

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Telecomunicaciones



TRABAJO DE DIPLOMA

Radio Definido por Software usando MATLAB

Autor: Carlos Manuel García Algora

Tutor: Dr. Vitalio Alfonso Reguera

Santa Clara

2011

"Año 53 de la Revolución"

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Telecomunicaciones



TRABAJO DE DIPLOMA

Radio Definido por Software usando MATLAB

Autor: Carlos Manuel García Algora

E-mail: cgalgora@uclv.edu.cu

Tutor: Dr. Vitalio Alfonso Reguera

Departamento de Telecomunicaciones

Facultad de Ingeniería Eléctrica

E-mail: vitalio@uclv.edu.cu

Santa Clara

2011

"Año 53 de la Revolución"

Hago constar que el presente trabajo fue realizado en la Universidad Central “Marta Abreu” de las Villas como parte de la culminación de los estudios de la especialidad de Telecomunicaciones y Electrónica autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

Firma del Autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del Tutor

Firma del Jefe de Dpto
donde se defiende el trabajo

Firma del Responsable de
Información Científico- Técnica

A mi familia, en especial a mis padres.

A mis amigos.

AGRADECIMIENTOS

Agradecimientos a todos los que, de una forma u otra, me han ayudado en la realización de este trabajo.

Agradecimientos especiales a:

- Mi madre, Ana Esther Algora, por el cariño, el ejemplo y la inspiración que representa.
- Mi padre, José Manuel García, por su ejemplo y su apoyo en todo momento, sin importar las circunstancias.
- Mi hermano y el resto de mi familia, por su apoyo constante.
- Mi tutor, Dr. Vitalio Alfonso Reguera, por los conocimientos y el ejemplo transmitidos, y la confianza depositada en mí.
- A los profesores por la enseñanza de sus conocimientos y su ejemplo de profesionalidad.
- Mis compañeros de aula por la ayuda prestada, en especial a Maddiel, Leiber, Heidy y Dayana.
- Jorge García, Arián, Raimel, Eduardo y Normita por su invaluable ayuda con el acceso a Internet.

Muchas gracias a mis amigos, dondequiera que estén, porque sin ellos no habría llegado hasta aquí; y a mis madres adoptivas por tantas horas de paciencia.

RESUMEN

El auge de la tecnología de Radio Definido por Software en los últimos años ha conducido a la necesidad de su estudio por parte de profesionales de las comunicaciones, incluidos aquellos que se desempeñan en el ambiente académico. Los profesionales de esta rama en Cuba, aun sin ser la excepción, no han desarrollado a profundidad el necesario estudio e implementación de esta tecnología. Además, los estudiantes de Ingeniería en Telecomunicaciones y Electrónica no reciben este contenido como parte de su plan de estudio. La intención de incluir SDR en la asignatura de Fundamentos de las Comunicaciones requiere implementaciones prácticas de SDR para utilizarlas en laboratorios de simulación de la materia.

Este trabajo desarrolla una explicación teórica de SDR incluyendo el estado del arte de esta tecnología y sus aplicaciones en los sistemas de telecomunicaciones actuales a nivel mundial. En este informe se explica el desarrollo de un sistema de comunicación digital transmisor-receptor usando MATLAB. Se incluyen explicaciones acerca de los diferentes algoritmos utilizados para la realización de las funciones de un radio digital a base de software. Los códigos de las funciones implementadas son utilizados como medio para la descripción del sistema y se incluyen íntegramente en los anexos. También se describen las pruebas realizadas para verificar el desempeño del sistema y se muestran los resultados de las mismas.

ÍNDICE

INTRODUCCIÓN	1
Estructura del Trabajo	2
CAPÍTULO 1. RADIO DEFINIDO POR SOFTWARE.....	4
1.1 Estructura de un Sistema SDR	4
1.2 Concepto de Radio Definido por Software	5
1.2.1 Niveles de SDR	6
1.3 Aplicaciones Prácticas de SDR.....	8
1.3.1 Tecnologías de Internet Inalámbrica y Telefonía Celular	8
1.3.2 Radio Cognitivo	11
1.3.3 Industria Automotriz	13
1.3.4 SDR Implementados.....	15
1.4 Estudio Académico de SDR.....	17
CAPÍTULO 2. IMPLEMENTACIÓN DE SDR USANDO MATLAB	19
2.1 Transmisor	19
2.1.2 Formato	20
2.1.3 Codificación de Canal	20
2.1.4 Sincronismo de Trama.....	23
2.1.5 Formación del Pulso	24
2.1.6 Modulación.....	28
2.1.7 Espectro Extendido.....	30
2.2 Receptor	34
2.2.1 Conversión Análogo-Digital	34
2.2.2 Sincronismo de Portadora	35
2.2.3 Espectro Extendido por Secuencia Directa	39
2.2.4 Demodulación	40
2.2.5 Filtro Acoplado.....	40
2.2.6 Sincronismo de Símbolo	41
2.2.7 Decodificación de Canal	43
CAPÍTULO 3. PRUEBAS EXPERIMENTALES Y RESULTADOS.....	47
3.1 Probabilidad de Error de Bit	47

3.2 Lazo de Costas	49
3.3 Algoritmo <i>Early/Late Gate</i>	51
3.4 Transmisión real de datos.....	52
3.4.1 Conexión directa entre interfaces de sonido.....	53
3.4.2 Audiofrecuencias	54
3.4.3 Transmisión de FM.....	55
CONCLUSIONES	57
REFERENCIAS BIBLIOGRÁFICAS.....	58
GLOSARIO.....	60
ANEXO	63
Código Transmisor.....	63
Código Receptor.....	64

INTRODUCCIÓN

Por el avance logrado en los últimos años en el desarrollo de la tecnología de Radio Definido por Software (SDR), llegando hasta la implementación práctica de sistemas de comunicaciones basados en el mismo, así como la expansión de su alcance en el terreno profesional y comercial; se hace necesario incorporar este importante tema de las telecomunicaciones del siglo XXI a los programas de enseñanza de la carrera de Ingeniería en Telecomunicaciones y Electrónica.

Sin embargo, hasta el momento la tecnología SDR no se incluye en los programas de estudio de la carrera. Este principio se ha desarrollado en los últimos años como iniciativa, en su fase inicial, de radioaficionados y desarrolladores individuales de la electrónica y las radiocomunicaciones; y posteriormente por empresas de renombre en el campo de las telecomunicaciones.[2-11]

La expansión de la tecnología SDR en la creación de sistemas de radio le debe mucho al desarrollo y abaratamiento consecuente de los microprocesadores, pues ha permitido el empleo de procesadores con buenas prestaciones en dispositivos de uso común como celulares, PDAs, computadoras portátiles, sistemas de navegación y tráfico, etc. Este empleo extensivo de procesadores potentes y baratos, ha posibilitado el surgimiento de la idea de sustituir parte de la circuitería de los dispositivos de radio por programas que realicen las mismas funciones pero a nivel de software.[12]

Actualmente se han desarrollado transmisores y receptores bajo este principio por parte de aficionados, entidades del sector de las comunicaciones y empresas de otras ramas con especial interés en brindar servicios vía radio. Estos sistemas han puesto en el mercado una variante barata, flexible y con amplias posibilidades de desarrollo en oposición a la tradicional forma de construir los sistemas de radio.

A su vez, por la importancia y el desarrollo que está teniendo SDR a nivel mundial, se hace indispensable para los graduados de Telecomunicaciones conocer dicha tecnología y estudiar el funcionamiento de sistemas de radio basados en ésta. La Facultad de Ingeniería Eléctrica pretende mantener la competitividad de sus graduados, así como formar profesionales adaptados a los avances de la tecnología a nivel mundial. Para ello se hace muy necesario poner a disposición de los estudiantes herramientas que le

permitan desarrollar el estudio de SDR, pues esta tecnología se avizora como el futuro de las radiocomunicaciones.

El estudio en Cuba del principio de SDR es muy escaso, principalmente por ser una tecnología emergente en el mercado. Algunos profesionales del ramo de las telecomunicaciones, en especial en el ambiente académico, se han interesado en el tema. Sin embargo, los estudiantes de estas especialidades no tienen acercamiento a SDR durante sus estudios de pregrado, a pesar de la vital importancia que tiene expandir el conocimiento al respecto en aras de una mejor formación de nuestros ingenieros. Los profesionales de las telecomunicaciones en Cuba tampoco están cercanos a estos temas y necesitan materiales de consulta e implementaciones prácticas de los mismos para su superación.

Este trabajo se orienta hacia la realización de un estudio teórico del SDR, la implementación práctica de funciones para construir sistemas de comunicaciones basados en este principio y la elaboración de prácticas de laboratorios que sirvan de apoyo para la enseñanza de este contenido en la carrera de Ingeniería en Telecomunicaciones y Electrónica, en la asignatura de Fundamentos de las Comunicaciones.

El objetivo general del trabajo es analizar desde el punto de vista teórico y práctico el SDR e implementar funciones en MATLAB para su uso con fines docentes en la Facultad de Ingeniería Eléctrica de la Universidad Central de Las Villas.

Los objetivos específicos del trabajo de diploma son implementar funciones en MATLAB de transmisores y receptores basados en el principio de SDR usando diferentes técnicas de modulación; y probar experimentalmente los transmisores y receptores creados, usando la interfaz de la tarjeta de audio de la computadora, y análisis de los resultados de éstas.

Estructura del Trabajo

El informe de la investigación consta de introducción, capitulario, conclusiones, recomendaciones, referencias bibliográficas, glosario y anexos. En el Capítulo 1 se realiza la descripción de la tecnología SDR, necesidad de estudio y ventajas de la utilización de SDR. El Capítulo 2 aborda la implementación en MATLAB de funciones para desarrollar transmisores y receptores basados en SDR. Y el Capítulo 3 resume los

resultados de la experimentación usando las funciones creadas como parte del Capítulo 2 y su análisis consecuente.

CAPÍTULO 1. RADIO DEFINIDO POR SOFTWARE

En este capítulo se abordan los aspectos teóricos esenciales de SDR tales como concepto, estructura de un sistema SDR, aplicaciones prácticas e implementación real, y la importancia, necesidad y viabilidad de su estudio académico. Además, se demuestra su importancia en los sistemas de comunicaciones por radio del mundo actual y sus proyecciones futuras.

1.1 Estructura de un Sistema SDR

Para describir la estructura de un sistema de Radio Definido por Software se debe partir desde los sistemas de comunicación digitales en general. La figura 1.1 muestra el diagrama en bloques de un sistema de este tipo. Algunos de estos bloques son imprescindibles para el funcionamiento de cualquier sistema de comunicaciones digitales, como por ejemplo conformación del pulso, filtro acoplado, modulación digital y la sección de radiofrecuencia. Otros son prescindibles pero necesarios para un mejor funcionamiento del sistema, por ejemplo codificación de canal, espectro extendido y sincronismo.

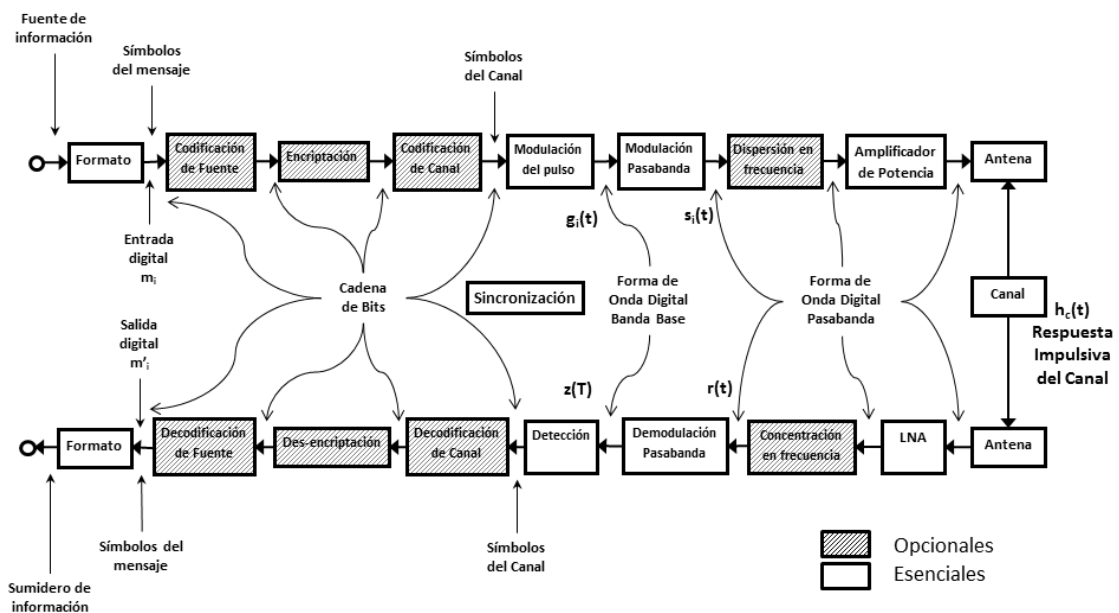


Figura 1.1 Diagrama en bloques de un Sistema Digital de Comunicación.

No se encuentra dentro del alcance de este trabajo el estudio sobre los sectores de amplificador de potencia, antenas, amplificador de bajo ruido (LNA) y canal –excepto su implicación en el diseño del filtro acoplado–. Esta sección se tiene en cuenta en la implementación y simulación del sistema solo en la medida en que estos aspectos afectan el proceso de transmisión de la información. Tampoco se incluyen en el sistema

creado los procesos de encriptación y codificación de fuente por considerarlos un aumento considerable del costo computacional, en relación a los beneficios que estos conllevan para el funcionamiento del sistema.

1.2 Concepto de Radio Definido por Software

Existen disímiles conceptos de SDR basados en enfoques diferentes del problema que dependen, fundamentalmente, del campo de estudio de quien establece la definición.

*Wireless Innovation Forum*¹ establece, en su sitio online, que un Radio Definido por Software es un “radio en el que algunas o todas las funciones de la capa física son definidas por software”. [13]

De acuerdo a Lee Pucker, colaborador de *Wireless Innovation Forum*, SDR es una tecnología aplicable en un amplio rango de áreas dentro la industria inalámbrica, capaz de proveer soluciones eficientes y comparativamente baratas a muchos problemas inherentes a arquitecturas de radio más tradicionales. En esencia, SDR es un término utilizado para describir una tecnología de radio donde algunas o todas las funciones de la capa física inalámbrica son definidas por software. Aunque a primera vista parezca reducido, el concepto de Pucker da una idea suficientemente clara de lo que propone la tecnología SDR. [14]

William A. Sethares y C. Richard Johnson, Jr. en [15] afirman: “Definido por Software significa que los elementos claves del radio están implementados en software”; y continúan: “Tomar un enfoque de ‘definido por software’ refleja la tendencia de los diseños modernos donde una parte cada vez mayor del sistema está diseñada y construida utilizando software reconfigurable, en vez de hardware fijo. Esto permite introducir, demostrar y entender los conceptos detrás de la transmisión a través de simulación. Por ejemplo, cuando se hable de traslación en frecuencia de una señal, los procedimientos se presentan matemáticamente a través de ecuaciones, gráficamente usando diagramas en bloques y luego concretamente como programas de MATLAB”. Estos autores proponen un concepto orientado hacia el desarrollo y aplicación de SDR en un ambiente comercial y profesional pero mostrando su alcance para estudio y

¹*Wireless Innovation Forum* –anteriormente *SDR Forum*: Corporación de beneficio mutuo sin fines de lucro a la cabeza del desarrollo de tecnologías inalámbricas emergentes.

simulación en ambientes investigativos, por tratarse de una obra destinada a la enseñanza de SDR usando MATLAB.

Tod Sizer, director y líder del dominio de tecnología inalámbrica de *Alcatel-Lucent Bell Lab*, establece su definición desde un enfoque de mercado donde muestra la ampliación de los horizontes de SDR: “Radio Definido por Software es un término difuso, es la habilidad de modificar un radio para actuar de un modo diferente, mejorar su comportamiento y ser utilizado para más de un propósito”. [16]

Un concepto de SDR que obedezca al perfil de esta investigación deviene una confluencia de los enfoques mostrados, pues ésta constituye un estudio académico de SDR para actualizar al lector sobre cada una de las tendencias actuales de esta tecnología. Teniendo en cuenta lo anterior, SDR se define como:

Tecnología de radio donde algunas o todas las funciones de la capa física están definidas por software, proveyéndole flexibilidad, capacidad de reconfiguración, actualización, y posibilidades de adaptación a diferentes ambientes y propósitos a los dispositivos de radio.

Esta tecnología tiene amplias aplicaciones en el área de las comunicaciones inalámbricas pues brinda soluciones eficientes a problemas inherentes a las arquitecturas de radio tradicionales. También ofrece facilidades para una investigación menos costosa a través de simulación, permitiendo su experimentación y estudio a profundidad en entidades profesionales, comerciales y académicas.

1.2.1 Niveles de SDR

Con el fin de establecer una clasificación de los equipos de radio, el *Wireless Innovation Forum* ha definido 5 niveles de utilización de software dentro del radio para controlar o realizar funciones de la capa física, determinando la frontera entre hardware y software en el equipo:

★ Nivel 0.

Radio construido utilizando hardware solamente, no puede cambiarse por software. En este nivel no hay software, ni en el control ni en la realización propia de las funciones de operación del radio.

★ Nivel 1

Radio controlado por software con limitaciones en cuanto a funciones controlables. Se controlan algunas como nivel de potencia, interconexiones, etc. pero nunca modo o frecuencia.

★ Nivel 2

En este nivel una parte significativa del radio es configurable por software. Usualmente se utiliza el término Radio Controlado por Software (SCR). Existe control de software de ciertos parámetros como frecuencia, modulación, generación/detección de forma de onda, seguridad, etc. La etapa de RF permanece en hardware y no puede ser reconfigurada. Es importante destacar que el software en este tipo de radios sólo controla funciones que están implementadas de modo físico dentro del radio, a hardware únicamente.

★ Nivel 3

Este nivel agrupa todos los radios en los que al menos una de sus funciones está definida por software, incluyendo lo que se conoce como Radio Definido por Software Ideal (ISR) donde la frontera entre la parte configurable y la no configurable se encuentra muy cercana a la antena, y la etapa final de RF es configurable. Se puede decir que el ISR es completamente programable.

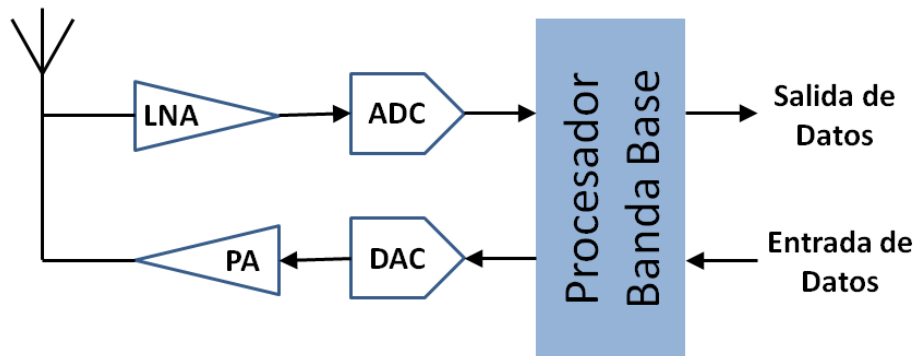


Figura 1.2 Diagrama en bloques simplificado de un SDR nivel 3.

