuesta es plana.

2.5.2.1.2 RUIDO DE DISPARO

Es causado por la llegada aleatoria de portadores en el elemento de salida de un dispositivo electrónico (diodos, BJT, FET, etc.). El ruido de disparo es un ruido electromagnético no correlacionado. Este también es conocido como ruido de transistor, producido por la llegada aleatoria de portadores tales como electrones y huecos en el elemento de salida de un dispositivo, tales como diodo, un transistor que puede ser de efecto de campo o bipolar, o un tubo de vacío. El ruido de disparo demuestra que es aditivo respecto al ruido térmico y al mismo.

2.5.2.1.3 RUIDO DE TIEMPO DE CONTACTO

La modificación de una corriente de portadores al pasar de la entrada a la salida de un dispositivo produce una variación aleatoria irregular que es igual al ruido de tránsito. Lo que produce una variación aleatoria irregular de la energía con respuesta plana.

2.5.2.2 RUIDO EXTERNO

Es generado externamente a un circuito y se introduce en él. Las señales externamente generadas se consideran ruido, sólo si sus frecuencias caen dentro de la banda útil del filtro de entrada del circuito. Puede ser:

2.5.2.2.1 RUIDO ATMOSFÉRICO

Es energía eléctrica que ocurre naturalmente y se origina en la atmósfera de la tierra. El ruido atmosférico es comúnmente llamado electricidad estática. El ruido atmosférico provoca energía estática, su fuente son las perturbaciones eléctricas naturales. Se presenta en forma de impulsos. Es inversamente proporcional a la frecuencia. A FM no le afecta el ruido estático, pues este solo afecta a la amplitud. El ruido estático depende de las condiciones del tiempo. El

ruido atmosférico es la suma de la energía eléctrica de todas las fuentes externas, locales y distantes. El ruido atmosférico se propaga por medio de la atmosfera de la tierra por medio de las ondas de radio. El ruido atmosférico es el tronido que se escucha en un receptor de radio predominantemente.

2.5.2.2.2 RUIDO EXTRATERRESTRE

Este se origina fuera de la atmosfera de la tierra y es conocido como ruido de espacio profundo o galáctico. El ruido extraterrestre se origina de la vía láctea, en otras galaxias y el sol. El ruido extraterrestre se divide en dos categorías: solar y cósmico (galáctico). El ruido extraterrestre contiene frecuencias de aproximadamente 8 MHz a 1.5 GHz, aunque las frecuencias menores a 20 MHz raramente penetran la atmósfera de la tierra y por lo tanto son insignificantes.

2.5.2.2.3 RUIDO HECHO POR EL HOMBRE

Es el ruido eléctrico que se puede atribuir al hombre. Sus fuentes pueden ser mecanismos que producen chispas, motores eléctricos, encendido de los autos, equipos de comunicación de potencia, luces fluorescentes, ruido industrial, entre otros. El ruido eléctrico también es impulsivo en su naturaleza y por lo tanto contiene un rango amplio de frecuencias que son propagadas por el espacio de la misma manera que las ondas de radio. El ruido eléctrico es más intenso en las áreas más pobladas, metropolitanas e industriales.

2.6 TRASLACIÓN DE FRECUENCIA

Es el Proceso de convertir una frecuencia o banda de frecuencias de un punto a otro en el espectro electromagnético. La base del multiplexado por división de frecuencia, es el desplazamiento de traslación o conversión de frecuencia. Traslación de frecuencia es un proceso que permite transmitir señales diferentes en diferentes bandas de frecuencias.

La traslación de frecuencia o traslación se emplea también para desplazar una señal modulada a otra nueva frecuencia portadora. Para amplificación o para algún otro procesamiento.

Transmisor (Modulador) AF - > RF

Receptor (Demodulador) RF - > AF

2.7 CANAL

Canal es una conexión entre los puntos de inicio y terminación de un circuito. Canal es un camino único facilitado mediante un medio de transmisión que puede ser:

- Con separación física, tal como un par de cables multiparados.
- Con separación eléctrica, tal como la multiplicación por división de frecuencia (MDF) o por medición de tiempo (MDT)

Cuando se define a una frecuencia o a una banda específica de frecuencias distribuidas para un servicio particular o para transmisión. Canal de voz: 4KHZ

VOZ: 50 HZ - >	15000 HZ
300HZ - >	3400HZ

2.8 ANCHO DE BANDA Y CAPACIDAD DE INFORMACIÓN

Las limitaciones principales de los sistemas de comunicación son dos:

- El ancho de banda.
- El Ruido.

El ancho de banda de un sistema de comunicaciones es la banda de paso mínima (Rango de frecuencias) requeridas para propagar una información de la fuente a través del sistema. El ancho de banda debe ser suficientemente grande (ancho) para pasar todas las frecuencias significativas de la información.

La capacidad de información de un sistema de comunicaciones es una medida de cuánta información de la fuente puede transportarse por el sistema, en un periodo dado de tiempo. La cantidad de información que puede propagarse a

través de un sistema de transmisión es una función del ancho de banda del sistema y el tiempo de transmisión.

2.9 MODO DE TRANSMISIÓN

Los sistemas de comunicaciones electrónicas pueden diseñarse para manejar la transmisión solamente en una dirección, en ambas direcciones pero sólo uno a la vez, o ambas direcciones al mismo tiempo. Estos se llaman modos de transmisión.

Los modos de transmisión son los siguientes:

2.9.1 *SIMPLEX*

La transmisión se realiza en una sola dirección (Ver figura 2.9). Los sistemas simplex son llamados a veces llamados sistemas de un sentido, sólo para recibir o sólo para transmitir. Una ubicación puede ser un transmisor o un receptor pero, no ambos. Un ejemplo de transmisión *simplex* es la radiodifusión de la radio comercial o de televisión; la estación de radio siempre transmite y el usuario siempre recibe.

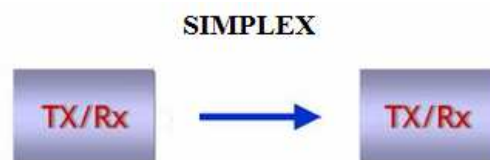


Figura 2.9. Gráfico de modelo de Tx SIMPLEX

Fuente: <http://elnafragodelared.blogspot.com/2011/02/modos-simplex-half-duplex-y-full-duplex.html>

2.9.2 *HALF DÚPLEX*

Se transmite en ambas direcciones pero no simultáneamente (Ver figura 2.10). Algunas veces se les llaman sistemas con alternativa de los sentidos, cualquier sentido, o cambio y fuera. Una ubicación puede ser un transmisor y un receptor, pero no los dos al mismo tiempo. Por ejemplo los sistemas de radio de doble sentido que utilizan los botones oprima para hablar (PTT), para operar sus transmisores, como los radios de banda civil y de banda policiaca.

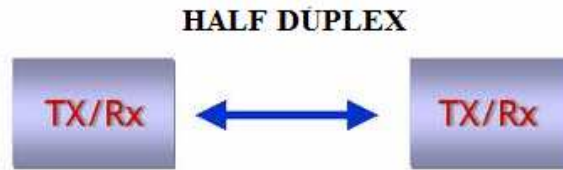


Figura 2.10. Gráfico de modelo de Tx HALF DUPLEX
 Fuente: <http://elnafragodelared.blogspot.com/2011/02/modos-simplex-half-duplex-y-full-duplex.html>

2.9.3 FULL DÚPLEX

Transmisión simultánea en ambas direcciones (Ver figura 2.11). Algunas veces se les llama líneas simultáneas de doble sentido, dúplex o de ambos sentidos. Una ubicación puede transmitir y recibir simultáneamente; sin embargo, la estación a la que se está transmitiendo también debe ser la estación de la cual está recibiendo. Por ejemplo un sistema telefónico estándar.

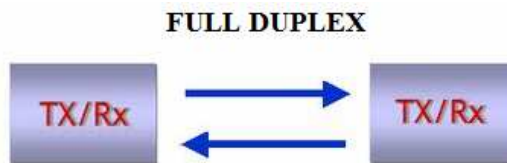


Figura 2.11. Gráfico de modelo de Tx FULL DUPLEX
 Fuente: <http://elnafragodelared.blogspot.com/2011/02/modos-simplex-half-duplex-y-full-duplex.html>

2.9.4 FULL/FULL DÚPLEX

Es posible transmitir y recibir simultáneamente, pero no necesariamente entre las mismas dos ubicaciones (es decir, una estación puede transmitir a una segunda estación y recibir de una tercera estación al mismo tiempo). Se utilizan exclusivamente con circuitos de comunicaciones de datos. Por ejemplo El Servicio Postal de Estados Unidos.

CAPÍTULO 3: PROGRAMA DE SIMULACIÓN *MATLAB* / *SIMULINK*

MATLAB fue creado en el año de 1984 por el matemático Cleve Moler, la primera versión surgió con la idea de emplear paquetes de subrutinas escritas en Fortran en los cursos de algebra lineal y análisis numérico. El lenguaje de programación **Matlab** fue creado en el año 1970 para facilitar un fácil acceso al software de matrices **Linpac** y **Eispac** sin tener que hacer uso de **Fortran**. Se estimaba que en 2004, **MATLAB**, era utilizado por más de un millón de usuarios en áreas académicas y empresariales.

MATLAB es una herramienta de cómputo que trabaja de la mano con cálculos y visualizaciones. Entre sus prestaciones se encuentran el análisis numérico, cálculo matricial, programación, procesamiento de señales y gráficos. Es una herramienta de fácil uso para el usuario, contiene un **HELP** bastante extenso que ayuda a resolver los problemas que se presentan.

El nombre **MATLAB** proviene de **Matrix Laboratory**, es un sistema interactivo cuyo elemento básico de datos es una matriz que no necesita dimensionamiento, ayuda a solucionar problemas en un menor rango de tiempo a diferencia de tradicionales lenguajes de programación como C++, Pascal, entre otros que no cuentan con esta ventaja. **MATLAB** nace como una solución a la necesidad de mejores y más poderosas herramientas para resolver problemas de cálculo complejos en los que es necesario aprovechar las amplias capacidades de proceso de datos de grandes computadores.

Este **software** viene creciendo a través de los años con entradas provenientes de muchos usuarios. A nivel universitario, **MATLAB** se ha convertido en la herramienta de enseñanza estándar para cursos de introducción en álgebra lineal aplicada, así como cursos avanzados en otras áreas como matemáticas, ingeniería y ciencia. A nivel industrial **MATLAB** se lo utiliza para investigaciones y para resolver problemas de ingeniería y matemática.

Proporciona una serie de soluciones específicas llamadas **TOOLBOXES**, que son conjuntos de funciones que extienden el entorno **MATLAB** para resolver problemas particulares como: procesamiento de señales, diseño de sistemas de control, la simulación de sistemas dinámicos, identificación de sistemas, redes neuronales y otros.

3.1 ORIGEN

Matlab se origina con la idea de crear una herramienta de cálculo confiable y muy poderoso, capaz de resolver, con poco esfuerzo, desde hacer cálculos rutinarios con matrices, a escribir programas para hacer tareas más complejas de forma muy sencilla.

La palabra **MATLAB** proviene de “*matrix laboratory*” con su traducción al español de “laboratorio matricial”. Fue creado originalmente como una interfaz para librerías de rutinas de **Fortran** como **IESPACK** y **LINPACK**, los cuales son considerados como el estado de las artes para resolver problemas de álgebra matricial. Hoy en día **MATLAB** no solamente se limita a resolver problemas numéricos sino que ofrece una gran cantidad de herramientas que permiten vincularse con otros programas, hacer adquisición de datos, control en tiempo real, hacer procesamiento simbólico, etc. El diseño o arquitectura de **MATLAB** es muy fácil para el usuario, a tal punto que él puede crear herramientas personalizadas para un mejor desempeño.

3.2 INICIACIÓN DE **MATLAB**

MATLAB es un lenguaje de computación técnico de alto nivel y un entorno interactivo para desarrollo de algoritmos, visualización de datos, análisis de datos y cálculo numérico. Sea cual sea la necesidad que se tenga (un análisis, una estadística, un algoritmo, informes o simulación), **MATLAB** está diseñado para esto. El lenguaje de programación de **MATLAB** es muy fácil e interactivo para el usuario, permite a estudiantes, ingenieros y científicos expresar con facilidad sus ideas técnicas al momento de crear algún cálculo en **MATLAB**. Es un lenguaje de programación que permite manipular vectores o matrices como simples variables.

La funcionalidad de **MATLAB** se agrupa en más de 35 cajas de herramientas (**toolboxes**) y paquetes de bloques (para **Simulink**). Dispone de herramientas para adquisición de datos, exploración y análisis de datos, visualización y procesado de imágenes, prototipaje y desarrollo de algoritmos, modelado y simulación, programación y desarrollo de aplicaciones. Para un buen manejo de **MATLAB** se requiere saber tres puntos claves: 1) cómo se introducen y manipulan las matrices; 2) cómo se utilizan en general las funciones; y 3) cómo obtener información de cada función en particular. El entorno **MATLAB** estimula la creatividad y permite comprobar y comparar rápidamente múltiples alternativas, permitiendo así la obtención de mejores soluciones

3.3 REQUISITOS DEL SISTEMA PARA PODER INSTALAR **MATLAB** EN CONSOLA **WINDOWS**.

1. Tener un PC o portátil con de 512 Mb de memoria RAM o más.
2. Espacio en disco duro de 1gb para **MATLAB** y 3-4 gb para una instalación típica.
3. **Windows XP**, Vista y **Seven**.
4. Procesador Celeron, **Intel Pentium** 4 en adelante, AMDx86 y otros compatibles para el sistema.
5. Puerto USB o CD-ROM para **drivers** de **MATLAB**.

3.4 PASOS A SEGUIR PARA LA INSTALACIÓN DE **MATLAB** VERSIÓN ESTUDIANTIL.

1. Introducir al PC el CD o memoria USB que contenga los instaladores de **MATLAB** y ejecutar el archivo **SETUP.EXE**
2. Se despliega una ventana y se elige la opción "instalar manualmente sin el uso de Internet y le damos clic al botón **NEXT**.
3. A continuación **Aceptar** los términos y condiciones para la licencia del programa y damos **click** al botón **NEXT** para seguir con los siguientes pasos.

4. En el siguiente paso introducimos la clave de activación del producto que normalmente se lo ubica en los archivos de instalación y damos **click** en el botón **NEXT**.
5. Seleccionamos la opción de **instalación típica** la cual nos instala todos los productos que trae el programa, la instalación personalizada nos permite seleccionar que productos instalar, damos **click** al botón **NEXT** para continuar.
6. Elegimos la ruta donde queremos que se instale **MATLAB**, normalmente es archivos de programas que contiene todos los programas y damos **click** a **NEXT**.
7. A continuación se muestra una ventana de confirmación de todos los pasos que se hicieron anteriormente y se continúa dando **click** en el botón de instalar, se espera que se instalen todos los componentes de **MATLAB**, se reinicia el computador y se procede a acceder a **MATLAB**.

3.5 INGRESO AL PROGRAMA

Click en Inicio – archivos de programa – **MATLAB**, o desde el escritorio, en donde se ha creado un acceso directo del programa. Al ejecutar **MATLAB** se abre una ventana como se muestra en la Figura 3.1.

