

**UNIVERSIDAD POLITÉCNICA SALESIANA**  
**SEDE QUITO**

**CARRERA:**  
**INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:**  
**INGENIERO ELECTRÓNICO**

**TEMA:**  
**ANÁLISIS COMPARATIVO DE LOS ALGORITMOS RADIX-2 Y SPLIT  
RADIX SOBRE EL ESTÁNDAR DE TELEVISIÓN DIGITAL ISDB-T**

**AUTOR:**  
**CARLOS ENRIQUE ALARCÓN ARTEAGA**

**TUTORA:**  
**VERÓNICA EMMA SORIA MALDONADO**

**Quito, julio del 2019**

## **CESIÓN DE DERECHOS DE AUTOR**

Yo, Carlos Enrique Alarcón Arteaga con documento de identificación N° 1720628104, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación intitulado: “ANÁLISIS COMPARATIVO DE LOS ALGORITMOS RADIX-2 Y SPLIT RADIX SOBRE EL ESTÁNDAR DE TELEVISIÓN DIGITAL ISDB-T”, mismo que ha sido desarrollado para optar por el título de Ingeniero Electrónico en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de la Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago la entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



---

Carlos Enrique Alarcón Arteaga  
C.I.: 1720628104

Quito, julio del 2019

## **DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR**

Yo declaro que bajo mi dirección y asesoría fue desarrollado el proyecto técnico, “ANÁLISIS COMPARATIVO DE LOS ALGORITMOS RADIX-2 Y SPLIT RADIX SOBRE EL ESTÁNDAR DE TELEVISIÓN DIGITAL ISDB-T”, realizado por Carlos Enrique Alarcón Arteaga, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerado como trabajo final de titulación.

Quito, julio del 2019



---

Verónica Emma Soria Maldonado

C.I.: 1715185961

## **DEDICATORIA**

A Dios por todas las bendiciones que ha derramado sobre mí con la familia que tengo, por la fortaleza y la fe que otorga a mi espíritu para salir adelante en los momentos difíciles.

A mis padres Carlos y Eugenia, a mi hermana Paulina, por todo el amor y apoyo incondicional que me han brindado durante toda la vida. Por todos los valores y buenas costumbres que he aprendido. Gracias a su esfuerzo he tenido la posibilidad de superarme y convertirme en una mejor persona cada día.

## **AGRADECIMIENTO**

Agradezco a mis padres y hermana por el apoyo que me han brindado siempre, en especial durante este último periodo que por circunstancias de la vida resultó ser una etapa difícil.

Agradezco a mis tíos, Marcelo, Rodrigo y Elsa por todo el afecto, cariño y apoyo que siempre han entregado a mi familia, por estar siempre junto a nosotros.

Agradezco a todas las personas que han sido parte de mi vida durante este proceso. A mis amigos por todos los buenos y malos momentos que pasamos, a los profesores por el tiempo que dedicaron para compartir con nosotros sus experiencias y conocimientos.

Un agradecimiento especial a los Ingenieros Verónica Soria y Luis Oñate por su asesoría, disposición y apoyo en el desarrollo de este proyecto.

## ÍNDICE DE CONTENIDO

CESIÓN DE DERECHOS DE AUTOR.....	I
DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR.....	II
DEDICATORIA .....	III
AGRADECIMIENTO .....	IV
ÍNDICE DE CONTENIDO.....	V
ÍNDICE DE FIGURAS.....	VIII
ÍNDICE DE TABLAS .....	X
RESUMEN.....	XI
ABSTRACT.....	XII
INTRODUCCIÓN .....	XIII
<b>CAPÍTULO 1.....</b>	<b>1</b>
<b>ANTECEDENTES .....</b>	<b>1</b>
1.1. Objetivos .....	1
1.1.1. Objetivo general .....	1
1.1.2. Objetivos específicos .....	1
1.2. Problema de estudio .....	1
1.3. Justificación.....	2
1.4. Grupo objetivo .....	2
1.5. Planteamiento del problema.....	2
<b>CAPÍTULO 2.....</b>	<b>4</b>
<b>MARCO TEÓRICO .....</b>	<b>4</b>
2.1. Televisión Digital Terrestre (TDT).....	4
2.2. Estándar ISDB-T.....	5
2.2.1. Características ISDB-T .....	6
2.2.2. Estructura del estándar ISDB-T .....	6
2.2.3. Parámetros del estándar ISDB-T.....	8
2.2.4. Diferencia entre ISDB-T e ISDB-T Internacional .....	8
2.3. OFDM .....	9
2.3.1. Ortogonalidad.....	10
2.4. Transformada de Fourier.....	10
2.4.1. Transformada Discreta de Fourier (TDF) .....	11
2.4.2. Transformada Rápida de Fourier (FFT).....	12

2.4.3.	Transformada Inversa Rápida de Fourier (IFFT).....	13
2.5.	Algoritmos de la FFT.....	13
2.5.1.	Algoritmo radix-2 .....	13
2.5.2.	Radix-2 diezmado en el tiempo .....	14
2.5.3.	Radix-2 diezmado en frecuencia.....	15
2.5.4.	Algoritmo Radix-4 .....	16
2.5.5.	Algoritmo Split Radix.....	17
2.5.6.	Comparación entre algoritmos de la FFT .....	18
<b>CAPÍTULO 3.....</b>		<b>20</b>
<b>SIMULACIÓN DEL ESTÁNDAR ISDB-T CON LOS ALGORITMOS RADIX-</b>		
<b>2 Y SPLIT RADIX .....</b>		<b>20</b>
3.1.	Simulación en MATLAB (Simulink).....	20
3.1.1.	Imagen de entrada .....	22
3.1.2.	Codificador Reed Solomon (n,k) .....	22
3.1.3.	Convertidor de entero a bits .....	23
3.1.4.	Bloque para la modulación de portadora .....	24
3.1.5.	Bloque para entrelazado de tiempo y frecuencia .....	26
3.1.6.	Bloque del modulador OFDM .....	26
3.1.7.	Bloque para el canal .....	27
3.1.8.	Bloque del demodulador OFDM.....	28
3.1.9.	Bloque para el desentrelazado de tiempo y frecuencia .....	31
3.1.10.	Bloque para la demodulación de portadora.....	31
3.1.11.	Convertidor de bits a enteros .....	32
3.1.12.	Decodificador Reed Solomon .....	32
3.1.13.	Imagen de salida.....	33
3.2.	Código del bloque Split Radix .....	33
3.3.	Código del bloque Radix-2 .....	38
<b>CAPÍTULO 4.....</b>		<b>41</b>
<b>ANÁLISIS Y RESULTADOS.....</b>		<b>41</b>
4.1.	Datos obtenidos con el algoritmo Radix-2.....	41
4.2.	Datos obtenidos con el algoritmo Split Radix .....	41
4.3.	Gráficas de los resultados obtenidos .....	42
4.4.	Eficiencia.....	45
4.5.	Retardo .....	47

CONCLUSIONES .....	49
RECOMENDACIONES .....	51
REFERENCIAS .....	52

## ÍNDICE DE FIGURAS

Figura 2.1. Modelo de televisión digital .....	4
Figura 2.2. Diagrama de bloques para el estándar ISDB-T .....	6
Figura 2.3. Estructura en bloques del estándar ISDB-T para transmisión .....	7
Figura 2.4. Transmisor OFDM .....	10
Figura 2.5. Receptor OFDM .....	10
Figura 2.6. Operación en mariposa .....	13
Figura 2.7. Operación mariposa en Radix-2 DIT para 8 puntos .....	15
Figura 2.8. Operación mariposa en Radix-2 DIF para 8 puntos .....	16
Figura 2.9 Operación mariposa en Radix-4 .....	17
Figura 2.10. Operación mariposa en Split Radix .....	18
Figura 3.1. Diagrama de bloques del estándar ISDB-T en Simulink.....	21
Figura 3.2. Parámetros del bloque Signal From Workspace.....	22
Figura 3.3. Bloque de Simulink para el codificador RS .....	23
Figura 3.4. Parámetros del codificador RS .....	23
Figura 3.5. Bloque de Simulink para el convertidor de entero a bit .....	23
Figura 3.6. Parámetros del convertidor de entero a bits.....	24
Figura 3.7. Bloque de modulación de portadora .....	24
Figura 3.8. Bloque del subsistema Mapping.....	25
Figura 3.9. Convertidor de bits a entero.....	25
Figura 3.10. Modulador de banda base QAM.....	26
Figura 3.11. Subsistema del modulador OFDM con Split Radix .....	27
Figura 3.12. Subsistema del modulador OFDM con Radix-2.....	27
Figura 3.13. Subsistema para el canal de comunicación.....	28
Figura 3.14. Demodulador OFDM con Split Radix.....	29
Figura 3.15. Bloque Multiport Selector .....	30
Figura 3.16. Bloque Matrix Concatenate .....	30
Figura 3.17. Demodulador OFDM con Radix-2 .....	31
Figura 3.18. Subsistema del bloque Demodulación de portadora.....	31
Figura 3.19. Bloque decodificador Reed Solomon .....	32
Figura 3.20. Parámetros decodificador RS .....	32
Figura 3.21. Script de MATLAB para el algoritmo Split Radix en el demodulador. 34	
Figura 3.22. Script para el algoritmo Split Radix inverso.....	36

Figura 3.23. Diagrama de flujo para el algoritmo Split Radix.....	37
Figura 3.24. Script de MATLAB para el algoritmo Radix-2 en el demodulador .....	38
Figura 3.25. Diagrama de flujo para el algoritmo Radix-2 .....	39
Figura 3.26. Script de MATLAB para el algoritmo Radix-2 en el modulador.....	40
Figura 4.1. BER vs SNR aplicando los algoritmos de la FFT .....	42
Figura 4.2. Imagen recibida .....	43
Figura 4.3. Imagen recibida con el algoritmo Radix-2 .....	44
Figura 4.4. Imagen recibida con el algoritmo Split Radix .....	44
Figura 4.5. Gráfica de la eficiencia de transmisión aplicando los algoritmos Radix-2 y Split Radix.....	47

## ÍNDICE DE TABLAS

Tabla 2.1. Parámetros de transmisión en el estándar ISDB-T .....	8
Tabla 2.2. Comparación entre el estándar ISDB-T y ISDB-Tb .....	9
Tabla 2.3. Comparación de cálculos entre la TDF y la FFT .....	12
Tabla 2.4. Comparación entre algoritmos de la FFT .....	19
Tabla 3.1. Especificaciones del canal SUI de tipo C .....	28
Tabla 4.1. Resultados obtenidos con el algoritmo Radix-2 .....	41
Tabla 4.2. Resultados obtenidos con el algoritmo Split Radix .....	42
Tabla 4.3. Valores calculados de la eficiencia con el algoritmo Radix-2 .....	45
Tabla 4.4. Valores calculados de la eficiencia con el algoritmo Split Radix.....	46
Tabla 4.5. Tiempos de simulación para los algoritmos.....	48

## RESUMEN

El presente trabajo consiste en el análisis de televisión digital terrestre, específicamente sobre el estándar ISDB-T, para comprender su funcionamiento en base a un esquema de modulación OFDM mediante la utilización de dos algoritmos de la transformada rápida de Fourier conocidos como Radix-2 y una variante que une a los algoritmos Radix-2 y Radix-4 denominada Split Radix. La finalidad del proyecto es establecer una comparación entre parámetros básicos de la transmisión y recepción de información aplicando dichos algoritmos. Por tal motivo, se analizó la estructura del estándar de televisión digital ISDB-T para conocer sus características, configuraciones y de esta forma comprender el bloque en el cual se aplican los algoritmos Radix-2 y Split Radix. La simulación se fundamenta en una estructura de bloques desarrollada en el software Simulink de MATLAB que fue modificada tanto en la modulación como en la demodulación para la utilización y análisis de los algoritmos en cuestión; este modelo contiene aspectos básicos del estándar con modulación OFDM. Con los resultados obtenidos de la simulación tales como la tasa de bit errado, retardo y eficiencia, se indican las diferencias o similitudes entre los algoritmos y la forma en la que estos afectan al rendimiento del sistema.

## **ABSTRACT**

The present work consists in the analysis of digital terrestrial television, specifically on the ISDB-T standard, looking for understanding its operation based on an OFDM modulation scheme using two algorithms of the Fast Fourier Transform known as Radix-2 and a variant that joins the Radix-2 and Radix-4 algorithms called Split Radix, the purpose of the project is to establish a comparison between parameters of the transmission and reception of information applying these algorithms. The structure of the digital television standard ISDB-T was analysed to know its characteristics and configuration. So, it was possible to understand the block in which the Radix-2 and Split Radix FFT algorithms are applied. The simulation is based on a block structure developed in MATLAB's Simulink software that was modified in both modulation and demodulation for the use and analysis of the algorithms in question. This model contains basic aspects of the standard with OFDM modulation. With the obtained results from the simulation such as BER, delay and efficiency, the differences or similarities between the algorithms are indicated and the way in which these algorithms affect the performance of the system.

## INTRODUCCIÓN

El estándar de televisión digital terrestre ISDB-T fue desarrollado e implementado en Japón para posteriormente ser adoptado por otros países. Utiliza modulación de tipo OFDM, en cuyo esquema se aplica la Transformada discreta de Fourier para la modulación y demodulación mediante algoritmos que permiten mejorar la eficiencia ya que computan la TDF utilizando un menor número de cálculos. Dichos algoritmos corresponden a la Transformada Rápida de Fourier. Por tal motivo, la IFFT y FFT han sido implementadas en los moduladores y demoduladores OFDM, respectivamente. Entre estos algoritmos de la FFT se encuentran dos conocidos como Radix-2 y Split Radix.

En el capítulo uno se presenta la información correspondiente al proyecto, es decir, objetivos, problema de estudio, planteamiento del problema, justificación y el grupo de personas al cual beneficiará el estudio.

En el capítulo dos se encuentra la información referente al tema de estudio, que consiste en conceptos de televisión digital terrestre, el estándar ISDB-T y sus características, el tipo de modulación aplicada al estándar y el análisis de los algoritmos de la Transformada Rápida de Fourier Radix-2 y Split Radix que son utilizados en el proyecto.

En el capítulo tres se encuentra la simulación del estándar ISDB-T con modulación OFDM, en esta sección se detallan los bloques que la componen, los parámetros de configuración y los scripts utilizados para cada algoritmo, además la estructura necesaria para insertarlos en el esquema del estándar.

El capítulo cuatro consta de los resultados obtenidos al ejecutar la simulación con cada algoritmo, en estos valores se muestra la tasa de bit errado, eficiencia, retardo en la transmisión de información y los tiempos de ejecución del sistema. Finalmente se presentan las conclusiones y recomendaciones del trabajo realizado.

# CAPÍTULO 1

## ANTECEDENTES

### 1.1. Objetivos

#### 1.1.1. Objetivo general

Determinar el grado de mejora de una señal recibida en el estándar de televisión digital ISDB-T al utilizar el algoritmo Split Radix con respecto al algoritmo Radix-2 al obtener la señal digital de las portadoras analógicas.

#### 1.1.2. Objetivos específicos

- Analizar el sistema de televisión digital ISDB-T para que se determine en que bloque se aplican los algoritmos Radix-2 y Split Radix.
- Determinar los parámetros necesarios como tasa de bit errado, retardo y eficiencia de transmisión en el sistema de televisión digital ISDB-T para compararlos usando ambos algoritmos.
- Simular el sistema ISDB-T para que se conozca en qué grado mejora la recepción de señales ISDB-T con Split Radix con respecto al Radix-2.
- Comparar los parámetros simulados de los algoritmos Split Radix y Radix-2 para que se conozcan las ventajas del uno respecto al otro.

### 1.2. Problema de estudio

La TDT (Televisión Digital Terrestre), es un mecanismo gratuito para la transmisión de señales en televisión que ofrece una mejor experiencia al usuario por las mejoras presentes en la imagen, video y sonido que se reciben en los distintos dispositivos compatibles con el sistema. Este formato digital tiene recursos que optimizan la utilización del espectro radioeléctrico, debido a que se consigue obtener una mayor cantidad de frecuencias disponibles para que puedan ser utilizadas por otras tecnologías. Tanto en nuestro país como en una gran mayoría de Sudamérica se ha optado por la implementación del estándar ISDB-T.

El estándar ISDB-Tb es el sistema escogido para televisión digital en el Ecuador, razón que lo convierte en un tema de interés para conocer sus características y funcionamiento. El tipo de modulación aplicada en este sistema es OFDM; en la

transmisión o recepción de los símbolos de este tipo de modulación se aplica la DFT. Entre los algoritmos existentes de la FFT para reducir los cálculos y mejorar los tiempos de respuesta de los sistemas, se encuentran el Radix-2 y una variación del Radix-2 y Radix-4 conocida como Split Radix. Mediante este proyecto se pretende establecer una comparación entre estos algoritmos para verificar cuál le brinda al estándar de televisión una mejor optimización de parámetros tales como BER, tiempo, eficiencia en la transmisión y recepción de la información.