★ Nivel 4

Los equipos pertenecientes a esta categoría reciben la denominación de *Ultimate Software Radio* (USR) y constituyen una etapa superior al ISR puesto que no solo son completamente programables sino que además son capaces de soportar un amplio rango de funciones y frecuencias simultáneamente. Dentro de este nivel se encuentran los teléfonos celulares con soporte para varios estándares.

Este trabajo de tesis se enfoca en el nivel tres, por ser aquellos en los que las funciones de la capa física del radio están definidas por software.

1.3 Aplicaciones Prácticas de SDR

Por el sutil modo en que la idea de SDR se fue introduciendo en la fabricación de equipos de comunicaciones por radio se hace difícil identificar un inicio cronológico del mismo. No obstante, su entrada en el ámbito comercial ha impactado en aspectos como eficiencia energética, costo y tamaño físico de los equipos, unido a las oportunidades que esta tecnología ofrece para la creación de radios baratos, flexibles y energéticamente económicos. Los acápites siguientes darán muestra de esta afirmación con ejemplos de la utilización de SDR en varios sectores relevantes de las comunicaciones radioeléctricas.

1.3.1 Tecnologías de Internet Inalámbrica y Telefonía Celular

Los usuarios, con la intención de participar competitivamente en la sociedad, quieren tener acceso libre a los servicios emergentes de triple/cuádruple *play* en todas sus vías de conexión, ya sean suscripciones cableadas o inalámbricas. A nivel internacional, los operadores de servicios de telecomunicaciones, sean éstos inalámbricos o no, han mostrado intenciones de evolucionar sus redes de conmutación de circuitos hacia redes multimedia basadas en el principio de “*All IP*”, y muchos de ellos tienen evoluciones en progreso.[17]

La industria del multiradio es una tendencia tentadora en los segmentos de redes celulares y de acceso inalámbrico de banda ancha (BWA). SDR es la más clara opción para lograrlo puesto que usando esta tecnología se puede hacer selección de forma de onda y se pretende lograr la operación simultánea de varios sistemas. Aunque existe diferencia de opiniones, fuentes de la industria comercial indican que en los futuros celulares podrían emerger hasta 11 radios. WiFi, WiMAX y Beyond 3G están desarrollando iniciativas para la utilización de SDR, unido a Multiplex por División Ortogonal en Frecuencia (OFDM), en sus estándares de evolución para lograr la simultaneidad de los servicios en los terminales de usuario de estas tecnologías.[17]

Los requerimientos claves de todo BWA incluyen:

1. Razones de bit en el orden de los megabit por segundo (Mbps).

2. Calidad de servicio. Habilitación de servicios multimedia en tiempo real tales como voz sobre IP (VoIP), video sobre IP (VioIP) y juegos multimedia.
3. Seguridad.
4. Movilidad.

La estandarización y despliegue del paradigma de movilidad de la telefonía celular en WiFi y WiMAX requerirá tiempo para lograr una funcionalidad competitiva. Sin embargo, la aplicación de SDR en las redes de celulares está brindando amplias posibilidades a los operadores en este sector, posibilitando la extensión de la vida útil de las estaciones base y el alivio de la congestión en el espectro, entre otras ventajas.[17]

Lisbet Van der Perre, de la sección de investigación inalámbrica en IMEC –centro de investigaciones sobre microelectrónica con sede en Bélgica- plantea el problema así: “los móviles cambian todos los años, y con las estaciones base es peor, mientras que los operadores las quieren instaladas por décadas”. Van der Perre muestra una posible solución al problema cuando dice que “con SDR una estación base instalada puede incorporar un nuevo protocolo inalámbrico usando solo una descarga de software. Diseños tan flexibles proporcionan ventajas significativas en cuanto al ahorro de capital y costo operacional, acortando el proceso de introducción de servicios y el tiempo que tarda la tecnología en llegar al mercado”. Los diseños de estaciones base con SDR están siendo adoptados por todos los fabricantes de equipos celulares. En 2004, los proveedores de telefonía celular mejoraron sus redes GSM hacia EDGE usando software solamente y, más recientemente, se añadieron a las redes 3G tecnologías de descarga y subida de Acceso por Paquetes a Alta Velocidad (HSPA) del mismo modo.[16]

De hecho, el IMEC tiene una línea de investigación profunda sobre Radio Cognitiva y SDR cuyos resultados se han concretado en varios dispositivos que funcionan con SDR y proporcionan alternativas de hardware y software para los SDR.[18-20]

En la actualidad los fabricantes de equipamiento están extendiendo la idea de modo que la misma estación base pueda soportar múltiples estándares inalámbricos. “Con las estaciones base SDR multiestándar un operador puede mantener la tecnología de radio existente tanto tiempo como lo necesite antes de migrar, por medio de actualización de software, cuando las condiciones sean las correctas”, según Zhongrong Liu, líder de la sección de estrategia de tecnologías de la red de radio en *T-Mobile International*. “Esto

permite la reducción progresiva del espectro para la tecnología vieja mientras se incrementa el espectro para la nueva”, plantea Liu.[16]

Los móviles también están adoptando SDR, aunque por razones diferentes. Los móviles necesitan agrupar más radios aun para soportar protocolos como 2G, 3G y WLAN, así como TV móvil, GPS y Bluetooth. Según investigaciones del IMEC, actualmente son comunes los móviles con seis o más radios. “Un OEM (*Original Equipment Manufacturer*), para dos o tres radios, necesita fabricar un nuevo chip mientras nosotros solo tenemos que cambiar el software contenido en el chip”, dice John Glossner, director tecnológico y cofundador de *Sandbridge Technologies*. “Esto representa importantes beneficios en costo de desarrollo mientras acorta el tiempo necesario para desarrollarlo”. [16]

Sin embargo, no todo es positivo en la aplicación de SDR a los móviles. Nokia acepta que el uso de SDR en estos dispositivos aumentará, pero gradualmente. Petteri Alinikula, director del centro de investigaciones de Nokia en Helsinki, aclara que SDR “no resuelve todos los problemas de implementación del radio. En principio, aunque el procesamiento en banda base puede ser capaz de implementar varios radios simultáneamente, allí se necesitan circuiterías de radiofrecuencia diferentes para reunir todas las señales de radio para su procesamiento”. De acuerdo al Vice-presidente de Nokia, Timo Ali-Vehmas, la relación entre el impacto de la reconfiguración en el costo de diseño y el consumo de energía es una ecuación extremadamente complicada que precisa de un análisis profundo. Y hay más, nuevos requerimientos para los móviles de liderazgo están surgiendo constantemente y SDR podría no ser la solución óptima. “SDR es un elemento en la caja de herramientas. No es una cuestión de blanco y negro”, agrega Ali-Vehmas.[16]

Alrededor de este tema existe gran polémica entre los líderes del mercado puesto que, independientemente de retos como este, ya SDR es una tecnología desplegada en las redes de los operadores móviles. Los defensores de SDR plantean que es un precursor importante en el advenimiento del radio cognitivo (CR), que promete un mejor uso del espectro radioeléctrico y, según Sizer, si los operadores móviles ven ahora siquiera una insignificante razón para perseguir CR, lo valorarán seriamente cuando se les empiece a agotar el espectro disponible. Mientras tanto Liu, de *T-Mobile International*, establece que la explotación comercial de CR no se espera hasta los próximos 8-10 años, por lo

menos. Por su parte, el profesor William Webb, destacado colaborador de Ofcom, se muestra escéptico acerca del impacto de CR cuando afirma: “No creo que esto conduzca a una revolución en la forma en la que utilizamos el espectro”. [16]

Las oportunidades de SDR son significativas en las tecnologías de BWA y las iniciativas para telefonía celular. SDR podría ser una herramienta facilitadora para alcanzar capacidades de multiradio y OFDM, además de constituir un punto de direccionamiento para los requerimientos de mercado emergentes.

1.3.2 Radio Cognitivo

Radio cognitivo es aquel en el cual los sistemas de comunicaciones están al tanto de las condiciones de su estado interno y del ambiente, tales como localización y utilización del espectro de frecuencia de RF. Estos sistemas pueden tomar decisiones sobre su comportamiento comparando esta información con objetivos predefinidos. Además, CR se define por su utilización de SDR para ajustar automáticamente su comportamiento u operaciones para alcanzar los objetivos deseados. La utilización de estos elementos es crítica para permitir a los usuarios finales hacer uso óptimo del espectro de frecuencias disponible y redes inalámbricas con un hardware común de radio. Esto reducirá el costo para el usuario final permitiéndole comunicarse con quienquiera que lo necesite, en el momento que lo necesite y de cualquier forma que estime adecuada. [21]

Los objetivos a corto plazo del empleo de esta tecnología son eficiencia espectral y disponibilidad de espectro para soportar los requerimientos emergentes en las tecnologías inalámbricas y las oportunidades de mercado. El objetivo a largo plazo es la capacidad cognitiva de los sistemas. Más allá de los propósitos en cuanto a eficiencia, mejoría en la efectividad de las regulaciones, licencias, y administración del espectro, se pretende que el radio en sí mismo sea capaz de anticipar y acomodar las necesidades de los usuarios y la eficiencia del sistema. La mayoría de las organizaciones de estándares consideran CR y SDR como tecnologías distintas pero complementarias e impulsoras una de la otra. [17]

Existen varios tipos de CR que agrupan a cada uno de estos equipos en dependencia de los parámetros que monitorea y el espacio del espectro que utilizan. Atendiendo a los parámetros de los que el radio es capaz de tener conocer el estado, los CR se dividen en dos grupos: CR Completo -conocido también como radio Mitola- y CR de Espectro. El primero se refiere a aquellos que tienen en cuenta todos los parámetros posibles,

incluyendo espectro de radiofrecuencia, estado de la red y comportamiento del usuario. Los CR de Espectro son aquellos que solo consideran la cuestión de espectro de radiofrecuencia. En cuanto al espectro de frecuencia que utilizan, los CR se clasifican como CR de Banda con Licencias y CR de Bandas sin Licencias. El primero opera indistintamente en bandas de frecuencia que son exclusivas para usuarios con licencia y en bandas que se pueden usar sin necesidad de poseer una. El segundo solo puede utilizar las regiones del espectro de radiofrecuencia que no están bajo el control de licencias.[22]

Los mercados para SDR y CR en conjunto están desarrollándose desde principio de los años 2000. Sus áreas más importantes son: tecnologías inalámbricas, seguridad pública y defensa. De ellas la más relevante es la de tecnologías inalámbricas, principalmente: mercado de telefonía celular, incluyendo 3G; Evolución a Largo Plazo para GSM (LTE) y 4G; Redes de Área Local Inalámbrica, o WiFi; y Acceso Inalámbrico de Banda Ancha (BWA), o WiMAX. Las oportunidades claves de SDR y CR en estos sectores comerciales son eficiencia y disponibilidad espectral, mantenimiento cognitivo para las configuraciones de red y terminal con el fin de dar soporte a iniciativas emergentes de triple/cuádruple *play*, multimedia y multiradio². Las características específicas de SDR y CR se tornan indispensables para el logro de los objetivos del mercado inalámbrico comercial.

La aplicación clave de SDR y CR en el ámbito de la seguridad pública es la interoperabilidad. En muchos eventos de desastre pueden no estar disponibles las comunicaciones debido a daños en su infraestructura. Por tanto el requerimiento esencial de las tecnologías de CR es reconfigurar los radios de los rescatistas como una extensión ad hoc a la red existente para permitir el relevo de las comunicaciones desde el sitio del desastre a través de una red de radios individuales, proveyendo acceso a la red principal de radio.

La comunidad de la defensa y seguridad nacional ha sido líder en las iniciativas de SDR y CR a nivel internacional. Aunque existen intereses en este sentido a nivel internacional, a la cabeza de estas iniciativas ha estado el Departamento de Defensa de

² Multiradio se refiere a la inclusión de múltiples estándares de radio (o formas de onda) en una misma plataforma, sea esta móvil, portable o de infraestructura. Pretende que al menos dos enlaces de radio trabajen simultáneamente.

los Estados Unidos de América con su proyecto *DARPA Next Generation* y otros programas relacionados. Los programas militares en esta dirección han estado enfocados en Redes Móviles Ad Hoc (MANETs) que no requieren infraestructura de soporte pues utilizan la tecnología de redes IP.[17]

De las iniciativas para el desarrollo de CR a nivel internacional la que más se destaca por su alcance es el proyecto Reconfiguración Extremo-a-Extremo (E²R) de la Unión Europea. El mismo se enfoca en idear, diseñar y validar soluciones de arquitectura, y respaldar algoritmos, protocolos y mecanismos en redes cognitivas, donde la noción de “radio cognitivo” incluye facilidades de software y radio cognitivo enriquecidas por capacidades de autogestión.[23]

Aunque las posibilidades de SDR en el área de CR parecen ser grandes, según un reporte de estudio de Ofcom, publicado en 2007, todavía existen dos obstáculos a salvar para alcanzar Radio Cognitivo total. El primero es hacer un dispositivo verdaderamente cognitivo con la habilidad y la inteligencia para tomar decisiones basados en el conocimiento de su propia situación. Y el segundo es la evolución de SDR para permitir la reconfiguración total. La solución a estos problemas es cuestión de tiempo, SDR se ha desarrollado lo suficiente para lograr un radio realmente cognitivo en el transcurso de las próximas dos décadas.[24]

1.3.3 Industria Automotriz

La aplicación de SDR en los automóviles va encaminada fundamentalmente a la utilización de las computadoras y las tecnologías de las comunicaciones para mejorar las experiencias de viaje en los automóviles. Las autopistas del mundo cada vez son más congestionadas, y añadir nuevas vías para aliviar este fenómeno se vuelve una opción cada día menos atractiva, por lo que se necesitan acercamientos alternativos al tema. La telemática ofrece buenas soluciones con la aplicación e integración de geo-localización, comunicaciones y tecnologías de la información dentro de la arquitectura de los vehículos y la infraestructura de las carreteras y autopistas. Las promesas de esta integración son conveniencia, seguridad y capacidad de tráfico incrementada sin construcciones nuevas.[17]

Un líder en servicios de este tipo en Norteamérica y China es la subsidiaria de *General Motors* OnStar, una compañía dedicada a ofrecer servicios de comunicaciones, seguridad, conversación manos libres, navegación, y ayuda remota.[9]

Internacionalmente, el foco de atención del mercado de la telemática y el de los gobiernos es diferente. En Norteamérica las preocupaciones son seguridad y comodidad; en Europa el énfasis ha estado en la información de tráfico y navegación, mientras en Asia, en especial Japón por ser líder tecnológico de la región, los intereses son navegación, información y entretenimiento. Hasta hace dos décadas nadie había obtenido beneficios relevantes produciendo servicios de telemática en esta área. Sin embargo, servicios de información de tráfico, regularmente gratuitos y en ocasiones con ganancias para alguna compañía, han sido bastante populares en algunas áreas geográficas. Un ejemplo es el servicio de Canal de Mensajes de Tráfico usando subportadoras de FM con amplio despliegue en Europa. Las unidades con bases de datos de mapas y pantalla están logrando popularidad como opciones de sobreprecio en todos los mercados.[17]

Un ejemplo notable de ese tipo de unidades es el sistema *Sync* de Ford que usa el sistema operativo *Microsoft Windows Embedded Automotive OS* para que los usuarios del vehículo puedan, mediante conexiones de tipo Bluetooth y a través de comandos de voz, acceder a teléfono, reproductor de mp3 y dispositivos USB; conectarse directamente con operadores de emergencias, recibir mensajes de texto audibles, y otras opciones. Este sistema salió a la venta en 2007 y está disponible en Estados Unidos de América en 20 modelos de automóviles.[8]

Los automóviles emplean Unidades de Control Electrónicas (ECU) para implementar ventajas efectivas y eficientes desde el punto de vista de costos, para una variedad de funciones mecánicas, hidráulicas y eléctricas/electrónicas tradicionales. Un vehículo moderno tiene, típicamente, entre 20 y 80 ECUs. Una tendencia emergente es consolidar menos y más poderosas ECUs y mejorar las funcionalidades de comunicación entre unidades a través de buses en el vehículo, usando radios definidos por software que aprovechen las potencialidades de las unidades electrónicas.[17]

Un estándar ya establecido en el mercado a nivel internacional con este propósito es *FlexRay*, sistema de bus serie digital robusto, escalable, determinístico y tolerante a fallas diseñado para utilizarse en aplicaciones automotrices. Algunos modelos de autos que utilizan *FlexRay* son: Audi A8, Bentley Mulsanne, BMW X5, Serie BMW 7, Serie BMW 5 Gran Turismo, Serie BMW 5, y Rolls-Royce Ghost.[25]

Otras iniciativas en la industria automotriz, agrupadas bajo el término Sistemas Inteligentes de Transportación, son una solución de sinergia con SDR. Un ejemplo de ellas es AUTOSAR, una arquitectura de software automotriz abierta y estandarizada, desarrollada por fabricantes, proveedores y desarrolladores de automóviles y de herramientas. El objetivo principal de AUTOSAR es establecer estándares abiertos para arquitecturas eléctricas/electrónicas en automóviles, que provean una infraestructura básica para el desarrollo de software destinado a vehículos.[7]

Aunque la telemática no ha logrado gran éxito en este terreno hasta el momento, la necesidad de mejorar la seguridad, información de tráfico y navegación, y entretenimiento, unida a la sinergia de las emergente iniciativas de telecomunicaciones de “en cualquier momento, en cualquier lugar, y cualquier servicio” con la tecnología SDR; crean un ambiente muy favorable para el desarrollo y éxito de la telemática y SDR en el mercado de la industria automotriz.

