ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

SIMULACIÓN DE DSSS Y FHSS EN MATLAB

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR EN REDES Y TELECOMUNICACIONES

BETTY MILENA ESPINOZA ESPINOZA

DIRECTOR: ANDRÉS FERNANDO REYES CASTRO

DMQ, marzo 2023

CERTIFICACIONES

Yo, BETTY MILENA ESPINOZA declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

BETTY MILENA ESPINOZA ESPINOZA

betty.espinoza@epn.edu.ec

betty.mileespinoza24@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por BETTY MILENA ESPINOZA, bajo mi supervisión.

ANDRES FERNANDO REYES CASTRO

andres.reyes@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

BETTY MILENA ESPINOZA ESPINOZA

betty.espinoza@epn.edu.ec

betty.mileespinoza24@gmail.com

DEDICATORIA

Este proyecto está dedicado a mi familia, principalmente a las mujeres de mi vida, a mi madre Margarita y a mi querida abuela Emma, a mi padre Armando y a mi querido hermano Brandon que fueron ejemplos de superación, me alentaron a seguir adelante y estuvieron junto a mí durante todo mi trayecto.

Betty Milena.

AGRADECIMIENTO

A mis profesores y a la Escuela Politécnica Nacional, por permitirme conocer y aprender más, a mi director de tesis el Ing. Andrés Reyes quien me guio durante el transcurso del proyecto, a mis queridos amigos que con sus conocimientos y apoyo me alentaron a seguir, a mis padres por darme la valentía y el deseo constante de superación, les agradezco siempre.

Betty Milena.

ÍNDICE DE CONTENIDOS

CERTIFICACIONES I

DECLARACIÓN DE AUTORÍA II

DEDICATORIA III

AGRADECIMIENTO IV

ÍNDICE DE CONTENIDOS V

RESUMEN VII

ABSTRACT VIII

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO 1

1.1 Objetivo general 2

1.2 Objetivos específicos 2

1.3 Alcance 2

1.4 Marco Teórico 2

Espectro Ensanchado 2

Características de las técnicas de SS 3

Espectro Ensanchado por Secuencia Directa 4

Espectro Ensanchado Salto de Frecuencias 4

Diagrama de Bloques 5

Códigos pseudoaleatorios 7

Modulación y demodulación BPSK 7

Espectro de frecuencia 8

Eficacia de utilización del espectro 8

Densidad Espectral de Potencia 8

Matlab 9

2 METODOLOGÍA 10

3 RESULTADOS 11

3.1 Principios de funcionamiento de DSSS y FHSS 11

Principio de DSSS 11

Principio de FHSS 11
3.2 Determinación de las herramientas de Matlab 12
Ventana de Comandos 12
Archivos*.m 12
Signal Analyzer 14
3.3 Implementación de los escenarios para la simulación 16
Implementación de la simulación para DSSS 16
Implementación de la simulación para FHSS 22
3.4 Comparaciones entre las de las simulaciones y sus resultados entre DSSS y FHSS 27
Resultados de DSSS 27
Resultados de FHSS 30
Comparación respecto al Espectro de Frecuencia 32
Comparación respecto a la Densidad Espectral de Potencia 32
Comparación en el dominio Tiempo-Frecuencia 33
4 CONCLUSIONES 34
5 RECOMENDACIONES 35
6 REFERENCIAS BIBLIOGRÁFICAS 36
7 ANEXOS 39
ANEXO I: Certificado de Originalidad i
ANEXO II: Enlaces ii
ANEXO III: Diagrama de flujo del código fuente de la simulación de DSSS en Matlab iii
ANEXO IV: Diagrama de flujo del código fuente de la simulación de FHSS en Matlab v
ANEXO V: Código fuente de la simulación de DSSS en Matlab vii
ANEXO VI: Código fuente de la simulación de FHSS en Matlab xv
ANEXO VII: Comandos empleados para la simulación de DSSS y FHSS en Matlab . xxii

RESUMEN

En el presente documento se evidencia el desarrollo del Trabajo de Integración Curricular que consiste en la simulación mediante la plataforma Matlab de los escenarios del método de Espectro Ensanchado (SS): Espectro Ensanchado por Secuencia Directa (DSSS) y Espectro Ensanchado por Salto de Frecuencia (FHSS), y establecer comparaciones entre ambas técnicas.

Se presenta en la primera sección una introducción sobre el proyecto y se continua con la descripción de los objetivos y el alcance que tendrá el mismo.

En la metodología se definen conceptos que complementan la implementación de las simulaciones y se inicia con una investigación de los fundamentos de SS y sus técnicas más conocidas DSSS y FHSS.

Para el desarrollo del proyecto se parte con la selección de las herramientas de Matlab y se explica de forma detallada la creación de dos entornos para implementar la simulación de cada técnica respectivamente. Como resultado se obtienen los componentes para realizar las comparaciones posteriores; para las cuales se analiza el dominio del tiempo y la frecuencia y se procede a concluir y recomendar sobre el proyecto.

PALABRAS CLAVE: Espectro Ensanchado, DSSS, FHSS, Matlab, Simulación.

ABSTRACT

In this document, the development of the final degree project is evidenced, which consists of simulating, by using Matlab, scenarios of the Spread Spectrum (SS) techniques: Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS) and establish comparisons between them.

The first section presents an introduction to the project, followed by a description of the objectives and scope of the project. The methodology defines concepts that complement the implementation of the simulations, starting with an investigation of the fundamentals of SS and its well-known techniques, DSSS and FHSS.

The project development begins with the selection of Matlab tools, and a detailed explanation of the creation of two environments to implement the simulation of each respective technique. As a result, components are obtained to carry out subsequent comparisons. The time and frequency domains are analyzed, and conclusions and recommendations are made regarding the project.

KEYWORDS: *Spread Spectrum, DSSS, FHSS, Matlab, Simulation.*

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Dada la importancia de las telecomunicaciones en la sociedad, es pertinente realizar el análisis de métodos de transmisión de las señales; uno de ellos el Espectro Ensanchado, *Spread Spectrum* (SS), que hace referencia a transmitir la señal variando la frecuencia a la que es enviada con la finalidad de brindar seguridad frente a interferencias.

La técnica de SS, inicialmente por sus altos costos, fue implementada en aplicaciones militares hasta comienzo de los años noventa debido a sus características de encriptar la información y brindar seguridad a los sistemas de comunicación. En la actualidad este método es usado por los fabricantes en su afán de aumentar la capacidad de los canales, que se hallan más congestionados debido a la creciente demanda de usuarios. SS permite que simultáneamente se use la banda ancha de ciertas frecuencias siendo útil para sistemas de redes inalámbricas para la transmisión de datos digitales y por radiofrecuencia. SS presenta ventajas respecto al ancho de banda, este es más amplio favoreciendo a la inmunidad contra los efectos de canal.

En el presente proyecto se realiza una simulación para el análisis de las técnicas *Direct Sequence Spread Spectrum* (DSSS) o técnica de Espectro Ensanchado por Secuencia Directa y *Frecuency Hopping Spread Spectrum* (FHSS) o técnica de Espectro Ensanchado por Salto de Frecuencia.

Para la simulación se aplicó la modulación por Desplazamiento de Fase Binario o *Binary Phase Shift Keying* (BPSK), cuya finalidad es transmitir un mensaje cambiando la fase de la señal portadora que representa cada símbolo de información de la señal moduladora. En BPSK se tendrán dos fases con una sola frecuencia, la potencia de todos los símbolos siempre es la misma y presenta una mayor inmunidad al ruido y a las interferencias.

El desarrollo de las simulaciones DSSS y FHSS se realizó mediante el uso de la plataforma Matlab, que ofrece un lenguaje de programación con un entorno grafico fácil de usar y permite el procesamiento de señales; el proyecto se llevó a cabo en la ventana de edición de *scripts* donde se expresan los comandos y operaciones; se puede escribir o modificar el código de las simulaciones, mediante las aplicaciones de la plataforma se obtuvo los gráficos y hacer un análisis del funcionamiento del *scrip*t, útil en este proyecto para establecer comparaciones entre ambas técnicas de SS.

1.1 Objetivo general

Simular DSSS y FHSS en Matlab.

1.2 Objetivos específicos

- Investigar principios de DSSS y FHSS
- Determinar las herramientas de Matlab
- Implementar escenarios para la simulación
- Fijar las comparaciones a partir de las simulaciones y sus resultados entre DSSS y FHSS

1.3 Alcance

A partir del proyecto y la previa investigación de los principios de DSSS y FHSS se obtiene la idea del funcionamiento de ambas técnicas partiendo del apoyo de las herramientas de Matlab, plasmando escenarios de simulación y concluyendo con los comparativos basados en dichas simulaciones y sus resultados.

1.4 Marco Teórico

Espectro Ensanchado

Es un método de transmisión en el que la señal a transmitir ocupa un ancho de banda más grande que el mínimo requerido para el envío de información [1].

El sistema de SS está diseñado considerando el ancho de banda y la potencia; en este la energía de la señal transmitida se distribuye en un ancho de banda mayor que el de la información a modularse, así pues, se intercambian un mayor ancho de banda con una densidad espectral de baja potencia con mayor inmunidad a interferencias de la misma banda de frecuencias [2].

En la Figura 1.1 se muestra el fundamento básico de los sistemas SS, que es el ensanchamiento de la señal a transmitirse en una gran banda ancha de frecuencias, incluso mayor al ancho de banda mínimo que se necesita para la transmisión de la información a enviar.

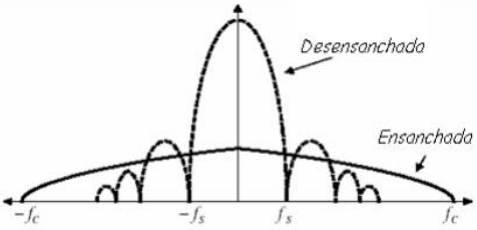


Figura 1.1 Densidad espectral de la potencia de la señal a transmitir [3].

El ensanchamiento de la señal se da antes de que la señal de información se transmita mediante un código pseudoaleatorio que es independiente de la secuencia de datos, este mismo código es usado en el receptor para obtener la señal original de datos.

Los sistemas de SS tienen la capacidad de una resistencia elevada al desvanecimiento profundo (variación de una señal con varios parámetros entre ellos el tiempo y la frecuencia de radio), beneficiando al diseño del sistema y usando de manera eficaz el espectro. La finalidad de estos sistemas es que usuarios simultáneos se hallen en la misma frecuencia asumiendo que todas las señales tienen la misma potencia, resultando una solución para el incremento de la capacidad de las comunicaciones móviles.

Características de las técnicas de SS

1. Permiten compartir el espectro con sistemas que cuenten con banda estrecha (convencional), debido a que en los sistemas SS pueden transmitir una potencia inferior por los receptores de banda estrecha. En estas técnicas la señal transmitida, es una señal con una baja densidad de potencia y con una banda ancha similar a la del ruido, así pues, un receptor convencional no la detecta con facilidad [4].
2. La información de la banda base se restituye partiendo de la señal de banda ancha transmitida únicamente por el proceso de correlación, así los usuarios de otros sistemas de transmisión no pueden percibir la información de la banda base y como cuenta con una baja densidad de potencia, no produce interferencia a sistemas que no sean de SS [2].
3. SS brinda un grado de confidencialidad a la información transmitida, respecto a sistemas que no pertenezcan al Espectro Ensanchado e incluso a otros sistemas SS que usen códigos diferentes y ningún procesamiento especial de la señal [2].
4. En los sistemas de SS utilizan señales de banda ancha semejante al ruido por ello son difíciles de detectar, también son difíciles de interceptar y demodular [5].

5. Presentan una ventaja sobre las técnicas de modulación, puesto que, brindan más fiabilidad de transmisión frente a desvanecimientos selectivos (anulación de una sola señal) y la propagación en múltiples trayectos [4].
6. Las técnicas de SS realizan un ensanchamiento de energía sobre una banda amplia es decir tiene una baja densidad de potencia espectral, característica que impide que dichos sistemas interfieran en los sistemas de comunicación con bandas más estrechas [5].

Espectro Ensanchado por Secuencia Directa

DSSS es una técnica del Espectro Ensanchado en donde la señal portadora es modulada por una señal de código pseudoaleatorio que consta de un determinado número de bits, denominados chips (indica un símbolo refiriéndose a señales de código). La tasa de chip de la señal debe ser más alta que la tasa de bit de información, para poder obtener el ensanchamiento requerido.

Para la modulación de códigos de pueden emplear las variantes de la modulación por cambio de fase (PSK) e incluso la modulación por cambio de fase mínimo.

En los sistemas de DSSS el flujo de información es dividido en pequeñas partes y se les asigna un canal de frecuencias mediante el espectro, en caso de que un bit resulte dañado en el proceso de la transmisión el código chip ayudaría debido a su característica de patrón de bits redundante se asocia a cada uno de los bits del código aumentando la resistencia de la señal a interferencias.

Todo el proceso de los sistemas DSSS se da multiplicando una señal portadora de radiofrecuencia y un código pseudoaleatorio (PN), este código es modulado mediante técnicas de modulación por desplazamiento de fase (PSK), del inglés *Phase Shift Keying.*

Espectro Ensanchado Salto de Frecuencias

FHSS, es una de las técnicas del Espectro Ensanchado (SS), donde la frecuencia de la señal portadora de la información modulada no es constante, varía en intervalos de tiempo, saltando la frecuencia de la señal portadora a otra frecuencia. El patrón de los saltos es establecido por la señal de código pseudoaleatorio (PN).

En este sistema el transmisor da saltos entre las frecuencias que se encuentren disponibles, este se haya sincronizado con el receptor y una cantidad de datos se transmiten en banda estrecha.

La frecuencia de la señal portadora aumenta y pasa a ser igual a la frecuencia de transmisión usando un sintetizador de frecuencia rápido, en cambio en el receptor se da el proceso opuesto, se usa una secuencia de código que se genera de manera local; la señal recibida en un inicio es bajada a la banda base y la información es recuperada después de la demodulación de igual manera en banda base.

El ancho de banda de la señal de alguno de los saltos dependerá del ancho de banda de la información; en un escenario donde la frecuencia de saltos sea menor que el ancho de banda de la información, el ancho de banda de la información es considerado como el factor principal para la decisión del ancho de banda a ocupar. Caso contrario si la frecuencia de saltos es mucho mayor que el ancho de banda de la información la frecuencia de salto prevalecerá para seleccionar el ancho de banda.

Otra condición es que tanto para el receptor como para el transmisor la secuencia de las frecuencias debe ser exactamente la misma, así se garantiza una señal coherente distinguiendo la señal del ruido.

Mediante el circuito de sincronización y rastreo se garantiza que el salto de la portadora creada de manera local se sincronice al patrón de saltos de la señal portadora que es recibida, así se asegura el desensanchamiento correcto.

En este tipo de sistemas existen la misma cantidad de señales portadoras como de canales cuyo ancho de banda corresponde al ancho de banda de la señal que ingrese. Durante un intervalo de tiempo la cantidad de bits posiblemente transmitidos usa un esquema de codificación analógico-digital como la Modulación por Desplazamiento de Frecuencia (FSK) o PSK.

También se usa una secuencia pseudoaleatoria que muestra la secuencia de los canales que se utilizó, el transmisor y el receptor usan la misma secuencia para mantener la sincronización. En el receptor se demodula la señal que contiene el espectro ensanchado con la misma secuencia de código pseudoaleatoria (PN), para obtener la señal de la salida.

Diagrama de Bloques

En el diagrama de bloques de la Figura 1.2 se muestra el sistema DSSS, se inicia en el transmisor con la modulación de la señal de datos binarios o de información d(t) junto con un código pseudoaleatorio PN(t); el proceso de multiplicación básicamente consiste en realizar una compuerta XOR o OR exclusiva (suma binaria cuyo resultado es verdadero o 1 si y solo sí uno de los dos operandos es verdadero), entre los bits a transmitir y los bits del código PN(t) [6], para luego modularla nuevamente con una onda

portadora llevando a cabo el proceso de expansión o ensanchamiento y como resultado se obtendrá la señal final que tendrá el mismo ancho de banda que el código PN(t). La señal a la salida contiene la señal original de información que fue modulada por BPSK.

En el receptor se inicia con un demodulador BPSK, y se demodula la señal de la salida del trasmisor con una onda portadora, luego se correlaciona la señal recibida con una copia sincronizada del código pseudoaleatorio PN(r); el proceso de sincronización debe iniciarse en el receptor y mantenerse hasta que se reciba la señal d(r).

Esta demodulación tiene el objetivo de concentrar la potencia de un usuario específico en el ancho de banda de la información, todo esto después de la transmisión de la señal.