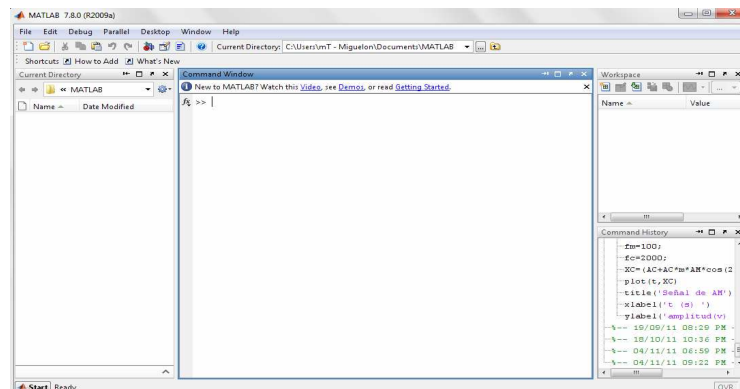


Figura 3.1. Modelo de Ventana inicial de MATLAB
Fuente: Print Screen del Programa MATLAB

Dado que la interfaz del escritorio de **MATLAB** puede ser personalizada para adaptarla a cualquier estilo de trabajo y usar selectivamente las funciones que

sean necesarias en cada fase del proyecto, es posible que en muchos casos, la apariencia de la ventana que abrimos sea diferente.

Lo primero que se le presenta al usuario al iniciar **MATLAB** es el denominado **Command Windows**, el cual cumple la función de intérprete entre el usuario y el núcleo de procesamiento. Esta ventana presenta en el margen izquierdo un prompt `>>` donde se escribirán las instrucciones. Cada línea será ejecutada luego de presionar la tecla **Enter**. En casi todos los casos, excepto los comandos gráficos, los resultados aparecerán en el mismo **Command Windows**. En la parte derecha de la fig. 3.1., aparecen dos ventanas muy útiles para esta aplicación y que son **workspace** y **command history**.

La **COMMAND HISTORY** ofrece acceso a las sentencias que se han ejecutado anteriormente en el **Command Windows**. **Workspace** contiene información sobre todas las variables que se hayan definido en esta sesión y acceder a **Array Editor** para ver y modificar los datos. Al lado izquierdo de la fig. 3.1 se encuentra la venta **CURRENT DIRECTORY** que permite seleccionar un directorio para trabajar en él. Es posible navegar por los archivos del directorio, ejecutarlo y modificarlo.

3.5.1. ACERCA DE MATLAB

Dentro de **MATLAB** existe otro punto muy fuerte que son las gráficas que se verán más detalladas en una posterior sección. Para un título se puede escribir la siguiente línea y pulsar intro: en la ventana de **Command Windows**.

```
>> x=-4: .01: 4; y=sin(x); plot(x,y), grid, title('FUNCION SENO(x)')
```

A continuación se despliega una nueva ventana en la que se muestra la gráfica 3.2 de la Función $\sin(x)$. Esta figura representa los datos ingresados anteriormente, muestra el título "Función seno(x)" y una cuadrícula o "**grid**". La línea anterior también contiene otras instrucciones como la de crear un vector x con 801 valores reales entre -4 y 4, separados por una centésima. Después se crea un vector X , se dibujan los valores de Y en ordenadas frente a los de X abscisas. Las dos últimas cuadrículas establecen la cuadrícula y el título.

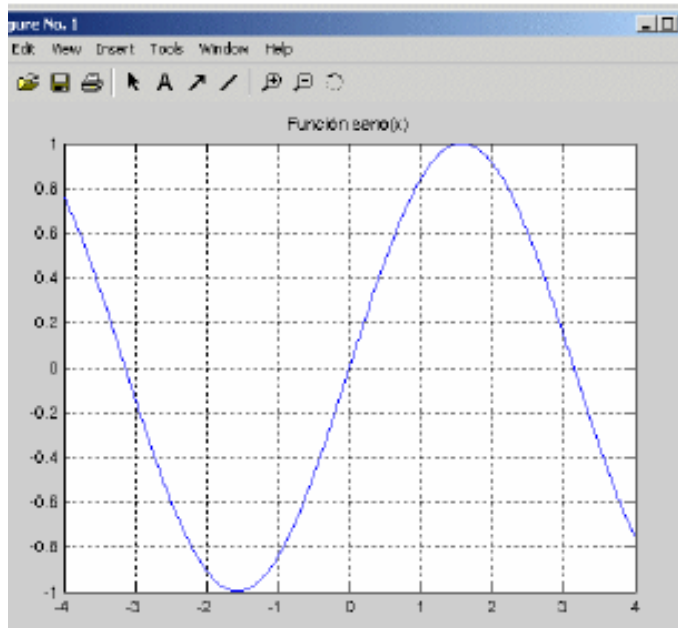


Figura 3.2. Gráfico de la Función Seno(X)
 Fuente: **Print Screen** del Programa MATLAB

Con el **Command History** se pueden recuperar comandos ingresados con anterioridad por medio de las flechas arriba y abajo, con el primer pulso de la flecha aparece el último comando ingresado. Se puede borrar toda una línea de programación utilizando la tecla ESC. Para realizar una limpieza del **Command History** se utilizan los comandos **CLC** y **HOME**. La función **CLC** borra todas las salidas anteriores en cambio **HOME** las mantiene, pero lleva el **PROMPT(>>)** a la primera línea de la ventana. Para salir de MATLAB utilizamos el comando **QUIET** o **EXIT** desde el menú **file**.

3.5.2. USO DEL **HELP** O AYUDA

MATLAB dispone de una ayuda muy completa y accesible, estructurada en varios niveles (línea de comandos en la **Command Window**, ventana **Help**, y manuales en formato PDF) con la que es muy importante estar familiarizado, porque hasta los más expertos programadores tienen que acudir a ella con una cierta frecuencia. La figura 3.3 muestra cómo tener acceso a esta herramienta.

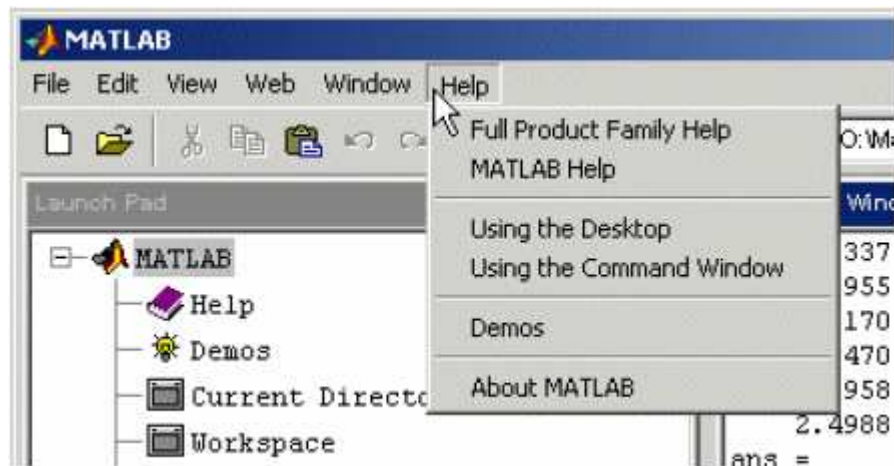


Figura 3.3. Gráfico de ubicación del HELP
Fuente: Print Screen del Programa MATLAB

3.5.3. FULL PRODUCT FAMILY HELP

En la barra de menú en el área de **HELP** encontramos esta opción que sirve para buscar información acerca de otros productos de la familia **MATLAB** a los que se tenga acceso. La gran mayoría de las páginas de ayuda están en formato HTML.

3.5.4. MATLAB HELP

Al ejecutar **MATLAB Help** se abrirá una ventana igual a las figuras 3.4 (ventana inicial de **help Windows**) y 3.5 (ventana inicial de **Help Desk**) respectivamente. Esta opción permite la búsqueda de cualquier información acerca de **Matlab** como: conceptos, funciones, comandos, etc. En esta portada encontramos tres ítems

- **Learning Matlab:** Basado en contenidos de **MATLAB**.
- **Finding Functions and Properties:** Da el acceso a la información de diferentes funciones o propiedades de los objetos Gráficos.
- **Printing the Documentation:** Da acceso a manuales PDF del programa que se pueden imprimir, si lo desea el usuario.



Figura 3.4. Ventana inicial de help windows.
Fuente: Print Screen del Programa MATLAB

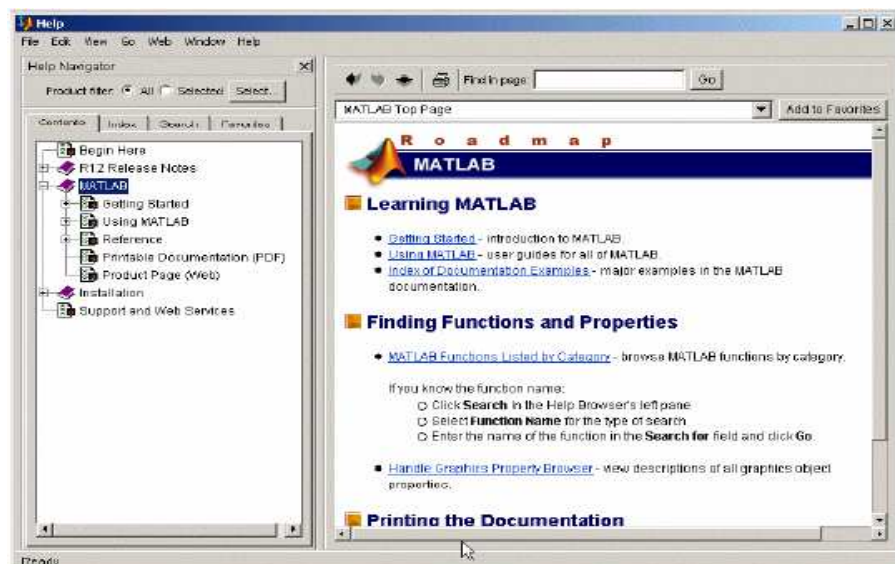


Figura 3.5. Ventana inicial de Help Desk.
Fuente: Print Screen del Programa MATLAB

3.5.5. USING THE DESKTOP

Se abre una ventana muy parecida a la que se muestra en la figura 3.3 y 3.4 con información sobre cómo utilizar y configurar el entorno de desarrollo o **Desktop**. Las distintas herramientas disponibles se describen sucesivamente. Cada página dispone de flechas y enlaces que permiten ir a la página siguiente o volver a la anterior. Una característica importante, es la posibilidad de organizar las ventanas con gran flexibilidad.

3.5.6. USING THE COMMAND WINDOWS

Command Windows es la base y lo más importante de **MATLAB** aquí encontramos toda la información necesario para aprovechar las capacidades de **command Windows**.

3.5.7. DEMOS

Da acceso a un gran número de ejemplos resueltos con **MATLAB**, cuyos resultados se presentan gráficamente de diversas formas. Es muy importante analizar estos ejemplos resueltos para poder captar más la idea de **MATLAB**.

3.6 EL ENTORNO DE TRABAJO DE **MATLAB**

Hay muchas mejoras a partir del **MATLAB** 6.0 dentro de ésta, está la del entorno de trabajo, haciéndose mucho más interactivo, similar a las aplicaciones profesionales de **Windows**. A continuación se explica con más detalle los componentes del entorno de trabajo. Los componentes más sobresalientes del entorno de trabajo de **MATLAB** son los siguientes:

- **Escritorio de **MATLAB** (*Matlab Desktop*)**
Es la ventana o contenedor de máximo nivel en la que se pueden situar (**to dock**) las demás componentes.

- **Componentes individuales orientados a tareas concretas**
 - a) La ventana de comandos (**Command Windows**).
 - b) La ventana histórica de comandos (**Command History Browser**).
 - c) El espacio de trabajo (**Workspace Browser**).
 - d) La plataforma de lanzamiento (**Launch Pad**).
 - e) El directorio actual (**Current Directory Browser**),
 - f) La ventana de ayuda (**Help Browser**)
 - g) El editor de ficheros y depurador de errores (**Editor&Debugger**).
 - h) El editor de vectores y matrices (**Array Editor**).
 - i) Empleo del tiempo de ejecución (**Profiler**)

A continuación se describen cada uno de los puntos anteriores para tenerlos en claro porque con el conocimiento de los mismos hará de **MATLAB** mucho más fácil en el entorno de trabajo.

3.6.1. ESCRITORIO DE **MATLAB**

El **MATLAB Desktop** es la ventana más general de la aplicación. Los demás componentes son alojarse o ejecutarse como ventanas independientes.

La figura 3.6 que se mostrara a continuación muestra la vista grafica de como se ve el menú **Desktop**, desde aquí controlamos la vista del **MATLAB**. Por ejemplo en la figura 3.7 tenemos activa la ventana de **command Window**, en la figura 3.7 aparece el menú del VIEW con la opción de dejar de alojar la ventana en el **MATLAB Desktop** su nombre se muestra **UNDOCK COMMAND WINDOW**. Este menú también da la opción de eliminar componentes visibles en el **desktop**. Con la configuración solo por defecto (**DEFAULT**) al cerrar **Matlab** y al volverlo a ejecutar se mantendrá la configuración que el usuario haya adoptado.

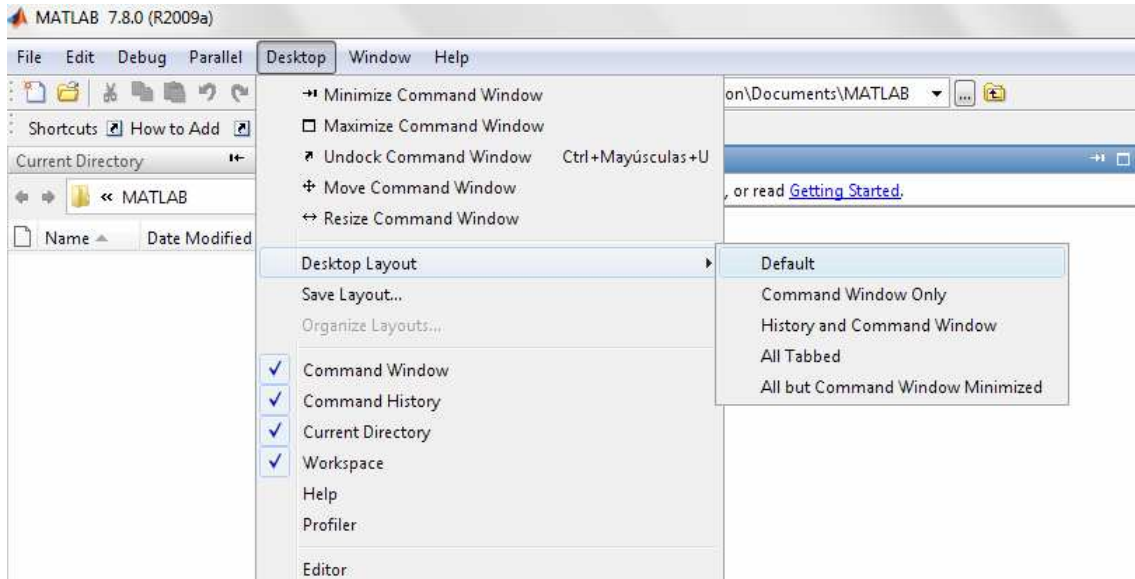


Figura 3.6. Menú para configurar el MATLAB Desktop.