1.3.4 SDR Implementados

Desde hace unos años se han venido desarrollando algunos radios basados en SDR por parte de aficionados, empresas privadas y desarrolladores libres. Ejemplo de esta afirmación son:

- *FUNcube Dongle*: Receptor de radio, o segmento de tierra, desarrollado por AMSAT-UK como parte de un proyecto educativo de colaboración global que sirve como receptor de señales de satélites *FUNcube* alrededor del mundo. Se le puede variar la frecuencia entre 64 y 1,700 MHz. Su precio está cercano a los \$200 USD.[4]



Figura 1.3 SDR FUNcube Dongle de AMSAT-UK

- PM-SDR: Desarrollado por Martin Pernter, utiliza el software libre WinRad de Jeffrey Pawlan, así como otros programas para funcionar en Linux. Se conecta por USB y puede configurarse en el rango de 100kHz - 50MHz.[6]

- Finningley 80m: Creado por *Microwave Roundtable* y reproducido por *Kanga Products*. Es un kit para la familiarización con SDR y su precio es de más de \$25 USD.[5]
- EXCALIBUR: Creado por WiNRADiO, es un receptor de onda corta entre 9KHz y 50 MHz. Incluye un analizador de espectro de tiempo real de 50MHz de ancho de banda y su sensibilidad ronda los 0.35 μ V. Su interfaz de software es una de las más completas creadas para SDR.[11]

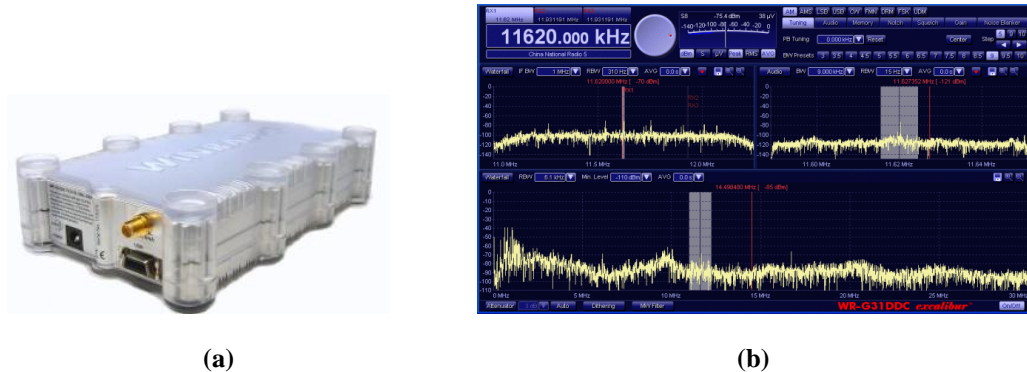


Figura 1.4 SDR EXCALIBUR de WiNRADiO. a) Dispositivo SDR. b) Analizador de espectro de tiempo real del EXCALIBUR.

- *Universal Software Radio Peripheral* (USRP): Tarjeta Madre desarrollada por GNU Radio a la que se le pueden incorporar otras tarjetas para darle funcionalidad. Es capaz de manejar muchas características que pueden ser consultadas en [3]. Tiene una versión mejorada, USRP2. Es uno de los SDR más potentes que se pueden encontrar en la actualidad y emplea software libre.[3]

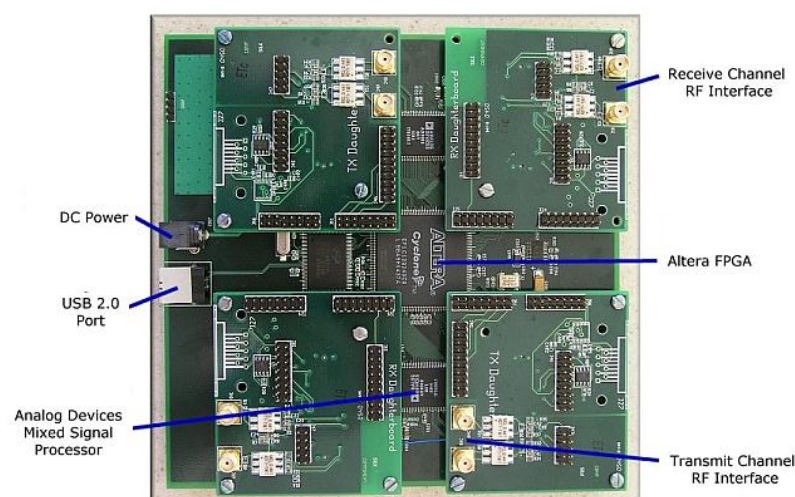


Figura 1.5 USRP fabricado por GNURadio.

- QS1R: Creado por *QuickSilver*, es el receptor SDR más completo que se puede encontrar en el mercado. Abundante en opciones y posibilidades de

configuración, utiliza el rango de frecuencias de 10 KHz a 62.5 MHz, que puede ser ampliado hasta 500 MHz con el empleo de sobremuestreo.[10]



Figura 1.6 Receptor QS1R de *QuickSilver*. Vista frontal y trasera.

1.4 Estudio Académico de SDR

El amplio impacto de SDR en las comunicaciones a nivel mundial ha provocado que este tema sea incorporado a los planes de estudios de las carreras de telecomunicaciones en muchas universidades, incluyendo prestigiosas instituciones como Instituto Tecnológico de Massachusetts (MIT), Universidad de Stanford, Instituto Tecnológico de Tokio, Universidad de Notre Dame y otros.[2, 26, 27]

Uno de los proyectos de mayor impacto en relación a SDR es el *Smart Radio Challenge* de *Wireless Innovation Forum*. El objetivo de esta competición es exponer a los estudiantes a los problemas del mundo real que encontrarán cuando se gradúen y fomentar la interacción entre la industria y la academia. En el desarrollo de la competición, equipos de estudiantes de ingeniería diseñan, desarrollan y prueban tecnologías SDR y CR dirigidos a relevantes problemas de las tecnologías inalámbricas. Los principales patrocinadores de este proyecto son los desarrolladores de MATLAB, *MathWorks Inc.*[27]

En los centros de altos estudios donde se imparte SDR y/o tecnologías relacionadas, los estudiantes reciben el contenido de forma teórica y después lo llevan a la práctica a través de prácticas de laboratorio y proyectos de investigación como [2, 26, 28]. En las investigaciones los estudiantes utilizan dispositivos de propósito especial como tarjetas de adquisición de datos (DAQ), procesadores de señales digitales (DSP), FPGA y otros, elementos cuyo costo económico dificulta su empleo.[2, 26]

El sistema creado en este trabajo ha sido pensado desde un inicio para su empleo en instituciones que, como la Facultad de Ingeniería Eléctrica de la Universidad Central de Las Villas, no disponen de estos recursos. Para suplir esta carencia se utilizan computadoras de propósito general existentes en los centros de estudios superiores y el MATLAB como herramienta de procesamiento y simulación que sustituye el funcionamiento de muchos de estos dispositivos. Los requerimientos de hardware se reducen a una computadora estándar con Sistema Operativo Microsoft Windows, MATLAB versión 7.8.0.347 (R2009a) –o superior– e interfaz de audio estándar, con entrada de línea en el caso del receptor.

CAPÍTULO 2. IMPLEMENTACIÓN DE SDR USANDO MATLAB

En este capítulo se desarrolla la explicación del sistema de comunicaciones digitales basado en SDR desarrollado durante la realización de este trabajo. Se incluye un breve preámbulo teórico al inicio de cada epígrafe en que se ha considerado necesario para un mejor entendimiento del funcionamiento del sistema. Sin embargo, no se pretende llevar a cabo explicaciones exhaustivas de temas de comunicaciones por alejarse del alcance de este trabajo.

En cada epígrafe se incluyen segmentos de las funciones en MATLAB que componen el sistema con el fin de ilustrar mejor los métodos y algoritmos utilizados. Todo el sistema ha sido programado utilizando MATLAB versión 7.8.0.347 (R2009a) por tanto algunas funciones empleadas presentan problemas de compatibilidad con versiones anteriores de este software.

Aunque la herramienta Simulink de MATLAB resulta mucho más amigable, el sistema se ha implementado como funciones por estar destinado a enseñar los conceptos de comunicaciones a estudiantes de Ingeniería en Telecomunicaciones que están familiarizados con el software, así como profundizar los conocimientos sobre el mismo que posean.

2.1 Transmisor

A continuación se muestra el diagrama en bloques del transmisor del sistema SDR implementado como parte de la elaboración de este trabajo. Los métodos y algoritmos utilizados para la realización de cada una de las funciones del mismo se explican detalladamente en los siguientes epígrafes.

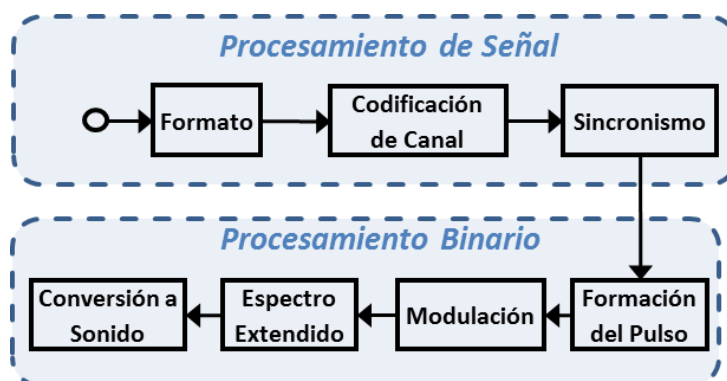


Figura 2.1 Diagrama en bloques del transmisor

2.1.2 Formato

Esta sección del transmisor se encarga de darle formato a la información de modo que ésta sea convertida en una cadena binaria. Es dependiente del tipo de información a tratar puesto que MATLAB importa de forma diferente cada uno de los tipos de archivo que soporta. En la elaboración de este transmisor se implementaron dos funciones diferentes que responden a este bloque.

La primera de las variantes convierte una cadena de caracteres introducida como parámetro a la función del transmisor en una cadena binaria empleando 8 bits para cada carácter. El procedimiento utilizado es el de convertir la cadena al tipo de datos de MATLAB “*double*”, llevar estos datos a binario, y luego transformar en una fila la matriz obtenida en este proceso.

La segunda variante transforma cualquier imagen introducida como parámetro a la función del transmisor en una cadena binaria usando 8 bits para la codificación de cada píxel. El proceso es similar al desarrollado por la variante descrita anteriormente pero en este caso primero hay que leer la imagen, usando la función ‘*imread*’ de MATLAB, para luego convertirla a binario.[29]

2.1.3 Codificación de Canal

Codificación de Canal se refiere a la clase de transformaciones de señal diseñadas para mejorar el rendimiento de las comunicaciones, facilitándole a las señales transmitidas contrarrestar mejor varios efectos negativos del canal como ruido, desvanecimiento y *jamming*.

Usualmente, el objetivo de la codificación de canal es reducir la probabilidad de error de bit (P_B) o la relación E_b/N_0 requerida, al precio de emplear más ancho de banda del necesario si no se utilizara esta codificación. La codificación de canal se convirtió en una forma muy popular de proveer mejoras de funcionamiento gracias al empleo de circuitos de gran escala de integración (LSI) pues éstos son capaces de proveer una mejora de hasta 8 dB a un costo mucho menor que con cualquier otro método como el empleo de transmisores de potencia o antenas mayores. La codificación de canal puede dividirse en dos áreas de estudio: codificación de forma de onda y secuencias estructuradas (o redundancia estructurada).[1]

La codificación de forma de onda se ocupa de transformar esta característica para hacer el proceso de detección menos sujeto a errores. Ejemplos de esta codificación son el empleo de señales antipodales, ortogonales, biortogonales, transortogonales y señalización M-aria.

Las secuencias estructuradas tratan la transformación de secuencias a través de la introducción de redundancia en las mismas. El proceso de codificación –por cualquiera de las dos vías– incorpora en la señal codificada mejores propiedades de distancia. Los bits redundantes pueden utilizarse para la detección y corrección de errores. Existen dos vías básicas para usar la redundancia en el control de errores.

La primera, denominada detección de error y retransmisión, generalmente utiliza bits de paridad para detectar la ocurrencia de errores, en tal caso el terminal receptor no intenta corregir el error y solicita una retransmisión de los datos. Esta vía necesita una conexión dúplex y se implementa usando técnicas de Solicitud Automática de Retransmisión (ARQ).[1]

La segunda vía para el control de error, denominada corrección de error hacia adelante (FEC), emplea la redundancia para tratar de corregir los errores en caso de que estos ocurran. ARQ tiene como ventajas que la detección de errores requiere menos redundancia, el proceso de detección es menos complejo y costoso, y es adaptativo puesto que la información solo se retransmite cuando ocurren errores. Sin embargo, FEC es una opción viable, o inevitable, cuando no se dispone de un canal de retorno al transmisor, el retardo con ARQ resulta excesivo, la estrategia de retransmisión no ha sido implementada convenientemente o el número de errores esperado si no se usara corrección requiriera retransmisiones excesivas. Por tanto, si se está diseñando un sistema de forma que solo se transmita en una dirección, entonces la opción obvia es FEC.[1]

Las secuencias estructuradas se dividen en codificadores de bloque y codificadores convolucionales. Aunque algunos codificadores de bloque solo son capaces de detectar errores, existen muchos capaces de corregirlos también, por ejemplo: matriz de paridad cruzada, código de Hamming, código Golay Extendido, y código BCH. Los códigos convolucionales tienen como característica principal que son sistemas con memoria puesto que la salida del codificador no depende exclusivamente del(los) bit(s) actual(es) pues está determinada por el(los) bit(s) actual(es) y varios bits anteriores, en

dependencia de la complejidad del codificador. Esta característica permite que los códigos convolucionales tengan un mejor rendimiento que códigos de bloque con igual complejidad de implementación de codificador-decodificador. Por esta razón, los códigos convolucionales son los más utilizados en la últimas tres décadas.[1]

En este sistema se utiliza un código convolucional implementado de modo que se emplee un decodificador basado en el algoritmo de Viterbi en el receptor. Para ello se crea una estructura de rejilla usando la función de MATLAB 'poly2trellis' que luego se utiliza en la codificación propiamente dicha a través de la función 'convenc'. [29]

Una característica importante de los códigos convolucionales es que existe la posibilidad de que propaguen el error catastrófico. Un error catastrófico se define como un evento en el cual un número finito de errores de símbolo de código provoca infinitos errores de bit en los datos decodificados. Una condición necesaria y suficiente para que un generador diseñado por polinomios propague el error catastrófico es que dichos generadores tengan un factor común.[22]

Aunque existe una clase de códigos que nunca propagan el error catastrófico, denominados códigos sistemáticos, solo una pequeña parte de los códigos no sistemáticos propaga el error catastrófico. No obstante, el sistema descrito aquí utiliza la combinación de dos codificadores óptimos ofrecidos por [22] y se puede verificar que no propaga el error catastrófico a través de la función de MATLAB 'iscatastrophic'. La representación en forma de registros de desplazamientos se muestra en la figura 2.2.

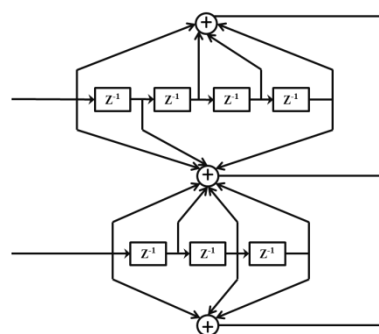


Figura 2.2 Representación por registros de desplazamiento del codificador convolucional.

Una vez que los datos han sido codificados, se introduce al inicio la longitud total de éstos con el fin de conocer el final de la cadena en el receptor.

```
l= length(x);  
trell = poly2trellis([5 4],[27 31 0; 0 17 13]); % Diseño del generador  
x = convenc(x, trell); % Codificación  
x = [de2bi(l, 128, 2) x];
```

2.1.4 Sincronismo

Cuando arriba al receptor, la señal es una forma de onda que debe ser muestreada con el fin de recuperar el mensaje transmitido. Existe cierto retardo por propagación, inherente a los sistemas de comunicación, que genera que estas muestras no se tomen en el instante adecuado. Por tanto se necesita algún medio de determinar cuándo tomar las muestras. Existen varios algoritmos diseñados para lograr el sincronismo de símbolo. La explicación de algunos de ellos y los fundamentos para la elección hecha en la realización de este sistema se pueden encontrar en el acápite de sincronismo de símbolo del receptor. Por ahora solo es necesario conocer que para un mejor desempeño del algoritmo implementado en este sistema, es importante introducir al inicio de los datos transmitidos una secuencia de entrenamiento consistente en una sucesión binaria que alterna ceros y unos. Esto se realiza al inicio de la sección del sistema descrita en este epígrafe.

El empleo de una cabecera para establecer el sincronismo de trama es una técnica ampliamente utilizada en casi cualquier medio de comunicaciones digitales. La cabecera usada es una secuencia binaria aleatoria conocida en ambos extremos de la transmisión por la introducción de ésta en el diseño e implementación práctica del sistema. La secuencia se genera utilizando la instrucción ‘fd=randint(1, longitud_deseada);’, usando el parámetro ‘longitud_deseada’ para definir la longitud de la secuencia[29]. Para garantizar que la secuencia es la misma en ambos extremos de la comunicación, se debe reiniciar la semilla del generador aleatorio de MATLAB, véase el código a continuación. Esta cabecera se encuentra a continuación de la secuencia de entrenamiento explicada en el párrafo anterior.

Sincronismo de Trama

```
RandStream.setDefaultStream(RandStream('mt19937ar', 'seed', 117));  
fd = randint(1, 2000);
```

2.1.5 Formación del Pulso

Debido a los efectos de filtrado de un sistema de comunicaciones, los pulsos recibidos se superponen entre sí y la energía secundaria de un pulso produce un efecto ruidoso en intervalos de símbolos adyacentes que interfiere en el proceso de detección. Este fenómeno recibe el nombre de Interferencia Intersímbolo (ISI).[1, 15, 30]

Matemáticamente la función de transferencia de un sistema se representa:

$$H(f) = H_t(f)H_c(f)H_r(f)$$

Donde $H_t(f)$ representa el filtro del transmisor, $H_c(f)$ el efecto de filtrado del canal, y $H_r(f)$ el filtro del receptor. Aun en ausencia de ruido y con perfecto sincronismo, las limitaciones de ancho de banda y el filtrado imperfecto del sistema generan ISI. En la práctica la situación más común es encontrar que $H_t(f)$ es conocida y el problema radica en determinar $H_c(f)$ y $H_r(f)$ de modo que se minimice la ISI de los pulsos a la salida de $H_r(f)$. Nyquist investigó el problema de especificar una forma del pulso para eliminar la ISI en el detector y demostró que cuando $H(f)$ es rectangular, entonces el ancho de banda mínimo para detectar R_s símbolos/s es $\frac{R_s}{2}$ Hz. Cuando esto ocurre, o sea $H(f)$ es un filtro ideal de ancho de banda $\frac{1}{2T}$ Hz $-T$ es el período de símbolo—, entonces la respuesta al impulso de $H(f)$ es $h(t) = \text{sinc}(t/T)$ y es la forma del pulso recibida en el detector como resultado de la aplicación de un impulso como estímulo para este sistema ideal.[1]

Basándose en lo anterior, Nyquist estableció que la mayor razón de transmisión de símbolos sin ISI para un sistema de ancho de banda $W = \frac{1}{2T}$ Hz es $R_s = \frac{1}{T}$ símbolos/s, en lo que se conoce como límite de ancho de banda de Nyquist. La razón por la que esto ocurre se muestra en la figura 2.3 donde se grafican dos pulsos con la forma $\text{sinc}(t)$ y se evidencia que los mínimos de las ondulaciones secundarias de cada pulso ocurren en el punto de máxima energía de los pulsos adyacentes.[1]

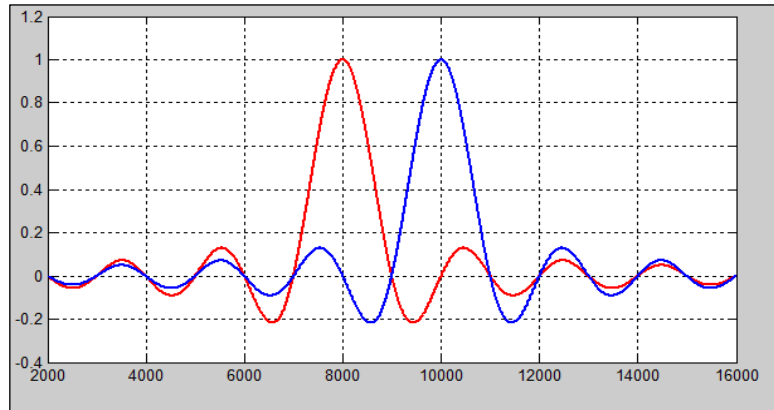


Figura 2.3 Pulsos $h(t)$ y $h(t - T)$ cuando $h(t) = \text{sinc}(t)$ sin ISI.

