

# **TRABAJO ESPECIAL DE GRADO**

## **SIMULACION DEL CODIGO LDPC DEL ESTANDAR AOS EN SISTEMAS DE TRANSMISION DE DATOS SATELITAL**

Presentado ante la Ilustre  
Universidad Central de Venezuela

Por la Bachiller:

Ortega M, Jessica A

Para optar al Título de

Ingeniero Electricista

Caracas, 2017

# **TRABAJO ESPECIAL DE GRADO**

## **SIMULACION DEL CODIGO LDPC DEL ESTANDAR AOS EN SISTEMAS DE TRANSMISION DE DATOS SATELITAL**

Tutor Académico:

Prof. Francisco Varela

Presentado ante la Ilustre  
Universidad Central de Venezuela

Por la Bachiller:

Ortega M, Jessica A

Para optar al Título de  
Ingeniero Electricista

Caracas, 2017

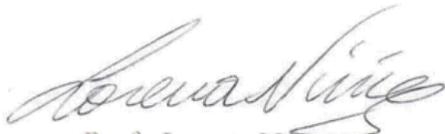
## CONSTANCIA DE APROBACIÓN

Caracas, 20 de marzo de 2017

Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por la Bachiller Jessica A. Ortega M., titulado:

### “SIMULACIÓN DEL CODIGO LDPC DEL ESTANDAR AOS EN SISTEMAS DE TRANSMISIÓN DE DATOS SATELITAL”

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la mención de Comunicaciones, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.



Prof. Lorena Núñez  
Jurado



Prof. Luis Fernández  
Jurado



Prof. Francisco Varela  
Prof. Guía

**Ortega M., Jessica A.**

## **SIMULACION DEL CODIGO LDPC DEL ESTANDAR AOS EN SISTEMAS DE TRANSMISION DE DATOS SATELITAL**

Tutor Académico: Prof. Francisco Varela.

Tesis. Caracas, U.C.V. Facultad de Ingeniería. Escuela de Ingeniería Eléctrica. Año  
2017, 97p.

**Palabras Claves:** Codificación de canal. Códigos de comprobación de paridad de baja densidad. Estándar sistema avanzado en órbita.

**Resumen:** El presente trabajo de grado consiste en la simulación de los códigos de comprobación de paridad de baja densidad (LDPC) bajo el estándar de sistema avanzado en órbita para la transmisión de datos satelitales. El trabajo se divide en tres partes fundamentales. La primera, donde se comprueban las utilidades de usar un diseño estructurado para el envío de datos satelitales y los beneficios de usar matrices de paridad de gran tamaño. La segunda parte, consta de tres simulaciones con las cuales se demuestra la correspondencia de los resultados a nivel de decodificación con la finalidad de recuperar la fuente de información enviada completamente, esto se realiza enviando en cada una de las distintas simulaciones las siguientes fuentes de información, una o dos letras, una imagen JPEG y una oración. La tercera y última parte, consiste en demostrar los beneficios que se obtienen de aplicar un bloque de aleatorización antes de la modulación, ya que este el código LDPC establecido por el estándar AOS no proporciona suficientes transiciones de bits necesarias para la correcta recuperación del reloj en el receptor. Todas las simulaciones que componen este trabajo de grado son realizadas en la herramienta MATLAB.

## INDICE

CAPITULO I.....	1
INTRODUCCION .....	1
CAPITULO II.....	3
FUNDAMENTO DE LA INVESTIGACION .....	3
2.1 Planteamiento del problema .....	3
2.2 Antecedentes.....	4
2.3 Objetivos.....	4
2.3.1 Objetivo General.....	4
2.3.2 Objetivos Específicos .....	4
CAPITULO III.....	6
MARCO TEORICO.....	6
3.1 La Comunicación Digital .....	6
3.1.1 Fuentes de información.....	7
3.1.2 Codificación y decodificación de fuente .....	8
3.1.3 Codificación y decodificación de canal .....	9
3.1.4 Modulación y demodulación digital .....	10
3.1.5 Canal de comunicaciones.....	10
3.2 Ruido .....	11
3.2.1 Tipos de ruido.....	12
3.2.2 Ruidos del bloque imnoise de la herramienta MATLAB [24] .....	14
3.3 Estándar AOS .....	15
3.3.1 Funciones generales.....	15
3.3.2 Composición de la trama [2].....	16