Fuente: Print Screen del Programa MATLAB

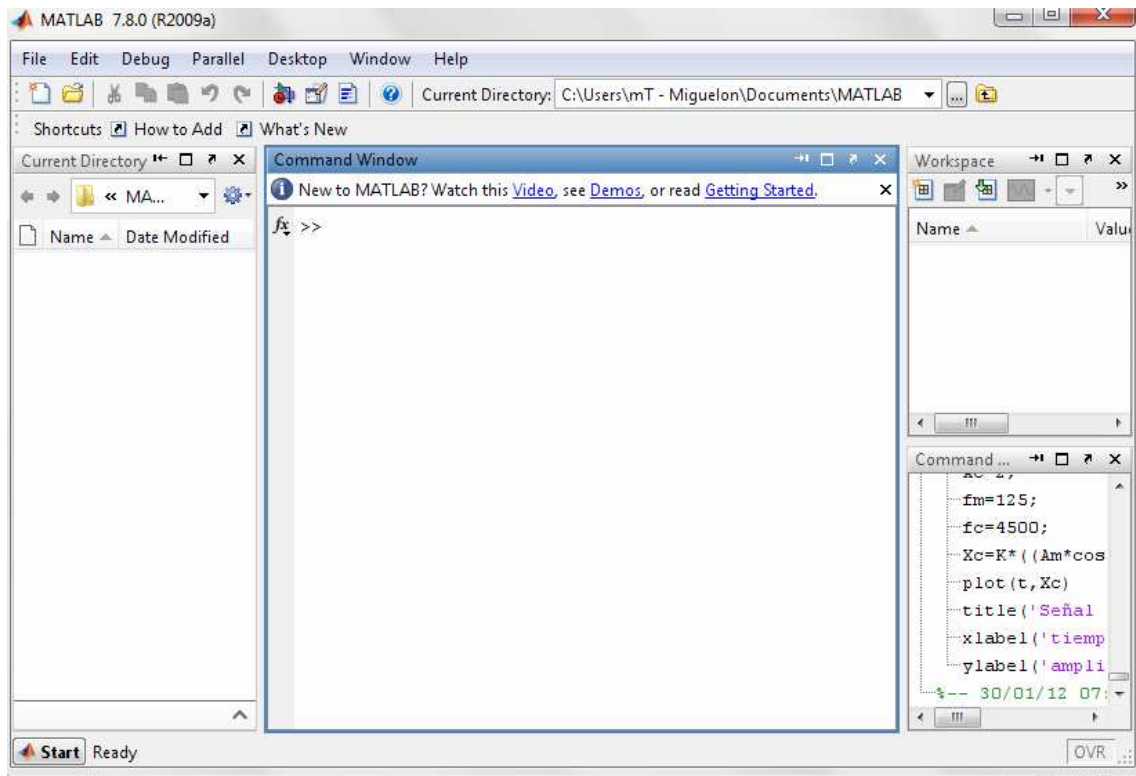


Figura 3.7. Configuración por defecto del MATLAB Desktop.

Fuente: Print Screen del Programa MATLAB

3.7 **COMMAND WINDOW**

En esta ventana se ejecutan interactivamente las instrucciones de **MATLAB** y en donde se visualizan los resultados correspondientes. Es la ventana más importante y la única que existe en versiones anteriores de la aplicación. En esta versión se han añadido otras mejoras tales, como:

- Permite líneas de comando muy largas que se corren automáticamente continúe a la línea siguiente cuando llega al margen derecho de la ventana. Para esta opción ahí que activar el **WRAP LINES**, que se encuentra en el menú **FILE/PREFERENCES/COMMAND/WINDOWS**.
- Dando click derecho sobre el nombre de una función que aparezca en esta ventana de comandos se puede tener acceso a la página de **HELP** y obtener información sobre dicha función.
- Escribiendo el nombre de una función sobre la ventana de comandos y pulsando **TAB, MATLAB** completa automáticamente el nombre de la función, o bien muestra todas las funciones disponibles en las líneas siguientes y que comiencen con las letras tecleadas por el usuario.

3.8 LAUNCH PAD

El **Launch Pad** se considera un recurso de **MATLAB** muy general que da acceso a todos los módulos o componentes que se tengan instalados como: **HELP BROWSER, DEMOS, CURRENT DIRECTORY, WORKSPACE, PATH y a GUIDE.**

3.9 COMMAND HISTORY BROWSER

Da acceso a los comandos que ejecutados con anterioridad en el **Command Windows**. También están accesibles por medio de las teclas de dirección arriba y abajo. Los comandos ejecutados anteriormente también pueden volverse a ejecutar mediante un doble click o también por medio del menú contextual que se abre al dar click derecho sobre ellos.

3.10 CURRENT DIRECTORY BROWSER

Esta opción da paso a exploraciones de los directorios del ordenador en forma análoga a la del explorador u otras aplicaciones de **Windows**. Cuando se encuentra el directorio deseado se muestran los ficheros y contenidos. También permite tener un orden de los directorios por fecha, tamaño nombre, etc. El directorio que se tiene actualmente cambia automáticamente en función del directorio seleccionado con este **browser** y también se puede modificar desde la barra de herramientas de **MATLAB Desktop**.

3.11 PATH DE MATLAB: Establecer el camino de búsqueda (Search Path)

Es una lista de directorios que se puede ver y modificar a partir de la línea de comandos, o utilizando el cuadro de diálogo **Set Patch** del menú **File**.

3.12 **WORKSPACE BROWSER**

El **workspace** es el conjunto de funciones y variables de usuario que en tiempo dado están definidas en la memoria del programa. Para obtener información acerca del **workspace** se puede acceder desde las líneas de comandos utilizando el **who** y el **whos**. El **whos** da una información más detallada que el **who**.

En la ventana **workspace browser** podemos visualizar las variables definidas en el espacio de trabajo. Esta se activa con el comando **VIEW/WORKSPACE**.

3.13 **ARRAY EDITOR**

Este permite visualizar los valores de elementos de cualquier matriz o vector que esté definido en el programa. Estos valores pueden ser modificados dando clic sobre la celda correspondiente. Es muy útil para la comprensión de algoritmos, ejecutando paso a paso un programa y viendo cómo cambian los valores de las distintas variables.

3.14 **EL EDITOR/DEBUGGER**

Son ficheros del texto ASCII con su extensión *.m, de tal manera que contienen conjuntos de comandos o definición de funciones. Su importancia radica en que al teclear el nombre en la línea de comandos y dar **Enter**, se ejecutan uno tras otro todos los comandos contenidos en dicho fichero, lo que es un ahorro de trabajo y tiempo.

3.15 **PREFERENCIAS: FORMATOS DE SALIDA Y DE OTRAS OPCIONES DE MATLAB**

MATLAB tiene un cuadro de diálogo donde el usuario tiene la opción de determinar los parámetros por su cuenta. Este cuadro de diálogo se abre con el comando **PREFERENCES** del menú **FILE**. A continuación mostraremos en la figura 3.8 el cuadro de dialogo **PREFERENCES** mostrado las opciones que ofrece. En la figura se muestra que son 18 cuadros de diálogos. En la figura 3.9

mostramos que hay la opción **COMMAND WINDOW/FONT&COLORS**, ofrece la posibilidad de elegir el tipo de letra, color y tamaño ya sea del fondo o de las letras.

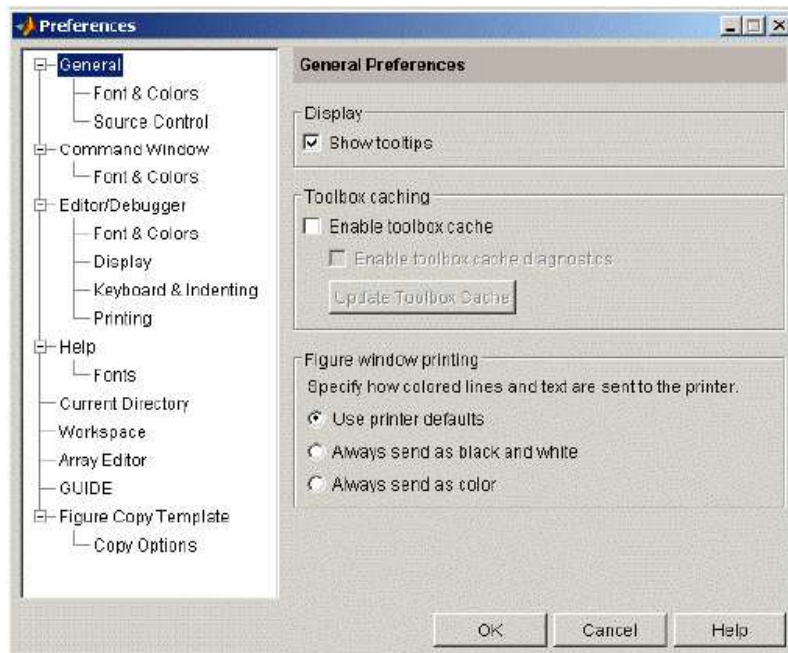


Figura 3.8. Cuadro de dialogo PREFERENCIAS/GENERAL.
Fuente: Print Screen del Programa MATLAB

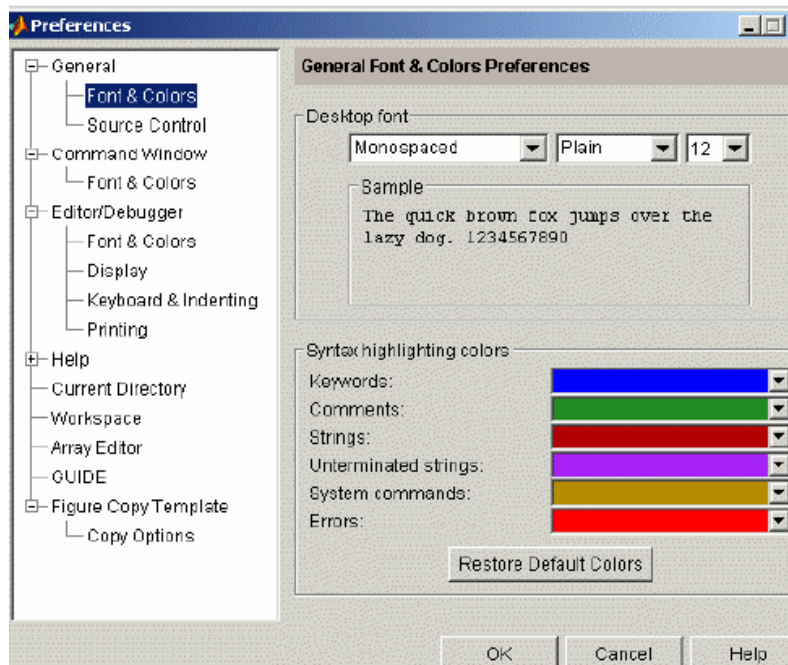


Figura 3.9. Cuadro de dialogo PREFERENCIAS/FONTS&COLOR.
Fuente: Print Screen del Programa MATLAB

3.16 GUARDAR VARIABLES Y ESTADOS DE UNA SESIÓN: COMANDOS *SAVE* Y *LOAD*

Para guardar el estado de una sesión de trabajo existe el comando **SAVE**. Hay que tener en cuenta que al salir del programa todo el contenido de la memoria se borra automáticamente, por lo que es importante guardar la información antes de salir. Se pueden también guardar matrices y vectores de forma selectiva.

Existe la forma de guardado por medio de **>>save** tecleándolo en la ventana y cerrar el programa y los datos quedan respaldados en su ordenador. Cuando se quiera volver a abrir el archivo se teclaea **>>load**. Existe también la opción de configurar el ordenador para que haga un guardado diario esta opción se la puede activar por medio de la función **>>diary filename.txt** y se la puede desactivar con **>>diary off** y se la reanuda con **>>diary on**.

3.17 LÍNEAS DE COMENTARIOS

MATLAB permite comentar bloques de sentencias, seleccionando las sentencias que se desean comentar. Dando clic en el botón derecho elegir la opción **Comment** en el menú que se abre, las sentencias seleccionadas se comentan individualmente con el carácter %.

Las líneas de comentarios ayudan mucho para programas largos y se los comenta para identificar cada cosa o cada programación.

3.18 FUNCIONES DE LIBRERÍAS

MATLAB tiene un gran número de funciones incorporadas, unas de ellas son **funciones intrínsecas**, es decir, funciones incorporadas en el propio código ejecutable del programa. Estas funciones son por lo general son rápidas y eficientes. **MATLAB** posee diferentes tipos de funciones. A continuación se nombrarán los tipos de funciones más importantes, clasificadas según su finalidad:

- 1.- Funciones matemáticas elementales.
- 2.- Funciones especiales.
- 3.- Funciones matriciales elementales.
- 4.- Funciones matriciales específicas.
- 5.- Funciones para la descomposición y/o factorización de matrices.
- 6.- Funciones para análisis estadístico de datos.
- 7.- Funciones para análisis de polinomios.
- 8.- Funciones para integración de ecuaciones diferenciales ordinarias.
- 9.- Resolución de ecuaciones no-lineales y optimización.
- 10.- Integración numérica.
- 11.- Funciones para procesamiento de señal.

3.19 FUNCIONES MATEMÁTICAS ELEMENTALES

Comprende funciones básicas y funciones matemáticas trascendentales. Cuando se aplican a una matriz actúan sobre cada elemento de la matriz como se tratase de un escalar. Estas funciones se aplican de la misma forma a escalares, vectores y matrices.

$\sin(x)$	seno
$\cos(x)$	coseno
$\tan(x)$	tangente

3.20 TIPOS DE DATOS DE MATLAB

MATLAB puede también trabajar con otros tipos de datos, veamos a continuación:

- Conjuntos o cadenas de caracteres, fundamentales en cualquier lenguaje de programación.
- Vectores o matrices de celdas (**cell arrays**), son vectores o matrices cuyos elementos pueden ser cualquier otro tipo de dato.
- Matrices dispersas o matrices dispersas, que son matrices que pueden ser de muy gran tamaño con la mayor parte de sus elementos cero.
- Hipermatrices, o matrices de más de dos dimensiones.

- Estructuras, o agrupaciones bajo un mismo nombre de datos de naturaleza diferente.

3.21 PROGRAMACIÓN DE *MATLAB*

MATLAB es una aplicación que se puede programar muy fácilmente y que no tiene tantas dificultades como otros lenguajes. Comencemos viendo las bifurcaciones y bucles, y la lectura y escritura interactiva de variables, estos son los elementos básicos de cualquier programa con cierta complejidad. Básicamente nosotros no nos vamos a profundizar en lo que es programación, dado que nuestros ejercicios se basan en ingresar una formula y mostrarlo ya con un aspecto más amigable que sería ingresando ciertas funciones.

3.22 FICHEROS *.M

Estos ficheros de texto sin formato con extensión (.m) (ficheros ASCII) que conforman el centro de la programación en *MATLAB*. Se crean y modifican con un editor de textos cualquiera. En el caso de *MATLAB*, ejecutado en un PC bajo *Windows*, lo mejor es utilizar su propio editor de textos, que es también *Debugger*. Existen dos tipos de ficheros *.m:

1. Los **ficheros de comandos** (llamados *scripts* en inglés). Contienen un conjunto de comandos que se ejecutan sucesivamente cuando se teclaea el nombre del fichero en la línea de comandos de *MATLAB* o se incluye dicho nombre en otro fichero *.m. Un fichero de comandos puede llamar a otros ficheros de comandos. Si un fichero de comandos se llama desde de la línea de comandos de *MATLAB*, las variables que crea pertenecen al espacio de trabajo base de *MATLAB*, y permanecen en él cuando se termina la ejecución de dicho fichero).
2. **Las funciones.** (permiten definir funciones enteramente análogas a las de *MATLAB*, con su nombre, sus argumentos y sus valores de retorno. Los ficheros *.m que definen funciones permiten extender las posibilidades de *MATLAB*; de hecho existen bibliotecas de ficheros *.m que se venden (*toolkits*) o se distribuyen gratuitamente a través de Internet).