En la figura 2.3 $h(t)$ tiene valor cero en el punto de detección del pulso adyacente, $h(t - T)$, cuando el muestreo se realiza con sincronismo perfecto entre transmisor y receptor pero pequeños errores de sincronismo producen ISI. Además, analizando la función *sinc* con detenimiento se puede notar que:

- Tiene duración infinita y por tanto en cualquier implementación real se hace necesario truncar la función.
- Es una función no causal³, por tanto el pulso truncado sufrirá un retardo.
- Los angostos límites de banda de su respuesta de frecuencia son difíciles de aproximar.
- La amplitud de la función *sinc* decae demasiado lento, a una razón de $1/t$. [15]

Por estas razones, la implementación de la función *sinc* no es utilizada en la práctica y se busca otra función de transferencia que genere pulsos un poco más limitados en tiempo sin implicar grandes pérdidas en eficiencia espectral, buscando una relación de compromiso de modo que la forma del pulso cumpla con algunos requerimientos básicos tales como:

- Los cruces por cero ocurran en los puntos de muestreo de los pulsos vecinos.
- Los límites de banda de su respuesta de frecuencia sean suaves.
- La decadencia de la amplitud en el dominio del tiempo sea más rápida que en la función *sinc* manteniendo una respuesta de frecuencia relativamente estrecha. [15]

³ Un sistema es causal si, para cualquier elección de n_0 , el valor de la secuencia de salida en el índice $n = n_0$ depende solamente de los valores de la secuencia de entrada para $n \leq n_0$. [31] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed., Upper Saddle River, NJ: Prentice Hall, 1999.

Una función de transferencia de sistema usada con frecuencia por cumplir con estos requerimientos es la denominada filtro coseno alzado, que puede expresarse analíticamente como:

$$H(f) = \begin{cases} 1 & \text{para } |f| < 2W_0 - W \\ \cos^2\left(\frac{\pi}{4} \frac{|f| + W - 2W_0}{W - W_0}\right) & \text{para } 2W_0 - W < |f| < W \\ 0 & \text{para } |f| < W \end{cases}$$

El factor de roll-off se define como $r = \frac{W - W_0}{W_0}$ y especifica el “exceso de ancho de banda” del filtro en relación al límite de Nyquist, para $r = 0$ el filtro coseno alzado responde a este límite. Además, el factor de roll-off es una medida de la pendiente del filtro y modifica la expresión para la restricción de la razón de transmisión de modo que para una razón de transmisión R_S el ancho de banda necesario será $W = \frac{1}{2}(1 + r)R_S$. [1]

Aplicando Transformada Inversa de Fourier (FFT) a su función de transferencia, se obtiene que la respuesta al impulso del filtro coseno alzado es:

$$h(t) = 2W_0 \operatorname{sinc}(2W_0 t) \frac{\cos[2\pi(W - W_0)t]}{1 - 4(W - W_0)^2 t^2}$$

La figura 2.4 muestra la representación gráfica de ambas funciones para diferentes valores de r y evidencia que su comportamiento cumple con los requerimientos planteados anteriormente. Nótese que la función decae a una razón de $1/|t|^3$, mucho más rápido que el $1/t$ de la función sinc .

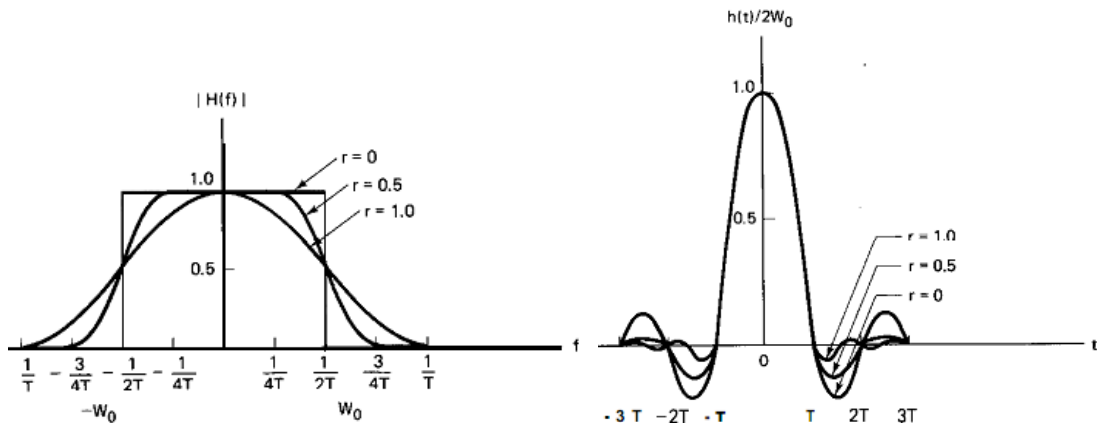


Figura 2.4 Características del filtro coseno alzado. a) Respuesta de frecuencia. b) Respuesta al impulso. [1]

Hasta el momento la conclusión en el diseño de filtros para evitar la ISI, es que a la entrada del detector se debe poner la forma de pulso de salida de un filtro coseno alzado puesto que ofrece el mejor compromiso entre eficiencia espectral y duración de los pulsos. Al inicio de este epígrafe se asume que el filtro en el transmisor está predeterminado y no se puede cambiar puesto que cuando se diseña un receptor este es el fenómeno más común. Sin embargo, en este trabajo se ha implementado un sistema transmisor-receptor y por tanto uno de los bloques a diseñar es, precisamente, el filtro del transmisor.

Si se tiene la posibilidad de diseñar ambos extremos, transmisor y receptor, la mejor opción es utilizar un par de filtros tal que el producto de sus funciones de transferencia –o la convolución de sus respuestas impulsivas en el dominio del tiempo– resulte en una forma del pulso a la entrada del detector que responda a la función coseno alzado. Cuando se desea este tipo de comportamiento en un sistema, la solución es utilizar la raíz de la función coseno alzado en el dominio de la frecuencia y definir una forma de pulso denominada raíz cuadrada de coseno alzado –o simplemente raíz de coseno alzado– cuya descripción en el tiempo se muestra a continuación.[1]

$$h_{srrc}(t) = \begin{cases} \frac{1}{\sqrt{T}} \frac{\sin\left[\pi(1-\beta)\frac{t}{T}\right] + \left(4\beta\frac{t}{T}\right)\cos\left[\pi(1+\beta)\frac{t}{T}\right]}{\left(\pi\frac{t}{T}\right)\left[1 - \left(4\beta\frac{t}{T}\right)^2\right]} & \text{para } t \neq 0, t \neq \pm\left(\frac{T}{4\beta}\right) \\ \frac{1}{\sqrt{T}} \left[1 - \beta + \left(4\frac{\beta}{\pi}\right)\right] & \text{para } t = 0 \\ \frac{\beta}{\sqrt{T}} \left[\left(1 + \frac{2}{\pi}\right)\sin\left(\frac{\pi}{4\beta}\right) + \left(1 - \frac{2}{\pi}\right)\cos\left(\frac{\pi}{4\beta}\right)\right] & \text{para } t = \pm\left(\frac{T}{4\beta}\right) \end{cases}$$

MATLAB contiene varias funciones a través de las cuales se pueden implementar filtros raíz de coseno alzado[29]. En el sistema descrito se empleó la función ‘rcosine’ para crear un filtro FIR especificando el factor de roll-off del filtro. El objeto creado con dicha función se utiliza como parámetro de entrada en la función ‘rcosflt’ para filtrar la señal. Este valor se puede cambiar accediendo al código del sistema y se establece por separado al transmisor y el receptor y debe ser el mismo para ambos.

Formación del Pulso

```
Rs = Fs/floor(Fs/Rs);           % Ajuste de Rs para hacer Fs/Rs entero
num = rcosine(Rs,Fs,'sqrt',r);   % Diseño del filtro
y = rcosflt(y,Rs,Fs,'filter',num); % Filtro raíz de coseno alzado
```

2.1.6 Modulación

La modulación –sea analógica o digital– es el proceso mediante el cual una señal de información es convertida a una forma de onda sinusoidal. Una senoide tiene tres características que la distinguen: amplitud, frecuencia y fase. Por tanto, la modulación puede definirse como el proceso mediante el cual la amplitud, fase o frecuencia de una portadora de RF, o la combinación de ellas, es variada de acuerdo a la información a ser transmitida.

La modulación analógica se divide en modulación de amplitud (AM), de fase (PM) y de frecuencia (FM). Cuando se estudia su funcionamiento, PM y FM son en esencia lo mismo y los autores las denominan modulación angular.[32]

En los sistemas de transmisión digital se utilizan técnicas de modulación digital. La diferencia básica entre la modulación analógica y la digital es que cuando se usa la última los períodos de T segundos representan un símbolo, mientras en la modulación analógica la idea del símbolo no tiene sentido por usar como moduladoras señales de información continuas en el tiempo.[1]

Las técnicas básicas de modulación digital son:

- Llaveo de Fase (PSK)
- Llaveo de Frecuencia (FSK)
- Llaveo de Amplitud (ASK)
- Modulación por Fase Continua (CPM)
- PSK Diferencial (DPSK)

De las mencionadas antes se derivan una serie de variaciones como Llaveo de Fase Mínimo (MSK), Modulación de Amplitud en Cuadratura (QAM), y otras. Una descripción exhaustiva los métodos de modulación mencionados se puede encontrar en textos como [1, 30, 33].

De todas las técnicas de modulación digital usadas la más sencilla de implementar a nivel de software es la PSK Binaria (BPSK). La característica principal de BPSK es el cambio de fase de π radianes de la portadora en dependencia de si la señal de información es un uno o un cero.

La modulación se realiza en dos etapas: Modulación por Amplitud del Pulso (PAM) en banda base y BPSK para elevar la frecuencia. La modulación PAM empleada convierte los bits a valores de la señal binaria obtenida de la información a -1, para el valor '0' del bit, y 1, para el valor '1' del bit. Hay diferentes formas de hacer esto, la más económica computacionalmente es utilizando un vector que posee los valores y luego utilizar los valores binarios de la señal para indicar el índice de dicho vector. El siguiente ejemplo ilustra lo anterior:

```
x = [0 1 1 1 1 0 0 0 0 1 1 1 1];  
s = [-1 1];
```

Entonces, la sentencia $y = s(x+1)$; devolvería los valores de la variable 'x' convertidos de la forma explicada en el párrafo anterior puesto que para un $x(n)=0$, $y(n)$ toma el valor de $s(1)$ y lo propio en caso de $x(n)=1$. En la implementación real se realiza la misma operación.

Modulación PAM

```
s = [-1,1];           % Constelación PAM  
y = s(x+1);          % Modulación Banda Base
```

Posteriormente se le da forma al pulso. En el epígrafe 2.1.5 se realiza la explicación de este procedimiento. Luego, la señal de salida del filtro de formación de pulso se utiliza como moduladora de un tono de portadora, cuya frecuencia puede variarse por software, para luego sacarla por la salida de audio del sistema operativo. Esta combinación de modulación PAM usando constelación [1, -1] y modulación de amplitud (AM) produce la onda modulada en PSK binaria (BPSK) que será transmitida.

Las características de los dispositivos de salida de sonido comunes en el mercado imponen ciertas restricciones para esta frecuencia de portadora. La salida de sonido estándar de las computadoras personales, ya sean éstas incluidas en el chipset o de una tarjeta de extensión para audio, tiene el ancho de banda limitado a 20KHz. Recordando el límite de ancho de banda de Nyquist se pueden deducir varias implicaciones de esta característica:

1. La máxima razón de transmisión teórica que se puede alcanzar es de 10Kbps.
2. La frecuencia de la portadora, f_c , tiene que cumplir con la condición:

$$\frac{1}{T_S} < f_c < 20\text{KHz} - \frac{1}{T_S} \text{ donde } T_S = \frac{1}{R_S}$$

3. El receptor requiere un ancho de banda mínimo de 20kHz .

Esta última consecuencia genera a su vez ciertas limitaciones en la computadora utilizada como receptor. Usualmente se utiliza la entrada de micrófono de la interfaz de sonido de la computadora como receptor para la comunicación vía audiofrecuencia. Esto es imposible en este sistema puesto que las entradas de micrófono, salvo raras excepciones, se diseñan con un ancho de banda de 4kHz , puesto que habilitar más sería derrocharlo debido a que la voz humana no excede ese límite. En consecuencia, se tiene que emplear la entrada de línea de la interfaz de sonido en el extremo receptor del sistema porque ésta sí está diseñada con un ancho de banda de 20kHz y la señal transmitida no se deteriora por causas de hardware.

2.1.7 Espectro Extendido

La aplicación inicial de técnicas de espectro extendido (SS) fueron los sistemas de comunicación militar, al punto que el final de la Segunda Guerra Mundial eran un concepto familiar para los ingenieros de radar. Posterior a esta aplicación inicial se desarrollaron investigaciones con el fin de aplicar estas técnicas en sistemas de comunicaciones fuera del ámbito militar por las ventajas que presentan. Se denominan espectro extendido porque el ancho de banda de transmisión empleado es mucho mayor que el mínimo requerido para transmitir la información.[30]

Un sistema se considera SS cuando cumple los siguientes requerimientos:

- La señal ocupa mucho ancho de banda en exceso del necesario para enviar la información.
- La extensión del espectro se logra a través de una señal de código que es independiente de los datos.
- En el receptor, la recuperación de los datos originales se logra por la correlación de la señal extendida recibida con una réplica sincronizada de la señal de código utilizada para expandir la información.

Algunos esquemas de modulación como FM y PCM también expanden el espectro de la señal de información pero no se clasifican como sistemas de espectro extendido por no satisfacer los requerimientos antes mencionados. Los atributos positivos de los sistemas

SS se resumen en supresión de interferencia, reducción de densidad de energía, buena resolución temporal y acceso múltiple.[1]

La supresión de interferencia se produce por la limitación de potencia de un sistema interferente. Es decir, cuando se expande la señal en el espectro, todo ente que intenta de interferir la señal queda en la disyuntiva de ampliar el espectro de la señal interferente, reduciendo la densidad espectral de interferencia, o dañar un ancho de banda menor pero con más potencia para cada componente de la banda. En cualquiera de los dos casos, el efecto de interferencia queda reducido.[1]

La reducción de densidad de energía se deriva del propio proceso de expansión de la señal porque ésta se esparce y disminuye su energía por cada componente del espectro. La mayor ventaja y aplicación de este atributo radica en la fabricación de sistemas de comunicaciones con baja probabilidad de detección (LPD), lo que se traduce en una gran dificultad para detectar si el sistema en funcionamiento o no para todo ente ajeno al mismo. Estos sistemas, aun suponiendo que fueran detectados, exhiben baja probabilidad de determinación de la posición (LPPF) y, en algunos casos, baja probabilidad de explotación de señal (LPSE), lo que significa que la identificación de la fuente es difícil. La otra aplicación que esta característica permite a los sistemas SS es el cumplimiento de regulaciones de potencia y operar sobre bandas que ya se encuentran en explotación.[1]

Las señales de SS también pueden utilizarse para determinar rango de distancia o posición espacial. La distancia se puede determinar midiendo el retardo de un pulso mientras atraviesa el canal. La inexactitud en la medición de retardo es inversamente proporcional al ancho de banda del pulso de señal. Todo esto se logra gracias a la buena resolución temporal que muestran los sistemas SS.[1]

Un atributo de los sistemas SS ampliamente aplicado es su capacidad de multiplexación. Solo por citar un ejemplo, los sistemas de Acceso Múltiple por División de Código (CDMA) permiten varios usuarios sobre el mismo canal asignando una secuencia de código para expandir la señal diferente para cada uno.[1]

El modelo para utilizar SS se basa en un principio relativamente sencillo. Multiplicando la señal de información por la señal de código una vez se logra extender el espectro de la señal de información, y multiplicando por segunda vez, con filtrado posterior, se

recupera la señal original. De este modo, la señal de información se multiplica dos veces y el ruido solo una.[1]

Existen dos métodos básicos para realizar SS: saltos en frecuencia (FH) y secuencia directa (DS). La primera generalmente va acompañada de modulación por Llaveo de Frecuencia M-ario (MFSK) pues $k = \log_2 M$ bits se utilizan para determinar una de las M frecuencias para transmitir esa secuencia. El espectro extendido por secuencia directa (DSSS) se realiza multiplicando la señal, como se explicaba antes, por una secuencia pseudo-aleatoria sincronizada en los extremos de la transmisión. Un tercer método podría ser saltos temporales (TH) pero esta técnica no expande el espectro sino reduce la razón de transmisión de datos transmitiendo a ráfagas espaciadas temporalmente atendiendo a algún código conocido solo por los autorizados a utilizar el sistema. Aunque estas técnicas suelen usarse por separado, también existen combinaciones híbridas, por ejemplo DS/FH, FH/TH y DS/FH/TH.[1, 30]

DSSS es la elección para este sistema puesto que computacionalmente es más económico y el grado de dificultad para manejar el ancho de banda a utilizar es menor. La idea detrás de DSSS es utilizar una señal pseudo-aleatoria con una razón de código mucho mayor que la señal de información para cambiar su fase dentro de un mismo símbolo, tal y como se muestra en la figura.[30]

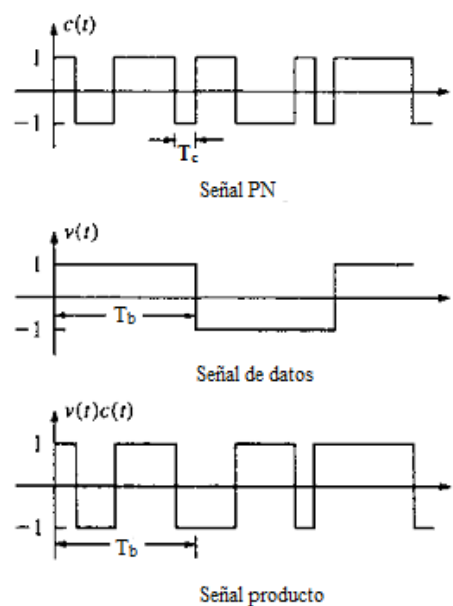


Figura 2.5 Espectro Extendido por Secuencia Directa.