### **1.3. Justificación**

No existen muchas fuentes que establezcan una comparación específica entre Split Radix y Radix-2 para conocer el desempeño del estándar ISDB-T con la utilización de estos algoritmos, por lo que es necesario realizar un análisis sobre el tema, desarrollando las comparaciones necesarias mediante software para conocer en qué medida mejoran los parámetros de los datos transmitidos en el estándar ISDB-T. Por lo tanto, se evaluarán los parámetros que permitan indicar una comparación entre los algoritmos Split Radix y Radix-2 y de esta forma concluir cuál de los dos presenta un mejor rendimiento en el cálculo de la Transformada Rápida de Fourier (FFT) sobre el estándar ISDB-T.

Las personas beneficiadas con este proyecto son los estudiantes de la carrera de Ingeniería Electrónica y todo aquel que esté interesado en TDT.

### **1.4. Grupo objetivo**

El grupo de personas a las cuales el presente proyecto de investigación brindará aporte son los interesados en la investigación sobre los temas referentes a televisión digital, así como son los estudiantes de la Universidad Politécnica Salesiana.

### **1.5. Planteamiento del problema**

La tecnología TDT está prevaleciendo en el ámbito de televisión a nivel mundial y se espera un cambio total del sistema analógico a un entorno digital, a través del denominado apagón analógico. Todo esto debido a la mejora que existe en los parámetros tales como video, imagen y sonido, mismos que permiten al usuario tener una mejor experiencia e interacción con el sistema de televisión, por tal motivo, con el paso del tiempo se han desarrollado varios estándares para el uso de TDT. Este

proyecto se basa específicamente en la comparación del desempeño en el estándar ISDB-T mediante los algoritmos Split Radix y Radix-2 para determinar las mejoras que podrían existir con la aplicación de estos algoritmos.

Una de las principales razones del por qué se realiza este estudio es debido a que el estándar ISDB-T es utilizado en gran parte de Latinoamérica, por ello es necesario realizar mediciones para observar el comportamiento del estándar de acuerdo al tipo de algoritmo que se aplique. Con los resultados obtenidos se pueden determinar factores que permitan obtener un mejor desempeño del sistema y conocer si el cambio de parámetros en los bloques de transmisión y recepción del estándar tienen una diferencia significativa causada por la aplicación de cada uno de los algoritmos en estudio.

## CAPÍTULO 2

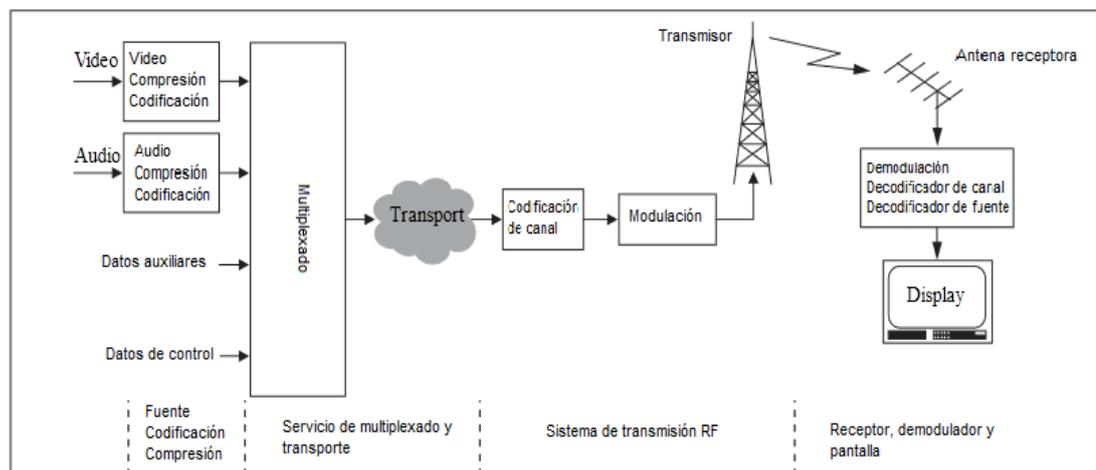
### MARCO TEÓRICO

En este capítulo se revisan conceptos básicos para comprender las características de la transmisión de la Televisión Digital Terrestre en el estándar ISDB-T. De acuerdo a la estructura que se presenta en el diagrama de bloques esencial del sistema, se generan los símbolos OFDM que contienen la información que es transmitida. Los algoritmos de la FFT pueden mejorar el rendimiento del sistema debido a que reducen la cantidad de cálculos necesarios para obtener la TDF. En este caso se analizan los algoritmos Radix-2, Split Radix y Radix-4, este último debido a que es aplicado en el algoritmo Split Radix para el cálculo de las muestras impares.

#### 2.1. Televisión Digital Terrestre (TDT)

Desde hace varios años se han elaborado estrategias e implementado estándares que permitan cambiar el sistema de televisión análogo a digital. El sistema de TDT brinda al usuario mejoras en la calidad de imagen y sonido que se recibe en los dispositivos, además le permite interactuar con el sistema. Dependiendo del estándar que se utilice, europeo o americano, los canales de televisión se transmiten con un ancho de banda entre 6 u 8 MHz. Mediante los procesos de digitalización se ha conseguido optimizar eficientemente el uso de frecuencias en el espectro radioeléctrico, por lo cual, en la televisión digital se puede incrementar el número de canales ofertados haciendo uso del mismo espectro que los sistemas analógicos. (Millan, 2010)

Figura 2.1. Modelo de televisión digital



Modelo en bloques para la transmisión y recepción de televisión digital. Fuente: (ITU-R, 2016)

El modelo que se describe en la figura 2.1, consta de cuatro bloques o divisiones. Inicialmente se encuentra la etapa de codificación y compresión, en la cual se establece la compresión de los datos y técnicas para la protección de errores en audio, video y datos. Después, la información pasa al servicio de multiplexación y transporte, en este bloque se procede a dividir la información digital en paquetes de información, también se consigue la identificación correspondiente al tipo de paquete (audio, video, datos) y la agrupación de dichos paquetes en un único flujo para que sean enviados al receptor través de algún mecanismo encargado de llevar la información. Los últimos dos bloques hacen referencia al sistema de transmisión de radio frecuencia y la recepción de la señal en los dispositivos, es decir, la señal es codificada y modulada en el transmisor para luego ser difundida y recibida por los dispositivos, en donde se realizan los procesos de decodificación y demodulación. (ITU-R, 2016)

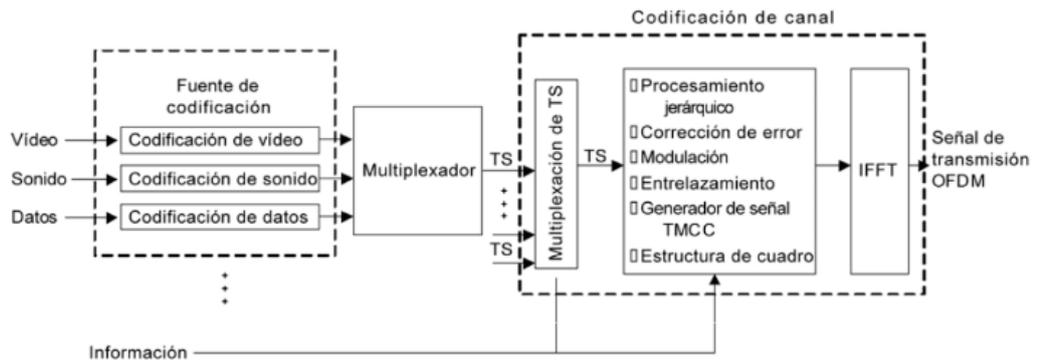
La transmisión de señales en el sistema de TDT es similar a la utilizada en el sistema análogo, es decir, la señal es difundida desde un punto para que pueda ser recibida por todos los usuarios que posean un televisor con el receptor requerido de acuerdo al estándar aplicado en cada país. En la mayor parte de Sudamérica los gobiernos optaron por el estándar brasilero ISDB-T internacional o ISDBT-Tb, que se diferencia del estándar japonés ISDB-T en la compresión que se maneja en audio y video.

## **2.2. Estándar ISDB-T**

El estándar ISDB-T fue desarrollado y opera desde el año 2003 en Japón, mediante el cual se han logrado servicios tales como movilidad, portabilidad e interactividad con el sistema. A mediados del año 2006, el estándar fue adoptado por Brasil, en donde se establecieron modificaciones para la compresión de audio y video en las transmisiones de SDTV y HDTV. El estándar brasilero es conocido como ISDB-T Internacional o SBTVD, mismo que ha sido adoptado por Ecuador y la mayoría de los países Sudamericanos. (DiBEG, Reporte Técnico de ARIB, 2007)

En la figura 2.2 se visualiza el esquema utilizado por ISDB-T, los TS (flujo de transporte) que contienen los distintos programas o servicios otorgados por el estándar, estos son remultiplexados para obtener un único flujo de información.

Figura 2.2. Diagrama de bloques para el estándar ISDB-T



Esquema de bloques del sistema ISDB-T. Fuente: (Sotelo, Durán, & Joskowicz, 2011)

### 2.2.1. Características ISDB-T

El estándar para televisión digital terrestre ISDB-T tuvo que cumplir con varios requerimientos para que sea implementado en las transmisiones de televisión digital en Japón, por lo cual, las características que se desarrollaron para ISDB-T son las siguientes:

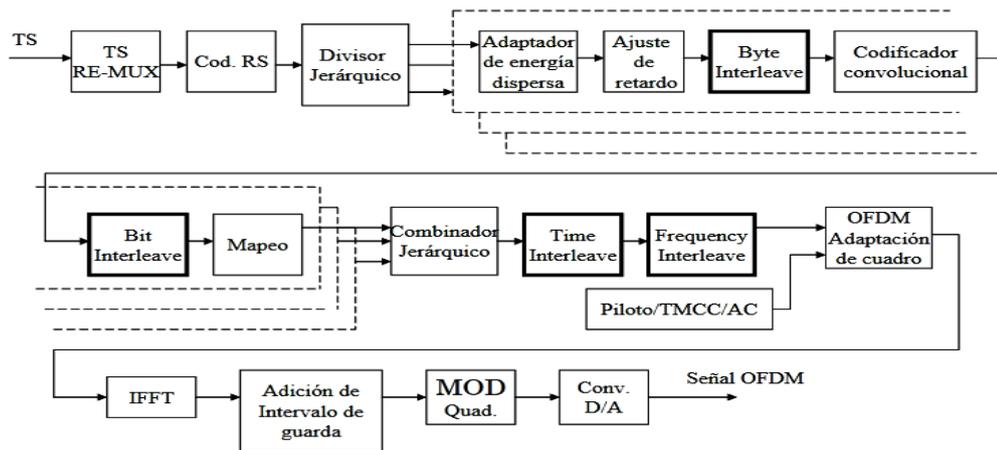
- La transmisión digital soporta sistemas HDTV, SDTV
- Los canales tienen un ancho de banda de 6MHz
- La señal es transmitida para diferentes tipos de dispositivos.
- Modulación OFDM
- El sistema posee una buena recepción de la señal, aunque exista un entorno con condiciones que puedan generar algún tipo de interferencia
- Es posible transmitir varios servicios utilizando el ancho de banda correspondiente al canal.

En el estándar ISDB-T es posible realizar distintos tipos de transmisión, mismos que pueden ser HDTV, HDTV+SDTV o varios canales SDTV. (DiBEG, Reporte Técnico de ARIB, 2007)

### 2.2.2. Estructura del estándar ISDB-T

El sistema ISDB-T esta constituido por la estructura que se presenta en la figura 2.3. Mediante la que se cumplen las distintas etapas para la generación de los símbolos OFDM para la transmisión de información.

Figura 2.3. Estructura en bloques del estándar ISDB-T para transmisión



Modelo aplicado en la transmisión de información en el estándar ISDB-T. Fuente: (DiBEG, 2007)

El flujo de transporte o TS (Transport Stream) está conformado por la sucesión de paquetes de 188 bytes cada uno, este flujo ingresa continuamente al re-multiplexador para generar un flujo binario conocido como BTS (Flujo de Transporte Broadcasting) el cual contiene toda la información que se va a transportar. Además, se añaden 16 bytes al final de cada paquete del flujo TS para formar un nuevo paquete denominado TSP que tiene 204 bytes de longitud. (Pisciotta N. O., 2014)

El codificador Red Solomon permite detectar y corregir errores en los bytes de los TSP generados en el flujo BTS. Otro factor importante es que se deben evitar patrones repetitivos de ceros y unos, asegurando la dispersión de energía del espectro, ya que, al existir irregularidades en el flujo de datos, la energía se concentraría en ciertos puntos del espectro, lo que origina la aparición de espacios vacíos y sin uso en otros puntos de este. (Pisciotta N. , 2010)

En el sistema ISDB-T la información puede ser organizada en tres capas jerárquicas que han sido denominada como A, B y C. A través de dichas capas el canal puede ser dividido en 13 segmentos. A cada capa le puede ser asignado un servicio de los que ofrece el sistema, por ejemplo, el servicio one-seg que utiliza un solo segmento se aplica en la capa A, los servicios de HDTV y SDTV pueden ser distribuidos en los 12 segmentos restantes. Es decir, las capas pueden estar conformadas por uno o más segmentos de OFDM dependiendo la cantidad de recursos que requiera el servicio a implementar.

El entrelazado de bytes es empleado para distribuir los errores de tipo ráfaga en el tiempo, mismos que pueden generarse en canales susceptibles a ellos. Las secuencias de bytes son entrelazadas antes de transmitirse (Pisciotta N. , 2010). Los tipos de entrelazado aplicados son de byte, bit, tiempo y frecuencia. La información es procesada, codificada y modulada para que puedan ser generados los símbolos OFDM y de esta forma pueda ser transmitida a los receptores.

### 2.2.3. Parámetros del estándar ISDB-T

El estándar ISDB-T permite la transmisión de información mediante tres modos de configuración diferentes, en la tabla 3.1 se muestran los parámetros configurables para cada uno.

Tabla 2.1. Parámetros de transmisión en el estándar ISDB-T

<b>Modo</b>	<b>Modo 1</b>	<b>Modo 2</b>	<b>Modo 3</b>
Número de segmentos OFDM	13		
Ancho de banda	5,575 MHz	5,573 MHz	5,572 MHz
Separación entre frecuencias portadoras	3,968 KHz	1,984 KHz	0,992 KHz
Esquema de modulación de las portadoras	QPSK, 16 QAM, 64 QAM, DQPSK		
Símbolos por cuadro	204		
Inner code	Código convolucional (1/2, 2/3, 3/4, 5/6, 7/8)		
Outer code	RS(204,188)		

Parámetros para los 3 modos del estándar ISDB-T. Fuente: (Sotelo, Durán, & Joskowicz, 2011)

### 2.2.4. Diferencia entre ISDB-T e ISDB-T Internacional

El estándar para televisión digital brasilero utiliza la misma arquitectura que el sistema japonés para la transmisión y recepción de señales, sin embargo, existen diferencias en las tecnologías utilizadas para la compresión de audio y video, que se resumen en la tabla 2.2. El middleware Ginga que se implementa en el estándar ISDB-T Internacional, hace referencia a un conjunto de funciones y métodos que permiten desarrollar aplicaciones para que el usuario tenga interacción con el sistema de TDT.

Tabla 2.2. Comparación entre el estándar ISDB-T y ISDB-Tb

<b>TECNOLOGÍAS</b>	<b>ISDB-T</b>	<b>ISDB-Tb</b>
<b>Aplicativos</b>	Interactivo	Interactivo
<b>Middleware</b>	ARIB	GINGA
<b>Compresión audio</b>	MPEG-2 AAC	MPEG-4 AAC
<b>Compresión video</b>	MPEG-2	MPEG-4
<b>Transporte</b>	MPEG-2	MPEG-2
<b>Modulación</b>	BST-OFDM	BST-OFDM
<b>Velocidad de transmisión</b>	15 fps	30 fps

Características del estándar ISDB-T japonés y brasilero. Fuente: (Alcocer & Arteaga, 2013)

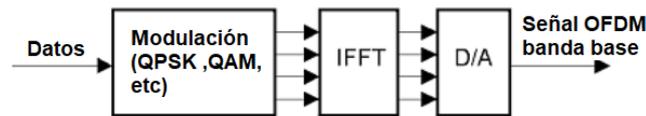
### 2.3. OFDM

La modulación OFDM (Orthogonal Frequency Division Multiplexing) es una técnica usualmente utilizada para transmitir inalámbricamente señales de banda ancha. Que ha sido implementada con diversos estándares para las transmisiones de televisión digital tales como DVB-T (Digital Video Broadcasting Terrestrial) e ISDB-T (Integrated Service of Digital Broadcasting Terrestrial). (Vargas, Lopez, & da Rocha, 2007)

Esta modulación permite enviar y repartir la información a través de varias subportadoras que se encuentran traslapadas bajo parámetros de ortogonalidad, lo que permite a un sistema ser robusto ante interferencias, desvanecimientos propios del trayecto en el que viaje la información, tal como el caso de la interferencia por propagación multitrayecto.