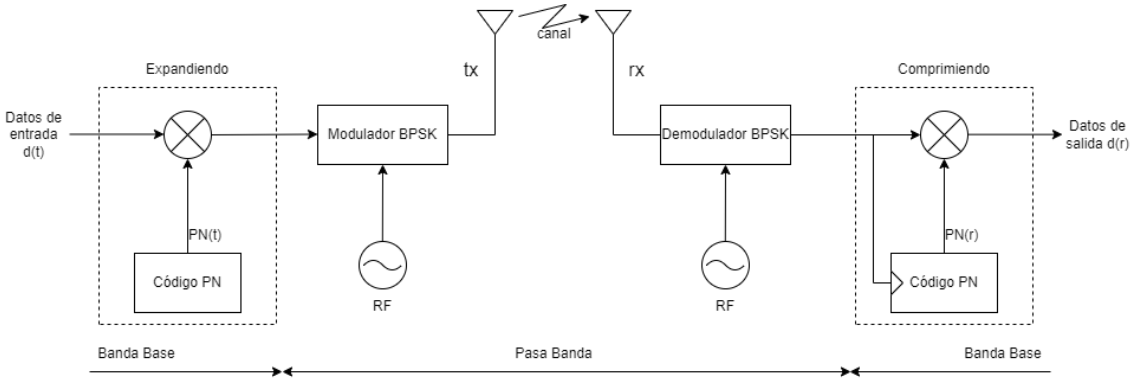


Figura 1.2 Diagrama de bloques de un sistema de DSSS [7].

En el diagrama de la Figura 1.3 se observa un sistema FHSS, en el proceso del transmisor el código pseudoaleatorio PN(t) ingresa a un sintetizador de frecuencias, con la finalidad de producir los saltos de frecuencia, permitiendo que la transmisión de cada bit o grupo de estos sea en señales portadoras diferentes [6]. Del sintetizador resulta un tono de frecuencia constante f(hi), que es la cantidad de variaciones que tiene la onda portadora, antes se aplica un proceso de conversión de la señal de Radio Frecuencia o *Radio Frequency* (RF) y el resultado será una señal de Frecuencia Intermedia o *Intermediate Frqcuency* (IF), que es f(hi).

La señal de datos binarios d(t) ingresa a un modulador BPSK, resultando una señal centrada en una de las frecuencias bases f(od), esto posibilita cambiar la frecuencia de la señal portadora junto con la señal de f(hi) y ambas ingresan al modulador de Salto de Frecuencia o *Frequency Hopping* (FH).

En el receptor la señal ensanchada es recibida por un modulador FH y junto a el código PN(r) que ingresa a un sintetizador, resultando una frecuencia f(hi), se obtiene la *frecuencia* f(od) ahora centrada en la frecuencia de otra señal portadora; y debe existir un sincronismo entre el código PN(r) y el código PN(t), para iniciar con la demodulación

de la señal reincida mediante un demodulador BPSK; en el receptor se debe realizar el proceso de saltos de frecuencias de manera sincronizada con el transmisor para lograr recuperar correctamente la señal de datos binarios d(r) [8].

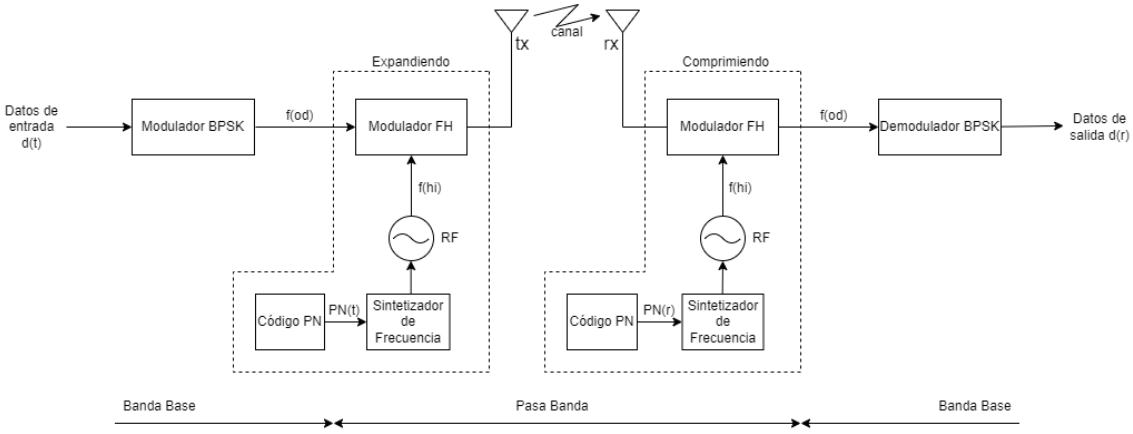


Figura 1.3 Diagrama de bloques de un sistema de FHSS [7].

Códigos pseudoaleatorios

En los sistemas digitales un código PN o secuencia de pseudoruido, del inglés *pseudo-noise* es una secuencia generada por un registro de cambio de retroalimentación es determinística y periódica, conformada por bits, con propiedades de aleatoriedad semejante al ruido y dificultad de reconstrucción [9].

En los sistemas de SS interviene un código pseudoaleatorio con la característica de que debe ser balanceado, es decir, la relación entre unos y ceros debe ser menor o igual a la unidad.

Modulación y demodulación BPSK

La modulación por Desplazamiento de Fase Binario o *Binary Phase Shift Keying* (BPSK), es una técnica de modulación digital cuya finalidad es transmitir una señal digital mediante una señal portadora analógica variando su fase [10].

Si la señal a transmitir permite únicamente un bit por símbolo, existirán dos fases a variar y mantener constante la amplitud. En la Figura 1.4 se observa las dos fases a la salida para una única frecuencia portadora, la fase de la señal portadora se representa con un 1 lógico o un 0 lógico y conforme la señal digital se modifique la fase de la señal portadora se desplaza entre dos ángulos desfasados entre si 180° [11].

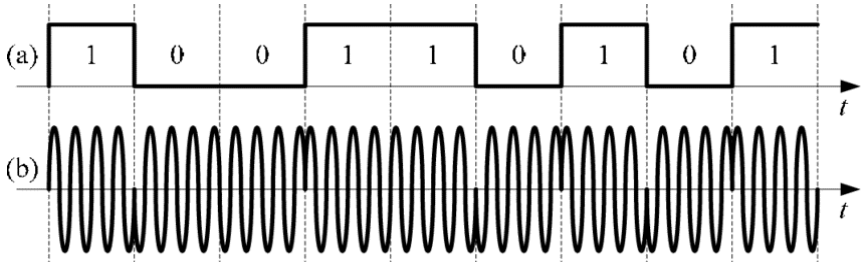


Figura 1.4 Representación de la modulación BPSK [12].

La señal a transmitir (a) es una secuencia de bits y la señal modulada mediante BPSK (b) es una señal senoidal con cambio de fase.

El proceso de demodulación consiste recuperar la información o la secuencia de bits que se transmitió mediante una señal portadora, que en el transmisor previamente fue modulada con dicha secuencia.

Espectro de frecuencia

En una representación gráfica de las ondas en función de la frecuencia, también denominado espectrograma, se muestra el nivel de intensidad de cada una de las frecuencias que emita una onda es el caso de las comunicaciones de radio y la transmisión de algunas redes móviles [13].

El espectrograma se basa en tomar una cierta cantidad de muestras en un periodo de tiempo que permite cálculos de contenido espectral con las muestras obtenidas partiendo de la Transformada Rápida de Fourier (FFT) del inglés *Fast Fourier Transform*. La suma de las representaciones de la FFT brinda información sobre el dominio de la frecuencia y la variación de la energía de la señal [14].

Eficacia de utilización del espectro

Debido a que los sistemas de SS trabajan con un ancho de banda mayor para la trasmisión de la información, se debe tomar en cuenta el grado de eficacia en la utilización del espectro, que se refiere a la capacidad para utilizar óptimamente el espectro y consiste en la cantidad de información que se puede transmitir en un determinado rango de frecuencias.

Densidad Espectral de Potencia

La Densidad Espectral de Potencia (PDS) del inglés *Power Spectral Density*, es la Transformada de Fourier de la función de autocorrelación, permitiendo determinar la distribución de la potencia de una señal en un rango de frecuencias [15]. Brinda

información sobre la potencia en el dominio de la frecuencia. Se expresa en unidades de potencia por unidad de frecuencia (dB/Hz).

Matlab

Es un sistema basado en el álgebra matricial y demás elementos matemáticos, usado para la programación y el cálculo, viene de *Matrix Laboratory* [16]. Cuenta con un lenguaje de cálculo técnico y aunque todos sus comandos se basen en el también utiliza lenguaje C, C++ o lenguaje Java, para crear un código básico con librerías y funciones predeterminadas.

Ofrece un entorno interactivo muy sencillo con una ventana para ejecutar los ficheros editables que tienen un formato *.m donde se realizan varias operaciones y procesos, además de cada ventana tiene una barra de menús con diversas funcionalidades. Cuenta con una interfaz que posibilita introducir comandos uno a uno o escribir un archivo y luego ejecutarlo, estos archivos pueden ser escritos en el editor de texto de la plataforma, los archivos de Matlab no aceptan ni devuelven argumentos de entrada y salida respectivamente, sino que operan con datos en el área de trabajo [17].

Esta plataforma entre sus múltiples ventajas ofrece una excelente capacidad grafica para la representación gráfica de los datos.

La vista por defecto como se muestra en la Figura 1.5 incluye varias ventanas de despliegue entre ellas la ventana principal o ventana de comandos *(Command Window)* a la derecha, el área de trabajo a la izquierda *(Workspace)*, donde se almacenan las variables creadas en la ventana de comandos; junto con el directorio actual *(Current Directory)* donde se almacenarán todos los archivos *.m [18].

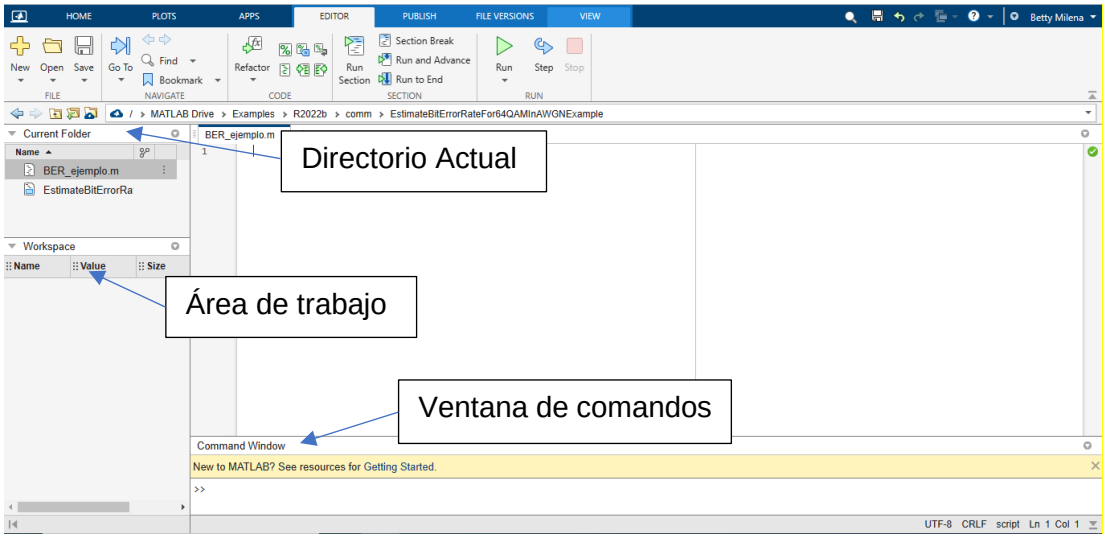


Figura 1.5 Ventana apertura de Matlab

2 METODOLOGÍA

El desarrollo del proyecto se basó en cuatro objetivos, iniciando con la investigación de los fundamentos del Espectro Ensanchado y sus dos técnicas DSSS y FHSS, continuando con la selección de las herramientas correctas del *software* Matlab para simulación y finalmente, la comparación entre ambas técnicas de SS. La metodología que se empleó se muestra en la Figura 2.1 Metodología empleada.

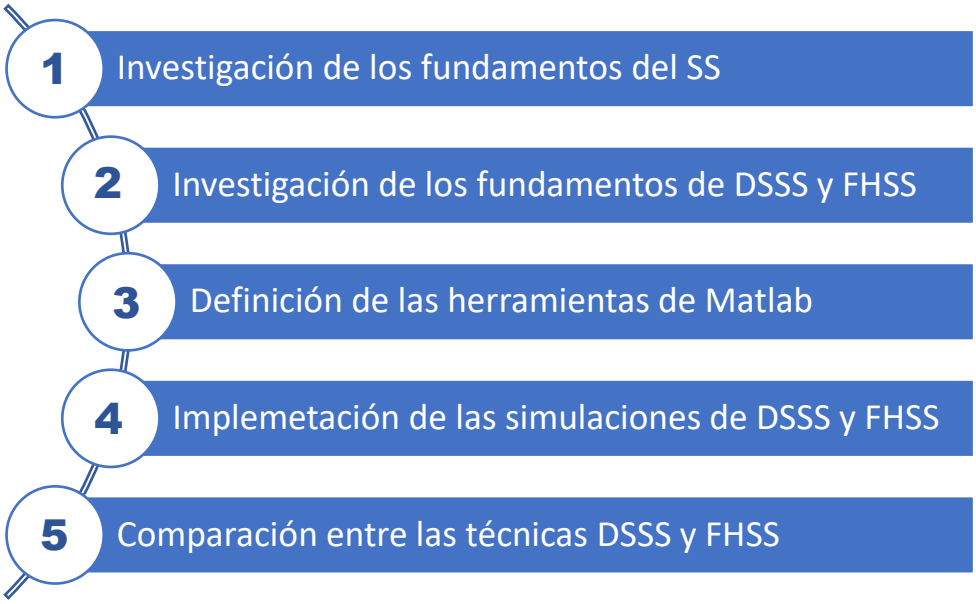


Figura 2.1 Metodología empleada

Para la selección de las herramientas en Matlab se realizó un análisis a la ventana de comandos y las diferentes funciones disponibles en Matlab para el procesamiento de señales, apta para implementar el código necesario para la simulación y analizar los gráficos obtenidos.

Se analizó el funcionamiento de los sistemas DSSS y FHSS, tanto el transmisor como el receptor, para poder simularlos de manera correcta con los comandos correspondientes. Además, se indagó sobre la funcionalidad de cada uno de los comandos y los elementos que formaron parte de los códigos como las estructuras de repetición y los condicionales; así como los arreglos matriciales.

Seleccionada la ventana de comandos, las aplicaciones y herramientas que nos ofrece Matlab, se implementó dos simulaciones por separado, mediante dos *scripts* para representar ambas técnicas y obteniendo gráficas para realizar un comparativo entre DSSS y FHSS en términos de espectrograma y densidad espectral de potencia. Con los

resultados obtenidos se puede llegar a las conclusiones pertinentes del análisis de las dos técnicas de Espectro Ensanchado.

3 RESULTADOS

Se procede a explicar los requerimientos necesarios para la simulación, además de una breve explicación de los fundamentos del Espectro Ensanchado y sus dos técnicas DSSS y FHSS, así como se describe la funcionalidad de Matlab y los códigos empleados en simulación. Por último, se establece un comparativo de los resultados obtenidos mediante la simulación.

3.1 Principios de funcionamiento de DSSS y FHSS

El fundamento básico de las técnicas de SS es el ensanchamiento de la señal que va a ser transmitida en un rango de frecuencias mayor que el ancho de banda mínimo que se requiere para la transmisión de la señal que contiene la información.

Ambas técnicas usan el código PN que logra aumentar el ancho de banda y reducir la densidad espectral, debido a que la señal resultante se asemeja al ruido, así pues, para todos los receptores cercanos a la trasmisión será una señal de ruido a excepción del receptor que forma parte del sistema, por lo tanto, el código PN dictamina el sincronismo entre receptor y transmisor.

Principio de DSSS

El Espectro Ensanchado por Secuencia Directa se basa en una secuencia de bits redundantes que es una secuencia pseudoaleatoria PN, cada uno de los bits de esa secuencia acompaña respectivamente a cada uno de los bits de entrada o la señal de la información a recibir; el sistema será más resistente a interferencias si el código pseudoaleatorio es largo lo más recomendable es 11 bits, según el estándar IEEE 802.11 que hace referencia a la terminología y explicaciones inalámbricas [3].

Principio de FHSS

En el Espectro Ensanchado por Salto de Frecuencia, el ensanchamiento se produce en un conjunto de frecuencias, la señal varia de una frecuencia a otra, es decir que la información durante varios intervalos de tiempo se transmite en frecuencias distintas.

La secuencia de saltos está determinada según el código PN que tanto transmisor y receptor deben conocer, es decir debe existir un sincronismo entre ambos.