Las funciones que se definen en ficheros *.m se caracterizan porque la primera línea (que no sea un comentario) comienza por la palabra **function**, seguida por los valores de retorno (entre corchetes [] y separados por comas, si hay más de uno), el signo igual (=) y el nombre de la función, seguido de los argumentos (entre paréntesis y separados por comas).

Un fichero *.m puede llamar a otros ficheros *.m, e incluso puede llamarse a sí mismo de forma recursiva. Los ficheros de comandos se pueden llamar también desde funciones, en cuyo caso las variables que se crean pertenecen a espacio de trabajo de la función. El espacio de trabajo de una función es independiente del espacio de trabajo base y del espacio de trabajo de las demás funciones.

Esto implica que no puede haber colisiones entre nombres de variables: aunque varias funciones tengan una variable llamada A, en realidad se trata de variables completamente distintas (a no ser que A haya sido declarada como variable global).

3.23 FICHEROS DE COMANDOS (SCRIPTS)

Son ficheros que contienen una sucesión de comandos análoga a la que se teclearía en el uso interactivo del programa. Esos comandos se ejecutan sucesivamente cuando se teclea el nombre del fichero que los contiene (sin la extensión), es decir, cuando se teclea **file1** con el ejemplo considerado.

Al ejecutar desde la línea de comandos, las variables creadas por **file1** pertenecen al espacio de trabajo base de **MATLAB**. En caso contrario, si se ejecuta desde una función, las variables que crea pertenecen al espacio de trabajo de la función de los ficheros de comandos conviene poner los puntos y coma (;) al final de cada sentencia, para evitar una salida de resultados demasiado cuantiosa. Un fichero *.m puede llamar a otros ficheros *.m, e incluso se puede llamar a sí mismo de modo recursivo. Sin embargo, no se

puede hacer **profile** de un fichero de comandos sólo se puede hacer de las funciones.

3.24 **HELP PARA LAS FUNCIONES DE USUARIO**

Las diferentes funciones creadas por el usuario pueden tener su **help**, análogo al que tienen las propias funciones de **MATLAB**. Esto se logra de la siguiente forma: las primeras líneas de comentarios de cada fichero de función son muy importantes, pues permiten construir un **help** sobre esa función. En otras palabras, cuando se teclea en la ventana de comandos de **MATLAB**:

```
>> help mi_func
```

El programa responde escribiendo las primeras líneas del fichero **mi_func.m** que comienzan por el carácter (%), es decir, que son comentarios.

3.25 **HELP DE DIRECTORIOS**

MATLAB permite que los usuarios creen una ayuda general para todas las funciones que están en un determinado directorio. Para ello se debe crear en dicho directorio un fichero llamado **contents.m**.

3.26 **INTEGRACIÓN NUMÉRICA DE FUNCIONES**

Primero se calcula la integral definida de esta función entre dos valores de la abscisa x. En inglés, al cálculo numérico de integrales definidas se le llama **quadrature**. Sabiendo eso, no resulta extraño el comando con el cual se calcula el área comprendida bajo la función entre los puntos 0 y 1 (obsérvese que la referencia de la función a integrar se pasa por medio del operador @ precediendo al nombre de la función. También podría crearse una variable para ello):

```
>> área = quad(@prueba, 0, 1)
```

```
área = 29.8583
```

Al teclear **help quad** se puede obtener más de información sobre esta función, incluyendo el método utilizado (**Simpson**) y la forma de controlar el error de la integración. La función **quadl()** emplea un método de orden superior (**Lobatto**), en tanto que la función **dblquad()** realiza integrales definidas dobles.

3.27 GRÁFICOS BIDIMENSIONALES

Los gráficos 2-D de **MATLAB** están orientados a la representación gráfica de vectores (y matrices). Los argumentos básicos de la función **plot** van a ser vectores. Cuando una matriz aparezca como argumento, se considerará como un conjunto de vectores columna (en algunos casos también de vectores fila).

En el programa de **MATLAB** se utiliza un tipo especial de ventanas para realizar las operaciones gráficas. Ciertos comandos abren una ventana nueva y otros dibujan sobre la ventana activa, bien sustituyendo lo que hubiera en ella, bien añadiendo nuevos elementos gráficos a un dibujo anterior.

3.28 FUNCIONES GRÁFICAS 2D ELEMENTALES

El programa **MATLAB** contiene cuatro funciones básicas para crear gráficos 2D. Éstas se diferencian por el *tipo de escala* que utilizan en los ejes de abscisas y de ordenadas. Estas cuatro funciones son las siguientes:

plot() Permite crear un gráfico a partir de vectores y/o columnas de matrices, con escalas lineales sobre ambos ejes

loglog() *Idem* con escala logarítmica en ambos ejes

semilogx() *Idem* con escala lineal en el eje de ordenadas y logarítmica en el eje de abscisas

Abscisas semilogy() *Idem* con escala lineal en el eje de abscisas y logarítmica en el eje de ordenadas.

Borrar texto (u otros elementos gráficos) es un poco más complicado; para ello, es necesario preverlo de antemano. Para poder hacerlo hay que recuperar previamente el valor de retorno del comando con el cual se ha creado. Después hay que llamar a la función **delete** con ese valor como argumento.

3.29 FUNCIÓN **PLOT**

Es la función clave de todos los gráficos 2-D en **MATLAB**. El elemento básico de los gráficos bidimensionales es el vector. También se emplean cadenas de 1, 2 ó 3 caracteres para indicar *colores* y *tipos de línea*. La función **plot()**, en sus diversas variantes, no hace otra cosa que dibujar vectores. **Plot** se aplica tomando los valores de (x,y) trazando los esquemas de ondas en el plano con sus respectivos valores asignado.

3.30 FUNCIÓN **TITLE**

Esta función establece una cadena en la parte superior y el centro de los ejes en una grafica o plano. (Ver figura 3.10)

title('Señal de AM')

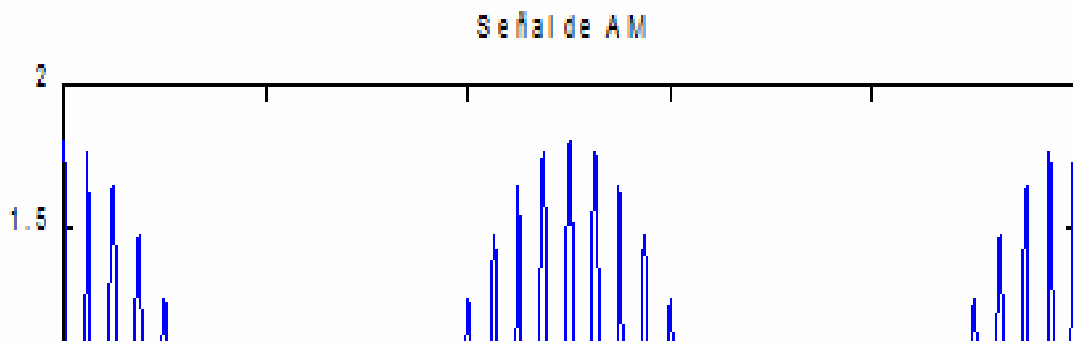


Figura 3.10. Grafico aplicando función **TITLE**.
Fuente: Print Screen del Programa **MATLAB**

3.31 FUNCIÓN **XLABEL** y **YLABEL**

La función **xlabel** y **ylabel** son parecida a la función **TITLE** ya que también establecen nombres en las graficas, la función **XLABEL** establece un nombre

en las graficas de las coordenadas de las “X” y la función **YLABEL** establece un nombre en las graficas de las coordenadas de las “Y”. (Ver figura 3.11)

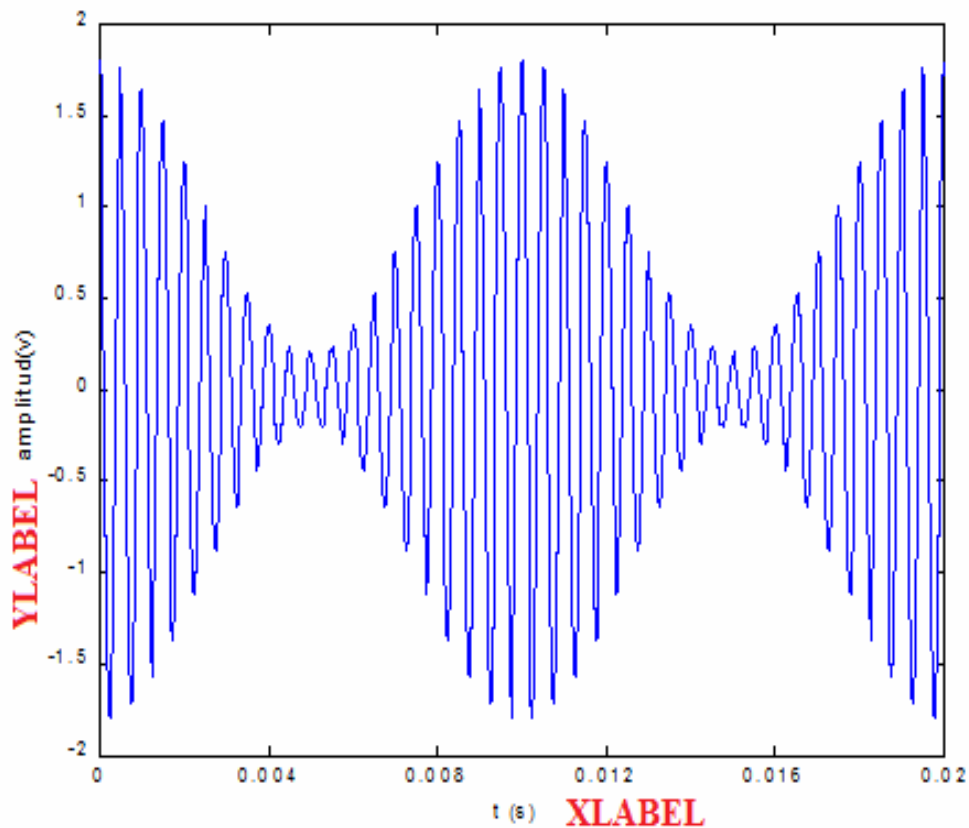


Figura 3.11. Grafico aplicando función XLABEL Y YLABEL.
Fuente: Print Screen del Programa MATLAB

3.32 ESTILOS DE LÍNEA Y MARCADORES EN LA FUNCIÓN **PLOT**

La tarea fundamental de la función **plot()** es dibujar los valores de un vector en ordenadas, frente a los valores de otro vector en abscisas. Esto exige que se pasen como argumentos un par de vectores. El conjunto básico de argumentos de esta función es una *tripleta* formada por dos vectores y una cadena de 1, 2 ó 3 caracteres que indica el color y el tipo de línea o de **marker**. En la tabla 3.1 se muestra los estilos de línea y marcadores de la función **plot**.

Tabla 3.1: Estilos de línea y marcadores en la función *plot*.

Símbolo	Color	Símbolo	Marcadores (<i>markers</i>)
Y	Yellow	.	Puntos
M	Magenta	O	Círculos
C	Cyan	X	Marcas en x
R	Red	+	Marcas en +
G	Green	*	Marcas en *
B	Blue	S	marcas en diamante (diamond)
W	White	D	triángulo apuntando arriba
K	Black	^	triángulo apuntando abajo
		V	
Simbolo	Estilo de línea	>	triángulo apuntando a la dcha.
-	Líneas continuas	<	triángulo apuntando a la izda.
:	Líneas a puntos	P	estrella de 5 puntas
-.	Líneas a barra-punto	H	estrella se seis puntas
--	Líneas a trazos		

En el caso de querer dibujar varias líneas, por defecto se van cogiendo sucesivamente los colores de la tabla comenzando por el azul, hacia arriba, y cuando se terminan se vuelve a empezar otra vez por el azul. Si el fondo es blanco, este color no se utiliza para las líneas.

3.33 CONTROL DE LOS EJES

El control de los ejes tiene sus opciones por defecto, que en algunas ocasiones puede interesar cambiar. Su comando básico es el *axis*. Por defecto, **MATLAB** ajusta la escala de cada uno de los ejes de modo que varíe entre el mínimo y el

máximo valor de los vectores a representar. Este es el llamado modo "auto", o modo automático.

3.34 **FUNCIÓN LINE()**

Permite dibujar una o más líneas que unen los puntos cuyas coordenadas se pasan como argumentos. Además, permite especificar el color, grosor, tipo de trazo, marcador, etc. Es una función de más bajo nivel que la función **plot ()**, pero ofrece una mayor flexibilidad. En su versión más básica, para dibujar un segmento de color verde entre dos puntos, esta función se llamaría de la siguiente manera: **line ([xini, xend]', [yini, yend]', 'color', 'g')**

Se puede también dibujar dos líneas a la vez utilizando la forma: **line ([xini1 xini2; xend1 xend2], ([yini1 yini2; yend1 yend2])**. Finalmente, si cada columna de la matriz X contiene la coordenada x inicial y final de un punto, y lo mismo las columnas de la matriz Y con las coordenadas y, la siguiente sentencia dibuja tantas líneas como columnas tengan las matrices X e Y:

Line ([X], [Y]); Se pueden controlar las características de la línea por medio de pares parámetro/valor.

3.35 **INTRODUCCIÓN A SIMULINK**

SIMULINK que es un programa de simulación con interfaz gráfico que suplementa al **MATLAB**. Esta nueva herramienta nos permite describir gráficamente un sistema dibujando su diagrama en bloques, su resultado es muy conveniente para la simulación de sistemas dinámicos.

3.36 **ACCESO Y DESCRIPCIÓN**

El acceso a **Simulink** es muy sencillo de emplear, basta con escribir **SIMULINK** en la ventana de comando de **MATLAB**, o también haciendo click en el icono (figura 3.12) que se localiza en la barra de herramientas de **MATLAB**.



Figura 3.12. Grafico de icono para acceso a SIMULINK.

Fuente: Print Screen del Programa MATLAB

Al arrancar **Simulink** se nos despliega una ventana gráfica que contiene todas las librerías que el entorno de **Simulink** bajo **MATLAB** soporta, la figura 3.13 muestra gráficamente como es la ventana.