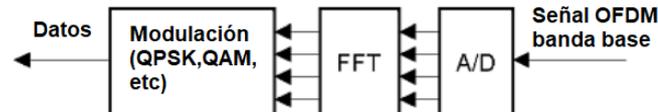
Una ventaja de OFDM es que se puede aplicar de una manera eficiente utilizando la IFFT (Transformada Inversa Rápida de Fourier) en el transmisor y la FFT (Transformada Rápida de Fourier) en el receptor (Rivas, 2012), como se observa en los diagramas de bloques de las figuras 2.4 y 2.5 respectivamente.

Figura 2.4. Transmisor OFDM



Estructura en bloques de un transmisor OFDM. Fuente: (Singh & Alphones, 2004)

Figura 2.5. Receptor OFDM



Estructura en bloques de un receptor OFDM. Fuente: (Singh & Alphones, 2004)

En el caso ISDB-T, el canal es dividido en trece segmentos y una parte adicional que no se atribuye a ningún segmento conocida como banda de guarda, por este motivo la modulación que se emplea es denominada BST-OFDM (Band Segmented Transmission - Orthogonal Frequency Division Multiplexing), cada segmento posee un ancho de banda correspondiente a 6/14 MHz o aproximadamente 428,57 KHz. La banda de guarda es aplicada en cada extremo del canal y no es necesario que se la distribuya simétricamente, dicha banda de guarda se adapta a las condiciones para reducir la probabilidad que exista interferencia con otros canales. (Sotelo, Durán, & Joskowicz, 2011)

### 2.3.1. Ortogonalidad

La ortogonalidad es una propiedad que posibilita transmitir múltiples señales de información sobre un mismo canal sin que se produzca interferencia, cuando no existe una adecuada ortogonalidad se degradan las señales y por tal motivo existen fallos en la comunicación. En OFDM se consigue ortogonalidad en el dominio de la frecuencia con las señales de información que son divididas entre distintas subportadoras. (Mejía, Garcia, & Ramos, 2014)

### 2.4. Transformada de Fourier

La transformada de Fourier ha sido un instrumento muy valioso en el desarrollo de sistemas para el procesamiento digital de imágenes, debido a que con su aplicación las señales que se encuentran en el dominio del tiempo pueden ser representadas en el

dominio de la frecuencia. Facilitando el análisis de las señales y los efectos que causan en los sistemas digitales. (Cassagnes, 2016)

#### 2.4.1. Transformada Discreta de Fourier (TDF)

La TDF es una variación de la Transformada de Fourier aplicada a secuencias que poseen una longitud finita, donde es posible observar el espectro para frecuencias específicas. Es de mucha importancia en el procesamiento digital de señales, se la utiliza en varias aplicaciones tales como filtrado y análisis espectral, además es implementada en modulaciones y demodulaciones de OFDM (Albertí, 2006). Para el cálculo de la TDF se requiere obtener una secuencia cuya longitud sea N, de acuerdo a la ecuación 2.1. (Lezana Illesca & Muratt Rodriguez, 2005)

$$F(k) = \sum_{n=0}^{N-1} f(n)W_N^{nk} \quad \text{Ec. (2.1)}$$

En la ecuación 2.2 se representa el factor conocido como constante Twiddle:

$$W_N = e^{-j\frac{2\pi}{N}} \quad \text{Ec. (2.2)}$$

Donde,

$$e^{-j\frac{2\pi}{N}} = \cos\left(\frac{2\pi}{N}\right) - j\sin\left(\frac{2\pi}{N}\right)$$

Por lo tanto,

$$W_N^n = \cos\left(\frac{2\pi n}{N}\right) - j\sin\left(\frac{2\pi n}{N}\right) = e^{-j\frac{2\pi n}{N}}$$

Entonces la ecuación 2.1 para el cálculo de un punto de la TDF se puede representar como:

$$F(k) = f(0)W^0 + f(1)W^k + f(2)W^{2k} \dots + f(N-1)W^{k(N-1)} \quad \text{Ec. (2.3)}$$

Se debe tomar en cuenta que el número máximo de muestra que se pueden obtener para  $F(k)$  es  $N$ . (Albertí, 2006)

La Transformada Discreta de Fourier Inversa está dada por la ecuación 2.4 que se muestra a continuación:

$$f(n) = \frac{1}{N} \sum_{k=0}^{N-1} F(k) e^{j\frac{2\pi}{N}nk} \quad \text{Ec. (2.4)}$$

#### 2.4.2. Transformada Rápida de Fourier (FFT)

La transformada rápida de Fourier es aplicada para calcular de una manera más eficiente a la transformada discreta de Fourier, debido a que los algoritmos que se utilizan en la FFT reducen los cálculos utilizados y el tiempo de ejecución de estos. En la tabla 2.3 se muestra una comparación entre la cantidad de cálculos requeridos con la TDF y la FFT para un máximo de 1024 puntos.

Tabla 2.3. Comparación de cálculos entre la TDF y la FFT

<b>N</b>	<b>DFT <math>N^2</math></b>	<b>FFT <math>\frac{N}{2}(\log_2 N)</math></b>
4	8	4
8	64	12
16	256	32
32	1024	80
64	4096	192
128	16384	448
256	65536	1024
512	262144	2304
1024	1048576	5120

Tabla comparativa de la cantidad de cálculos utilizados tanto en la TDF como en la FFT.

Elaborado por: Carlos Alarcón

### 2.4.3. Transformada Inversa Rápida de Fourier (IFFT)

La Transformada Inversa Discreta de Fourier, es la encargada de convertir una secuencia en el dominio de la frecuencia a una secuencia en el dominio del tiempo. Comparando la ecuación 2.4 con la TDF, se puede establecer que para el cálculo de la IFFT se debe añadir el factor  $1/N$  y cambiar el signo del exponente en el factor  $W^{-nk}$ . (Chassaing, 2002)

### 2.5. Algoritmos de la FFT

Los algoritmos que se implementan para el cálculo de la FFT utilizan las propiedades de simetría y periodicidad de  $W_N$  disminuyendo la cantidad de operaciones utilizadas en la DFT. Estos algoritmos se encuentran clasificados de acuerdo a los cálculos que se realicen ya sean en tiempo o frecuencia. Además, es importante que la longitud de la secuencia sea una potencia de 2. (Albertí, 2006)

Con la aplicación de los algoritmos de la FFT como Radix-2 y Split Radix en el estándar ISDB-T se realiza el cálculo de la IFFT en la modulación OFDM y FFT en la demodulación para conocer el que genera un mejor desempeño en los distintos parámetros medidos en la transmisión y recepción de señales.

#### 2.5.1. Algoritmo radix-2

El algoritmo Radix-2 se basa en que los elementos  $N$  de la señal sean potencia de 2. En este método se aplica la operación conocida como mariposa, que básicamente es un bloque de cálculos llevados a cabo entre un par de números reales o complejos. Dependiendo del método que se aplique en los algoritmos, los cálculos pueden ser efectuados para tiempo y frecuencia. En cada operación de este tipo se requieren cálculos de multiplicaciones, sumas y restas. (Guoan & Yonghong, 2004)

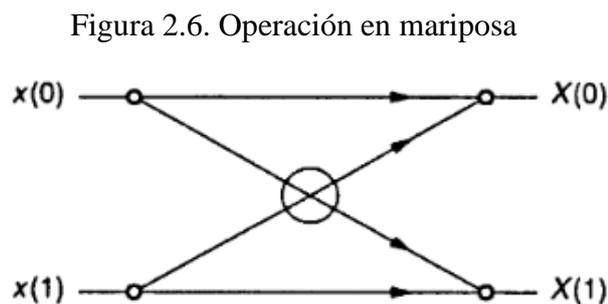


Diagrama de flujo para una operación mariposa básica. Fuente: (Chassaing, 2002)

En la figura 2.6 se muestra el diagrama de flujo correspondiente a una operación de mariposa básica, en la cual los resultados obtenidos aplicando la secuencia son los siguientes:

$$X(0) = x(0) + x(1)$$

$$X(1) = x(0) - x(1)$$

### 2.5.2. Radix-2 diezmado en el tiempo

En este método del algoritmo Radix-2, se reordenan las muestras en el tiempo. Mediante este proceso la señal de entrada es dividida en secuencias más pequeñas. Es decir, se consigue tener la señal en dos grupos, uno con la sumatoria de los índices pares y otro con la sumatoria de los índices impares. Por lo tanto, se descompone la transformada en dos sumatorios con longitud de  $N/2$  puntos, como se observa en la ecuación 2.6. Las ecuaciones se tomaron de la referencia (Guoan & Yonghong, 2004).

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(n) W_N^{nk} + \sum_{n=0}^{\frac{N}{2}-1} x(n) W_N^{nk} \quad \text{Ec. (2.5)}$$

Realizando las sustituciones de  $n=2n$  para un índice par y  $n=2n+1$  para un índice impar, en la ecuación 2.5 se obtiene:

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n) W_N^{2nk} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) W_N^{(2n+1)k}$$

Factorizando la ecuación anterior, resulta:

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n) W_N^{2nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) W_N^{2nk}$$

Reemplazando  $W_N^2 = W_{\frac{N}{2}}$ , la ecuación queda de la forma :

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n) W_N^{nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) W_N^{nk} \quad \text{Ec. (2.6)}$$

Donde k puede tomar valores de 0,1,2, .... hasta  $\frac{N}{2} - 1$ .

Por ejemplo, para el caso de una secuencia con 8 puntos, los grupos son organizados en índices pares [0 2 4 6] e impares [1 3 5 7], una vez agrupados se aplican las operaciones de acuerdo al diagrama de flujo mostrado en la figura 2.7.

Figura 2.7. Operación mariposa en Radix-2 DIT para 8 puntos

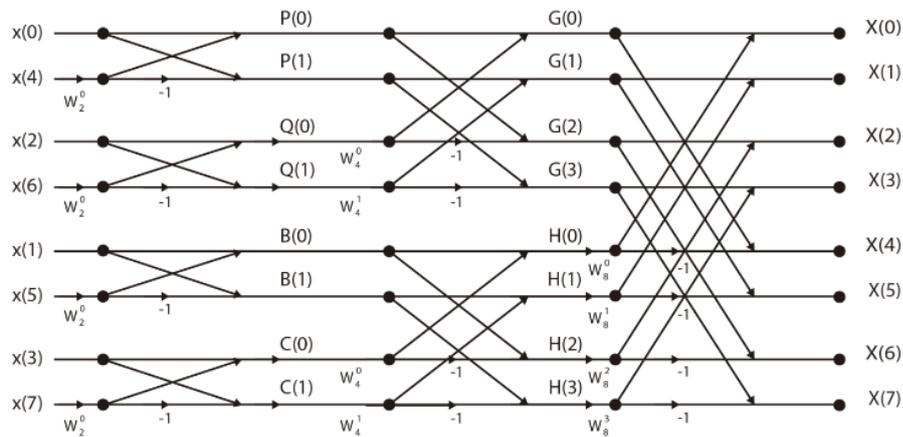


Diagrama de flujo para la operación mariposa en el cálculo de la FFT aplicando el algoritmo Radix-2 con diezmado en el tiempo con  $N=8$ . Fuente: (Cruz, 2013)

### 2.5.3. Radix-2 diezmado en frecuencia

El método de Radix-2 para la descomposición en frecuencia, es similar al diezmado en el tiempo, pero en esta aplicación se divide la secuencia de salida. Se separan los índices con los valores discretos de frecuencia pares y los índices con los valores discretos de frecuencia impares. (Cruz, 2013)

Las ecuaciones utilizadas en el algoritmo Radix-2 con diezmado en frecuencia se pueden observar en las ecuaciones 2.7 y 2.8 para las muestras pares e impares respectivamente.

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} \left[ x(n) + x\left(n + \frac{N}{2}\right) \right] W_N^{nk} \quad \text{Ec. (2.7)}$$

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} \left[ x(n) + x\left(n + \frac{N}{2}\right) \right] W_N^n W_{\frac{N}{2}}^{nk} \quad \text{Ec. (2.8)}$$

Para  $k = 0, 1, \dots, \frac{N}{2} - 1$ .

Figura 2.8. Operación mariposa en Radix-2 DIF para 8 puntos

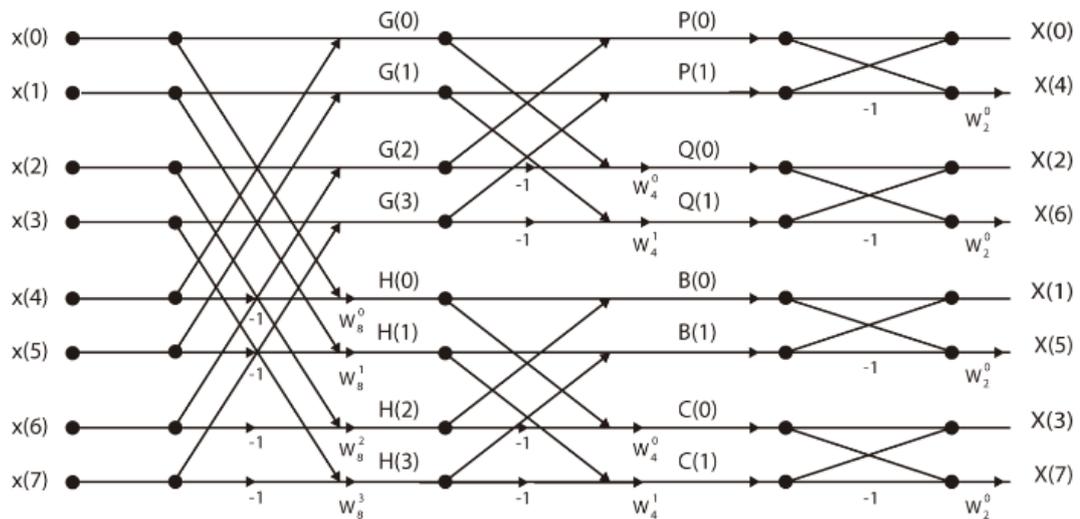


Diagrama de flujo para la operación mariposa en el cálculo de la FFT aplicando el algoritmo Radix-2 con diezmado en frecuencia con  $N=8$ . Fuente: (Cruz, 2013)

#### 2.5.4. Algoritmo Radix-4

El algoritmo Radix-4 puede aplicarse cuando los datos en la secuencia de entrada son múltiplos de 4. Para el diezmado en tiempo se divide a la secuencia de entrada en los cuatro grupos que se indican a continuación:  $x(4n), x(4n+1), x(4n+2), x(4n+3)$ , con  $n = 0, 1, \dots$  hasta  $\frac{N}{4} - 1$ . En el diezmado de frecuencia se divide la secuencia de salida en grupos, para reducir el número de cálculos empleados y obtener una mayor eficiencia que en el algoritmo Radix-2. (Proakis & Maniatis, Digital Signal Processing Principles, Algorithms, and Applications, 1996)

En el algoritmo Radix-4 son recopiladas dos etapas del algoritmo Radix-2 en una sola. Por ejemplo, para calcular una FFT de 16 puntos mediante Radix-2 se necesitan cuatro etapas ( $\log_2 16 = 4$ ), sin embargo, mediante Radix-4 son necesarias solo dos etapas ( $\log_4 16 = 2$ ). En la figura 2.9 se visualiza la operación mariposa que se aplica en el algoritmo Radix-4. (Cheng, 2000)

Figura 2.9 Operación mariposa en Radix-4

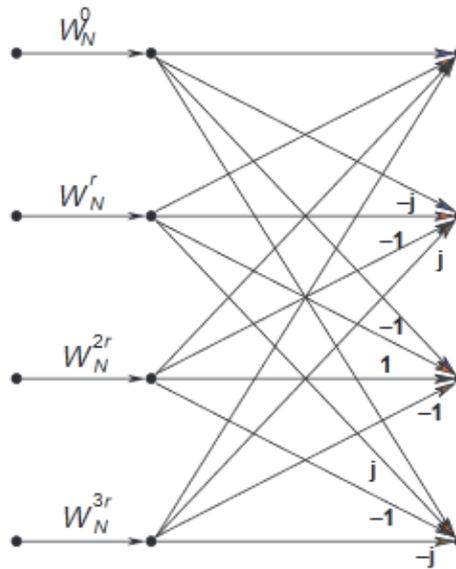


Diagrama de flujo de la operación mariposa para el algoritmo Radix-4. Fuente: (Cheng, 2000)

### 2.5.5. Algoritmo Split Radix

El algoritmo Split Radix fue desarrollado inicialmente por Duhamel y Hollmann en el año de 1984. En 1986 Duhamel amplió e implementó el algoritmo para datos simétricos, complejos y reales. Este algoritmo es aplicado en el cálculo de la TDF y es una variante de otros algoritmos empleados en la FFT. (George & Chu, 2000)

En este algoritmo se combinan los algoritmos Radix-2 y Radix-4, por tal motivo se convierte en un algoritmo muy eficaz para realizar los cálculos de la TDF. Mediante los cuales, para las muestras con índice par se implementa el algoritmo Radix-2 con la ecuación 2.9. El algoritmo Radix-4 es utilizado con las muestras de índice impar de acuerdo a las ecuaciones 2.10 y 2.11. (Watanabe, 2009)

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} \left[ x(n) + x\left(n + \frac{N}{2}\right) \right] W_N^{nk} \quad \text{Ec. (2.9)}$$

$$X(4k + 1) = \sum_{n=0}^{\frac{N}{4}-1} \left\{ \left[ x(n) - x\left(n + \frac{N}{2}\right) \right] - j \left[ x\left(n + \frac{N}{4}\right) - x\left(n + \frac{3N}{4}\right) \right] \right\} W_N^n W_N^{\frac{nk}{4}} \quad \text{Ec. (2.10)}$$

$$X(4k + 3) = \sum_{n=0}^{\frac{N}{4}-1} \left\{ \left[ x(n) - x\left(n + \frac{N}{2}\right) \right] + j \left[ x\left(n + \frac{N}{4}\right) - x\left(n + \frac{3N}{4}\right) \right] \right\} W_N^{3n} W_N^{\frac{nk}{4}} \quad \text{Ec. (2.11)}$$

Para  $k = 0, 1, \dots, \frac{N}{4} - 1$ .