En la Figura 3.1 Representación de los sistemas FHSS y DSSS respecto a el tiempo vs la frecuencia se representa gráficamente la representación de las técnicas FHSS y DSSS en el dominio del tiempo versus la frecuencia.

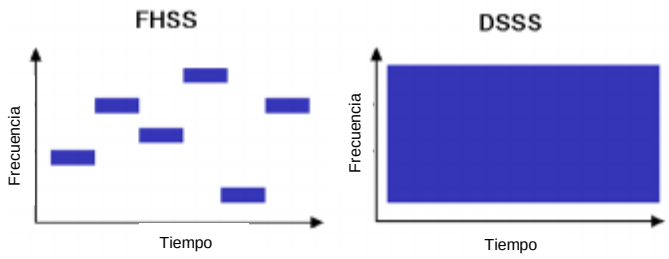


Figura 3.1 Representación de los sistemas FHSS y DSSS respecto a el tiempo vs la frecuencia [19]

3.2 Determinación de las herramientas de Matlab

Ventana de Comandos

Es la ventana principal de Matlab y por ello se abre cuando el *software* se ejecuta [20]. Está en la vista por defecto de Matlab formando parte de su ventana de apertura junto a cuatro de las ventanas que se abren por defecto [18]. La ventana de comandos permite como su nombre lo indica introducir comandos y se identifica con el símbolo >> [21].

Si se ejecuta un archivo *.m, escrito anteriormente, los resultados también se presentarán en esta ventana.

Archivos*.m

Son los denominados *M-files,* son *scripts* que cuentan con una extensión de archivos propia de Matlab, denominada *.m, permiten escribir los comandos en un editor de texto y ejecutarlos posteriormente y ya no existe la necesidad de escribirlos en la ventana de comandos [22].

Para ejecutar un archivo*.m se escribe su nombre en la ventana de comandos o se selecciona la opción del menú suprior que permite abrir los archivos almacenados en Matlab. En la Figura 3.2 se puede ver un listado de archivos *.m, que son almacenados en la plataforma de Matlab, estos pueden ser creados por el usuario o subidos a la misma.

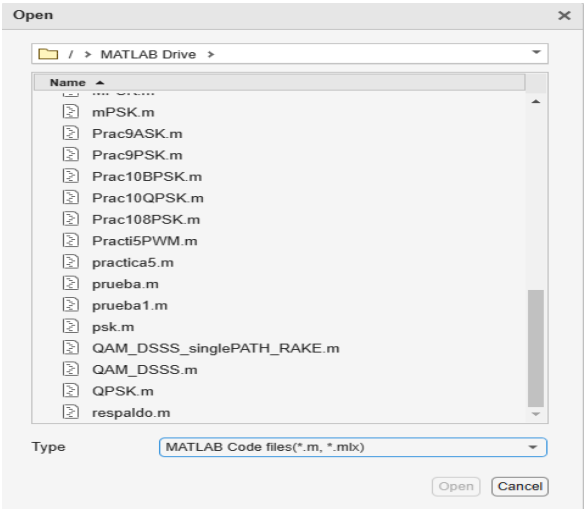


Figura 3.2 Listado de archivos*.m

Al llamar un *script* con extensión, denominado archivo*.m, se ejecutan los comandos que se hallen en el archivo; tienen la posibilidad de emplear variables que se hallen en el área de trabajo o pueden crear variables nuevas sobre las cuales trabajar y se almacenaran en el área de trabajo para ser usadas en cálculos futuros. Pueden ejercer comandos y funciones. En la Figura 3.3 se observa la ejecución de un *script* y la ventana de comandos en acción.

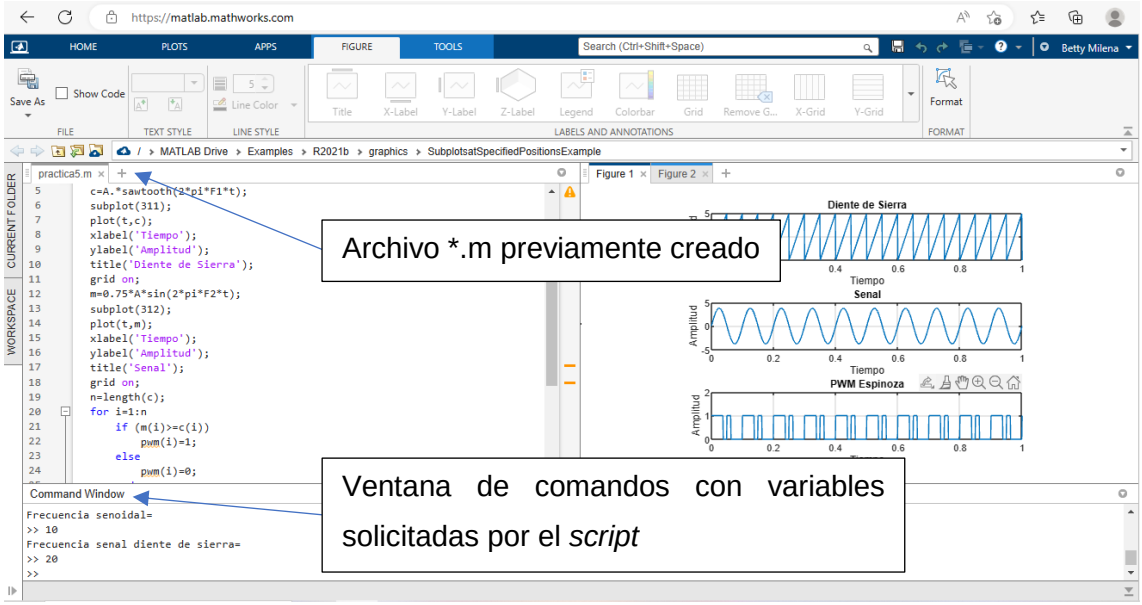


Figura 3.3 Ejecución de un M-file

En el caso de las simulaciones para las técnicas DSSS y FHSS, en el ANEXO V se destallan los comandos empleados para el desarrollo de los archivos DSSS.m y FHSS.m.

Signal Analyzer

Es una herramienta interactiva está integrada en Matlab, cuya funcionalidad es permitir visualizar, procesar, medir, analizar y comparar las señales en el dominio del tiempo y de la frecuencia [23], con esto se puede detectar sus patrones y tendencias y caracterizarlas y a los sistemas de procesamiento de las mismas.

Con esta aplicación se puede:

- Tener fácil acceso a las señales que se estén procesando en el área de trabajo
- Personalizar las funciones de procesamiento
- Operar con varias señales con distinta duración de tiempo
- Visualizar y comparar simultáneamente varias señales, espectrogramas, formas de onda, etc.
- Medir datos y realizar estadísticas de las señales [23]

En la Figura 3.4 se observa el entorno de la aplicación y funcionalidad del analizador de señales, como se puede analizar en una misma vista se hallan varias señales en diferentes planos, en la sección izquierda la frecuencia en función del tiempo y a la derecha la potencia espectral en función del tiempo.

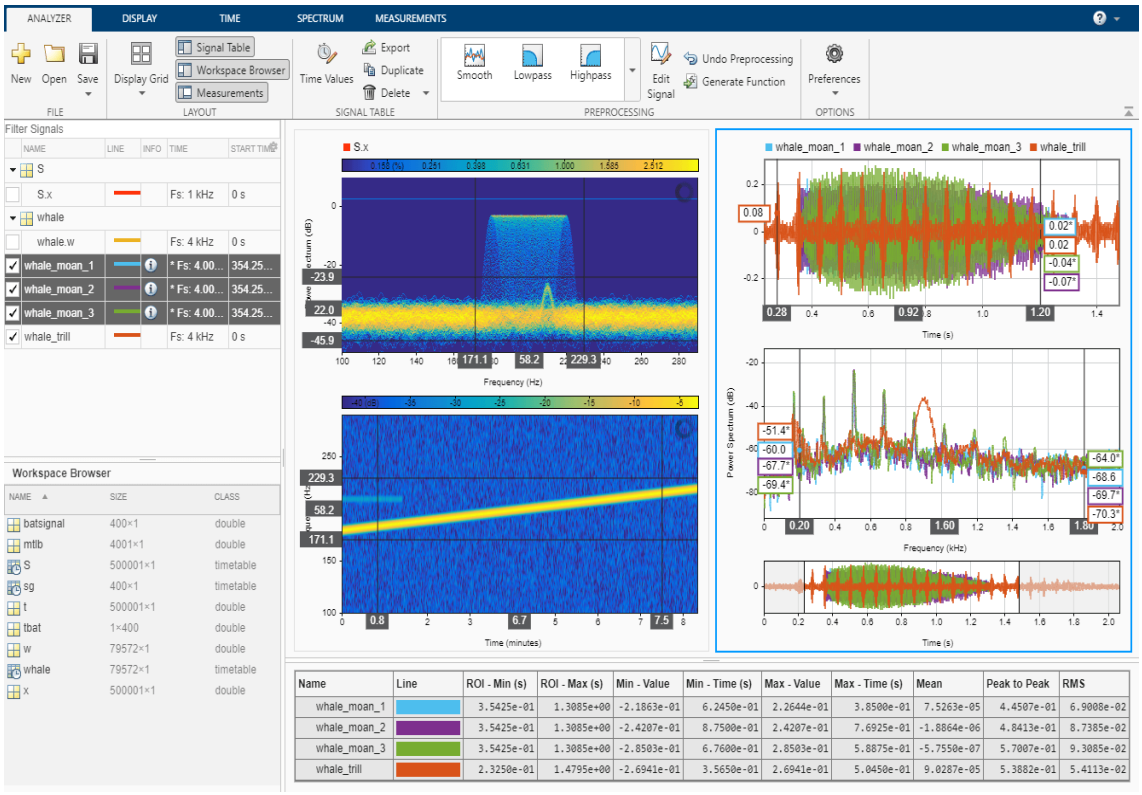


Figura 3.4 Aplicación *Signal Analyzer* en Matlab [23]

El procedimiento para trabajar con la aplicación es:

1. Seleccionar las señales a analizar, estas aparecerán en el área de trabajo, deben contener información de tiempo como tasas de muestreo, vectores, arreglos o expresiones.
2. Procesar las señales aplicando filtros pasa bajo o pasa alto, filtros pasa banda o eliminar la banda de las señales, Matlab permite automatizar las opciones y personalizar las funciones de procesamiento.
3. Explorar las señales mediante representaciones, mediciones y comparaciones de datos, espectrogramas o sus escalogramas.

   Se requiere hallar las características en el dominio del tiempo, de la frecuencia y el dominio tiempo-frecuencia [24]. Para ello se usan las funciones que se muestran en la Figura 3.5.

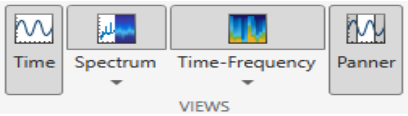


Figura 3.5 Dominios que permite analizar *Signal Analyzer*

Los archivos*.m para las simulaciones requieren de operadores y caracteres especiales que se muestran en la Tabla 3.1:

Tabla 3.1 Operadores y caracteres especiales de Matlab para la simulación de DSSS y FHSS

Operadores y caracteres	Funciones que realizan
.*	Se da la multiplicación de arreglos elemento por elemento.
./	Se da la división de los arreglos elemento por elemento. Es denominada como la división por la derecha.
/	Resuelve un sistema de ecuaciones lineales, para ello los arreglos deben tener el mismo número de columnas.
*****	Permite el producto matricial.
.	El punto representa un separador decimal, los operadores que contengan un punto realizan sus operaciones elemento a elemento.
==	Permite determinar una igualdad.
=	Permite asignar valores a una variable.

Tabla 3.1 Operadores y caracteres especiales de Matlab para la simulación de DSSS y FHSS

Operadores y caracteres	Funciones que realizan
;	El punto y coma sirve para eliminar la salida de una línea de comandos [25].
[ ]	Los corchetes permiten la construcción y concatenación de arreglos y la creación de matrices vacías.
( )	Los paréntesis especifican la precedencia de ciertas operaciones.
{ }	Las llaves permiten construir y asignar arreglos de celdas [25].
>	El mayor que permite comparar elementos o arreglos y devuelve un valor lógico 1 si la expresión es verdadera y 0 si se da el caso contrario.
,	La coma separa elementos de un arreglo.
' '	Las comillas simples permiten crear arreglos de caracteres.
%	El porcentaje permite comentar un texto, así se indica que no es ejecutable.

3.3 Implementación de los escenarios para la simulación

Para implementar las simulaciones se hizo uso de la plataforma Matlab en su versión en línea, que permite acceder a los recursos de la plataforma sin descargarlos e instalarlos; se almacena y gestiona los archivos desde cualquier lugar y navegador solo iniciando sesión.

Se emplearon dos *script*s para implementar las simulaciones de ambas técnicas:

Implementación de la simulación para DSSS

Para iniciar con la simulación se emplean los dos comandos utilizados para eliminar el rastro de cualquier otra variable ajena al *script* y se crean los parámetros principales, como la cantidad de bits *N*, la frecuencia de la señal de entrada *Fs* y la frecuencia de la señal portadora *Fp* que debe ser mayor a *Fs*, el periodo del bit *t_bit* medido en segundos y el parámetro *fcp* por el que se multiplicará la frecuencia de la señal de entrada, se muestran en la Figura 3.6:

```
clc
clear all;

% Se establecen los parámetros

N=20;

Fs = 100;
Fp = 1000;

fcp = 4;



t_bit=1;
```

Figura 3.6 Comandos iniciales y definición de los parámetros

En la Figura 3.7 se fija la secuencia de bits de entrada que puede ser aleatoria, en este caso se selecciona dicha secuencia para realizar las respectivas comparaciones y se convierte los bits de entrada a valores binarios mediante un bucle con el comando *for,* en el que se fija la variable *b* en la que se almacenaran los valores ya sea 1 o 0, mediante la instrucción *if:*

```
b_entrada = [1 1 1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 1];



%Se pasa los bits de entrada a valores binarios

for b = 1:length(b_entrada)
   if(b_entrada(b)==0)
       b_entrada(b)=-1;
   end

end
```

Figura 3.7 Secuencia de entrada y conversión a datos binarios

Como se aprecia en la Figura 3.8 se establece la señal de entrada mediante los comandos que se usan para arreglos de matrices y basándose en la secuencia de en la secuencia de bits de entrada.

Aplicando los comandos *figure* y *stem* se procede a crear un diagrama que represente a los bits de entrada, estableciendo los límites de los ejes mediate el comando *axis,* y los títulos de la gráfica y ejes con los comandos *title*, *xlabel* y *ylabel* respectivamente:

```
snl_entrada =  repmat(b_entrada,fcp,1);

snl_entrada =  reshape(snl_entrada,1,[]);

% Se grafican los datos de entrada como unos y ceros
figure(1);
stem(b_entrada);
axis([0 length(b_entrada)*t_bit 0 1]);
title('Datos binarios de entrada');
ylabel('Amplitud');
xlabel('Tiempo');
```

Figura 3.8 Señal de entrada y grafica de la secuencia de entrada

Se crea una secuencia de bits aleatorios para establecer el código pseudoaleatorio y se procede a convertirlos en datos binarios (Figura 3.9):

```
codi_pn = randi([0,1],1,length(b_entrada)*fcp);



for b = 1:length(codi_pn)
   if(codi_pn(b)==0)
         codi_pn(b)=-1;
   end
end
```

Figura 3.9 Código pseudoaleatorio y su conversión a datos binarios

Se establece la señal portadora una onda sinusoidal y su periodo *t* que se relaciona con la frecuencia fijada en los parámetros iniciales *Fp*.

Se crea la señal ensanchada por secuencia directa partiendo del producto de la señal de entrada y la señal portadora, como se observa en Figura 3.10:

```
t = 0:1/Fs:(t_bit-1/Fs);



p0 = -1*cos(2*pi*Fp*t);
p1 = cos(2*pi*Fp*t);
portadora = [];


snl_dsss=snl_entrada.*codi_pn;
```

Figura 3.10 Señal portadora y señal DSSS

En la Figura 3.11 se inicializando la señal modulada y se muestra el proceso de modulación de la misma mediante BPSK, como la técnica emplea dos símbolos se utilizan las sentencias *if* y *elseif* para representar a ambos símbolos desfasados con un bit de información cada uno.

Y finalizando se establece la señal portadora, que se inicializo en cero, se fija mediante *p1* para el análisis:

```
snl_bpsk=[];
for i = 1:length(snl_dsss)
    if (snl_dsss(i)==1)
        snl_bpsk=[snl_bpsk p1];
    elseif (snl_dsss(i)==-1)
        snl_bpsk=[snl_bpsk p0];
    end
    portadora=[portadora p1];

end
```

Figura 3.11 Modulación BPSK

A continuación, se inicia con el proceso de demodulación para ello se estable una variable que almacenara la señal de salida y se establece el proceso con una sentencia de bucle, mediante *for*, fijando la variable *i* para almacenar los bits del código pseudoaleatorio y los comandos *if* y *else* para que se cumplan las condiciones dentro del bucle. Este proceso se muestra en la Figura 3.12 donde la señal demodulada será el producto de la señal de salida y la señal portadora.