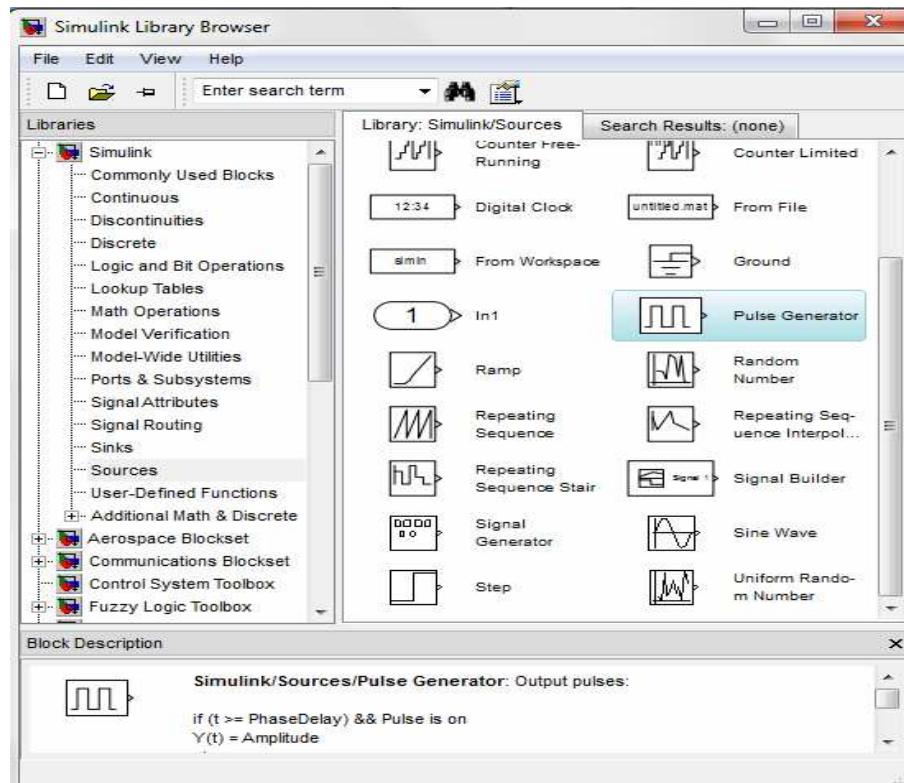


Figura 3.13. Ventana de librería del entorno SIMULINK

Fuente: Print Screen del Programa MATLAB/SIMULINK

Para crear un proyecto o un esquema de trabajo nos ubicamos en la librería que deseamos utilizar. Nos ubicamos en la barra de menú en la parte de **FILE** y seleccionamos la opción **new model** y se nos desplegará una ventana en blanco que es donde realizaremos nuestro diseño.

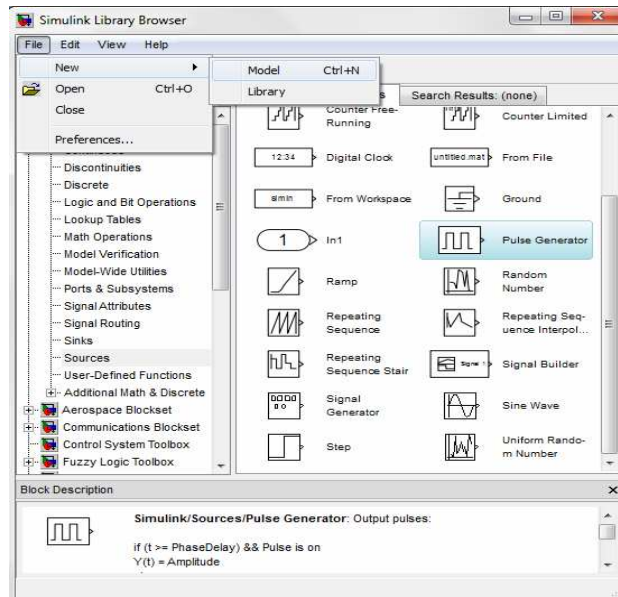


Figura 3.14. Apertura de un nuevo fichero de modelo
Fuente: Print Screen del Programa MATLAB/SIMULINK

Entre las diferentes librerías que tiene **SIMULINK** son:

- **Continuous** (Bloques para sistemas en tiempo continuo)
- **Discrete** (Bloques para sistemas en tiempo discretos)
- **Functions & Tables**
- **Math** (Sumadores, Ganancias matriciales o constantes, etc.)
- **Nonlinear**
- **Signals & Systems** (multiplexores, demultiplexores, etc.)
- **Sinks** (Graficadores, etc.)
- **Sources** (Varias fuentes de entradas)

Con un doble click sobre la librería podemos visualizar los elementos que posee. Por ejemplo si ingresamos a **Continuous**, entre los elementos disponibles utilizaremos los siguientes:

Derivative: bloque derivador, es decir du/dt .

Integrator: bloque integrador, función transferencia 1s.

State-Space: bloque para expresar al sistema en modelo de estados.

Transfer Fnc: bloque para expresar al sistema como cociente de polinomios.

Zero-pole: bloque para expresar al sistema con ceros, polos y una ganancia.

Ver diferentes **toolboxes** del **Simulink** (figuras 3.15, 3.16 y 3.17)

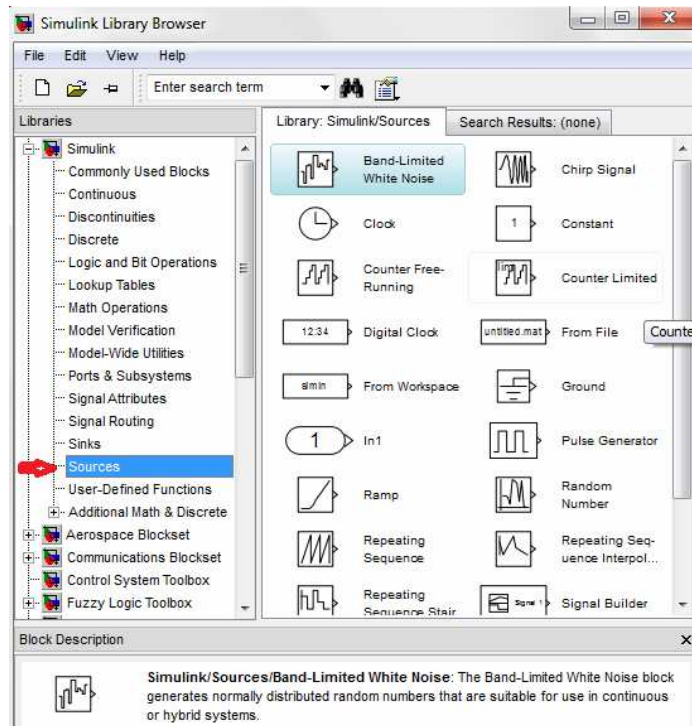


Figura 3.15. Diferentes toolboxes del Simulink. toolbox básica.
Fuente: Print Screen del Programa MATLAB/SIMULINK

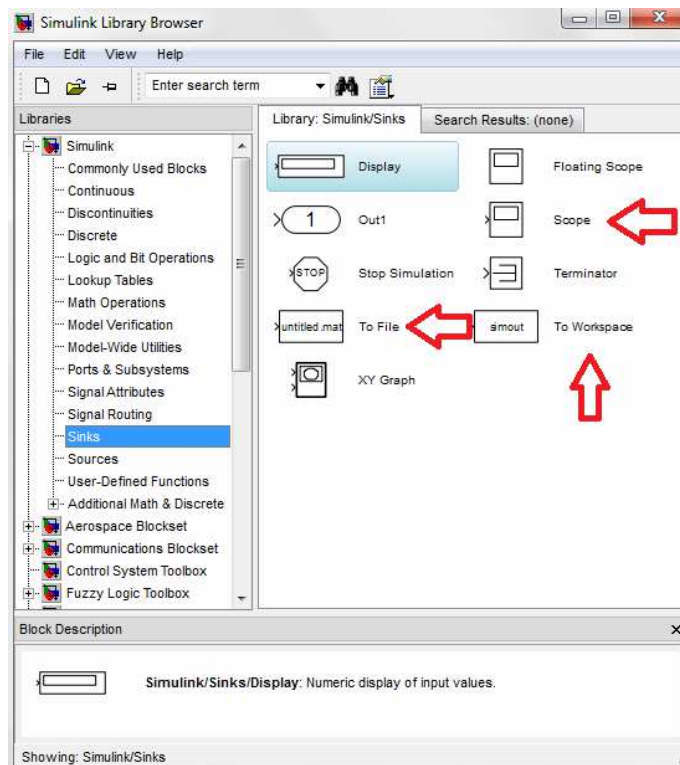


Figura 3.16. Diferentes toolboxes del simulink. toolbox básico.
Fuente: Print Screen del Programa MATLAB/SIMULINK

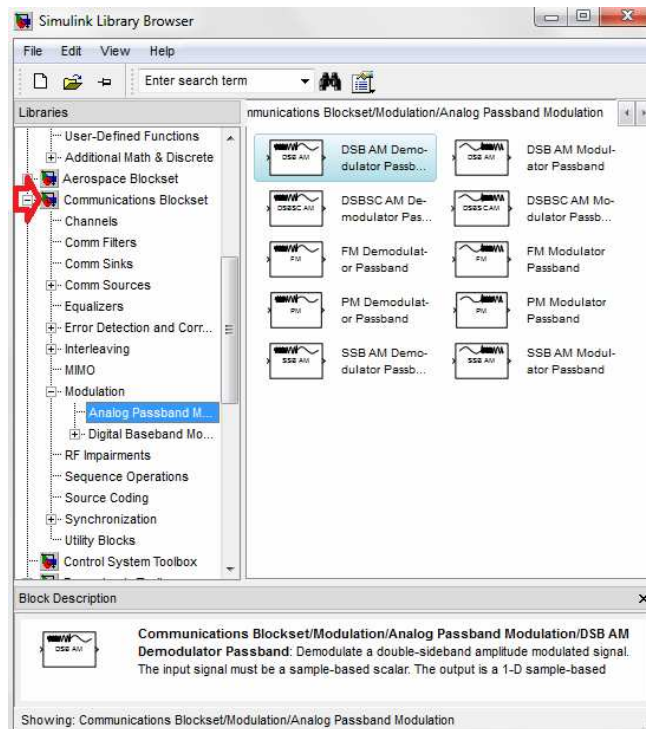


Figura 3.17. Diferentes toolboxes del Simulink. toolbox básica.
Fuente: Print Screen del Programa MATLAB/SIMULINK

3.37 CONSTRUCCIÓN DE UN MODELO EN *SIMULINK*

Supongamos que tenemos el siguiente ejemplo de generar una señal sinusoidal y los datos serian:

- ✓ Amplitud pico: 2 V
- ✓ Frecuencia: 2 kHz
- ✓ Offset (bias): 0 V
- ✓ Desfasaje : 0 radianes
- ✓ Tiempo de muestreo: 0.00001 s (0.01ms)
- ✓ Tiempo de simulación: 5 ms (stop time) (Menú Simulación/**Configuration parameters**)

Ingresamos las librerías correspondientes, en este caso estaríamos utilizando un **sine wave** y un **scope** (osciloscopio). (Ver figura 3.18)



Figura 3.18. Ejemplo de generación de una señal sinusoidal.
Fuente: Print Screen del Programa MATLAB/SIMULINK

Básicamente toda la codificación para realizar los ejercicios se hace interactivamente, en la figura 3.19 se muestra la configuración de los parámetros de la señal sinusoidal.

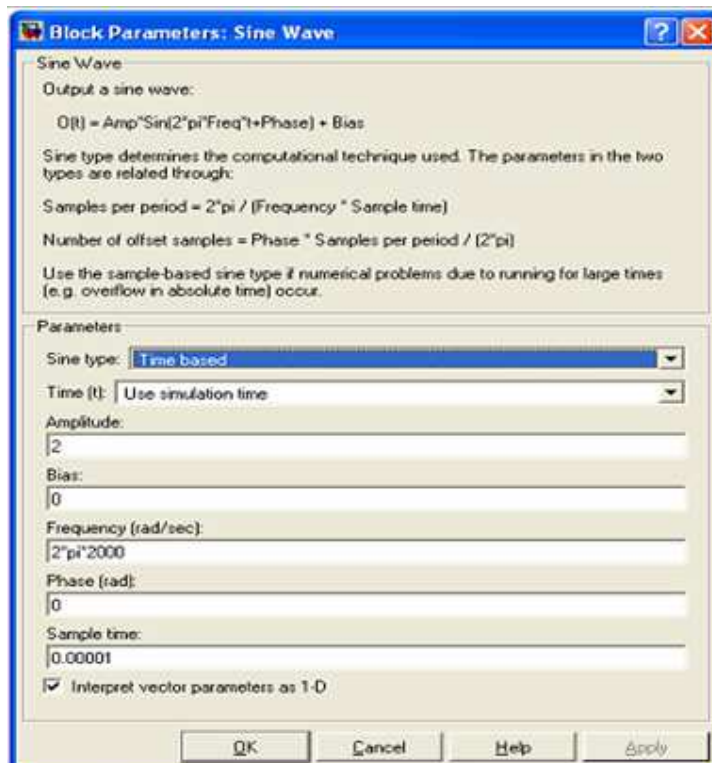


Figura 3.19. Parámetros o configuración del Sine Wave (librería).
Fuente: Print Screen del Programa MATLAB/SIMULINK

A continuación de ingresar todos los datos en la configuración se da aceptar. Luego corremos el diseño que armamos y por medio del **scope** o osciloscopio vamos a visualizar como se ve la señal en una ventana que ejecuta **SIMULINK**. (Ver figura 3.20)

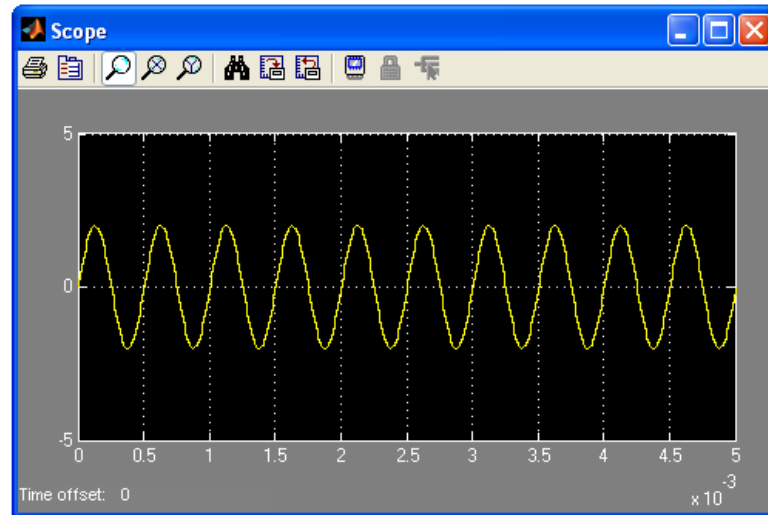


Figura 3.20. Muestra grafica de un Scope de SIMULINK.
Fuente: Print Screen del Programa MATLAB/SIMULINK

3.38 REFERENCIAS DE LIBRERÍAS DE **SIMULINK**

A continuación se van a detallar teóricamente todas las librerías que son de aplicación indispensables en los fundamentos de comunicación. Se a dejar claro el funcionamiento y concepto de cada librería que se ocupe durante el proyecto.

3.38.1 **SINKS**

Aquí mencionamos los dos tipos de **SINKS**:

- **SCOPE**
- **TO WORKSPACE**

3.38.1.1 **SCOPE**

El bloque **Scope** muestra su entrada con respecto al tiempo de simulación. El bloque **scope** puede tener varios ejes (uno por cada puerto) y todos los ejes tienen un rango de tiempo común con los independientes y los ejes. El bloque

scope le permite ajustar la cantidad de tiempo y el rango de valores de entrada que se muestra. Puede mover y redimensionar la ventana del alcance y se pueden modificar los valores del alcance de los parámetros durante la simulación. Al final, este bloque a nivel de prácticas muestra la forma de las ondas de las señales en forma grafica.

3.38.1.2 **TO WORKSPACE**

To workspace muestra el navegador de espacio, una interfaz gráfica de usuario que le permite ver y administrar el contenido del espacio de trabajo de **MATLAB**. Proporciona una representación gráfica de la pantalla **whos**, y le permite realizar el equivalente de las funciones **claras**, de **carga**, **abrir** y **guardar**.