Este es el método utilizado en nuestro sistema. El código muestra el procedimiento empleado para lograrlo. Se fija siempre la semilla para el generador de la secuencia de

código para poder utilizar la misma en el receptor sin necesidad de guardarla en memoria.

Espectro Extendido por Secuencia Directa

```
chip = 5; % Número de chips por bit de información
rep_x = rectpulse(x, chip); % Convierte cada bits de x en 5 iguales
RandStream.setDefaultStream(RandStream('mt19937ar', 'seed', 117));
pn_seq = randint(1, length(rep_x)); % Creación del vector PN
x = xor(rep_x, pn_seq);
```

2.2 Receptor

El receptor ha sido implementado como una función en cuya invocación se introducen como parámetros la razón de símbolos y la frecuencia de la portadora utilizada para la modulación. El diagrama en bloques del receptor se muestra a continuación.

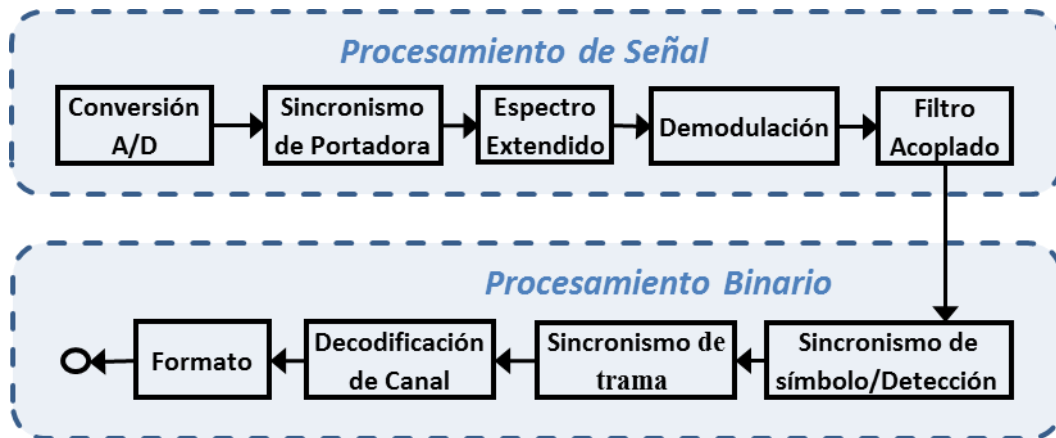


Figura 2.6 Diagrama en bloques del receptor

2.2.1 Conversión Análogo-Digital

La sección de conversión análogo-digital comprende la configuración del convertidor análogo/digital de la interfaz de sonido de Windows a través de MATLAB y la captura de la señal a través de ésta. Para la configuración de la interfaz de sonido se emplea un solo canal configurado para activarse por software cuando la señal sobrepase un nivel positivo de 0.1 volts y una duración máxima de 12 segundos. Este último es un parámetro que se utiliza para fijar un tiempo máximo para que el sistema esté en recepción. Si al agotarse el tiempo no se ha activado el sistema, la función aborta el proceso y emite un mensaje de error.

La captura de la señal es relativamente sencilla, solo es importante destacar que la función 'getdata' devuelve una columna y por tanto es necesario trasponer la matriz generada para convertirla en una fila, puesto que en todo el procesamiento ulterior de la señal ésta es tratada como tal.

Configuración Conversor Análogo/Digital de la Interfaz de Sonido

```
AI=analoginput('winsound');
chan=addchannel(AI,1);
duration=40;
SampleRate=Fs;
set(AI,'SampleRate',SampleRate);
set(AI,'SamplesPerTrigger',duration*SampleRate);
set(AI,'TriggerChannel',chan)
```

```
set(AI, 'TriggerType', 'Software')
set(AI, 'TriggerCondition', 'Rising')
set(AI, 'TriggerConditionValue', 0.1)
set(AI, 'Timeout', inf)

disp('Waiting .....');
start(AI);
data=getdata(AI);
disp('Processing .....');
y=data';
```

2.2.2 Sincronismo de Portadora

En un sistema de comunicaciones, aun si el canal de transmisión fuera ideal, la señal que llega al receptor es una forma de onda analógica complicada que ha de ser demodulada y muestreada antes de poder recuperar el mensaje. En el propio proceso de transmisión, propagación y recepción siempre ocurren desplazamientos de frecuencia y fase entre la portadora utilizada para la modulación y la que llega al receptor. Estos desplazamientos se deben, fundamentalmente, a la distancia entre los extremos del sistema, las inestabilidades relativas de los osciladores del transmisor y el receptor; y las velocidades relativas de los extremos, que pueden provocar desviaciones de frecuencia por efecto Doppler. Entonces, para que la demodulación tenga éxito, el receptor tiene que ser capaz de precisar la frecuencia y fase de la sinusoide utilizada para la modulación en el transmisor. El proceso de estimación de la frecuencia y la fase de la portadora recibe el nombre de Recuperación de Portadora.[34]

En este sistema de comunicación, como en muchos otros, es necesario realizar dos procesos de conversión de frecuencia: uno analógico y otro digital. El primero debe trasladar el espectro de la señal hasta una frecuencia intermedia y el segundo debe digitalizar la señal en esta frecuencia. La razón por la que, a nivel mundial, se hace de esta forma es el costo económico de hacer un convertidor analógico de frecuencia altamente preciso y el costo económico y computacional de muestrear a altas frecuencias sin reducir a frecuencias intermedias.

El compromiso utilizado en la amplia mayoría de los casos es emplear un convertidor analógico barato aunque no muy preciso para trasladar a frecuencia intermedia y luego emplear procesamiento digital para compensar las inexactitudes de fase de la portadora, puesto que esta última etapa es capaz de compensar el retardo de fase provocado por la transmisión y, además, cualquier inexactitud provocada por la porción analógica del receptor.[15]

Normalmente, el transmisor y el receptor acuerdan una frecuencia en particular para la portadora y, en un mundo ideal, la frecuencia de la portadora transmitida es exactamente la acordada. El problema aparece por fenómenos que desvían la frecuencia de la señal y el hecho de que incluso los osciladores caros tienen pequeños desplazamientos de frecuencia, por tanto los osciladores baratos se convierten en una necesidad y se requiere alguna forma de alinear las frecuencias en los extremos de la transmisión. Si lo que se desea es, precisamente, encontrar la componente de frecuencia fundamental de la portadora, la vía más obvia pareciera ser utilizar la Transformada de Fourier usando directamente Transformada Rápida de Fourier (FFT) para hallar la frecuencia y la fase de la portadora. Aunque parece un camino viable, este método se usa muy raramente debido al costo computacional que implica. La estrategia en este sentido ha sido sustituir la FFT por elementos adaptativos que logren un valor óptimo cuando la fase de una portadora estimada iguale la fase de la portadora real. Moviendo un estimado en la dirección del gradiente, este elemento puede llegar al valor correcto de forma recursiva. Asumiendo inicialmente que la frecuencia es conocida, existen varias formas de estructurar elementos adaptativos que estimen la fase de una portadora de forma iterativa.[15]

Una de estas funciones se basa en la utilización del cuadrado de la diferencia de fase entre la señal recibida y una senoide generada localmente. Este método consiste en minimizar el promedio del cuadrado de la diferencia entre una versión de una senoide en la frecuencia acordada entre los extremos del sistema y la senoide de la portadora recibida, trasladando la primera al doble de la frecuencia a la vez que duplica su fase en el proceso. Esta señal sería:

$$r_p(t) = \cos(4\pi f_0 t + 2\varphi)$$

Y la función del método:

$$J_{SD}(\theta) = \text{avg}\{e^2(\theta, k)\} = \frac{1}{4} \text{avg}\{(r_p(kT_s) - \cos(4\pi f_0 T_s + 2\theta))^2\}$$

La razón por la que el método funciona es que si se puede encontrar un θ tal que $\varphi = \theta + n\pi$, entonces el valor de la función sería cero y las frecuencias se sincronizan perfectamente.[15]

Otro método muy empleado para la recuperación de portadora es el conocido Lazo de Fase Cerrada (PLL). Por el conocimiento y amplio estudio que de este método existe, no se discute aquí su funcionamiento e implementación. Su funcionamiento puede encontrarse en una gran variedad de libros de electrónica y procesamiento digital de señales, y una implementación en MATLAB aplicada a la recuperación de portadora puede encontrarse en [15].

El método escogido para la elaboración del sistema de comunicaciones que se describe en este trabajo es el llamado Lazo de Costas. Los algoritmos PLL y Diferencia Cuadrada constituyen dos formas de sincronizar la fase en el receptor con la fase en el transmisor. Ambos requieren pre-procesamiento de la señal recibida para extraer una versión "limpia" de la portadora, y además lo hacen al doble de la frecuencia y fase.

El Lazo de Costas opera directamente sobre la señal recibida, $r(kT_s) = s(kT_s)\cos(2\pi f_0 kT_s + \varphi)$, a través de la inversión del orden de procesamiento: primero modular a DC, luego hacer filtrado paso bajo, y finalmente elevar al cuadrado.[15]

Este orden de las operaciones en reversa lleva a la función de desempeño conocida como Lazo de Costas:

$$J_c(\theta) = \text{avg}\{\text{LPF}(r(kT_s) \cos(2\pi f_0 kT_s + \theta))^2\} \quad (2.5)$$

Por la forma en que esta función utiliza las no-linealidades, ésta puede operar sin hacer procesamiento previo de la señal. Detallando la afirmación anterior, se puede sustituir $r(kT_s)$ en (2.5), entonces:

$$J_c(\theta) = \text{avg}\{\text{LPF}(s(kT_s) \cos(2\pi f_0 kT_s + \varphi) \cos(2\pi f_0 kT_s + \theta))^2\} \quad (2.6)$$

El filtro paso bajo no es parte del lazo de control, por tanto no es necesaria una respuesta de frecuencia que se encuentre dentro del ancho de banda del lazo. Su único propósito es eliminar el exceso de ruido generado por los multiplicadores de frecuencia anteriores a su operación, por tanto su frecuencia de corte inferior es menor que $2f_0$. Consecuentemente, las componentes del doble de la frecuencia de portadora pueden ser eliminadas de la ecuación.[35]

Aplicando lo anterior y algunas identidades trigonométricas⁴, se llega a que:

$$\text{LPF}\{s(kT_s) \cos(2\pi f_0 kT_s + \varphi) \cos(2\pi f_0 kT_s + \theta)\} = \frac{1}{2} s(kT_s) \cos(\varphi - \theta) \quad (2.7)$$

Sustituyendo (2.7) en (2.5) y resolviendo el cuadrado la función queda:

$$J_C(\theta) = \frac{1}{4} s_{\text{avg}}^2 \cos^2(\varphi - \theta) \quad (2.8)$$

Por tanto $J_C(\theta)$ es dependiente de $\cos^2(\varphi - \theta)$ y tiene máximos en cada punto en que $\theta = \varphi + n\pi$ para todo $n \geq 0$.

En la práctica el promedio puede ser eliminado si se tiene en cuenta el proceso de promedio que realiza el filtro paso bajo. Teniendo en cuenta esto, el Lazo de Costas puede implementarse como un elemento adaptativo usando el siguiente algoritmo:

$$\theta[k+1] = \theta[k] - \varepsilon \text{LPF}\{r(kT_s) \cos(2\pi f_0 kT_s + \theta[k])\} \text{LPF}\{r(kT_s) \sin(2\pi f_0 kT_s + \theta[k])\}$$

donde ε es el tamaño del incremento del algoritmo en cada iteración.[15]

Esta es la expresión matemática del algoritmo implementado para Recuperación de Portadora en el sistema creado puesto que brinda mayor robustez en presencia de ruido que el PLL y es computacionalmente más eficiente. La explicación ampliada de los métodos descritos aquí se puede encontrar en [15]. La implementación realizada en este trabajo se muestra a continuación.

Sincronismo de Portadora por Lazo de Costas

```
t = (0:length(y) - 1)/Fs;           % Puntos de muestreo
fle=64;                             % Orden del filtro
h=fir1(fle,0.001);                  % Diseño del filtro Paso Bajo
mu=0.01;                             % Tamaño del paso del algoritmo
theta=zeros(1,length(t));           % Inicialización del vector de
                                     % estimación
theta(1) = -1;                       % Fase inicial
zs=zeros(1,fle+1);zc=zeros(1,fle+1); % Inicialización de los buffers
                                     % para el filtro Paso Bajo
for k=1:length(t)-1
```

⁴ $\cos^2(x) = \frac{1}{2}(1 + \cos(2x)); \cos(x+y) = \cos(x)\cos(y) - \sin(x)\sin(y);$

$\cos(x)\cos(y) = \frac{1}{2}[\cos(x-y) + \cos(x+y)]$

```
zs=[zs(2:file+1), 2*y(k)*sin(2*pi*Fc*t(k)+theta(k))];  
zc=[zc(2:file+1), 2*y(k)*cos(2*pi*Fc*t(k)+theta(k))];  
lpfs=fliplr(h)*zs'; % Nueva salida de los filtros  
lpfc=fliplr(h)*zc'; % Nueva salida de los filtros  
theta(k+1)=theta(k)-mu*lpfs*lpfc; % Actualización del algoritmo  
end
```

2.2.3 Espectro Extendido por Secuencia Directa

Para recuperar la señal de datos que llega expandida en frecuencia desde el transmisor el procedimiento es multiplicarla por la misma secuencia pseudo-aleatoria utilizada en el transmisor para expandirla. El procedimiento multiplica también el ruido pero si se analiza el proceso completo desarrollado en los dos extremos, se aprecia que la señal de datos ha sido multiplicada dos veces, una en el transmisor y otra en el receptor, mientras el ruido solo es multiplicado en el receptor.[1]

Es importante analizar en el receptor la polaridad de la señal que llega pues las transformaciones debidas al uso de técnicas de espectro extendido no tienen que ser simétricas en los extremos en cuanto al punto en que se hace el proceso. Por ejemplo, en esta implementación se expande la señal en un punto del transmisor en que ésta tiene una forma de onda unipolar; y en el receptor se completa la técnica en un punto anterior a la detección y, por consiguiente, la señal tiene forma bipolar. De hecho, la aplicación de DSSS en el transmisor funciona como detector integrando los valores según la cantidad de *chips* utilizados para cada bit.

El otro aspecto importante en las técnicas de espectro extendido es la sincronización de la señal pseudo-aleatoria usada en el receptor con la generada en el transmisor. Para eso se correlaciona la secuencia de entrenamiento para el sincronismo de símbolo, procesada cual si hubiera sido transmitida, con la señal de entrada. Así se elimina el ruido que precede a la señal que contiene la información y cuando se efectúa la multiplicación por la secuencia pseudo-aleatoria la señal que se tiene comienza en el mismo punto que en el transmisor.

En el sincronismo es necesario tener en cuenta que las secuencias pseudo-aleatorias empleadas en los dos extremos sean iguales. Esto se logra poniendo la semilla del generador aleatorio de MATLAB en el mismo valor en ambos extremos del sistema justo antes de generar la secuencia. Véase el siguiente código para un mejor entendimiento del proceso.

Sincronismo de Señal

```
s_inic = upsample(ones(1,20),2);      % Secuencia de entrenamiento
s = [-1,1];
s_inic = s(s_inic+1);
Rs = Fs/floor(Fs/Rs);                % Ajuste para hacer Fs/Rs entero
num = rcosine(Rs,Fs,'sqrt',r);        % Diseño del filtro
s_inic = rcosflt(s_inic,Rs,Fs,'filter',num)'; % Filtro raíz cuadrada
                                         % de coseno alzado
t_aux = (0:length(s_inic) - 1)/Fs;    % Puntos de muestreo
s_inic = s_inic.*cos(2*pi*Fc*t_aux);    % Modulación
[a,i] = max(xcorr(s_inic,double(y)));
i = length(y) - i;
y = y(i+1:end);
```

Espectro Extendido por Secuencia Directa

```
RandStream.setDefaultStream(RandStream('mt19937ar','seed',117));
pn_seq = randint(1,length(x));        % Inicialización del vector PN
x = xor(x,pn_seq);
no_bits = floor(length(x)/chip);
x = x(1:(floor(length(x)/3)*3));
x_temp = zeros(1,6);
for i=1:no_bits
    x_temp(i) = intdump(double(x(1:chip)),chip);
    x = x((chip+1):end);
end;
x = x_temp;
x = x > 0;
```

La última línea del código anterior se emplea en dependencia del tipo de decodificación que se use, *soft* o *hard*. En caso de usarse *soft decision decoding* no se incluye esta línea y los valores que se pasan al decodificador provienen del proceso de integración. Si se usa *hhard decision decoding*, se incluye esa sentencia y se pasan valores binarios al decodificador.

2.2.4 Demodulación

El proceso de demodulación consiste en la traslación en frecuencia a banda base utilizando un tono de la frecuencia de la portadora y el filtro acoplado para eliminar la componente de suma de frecuencias.

2.2.5 Filtro Acoplado

En la explicación desarrollada en el epígrafe 2.1.5 se arriba a la conclusión de que para evitar la ISI cuando se tiene control sobre el filtrado en el transmisor y en el receptor, la práctica más adecuada es el empleo de filtros raíz de coseno alzado en la formación del

pulso en el transmisor y en el filtro previo al detector en el extremo receptor. Así se ha implementado en el sistema creado como parte de la realización de este trabajo.

Para filtrar la señal de entrada se utilizan las funciones ‘rcosine’, con el parámetro ‘sqrt’, para crear el filtro raíz de coseno alzado y ‘rcosflt’ para filtrar la señal utilizando el filtro anterior. A la salida de este último se obtiene una señal con la menor ISI posible, lista para ser empleada por el detector en la recuperación de la información.

2.2.6 Sincronismo de Símbolo

Debido al ruido inherente a los sistemas de comunicaciones, la forma de onda obtenida después de la demodulación sufre deformaciones provocadas por los efectos del ruido. Además, los relojes de los extremos del sistema de transmisión son difíciles de sincronizar, aun cuando se tenga la frecuencia de portadora corregida; y el filtrado introduce retardos. Por tanto, se hace necesario crear algún mecanismo para muestrear cada símbolo en el punto de mayor energía. Este proceso se conoce como sincronización de reloj o sincronización de símbolo.[30]

Los sincronizadores de símbolo se dividen en dos grupos básicos. Los que integran el primero de estos grupos se conocen como sincronizadores de lazo abierto y su principio de funcionamiento es la recuperación de una réplica de la salida del reloj del transmisor a través de operaciones en la cadena de datos de entrada. El segundo se denomina sincronizadores de lazo cerrado y se basa en la comparación de mediciones de la señal de entrada para llevar a sincronismo una señal generada localmente con las transiciones de los datos entrantes.[1]

Según Sklar en [1], en la implementación en hardware de estos mecanismos los métodos de lazo cerrado tienden a ser más precisos pero mucho más complejos y costosos. Sin embargo, los métodos de sincronización de símbolo de lazo abierto tienen un inevitable error de rastreo de promedio no cero. Este error puede hacerse pequeño para razones de señal a ruido (SNR) grandes, pero mientras la forma de onda de la señal de sincronización dependa directamente de la señal de entrada el error nunca desaparecerá.