También se estable un arreglo para almacenar el mensaje de salida basándose en el resultado obtenido de la demodulación y se simula la correlación entre la señal portadora y la señal demodulada en el receptor mediante el comando *sum*.

Como último paso se condiciona a que se almacene el resultado y se exprese un mensaje a la salida (Figura 3.12).

```
snl_salida=[];
for i = 1:length(codi_pn)

    if(codi_pn(i)==1)
        snl_salida=[snl_salida snl_bpsk((((i-1)*length(t))+1):i*length(t))];
    else
        snl_salida=[snl_salida (-1)*snl_bpsk((((i-1)*length(t))+1):i*length(t))];
    end
end
snl_demod = snl_salida.*portadora;

result=[];
for i=1:length(b_entrada)
   x = length(t)*fcp;

   cx = sum(portadora(((i-1)*x)+1:i*x).*snl_demod(((i-1)*x)+1:i*x));


   if(cx>0)
       result=[result 1];
   else
       result=[result -1];
   end
end
% Se establecen los mensajes de salida, con areglos
mensaje1=repmat(result,fcp,1);
mensaje1=reshape(mensaje1,1,[]);
```

Figura 3.12 Proceso de demodulación y establecimiento de una señal de salida

En la Figura 3.13 se fija una variable que almacene la cantidad de elementos del código pseudoaleatorio para establecer su periodo *tpn* y mediante el parámetro establecido inicialmente se dispone el periodo de la señal de entrada *t_sentra*:

```
cod_pn_ele=length(codi_pn);


tpn = linspace(0,length(b_entrada)*t_bit-t_bit/fcp,cod_pn_ele);
t_sentra = 0:t_bit/fcp:length(b_entrada)*t_bit-t_bit/fcp;
```

Figura 3.13 Se establece la longitud y el periodo del código pseudoaleatorio

Se procede a graficar la señal de entrada respecto a su periodo de tiempo y de igual manera el código pseudoaleatorio, como se aprecia en la Figura 3. **14**. La señal ensanchada por secuencia directa se grafica respecto al periodo del código pseudoaleatorio y el mensaje de salida respecto al periodo de la señal de entrada:

```
% Secuencia original de bits
figure(2);
stairs(t_sentra,snl_entrada,'linewidth',2)
title('Secuencia de bits de la Señal de Entrada')
axis([0 length(b_entrada)*t_bit -1 1]);
ylabel('Amplitud');
xlabel('Tiempo');

% Código Pseudoaleatorio
figure(3);
stairs(tpn,codi_pn,'linewidth',2)
title('Código Pseudoaleatorio');
axis([0 length(b_entrada)*t_bit -1 1]);
ylabel('Amplitud');
xlabel('Tiempo');

% Señal ensanchada
figure(4);
stairs(tpn,snl_dsss,'linewidth',2)
title('Señal modulada');
axis([0 length(b_entrada)*t_bit -1 1]);
ylabel('Amplitud');
xlabel('Tiempo');

% Mensaje con el código Pseudoaleatorio
figure(5);
stairs(t_sentra,mensaje1,'linewidth',2)
title('Mensaje recibido con el código pseudoaleatorio')
axis([0 length(b_entrada)*t_bit -1 1]);
ylabel('Amplitud');
xlabel('Tiempo');
```

Figura 3. 14 Gráficas de las señales respecto a sus periodos

Se procede a realizar las gráficas en el dominio de la frecuencia apoyándose de la herramienta de la Transformada Rápida de Fourier de Matlab, como se muestra en la Figura 3.15. Inicialmente se grafica por separado el espectro de frecuencias de la señal de entrada, el código pseudoaleatorio y de la señal ensanchada.

```
% Graficas en el dominio de la frecuencia
fre = linspace(-Fs/2,Fs/2,1024);
figure(6);
plot(fre,abs(fftshift(fft(snl_entrada,1024))),'linewidth',2);
title('Espectro de Frecuencia de la señal de entrada');
xlabel('Frecuencia (Hz)');
ylabel('Tiempo(s)')
% La fft se usa para mostrar características de la frecuencia
%
figure(7);
plot(fre,abs(fftshift(fft(codi_pn,1024))),'linewidth',2);
title('Espectro de Frecuencia del Código Pseudoaleatorio');
xlabel('Frecuencia (Hz)');
ylabel('Tiempo(s)')
%
figure(8);
plot(fre,abs(fftshift(fft(snl_dsss,1024))),'linewidth',2);
title('Espectro de Frecuencia de la señal de Espectro Ensanchado');
xlabel('Frecuencia (Hz)');
ylabel('Tiempo(s)')
```

Figura 3.15 Gráficas de los Espectros de Frecuencia de las señales

Para finalizar se realiza una gráfica que represente la densidad espectral de potencia de la señal ensanchada, mediane los comandos *fft* y *xcorr.* Se normalizan los valores obtenidos (Figura 3.16):

```
z = abs(fft(xcorr(snl_dsss)));

minz=min(z);
z1=z-minz;
maxz=max(z1);
z_normal=z1/maxz;

figure(10);
plot(z_normal);
title('Densidad Espectral de Potencia para la señal ensanchada por Secuencia Directa')
xlabel('Frecuencia (Hz)');
ylabel('Potencia (dBm/Hz)');
```

Figura 3.16 Se plantea y gráfica la Densidad Espectral de Potencia de la señal DSSS

El *script* completo se ubica en la sección de ANEXOS, en el ANEXO V en conjunto con el diagrama de flujo en el ANEXO III.

Implementación de la simulación para FHSS

De igual manera se inicia con la eliminación de variables o funciones de otros *scripts* y con la definición de los parámetros iniciales, como la cantidad de bits *N*, la cantidad de muestras por segundo *T*, la duración de la secuencia, así como la secuencia de bits de entrada, la señal de entrada y la señal portadora, establecidos en la Figura 3.17:

```
clc
clear all;
% Estacblecen los parámetros
N=20;
T=120;
t_bit=N*T;
d_bit=T/20;

b_entrada = [1 1 1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 1];
senal_entrada=[];
senal_portadora=[];

t=0:2*pi/(T-1):2*pi;
```

Figura 3.17 Comandos iniciales y definición de los parámetros

En la Figura 3.18 muestra cómo se convierte los bits de la secuencia de entrada a datos binarios mediante un bucle creado por el comando *for* y complementado con *if* y *else.*

Acto seguido se fijan la señal portadora y la señal de entrada:

```
for k = 1 :N
    if b_entrada(1,k)==0
        snl=-ones(1,T);
    else
        snl=ones(1,T);
    end
    %
    p=cos(t);
    senal_entrada = [senal_entrada snl];
    senal_portadora = [senal_portadora p];

end
```

Figura 3.18 Conversión a datos binarios de la secuencia de entrada y creación de las señales portadora y, de entrada

Se grafican los datos de la secuencia de entada como bits mediante el comando *stem* y se representan como una secuencia mediante el comando *plot,* como se muestra en la Figura 3.19:

```
% Se grafica la señal de datos binarios
figure(1);
stem(b_entrada,'linewidth',2);
title('Datos binarios de entrada');
ylabel('Amplitud');
xlabel('Tiempo');
% Se grafica la señal de entrada
figure(2);
plot(senal_entrada,'linewidth',2);
axis([-100 2400 -1 1]);
title('Secuencia Original de Bits');
ylabel('Amplitud');
xlabel('Tiempo');
```

Figura 3.19 Gráficas de los bits y de la secuencia de bits de entrada

Como se observa en la Figura 3.20, se realiza el proceso de la modulación mediante el producto de la señal de entrada con la señal portadora y mediante *figure(3),* se grafica el resultado:

```
bpskmod_senal=senal_entrada.*senal_portadora;


figure(3);

plot(bpskmod_senal,'linewidth',2);
axis([-100 2400 -1 1]);
title('Señal BPSK modulada');
ylabel('Amplitud');
xlabel('Tiempo');
```

Figura 3.20 Proceso de modulación BPSK y gráfica de la señal obtenida

A continuación, se determinan los periodos de las diferentes señales portadoras para los diferentes saltos de frecuencias, se definen seis periodos para las seis frecuencias, ver Figura 3.21.

```
t1=0:2*pi/9:2*pi;
t2=0:2*pi/19:2*pi;
t3=0:2*pi/29:2*pi;
t4=0:2*pi/39:2*pi;
t5=0:2*pi/59:2*pi;
t6=0:2*pi/119:2*pi;
```

Figura 3.21 Seis periodos distintos

La Figura 3.22 muestra la generación de las seis señales portadoras con funciones senoidales para sus respectivos periodos, mediante arreglos de vectores, donde la dimensión del vector indica la frecuencia con la que se repetirá la señal portadora.

```
portadora1=cos(t1);
portadora1=[portadora1 portadora1 portadora1 portadora1 portadora1 portadora1 portadora1 portadora1 portadora1 portadora1 portadora1 portadora1]
portadora2=cos(t2);
portadora2=[portadora2 portadora2 portadora2 portadora2 portadora2 portadora2];
portadora3=cos(t3);
portadora3=[portadora3 portadora3 portadora3 portadora3];
portadora4=cos(t4);
portadora4=[portadora4 portadora4 portadora4];

portadora5=cos(t5);
portadora5=[portadora5 portadora5];
portadora6=cos(t6);
```

Figura 3.22 Señales portadoras

Partiendo de un bucle se crean seis casos para establecer el valor de la señal ensanchada en función de las seis señales portadoras creadas posteriormente (Figura 3.23):

```
for n=1:N
    p=randi([1 6],1,1);

    switch(p)
        case(1)
            senal_ensa=[senal_ensa portadora1];
        case(2)
            senal_ensa=[senal_ensa portadora2];
        case(3)
            senal_ensa=[senal_ensa portadora3];
        case(4)
            senal_ensa=[senal_ensa portadora4];
        case(5)
            senal_ensa=[senal_ensa portadora5];
        case(6)
            senal_ensa=[senal_ensa portadora6];
    end
end
```

Figura 3.23 Casos para seleccionar la señal de salida

En Figura 3.24 se procede a representar la señal ensanchada mediante *figure(4)* y el comando *plot*:

```
figure(4);
plot(senal_ensa);
axis([-100 2400 -1 1]);
title('Señal portadora');
ylabel('Amplitud');
xlabel('Tiempo');
```

Figura 3.24 Gráfica de la señal ensanchada

A continuación, se fija la señal con saltos de frecuencias mediante el producto de la señal modulada mediante BPSK y la señal ensanchada y se grafica el resultado. El código de este proceso se muestra en la Figura 3.25.

```
senal_salto_fre=bpskmod_senal.*senal_ensa;

figure(5);
plot(senal_salto_fre);
axis([-100 2400 -1 1]);
title('Señal con salto de Frecuencia');
ylabel('Amplitud');
xlabel('Tiempo');
```

Figura 3.25 Señal FHSS y respectiva gráfica

Se inicia con el proceso de demodulación entre la señal con saltos de frecuencias y la señal ensanchada y se grafica el resultado, como se muestra en la Figura 3.26:

```
bpsk_demo=senal_salto_fre./senal_ensa;
figure(6)
plot(bpsk_demo);
axis([-100 2400 -1 1]);
title('Señal Demodulada por BPSK mediante el Espectro ensanchado');
ylabel('Amplitud');
xlabel('Tiempo');
```

Figura 3.26 Proceso de demodulación y gráfica de la señal demodulada mediante BPSK

En la Figura 3.27 se recupera la señal original partiendo de la operación de división por la derecha de la señal demodulada y la señal portadora y se grafica el resultado en *figure(7)*:

```
original_BPSK_signal=bpsk_demo./senal_portadora;
figure(7)
plot(original_BPSK_signal,'linewidth',2);
axis([-100 2400 -1 1]);
title('Transmisión de la secuencia Original de Bits');
ylabel('Amplitud');
xlabel('Tiempo');
```

Figura 3.27 Señal original y gráfica de la secuencia original de bits

Se establece la frecuencia de la señal de entrada *Fs* que será empleada para realizar la grafico del espectro de frecuencia de la señal ensanchada *fre*, partiendo del comando que se muestra en la Figura 3.28. Y en la gráfica 8 mediante *figure(8)* se grafica el espectro de frecuencia de la señal ensanchada usando *fftshift* y *fft* que en Matlab representan la Transformada Rápida de Fourier.

```
Fs = 100;
fre = linspace(-Fs/2,Fs/2,1024);



figure(8);
plot(fre,abs(fftshift(fft(senal_ensa,1024))),'linewidth',2);
title('Espectro de Frecuencia de la señal de Espectro Ensanchado');
xlabel('Frecuencia (Hz)');
ylabel('Tiempo(s)')
```

Figura 3.28 Frecuencia de la señal de entrada y gráfica del Espectro de Frecuencia de la señal ensanchada mediante FHSS

Para finalizar se realiza la gráfica en la Figura 3.29 de la densidad espectral de potencia partiendo de los comandos *fft* y *xcorr*.

```
z = abs(fft(xcorr(senal_ensa)));

figure(9);
plot(z);
axis([0 840 0 1.1*max(z)]);
title('Densidad Espectral de Potencia para la señal ensanchada');
xlabel('Frecuencia (Hz)');
ylabel('Potencia (dBm/Hz)')
```

Figura 3.29 Se plantea y gráfica la Densidad Espectral de Potencia

Para evidenciar el *script* completo revisar el ANEXO VI junto con el diagrama de flujo del ANEXO IV.

3.4 Comparaciones entre las de las simulaciones y sus resultados entre DSSS y FHSS

Resultados de DSSS

En la **Figura 3.30**, se puede observar que la secuencia de la señal de entrada tiene una duración del bit de un segundo, mientras que la Figura 3.31 se puede analizar la secuencia del código pseudoaleatorio que el tiempo de duración del bit y es menor en comparación con la primera (250 milisegundos); siendo así que la velocidad de transmisión de la secuencia de entrada es menor que la de la secuencia de la señal modulada y el código pseudoaleatorio.

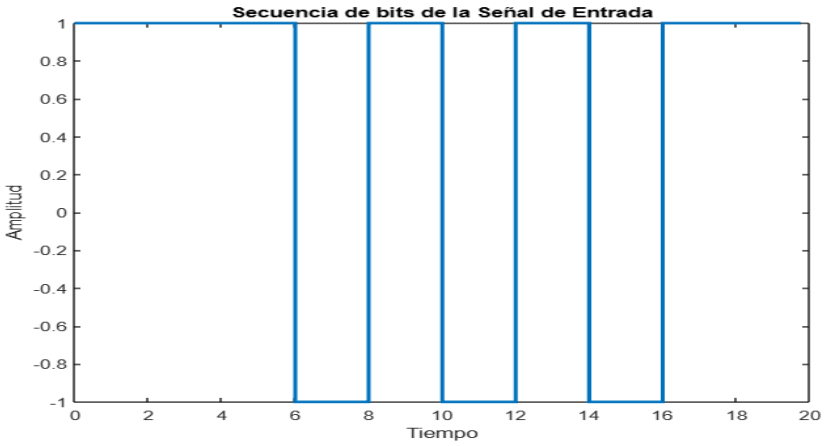


Figura 3.30 Secuencia de la señal de entrada

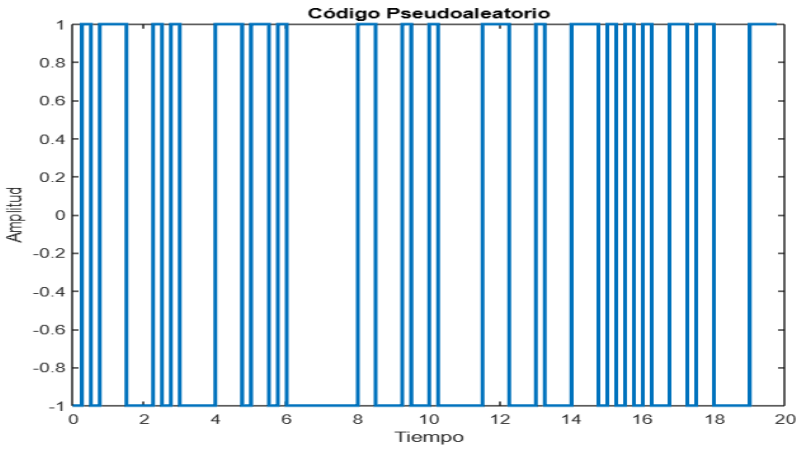


Figura 3.31 Secuencia del código pseudoaleatorio

En la Figura 3.32 se obtiene después de la demodulación exactamente la misma secuencia de la señal de entrada de la Figura 3.30, comprobando que la transmisión de la información es correcta.