En la figura 2.10 se presenta el diagrama de operación mariposa utilizada en el algoritmo Split Radix.

Figura 2.10. Operación mariposa en Split Radix

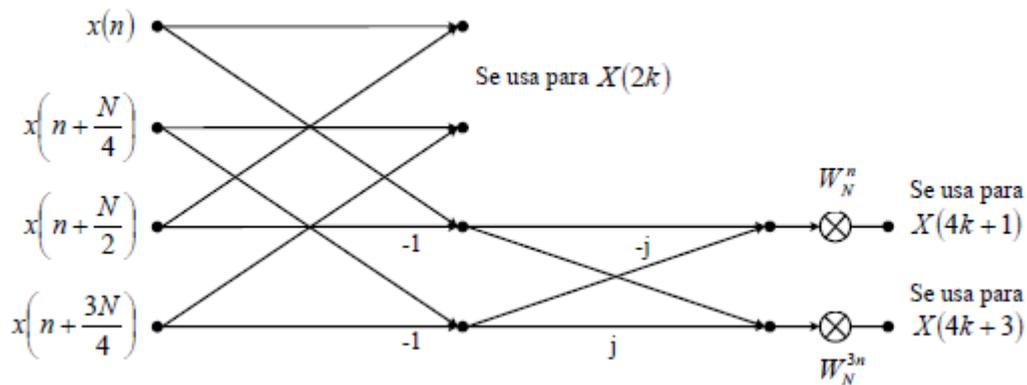


Diagrama de flujo de la operación mariposa para el algoritmo Split Radix. Fuente: (Watanabe, 2009)

### 2.5.6. Comparación entre algoritmos de la FFT

En la tabla 2.4 se muestra la comparación entre el número de operaciones que se ejecutan con los algoritmos Radix-2, Radix-4 y Split Radix para el cálculo de la FFT. Donde se puede apreciar que teóricamente el algoritmo Split Radix es el que posee una mayor eficiencia ya que utiliza el menor número de cálculos.

Tabla 2.4. Comparación entre algoritmos de la FFT

N	Multiplicaciones			Sumas		
	Radix-2	Radix-4	Split Radix	Radix-2	Radix-4	Split Radix
16	24	20	20	152	148	148
32	88		68	408		388
64	264	208	196	1032	976	964
128	712		516	2504		2308
256	1800	1392	1284	5896	5488	5380
512	4360		3076	13566		12292
1024	10248	7856	7172	30728	28336	27652

Comparación de cálculos entre algoritmos de la FFT. Fuente: (Proakis & Maniatis, 1996)

### **CAPÍTULO 3**

## **SIMULACIÓN DEL ESTÁNDAR ISDB-T CON LOS ALGORITMOS RADIX-2 Y SPLIT RADIX**

En el presente capítulo se detallan los bloques que componen el esquema del estándar ISDB-T aplicado en Simulink y los scripts correspondientes a cada algoritmo utilizados en las etapas de modulación y demodulación OFDM.

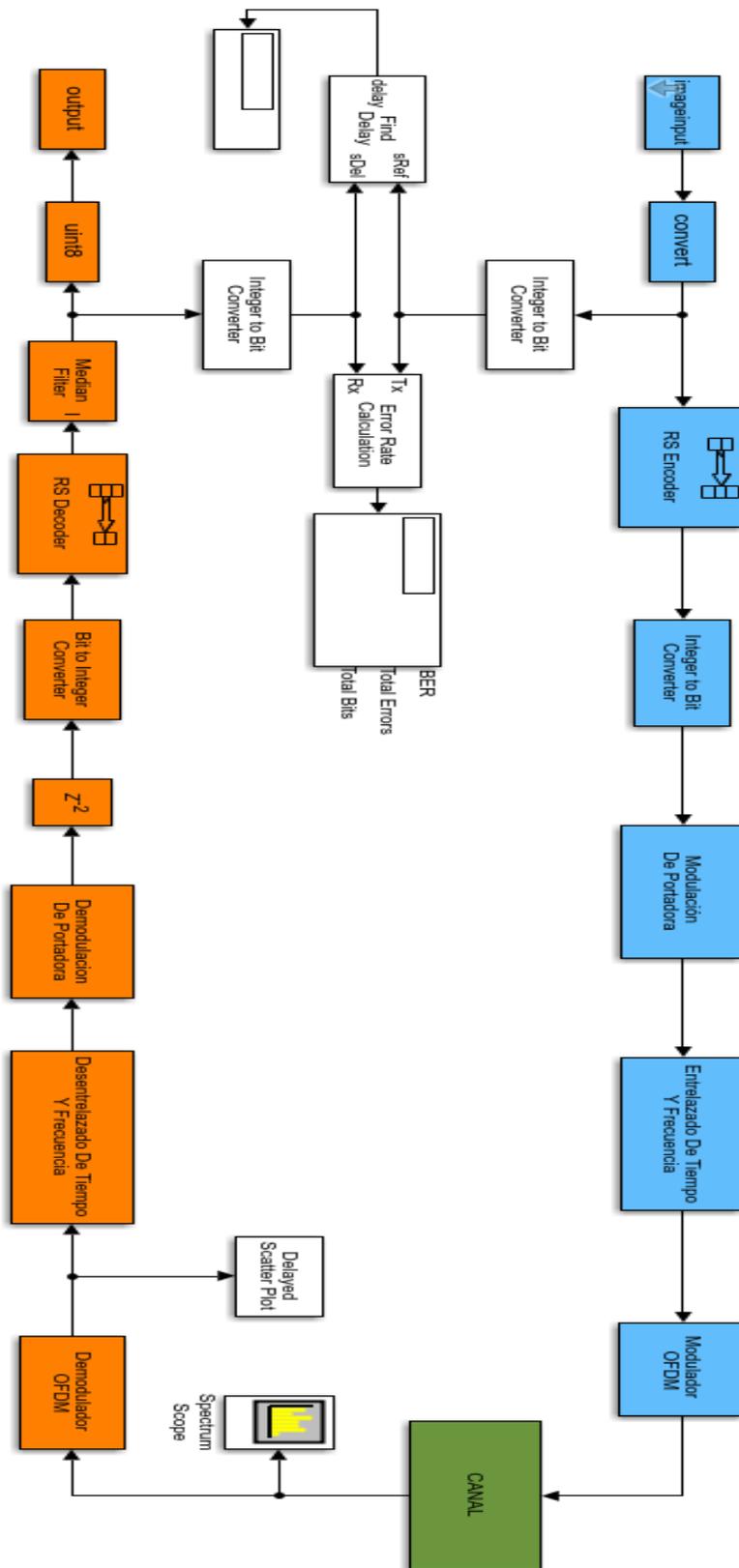
#### **3.1. Simulación en MATLAB (Simulink)**

La simulación del estándar ISDB-T aplicada en este proyecto se fundamenta en el proyecto de grado realizado por estudiantes de la Universidad Politécnica Salesiana, publicado con el título “Análisis comparativo entre DVB-T e ISDB-T utilizando modulación CC-OFDM” (Gomez & Lapo, 2019). En el cual, se implementa el diagrama de bloques en Simulink que se muestra en la figura 3.1 bajo parámetros de modulación OFDM. En este esquema se efectuaron modificaciones en el modulador y demodulador OFDM, para aplicar los algoritmos Radix-2 y Split Radix correspondientes a la FFT, este esquema funciona en el modo 1 del estándar ISDB-T.

La simulación consta de bloques que representan la transmisión(celeste), canal(verde) y recepción(naranja) de la información. Inicialmente, se envía una imagen, misma que circula a través de los distintos bloques del transmisor, donde es codificada y modulada para ser transmitida. La imagen atraviesa el canal que contiene parámetros para simular un ambiente urbano. Luego llega al receptor donde es demodulada, decodificada para que pueda ser visualizada.

El objetivo de este proyecto es la comparación que existe entre los algoritmos de la transformada rápida de Fourier, Radix-2 y Split Radix. Como se mencionó al inicio del capítulo, estos algoritmos son aplicados en el modulador y demodulador OFDM; donde se realizan los cálculos de la FFT. En las siguientes secciones se indican los parámetros configurados de los bloques que conforman la simulación.

Figura 3.1. Diagrama de bloques del estándar ISDB-T en Simulink

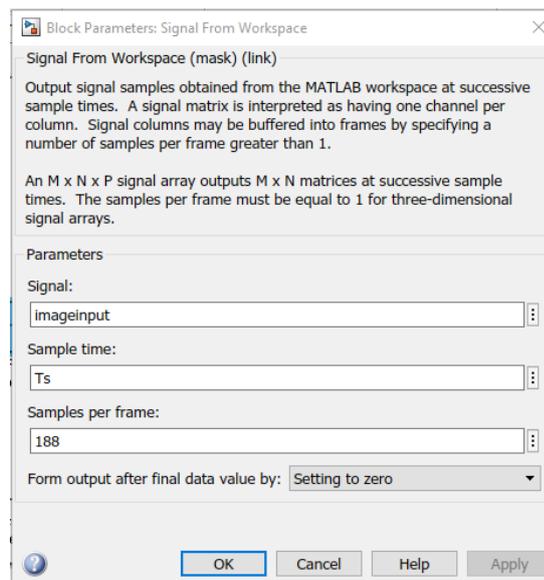


Bloques de simulación del estándar ISDB-T utilizados en Simulink. Fuente: (Gomez & Lapo, 2019)

### 3.1.1. Imagen de entrada

Los datos de la imagen que ingresa al transmisor son recopilados mediante el bloque denominado Signal From Workspace. Este bloque se encarga de importar los datos que se generan en el espacio de trabajo de MATLAB al modelo elaborado en Simulink. En el estándar ISDB-T cada flujo de transporte que ingresa tiene una longitud de 188 bytes, por lo cual, en este bloque se configura el número de muestras con dicho valor.

Figura 3.2. Parámetros del bloque Signal From Workspace

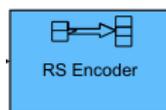


Configuración del bloque en Simulink encargado de importar la imagen desde el Workspace de MATLAB. Fuente: Simulink

### 3.1.2. Codificador Reed Solomon (n,k)

El codificador Red Solomon (n,k) donde n representa la cantidad de símbolos del código y k la cantidad de símbolos de datos permite detectar y corregir errores en los bytes del flujo de datos, se brinda protección a errores que pueden producirse en la transmisión. En el estándar de TDT, a los 188 bytes que contiene el flujo de transporte (TS) son añadidos 16 bytes, estableciendo de esta forma grupos de 204 bytes (Millán, 2018) . En la figura 3.3 se muestra el bloque de Simulink del codificador RS.

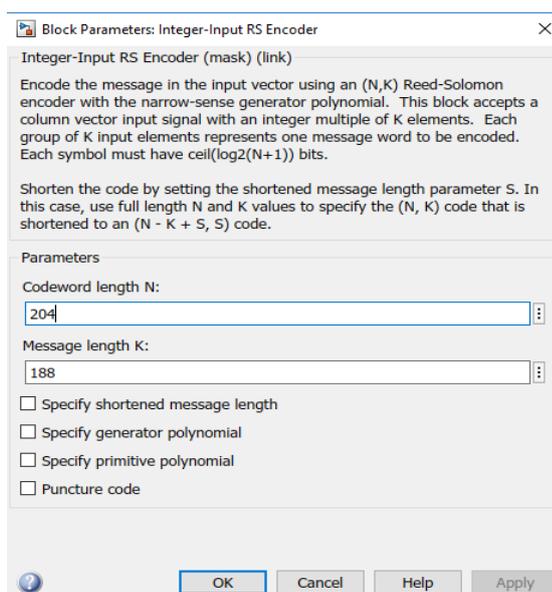
Figura 3.3. Bloque de Simulink para el codificador RS



Codificador Reed Solomon. Fuente: Simulink

En la figura 3.4 se muestran los parámetros de configuración para el codificador RS.

Figura 3.4. Parámetros del codificador RS



Configuración del codificador RS en el modelo de Simulink. Fuente: Simulink

### 3.1.3. Convertidor de entero a bits

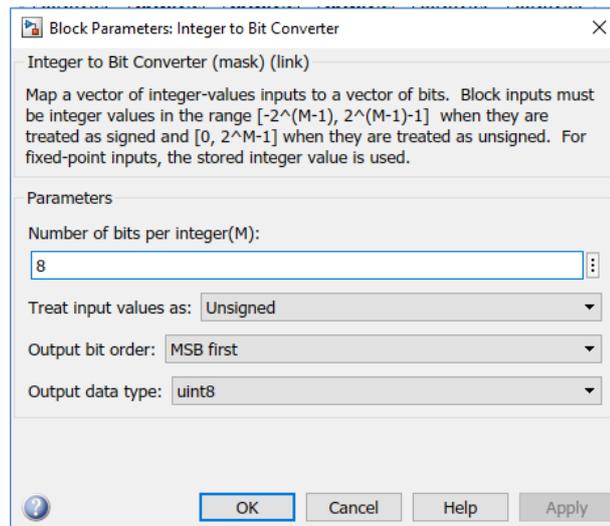
En este bloque se realiza la conversión de los valores que ingresan al bloque convertidor a bits, en este caso en la salida se obtienen 1632 bits, correspondiente a los 204 bytes del flujo TSP que salen del codificador RS, cada byte contiene 8 bits. En la figura 3.5 y 3.6 se muestra el bloque y los parámetros configurados correspondientes al convertidor de enteros a bits, respectivamente.

Figura 3.5. Bloque de Simulink para el convertidor de entero a bit



Convertidor de enteros a bits. Fuente: Simulink

Figura 3.6. Parámetros del convertidor de entero a bits

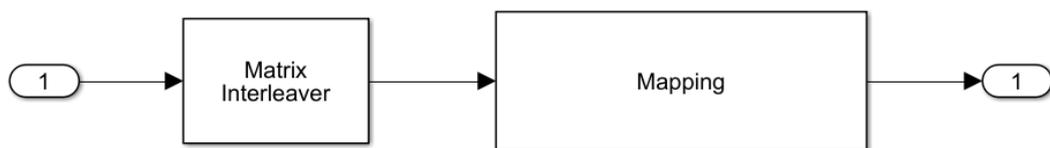


Configuración del convertidor de entero a bits en el modelo de Simulink. Fuente: Simulink

### 3.1.4. Bloque para la modulación de portadora

En este bloque, se encuentra un subsistema que contiene el entrelazado de bits y el diseño necesario para obtener la modulación 64-QAM.

Figura 3.7. Bloque de modulación de portadora



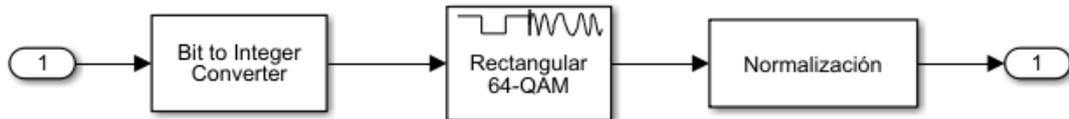
Subsistema correspondiente al bloque de modulación de portadora. Fuente: (Gomez & Lapo, 2019)

El entrelazado de bits se efectúa mediante el bloque denominado Matrix Interleaver en el cual se configura como parámetro los 1632 bits que se reciben del bloque convertidor de enteros a bits.