El navegador de espacio también muestra y actualiza automáticamente los cálculos estadísticos para cada variable, que puede optar por mostrar u ocultar

3.38.2 **SOURCES**

Aquí mencionamos los dos tipos de **SOURCES**:

- **SINE WAVE**
- **PULSE GENERATOR**

3.38.2.1 **SINE WAVE**

El bloque de onda sinusoidal (ver figura 3.21) ofrece una senoide. El bloque puede funcionar tanto en modo basado en el tiempo o la muestra de base.



Figura 3.21. Grafico del sine wave.
Fuente: Print Screen del Programa MATLAB/SIMULINK

La salida del bloque de onda sinusoidal se determina por

$$y = \text{Amplitud} \times \sin(\text{frecuency} \times \text{time} + \text{phase}) + \text{bias}$$

Basada en el **tiempo modo** tiene dos submodos: modo continuo o modo discreto. El valor del parámetro de tiempo de la muestra se determina si el bloque funciona en modo continuo o modo discreto:

0 (por defecto) hace que el bloque funcione en modo continuo.

> 0, el bloque opera en modo discreto

3.38.2.1.1 USO DEL BLOQUE DE ONDA SINUSOIDAL EN MODO CONTINUO

El tiempo de la muestra en valor 0 hace que el bloque funcione en modo continuo. Cuando se opera en modo continuo, el bloque de onda sinusoidal puede ser inexacto debido a la pérdida de precisión en cuanto tiempo se hace muy grande.

3.38.2.1.2 USO DEL BLOQUE DE ONDA SINUSOIDAL EN MODO DISCRETO

El tiempo de la muestra en valores mayores que cero hace que el bloque se comporte como si se tratara de la conducción de un bloque de mantenedor de orden cero, cuya muestra es el tiempo establecido en ese valor. Utilizando el bloque de onda sinusoidal de esta manera le permite construir modelos con las fuentes de ondas sinusoidales que son puramente discretos, en lugar de modelos que son los sistemas híbridos de continuo / discreto. Los sistemas híbridos son por naturaleza más complejos y, en consecuencia toman más tiempo para simular.

El bloque de onda sinusoidal en modo discreto utiliza un algoritmo incremental en lugar de uno basado en el tiempo absoluto. Como resultado, el bloque puede ser útil en los modelos con la intención de postularse para un periodo indefinido de tiempo, como en las vibraciones o los ensayos de fatiga. El algoritmo calcula el seno incremental basado en el valor calculado en el momento de la muestra anterior. Este método hace uso de las siguientes identidades:

$$\sin(t + \Delta t) = \sin(t) \cos(\Delta t) + \sin(\Delta t) \cos(t)$$

$$\cos(t + \Delta t) = \cos(t) \cos(\Delta t) - \sin(\Delta t) \sin(t)$$

Estas identidades se pueden escribir en forma matricial:

$$\begin{bmatrix} \sin(t + \Delta t) \\ \cos(t + \Delta t) \end{bmatrix} = \begin{bmatrix} \cos(\Delta t) & \sin(\Delta t) \\ -\sin(\Delta t) & \cos(\Delta t) \end{bmatrix} \begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix}$$

Desde Δt es constante, la siguiente expresión es una constante:

$$\begin{bmatrix} \cos(\Delta t) & \sin(\Delta t) \\ -\sin(\Delta t) & \cos(\Delta t) \end{bmatrix}$$

Por lo tanto, el problema se convierte en uno de multiplicación de la matriz del valor de $\sin(t)$ por una matriz constante para obtener $\sin(t + \Delta t)$. Modo discreto reduce pero no elimina la acumulación de errores de redondeo. Esto es así porque el cálculo de la producción del bloque en cada paso de tiempo depende del valor de la salida en el paso de tiempo anterior.

3.38.2.2 PULSE GENERATOR

En la figura 3.22 se muestra el bloque generador de impulsos de ondas cuadradas a intervalos regulares. Los parámetros del bloque a considerar son: forma de onda, amplitud, ancho de pulso, de época, y el retardo de fase, para determinar la forma de la onda de salida.



Figura 3.22. Grafico del generador de pulso.
Fuente: Print Screen del Programa MATLAB/SIMULINK

En la figura 3.23 se muestra la gráfica del generador de pulso.

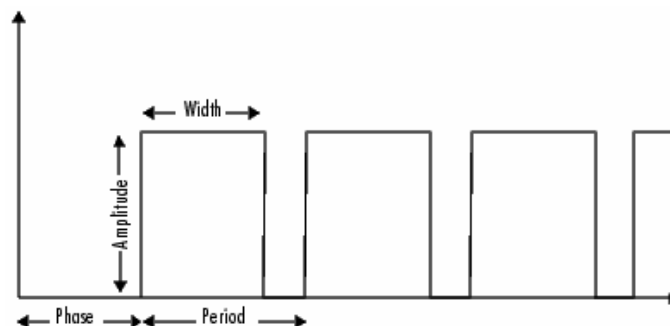


Figura 3.23. Grafico del generador de pulso.
Fuente: Print Screen del Programa MATLAB/SIMULINK

El generador de impulsos puede emitir señales de escalar, vector, o matriz de cualquier tipo de datos real. Para hacer que el bloque emita una señal escalar, usar escalares para especificar los parámetros de forma de onda. Para hacer que el bloque emita una señal vector o matriz, el uso de vectores o matrices, respectivamente, para especificar los parámetros de forma de onda.

Cada elemento de los parámetros de forma de onda afecta al elemento correspondiente de la señal de salida. Por ejemplo, el primer elemento de un parámetro de la amplitud del vector determina la amplitud del primer elemento de un pulso de salida vectorial. Todos los parámetros de forma de onda deben tener las mismas dimensiones que después de la expansión escalar. El tipo de datos de la salida es el mismo que el tipo de datos del parámetro de amplitud.

3.38.3 OPERACIONES MATEMÁTICAS

En las operaciones matemáticas las más utilizadas son:

- Producto
- Suma

3.38.3.1 PRODUCTO

La figura 3.24 es la representación grafica del bloque de **producto**. Cuando este bloque se utiliza con los parámetros por defecto, el bloque de productos muestra el resultado de la multiplicación de sus dos entradas, que pueden ser dos escalares, un escalar y no escalar. A la vez se pueden modificar las entradas de este bloque.



Figura 3.24. Grafico del PRODUCTO.
Fuente: Print Screen del Programa MATLAB/SIMULINK

3.38.3.2 SUMA

El bloque de Suma como se muestra en la figura 3.25, es aquella que realiza la operación de sumatoria o resta en sus entradas. Este bloque puede sumar o restar escalar, vector o matriz de entradas. También se pueden colapsar los elementos de una señal.

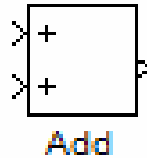


Figura 3.25. Grafico del SUMA.

Fuente: Print Screen del Programa MATLAB/SIMULINK

3.38.4 DISEÑO DE FILTRO ANALÓGICO

El diseño de filtro analógico (ver figura 3.26) permite configurar el diagrama de bloque para aplicarlos como filtro pasa alto, filtro pasa bajo y filtro pasa banda. La entrada debe ser una muestra basada en tiempo continuo, con valores reales o señal de escalar. Los parámetros que aparecen en el cuadro de diálogo varían para diferente diseño de banda de combinaciones. Sólo algunos de los parámetros que se enumeran a continuación se pueden ver en el cuadro de diálogo.

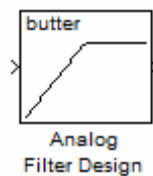


Figura 3.26. Grafico del diseño de filtro analógico.

Fuente: Print Screen del Programa MATLAB/SIMULINK

3.38.4.1 DESIGN METHOD

En este cuadro de configuración (ver figura 3.27) se puede obtener estas opciones: **Butterworth**, **Chebyshev** tipo I, **Chebyshev** tipo II o elíptica sintonizable.

3.38.4.2 FILTER TYPE

Muestra las siguientes opciones (ver figura 3.27): paso bajo, paso alto, pasó banda o banda de detención. Sintonizable.

3.38.4.3 **FILTER ORDER**

El orden del filtro, sirve para las configuraciones de paso bajo y paso alto, también para las configuraciones de banda de detención, a la orden del filtro final es el doble de este valor.

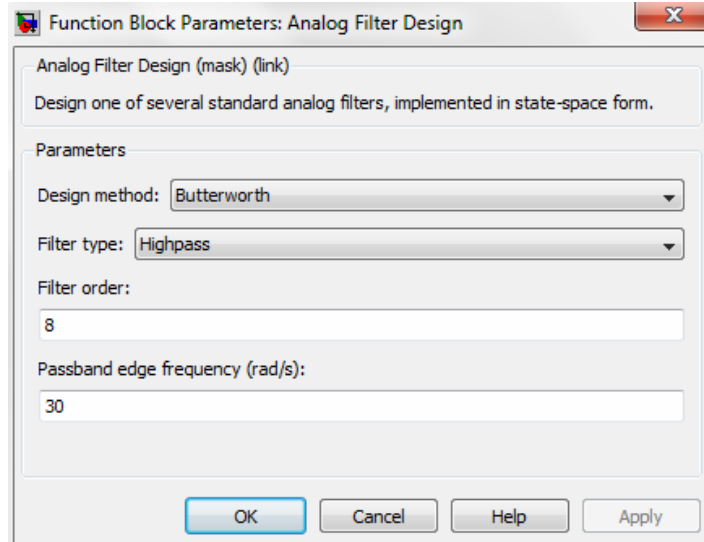


Figura 3.27. Grafico de los parámetros del filtro analógico.
Fuente: Print Screen del Programa MATLAB/SIMULINK

3.38.5 **FM MODULATOR PASSBAND**

El resultado es una representación de paso banda de la señal modulada. La frecuencia de la señal de salida varía con la amplitud de la señal de entrada. Los parámetros de este bloque se muestran en la figura 3.28 siguiente.

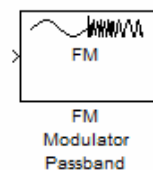


Figura 3.28. Grafico del modulador pasa banda
Fuente: Print Screen del Programa MATLAB/SIMULINK

En la figura 3.29, se muestra el grafico de los parámetros del filtro pasa banda.

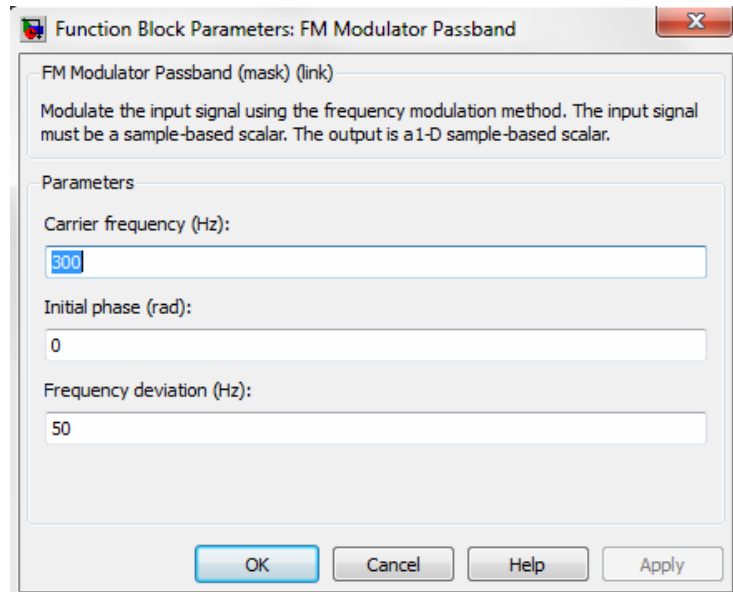


Figura 3.29. Grafico de los parámetros del filtro pasa banda
Fuente: Print Screen del Programa MATLAB/SIMULINK

3.38.6 **POWER SPECTRAL DENSITY**

Este bloque está conectado a la salida de un sistema que muestra el contenido de frecuencia de la memoria intermedia en la ventana gráfica. (Ver figura 3.30)



Figura 3.30. Grafico del power spectral density
Fuente: Print Screen del Programa MATLAB/SIMULINK

3.38.7 **TO WORKSPACE**

EL bloque TO WORKSPACE envía las variables generadas en el **SIMULINK** al **MATLAB**, esta se genera en la ventana de comandos de **Matlab**. (Ver figura 3.31 y 3.32).

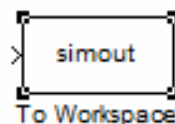


Figura 3.31. Grafico del to Workspace
Fuente: Print Screen del Programa MATLAB/SIMULINK

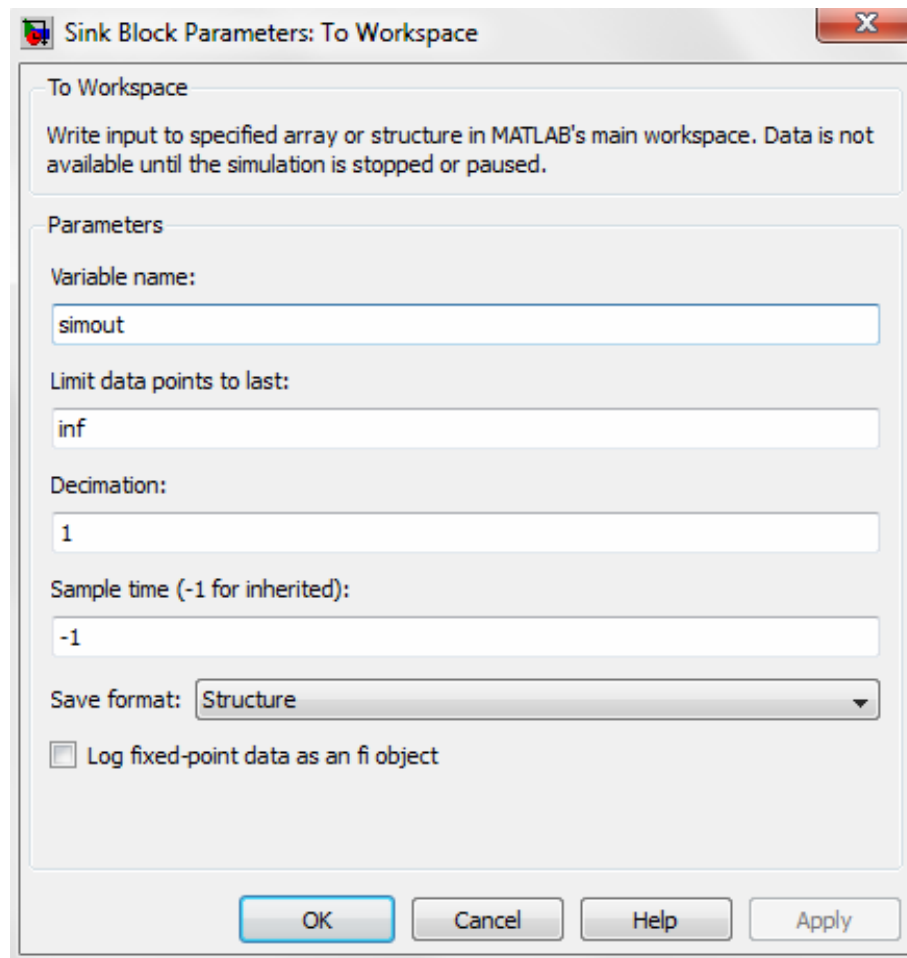


Figura 3.32. Grafico de los parámetros del to workspace.
Fuente: Print Screen del Programa MATLAB/SIMULINK

CAPITULO 4 APLICACIÓN DEL SOFTWARE *MATLAB* / *SIMULINK* EN LA ASIGNATURA DE FUNDAMENTOS DE COMUNICACIÓN.