Figura 3.32 Secuencia del mensaje recibido

Analizando la Figura 3.33, respecto al espectro de potencia con la Figura 3.34 y la Figura 3.35 se puede observar que el Espectro de Frecuencia de la señal de entrada se ha ensanchado en la señal del código pseudoaleatorio y de la señal ensanchada y modulada mediante BPSK.

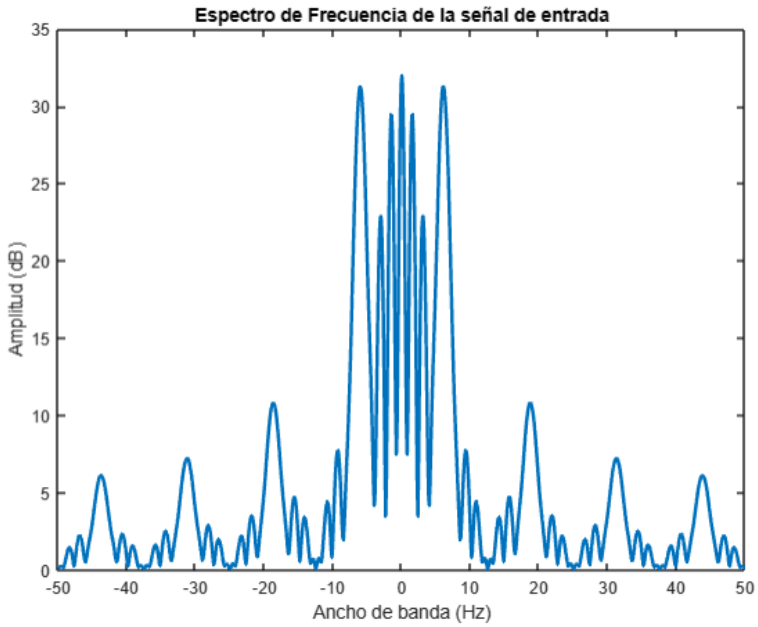


Figura 3.33 Espectro de Frecuencia de la señal de entrada

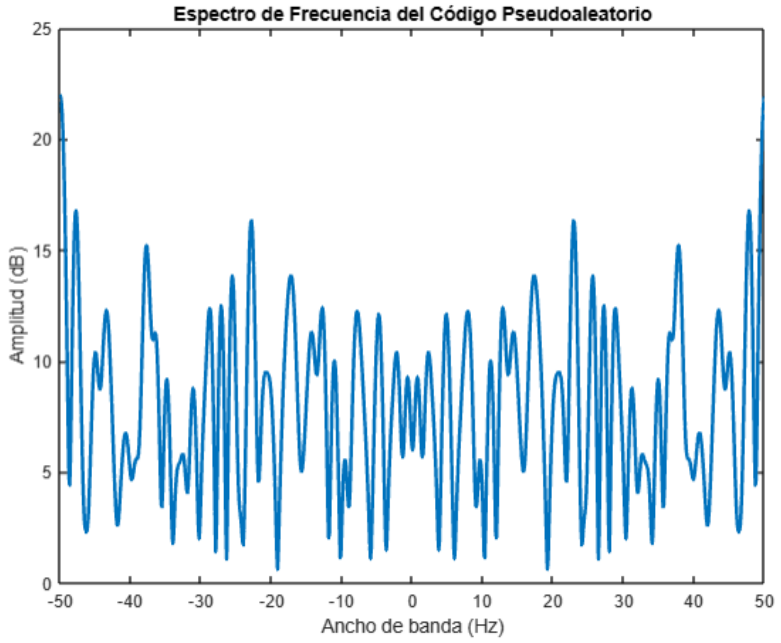


Figura 3.34 Espectro de Frecuencia del código pseudoaleatorio

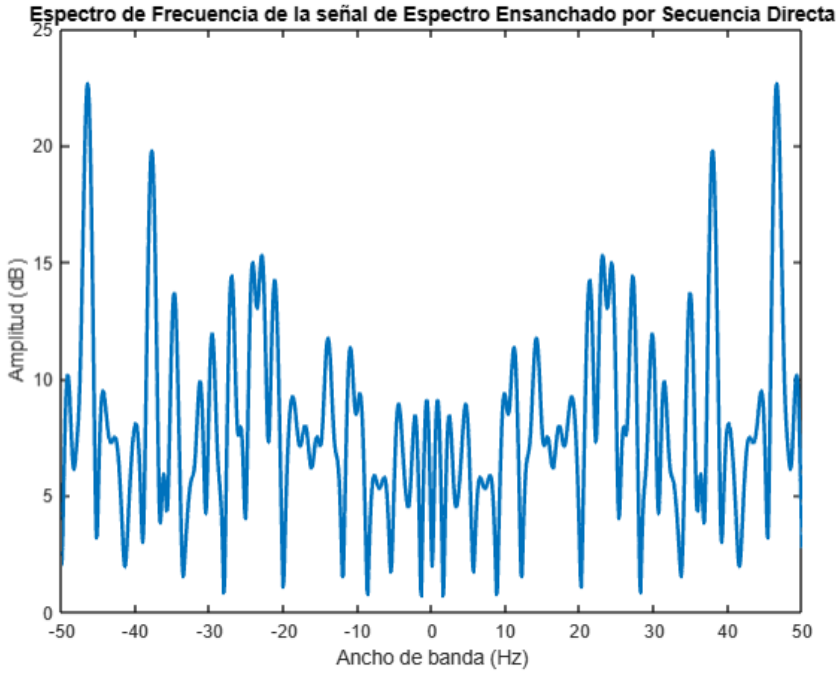


Figura 3.35 Espectro de Frecuencia de la señal ensanchada por secuencia directa

En la Figura 3.36 se comparan los tres espectros y se puede evidenciar el ensanchamiento del espectro al realizar la operación establecida por la compuerta XOR de la señal de entrada junto con el código pseudoaleatorio.

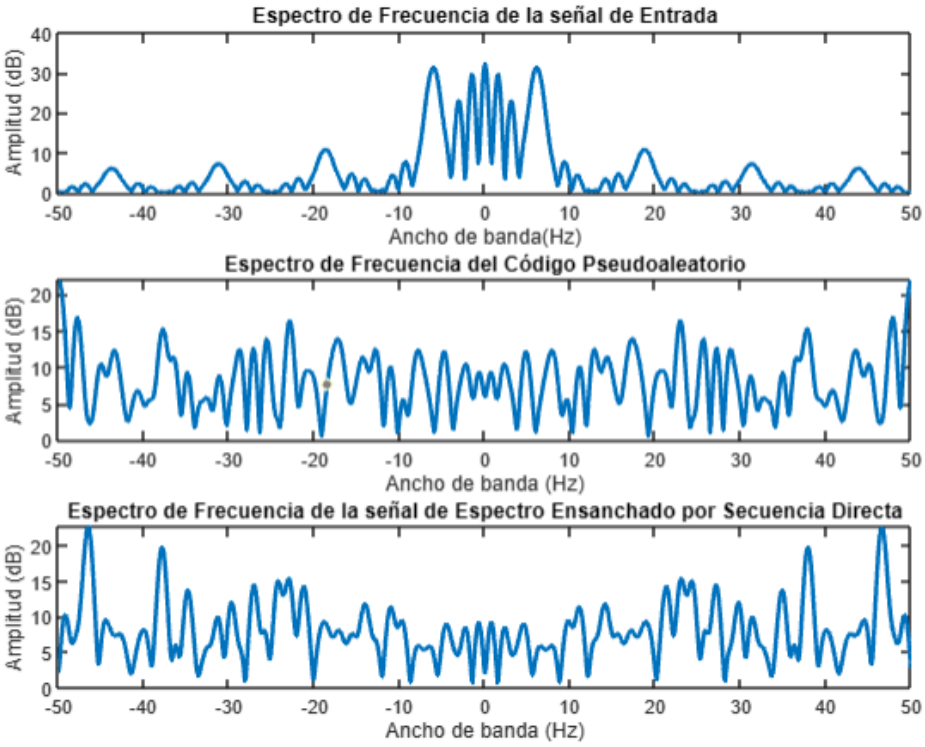


Figura 3.36 Representación del espectro de las tres señales

Resultados de FHSS

Para el análisis de la simulación para la técnica FHSS es pertinente considerar que al igual que en la simulación de DSSS la secuencia de bits de entrada y la secuencia de bits a la salida será la misma gracias a los procesos de modulación y demodulación BPSK, que presenta una ventaja frente a la inmunidad al ruido, lo que significa que se puede transmitir de manera confiable.

En la Figura 3.37 a la señal modulada mediante BPSK se puede analizar que hay una variación en la fase de la señal y comparando con la señal portadora de la Figura 3.38 existe una variación de fase a distintos periodos, en este caso en 6 periodos debido al número de señales portadoras establecidas y en comparación a la Figura 3.39 que muestra a la señal ensanchada por salto de frecuencias se la misma variación en la fase, pero a diferentes amplitudes, mismas dictaminadas por las señales portadoras.

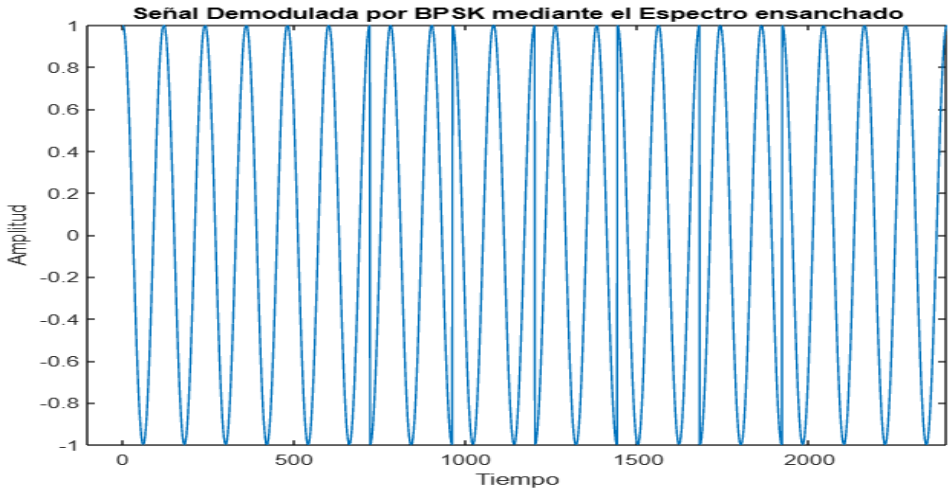


Figura 3.37 Señal modulada mediante BPSK

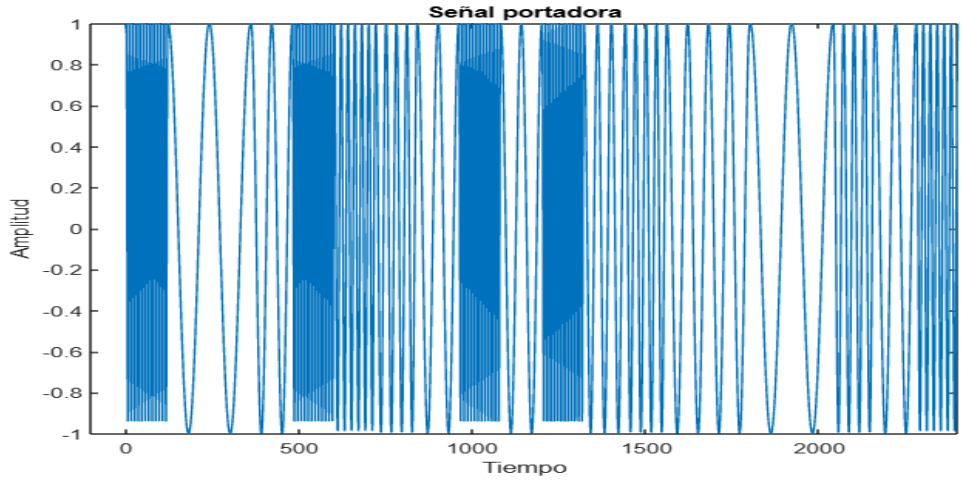


Figura 3.38 Señal portadora

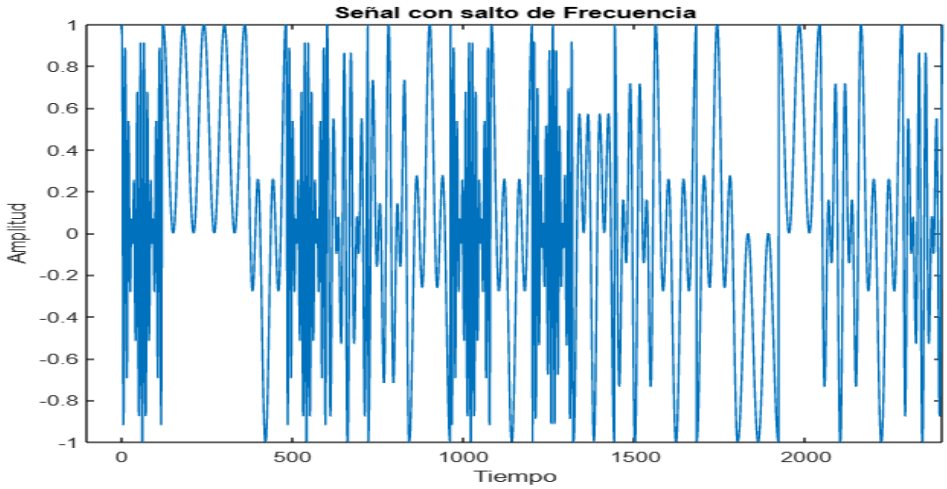


Figura 3.39 Señal ensanchada con salto de frecuencia

Comparación respecto al Espectro de Frecuencia

En la Figura 3.40 se gráfica el espectro de la señal ensanchada donde se presenta un mayor ancho de banda que la señal de entrada y en comparación con la Figura 3.41 se muestra un espectro de frecuencia resultando del salto entre frecuencias que cambian en el tiempo, mostrándose un conjunto de picos en diferentes frecuencias.

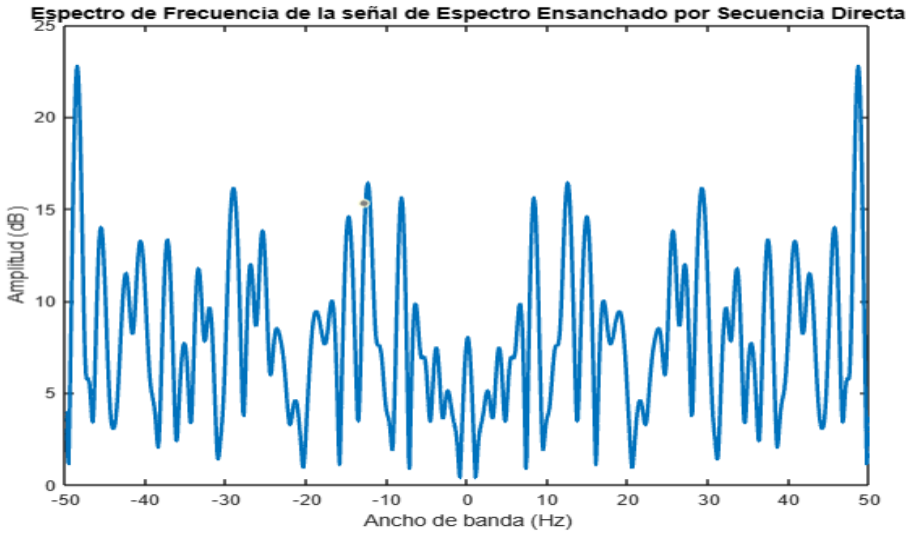


Figura 3.40 Espectro de Frecuencia de la señal ensanchada por Secuencia Directa

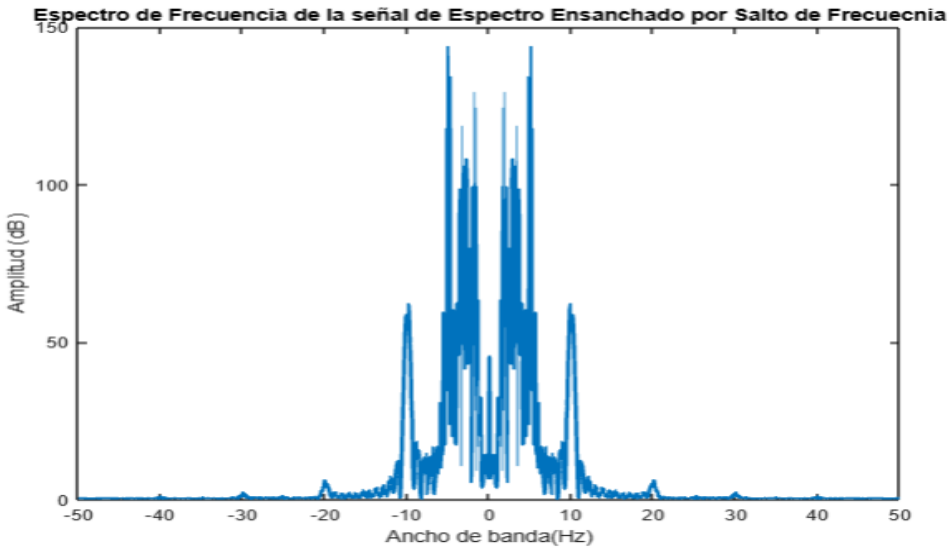


Figura 3.41 Espectro de la frecuencia de la señal ensanchada con salto de frecuencia

Comparación respecto a la Densidad Espectral de Potencia

Respecto al Densidad Espectral de Potencia FHSS consume más potencia que DSSS debido a que se necesitan múltiples frecuencias para transmitir la señal. DSSS es más

eficiente en el uso del espectro que FHSS ya que aplica técnicas de procesamiento de la señal para transmitir la información en una única frecuencia de portadora.