En el subsistema denominado Mapping que se muestra en la figura 3.8, se realiza el proceso para la modulación 64-QAM. En el bloque para la conversión de bits a enteros, se asigna un valor de 6, esto significa que cada entero tendrá un total de 6 bits. Esto debido a que,  $2^N$  donde N representa el número de bits; en la modulación 64-QAM adquiere el valor de 6 ( $2^N = 64$ ). Por lo tanto, se transmiten 6 bits por cada símbolo de la constelación. El bloque de normalización multiplica los valores obtenidos por un

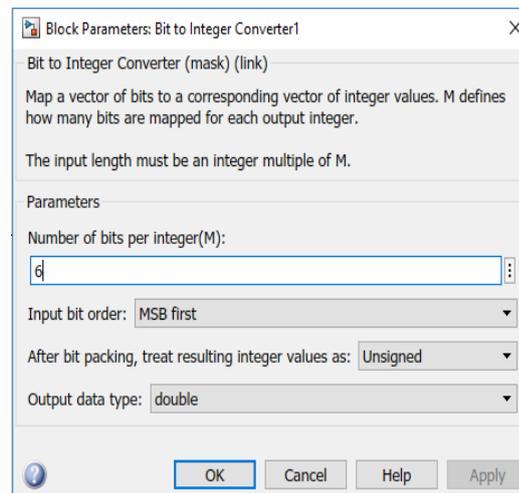
factor de  $\frac{1}{\sqrt{42}}$  establecido para la modulación 64-QAM según las especificaciones del estándar. (ARIB, 2005)

Figura 3.8. Bloque del subsistema Mapping



Esquema de bloques correspondientes al subsistema Mapping que compone al bloque de modulación de portadora. Fuente: (Gomez & Lapo, 2019)

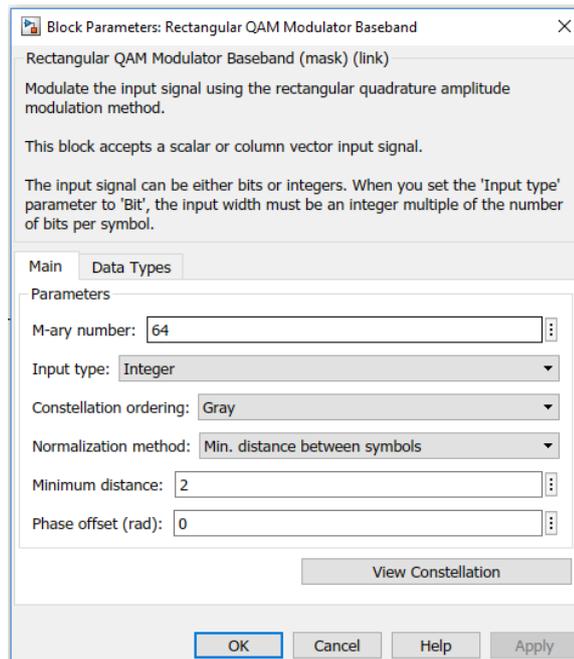
Figura 3.9. Convertidor de bits a entero



Configuración de los parámetros en el bloque convertidor de bits a enteros. Fuente: Simulink

En las figuras 3.9 y 3.10 se muestran los parámetros configurados para la conversión y la modulación 64-QAM, respectivamente. En el bloque del modulador QAM se asigna el valor de 64 a la modulación.

Figura 3.10. Modulador de banda base QAM



Configuración parámetros del modulador QAM. Fuente: Simulink.

### 3.1.5. Bloque para entrelazado de tiempo y frecuencia

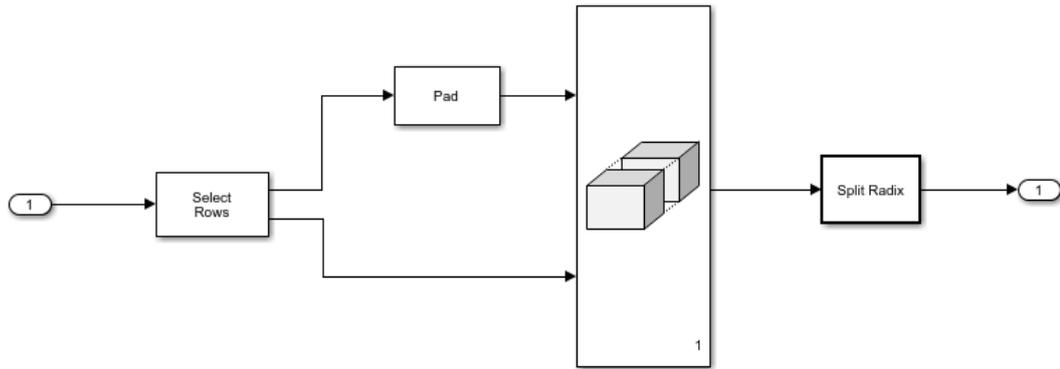
El bloque realiza el proceso de entrelazado que permite tener una mayor corrección de errores, está compuesto por buffers que contienen los datos de acuerdo al modo de operación del estándar que se está utilizando, al final del subsistema se almacenan la cantidad de portadoras que serán utilizadas para la IFFT.

### 3.1.6. Bloque del modulador OFDM

Los datos que ingresan al modulador son 1248 puntos, por lo tanto, en este bloque se configura una IFFT de 2048, debido a que, el valor de 1248 se debe aproximar al valor múltiplo de  $2^n$  más cercano que es 2048.

En la figura 3.11 se muestra el subsistema utilizado para el modulador, mediante el bloque Select Rows se divide la cantidad de muestras que ingresan en dos grupos, a uno de los cuales con el bloque Pad se le añaden los ceros necesarios para poder ejecutar una transformada de 2048 muestras. El bloque Matrix Concatenate sirve para unir nuevamente las muestras con los ceros que fueron añadidos al vector.

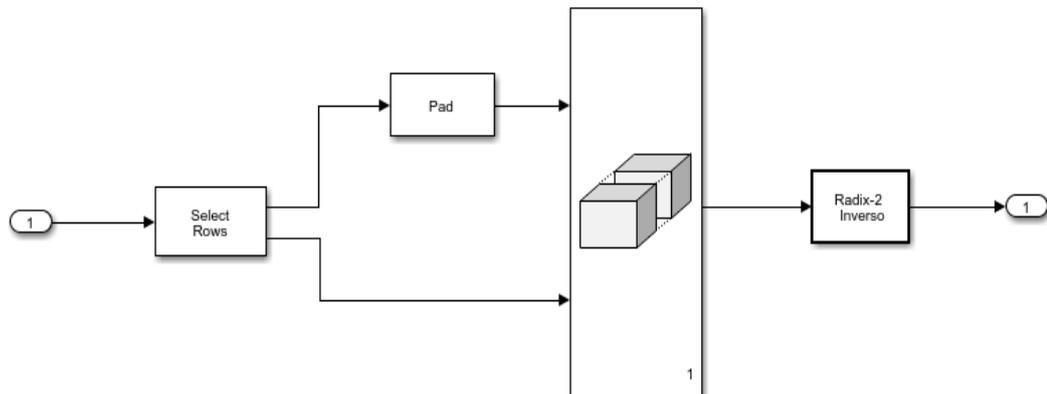
Figura 3.11. Subsistema del modulador OFDM con Split Radix



Subsistema para el modulador OFDM con el algoritmo Split Radix. Elaborado por: Carlos Alarcón

Para el algoritmo Radix-2 se utiliza el mismo esquema, cambiando el bloque Split Radix por el bloque Radix-2 inverso que contiene el script para efectuar la IFFT, como se observa en la figura 3.12.

Figura 3.12. Subsistema del modulador OFDM con Radix-2



Subsistema para el modulador OFDM con el algoritmo Radix-2. Elaborado por: Carlos Alarcón

### 3.1.7. Bloque para el canal

En la simulación se emplea un canal multicamino, el cual se encarga de generar retrasos, alteraciones en la señal que se transmite. El modelo aplicado corresponde al canal inalámbrico SUI de tipo C, el cual está diseñado para entornos urbanos con poca vegetación, las especificaciones requeridas por el canal se presentan en la tabla 3.1. En la figura 3.13 se muestra el subsistema generado para simular el canal de este tipo.

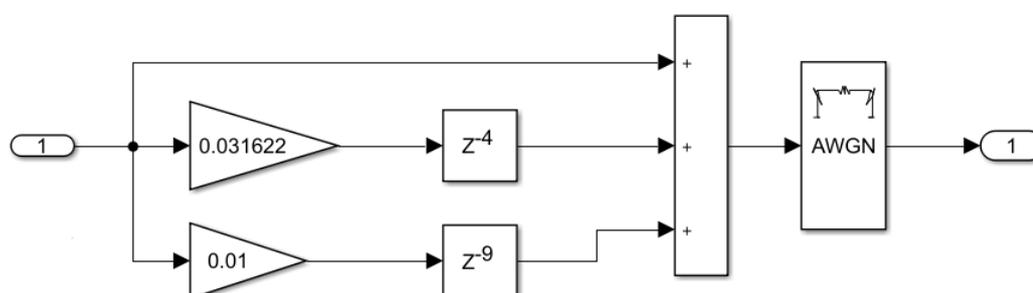
Tabla 3.1. Especificaciones del canal SUI de tipo C

	SUI-1			SUI-3		
	Tap 1	Tap 2	Tap 3	Tap 1	Tap 2	Tap 3
<b>Delay [us]</b>	0	14	20	0	4	9
<b>Ganancia [dB]</b>	0	0,1	0,398	0	0,0316	0,01
<b>Terreno</b>	Tipo A			Tipo C		

Especificaciones que se aplican en el subsistema de Simulink para el canal de comunicaciones.

Fuente: (Gomez & Lapo, 2019)

Figura 3.13. Subsistema para el canal de comunicación



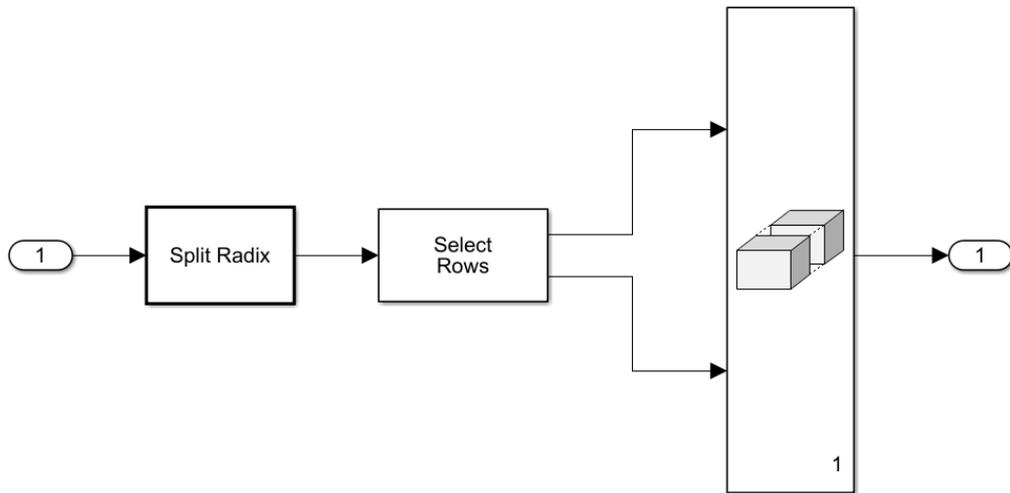
Subsistema en el cual se aplica el canal SUI de tipo c. Fuente: (Gomez & Lapo, 2019)

En el subsistema se utilizan las ganancias y delay determinados por la tabla 3.1, mediante el bloque AWGN se añade ruido Gaussiano aditivo blanco al canal.

### 3.1.8. Bloque del demodulador OFDM

El demodulador OFDM es un subsistema, en el cual se realiza el proceso inverso al modulador. Es decir, primero se ejecuta la FFT, se descomponen los datos para quitar los ceros que fueron añadidas por el modulador y posteriormente se vuelve a organizar los datos para que la imagen sea reconstruida. En la figura 3.14 se muestran los bloques que componen al subsistema para su aplicación con el algoritmo Split Radix.

Figura 3.14. Demodulador OFDM con Split Radix

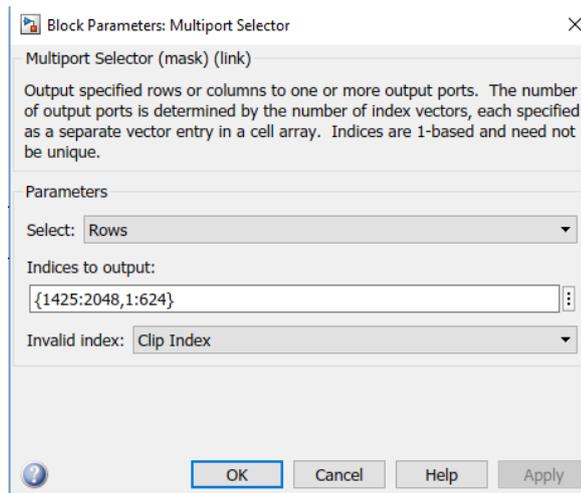


Subsistema correspondiente al demodulador OFDM con el algoritmo Split Radix. Elaborado por: Carlos Alarcón

Es importante recalcar que en este bloque son aplicados los dos algoritmos de la FFT correspondientes al estudio que se pretende con este proyecto. En el bloque denominado Split Radix se ejecuta el programa que contiene al algoritmo; desarrollado en un archivo.m. Esta función es llamada desde MATLAB hacia Simulink mediante el bloque MATLAB function.

En la salida del bloque de la FFT (Split Radix) se obtienen 2048 puntos, inicialmente antes del modulador OFDM, en el transmisor se tenían 1248 puntos, por este motivo mediante el bloque Multiport Selector, se eliminan los 800 elementos que se adicionaron para en el modulador. Se crean dos grupos con 624 puntos cada uno, mismos que son concatenados en el bloque Matrix Concatenate, en dicho bloque se configuran los parámetros para que permita el ingreso de los dos grupos de datos creados en el bloque selector. De esta forma, la imagen que fue enviada es reconstruida. En las figuras 3.15 y 3.16 se muestran los parámetros configurados para estos bloques.

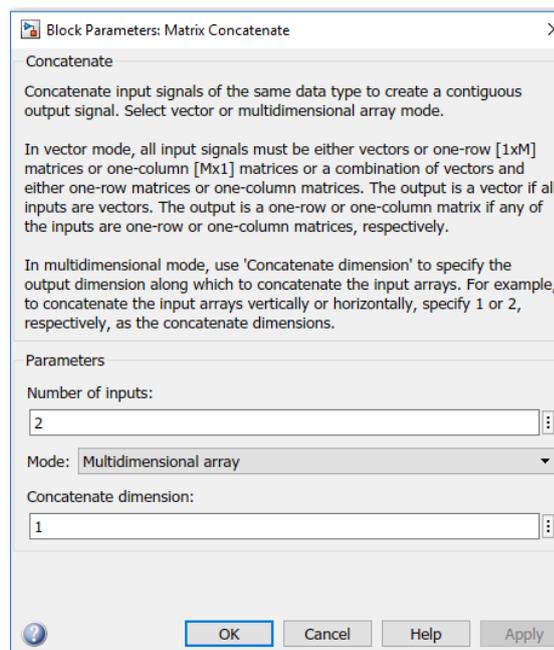
Figura 3.15. Bloque Multiport Selector



Configuración del rango de valores que tiene cada salida del bloque para reconstruir los 1248 puntos.

Fuente: Simulink

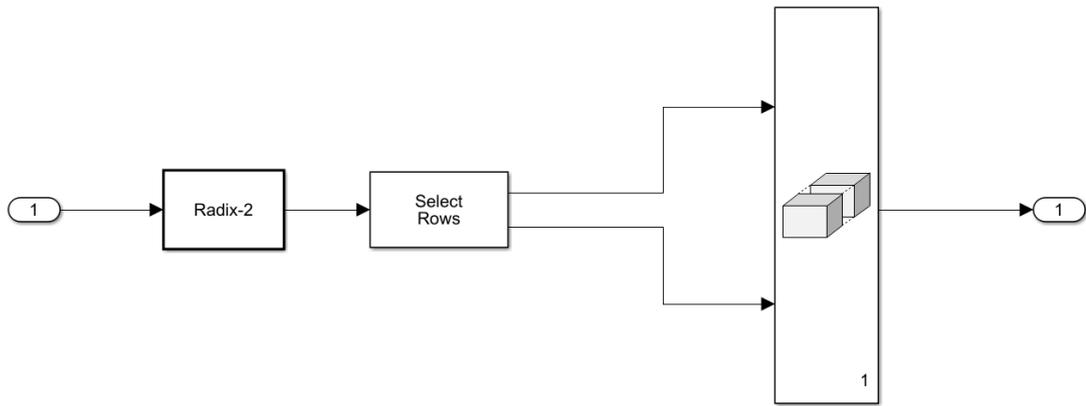
Figura 3.16. Bloque Matrix Concatenate



Configuración de los parámetros para dos entradas. Fuente: Simulink

Para el algoritmo Radix-2, de igual forma se creó una script para realizar los cálculos, este bloque reemplaza al de Split Radix; el procedimiento efectuado y la configuración de los parámetros es la misma, únicamente se cambia el tipo de algoritmo, a este subsistema se lo puede apreciar en la figura 3.17.

Figura 3.17. Demodulador OFDM con Radix-2



Bloques del subsistema correspondiente al demodulador OFDM con el algoritmo Radix-2. Elaborado por: Carlos Alarcón

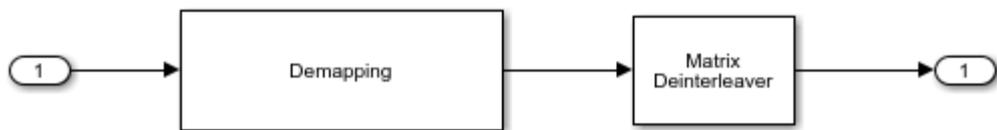
### 3.1.9. Bloque para el desentrelazado de tiempo y frecuencia

En este bloque se desarrolla un mecanismo contrario al efectuado en el entrelazado de tiempo y frecuencia para generar un vector que contenga la cantidad de bits requeridos por el bloque de demodulación de portadora.