Además, se han logrado implementaciones de métodos de lazo cerrado a nivel de software con una muy buena eficiencia computacional. Por estas razones, se decide

emplear un método de lazo cerrado para el sincronismo de símbolo del sistema implementado en la realización de este trabajo.

Uno de los métodos más utilizados, tanto en implementaciones de hardware como de software, es el denominado *early/late gate*. Cuando se hace usando hardware, el principio de funcionamiento utilizado es la integración de la energía de la señal en dos porciones diferentes de duración $(T - d)$ segundos del intervalo de símbolos, tal que $d < T$ segundos. La primera integración (*early gate*) comienza en el punto de inicio del símbolo estimado por el lazo e integra por los siguientes $(T - d)$ segundos. La segunda integral (*late gate*) demora el inicio de la integración por d segundos y luego integra hasta el final del período de símbolo (el tiempo nominal T). La diferencia del valor absoluto de la salida de estas dos integraciones es una medida del error de sincronismo del receptor, y se realimenta a la referencia de tiempo del lazo para corregirla. Existen numerosas variantes de este método, tal y como se ha explicado hasta ahora, que difieren fundamentalmente en la elección del valor para d . [1, 36]

Sin embargo, las implementaciones a nivel de software de este método tienden a evitar el proceso de integración usando muestras individuales separadas entre sí por intervalos cercanos a la mitad de la duración del pulso, generalmente menores. Los valores absolutos de estas muestras se comparan y, en dependencia del resultado de la comparación, se desplaza el punto de muestreo. A partir de este funcionamiento básico se desglosan dos variantes fundamentales: una que desplaza el punto de muestreo iterativamente en un mismo símbolo hasta que la diferencia rebase un error predeterminado durante la implementación del algoritmo, buscando el punto de muestreo óptimo para cada símbolo, y luego salta al siguiente para realizar el mismo proceso; y otra en la que la idea es utilizar la comparación para desplazar o no, el punto de muestreo una cantidad de muestras definida durante el diseño del algoritmo, para luego seguir hacia el siguiente símbolo sin iterar sobre cada uno de ellos. [37]

Esta última variante hace necesario introducir una secuencia de entrenamiento antes de la cadena de datos de forma que, cuando el algoritmo comience a procesar la sección de la señal donde se encuentran los datos, el punto óptimo de muestreo haya sido corregido; siendo mucho más rápida y eficiente, y menos costosa computacionalmente. Por estas razones es la utilizada en el sistema descrito en este capítulo. Su implementación se muestra a continuación.

Sincronismo de Símbolo por Early/Late Late

```
st = ceil((Fs/Rs)); t = st;           % Punto de muestreo
d = 1;                               % Tamaño del paso del algoritmo
err = 0.01;                           % Error permitido
si = ones(1,ceil(length(y)/(st-1))); % Índices de muestreo
for i = 1:length(si)
    dif=abs(y(t-d))-abs(y(t+d));
    if dif > err
        t = t - 1;
    elseif dif < -err
        t = t + 1;
    end
    si(i) = t;
    t = t + st;
    if t > length(y) - st + 1
        break
    end
end
si = si(~(si > length(y)));
y = y(si);                           % Tomando los valores de la señal
                                     % en los puntos de muestreo
```

La sección de sincronismo de símbolo y detección ha sido implementada de modo que el algoritmo para determinar los puntos de muestreo los va almacenando en un arreglo, que luego es utilizado para tomar solamente las muestras de la señal correspondientes a dichos puntos.

Una vez que se tienen las muestras se procede al proceso de detección. La forma en la que se lleva a cabo este proceso está fuertemente ligada a la decodificación de canal que se haga. En caso de utilizar el algoritmo de Viterbi con *hard decision* se verifica la condición de que sean mayores que cero y la salida de este paso es la secuencia binaria, realizándose así la detección y la demodulación PAM en el mismo paso. Si se emplea *soft decision*, entonces las muestras son cuantificadas con 512 niveles –9 bits– para usarlo como entrada del decodificador de Viterbi.

2.2.7 Decodificación de Canal

Cuando se realiza la decodificación de canal en los sistemas de comunicaciones digitales, el objetivo principal es reducir la probabilidad de error de bit y para lograrlo se utiliza un concepto denominado máxima verosimilitud. Este concepto se traduce en un algoritmo que compara la secuencia recibida, Z , con las posibles secuencias transmitidas, $U^{(m)}$, a través del cálculo del máximo de las probabilidades condicionales de modo que la elección del decodificador, $U^{(m')}$, estará dada por la ecuación:

$$P(Z|U^{(m')}) = \max P(Z|U^{(m)}) \quad \forall U^{(m)}$$

Esto no es más que la traducción de la forma que dicta el sentido común de escoger la secuencia transmitida más parecida a la que se recibió. Cuando esto se aplica a los decodificadores convolucionales entonces se toma la secuencia $U^{(m')}$ en la que $P(Z|U^{(m')})$ sea mayor. Este decodificador minimiza la probabilidad de error de bit y se conoce como decodificador de máxima verosimilitud.[1]

Normalmente, los decodificadores convolucionales se representan a través de diagrama de árbol o de rejilla para su mejor comprensión e implementación computacional. El problema ahora consiste en escoger el camino a través del árbol o la rejilla tal que esta probabilidad sea la máxima. Si se usa un diagrama de árbol, entonces el número de secuencias posibles con secuencias de L símbolos es 2^L . Por tanto emplear decodificación por máxima verosimilitud implica comparación por "fuerza bruta" de la secuencia de longitud 2^L con todas las posibles secuencias transmitidas. Por tanto no es práctico utilizar este tipo de decodificación con diagrama de árbol por su costo computacional.[1]

Por otra parte, los decodificadores basados en diagrama de rejilla descartan todos los caminos que van resultando imposibles y luego escoge la mejor opción dentro del reducido grupo de caminos sobrevivientes. Este método no tiene desventaja alguna en relación al de árbol pues el camino escogido es el mismo que si se usara "fuerza bruta", mientras reduce la complejidad de decodificación al eliminar, previo a la comparación, los caminos no elegibles.[1]

Existen muchos algoritmos que obtienen soluciones aproximadas al problema de decodificación por máxima verosimilitud. Todos se ajustan a determinadas aplicaciones especiales pero todos son no óptimos. En cambio, el algoritmo de decodificación de Viterbi ofrece una solución completa a este problema y, por tanto, se considera óptimo.[38]

El algoritmo de Viterbi reduce el costo computacional aprovechando la estructura del diagrama de rejilla. La ventaja principal de este algoritmo en relación al método de "fuerza bruta" es que la complejidad del decodificador de Viterbi no es una función del número de símbolos en una secuencia de palabra de código. Su principio de funcionamiento es el cálculo de una medida de similitud entre la señal recibida, en el

instante t_i , y todas las trayectorias de malla entrando en cada estado en t_i . Cuando dos trayectorias entran al mismo estado, se escoge el que mejor métrica tenga como "sobreviviente" al siguiente estado. La selección se realiza en todos los estados hasta un punto predeterminado en la configuración del decodificador denominado profundidad de decodificación.[1]

La eliminación de trayectorias no óptimas en cada estado reduce el costo computacional del algoritmo de Viterbi, elimina la dependencia que tiene la complejidad del decodificador de la longitud de la palabra de código, y relaciona la complejidad del decodificador y el costo computacional con la profundidad de decodificación.[1]

El demodulador banda base de un receptor de comunicaciones convierte la señal introducida en él, $\{z(T)\}$, en una secuencia de código, $z(T)$, y la pasa al decodificador. La salida del demodulador puede configurarse de diferentes formas. Puede hacerse de modo que la decisión sea estricta (*hard decision*) sobre si $z(T)$ representa un cero o un uno. En este caso la salida del demodulador está cuantificada en dos niveles, cero y uno, y por tanto, al ser pasada de este modo, la decodificación se conoce como *hard decision decoding*. [1]

El demodulador también puede configurarse para introducir en el decodificador un valor cuantificado de $z(T)$. Esta implementación provee al decodificador más información que con *hard decision* al ofrecerle una medida de confiabilidad del valor muestreado en cada símbolo. Cuando el número de niveles de cuantificación a la salida del demodulador es mayor que dos, entonces se está realizando lo que se denomina *soft decision decoding*. Por ejemplo, la ganancia de codificación asociada a la utilización de *soft decision*, con 3 bits de cuantificación, es de 2 dB en relación a *hard decision*. [1, 29]

No obstante, la decodificación por *soft decision* no siempre se puede utilizar debido a la presencia de otras funciones dentro del receptor, que puede implicar que al decodificador lleguen valores de cero y uno inevitablemente. Un ejemplo de sistema en que esto ocurre es el estándar GSM donde el empleo de intercaladores de bloques obliga a la utilización de *hard decision*. [39]

La función 'vitdec' de MATLAB realiza la decodificación usando el algoritmo de Viterbi y puede configurarse para utilizar los dos modos de decisión y en dependencia de la elección se configura el modo de detección. [29]

La decodificación por *soft decisión* se muestra a continuación.

Soft Decision Decoding

```
x=x+1; % Garantizar x(i) > 0  $\forall i \in \mathbb{Z}$ 
partition=(0.00391389:0.00391389:2); % Niveles de cuantificación
codebook=(0:511);
[index, x]=quantiz(x, partition, codebook);
trel = poly2trellis([5 4],[23 35 0; 0 5 13]);
x = vitdec(x, trel, 25, 'trunc', 'soft', 9);
```

El proceso utilizado con *hard decision decoding* es mucho más sencillo aunque no muestra la robustez del *soft decisión decoding*. A continuación se muestra el código en MATLAB utilizado para *hard decision decoding* en el sistema SDR implementado.

Hard Decision Decoding

```
x = x > 0; % Decisión
trel = poly2trellis([5 4],[23 35 0; 0 5 13]); % Decodificador
x = vitdec(x, trel, 25, 'trunc', 'hard'); % Decodificación de Viterbi
```

CAPÍTULO 3. PRUEBAS EXPERIMENTALES Y RESULTADOS

En el desarrollo de este capítulo se mostrarán las pruebas experimentales realizadas con el sistema creado probando los métodos empleados más importantes, en especial el lazo de Costas, el algoritmo *early/late gate* y la codificación convolucional con decodificador de Viterbi. Se inicia con un análisis de probabilidad de error de bit en función de la SNR por constituir una herramienta esencial e imprescindible para caracterizar los sistemas de comunicaciones digitales. También se describirá el rendimiento del sistema utilizando como herramienta de estudio pruebas realizadas ejecutando MATLAB en una misma computadora, para tener control total de las afectaciones que puede recibir la señal, y usando diferentes medios de comunicación como: cable de cobre, espacio libre con audiofrecuencias, y espacio libre usando transmisión de FM.

Las características de estos medios de propagación no son objeto de estudio de este trabajo. El análisis de éstos quedará reducido a su comparación atendiendo al nivel de degradación que la señal sufre cuando pasa a través de ellos. La comparación no pretende ser exhaustiva, por tanto se basará en los resultados empíricos obtenidos en las abundantes pruebas realizadas durante la realización de este trabajo.

3.1 Probabilidad de Error de Bit

La probabilidad de error de bit (P_B) en un sistema de transmisión digital está condicionada por el umbral de decisión utilizado en la detección y la varianza del ruido. El umbral de detección óptimo para una señal digital es:

$$\gamma_0 = \frac{a_1 + a_2}{2}$$

La función para hallar P_B dada por:

$$P_B = \int_{\gamma_0}^{\infty} \frac{1}{\sigma_0 \sqrt{2\pi}} e^{\left[-\frac{1}{2} \left(\frac{z-a_2}{\sigma_0}\right)^2\right]} dz \text{ donde } z \text{ es el valor muestreado del símbolo y } \sigma_0^2 \text{ es la varianza del ruido. [1]}$$

Si se asume $\mu = \frac{z-a_2}{\sigma_0}$, entonces $\sigma_0 d\mu = dz$, por tanto:

$P_B = \int_{\mu=\frac{a_1-a_2}{2\sigma_0}}^{\mu=\infty} \frac{1}{\sqrt{2\pi}} e^{\left[-\frac{\mu^2}{2}\right]} d\mu = Q\left(\frac{a_1-a_2}{2\sigma_0}\right)$ donde $Q(x)$ es la llamada función complementaria de error, símbolo comúnmente usado para la probabilidad bajo la cola de una distribución gaussiana, y está definida como:

$$Q = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{\left[-\frac{\mu^2}{2}\right]} d\mu$$

Cuando se aplica filtro acoplado, la SNR está dada por:

$SNR = \frac{(a_1-a_2)^2}{\sigma_0^2} = \frac{2E_d}{N_0}$ donde $\frac{N_0}{2}$ es la densidad espectral de ruido a la entrada del filtro y E_d es la energía de la señal diferencial en la entrada del filtro.

Entonces $P_B = Q\left(\sqrt{\frac{E_d}{2N_0}}\right)$ para cualquier sistema digital con filtro acoplado y para el caso de señalización bipolar $P_B = Q\left(\sqrt{\frac{E_b}{N_0}}\right)$ y $SNR = \frac{E_b}{N_0}$, donde E_b es la energía promedio del bit.[1]

La medida de calidad más empleada para los sistemas digitales de transmisión es la relación $\frac{E_b}{N_0}$ vs P_B . Normalmente, esta relación se describe usando una gráfica logarítmica que brinda información sobre el desempeño del sistema. La figura 3.1 muestra la forma general de dichas gráficas. Cuando se trazan dos de sobre los mismos ejes, se pueden obtener dos informaciones muy importantes: la diferencia de probabilidad de error para igual SNR (comparando verticalmente), y la diferencia de SNR para una misma probabilidad de error (comparando horizontalmente).

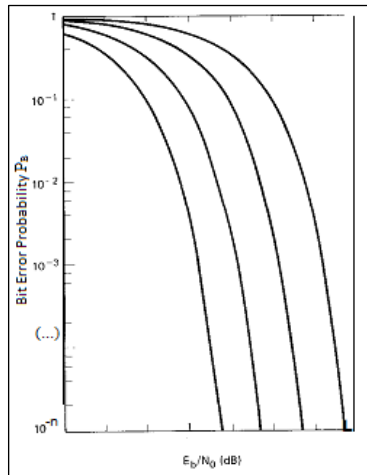


Figura 3.1 Forma general de gráficas de $\frac{E_b}{N_0}$ vs P_B

Las pruebas realizadas al sistema arrojan resultados consistentes con la realidad teórica. La figura 3.2 muestra las gráficas de la P_B teórica, la real del sistema sin codificación de canal y la del sistema usando código convolucional con decodificación de Viterbi.

En la gráfica se puede apreciar la ganancia de codificación asociada a la codificación convolucional utilizada y la correspondencia de la gráfica empírica del sistema y la teórica usando la función complementaria de error.

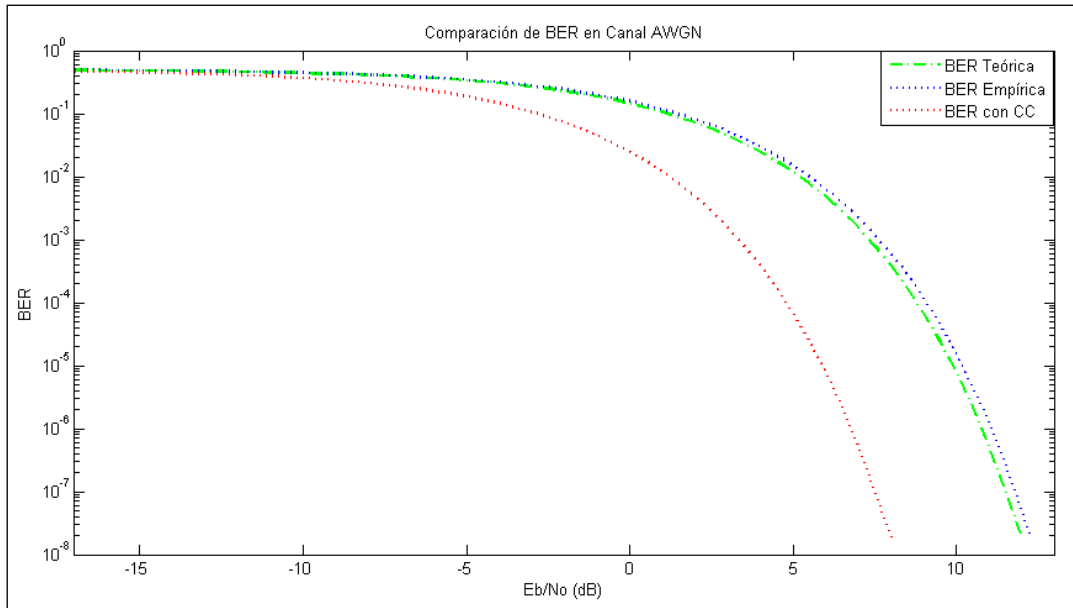


Figura 3.2 Gráfica de Razón de Error de Bit (BER) vs E_b/N_0

Por lo resultados de las pruebas experimentales se puede concluir que el sistema SDR implementado cumple con los requerimientos necesarios para su empleo futuro en los laboratorios de Fundamentos de las Comunicaciones. Además, los algoritmos implementados demuestran ser una elección acertada para el sistema atendiendo a su desempeño en la aplicación para la está fue diseñado.

3.2 Lazo de Costas

Como se describió en el epígrafe 2.2.2, el lazo de Costas se implementa para lograr que la fase de la portadora recibida y la de otra generada localmente sean iguales o separadas por un múltiplo entero de π . Esto se traduce en la perfecta sincronización de las frecuencias de estas señales necesaria para lograr una demodulación que permita la recuperación de la señal adecuadamente.

En el capítulo anterior solo se hizo una introducción teórica de este método. Ahora se presentan pruebas experimentales realizadas con el lazo de Costas variando el punto de

partida del algoritmo y la desviación de fase. Normalmente, los experimentos realizados para comprobar el funcionamiento del lazo de Costas no varían la desviación de fase, sin embargo en este trabajo pretendemos demostrar que su empleo logra, efectivamente, el sincronismo de la portadora recibida con otra generada localmente basada en el parámetro de frecuencia de portadora introducido a la función del receptor.

A continuación se muestra la gráfica de una simulación en la que se varió el punto de partida del lazo entre 4 y -4 para verificar que se ata a la desviación de fase introducida, en este caso 0.5, o a valores separados de ésta un múltiplo entero de π . El eje vertical representa la fase y el horizontal el tiempo de señal analizado. Se puede constatar la calidad del algoritmo si se analiza el tiempo que demora en sincronizar las frecuencias de las señales. Para ello nótese el tiempo que demora en aproximarse a los valores adecuados.