De la Figura 3.42 y de la Figura 3.43 se puede concluir que FHSS tiene mayor densidad espectral y ancho de banda que la técnica DSSS, puesto que la señal ensanchada por secuencia directa ocupa aproximadamente 160 (Hz), mientras que la señal ensanchada por salto de frecuencia ocupa de 5000 (Hz).

Respecto al ancho de banda DSSSS utiliza un ancho de banda fijo para transmitir la información y FHSS utiliza múltiples frecuencias con un ancho de banda más amplio.

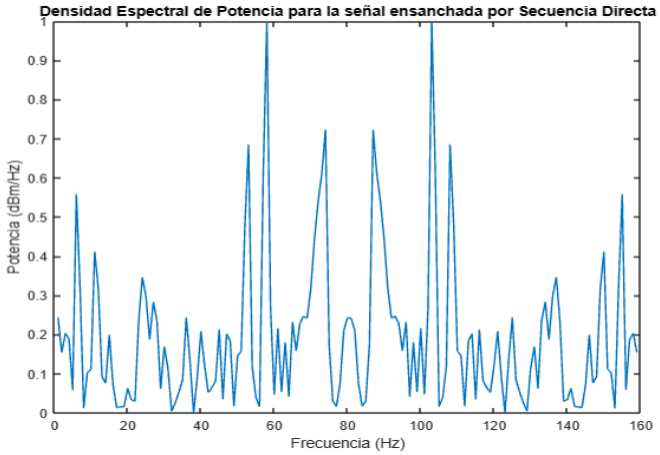


Figura 3.42 Espectro de la Densidad de Potencia de DSSS

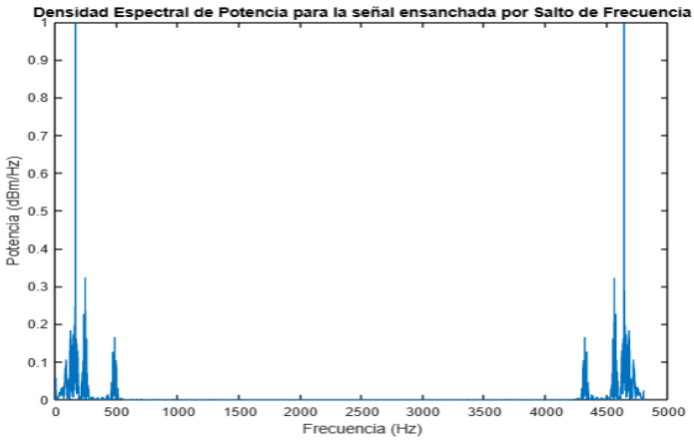


Figura 3.43 Espectro de la Densidad de Potencia de FHSS

Comparación en el dominio Tiempo-Frecuencia

Estableciendo una comparación en el análisis del dominio de Tiempo-Frecuencia en la Figura 3.44 se observa que la técnica DSSS opera en una única frecuencia para 75 muestras, en cambio en la Figura 3.45 la técnica FHSS cambia de frecuencias para

alrededor de 2400 muestras debido a una secuencia predeterminada (código pseudoaleatorio).

DSSS divide transmite la señal portadora mediante una única frecuencia, mientras que FHSS cambia frecuentemente la señal portadora de manera aleatoria en un rango de frecuencias establecido.

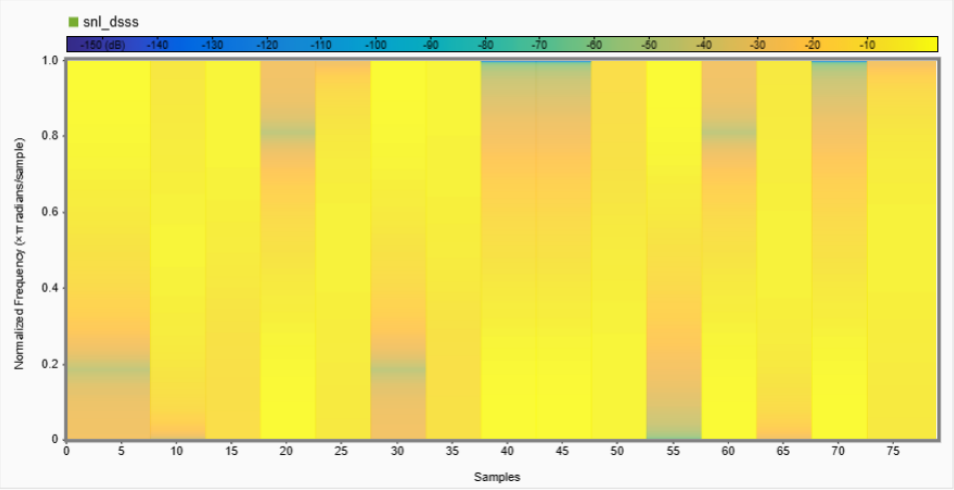


Figura 3.44 Dominio Tiempo-Frecuencia de DSSS

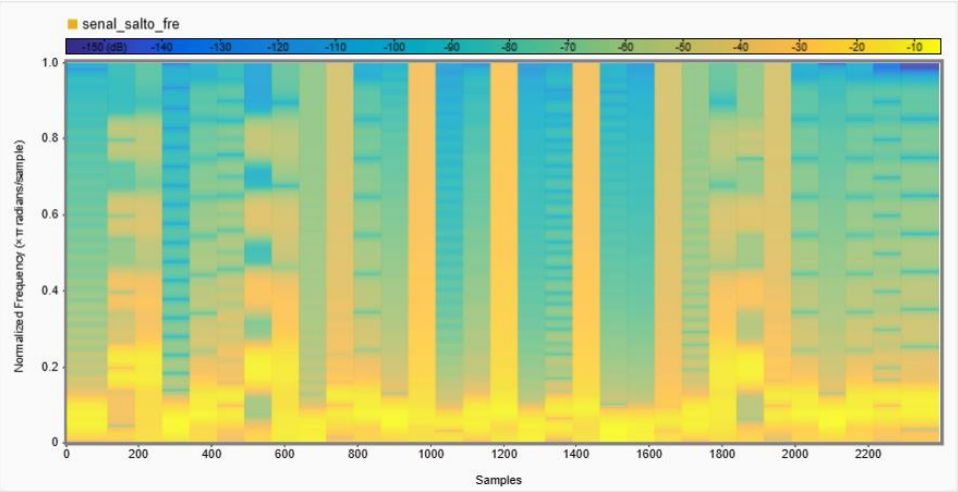


Figura 3.45 Dominio Tiempo -Frecuencia de FHSS

4 CONCLUSIONES

- Las técnicas de SS se fundamentan en que la señal transmitida tenga un ancho de banda mayor al ancho de banda mínimo requerido para transmitir la información, DSSS lo lleva a cabo mediante el cambio de la secuencia de datos y FHSS cambiando la frecuencia de la señal portadora.

- Matlab es una plataforma interactiva apta para comprender el funcionamiento de varios sistemas debido a la flexibilidad e interactividad que brindan sus herramientas y aplicaciones, que permiten analizar, procesar y visualizar los datos mediante gráficos.
- Matlab posibilita diseñar y simular sistemas partiendo de su variedad de funciones y herramientas, mediante las cuales se realiza el procesamiento y el análisis de señales, como el análisis de espectral y el análisis en el dominio tiempo-frecuencia.
- Mediante las gráficas obtenidas de las simulaciones es posible extraer características de las señales, lo que se traduce a obtener información destacada de la señal, útil en el caso de simular las técnicas de SS para establecer las respectivas comparaciones entre ambas.
- En los sistemas DSSS se ocupa la banda de frecuencia entera para transmitir, mientras que en un sistema FHSS de espectro ensanchado se usa una pequeña porción del ancho de banda al transmitir, lo que se traduce en un mayor ancho de banda para DSSS.
- FHSS tienen una mayor densidad espectral que DSSS, siendo así que DSSS es más eficiente en el uso del espectro, puesto que la señal se transmite usando todo el ancho de banda disponible, FHSS en cada salto de frecuencia consume potencia.

5 RECOMENDACIONES

- Es recomendable comprender el funcionamiento de los sistemas que emplean las técnicas de SS para poder diseñar los entornos de simulación de una manera más efectiva y relacionando los parámetros establecidos con las etapas que forman parte del sistema, por ejemplo, el proceso de modulación mediante BPSK, con la señal de entrada y la señal portadora que participan directamente para obtener la señal modulada.
- Se recomienda realizar un análisis previo a las herramientas y aplicaciones de la plataforma para verificar si cumplen con los requerimientos necesarios de las simulaciones de las técnicas de SS y obtener el mayor beneficio de las mismas.
- Se recomienda estudiar la función que cumple cada comando empleado en los scripts para las simulaciones, puesto que se hará más eficiente la codificación y establecer una semejanza a los sistemas de las técnicas FHSS y DSSS.

6 REFERENCIAS BIBLIOGRÁFICAS

[1] W. Suárez Quinteros, «ESTUDIO DE FACTIBILIDAD DE UNA RED INALÁMBRICA DE ÁREA METROPOLITANA (MAN) EN BASE A LA TECNOLOGÍA DE ESPECTRO DISPERSO, APLICABLE PARA UNA INSTITUCIÓN BANCARIA DENTRO DE LA CIUDAD DE QUIT,» Quito, 2001.

[2] Unión Internacional de Telecomunicaciones, «UTILIZACIÓN DE TÉCNICAS DE ESPECTRO ENSANCHAD,» de *RECOMENDACIÓN UIT-R SM.1055-0**, vol. 1, 1994, pp. 1-3.

[3] J. Colorado y R. Hurberto, «TRANSMISIÓN DE INFORMACIÓN USANDO LA MODULACIÓN (DSSS),» Pereira, 2011.

[4] F. P. Chiquin G y G. E. Pazmay G, «Diseño de un Sistema para el monitoreo y control de los servicios sobre las tecnologías inalámbricas: WLL, LMDS y Spreas Spectrum,» Quito, 2003.

[5] A. Fox y R. Llamas, «ENLACES INLAMBRICOS SPREAD SPECTRUM COMO SOLUCION DE ULTIMO KILOMETRO,» Cartagena de Indias, 2003.

[6] J. Mera Medina y M. David Manzano, «SIMULACIÓN Y ANÁLISIS DEL NIVEL FÍSICO DEL ESTÁNDAR IEEE 802.11B,» Popayán, 2007.

[7] H. Córdova y C. Patricia, «Estudio, Modelamiento y Simulación de Sistemas de Espectro Ensanchado Secuencia Directa y Salto De Frecuencia,» *Revista Tecnológica ESPOL,* vol. 18, nº 1, pp. 1-8, 14 Enero 2013.

[8] A. Agudelo, P. Bernal y E. Quintero, «Espectro Ensanchado por Saltos de Frecuencia para,» *Scientia et Technica ,* vol. 18, nº 1, pp. 255-232, Abril 2013.

[9] M. Cubos, S. Sánchez Urrieta, M. Lara Barrón y V. Kontorovich, «Comparación del desempeño (BER) de un sistema DS-CDMA empleando códigos de dispersión de Máxima Longitud, Gold y Kasami evaluado mediante simulación,» México, D.F, 1999.

[10] J. P. Cortez Mosquera, «COMPARACIÓN DE LAS TÉCNICAS DE CODIFICACIÓN DE CANAL DE QUINTA GENERACIÓN NR-LDPC Y NR-POLAR

EN UN CANAL AWGN CON MODULACIÓN BPSK Y CUANTIZACIÓN DE PUNTO FIJO MEDIANTE UNA SIMULACIÓN EN MATLAB,» Quito, 2021.

[11] A. Torres y L. Herrera, «DISEÑO E IMPLEMENTACIÓN DE UN MODULADOR Y DEMODULADOR BPSK,» Cartagena de Indias, 2011.

[12] G. Amouzad Mahdiraji y A. Fauzi Abas, «Advanced Modulation Formats and Multiplexing Techniques for Optical Telecommunication Systems,» de *Trends in Telecommunications Technologies*, 2010.

[13] «Colombia suma móvil,» SUMA móvil S.A.S., 2023. [En línea]. Available: https://sumamovil.com.co/glosario/espectro-de-frecuencia-que-es-para-que-sirve/. [Último acceso: 20 Enero 2023].

[14] M. Paz, G. Friedrich y C. Galasso, «Procesamiento de señal visualizado sobre un espectrograma,» *Revista elektron,* vol. 4, nº 1, pp. 35-39, 2020.

[15] J. Echeverry, C. Lemus y Á. Orozco, «ANÁLISIS DE LA DENSIDAD ESPECTRAL DE POTENCIA EN REGISTROS MER,» 2007.

[16] J. Benítez y J. Hueso, «Introducción a MATLAB».

[17] The MathWorks, Inc., « MathWorks,» MATLAB®, 2023. [En línea]. Available: https://la.mathworks.com/help/matlab/learn_matlab/scripts-and-functions.html. [Último acceso: 05 Febrero 2023].

[18] H. Moore, MATLAB para ingenieros, PRENTICE HALL MEXICO, 2009.

[19] D. Kyle, J.-F. Levesque y I. Mas, «Diseño, modelado y evaluación de un sistema de comunicaciones FHSS de 2,4 GHz para NarcisSat,» Sranford.

[20] A. Gilat, Matlab Una introducción con ejemplos prácticos, Barcelona: REVERTÉ, S.A., 2006, pp. 1-7.

[21] The MathWorks, Inc., «MathWorks,» MATLAB®, 2023. [En línea]. Available: https://es.mathworks.com/help/matlab/learn_matlab/desktop.html. [Último acceso: 05 Febrero 2023].

[22] A. Elizalde, Introducción a Matlab∗, 2005, pp. 1-10.

[23] The MathWorks, Inc., «MathWorks,» MATLAB®, 2023. [En línea]. Available: https://es.mathworks.com/help/signal/ref/signalanalyzer-app.html. [Último acceso: 05 Febrero 2023].

[24] The MathWorks, Inc., «MathWorks,» MATLAB®, 2022. [En línea]. Available: https://la.mathworks.com/help/signal/ug/using-signal-analyzer-app.html. [Último acceso: 15 Febrero 2023].

[25] The MathWorks, Inc., «MathWorks,» MATLAB®, 2022. [En línea]. Available: https://la.mathworks.com/help/matlab/matlab_prog/matlab-operators-and-special-characters.html. [Último acceso: 05 Febrero 2023].

[26] The MathWorks, Inc., «MathWorks,» MATLAB®, 2022. [En línea]. Available: https://la.mathworks.com/help/matlab/ref/randi.html. [Último acceso: 05 Febrero 2022].

[27] The MathWorks, Inc., «MathWorks,» MATLAB®, 2022. [En línea]. Available: https://la.mathworks.com/help/matlab/ref/stem.html. [Último acceso: 05 Febrero 2023].