A continuación se detallará el programa y el syllabus de la asignatura Fundamentos de Comunicación, en la cual se incluirá el uso de las simulaciones en *Matlab/Simulink* a través de prácticas que permitirá a los estudiantes un mejor entendimiento de la teoría.

4.1. PROGRAMA ACTUAL DE LA ASIGNATURA

Diseño de las unidades de estudio (UE)

UE I: Introducción a las comunicaciones electrónicas

Contenidos específicos: Introducción. Modulación y demodulación. El espectro electromagnético. Frecuencias de transmisión. Ancho de banda y capacidad de información. Modos de transmisión. Ruido eléctrico.

UE II: Modulación de Amplitud.

Contenido Específico: Modulación de amplitud. La envolvente de AM. Espectro de frecuencia de AM. Ancho de banda. Representación fasorial de una onda AM. Coeficiente de modulación y porcentaje de modulación. Distribución de voltaje AM. Análisis de AM en el dominio del tiempo. Distribución de la potencia de AM. Cálculo de corriente de AM. Modulación por medio de una señal de información compleja. Recepción de AM. Parámetros del receptor. Sistemas de banda lateral única. Análisis matemático de la AM con portadora suprimida. Generación de banda lateral única.

UE III: Modulación Angular

Contenido específico: Modulación angular. Análisis matemático. Formas de onda FM y PM. Desviación de fase, índice de modulación y desviación de frecuencia. Porcentaje de modulación. Moduladores y demoduladores de fase y de frecuencia. Análisis de frecuencia de las ondas con modulación angular. Requerimientos de ancho de banda. Relación de desviación. Potencia

promedio de una onda con modulación angular. Modulación angular y ruido. Preénfasis y De énfasis.

UE IV: Modulación de pulsos

Contenido específico: Transmisión digital. Ventajas y desventajas. Modulación de pulsos. Modulación de pulsos codificados. Tasa de muestreo. Rango dinámico. Eficiencia de codificación. Relación señal a ruido de cuantización. Códigos PCM lineales contra no lineales. Ruido del canal inactivo. Métodos de codificación. Compansión. Compansión analógica y digital.

4.2. SYLLABUS DE LA ASIGNATURA

El **Syllabus** es la programación detallada de la gestión del día en el aula de clase.

Tabla 4.1: Tabla **Syllabus** actual de la asignatura de Fundamentos de comunicación:

UE	Objetivo	Sistema de Clases	No. De Horas	Evaluación	Observaciones
I	Introducir al estudiante a los conceptos fundamentales de los sistemas de comunicaciones electrónicas y explicar la terminología básica necesaria para entender los temas relacionados con las telecomunicaciones	C1.- Introducción. Historia. Modulación y demodulación.	2		
		C2.- El espectro electromagnético. Frecuencias de transmisión.	2		
		C3.- Ancho de banda y capacidad de información.	2		
		C4.- Modos de transmisión.	2		

UE	Objetivo	Sistema de Clases	No. De Horas	Evaluación	Observaciones
		C5.- Ruido eléctrico: no correlacionado, externo	2		
		C6.- Ruido eléctrico no correlacionado, interno	2		
		C7.- Ruido eléctrico correlacionado	2		
		CP1.- Ejercicios de aplicación	2	Desarrollo individual	
II	Introducir al estudiante a los conceptos fundamentales de la transmisión AM, describir algunos de los circuitos usados en los moduladores AM y describir los tipos diferentes de transmisores AM.	C8.- Modulación de amplitud. La envolvente de AM. Espectro de frecuencia de AM. Ancho de banda.	2		
		C9.- Coeficiente de modulación y porcentaje de modulación.	2		
		C10.- Distribución de voltaje AM. Análisis de AM en el dominio del tiempo.	2		
		C11.- Distribución de la potencia de AM. Recepción de AM. Parámetros del receptor.	2		

UE	Objetivo	Sistema de Clases	No. De Horas	Evaluación	Observaciones
		C12.- Sistemas de banda lateral única.	2		
		C13.- Análisis matemático de la AM con portadora suprimida.	2		
		C14.- Generación de banda lateral única.	2		
		CP2.- Ejercicios de aplicación	2	Desarrollo individual	
III	Describir los conceptos básicos de la modulación en frecuencia y en fase y como se relacionan uno con otro. Mostrar algunos de los circuitos más usados comúnmente para producir las ondas con modulación angular y comparar el rendimiento de la modulación angular con la modulación en amplitud.	C15- Modulación angular. Análisis matemático	2		
		C16- Formas de onda FM y PM. Desviación de fase, índice de modulación y desviación de frecuencia. Porcentaje de modulación	2		
		C17- Moduladores y demoduladores de fase y de frecuencia. Análisis de frecuencia de las ondas con modulación angular.	2		

UE	Objetivo	Sistema de Clases	No. De Horas	Evaluación	Observaciones
		C18- Requerimientos de ancho de banda. Relación de desviación.	2		
		C19.- Potencia promedio de una onda con modulación angular.	2		
		C20.- Modulación angular y ruido.	2		
		C21.- Preenfasis y Deenfasis	2		
		CP3- Ejercicios de aplicación.	2	Desarrollo individual	
IV	Describir la estructura básica de los sistemas de comunicación digital y sus funciones. Describir y evaluar las transformaciones a ejecutar sobre la señal para su transmisión en forma de señal digital, es decir la codificación digital de las señales de	C22- Transmisión digital. Ventajas y desventajas	2		
		C23.- Modulación de pulsos. Modulación de pulsos codificados.	2		
		C24.- Tasa de muestreo. Rango dinámico	2		
		C25.- Eficiencia de codificación. Relación señal a ruido de cuantización	2		

UE	Objetivo	Sistema de Clases	No. De Horas	Evaluación	Observaciones
	naturaleza análoga	C26.- Códigos PCM lineales contra no lineales. Ruido del canal inactivo	2		
		C27.- Métodos de codificación. Compansión	2		
		C28.- Compansión analógica y digital	2		
		CP4- Ejercicios de aplicación.	2		

4.3 PRÁCTICAS RECOMENDADAS

Para la correcta instalación y evitar cualquier problema se los requerimientos básicos son:

- Computador **Intel Pentium** IV, 512 MB RAM (Memória de Acesso Aleatório), Mínimo 2 Gb de espacio libre en el disco duro, **Windows** XP o Vista.
- Instalador de programa **MATLAB** 7.8.0(R2009a).

A continuación se plantearan varias practicas que se realizaron a lo largo de esta investigación, recomendando se incluyan al programa de estudio de la materia de Fundamentos de Comunicaciones.

4.3.1 DESARROLLO EXPERIMENTAL 1: GENERACIÓN DE UNA SEÑAL SINUSOIDAL

Las señales sinusoidales específicamente son señales analógicas debido a que poseen valores infinitos entre dos puntos. Las señales sinusoidales son caracterizadas por su amplitud, frecuencia, fase y periodo. Estas cuatro características son aquellas que la caracterizan a una señal sinusoidal, dado que la amplitud es el máximo voltaje que la señal puede tener, la frecuencia se refiere a la cantidad de veces que se repite un ciclo en un segundo de tiempo (medida en HZ), la fase se identifica como el ángulo de fase inicial (medido en radianes) y el periodo es el trayecto de tiempo que se demora en completar un ciclo.

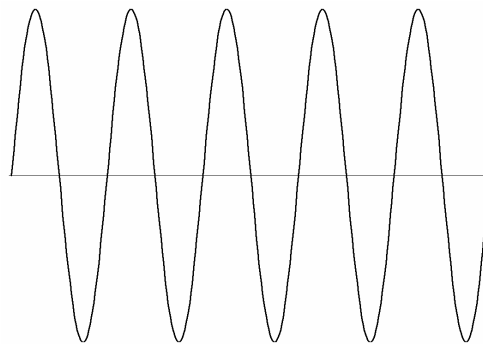


Figura 4.1: Señal Sinusoidal
Fuente: Los Autores

Desarrollo de la práctica.

En esta práctica se procederá básicamente al análisis de una señal sinusoidal utilizando como material didáctico el **Simulink**, así también como la descripción de librerías utilizadas en el proceso de la práctica.

Librerías: se utilizarán las siguientes:

- **Sources**
 - **Sine Wave**
- **Sinks**
 - **Scope**

La disposición de los elementos citados anteriormente debe ser la misma que se detalla en las figuras 4.2 y 4.3.

Señal Sinusoidal



Figura 4.2: Señal Sinusoidal

Fuente: Resultado de la práctica en Simulink en MATLAB

Parámetros

Amplitud: 3.8

Frecuencia: 2

Offset: 0 V

Desfase: 0 radianes

Tiempo de muestreo: 0.00001

Tiempo de simulación: 8 ms (Tiempo de parada)

Sine Wave1 (Modulador de la señal analógica)

Amplitud: 3.8

Bias: 0

Frequency: $2 \cdot \pi \cdot 2000$

Phase: 0

Sample Time: 0.00001

Señal de salida

Señal Sinusoidal

En la figura 4.3 muestra la señal de salida obtenida de la configuración de la figura 4.2:

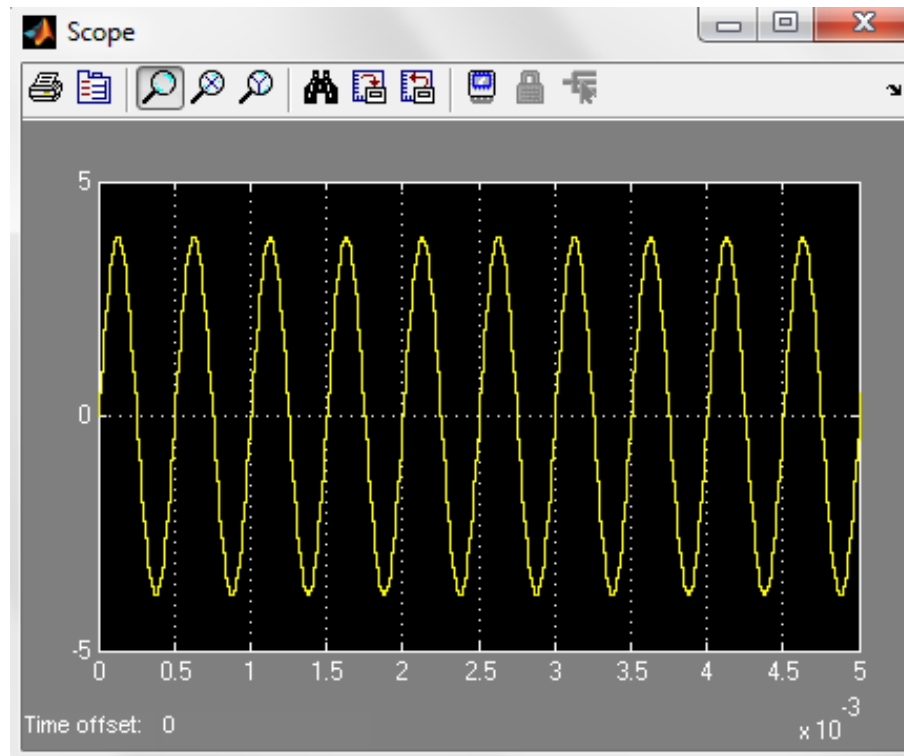


Figura 4.3: Visualización de señal sinusoidal
Fuente: Resultado de la práctica en Simulink de MATLAB

En la figura 4.3 muestra al osciloscopio que permite visualizar gráficamente una señal sinusoidal, en nuestra práctica la frecuencia es más baja por lo tanto la senoide no es una muestra periódica pero la representación grafica es la misma.

4.3.2 PRÁCTICA # 2: MODULACIÓN DE AMPLITUD (AM) EN SIMULINK

La modulación de amplitud se puede definir como modulación lineal cuyo procedimiento consiste en hacer variar la amplitud de la señal portadora, de tal manera que esta refleje un cambio en el momento que la señal moduladora presente variaciones de niveles que seria los datos que se transmitirían.

Desarrollo de la práctica.

En esta práctica se procederá al análisis del circuito con el cual se genera la modulación de amplitud de una señal, así también como la descripción de librerías utilizadas de la aplicación **Simulink** de **Matlab**:

Librerías: se utilizarán las siguientes:

- **Sources**
 - **Sine Wave**
- **Match Operations**
 - **Products**
 - **Sum**
- **Sinks**
 - **Scope**

La disposición de los elementos citados anteriormente debe ser la misma que se detalla en las figuras 4.1 y 4.2.

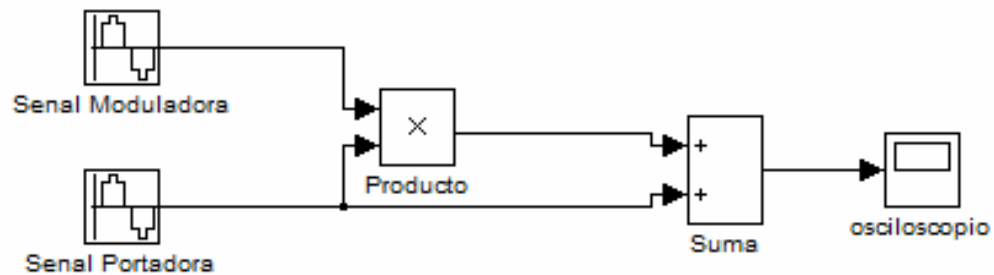


Figura 4.4: Modulación AM
Fuente: Resultado de la práctica en Simulink en MATLAB

Parámetros

Sine Wave1 (Señal moduladora)

Amplitud: 0.5

Bias: 0

Frecuency: 1

Phase: 0

Sample Time: 0.1

Sine Wave2 (Señal portadora)

Amplitud: 2

Bias: 0

Frecuency: 10

Phase: 0

Sample Time: 0.01

Producto

Number of inputs: 2

Multiplication: Element-wise(.*)

Sample time: -1

Suma

Lists of signs: ++

Sample Time: -1

Señal de salida

Modulación de Amplitud

En la figura 4.5 se muestra la señal de salida del circuito modulador de amplitud

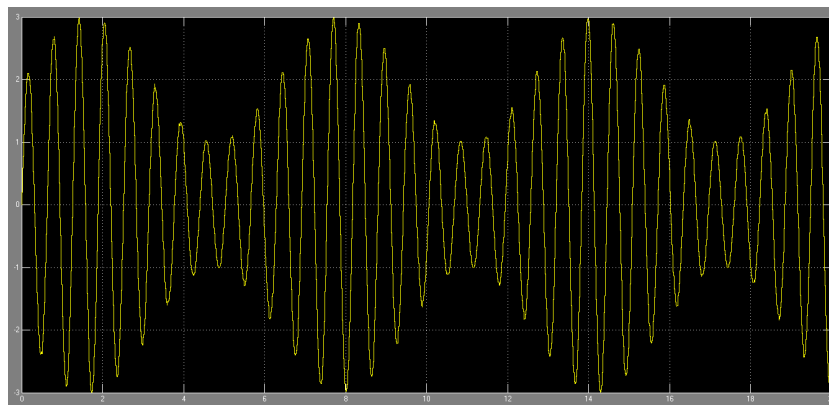


Figura 4.5: Visualización de señal modulada por amplitud
Fuente: Resultado de la práctica en Simulink de MATLAB

En la grafica que desplego el simulador del osciloscopio se visualiza como es representada la onda modulada, en AM la frecuencia no varía con la modulación, lo único que varía es la amplitud. La portadora modulada en el grafico vendría a ser toda la secuencia de la onda y la onda envolvente visualmente en la grafica es la sinusoide que se forma.