### 3.1.10. Bloque para la demodulación de portadora

En esta etapa se ejecuta una secuencia inversa a la descrita en la modulación de portadora, primero la entrada pasa por el subsistema Demapping en la cual se realiza una conversión de enteros a bits estableciendo un valor de 6 por los requerimientos de la demodulación 64-QAM, luego los datos circulan por al bloque Matrix Deinterleaver.

Figura 3.18. Subsistema del bloque Demodulación de portadora



Bloques que conforman el proceso para demodulación de portadora. Fuente: (Gomez & Lapo, 2019)

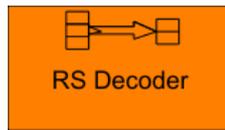
### 3.1.11. Convertidor de bits a enteros

En el convertidor se transforman los 1632 bits del flujo procedente del bloque demodulación de portadora y así obtener en su salida un valor de 204 que es requerido para el proceso que realiza el decodificador Reed Solomon.

### 3.1.12. Decodificador Reed Solomon

Este bloque se encarga del proceso de decodificación de los datos recibidos para recuperar la imagen original que fue enviada, realiza la detección y corrección de errores que puedan producirse en la transmisión. Este procedimiento es inverso al codificador, al decodificador ingresa una cantidad de símbolos para obtener la cantidad de símbolos correspondientes a los datos originales.

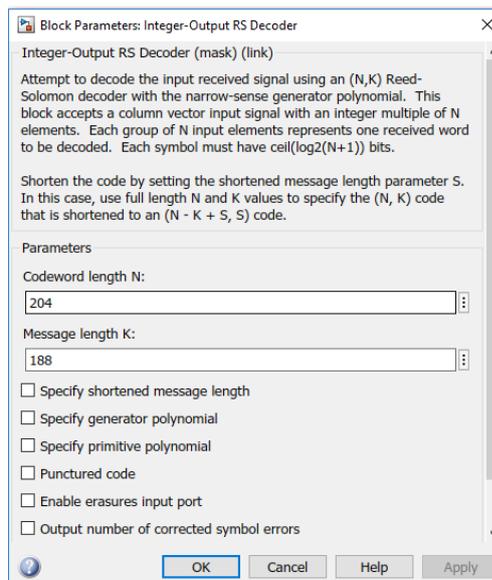
Figura 3.19. Bloque decodificador Reed Solomon



Bloque de Simulink para el decodificador Reed Solomon. Fuente: Simulink

En la figura 3.20 se muestra los parámetros de configuración para el decodificador RS.

Figura 3.20. Parámetros decodificador RS



Parámetros utilizados en el decodificador Reed Solomon. Fuente: Simulink

### 3.1.13. Imagen de salida

Los datos de la imagen obtenida luego de pasar por todos los procesos de codificación/decodificación, modulación/demodulación llega desde el transmisor al receptor y es enviada al Workspace de MATLAB mediante el bloque To Workspace para ser visualizada.

## 3.2. Código del bloque Split Radix

Como se mencionó en la sección 3.1.8, los algoritmos de la FFT fueron aplicados en el bloque del demodulador y modulador OFDM; para el caso del algoritmo Split Radix se utilizó el código que se presenta en la figura 3.21. Mismo que se basa en la implementación de las ecuaciones 2.9, 2.10 y 2.11. En cada ecuación se reemplazó el factor Twiddle  $W_N^n = e^{-j\frac{2\pi n}{N}}$  correspondiente a la ecuación 2.2. Obteniendo de esta manera las ecuaciones 3.1 para las muestras pares, 3.2 y 3.3 para las muestras impares de la función.

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} \left[ x(n) + x\left(n + \frac{N}{2}\right) \right] e^{-j\frac{4\pi nk}{N}} \quad \text{Ec. (3.1)}$$

$$X(4k + 1) = \sum_{n=0}^{\frac{N}{4}-1} \left\{ \left[ x(n) - x\left(n + \frac{N}{2}\right) \right] - j \left[ x\left(n + \frac{N}{4}\right) - x\left(n + \frac{3N}{4}\right) \right] \right\} e^{(1+4k)\left(-j\frac{2\pi n}{N}\right)} \quad \text{Ec. (3.2)}$$

$$X(4k + 3) = \sum_{n=0}^{\frac{N}{4}-1} \left\{ \left[ x(n) - x\left(n + \frac{N}{2}\right) \right] + j \left[ x\left(n + \frac{N}{4}\right) - x\left(n + \frac{3N}{4}\right) \right] \right\} e^{(3+4k)\left(-j\frac{2\pi n}{N}\right)} \quad \text{Ec. (3.3)}$$

El Script recibe como datos de entrada los valores de las portadoras que son generadas en el modulador OFDM, las cuales atraviesan el canal de transmisión para ser receptados en el bloque de demodulación OFDM. Este valor se almacena en el vector x. La cantidad total de portadoras que serán empleadas en los cálculos de la FFT con Split Radix se almacenan en la variable N, tomando el dato de la longitud del vector x que para este caso es de 2048. El vector de salida que contiene los datos de la FFT es X, inicialmente está compuesto por ceros que se llenarán con los valores de tipo complejo calculados por el algoritmo, por esta razón se utiliza la función complex para

que los datos sean del mismo tipo y no se ocasionen errores al ejecutar el Script; el vector está compuesto por N filas y 1 columna.

Figura 3.21. Script de MATLAB para el algoritmo Split Radix en el demodulador

```

function X = SplitRadix(x)
%Número de muestras obtenidas de la longitud del vector x
N = length(x);
%Creación del vector donde se almacenan los resultados de la transformada
X=complex(zeros(N,1));
%Calculo de las muestras pares con Radix-2
for k1 = 0:N/2-1
    X(2*k1+1) = 0;
    for n1 = 0:N/2-1
        %Ecuacion 3.1
        X(2*k1+1) = X(2*k1+1)+X(n1+1)+X(n1+1+N/2))*exp(-4*pi*i*i*n1*k1/N);
    end
end
%Calculo de las muestras impares con Radix-4
for k2 = 0:N/4-1
    X(4*k2+2) = 0;
    X(4*k2+4) = 0;
    for n2 = 0:N/4-1
        %Ecuacion 3.2
        X(4*k2+2) = X(4*k2+2) + ((X(n2+1) - X(n2+1+N/2)) - 1i * (X(n2+1+N/4) - X(n2+1+3*N/4))) * exp((1+4*k2) * ((-2*1i*pi*n2)/N));
        %Ecuacion 3.3
        X(4*k2+4) = X(4*k2+4) + ((X(n2+1) - X(n2+1+N/2)) + 1i * (X(n2+1+N/4) - X(n2+1+3*N/4))) * exp((3+4*k2) * ((-2*1i*pi*n2)/N));
    end
end

```

Código de programación que utiliza el algoritmo Split Radix en el demodulador OFDM. Elaborado por: Carlos Alarcón

Con la ecuación 3.1 se calculan las muestras pares de la transformada, por ejemplo,  $X(0), X(2), X(4), \dots$ , esto se logra mediante un proceso de bucles for anidados, donde el primero incrementa los valores de  $k$  desde 0 hasta  $\frac{N}{2} - 1$ , en el segundo for se implementa la ecuación mencionada, la sumatoria se ejecuta en cada iteración de  $k$  la cantidad de  $n$  veces,  $n$  está comprendido en el rango de 0 a  $\frac{N}{2} - 1$ . Cabe recalcar que MATLAB toma como primer valor de cada vector la posición 1, a diferencia de otros lenguajes de programación en los cuales la posición inicial es 0. Por esta razón,  $X(2k)$  se representa como  $X(2k+1)$ , entonces los valores del vector  $x$  correspondientes a las muestras pares se toman de las posiciones  $X(1), X(3), X(5), \dots, X\left(\frac{N}{2} - 1\right)$ .

Las muestras impares son calculadas con las ecuaciones 3.2 y 3.3, la primera realiza la sumatoria de las muestras  $X(1), X(5), X(9), \dots$ , la segunda se encarga del sumatorio de las muestras  $X(3), X(7), X(11), \dots$ , por lo expuesto en el párrafo anterior con respecto a las posiciones del vector que se consideran en MATLAB,  $X(4k+1)$  equivale a  $X(4k+2)$  donde las posiciones que se toman del vector son  $X(2), X(6), X(10), \dots, X\left(\frac{N}{4} - 1\right)$ . El equivalente de la expresión  $X(4k+3)$  es  $X(4k+4)$  donde las posiciones que se toman del vector son  $X(4), X(8), X(12), \dots$ , una vez finalizada la ejecución de los bucles los datos son almacenados en cada posición del vector  $X(\ )$  obteniéndose de esta manera la FFT. El algoritmo IFFT correspondiente a Split Radix se aplicó en el modulador OFDM efectuando variaciones en las ecuaciones que permiten calcular la FFT inversa. Estas modificaciones se presentan en las ecuaciones 3.4, 3.5 y 3.6. La figura 3.22 muestra el script de la función creada para el algoritmo Split Radix inverso.

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} \frac{1}{N} \left[ x(n) + x\left(n + \frac{N}{2}\right) \right] e^{j\frac{4\pi nk}{N}} \quad \text{Ec. (3.4)}$$

$$X(4k + 1) = \sum_{n=0}^{\frac{N}{4}-1} \frac{1}{N} \left\{ \left[ x(n) - x\left(n + \frac{N}{2}\right) \right] + j \left[ x\left(n + \frac{N}{4}\right) - x\left(n + \frac{3N}{4}\right) \right] \right\} e^{(1+4k)\left(-j\frac{2\pi n}{N}\right)} \quad \text{Ec. (3.5)}$$

$$X(4k + 3) = \sum_{n=0}^{\frac{N}{4}-1} \frac{1}{N} \left\{ \left[ x(n) - x\left(n + \frac{N}{2}\right) \right] - j \left[ x\left(n + \frac{N}{4}\right) - x\left(n + \frac{3N}{4}\right) \right] \right\} e^{(3+4k)\left(-j\frac{2\pi n}{N}\right)} \quad \text{Ec. (3.6)}$$

Figura 3.22. Script para el algoritmo Split Radix inverso

```

function X = sifftinv(x)
%Número de muestras obtenidas de la longitud del vector x
N = length(x);
%Creación del vector donde se almacenan los resultados de la transformada
X=complex(zeros(N,1));
%Calculo de las muestras pares con Radix-2
for k1 = 0:N/2-1
    X(2*k1+1) = 0;
    for n1 = 0:N/2-1
        %Ecuacion 3.4
        X(2*k1+1) = X(2*k1+1)+(x(n1+1)+x(n1+1+N/2))/N*exp(4*pi*1i*n1*k1/N);
    end
end
%Calculo de las muestras impares con Radix-4
for k2 = 0:N/4-1
    X(4*k2+2) = 0;
    X(4*k2+4) = 0;
    for n2 = 0:N/4-1
        %Ecuacion 3.5
        X(4*k2+2) = X(4*k2+2)+(X(n2+1)-X(n2+1+N/2))/N+1i*(X(n2+1+N/4)/N-x(n2+1+3*N/4)/N)*exp(2*pi*1i*n2/N)*exp(8*pi*1i*n2*k2/N);
        %Ecuacion 3.6
        X(4*k2+4) = X(4*k2+4)+(X(n2+1)-X(n2+1+N/2))/N-1i*(X(n2+1+N/4)/N-x(n2+1+3*N/4)/N)*exp(6*pi*1i*n2/N)*exp(8*pi*1i*n2*k2/N);
    end
end
end

```

Script del algoritmo Split Radix inverso en el modulador OFDM. Elaborado por: Carlos Alarcón

En la figura 3.23 se presenta el diagrama de flujo correspondiente al script utilizado para el algoritmo Split Radix. Este diagrama se aplica también para la inversa del algoritmo Split Radix utilizado en el modulador OFDM, únicamente difieren con las modificaciones que se efectuaron en las ecuaciones utilizadas.

Figura 3.23. Diagrama de flujo para el algoritmo Split Radix

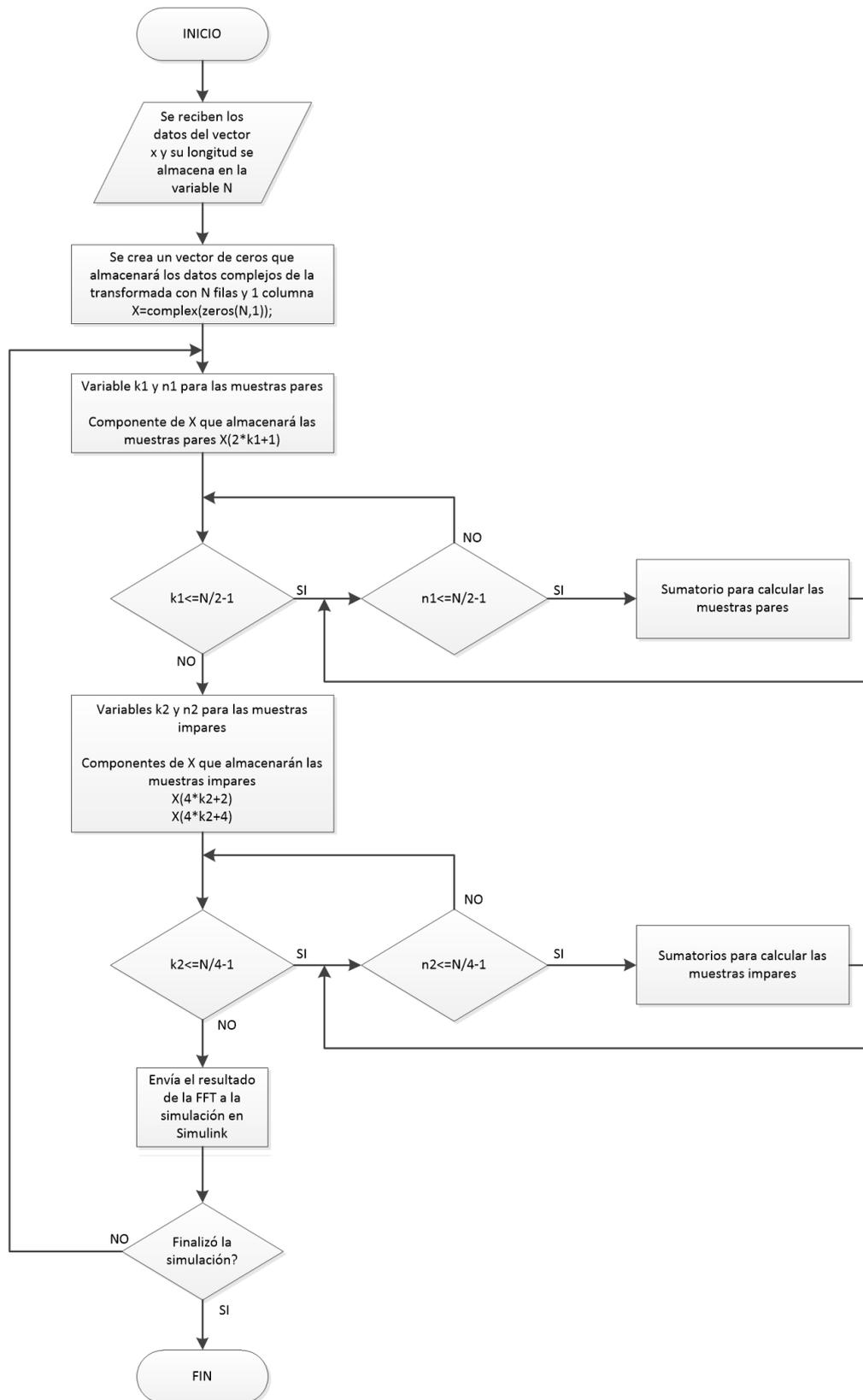


Diagrama de flujo para el script utilizado en el algoritmo Split Radix. Elaborado por: Carlos Alarcón

### 3.3. Código del bloque Radix-2

El procedimiento descrito en la sección 3.2 también es aplicado en el algoritmo Radix-2, en este caso se utilizaron las ecuaciones 2.7 y 2.8 para el cálculo de las muestras pares e impares. El script utilizado para la FFT con Radix-2 se observa en la figura 3.24.

Figura 3.24. Script de MATLAB para el algoritmo Radix-2 en el demodulador

```
function X = Radix2(x)
function X = SplitRadix(x)
% Número de muestras obtenidas de la longitud del vector x
N = length(x);
%
% Creación del vector donde se almacenan los resultados de la transformada
X=complex(zeros(N,1));
%
% Cálculo de las muestras pares con Radix-2
for k = 0:N/2-1
    X(2*k+1) = 0;
    for n = 0:N/2-1
        %Ecuación 2.7
        X(2*k+1) = X(2*k+1) + (x(n+1) + x(n+1+N/2)) * exp(-4*pi*li*n*k/N);
    end
end
% Cálculo de las muestras impares con Radix-2
for k1 = 0:N/2-1
    X(2*k1+2) = 0;
    for n1 = 0:N/2-1
        %Ecuación 2.8
        X(2*k1+2) = X(2*k1+2) + (x(n1+1) - x(n1+1+N/2)) * (exp((-2*li*pi*n1/N) * (1+2*k1)));
    end
end
end
```

Código de programación que utiliza el algoritmo Radix-2 en el demodulador OFDM. Elaborado por: Carlos Alarcón

El diagrama de flujo correspondiente al script del algoritmo Radix-2 es similar al utilizado en Split Radix, únicamente se realizan modificaciones en las variables y ecuaciones implementadas, esto se lo observa en la figura 3.25. Para el cálculo de la IFFT en las ecuaciones 2.7 y 2.8 se divide a las componentes para N y el factor twiddle es reemplazado por  $W_N^n = e^{j\frac{2\pi n}{N}}$ , como se observa en las ecuaciones 3.7 y 3.8 para las componentes pares e impares respectivamente. El script utilizado se muestra en la figura 3.26.