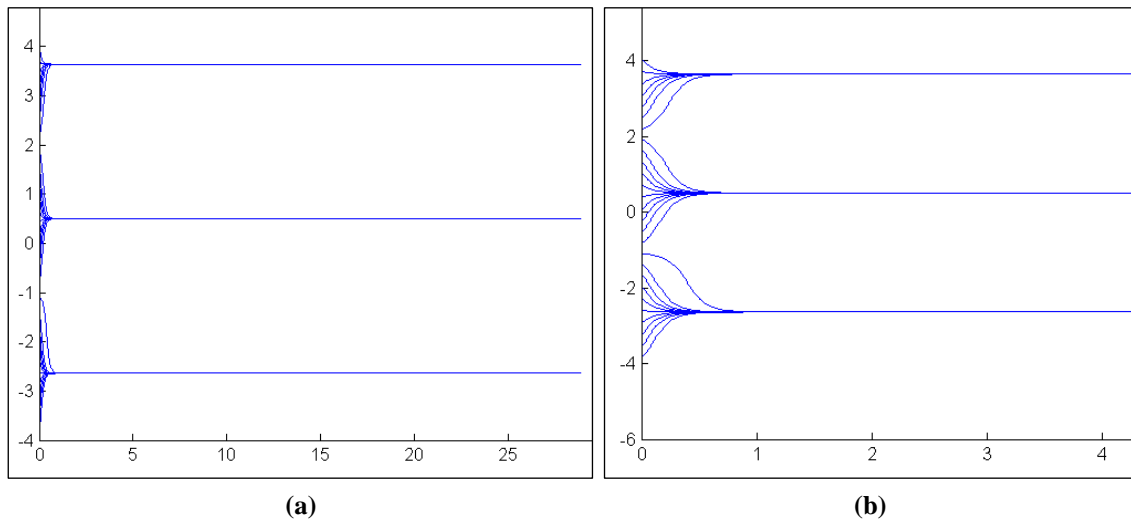


Figura 3.3 Desviación de fase instantánea de la portadora a medida que se sincroniza usando Lazo de Costas. a) Señal completa. b) Señal de 0 a 5 segundos.

Al hacer un acercamiento a la imagen graficando solo para los tiempos menores de 5 segundos se puede apreciar la aguda pendiente que este algoritmo genera en el desplazamiento de fase en los primeros instantes. Por esta disminución brusca de la desviación de fase se alcanza el valor cero en menos de un segundo.

No obstante, es importante aclarar que este primer segundo de señal no es demodulada con este error relativo sino que dicho error es sumado como variación de fase al coseno usado para demodular la señal. De este modo, la señal queda demodulada como si hubiera estado sincronizada desde que se recibió.

Por ejemplo, véase la gráfica de la figura 3.4 para verificar las desviaciones de fase de la portadora con $SNR = 10 \text{ dB}$. El tiempo de caída del algoritmo sigue siendo menor a un segundo.

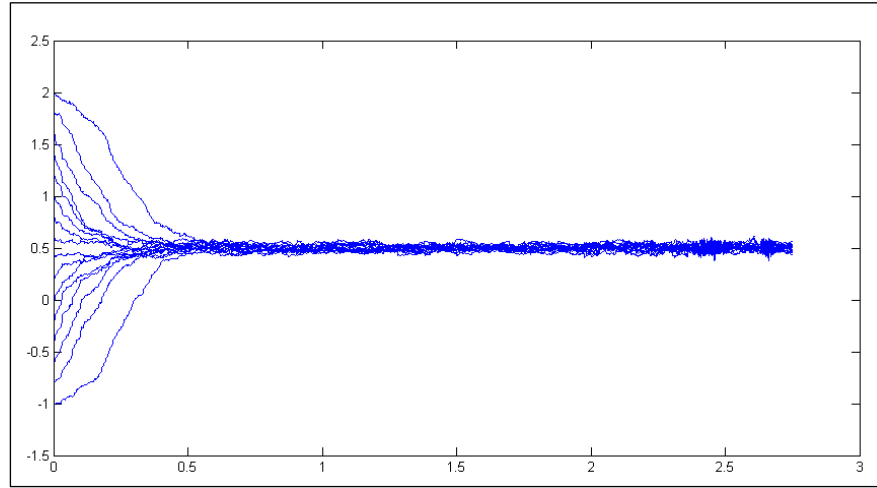


Figura 3.4 Desviación de fase instantánea de la portadora a medida que se sincroniza usando Lazo de Costas para $SNR = 10 \text{ dB}$

El lazo de Costas implementado demuestra sincronizar la frecuencia de la señal recibida a través de la corrección de la desviación de fase.

3.3 Algoritmo *Early/Late Gate*

Durante la explicación del funcionamiento del algoritmo *early/late gate* se planteó que, por la implementación que se realizó en este sistema, se hacía necesario el empleo de una secuencia de entrenamiento para poner el muestreo en sincronismo desde el inicio de los datos. Esto no es más que el pequeño precio a pagar por la utilización de una implementación de *early/late gate* tan ligera computacionalmente.

Por las pruebas realizadas, el sincronismo de símbolo muestra un muy buen funcionamiento. Sin importar el ruido precedente al inicio de la señal, el algoritmo siempre logra el sincronismo en la secuencia de entrenamiento.

En la figura 3.5a se muestra el desempeño del algoritmo implementado con la señal limpia. La forma de onda que se aprecia es la secuencia de entrenamiento y parte del inicio de la trama de datos. Los círculos rojos representan el punto de muestreo en cada iteración del algoritmo. Nótese cómo estos puntos se van acercando al máximo del valor absoluto de cada símbolo a pesar de tener un inicio difícil por la influencia del ruido de las muestras precedentes a la secuencia.

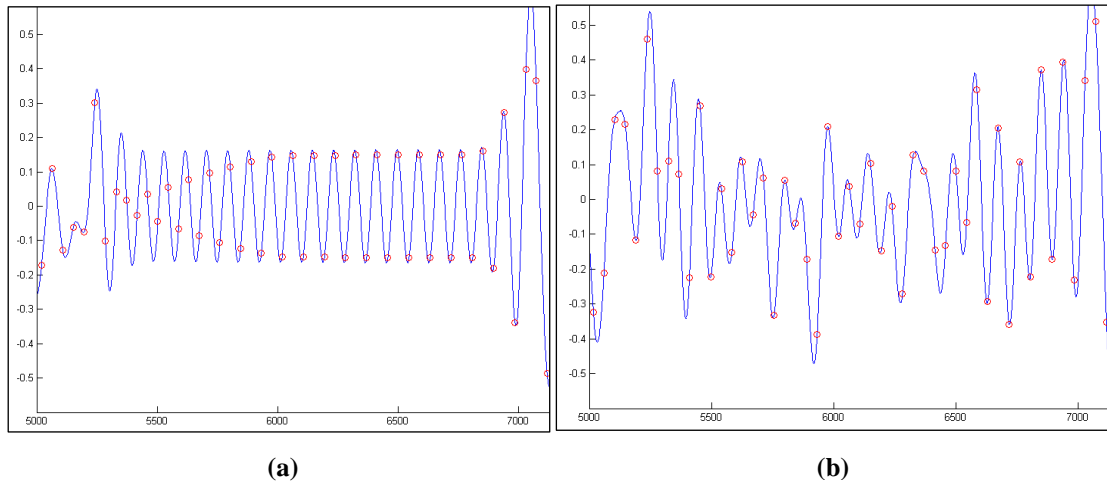


Figura 3.5 Puntos de muestreo (en rojo en a gráfica) en la secuencia de entrenamiento para algoritmo *early/late gate*. a) Señal sin ruido. b) Señal con ruido AWGN con $SNR = -5$ dB.

Esta secuencia de entrenamiento, se muestra en la 3.5b dañada por ruido blanco gaussiano aditivo (AWGN) con $SNR = -5$ dB. Es evidente que el ruido afecta la señal de manera considerable y, por tanto, se dificulta aún más lograr el sincronismo. Sin embargo, se nota que los puntos de muestreo de los últimos símbolos de la secuencia tienen sincronismo casi perfecto con los puntos de máxima energía del símbolo.

Las pruebas experimentales han demostrado que la elección del algoritmo *early/late gate* para el sincronismo de símbolo y la implementación del mismo utilizada son decisiones acertadas para este sistema.

3.4 Transmisión real de datos

Para comprobar el funcionamiento del sistema en el ambiente para el que está destinado, laboratorios de computación, se conectan dos computadoras personales de propósito general. Para la utilización del sistema, la transmisión entre computadoras se puede realizar de tres formas:

1. Directa entre interfaces de sonido.
2. Bocina-Micrófono.
3. Utilizando transmisión de FM.

La imagen transmitida y la señal recibida en un canal ideal no ruidoso se muestran en la figura 3.6 incluyendo los puntos de muestreo del detector en el receptor. Para simular un canal ideal se ejecuta la función del transmisor en una computadora y se guarda la señal de salida como una variable que luego es cargada ejecutando el receptor en la misma computadora. De este modo, la señal de entrada para la función del receptor es exactamente la misma que la entregada por el transmisor.

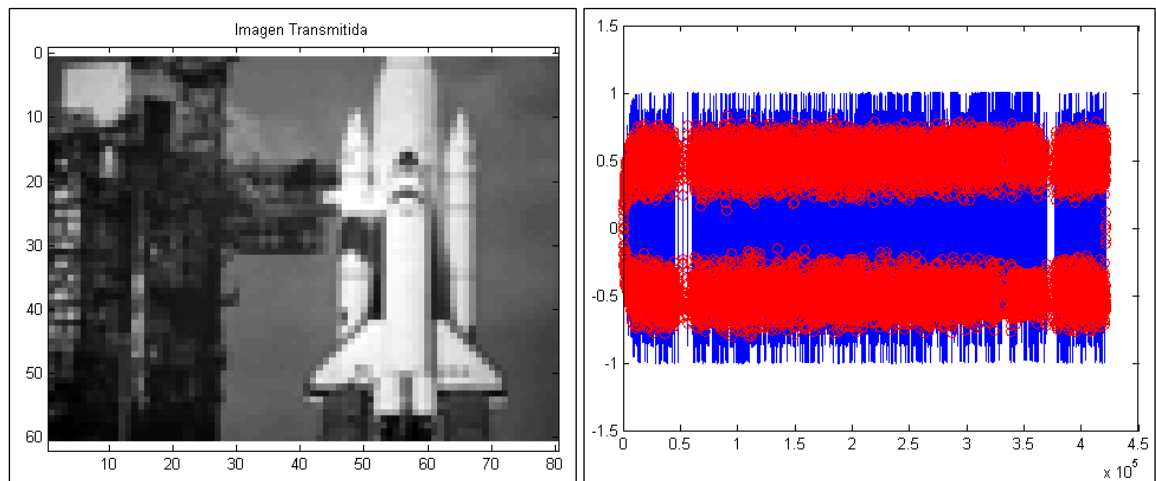


Figura 3.6 Imagen transmitida y señal recibida en canal ideal con puntos de muestreo en el detector (círculos rojos).

3.4.1 Conexión directa entre interfaces de sonido.

La primera de las configuraciones se realiza utilizando un cable puente con un conector de audio 3.5 mm macho en cada extremo. Se conecta un extremo del cable en la salida de audio de la computadora en función de transmisor. El otro extremo en la entrada de línea (*line-in*) de la computadora que realiza la función de transmisor.



Figura 3.6 Puente de conectores de audio 3.5 mm.

Las implicaciones de emplear la configuración directa entre interfaces de sonido son positivas para el sistema puesto que es menos afectado por el ruido, no hay que

configurar el volumen de las interfaces de audio y ofrece una sencillez de interconexión que la hace cómoda para su empleo. Sin embargo, este tipo de interconexión tiene las desventajas de que resulta difícil verificar la respuesta del sistema ante el ruido en entorno real pues casi no es afectado por este factor, se necesita un cable del tipo descrito, se necesita cierta cercanía entre las computadoras pues la longitud de un cable de este tipo no suele sobrepasar los 1.5 m, y solo se puede hacer transmisión punto-a-punto.

Cuando se utiliza esta configuración en pruebas experimentales, tal y como se describió antes, no ocurren errores de transmisión. Aunque la señal se deforma ligeramente, la figura recibida es la misma que la transmitida.

3.4.2 Audiofrecuencias

La configuración bocina-micrófono funciona básicamente del siguiente modo. Se ejecuta el transmisor en una PC que tenga algún medio de reproducción de audio, digamos bocinas. Se graba la señal de audio de la información creada por el transmisor. Luego esta señal es reproducida y grabada por micrófono para una computadora ejecutando la función del receptor. De este modo se puede transportar la información sin importar la distancia, solo necesita un dispositivo de grabación para transportarla en forma de señal de audio.

En el mundo real, esto es una opción poco factible. Sin embargo, al utilizarla como prueba experimental permite examinar los efectos del ruido de la banda audible por el ser humano sobre la información transmitida en dicha banda. Es importante destacar que la calidad del medio de reproducción es importante para esta configuración puesto que los efectos de distorsión y ruido de la bocina, por poner un ejemplo, van a afectar directamente la señal grabada.

En las pruebas experimentales realizadas se comprobó que la afectación del ruido era considerable –véase figura 3.7–, sobre todo si se compara con los resultados del modelo de conexión directa, por lo que ocurrían errores a una razón de $1,2 \times 10^{-3}$. En todas las ocasiones en que se realizó el experimento, los errores ocurridos fueron corregidos por el código convolucional utilizado.

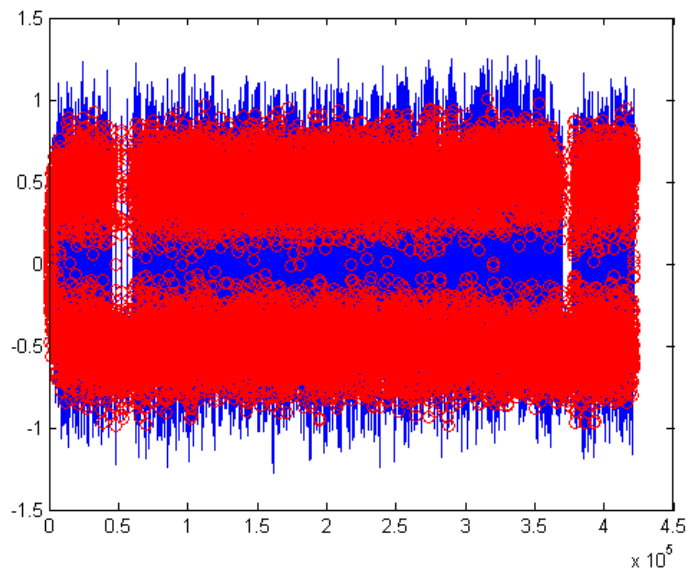


Figura 3.7 Señal recibida por la configuración bocina/micrófono

3.4.3 Transmisión de FM.

El empleo de transmisión de FM para el funcionamiento del sistema implica un esquema de conexión un poco más complicado. Se utiliza un modulador que pueda conectarse a la salida de sonido de la computadora transmisora. En el otro extremo se conecta un receptor de FM comercial con salida de audio por interfaz 3.5 mm utilizando el cable mencionado en la configuración directa, para conectar el receptor de FM con la entrada de línea de la computadora en función de recepción. La figura 3.8 muestra un diagrama ilustrativo del uso de modulación FM en el sistema.

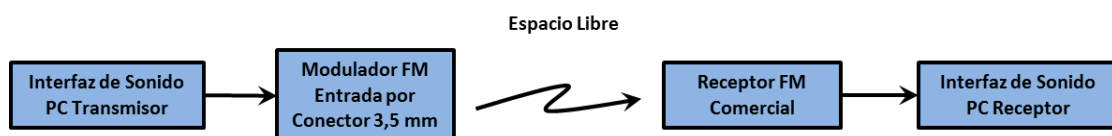


Figura 3.8 Diagrama en bloques de conexión usando transmisión FM

Esta configuración tiene la ventaja para el estudio académico de tener menor inmunidad al ruido y, por tanto, se puede verificar el desempeño del sistema en ambiente más hostil que el de conexión directa. Además, elimina la restricción de cercanía en dependencia del alcance del transmisor de FM y permite la transmisión punto-a-zona. Sin embargo presenta la desventaja de ser más caro construir el transmisor de FM, se sigue utilizando el mismo cable puente por tanto los costos solo aumentan y, evidentemente, la presencia de ruido nunca es deseable para un funcionamiento real de un sistema de comunicaciones.

En el desarrollo de los experimentos utilizando esta configuración la distorsión de la señal por causa del ruido es grande –véase figura 3.9–. Esto provocó la ocurrencia de errores a una razón de 9.6×10^{-3} . Ya esto es una razón de error considerable pero el código corrector de errores implementado la reduce hasta 1.8×10^{-3} .

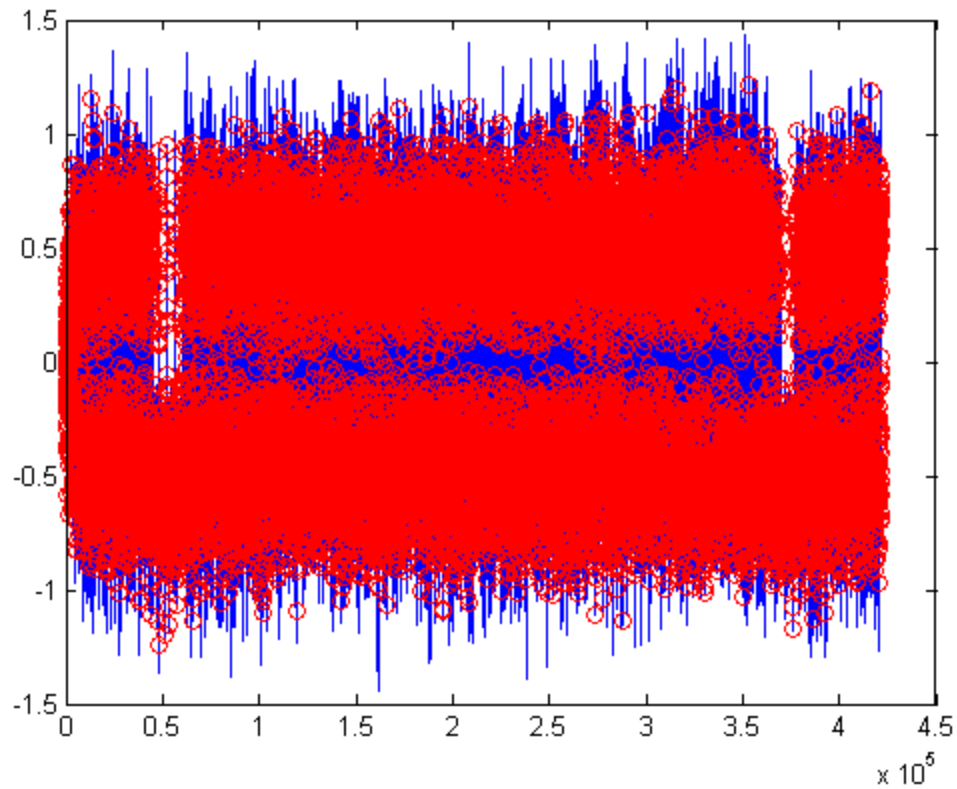


Figura 3.9 Señal recibida usando transmisión de FM.