3.3.3 Aleatorización [7] .....	20
3.4 Códigos.....	21
3.4.1 Códigos de bloques lineales.....	22
3.4.2 Códigos de paridad de baja densidad.....	23
CAPITULO IV .....	40
METODOLOGIA .....	40
4.1 Estudio Documental .....	40
4.2 Pruebas Prácticas.....	40
CAPITULO V .....	51
RESULTADOS.....	51
5.1 Simulación 1. Trama del estándar AOS codificada por el código LDPC. ....	51
5.2 Simulación 2. Transmisión de letras codificadas y decodificadas con el código LDPC.....	60
5.3 Simulación 3. Transmisión de imagen codificada y decodificada con el código LDPC.....	66
5.4 Simulación 4. Transmisión de mensajes codificados y decodificados con el código LDPC.....	71
5.5 Simulación 5 y 6. Bloque de aleatorización.....	72
CAPITULO VI.....	77
CONCLUSIONES Y RECOMENDACIONES.....	77
6.1 Conclusiones.....	77
6.2 Recomendaciones .....	78
CAPITULO VII .....	79
MATERIALES DE REFERENCIA .....	79
7.1 Referencias bibliográficas .....	79

7.2 Apéndice.....	82
7.2.1 Acrónimos.....	82
7.3 Anexos.....	83
7.3.1 Funcionamiento de la simulación 1 .....	83
7.3.2 Funcionamiento de la interfaz gráfica de la simulación 2 .....	86
7.3.3 Funcionamiento de la interfaz gráfica de la simulación 3 .....	88
7.3.4 Funcionamiento de la interfaz gráfica de la simulación 4 .....	90
7.3.5 Funcionamiento de la interfaz gráfica de la simulación 5 .....	92
7.3.6 Funcionamiento de la interfaz gráfica de la simulación 6 .....	94
7.4 Glosario .....	96

## LISTA DE TABLAS

### CAPITULO III

#### MARCO TEORICO

Tabla 3. 1: Especificaciones de las sub-matrices cíclicas. ....	29
Tabla 3. 2: Especificaciones de las sub-matrices de la matriz generadora. ....	32

### CAPITULO IV

#### METODOLOGIA

Tabla 4. 1: Valores de los campos de la trama para la simulación 1. ....	41
Tabla 4. 2: Valores de media y varianza suministrados a la herramienta MATLAB para hacer uso del ruido gaussiano en la simulación 1. ....	42
Tabla 4. 3: Valores suministrados a la herramienta MATLAB para hacer uso del ruido salt & pepper en la simulación 1. ....	42
Tabla 4. 4: Valores suministrados a la herramienta MATLAB para hacer uso del ruido speckle en la simulación 1. ....	43
Tabla 4. 5: Fuentes de información para la simulación 2 de transmisión de letras codificados y decodificados con el código LDPC. ....	43
Tabla 4. 6: Alfabeto de las letras y el espacio entre ellas en la simulación 4. ....	47
Tabla 4. 7: Trama utilizada para la simulación 5 y 6 del bloque de aleatorización. ....	50

### CAPITULO V

#### RESULTADOS

Tabla 5. 1: Degradación de la imagen obtenida luego de ser sometida a distintos valores para una fuente de ruido tipo gaussiana. ....	52
Tabla 5. 2: Cantidad de bits errados en cada una de las muestras de la imagen afectada por distintos valores de ruido tipo gaussiano. ....	53

Tabla 5. 3: Degradación de la imagen obtenida luego de ser sometida a distintos valores para una fuente de ruido tipo salt & pepper. ....	55
Tabla 5. 4: Cantidad de bits errados en cada una de las muestras de la imagen afectada por distintos valores de ruido tipo salt & pepper. ....	56
Tabla 5. 5: Degradación de la imagen obtenida luego de ser sometida a distintos valores para una fuente de ruido tipo speckle.....	58
Tabla 5. 6: Cantidad de bits errados en cada una de las muestras de la imagen afectada por distintos valores de ruido tipo speckle.....	59
Tabla 5. 7: Resultados obtenidos para la transmisión de una o dos letras. ....	61
Tabla 5. 8: Resultados de la imagen enviada para distintos valores de ruido gaussiano. ....	67
Tabla 5. 9: Comportamiento de 10 iteraciones en la simulación 3 para distintos valores de ruido gaussiano.....	69
Tabla 5. 10: Resultados del mensaje enviado para distintos valores de ruido gaussiano. ....	71
Tabla 5. 11: Bloque de entrada y de salida para la simulación 5 del bloque de aleatorización de la figura 4.12. ....	73
Tabla 5. 12: Bloque de entrada y de salida para la simulación 6 del bloque de aleatorización de la figura 4.13. ....	74