7 ANEXOS

La lista de los **Anexos** se muestra a continuación:

ANEXO I. Certificado de originalidad

ANEXO II. Enlaces

ANEXO III. Diagrama de flujo del código fuente de la simulación de DSSS en Matlab

ANEXO IV. Diagrama de flujo del código fuente de la simulación de FHSS en Matlab

ANEXO V. Código fuente de la simulación de DSSS en Matlab

ANEXO VI. Código fuente de la simulación de FHSS en Matlab

ANEXO VII. Comandos empleados para la simulación de DSSS y FHSS en Matlab

ANEXO I: CERTIFICADO DE ORIGINALIDAD

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 3 de marzo de 2023

De mi consideración:

Yo, ANDRÉS FERNANDO REYES CASTRO, en calidad de Director del Trabajo de Integración Curricular titulado SIMULACIÓN DE DSSS Y FHSS EN MATLAB elaborado por la estudiante BETTY MILENA ESPINOZA ESPINOZA de la carrera en TECNÓLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 19%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el link del informe generado por la herramienta Turnitin.

https://ev.turnitin.com/app/carta/es/?lang=es&o=2028023517&u=1087149605&student_user=1

Atentamente,

ANDRÉS FERNANDO REYES CASTRO

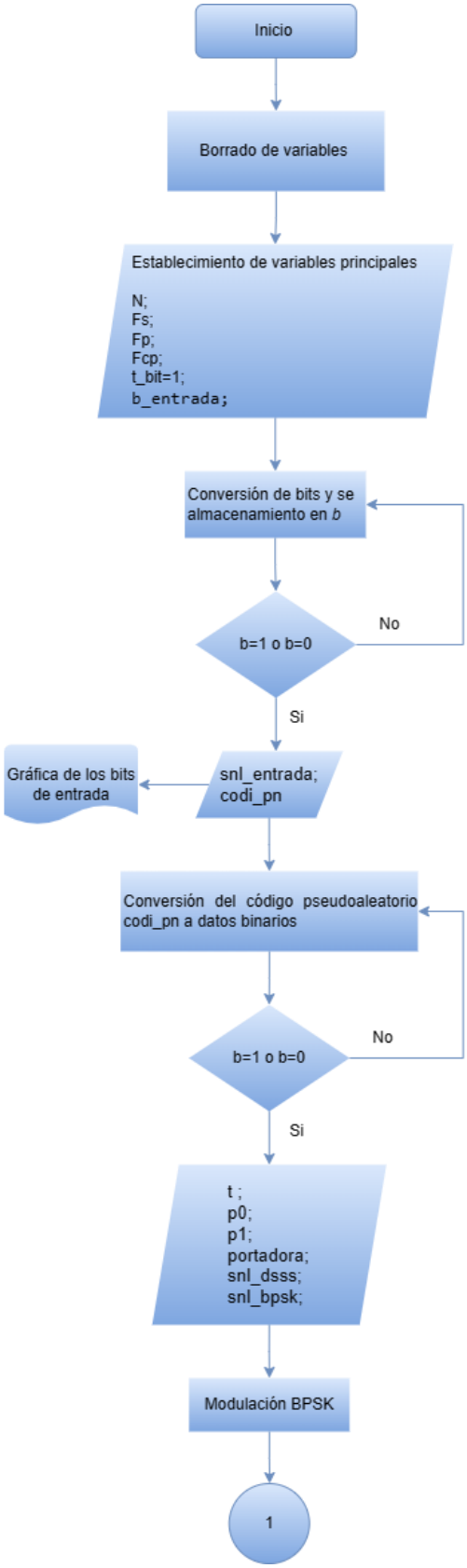
Docente

Escuela de Formación de Tecnólogos

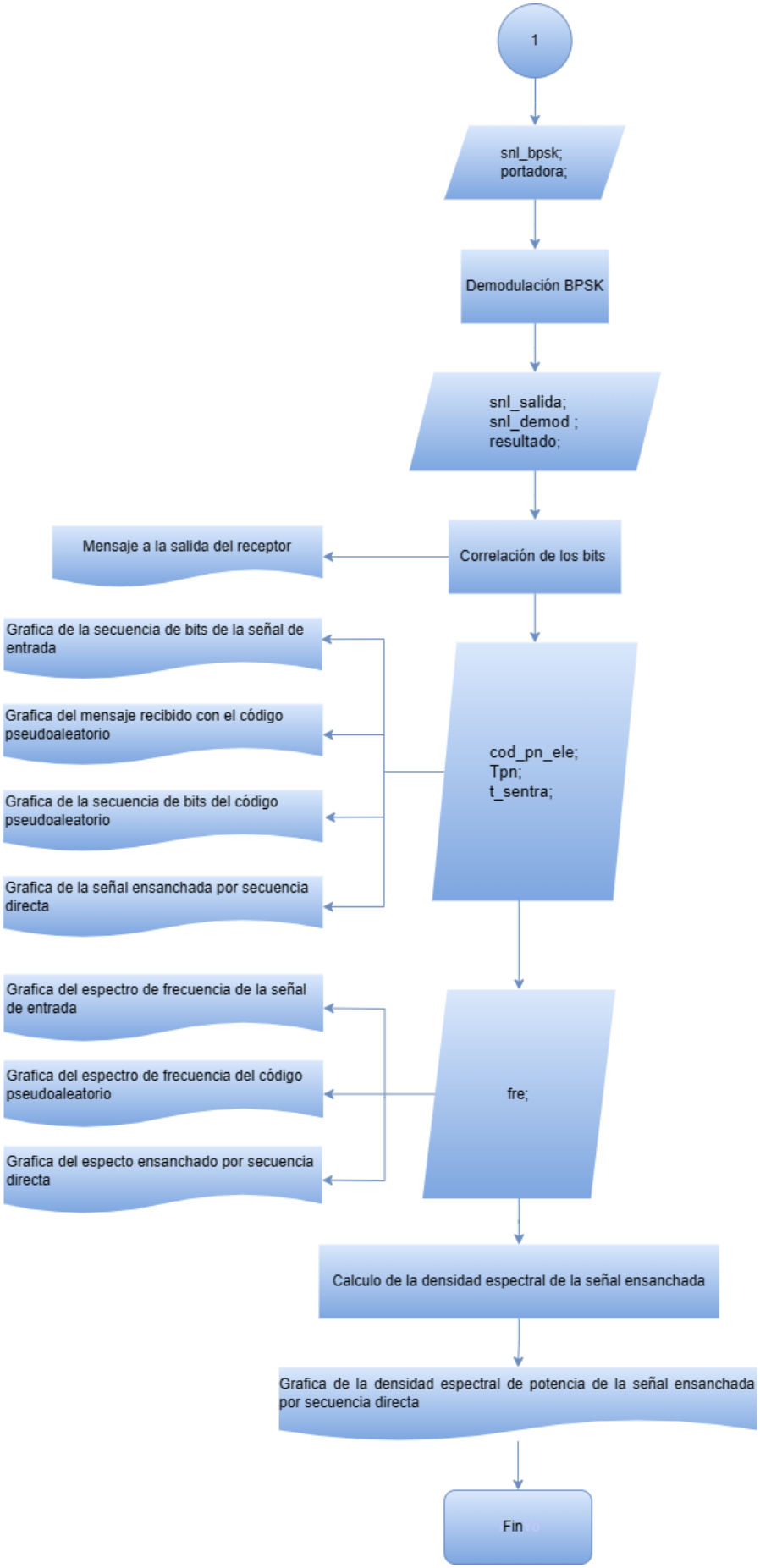
ANEXO II: Enlaces

Anexo II.I Código QR de la simulación y pruebas de funcionamiento

ANEXO III: Diagrama de flujo del código fuente de la simulación de DSSS en Matlab

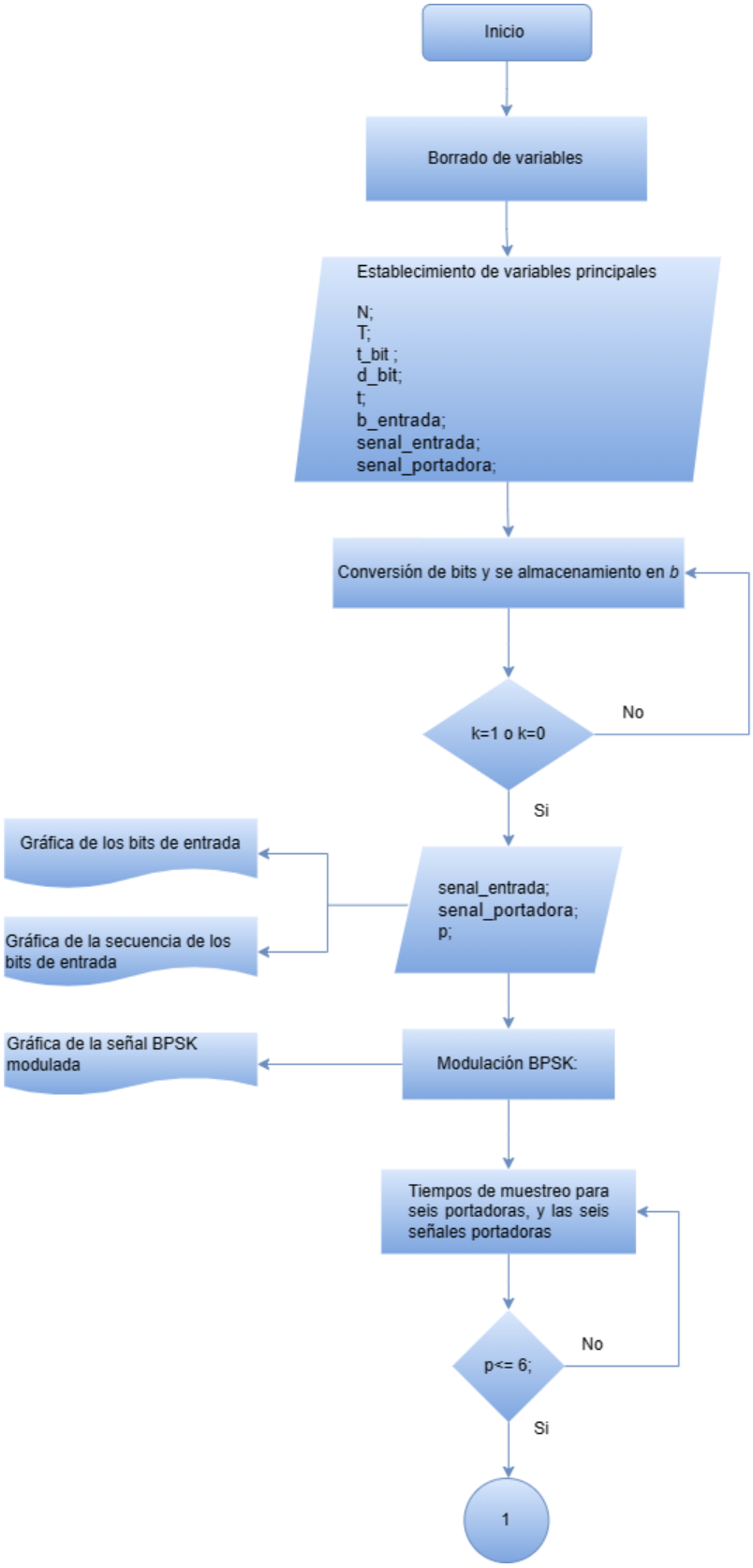


Anexo III. I Diagrama de flujo del código fuente de la simulación de DSSS en Matlab

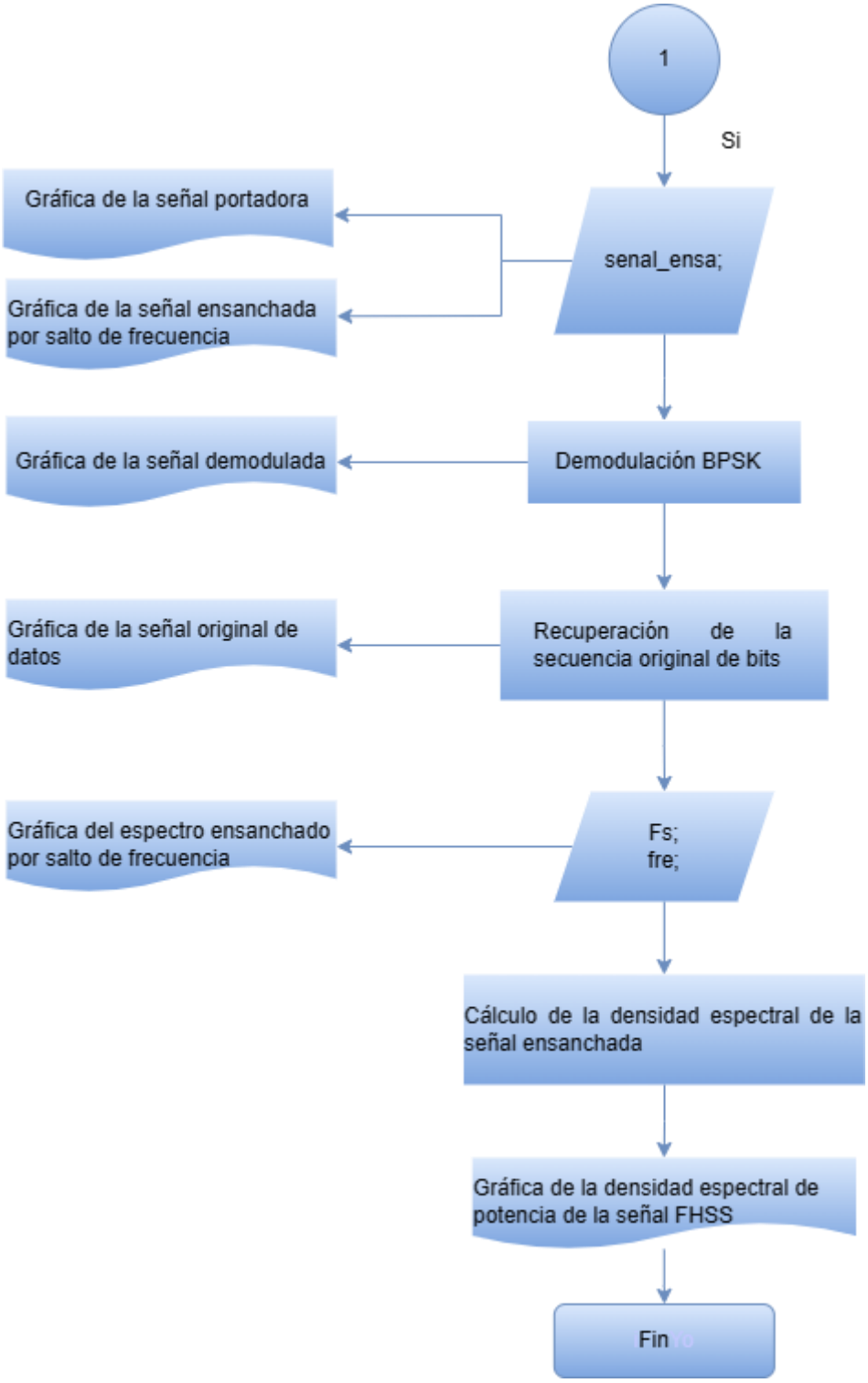


Anexo III.I Diagrama de flujo del código fuente de la simulación de DSSS en Matlab

ANEXO IV: Diagrama de flujo del código fuente de la simulación de FHSS en Matlab



Anexo IV.I Diagrama de flujo del código fuente de la simulación de FHSS en Matlab



Anexo IV.II Diagrama de flujo del código fuente de la simulación de FHSS en Matlab