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} \frac{1}{N} \left[ x(n) + x\left(n + \frac{N}{2}\right) \right] e^{j\frac{4\pi nk}{N}} \quad \text{Ec. (3.7)}$$

$$X(2k + 1) = \sum_{n=0}^{\frac{N}{2}-1} \frac{1}{N} \left[ x(n) - x\left(n + \frac{N}{2}\right) \right] e^{j\frac{2\pi n}{N}(1+2k)} \quad \text{Ec. (3.8)}$$

Figura 3.25. Diagrama de flujo para el algoritmo Radix-2

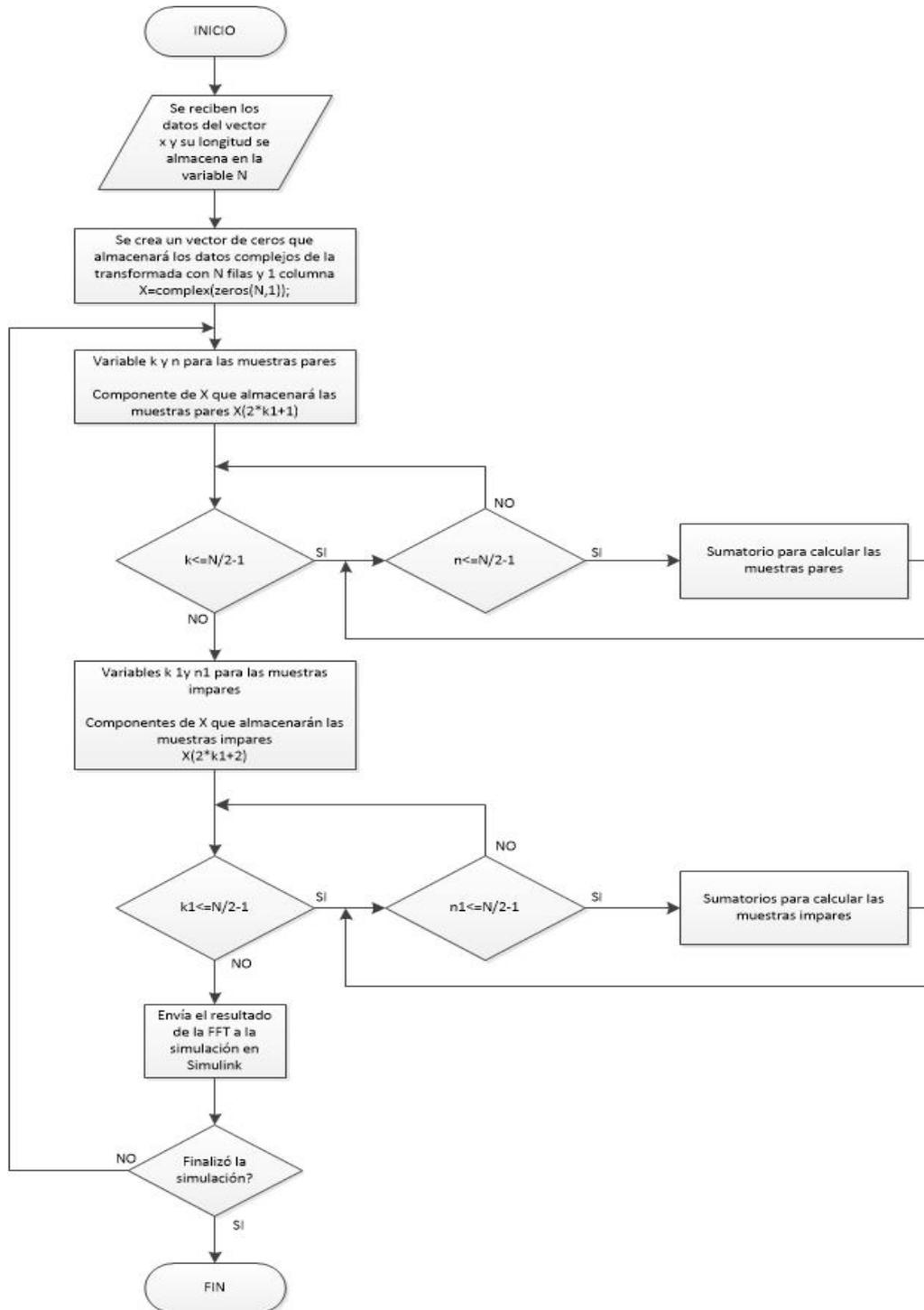


Diagrama de flujo para el script utilizado en el algoritmo Radix-2. Elaborado por: Carlos Alarcón

Figura 3.26. Script de MATLAB para el algoritmo Radix-2 en el modulador

```

function X = radix2inv(x)
    %Número de muestras obtenidas de la longitud del vector x
    N = length(x);
    %Creación del vector donde se almacenan los resultados de la transformada
    X=complex(zeros(N,1));
    %Cálculo de las muestras pares con Radix-2
    for k = 0:N/2-1
        X(2*k+1) = 0;
        for n = 0:N/2-1
            %Se dividen las componentes para N y en el factor twiddle se cambia
            %el signo a positivo
            X(2*k+1) = X(2*k+1)+(x(n+1)+x(n+1+N/2))/N*exp(4*pi*1i*n*k/N);
        end
    end
    %Cálculo de las muestras impares con Radix-2
    for k1 = 0:N/2-1
        X(2*k1+2) = 0;
        for n1 = 0:N/2-1
            %Se dividen las componentes para N y en el factor twiddle se cambia
            %el signo a positivo
            X(2*k1+2) = X(2*k1+2)+(x(n1+1)-x(n1+1+N/2))/N*(exp((2*1i*pi*n1/N)*(1+2*k1)));
        end
    end
end

```

Código de programación que utiliza el algoritmo Radix-2 inverso en el modulador OFDM. Elaborado por: Carlos Alarcón

## CAPÍTULO 4

### ANÁLISIS Y RESULTADOS

En el presente capítulo se muestran los datos obtenidos mediante la simulación del estándar ISDB-T de Simulink con los algoritmos de la FFT Radix-2 y Split Radix. Para verificar los parámetros obtenidos tales como BER, eficiencia y retardo que se generó en la transmisión de la imagen.

#### 4.1. Datos obtenidos con el algoritmo Radix-2

En la tabla 4.1 se presentan los resultados que se obtuvieron al ejecutar la simulación con diferentes valores de SNR en el canal, aplicando la IFFT y FFT mediante el algoritmo Radix-2.

Tabla 4.1. Resultados obtenidos con el algoritmo Radix-2

<b>B E R</b>	<b>BITS ERRADOS</b>	<b>BITS TOTALES</b>	<b>SNR [dB]</b>
0,1329	1488224	11197280	10
0,1075	1203886	11197280	12
0,0851	952968	11197280	14
0,0666	745575	11197280	16
0,0536	600314	11197280	18
0,0471	527384	11197280	20

Resultados del BER y eficiencia que se obtuvieron mediante la simulación del estándar aplicando el algoritmo Radix-2. Elaborado por: Carlos Alarcón

#### 4.2. Datos obtenidos con el algoritmo Split Radix

En la tabla 4.1 se presentan los resultados que se obtuvieron al ejecutar la simulación con diferentes valores de SNR en el canal, aplicando la IFFT y FFT mediante el algoritmo Split Radix.

Tabla 4.2. Resultados obtenidos con el algoritmo Split Radix

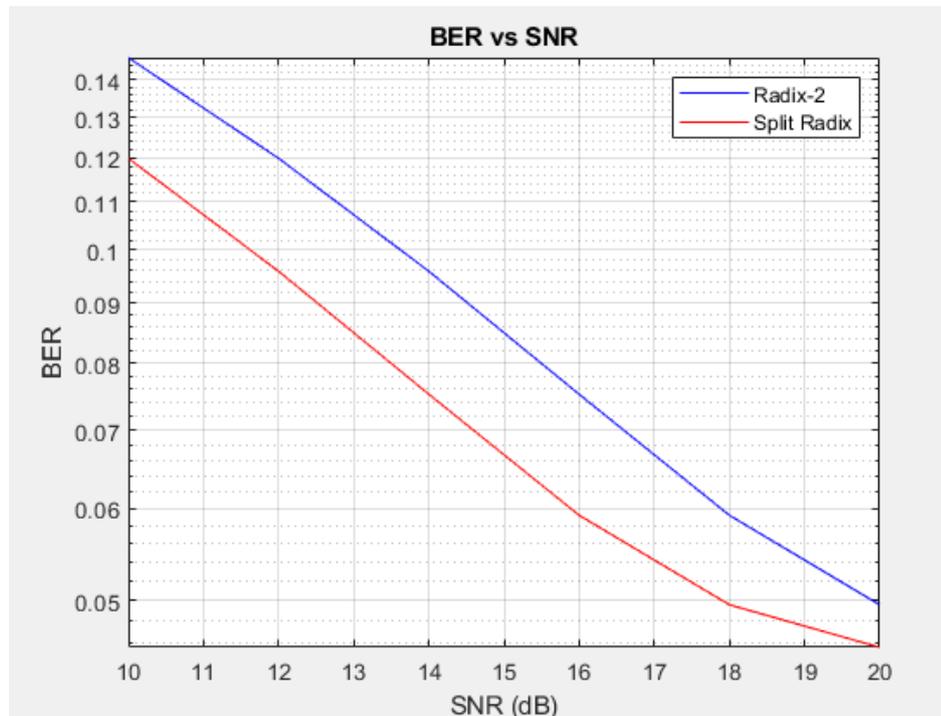
BER	BITS ERRADOS	BITS TOTALES	SNR [dB]
0,1199	1342897	11197280	10
0,0959	1073946	11197280	12
0,0753	843081	11197280	14
0,0593	663528	11197280	16
0,0496	555633	11197280	18
0,0456	511115	11197280	20

Resultados del BER y eficiencia que se obtuvieron mediante la simulación del estándar aplicando el algoritmo Split Radix. Elaborado por: Carlos Alarcón

### 4.3. Gráficas de los resultados obtenidos

En la figura 4.1 se muestran las gráficas obtenidas para del BER con relación a los distintos valores de SNR que se tomaron como referencia para poder establecer una diferencia.

Figura 4.1. BER vs SNR aplicando los algoritmos de la FFT



Gráfica obtenida de BER con los datos correspondientes a los algoritmos Radix-2 y Split Radix. Elaborado por: Carlos Alarcón

En la simulación se cambiaron los valores de SNR en el canal de comunicación para establecer la diferencia que se genera en la recepción. Como se aprecia en la figura

4.1, en ambos casos el valor del BER decreta exponencialmente con el aumento en la relación de señal a ruido, esto significa que cuando en el canal se tienen valores de SNR aceptables para el medio la cantidad de errores en la transmisión disminuye. Según la gráfica de la simulación obtenida, la diferencia entre los dos algoritmos no es muy grande pero el algoritmo Split Radix presenta una menor cantidad de pérdidas, además entre los 18 a 20 dB en el algoritmo Split Radix el decremento disminuye y el BER presenta valores más estables.

Estas cifras oscilan en el rango de  $10^{-2}$  para los valores de SNR que se recomiendan como buenos para el canal de transmisión, esto es entre 16 a 20 dB. Sin embargo, la señal recibida no es óptima con Split Radix en 20 dB se tiene un resultado de 4 bits erróneos por cada 100 que se reciben. En este tipo de comunicaciones un valor aceptable para la tasa de bit errado se encuentra en el rango de  $10^{-3}$ .o inferior.

En la figura 4.2, se muestra la imagen recibida con un SNR de 10, con el algoritmo Radix-2.

Figura 4.2. Imagen recibida

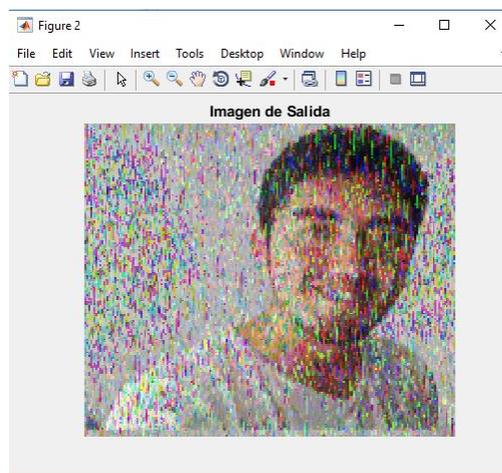


Imagen que se recibe luego de todo el proceso al utilizar Radix-2. Elaborado por: Carlos Alarcón

A continuación, se presentan las imágenes recibidas con un SNR de 20, aplicando en la simulación los algoritmos Radix-2 y Split Radix, en las figuras 4.3 y 4.4 respectivamente.

Figura 4.3. Imagen recibida con el algoritmo Radix-2

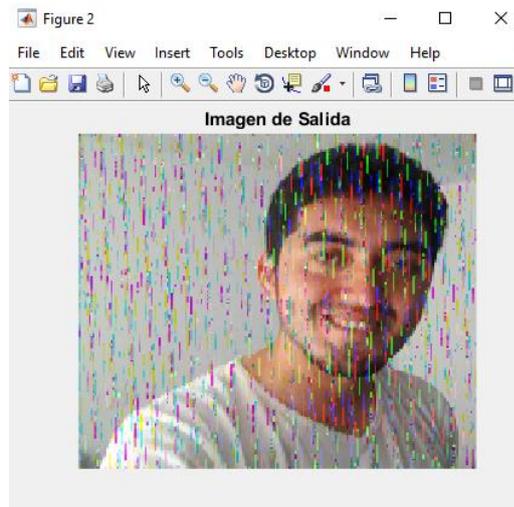


Imagen que se recibe luego de todo el proceso con Radix-2. Elaborado por: Carlos Alarcón

Figura 4.4. Imagen recibida con el algoritmo Split Radix

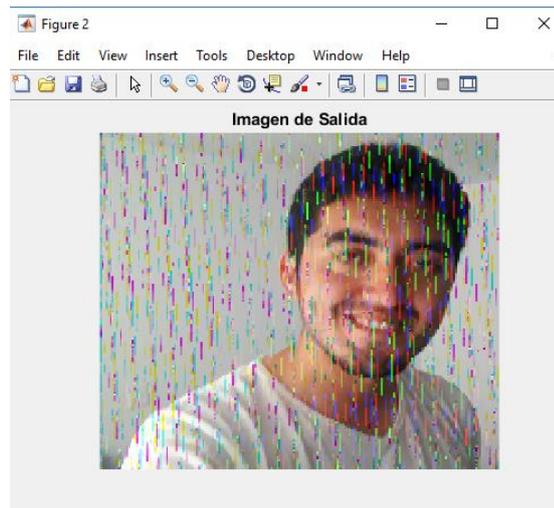


Imagen que se recibe luego de todo el proceso con Split Radix. Elaborado por: Carlos Alarcón

En las imágenes mostradas, la imagen recibida con un valor de SNR 10 presenta mucho ruido y una calidad muy baja. La calidad de la imagen obtenida con un SNR de 20 presenta una mejor resolución, la diferencia entre los algoritmos no es muy notoria, con el algoritmo Split Radix existe una pequeña mejora debido a que hay una menor cantidad de bits perdidos durante la transmisión. Sin embargo, la calidad de la imagen obtenida no es óptima, en el canal según lo observado se está produciendo un efecto de sal y pimienta que se ha reducido mediante la implementación de un filtro mediano. Además, la tasa de bit errado no se encuentra dentro de los valores

recomendados, hay pixeles que no están siendo reconstruidos adecuadamente por la cantidad de bits perdidos en el trayecto.

#### 4.4. Eficiencia

La eficiencia de la transmisión se la calculó tomando en cuenta los valores entregados por la simulación, tales como, el total de bits transmitidos y la cantidad de bits erróneos para cada valor de SNR que se asignó en el canal de comunicación. Esto se lo efectuó mediante la ecuación 4.1.

$$\eta\% = \frac{|Total\ de\ bits - Bits\ erróneos|}{|Total\ de\ bits|} \quad Ec.\ (4.1)$$

Aplicando esta ecuación se puede estimar la relación entre el valor real de bits que se transmiten y la cantidad aproximada que está llegando al receptor, este valor multiplicó por 100 nos da el resultado en porcentaje. Los valores calculados para Radix-2 se muestra en la tabla 4.3, los datos son tomados de la tabla 4.1. Por ejemplo, para un SNR=10, la eficiencia de transmisión es la siguiente:

$$\eta\% = \frac{|11197280 - 1488224|}{|11197280|}$$

$$\eta = 0,86709 \times 100\%$$

$$\eta = 0,8671$$

Tabla 4.3. Valores calculados de la eficiencia con el algoritmo Radix-2

SNR [dB]	EFICIENCIA	%
10	0,8671	86,71
12	0,8925	89,25
14	0,9149	91,49
16	0,9334	93,34
18	0,9464	94,64
20	0,9529	95,29

Valores de la eficiencia de transmisión con el algoritmo Radix-2 calculados con la ecuación 4.1.