4.2.3 PRÁCTICA # 3: SEÑAL AM CON MODULACION DE TONO

Para la señal AM con modulación de tono, debemos considerar el análisis del dominio del tiempo, cuya señal de portadora es $x_p(t) = A_c \cos(\omega_c t) = A_c \cos(2 \pi f_c t)$, donde la función $x(t)$ es la señal moduladora y viene dada por $x_c(t) = A_c [1 + mx(t)] \cos(\omega_c t)$. Asimismo el tono modulador ilustrado por la figura 4.6 viene dado por $x(t) = A_m \cos(2 \pi f_m t + \phi)$, considerando que:

$$A_m \leq 1$$
$$f_m \leq W$$

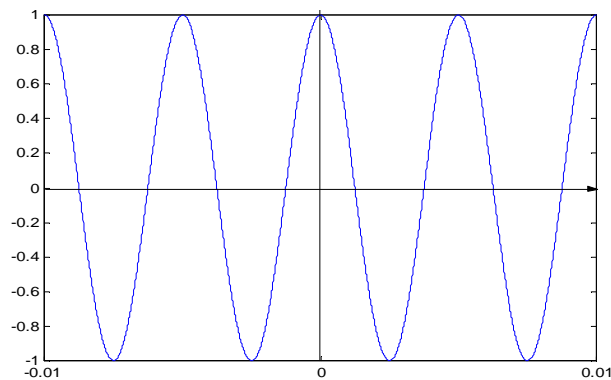


Figura 4.6 Tono modulador

Desarrollo de la práctica.

En esta práctica se procederá al análisis del circuito con el cual se genera una señal AM con modulación de tono, adicional se mostrará la programación de **Matlab** aplicando específicamente los principios de Fundamentos de comunicaciones.

Librerías: Se utilizarán las siguientes.

- **Plot**
- **Title**

- ***Xlabel***
- ***Ylabel***

Parámetros de la modulación:

$A_c = 1 \text{ V}$

$A_m = 1 \text{ V}$

$m = 0.9$ (90 %)

$f_m = 80 \text{ HZ}$

$f_c = 1800 \text{ HZ}$

Tiempo simulación = 30 ms

Script de *Matlab*

```
% Señal de AM con modulación de tono
% m: Índice de modulación
% ap: Amplitud de la portadora
% am: Amplitud de la moduladora
% fm: Ffrecuencia del tono modulador
% fc: Ffrecuencia de la portadora
t=0:0.0000001:0.03;
m=0.9;
AP=1;
AM=1;
fm=80;
fc=1800;
MP=(AP+AP*m*AM*cos(2*pi*fm*t)).*cos(2*pi*fc*t);
plot(t,MP)
title('Señal AM con modulación de tono')
xlabel('t (s)')
ylabel('Amplitud(v)')
```

En la figura 4.7 se muestra gráficamente mediante el **Scope** de **Matlab** la señal de la amplitud modulada con una modulación de tono.

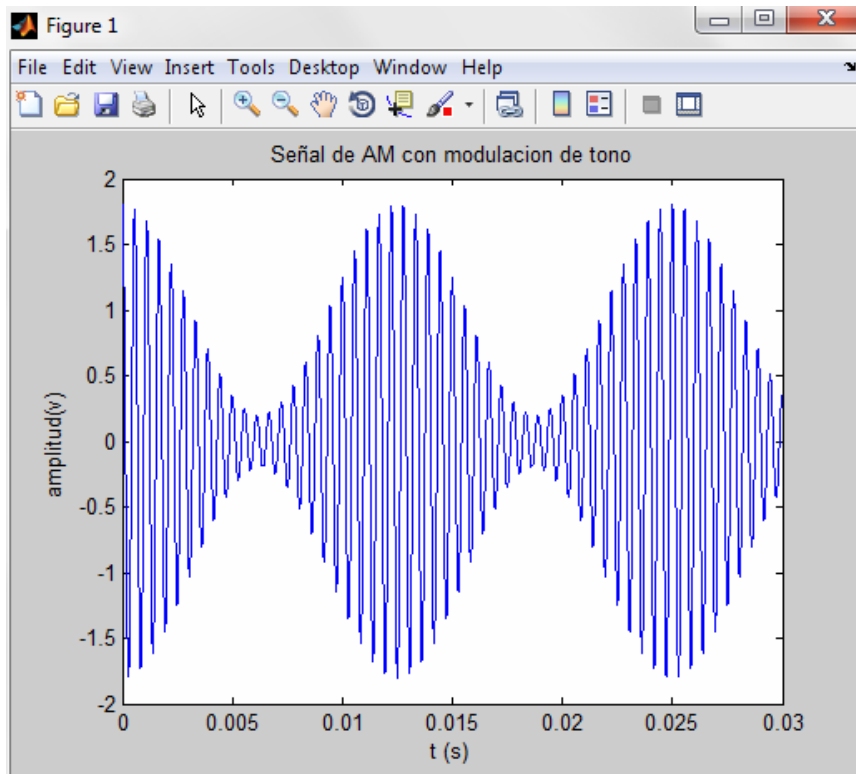


Figura 4.7: Grafico de la señal AM con modulación de tono
Fuente: Resultado de la práctica en MATLAB

Para concluir en la figura 4.7 se observa gráficamente la señal de AM obtenida con una modulación de tono, tomando en cuenta que la frecuencia del tono modulador siempre tiene que ser mucho más bajo que la frecuencia de la portadora y el índice de la modulación que vendría a ser m es el punto más importante y mide el grado de la profundidad de la modulación.

4.3.4 PRÁCTICA # 4: MODULACION DE DOBLE BANDA LATERAL CON PORTADORA SUPRIMIDA (DBLPS).

Para el caso de la modulación de doble banda lateral con portadora suprimida y de acuerdo a la ley de modulación, esta parte de la señal $x_c(t) = Kx(t)$, en la cual el factor de multiplicación del modulador vendría a ser la constante $K = K_m A_c$. En la mayoría de los casos $K_m=1$ y $A_c=1$.

La señal modulada está dada por $x_c(t) = Kx(t)\cos(\omega_c t)$, donde señal modulada es cero en los casos cuando $x(t)$ también es igual a 0. El espectro sale aplicando directamente el teorema de modulación dado por

$$X_c(f) = \frac{K}{2} [X(f + f_c) + X(f - f_c)]$$

Desarrollo de la práctica.

En esta práctica se procederá al análisis de una señal de DBLPS utilizando un tono como señal moduladora, la práctica se desarrollara específicamente en **MATLAB**, detallando cada aspecto de la programación en todo lo largo del ejercicio.

Librerías: se utilizarán las siguientes:

- **Plot**
- **Title**
- **Xlabel**
- **Ylabel**

La disposición de los elementos citados anteriormente debe ser la misma que se detalla en las figuras 4.7.

Parámetros

t=0:0.0000001:0.01;

Am=2;

K=5;

Ac=2;

fm=125;

fc=4500;

Script de Matlab

%Am:amplitud de la moduladora

```

%fm:frecuencia del tono modulador
%fc:frecuencia de la portadora
t=0:0.0000001:0.01;
Am=2;
K=5;
Ac=2;
fm=125;
fc=4500;
Xc=K*((Am*cos(2*pi*fm*t)).*(Ac*cos(2*pi*fc*t)));
plot(t,Xc)
title('Señal de DBLPS con un tono')
xlabel('tiempo (s) ')
ylabel('amplitud(v) ')

```

En la figura 4.8 se muestra gráficamente mediante el **Scope** de **Matlab** la señal de DBLPS con un tono.

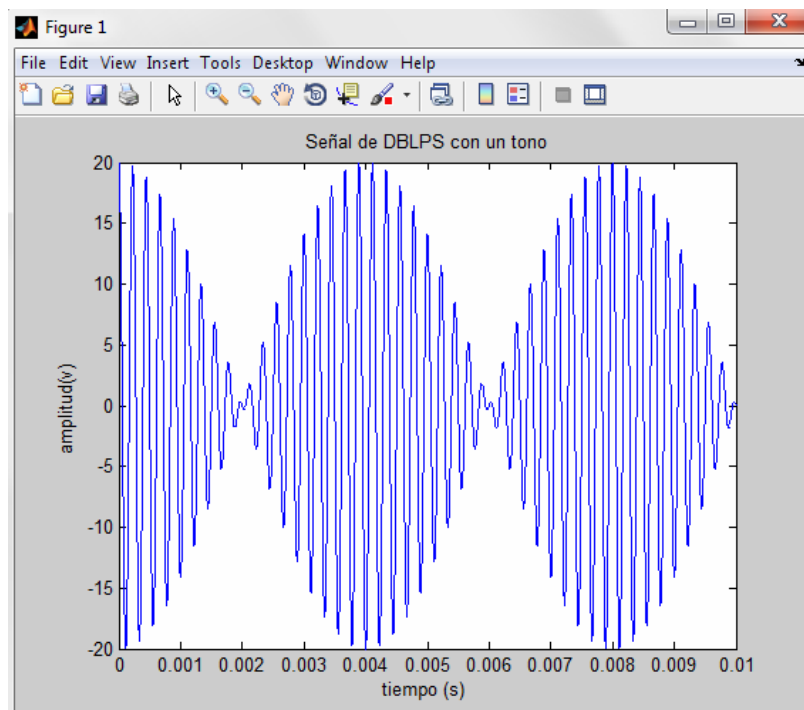


Figura 4.8: Gráfico de la señal de DBLPS con un tono
Fuente: Resultado de la práctica en MATLAB

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- Que mediante el adiestramiento teórico y práctico del software **MATLAB/SIMULINK**, el estudiante estará en capacidad de desarrollar sus habilidades y destrezas en su contexto profesional.
- Durante el desarrollo de enseñanza-aprendizaje del software **MATLAB/SIMULINK**, específicamente el educando opta en asimilar los principales objetivos cognitivos y procedimentales con el fin de alcanzar una optimización en el campo profesional.
- Cumpliendo el adiestramiento semestral de esta área técnica, el estudiante luego de haber intensificado el análisis, diferencias, semejanzas, ventajas y desventajas del software **MATLAB/SIMULINK**, estará en capacidad no solamente de crear o de incitar uno o dos cambios sino que tendrá el conocimiento y la motivación de propender a desarrollar transformaciones siempre que tengan el objetivo de mejorar, aprovechar y ampliar la intuición humana para dejar un legado a nuestra futura generación.
- Al culminar los nueve semestres efectivo, activo y motivacional de estudio el estudiante egresado estará en capacidad dentro de la correspondiente ética profesional en elaborar el respectivo manual de prácticas que sirva para simular las aplicaciones en el campo de la comunicación a través del software **MATLAB/SIMULINK**.

5.2 RECOMENDACIONES

- Impartir en forma técnica los conocimientos teóricos y prácticos para satisfacer los aprendizajes basados en los contenidos innovadores en la asignatura fundamentos de comunicación.
- Revisar los contenidos de las asignaturas técnicas que están incluida en el pensum de esta carrera, ya que es importante implementar estrategias innovadoras como el software MATLAB.
- Que los docentes de esta carrera deben generar espacios mayores a través del conocimiento de los contenidos de la asignatura de fundamento de comunicación para fortalecer y enriquecer la formación académica y obtener resultados positivos en los futuros profesionales.

5.3 GLOSARIO

- **AC.-** (*Alternating Current*), corriente alterna.
- **DC.-** (*Direct Current*), corriente directa.
- **AM.-** (*Modulation amplitude*), Amplitud modulada.
- **FM.-** (*Frecuency modulation*), Frecuencia modulada.
- **RF.-** (*Radio Frecuency*), Radio frecuencia.
- **PDM.-** (*Pulse duration modulation*), Modulaci3n de duraci3n de pulso.
- **PLM.-** (*Pulse length modulation*), Modulaci3n por longitud de pulso.
- **PSK.-** (*Phase shift keying*), Modulaci3n por desplazamiento de fase.
- **DIODO.-** Componente electr3nico.
- **BJT.-** (*Bipolar Junction transistor*), Transistor de uni3n bipolar es un Dispositivo electr3nico.
- **FET.-** (*Field effect transistor*), transistor de efecto de campo.
- **FORTTRAN:** lenguaje de programaci3n alto nivel.
- **AMD.-** (*Advanced Micro Devices*),micro equipos avanzados.
- **USB:** Bus universal en serie.
- **PC.-** (*Personal Computer*), Computadora Personal u Ordenador Personal.
- **HTML.-** (*Hypertext markup language*), Lenguaje de marcado de hipertexto.
- **ASCII.-** (*American standard code for information interchange*), C3digo est3ndar Estadounidense para el intercambio de informaci3n.
- **UCSG.-** Universidad Cat3lica de Santiago de Guayaquil.

5.2 BIBLIOGRAFIA

Wayne Tomasi. *Sistemas de Comunicaciones Electrónicas Segunda edición (2006)*. Recuperado el 16 de septiembre del 2011

William Stallings. *Comunicaciones y Redes de Computadores Quinta (2008)*. Recuperado el 05 de septiembre del 2011

Amos Gilat. *Matlab (2009)*, Recuperado el 28 de septiembre del 2011:
www.books.google.com/books?id=dAvjPaUFdAMC&printsec=frontcover&dq=MATLAB&hl=es&ei=m2q-TYHrPOfd0QH11MzPBQ&sa=X&oi=book_result&ct=result&resnum=1&ved=0CkQ6AEwAA#v=onepage&q&f=false

Luis Montoto. *Fundamentos físicos de la información y las comunicaciones (2004)*. Recuperado el 05 de octubre del 2011:
www.books.google.com/books?id=45HN_RhzdkYC&printsec=frontcover&dq=fundamentos+de+comunicaciones&hl=es&ei=MCK-TeWzFtHr0QGdq4nbBQ&sa=X&oi=book_result&ct=result&resnum=3&ved=0CDoQ6AEwAg#v=onepage&q&f=false

Enrique Sanchis. *Fundamentos y Electrónica de las comunicaciones (2004)*. Recuperado el 10 de octubre del 2011:
www.books.google.com/books?id=kZKsps2_F4YC&printsec=frontcover&dq=fundamentos+de+comunicaciones&hl=es&ei=MCK-TeWzFtHr0QGdq4nbBQ&sa=X&oi=book_result&ct=result&resnum=1&ved=0CDAQ6AEwAA#v=onepage&q&f=false

Modulación (2011). Recuperado el 08 de septiembre del 2011, de wikipedia.org:
http://es.wikipedia.org/wiki/Modulaci%C3%B3n_%28telecomunicaci%C3%B3n%29

Modos de transmisión (2006). Recuperado el 19 noviembre del 2011:
<http://www.scribd.com/doc/41793602/5/Modos-de-transmision-Simplex-half-duplex-y-full-duplex>

Comunicación Serial (2008). Recuperado el 19 de noviembre del 2011:
http://galia.fc.uaslp.mx/~cantocar/microprocesadores/EL_Z80_PDF_S/21_UAR_T_1.PDF

Introducción a matlab y simulink (2006/2007). Recuperado el 15 de noviembre del 2011: http://www.esi2.us.es/~fabio/apuntes_matlab.pdf