CONCLUSIONES

Como resultado del desarrollo del trabajo de diploma se ha logrado hacer un estudio teórico de la tecnología SDR y sus aplicaciones en ámbitos académicos, profesionales y comerciales. Además, se implementaron funciones de transmisores y receptores basados en SDR usando MATLAB para su empleo con fines docentes en la facultad de Ingeniería Eléctrica.

Las funciones de MATLAB componen un sistema de SDR transmisor-receptor en el que se emplean algunas de las técnicas de modulación más utilizadas en los sistemas de transmisión digital.

Las funciones programadas muestran conceptos fundamentales en el aprendizaje de Fundamentos de las Comunicaciones y fueron sometidas a pruebas experimentales para demostrar su funcionamiento. Además de las funciones, se probaron los algoritmos más importantes empleados en el funcionamiento del sistema, tales como sincronismo de portadora por lazo de Costas, sincronismo de símbolo con *early/late gate* y DSSS. Para el funcionamiento del sistema se utiliza la tarjeta de sonido como interfaz de entrada/salida y se incorpora transmisión con modulación de FM en pruebas experimentales.

Este trabajo brinda herramientas para el desarrollo de prácticas de laboratorio de Fundamentos de las Comunicaciones por parte de los estudiantes de Ingeniería en Telecomunicaciones y Electrónica de la Facultad de Ingeniería Eléctrica. En la creación del sistema se respeta el empleo de los recursos disponibles en la facultad y soluciona la deficiencia de recursos para realizar este tipo de laboratorios.

El informe del trabajo de diploma queda disponible para el estudio teórico de SDR y como material de consulta para la programación en MATLAB de sistemas de radio usando dicha tecnología.

Los autores recomiendan la ampliación de este trabajo, tanto en su parte teórica como en su componente práctica. La ampliación del sistema SDR implementado debe incorporar funciones en los diferentes bloques utilizando, por ejemplo, otras técnicas de modulación, codificación y espectro extendido. Se propone además, enriquecer el sistema añadiéndole otros bloques y funcionalidades como encriptación, codificación de fuente e intercalación de bits.

REFERENCIAS BIBLIOGRÁFICAS.

- [1] B. Sklar, *Digital Communications. Fundamentals and Applications.*, First ed., Upper Saddle River, NJ: Prentice Hall, 1988.
- [2] H. Zeng, J. W. Lockwood, G. A. Covington *et al.*, "AirFPGA: A Software Defined Radio Platform Based on NetFPGA ", <http://www.stanford.edu/~hyzeng/paper/airfpga.pdf>, [Mayo 2011, 2009].
- [3] E. Blossom. "GNU Radio: the open-source software defined Radio," <http://dev.emcelettronica.com/gnu-radio-open-source-software-defined-radio> <http://gnuradio.org/redmine/wiki/gnuradio>.
- [4] AMSAT-UK, "Sitio Web Oficial de FUNcube Dongle," 2011.
- [5] Kanga. "Sitio Web Oficial de Kanga Products," <http://www.kanga-products.co.uk/index.php/products/finningley-80m-sdr>.
- [6] M. Pernter. "Sitio Web Oficial de PM-SDR," <http://www.iw3aut.altervista.org/>.
- [7] AUTOSAR. "Sitio Web Oficial de AUTOSAR," Abril, 2011; <http://www.autosar.org>.
- [8] Ford. "Sitio Web Oficial de Ford Sync," Abril, 2011; <http://www.ford.com/technology/sync/>.
- [9] G. Motors. "Sitio Web Oficial de OnStar," Marzo, 2011; <http://www.onstar.com>.
- [10] S. R. L. LLC. "SRL QuickSilver QS1R Receiver," <http://www.srl-llc.com>.
- [11] WiNRADiO. "WiNRADiO... the future of radio. WR-G31DDC 'EXCALIBUR'," <http://www.winradio.com/home/g31ddc.htm>.
- [12] J. Mitola, "The Software Radio," in IEEE National Telesystems Conference, 1992.
- [13] W. I. Forum. "Software Defined Radio," Abril, 2011; <http://data.memberclicks.com/site/sdf/SoftwareDefinedRadio.pdf>.
- [14] L. Pucker, "SDR Architecture," http://data.memberclicks.com/site/sdf/tut-SDR_Architectures.pdf [Marzo 2011, 2010].
- [15] W. A. Sethares, and C. R. Johnson, Jr., *Telecommunication Breakdown. Concepts of Communication Transmitted via Software-Defined Radio.*, First ed., Upper Saddle River, NJ: Prentice Hall, 2003.
- [16] R. Rubenstein, "SOFTWARE-DEFINED RADIO: Softly does it," Technology Trends, http://data.memberclicks.com/site/sdf/tut-SDR_article.pdf, [Marzo 2011, 2010].
- [17] J. E. Gunn, "SDR Market Studies Overview," http://data.memberclicks.com/site/sdf/tut-Market_Studies_Overview.pdf, [Abril 2011, 2007].
- [18] IMEC. "Wireless Communication. Cognitive Radio," Mayo 10, 2011; http://www2.imec.be/be_en/research/green-radios/cognitive-radio.html.
- [19] A. Dejonghe, "Digital Baseband Solutions For Cognitive Radio," http://www2.imec.be/content/user/File/Brochures/GR2011_Leaflet_COBRA.pdf, 2011].
- [20] J. V. Driessche, "Scaldio Reconfigurable Radio Transceiver Ip," http://www2.imec.be/content/user/File/Brochures/GR2010_Leaflet%20Scaldio.pdf, 2011].
- [21] W. I. Forum. "What are Cognitive Radio and Dynamic Spectrum Access?," Abril, 2011; <http://data.memberclicks.com/site/sdf/Introduction%20to%20CR%20and%20DSA.pdf>.
- [22] J. L. Massey, and M. K. Sain, "Inverse of Linear Sequential Circuits," *IEEE Trans. Comput.*, vol. C17, pp. 330-337, 1968.
- [23] D. Bourse, "End-to-End Reconfigurability. Concertation Meeting," ftp://ftp.cordis.europa.eu/pub/ist/docs/directorate_d/cnt/neweve/e2r_en.pdf, [Febrero 2011, 2005].

- [24] A. Shukla, A. Alptekin, J. Bradford *et al.*, "Cognitive Radio Technology – A Study for Ofcom," http://stakeholders.ofcom.org.uk/binaries/research/technology-research/cograd_main.pdf, [Febrero 2011, 2007].
- [25] FlexRay.com. "Sitio Web Oficial de FlexRay," Abril, 2011; <http://www.flexray.com/>.
- [26] J. P. Campbell, W. M. Campbell, D. A. Jones *et al.*, "Biometrically Enhanced Software-Defined Radios," http://www.ll.mit.edu/mission/communications/ist/publications/031118_CampbellJ.pdf, [Marzo 2011, 2010].
- [27] W. I. Forum. "Smart Radio Challenge," Marzo, 2011; <http://www.radiochallenge.org>.
- [28] T. Irving, "3SA (Spread Spectrum Simulation Application) ", National Science Foundation Research Experience for Undergraduates in Software-Defined Radio. Stevens Institute of Technology, 2005.
- [29] MathWorks, "MATLAB Help," 2008].
- [30] J. G. Proakis, and M. Salehi, "Contemporary Communication Systems Using Matlab," Bookware Companion Series T. P. Inc., ed., PWS Publishing Company, 1998.
- [31] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed., Upper Saddle River, NJ: Prentice Hall, 1999.
- [32] H. L. Krauss, C. W. Bostian, and F. H. Raab, *Estado Sólido en Ingeniería de Radiocomunicaciones*, New York: Wiley, 1980.
- [33] J. G. Proakis, *Digital Communications*, 4ta ed., New York: McGraw Hill, 2001.
- [34] L. Goleniewski, "Telecommunications Essentials, Second Edition: The Complete Global Source," K. W. Jarrett, ed., Addison Wesley, 2007.
- [35] J. Feigin, "Practical Costas Loop Design.," <http://rfdesign.com/images/archive/0102Feigin20.pdf>, [Enero 2011, 2002].
- [36] P.N.Ravichandran, S. Sharma, S. Kulkarni *et al.*, "Design and Implementation of Early-Late Gate Bit Synchronizer for Satellite Communication," <http://www.ncc.org.in/download.php?f=NCC2009/file4.pdf>, [Mayo 2011, 2009].
- [37] S. Pasupathy, "Minimum Shift Keying: A Spectrally Efficient Modulation," *IEEE Communications Magazine*, 1979.
- [38] J. K. Omura, "On the Viterbi Decoding Algorithm," *IEEE Trans. Inf. Theory*, vol. IT15, 1969.
- [39] S. Shah, and V. Sinha, "Iterative Decoding vs. Viterbi Decoding: A Comparison," <http://2422055075149222849-a-1802744773732722657-sites.googlegroups.com/site/santoshjnt/santoshshah.pdf>, [Abril 2011, 2002].

GLOSARIO

AM: Amplitude Modulation.

ARQ: Automatic Repeat Request.

ASK: Amplitude Shift Keying.

AUTOSAR: Automotive Open System Architecture.

AWDN: Additive White Gaussian Noise.

ISR: Ideal Software Radio.

BWA: Broadband Wireless Access.

BPSK: Binary PSK.

Code Division Multiple Access.

CR: Cognitive Radio.

DARPA XG: DARPA Next Generation.

DAQ: Data Acquisition Card.

DSP: Digital Signal Processor.

DS(SS): Direct Sequence (Spread Spectrum).

E²R: End-to-End Reconfigurability

ECU: Electronic Control Unit.

EDGE: Enhanced Data for GSM Evolution.

FEC: Forward Error Correction.

FFT: Fast Fourier Transform.

FH(SS): Frequency Hopping (Spread Spectrum).

FPGA: Field Programmable Gate Array.

FSK: Frequency Shift Keying.

GSM: Global System for Mobile Communications.

HSPA: High Speed Packet Access.

IF: Intermediate Frequency.

ISI: Intersymbol Interference.

ISR: Ideal Software Radio.

LNA: Low Noise Amplifier.

LPD: Low Probability of Detection.

LPPF: Low Probability of Position Fix.

LPSE: Low Probability of Signal Exploitation.

LSI: Large Scale of Integration.

LTE: Long Term Evolution.

MANET: Mobile Ad-hoc Network.

MFSK: M-ary Frequency Shift Keying.

MIT: Massachusetts Institute of Technology.

OEM: Original Equipment Manufacturer.

OFDM: Orthogonal Frequency Division Multiplex.

PAM: Pulse Amplitude Modulation.

PCM: Pulse Code Modulation.

PLL: Phase Locked Loop.

PSK: Phase Shift Keying.

SDR: Software Defined Radio.

SCR: Software Controlled Radio.

SNR: Signal-to-Noise Ratio.

SS: Spread Spectrum.

TH: Time Hoping.

USR: Ultimate Software Radio.

USRP: Universal Software Radio Peripheral.

VoIP: Voice over Internet Protocol.

VioIP: Video over Internet Protocol.

ANEXO

Código Transmisor

```
function y = psk_tx_im_cc_soft(Rs, Fc)
% Rs razón de datos
% Fc frecuencia de portadora

Fs = 44100;           % Frec. muestreo para tarjeta de sonido
ini_phase = 0;        % Fase inicial
r = 0;                % Factor de roll-off del filtro
close all;

% Lectura de la imagen a ser transmitida
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
im_in=imread('shuttle_80x60.tif');
L=8;
[size_r,size_c]=size(im_in);
im_size=size_r*size_c;
im_vec=reshape(im_in,1,im_size);
bit_matrix=de2bi(im_vec);
bit_symbols=reshape(bit_matrix, im_size*L, 1); % Bits a ser transmitidos
x=bit_symbols';
x = [de2bi(im_size,128,2) de2bi(size_r,128,2) de2bi(size_c,128,2)...
     de2bi(L,8,2) x];
x = [de2bi(length(x),40,2) x];

% Convolutional Coding
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
trell = poly2trellis([5 4],[23 35 0; 0 5 13]);
x = convenc(x, trell);

% Preámbulo y Delimitador de Trama
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RandStream.setDefaultStream(RandStream('mt19937ar', 'seed', 10));
fd = randint(1, 2000);
x = [fd x];

% Espectro Extendido por Secuencia Directa (DSSS)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
chip = 5;           % Número de chips por bit de información
rep_x = rectpulse(x, chip);
RandStream.setDefaultStream(RandStream('mt19937ar', 'seed', 117));
pn_seq = randint(1, length(rep_x)); % Inicialización del vector PN
x = xor(rep_x, pn_seq);

% Modulación PAM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ss = upsample(ones(1,20),2); % Secuencia de entrenamiento
x= [ss x];
```

% Modulación PAM

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s = [-1,1];
y = s(x+1); % Modulación Banda Base

```

% Visualización de la imagen transmitida

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
colormap(gray);
h=image(im_in);
set(h,'CDataMapping','scaled')
axis('equal');
title('Imagen Transmitida');
hold;

```

% Formateo del Pulso

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Rs = Fs/floor(Fs/Rs); % Ajuste de Rs para Fs/Rs entero
num = rcosine(Rs,Fs,'sqrt',r); % Diseño del filtro
y = rcosflt(2*y,Rs,Fs,'filter',num); % Filtro raíz de coseno alzado

```

% Modulación Pasabanda

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t = (0:length(y) - 1)/Fs; % Puntos de muestreo
y = y.*cos(2*pi*Fc*t + ini_phase); % Modulación

```

% Salida por tarjeta de sonido escalada en el rango [-1 1]

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
soundsc(real(y),Fs);
end

```

Código Receptor

```

function x = psk_rx1_im_cc_soft(y, Rs, Fc)

```

```

% y datos
% Rs razón de datos
% Fc frecuencia de portadora

```

```

Fs = 44100; % Frec. muestreo de tarjeta de sonido
r = 0; % Factor de roll-off del filtro

```

% Lazo de Costas - Sincronismo de Portadora

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t = (0:length(y) - 1)/Fs; % Puntos de muestreo
fle=64; % Orden del filtro
h=fir1(fle,0.001); % Diseño del filtro Paso Bajo
mu=0.01; % Tamaño del paso del algoritmo
theta=zeros(1,length(t)); % Inicialización de vector de estimación
theta(1) = -1; % Fase inicial
zs=zeros(1,fle+1);zc=zeros(1,fle+1);% Inicialización de buffers de LPF

```



```

for k=1:length(t)-1
    zs=[zs(2:fle+1), 2*y(k)*sin(2*pi*Fc*t(k)+theta(k))];
    zc=[zc(2:fle+1), 2*y(k)*cos(2*pi*Fc*t(k)+theta(k))];
    lpfs=flipplr(h)*zs';           % Nueva salida de los filtros
    lpfc=flipplr(h)*zc';           % Nueva salida de los filtros
    theta(k+1)=theta(k)-mu*lpfs*lpfc; % Actualización del algoritmo
end

% Sincronismo de señal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s_inic = upsample(ones(1,20),2); % Secuencia de entrenamiento
s = [-1,1];
s_inic = s(s_inic+1);
Rs = Fs/floor(Fs/Rs);           % Ajuste de Rs para Fs/Rs entero
num = rcosine(Rs,Fs,'sqrt',r);   % Diseño del filtro
s_inic = rcosflt(s_inic,Rs,Fs,'filter',num)'; % Filtrado
t_aux = (0:length(s_inic) - 1)/Fs; % Puntos de muestreo
s_inic = s_inic.*cos(2*pi*Fc*t_aux); % Modulación
[a,i] = max(xcorr(s_inic,double(y)));
i = length(y) - i;
y = y(i+1:end);

% Demodulación pasabanda (a banda base)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t = (0:length(y) - 1)/Fs;
y = y.*cos(2*pi*Fc*t + theta(1:length(y))); % Demodulación

% Filtro Paso Bajo de Entrada (Filtro Acoplado)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Rs = Fs/floor(Fs/Rs);           % Ajuste de Rs para Fs/Rs entero
n = rcosine(Rs,Fs,'sqrt',r);   % Diseño del filtro
y = rcosflt(y,Rs,Fs,'fir/filter/Fs',n)'; % Filtrado

% Symbol synchronization & Detection
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
st = ceil((Fs/Rs)); t = st;      % Punto de muestreo
d = 1;                           % Tamaño del paso del algoritmo
err = 0.01;                       % Error permitido
si = ones(1,ceil(length(y)/(st-1))); % Índices de muestreo
for i = 1:length(si)
    dif=abs(y(t-d))-abs(y(t+d));
    if dif > err
        t = t - 1;
    elseif dif < -err
        t = t + 1;
    end
    si(i) = t;
    t = t + st;
    if t > length(y) - st + 1
        break
    end
end

```

```

    end
end

si = si(~(si > length(y)));
x = y(si)>0;

% Eliminación de secuencia de entrenamiento
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ss = upsample(ones(1,20),2);    % Secuencia de entrenamiento
s = [-1,1];
ss = s(ss+1);
[a,i] = max(xcorr(ss,double(x)));
i = length(x) - i + length(ss);
x = x(i+1:end);

% Espectro Extendido por Secuencia Directa (DSSS)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RandStream.setDefaultStream(RandStream('mt19937ar', 'seed', 117));
pn_seq = randint(1, length(x)); % Inicialización del vector PN
x = xor(x, pn_seq);
chip = 5;
no_bits = floor(length(x)/chip);
x = x(1:((floor(length(x)/5))*5));
x_temp = zeros(1, 6);
for i=1:no_bits
    x_temp(i) = intdump(double(x(1:chip)), chip);
    x = x((chip+1):end);
end;
x = x_temp;
x = x > 0;

% Sincronismo de trama
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RandStream.setDefaultStream(RandStream('mt19937ar', 'seed', 10));
fd = randint(1, 1500);
[a,i] = max(xcorr(fd, double(x)));
i = length(x) - i + length(fd);
figure; plot(xcorr(fd, double(x)))
x = x(i+1:end);
x = x(1:3*floor(length(x)/3));

% Corrección de errores usando algoritmo de Viterbi
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
trel = poly2trellis([5 4],[23 35 0; 0 5 13]);
x = vitdec(x, trel, 25, 'trunc', 'hard');
l = bi2de(x(1:40), 2);
x = x(41:40+l);

% Reconstrucción y visualización de la Imagen Recibida
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
im_size = bi2de(x(1:128), 2);
x = x(129:end);
size_r = bi2de(x(1:128), 2);
x = x(129:end);
size_c = bi2de(x(1:128), 2);
x = x(129:end);
L = bi2de(x(1:8), 2);
x = x(9:end);
bit_matrix_rx=reshape(x, im_size, L);
im_vec_rx=bi2de(bit_matrix_rx);
im_in_rx=reshape(im_vec_rx,size_r,size_c);

% Visualización de la imagen recibida
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
colormap(gray);
h=image(im_in_rx);
set(h,'CDataMapping','scaled')
axis('equal');
title('Imagen Recibida');
end
```