Elaborado por: Carlos Alarcón

Los valores calculados con el algoritmo Split Radix se muestra en la tabla 4.4, los datos son tomados de la tabla 4.2.

Por ejemplo, para un SNR=10, la eficiencia de transmisión es la siguiente:

$$\eta\% = \frac{|11197280 - 1342897|}{|11197280|}$$

$$\eta = 0,88006 \times 100\%$$

$$\eta = 0,8801$$

Tabla 4.4. Valores calculados de la eficiencia con el algoritmo Split Radix

SNR [dB]	EFICIENCIA	%
10	0,8801	88,01
12	0,9041	90,41
14	0,9247	92,47
16	0,9407	94,07
18	0,9504	95,04
20	0,9544	95,44

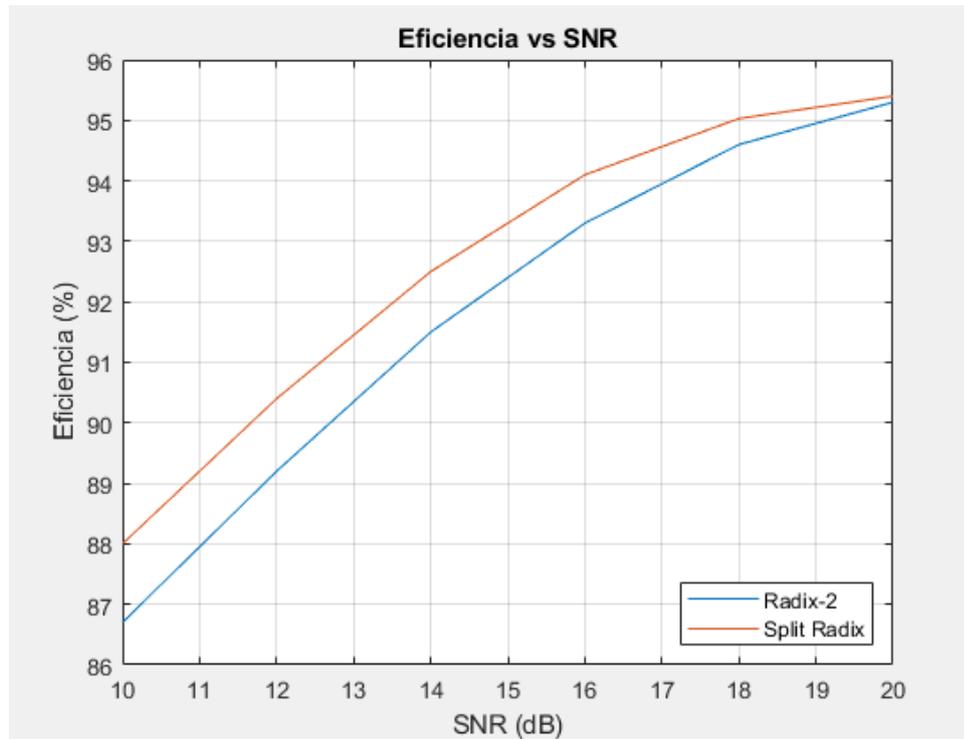
Valores de la eficiencia de transmisión con el algoritmo Split Radix calculados con la ecuación 4.1.

Elaborado por: Carlos Alarcón

En la figura 4.5 se presenta una gráfica comparativa de la eficiencia entre ambos algoritmos para distintos valores de SNR.

De acuerdo a la gráfica mostrada, el algoritmo Split Radix posee una mayor eficiencia en todos los niveles de SNR, la diferencia no es tan grande; con un SNR de 10 aproximadamente un 2% , que va reduciéndose a medida que se incrementa el valor de la relación de señal a ruido, lo cual se visualiza de mejor forma con un SNR=20 donde la eficiencia de transmisión difiere de un algoritmo a otro por 0,13%. Estos resultados significan que en el algoritmo Split Radix existe una menor pérdida de bits durante la transmisión.

Figura 4.5. Gráfica de la eficiencia de transmisión aplicando los algoritmos Radix-2 y Split Radix



Gráficas comparativas de la eficiencia obtenida con los algoritmos Radix-2 y Split Radix con distintos escenarios de SNR. Elaborado por: Carlos Alarcón

#### 4.5. Retardo

El retardo o delay que se tiene en la recepción de la imagen, en la simulación es visualizado mediante el bloque Find Delay de Simulink, el valor que arroja se encuentra en una medida de número de muestras o bits; el bloque realiza una comparación entre los bits que se transmiten y los bits que se reciben según la imagen que se envíe.

Este valor sirve para realizar el cálculo del delay que existe con cada algoritmo utilizando la ecuación 4.2. En donde la velocidad de transmisión para el estándar es de 19,6 Mbps. (Gomez & Lapo, 2019)

$$delay = \frac{Total\ de\ bits\ [bits]}{Velocidad\ de\ transmisión\ [Mbps]} \quad Ec.\ (4.2)$$

El resultado obtenido de la función find delay para los algoritmos fue el mismo, con un valor de 27270 muestras o bits, realizando el cálculo reemplazando los datos en la ecuación 4.2, se observa que el retraso es de 1,39 ms.

$$delay = \frac{27270 \text{ bits}}{19,6 \times 10^6 \text{ bps}}$$

$$delay = 1,39 \times 10^{-3} \text{ s} = 1,39 \text{ ms}$$

Mediante la herramienta de análisis de Simulink denominada Profiler Report, se obtuvieron los valores mostrados en la tabla 4.5, correspondientes al tiempo de ejecución de cada algoritmo en la simulación.

De acuerdo a los valores conseguidos, la simulación con el algoritmo Split Radix presenta un menor tiempo de ejecución total que con el algoritmo Radix-2. Además, en los bloques de modulador y demodulador OFDM donde se aplican los algoritmos, con Split Radix la diferencia del tiempo en que se realizan las operaciones es similar en el modulador y significativa en el demodulador, observándose que el tiempo utilizado por Split Radix en el total de la simulación es del 32,7 % para el modulador y 33,2% para el demodulador. Mientras que en Radix-2 estos porcentajes para modulador y demodulador son del 32,4% y 40,9 % respectivamente.

Tabla 4.5. Tiempos de simulación para los algoritmos

Algoritmo	Tiempo de simulación	Proceso	Tiempo de ejecución	Porcentaje
Radix-2	1148,52 [s]	Modulador OFDM	372,35 [s]	32,4 %
		Demodulador OFDM	470,18 [s]	40,9 %
Split Radix	891,84 [s]	Modulador OFDM	291,93 [s]	32,7 %
		Demodulador OFDM	295,5 [s]	33,2 %

Tiempos utilizados por los algoritmos Radix-2 y Split Radix en la simulación de Simulink para el estándar ISDB-T. Elaborado por: Carlos Alarcón

## CONCLUSIONES

Mediante la aplicación de los algoritmos Radix-2 y Split Radix en el esquema de bloques de Simulink, se concluye que los algoritmos, a través de software tienen aproximadamente el mismo comportamiento. Los valores obtenidos de la FFT son iguales, sin embargo, se observaron ciertas diferencias en la tasa de bit errado y eficiencia, puesto que con Split Radix el estándar resulta un 0,13% más eficiente que Radix-2 ya que existe una menor pérdida de bits durante la transmisión de la imagen. No obstante, el BER no presenta valores óptimos, porque con cada algoritmo de 100 bits transmitidos existe una pérdida de 4 y 5 bits, respectivamente. Para que la transmisión sea óptima estas cantidades deberían oscilar con una relación entre  $10^{-3}$  a  $10^{-4}$ . El delay resultante entre la señal transmitida y recibida es el mismo con ambos algoritmos, teniendo 1.39 ms.

Según el análisis del esquema para el estándar ISDB-T, los algoritmos se aplican en las etapas de modulación y demodulación para la transmisión y recepción respectivamente, el uso de la transformada rápida de Fourier mejora la eficiencia computacional necesaria para realizar los cálculos TDF en el esquema de OFDM, con su utilización los datos que ingresan al modulador se colocan en subportadoras, luego son combinados, transmitidos y reconstruidos en el receptor, permitiendo que estas subportadoras sean ortogonales y no causen interferencia entre sí. Por otro lado, las ecuaciones aplicadas en los scripts para realizar la FFT e IFFT con el algoritmo Split Radix y Radix-2 arrojaron los mismos resultados que los bloques FFT e IFFT propios de Simulink y que las funciones “fft” e “ifft” de MATLAB, comprobando de esta manera que el algoritmo funciona correctamente en la simulación.

Teóricamente el algoritmo Split Radix es más eficiente que Radix-2 ya que utiliza una menor cantidad de sumas y multiplicaciones complejas por la forma en la que se encuentra estructurado gracias a la optimización que adquiere del algoritmo Radix-4. En la simulación el algoritmo Radix-2 utilizó más recursos de la máquina para ejecutar el script; necesitando de 1148,52 segundos para completar la simulación, de los cuales 842,53 segundos corresponden a los bloques de modulación y demodulación OFDM donde se aplica el algoritmo, siendo el 73,3 % del tiempo total empleado. El script utilizado con Split Radix tiene un mejor desempeño, ya que requirió de 891,84

segundos para completar la simulación, siendo un 22,35 % más eficiente, en este algoritmo los bloques de modulación y demodulación utilizaron 587,43 segundos o el 65,9 % del tiempo total.

## RECOMENDACIONES

Para la obtención de ciertos valores al momento de ejecutar la simulación, se recomienda comentar los bloques que no se consideren necesarios dependiendo del valor que se requiera, para acelerar el tiempo de ejecución de la simulación.

Es recomendable probar las configuraciones con los otros modos de operación que posee el estándar ISDB-T para observar si se presentan mejoras en la imagen recibida, puesto que el aplicado en este trabajo corresponde al modo 1, el cual se adapta de mejor forma a los dispositivos móviles; el sistema móvil al ser baja resolución presenta una mayor cantidad de pérdidas en la imagen recibida.

Para observar de forma adecuada el funcionamiento del algoritmo Split Radix, este debería ser aplicado en hardware como una FPGA de alto rendimiento para utilizar su procesamiento como base para el cálculo de la FFT; y de esta forma obtener diferencias más claras del tiempo y recursos de cómputo que emplea con respecto a Radix-2, debido a que en la simulación los resultados en todo el proceso son muy similares exceptuando los tiempos de ejecución para cada algoritmo.

## REFERENCIAS

- Albertí, E. B. (2006). *Procesado digital de señales. Fundamentos para comunicaciones y control - II*. Barcelona: Ediciones UPC.
- Alcocer, J., & Arteaga, J. (2013). Análisis del estándar de televisión digital terrestre brasileño. Guayaquil, Ecuador: Escuela Superior Politécnica del Litoral. Obtenido de <https://www.dspace.espol.edu.ec/retrieve/130413/D-83122.pdf>
- ARIB. (2005). *TRANSMISSION SYSTEM FOR DIGITAL TERRESTRIAL TELEVISION BROADCASTING*. Obtenido de ARIB: [http://www.arib.or.jp/english/html/overview/doc/6-STD-B31v1\\_6-E2.pdf](http://www.arib.or.jp/english/html/overview/doc/6-STD-B31v1_6-E2.pdf)
- Cassagnes, A. D. (2016). Implementación y análisis de algoritmos de cálculo de Transformada Rápida de Fourier para su aplicación en sistemas OFDM. Buenos Aires: Universidad de Buenos Aires. Obtenido de <http://laboratorios.fi.uba.ar/lse/tesis/LSE-FIUBA-Tesis-Grado-Andres-Cassagnes-2016.pdf>
- Chassaing, R. (2002). *DSP Applications Using C and the TMS320C6x DSK*. John Wiley & Sons.
- Cheng, Y.-T. (Marzo de 2000). *Texas Instruments*. Obtenido de <http://www.ti.com/lit/an/spra654/spra654.pdf>
- Cruz, W. H. (2013). Desarrollo y puesta en marcha de un sistema embebido en FPGA para el procesamiento de datos en 2D con potenciales aplicaciones en óptica adaptativa. México: Universidad Nacional Autónoma de México. Obtenido de <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/6354/Tesis.pdf?sequence=1>
- DBEG. (2007). *Digital Broadcasting Experts Group*. Obtenido de [https://www.dibeg.org/techp/feature/ANNEX-AA\\_spanish.pdf](https://www.dibeg.org/techp/feature/ANNEX-AA_spanish.pdf)
- DiBEG. (2007). Obtenido de Digital Broadcasting Experts Group (DiBEG): [https://www.dibeg.org/techp/feature/isdb-t\\_Spanish.pdf](https://www.dibeg.org/techp/feature/isdb-t_Spanish.pdf)
- DiBEG. (2007). Obtenido de [https://www.dibeg.org/techp/feature/ANNEX-AA\\_spanish.pdf](https://www.dibeg.org/techp/feature/ANNEX-AA_spanish.pdf)
- George, A., & Chu, E. (2000). *Inside the FFT Black Box*. Boca Raton: CRC Press.
- Gomez, J., & Lapo, H. (2019). *Análisis comparativo entre DVB-T e ISDB-T utilizando modulación CC-OFDM*. Obtenido de Repositorio Digital-UPS: <https://dspace.ups.edu.ec/handle/123456789/16693>

- Guoan , B., & Yonghong, Z. (2004). *Transforms and Fast Algorithms for Signal Analysis and Representations* (Primera ed.). Springer Science+Business Media.
- ITU-R. (2016). *Handbook on Digital Terrestrial Television Broadcasting Networks and Systems Implementation*. Geneva, Suiza.
- Juan, C., Sara, R., James , L., & José Molina. (2008). *International Symposium on Distributed Computing and Artificial Intelligence*. Springer.
- Lezana Illesca, P., & Muratt Rodriguez, J. (20 de Mayo de 2005). *Universidad Técnica Federico Santa María*. Obtenido de [http://www2.elo.utfsm.cl/~elo385/docs/Biblio/Lab3/Lectura\\_FFT.pdf](http://www2.elo.utfsm.cl/~elo385/docs/Biblio/Lab3/Lectura_FFT.pdf)
- Mejía, J., Garcia, R., & Ramos, B. (2014). Diseño, simulación e implementación del esquema de modulación OFDM y de la ecualización en el dominio de la frecuencia. ESPOL. Obtenido de <http://www.dspace.espol.edu.ec/xmlui/handle/123456789/25437>
- Millán, J. (2018). *Técnicas y procesos en infraestructuras de telecomunicaciones*. Madrid: Paraninfo.
- Millan, R. (2010). *Televisión Digital Terrestre (TDT)*. Obtenido de ramonmillan.com: <https://www.ramonmillan.com/documentos/tdt.pdf>
- Pisciotta, N. (2010). Sistema ISDB-Tb (Primera parte). *Serie Materiales de Investigación*(9), 26-44.
- Pisciotta, N. O. (2014). Remultiplexor ISDB-Tb. *Memoria Investigaciones en Ingeniería*(12), 58.
- Proakis, J., & Manoakis, D. (1996). *Digital Signal Processing Principles, Algorithms, and Applications*. Prentice-Hall.
- Rivas, O. N. (Mayo de 2012). Sistemas OFDM de alta eficiencia espectral. Obtenido de [https://e-archivo.uc3m.es/bitstream/handle/10016/16948/TFG\\_Olaia\\_Nehme\\_Rivas.pdf](https://e-archivo.uc3m.es/bitstream/handle/10016/16948/TFG_Olaia_Nehme_Rivas.pdf)
- Singh, G., & Alphones, A. (2004). OFDM Modulation Study for a Radio-over-Fiber System for Wireless LAN (IEEE 802.11a). *ICICS-PCM 2003*, 3, 1460-1464.
- Sotelo, R., Durán, D., & Joskowicz, J. (2011). Sistema de Transmisión ISBD-T. *Memoria de trabajos de difusión científica y técnica*(9), 69. Obtenido de [http://www.um.edu.uy/\\_upload/\\_descarga/web\\_descarga\\_240\\_SistemadetransmisinISDB-T.-Sotelo\\_Durn\\_Joskowicz.pdf](http://www.um.edu.uy/_upload/_descarga/web_descarga_240_SistemadetransmisinISDB-T.-Sotelo_Durn_Joskowicz.pdf)

- Vargas, C., Lopez, W., & da Rocha, C. (2007). Sistemas de Comunicación Inalámbrica MIMO-OFDM. *Acta Nova*, 3(4), 762. Obtenido de <http://www.scielo.org.bo/pdf/ran/v3n4/v3n4a09.pdf>
- Watanabe, C. (2009). Diseño de la transformada rápida de Fourier con algoritmo Split-Radix en FPGA. Lima: Pontificia Universidad Católica del Perú. Obtenido de <http://hdl.handle.net/20.500.12404/295>