ANEXO V: Código fuente de la simulación de DSSS en Matlab

```
%Se inicializa eliminando todo rastro de otro script

clc

clear all;

% Se establecen los parámetros

N=20;

% Cantidad de bits

Fs = 100;

% Frecuencia de la señal de entrada o señal moduladora

Fp = 1000;

% Frecuencia de la señal portadora

% La señal portadora tiene la frecuencia mayor que la señal moduladora

fcp = 4;

% Este es un valor que sera multiplicado a la frecuencia de la señal de entrada

t_bit=1;

% Duración del bit en segundos

b_entrada = [1 1 1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 1];

% Se establece la secuencia de bits de entrada

%Se pasa los bits de entrada a valores binarios

for b = 1:length(b_entrada)

% Sea b = bit

  if(b_entrada(b)==0)

     b_entrada(b)=-1;

  end

end
```

```
% Los comandos remat y reshape son para arreglos de matrices

snl_entrada =  repmat(b_entrada,fcp,1);

% Comando repmat sirve para crear copias de valores

snl_entrada =  reshape(snl_entrada,1,[]);

% Comando usado para cambiar la matriz manteniendo los elementos

% Se grafican los datos de entrada como unos y ceros

figure(1);

stem(b_entrada);

axis([0 length(b_entrada)*t_bit 0 1]);

% El comado stem hace un diagrama de tallo

title('Datos binarios de entrada');

% Se establecen los títulos en los ejes y de la gráfica

ylabel('Amplitud');

xlabel('Tiempo');

% Se genera un código pseudoaleatorio

codi_pn = randi([0,1],1,length(b_entrada)*fcp);

% Código pseudo aleatorio con mayor frecuencia que la frecuencia de la señal original
o de entrada

% Esto garantiza que cada bit de la señal sea cifrado por una mayor cantidad de bits

for b = 1:length(codi_pn)

  if(codi_pn(b)==0)

% Bit cero

      codi_pn(b)=-1;

% Bit uno

  end

end

% Se crea la señal portadora para modularla (multiplicarla con la señal de entrada)
```

```
t = 0:1/Fs:(t_bit-1/Fs);

% Se establece un periodo en relación con el tiempo del bit y la frecuencia de la señal de entrada para la señal portadora

% Se establece la señal portadora de la señal con la frecuencia establecida en los parámetros

p0 = -1*cos(2*pi*Fp*t);

% También puede ser seno

p1 = cos(2*pi*Fp*t);

portadora = [];

% Se inicializa la señal portadora

% Se ensancha a la señal

snl_dsss=snl_entrada.*codi_pn;

% Se establece la señal DSSS producto de la señal de entrada y el código pseudoaleatorio

% Es la señal ensanchada

% Se empieza con la modulación BPSK

snl_bpsk=[];

for i = 1:length(snl_dsss)

% Este comando permite obtener los elementos de la señal ensanchada

   if (snl_dsss(i)==1)

      snl_bpsk=[snl_bpsk p1];

% Se toma la parte positiva de la portadora

   elseif (snl_dsss(i)==-1)

      snl_bpsk=[snl_bpsk p0];

% Se toma la parte negativa de la portadora

   end

   portadora=[portadora p1];

% Se establece la primera portadora, o parte positiva útil para el análisis
```

```
end

% Se inicia el proceso de demodulación

snl_salida=[];

for i = 1:length(codi_pn)

% Este comando permite obtener los elementos de la señal recibida por el recepto

    if(codi_pn(i)==1)

        snl_salida=[snl_salida snl_bpsk((((i-1)*length(t))+1):i*length(t))];

% Parte positiva

    else

        snl_salida=[snl_salida (-1)*snl_bpsk((((i-1)*length(t))+1):i*length(t))];

 % Parte negativa

    end

end

snl_demod = snl_salida.*portadora;

% La operación .* es un producto entre matrices se multiplican elementos por elementos en las matrices

result=[];

for i=1:length(b_entrada)

   x = length(t)*fcp;

% Se crea una variable para almacenar la cantidad (longitud) de elementos del periodo de tiempo con fcp

   cx = sum(portadora(((i-1)*x)+1:i*x).*snl_demod(((i-1)*x)+1:i*x));

% Con estas operaciones se conserva la longitud al multiplicar la señal portadora y la señal demodulada

     if(cx>0)

       result=[result 1];

   else

       result=[result -1];
```

```
  end
end
% Se establecen los mensajes de salida, con areglos
mensaje1=repmat(result,fcp,1);
% Las frecuencias van a determinar las dimensiones
mensaje1=reshape(mensaje1,1,[]);
% Son mensajes producto de la secuencia de entrada junto con el código pseudoaleatorio
% Se gráfica la señal de entrada, el codigo pseudoaleatorio en el dominio del tiempo
cod_pn_ele=length(codi_pn);
% Se establece la cantidad de elementos con el comando length para el código pseudoaleatorio
% El comando linspace devuelve valores espaciados entre las variables
tpn = linspace(0,length(b_entrada)*t_bit-t_bit/fcp,cod_pn_ele);
% Periodo del código pseudoaleatorio
t_sentra = 0:t_bit/fcp:length(b_entrada)*t_bit-t_bit/fcp;
% Se establece el periodo para la señal de entrada
% Secuencia original de bits
figure(2);
stairs(t_sentra,snl_entrada,'linewidth',2)
% El comando linewidth se establece las características de las líneas o curvas
title('Secuencia de bits de la Señal de Entrada')
axis([0 length(b_entrada)*t_bit -1 1]);
ylabel('Amplitud');
xlabel('Tiempo');


% Código Pseudoaleatorio
```

```
figure(3);
stairs(tpn,codi_pn,'linewidth',2)
title('Código Pseudoaleatorio');
axis([0 length(b_entrada)*t_bit -1 1]);
ylabel('Amplitud');
xlabel('Tiempo');
% Señal ensanchada
figure(4);
stairs(tpn,snl_dsss,'linewidth',2)
% Se gráfica la señal ensanchada
title('Señal modulada');
axis([0 length(b_entrada)*t_bit -1 1]);
% Se establecen los límites de los ejes
ylabel('Amplitud');
xlabel('Tiempo(s)');
% Mensaje con el código Pseudoaleatorio
figure(5);
stairs(t_sentra,mensaje1,'linewidth',2)
title('Mensaje recibido con el código pseudoaleatorio')
axis([0 length(b_entrada)*t_bit -1 1]);
ylabel('Amplitud');
xlabel('Tiempo(s)');
% Graficas en el dominio de la frecuencia
fre = linspace(-Fs/2,Fs/2,1024);
figure(6);
% Se establece la frecuencia en funcion de la frecuencia de la señal
plot(fre,abs(fftshift(fft(snl_entrada,1024))),'linewidth',2);
```

```
% De entrada con 1024 valores entre las frecuencias inicial y final
title('Espectro de Frecuencia de la señal de entrada');
xlabel('Frecuencia (Hz)');
ylabel('Tiempo(s)')
% La fft se usa para mostrar características de la frecuencia
figure(7);
plot(fre,abs(fftshift(fft(codi_pn,1024))),'linewidth',2);
title('Espectro de Frecuencia del Código Pseudoaleatorio');
xlabel('Frecuencia (Hz)');
ylabel('Tiempo(s)')
figure(8);
plot(fre,abs(fftshift(fft(snl_dsss,1024))),'linewidth',2);
title('Espectro de Frecuencia de la señal de Espectro Ensanchado');
xlabel('Frecuencia (Hz)');
ylabel('Tiempo(s)')
figure(9);
subplot(311)
plot(fre,abs(fftshift(fft(snl_entrada,1024))),'linewidth',2);
title('Espectro de Frecuencia de la señal de Entrada')
xlabel('Frecuencia (Hz)');
ylabel('Tiempo(s)')
subplot(312)
plot(fre,abs(fftshift(fft(codi_pn,1024))),'linewidth',2);
title('Espectro de Frecuencia del Código Pseudoaleatorio');
xlabel('Frecuencia (Hz)');
ylabel('Tiempo(s)')
subplot(313)
```

```
plot(fre,abs(fftshift(fft(snl_dsss,1024))),'linewidth',2);

title('Espectro de Frecuencia de la señal de Espectro Ensanchado');

xlabel('Frecuencia (Hz)');

ylabel('Tiempo(s)')

% Densidad espectral para la señal ensanchada

z = abs(fft(xcorr(snl_dsss)));

minz=min(z);

% Se normaliza a la señal para facilitar el análisis

z1=z-minz;

maxz=max(z1);

z_normal=z1/maxz;

figure(10);

plot(z_normal);

title('Densidad Espectral de Potencia para la señal ensanchada por Secuencia Directa');

% Se establecen los ejes

xlabel('Frecuencia (Hz)');

ylabel('Potencia (dBm/Hz)');
```

ANEXO VI: Código fuente de la simulación de FHSS en Matlab

```
% Se inicializa eliminando todo rastro de otro script

clc

clear all;

% Se establecen los parámetros

N=20;

% Generación de los bits de entrada en total 20

T=120;

% Creación de T muestras por segundo

t_bit=N*T;

% La duración de la transmisión será 120 segundos * 20 que es la cantidad de bits =
2400 segundos

d_bit=T/20;

% Duración de cada bit

b_entrada = [1 1 1 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 1];

senal_entrada=[];

% Se establece una señal de entrada

senal_portadora=[];

% Se establece una señal portadora

t=0:2*pi/(T-1):2*pi;

% Se establece el periodo de muestreo para la señal coseno (podrá ser seno)

% Convirtiendo los bits de 1 a 0, secuencia original

for k = 1 :N

   if b_entrada(1,k)==0

      snl=-ones(1,T);

% Condición para que el bit de entrada sea cero

   else
```

```
        snl=ones(1,T);
% Condición para que el bit de entrada sea uno
    end
    p=cos(t);
% Se establece la señal senoidal para la señal portadora
    senal_entrada = [senal_entrada snl];
% Se establece la señal original a transmitir
    senal_portadora = [senal_portadora p];
% Se establece la señal de la portadora a transmitir
end
% Se grafica la señal de datos binarios
figure(1);
% Se grafican los datos de entrada como unos y ceros
stem(b_entrada,'linewidth',2);
title('Datos binarios de entrada');
% Se establecen los títulos en los ejes
ylabel('Amplitud');
xlabel('Tiempo');
% Se grafica la señal de entrada
figure(2);
plot(senal_entrada,'linewidth',2);
% Se grafican los datos de entrada como una secuencia de unos y ceros
axis([0 t_bit -1 1]);
% Se establecen los límites de los ejes
title('Secuencia Original de Bits');
% Se establecen los títulos en los ejes
ylabel('Amplitud');
```

```
xlabel('Tiempo(s)');

% Se realiza el proceso de modulación

bpskmod_senal=senal_entrada.*senal_portadora;

% Se multiplican los datos de las señales de entrada y la portadora para proceder a la
modulación BPSK

figure(3);

% Grafica de la señal modulada en BPSK

plot(bpskmod_senal,'linewidth',2);

% Se grafica la señal modulada

axis([0 t_bit -1 1]);

% Se establecen los límites de los ejes

title('Señal BPSK modulada');

% Se establecen los títulos en los ejes

ylabel('Amplitud');

xlabel('Tiempo(s)');

% Se establecen los tiempos de muestreo para las 6 portadoras

t1=0:2*pi/9:2*pi;

% Los periodos son distintos porque las portadoras son distintas

t2=0:2*pi/19:2*pi;

t3=0:2*pi/29:2*pi;

t4=0:2*pi/39:2*pi;

t5=0:2*pi/59:2*pi;

t6=0:2*pi/119:2*pi;

% Se establecen 6 frecuencias


% Son arreglos de matrices para establecer los saltos de frecuencias

portadora1=cos(t1);
```

```
portadora1=[portadora1 portadora1 portadora1 portadora1 portadora1 portadora1 portadora1 portadora1 portadora1 portadora1 portadora1 portadora1];

portadora2=cos(t2);

portadora2=[portadora2 portadora2 portadora2 portadora2 portadora2 portadora2];

portadora3=cos(t3);

portadora3=[portadora3 portadora3 portadora3 portadora3];

portadora4=cos(t4);

portadora4=[portadora4 portadora4 portadora4];

% Con este arreglo de matrices se genera un bucle lo que se genera el periodo de cada portadora

portadora5=cos(t5);

portadora5=[portadora5 portadora5];

portadora6=cos(t6);

senal_ensa=[];

% Se crea la señal ensanchada

% Se crean los saltos de frecuencia de manera aleatoria

for n=1:N

   p=randi([1 6],1,1);

% Se generan de manera aleatoria las frecuencias de 1 a 6 porque ese es el número de portadoras

   switch(p)

% Se ensancha a la señal con en salto de frecuencia de las seis portadoras

      case(1)

         senal_ensa=[senal_ensa portadora1];

      case(2)

         senal_ensa=[senal_ensa portadora2];

      case(3)

         senal_ensa=[senal_ensa portadora3];
```

```
        case(4)
            senal_ensa=[senal_ensa portadora4];
        case(5)
            senal_ensa=[senal_ensa portadora5];
        case(6)
            senal_ensa=[senal_ensa portadora6];
    end
end
figure(4);
plot(senal_ensa);
% Se grafica la señal de espectro ensanchado con el salto de seis frecuencias
axis([0 t_bit -1 1]);
% Se establecen los límites de los ejes
title('Señal portadora');
% Se establecen los títulos en los ejes
ylabel('Amplitud');
xlabel('Tiempo(s)');
senal_salto_fre=bpskmod_senal.*senal_ensa;
% Se establece la señal con salto de frecuencia como el producto de la señal modulada
con la señal ensanchada
figure(5);
% Se grafica la señal con salto de frecuencia
plot(senal_salto_fre);
axis([0 t_bit -1 1]);
% Se establecen los límites de los ejes
title('Señal con salto de Frecuencia');
% Se establecen los títulos en los ejes
```

```
ylabel('Amplitud');
xlabel('Tiempo(s)');
% Proceso de demodulación
bpsk_demo=senal_salto_fre./senal_ensa;
% En el receptor se recibe la señal FH modulada y se demodula
figure(6)
% Se grafica la señal con salto de frecuencia
plot(bpsk_demo);
axis([0 t_bit -1 1]);
% Se establecen los límites de los ejes
title('Señal Demodulada por BPSK mediante el Espectro ensanchado');
ylabel('Amplitud');
% Se establecen los títulos en los ejes
xlabel('Tiempo(s)');
original_BPSK_signal=bpsk_demo./senal_portadora;
% Se recupera la señal original de bits
figure(7)
% Se grafica la señal original
plot(original_BPSK_signal,'linewidth',2);
axis([0 t_bit -1 1]);
% Se establecen los límites de los ejes
title('Transmisión de la secuencia Original de Bits');
% Se establecen los títulos en los ejes
ylabel('Amplitud');
xlabel('Tiempo(s)');
Fs = 100;
% Frecuencia de la señal de entrada o señal moduladora
```

```
fre = linspace(-Fs/2,Fs/2,1024);

% Se establece la frecuencia en función de la frecuencia de la señal de entrada con 1024 valores entre las frecuencias inicial y final

% La fft se usa para mostrar características de la frecuencia

figure(8);

plot(fre,abs(fftshift(fft(senal_ensa,1024))),'linewidth',2);

title('Espectro de Frecuencia de la señal de Espectro Ensanchado');

% Señal codificada

xlabel('Frecuencia (Hz)');

ylabel('Tiempo(s)')

% Densidad espectral de potencia

z = abs(fft(xcorr(senal_ensa)));

% PSD (densidad espectral de potencia/power spectral density)

minz=min(z);

z1=z-minz;

maxz=max(z1);

z_normal=z1/maxz;

figure(9);

% Se establecen los ejes

plot(z_normal);

title('Densidad Espectral de Potencia para la señal ensanchada por Salto de Frecuencia');

xlabel('Frecuencia (Hz)');

ylabel('Potencia (dBm/Hz)');

% Para la señal ensanchada
```

ANEXO VII: Comandos empleados para la simulación de DSSS y FHSS en Matlab

Comandos	Función que cumplen
abs	Devuelve el valor absoluto de un arreglo y en caso de ser un elemento complejo se devuelve una magnitud compleja.
axis	Especifica los límites de los ejes actuales de los gráficos, puede representarse con un vector de ochos elementos donde los dos primeros representan al eje x y los dos siguientes al eje y.
clc	Borra todo el texto de la ventana de comandos, las entradas y las salidas obteniendo una ventana despejada.
clear all	Elimina todas las variables globales, las variables persistentes, los archivos y funciones del área de trabajo, también elimina la memoria caché.
else, elseif	Las instrucciones se ejecutan si la condición expresada en el comando *if* es falsa.
end	Comando clave para terminar las instrucciones que se ejecuten mediante los comando o sentencias *for*, *while*, *switch,* etc.
fft	Calcula la Transformada rápida de Fourier mediante un algoritmo.
fftshift	Permite desplazar un elemento de frecuencia cero al centro del espectro, reorganizando la transformada de Fourier.
figure	Se abre una ventana para desplegar una nueva figura.
for	Usado para crear un bucle para repetir una acción una cantidad determinada de veces
if	Ejecuta instrucciones establecidas si la condición es verdadera.
length	Devuelve la mayor longitud de la dimensión de una matriz, y en caso de que el arreglo resulte ser un vector la longitud es el número de elementos.
linewidth	Permite especificar las características de las líneas y marcadores en los gráficos, se puede seleccionar el estilo, los colores y el acho de las líneas.
linspace	Crea un vector con sus elementos separados linealmente.
max	Selecciona el mayor elemento de un arreglo.
min	Selecciona el menor elemento de un arreglo.
ones	Permite crear arreglos que como elementos contengan el número uno.

plot	Establece un gráfico de dos dimensiones, se deben establecer los ejes para posicionar correctamente los elementos.
rand	Crea números pseudoaleatorios y los distribuye de manera uniforme, pueden ser matrices o escalares.
randi	Crea enteros pseudoaleatorios distribuidos de manera uniforme [26]. Permite establecer un intervalo para limitar el rango de selección.
repmat	Devuelve un arreglo (una matriz) con copias de un elemento.
reshape	Cambia las dimensiones de un arreglo manteniendo los elementos.
round	Redondea los elementos a su entero más cercano.
stairs	Representa un gráfico escalonado de los elementos de un arreglo.
stem	Representa una secuencia de datos discretos, con una línea vertical a lo largo del eje x, indicando los valores de los datos con círculos al finalizar cada línea [27].
subplot	Permite segmentar una figura creando ejes en posiciones distintas, pueden ser cuadraturas en las que se especifiquen sus dimensiones, que indicaran cada subgráfico.
sum	Se suman todos los elementos de un arreglo.
switch, case	Evalúa una expresión y selecciona las instrucciones a ejecutar en el caso de elección. Se prueba cada caso hasta que la expresión sea verdadera.
title	Especifica el título del gráfico.
xcorr	Devuelve una correlación cruza o autocorrelación, de secuencias en tiempos diferenciados, comparando las similitudes.
xlabel	Establece al título del eje x del gráfico.
ylabel	Establece al título del eje y del gráfico.