## LISTA DE FIGURAS

### CAPITULO III

#### MARCO TEORICO

Figura 3. 1: Diagrama de bloques de un sistema de comunicación digital. ....	7
Figura 3. 2: Formas del ruido eléctrico. ....	12
Figura 3. 3: Composición trama estándar AOS. ....	16
Figura 3. 4: Patrón asignado a la Sincronización. ....	17
Figura 3. 5: Diagrama lógico del pseudo-aleatorizador. ....	21
Figura 3. 6: Matriz LDPC regular (3,4) de orden 12x16. ....	25
Figura 3. 7: Matriz de paridad usada en el gráfico de Tanner de la figura 3.8. ....	26
Figura 3. 8: Gráfico de Tanner de la figura 3.7. ....	27
Figura 3. 9: Ejemplo de una sub-matriz cíclica 15x15. ....	28
Figura 3. 10: Matriz de paridad base para el código LDPC (8176,7156). ....	29
Figura 3. 11: Gráfico de dispersión de la matriz de comprobación de paridad. ....	31
Figura 3. 12: Matriz generadora para el código LDPC (8176,7156). ....	32
Figura 3. 13: Palabra de código abreviada. ....	35
Figura 3. 14: Diagrama de flujo de los algoritmos de decodificación. ....	36
Figura 3. 15: Gráfico de Tanner con ubicación de coeficientes. ....	37

### CAPITULO IV

#### METODOLOGIA

Figura 4. 1: Fuente de información para la simulación 1 de la trama del estándar AOS codificada por el código LDPC. ....	41
Figura 4. 2: Matriz de paridad para la transmisión de una letra, simulación 2. ....	44
Figura 4. 3: Matriz generadora para la transmisión de una letra, simulación 2. ....	44
Figura 4. 4: Gráfico de Tanner de la figura 4.2, simulación 2. ....	45
Figura 4. 5: Matriz de paridad para la transmisión de dos letras, simulación 2. ....	45
Figura 4. 6: Matriz generadora para la transmisión de dos letras, simulación 2. ....	45
Figura 4. 7: Gráfico de Tanner de la figura 4.5, simulación 2. ....	46

Figura 4. 8: Fuente de información de la simulación 3 de transmisión de una imagen. .....	47
Figura 4. 9: Matriz de paridad de la simulación 4 de envío de un mensaje. ....	48
Figura 4. 10: Matriz generadora de la simulación 4 de envío de un mensaje. ....	49
Figura 4. 11: Gráfico de Tanner de la figura 4.9, simulación 4. ....	49
Figura 4. 12: Bloque de aleatorización de la ecuación 4.17, simulación 5. ....	50
Figura 4. 13: Bloque de aleatorización de la ecuación 4.18, simulación 6. ....	50

## CAPITULO V

### RESULTADOS

Figura 5. 1: Promedio de bits errados para distintos valores de ruido tipo gaussiano.	54
Figura 5. 2: Promedio de bits errados para distintos valores de ruido tipo salt & pepper. .....	57
Figura 5. 3: Promedio de bits errados para distintos valores de ruido tipo speckle. ....	60
Figura 5. 4: Resultados de los nodos de chequeo y nodos variables para la letra enviada A. ....	62
Figura 5. 5: Resultados de los nodos de chequeo y nodos variables para la letra enviada D. ....	63
Figura 5. 6: Resultados de los nodos de chequeo y nodos variables para la letra enviada EC. ....	64
Figura 5. 7: Resultados de los nodos de chequeo y nodos variables para la letra enviada BF. ....	65
Figura 5. 8: Modulación QPSK de la entrada en el bloque de aleatorización. ....	73
Figura 5. 9: Modulación QPSK de la salida de la simulación 5 en el bloque de aleatorización. ....	74
Figura 5. 10: Modulación QPSK de la salida de la simulación 6 en el bloque de aleatorización. ....	75

## CAPITULO VII

## MATERIALES DE REFERENCIA

Figura 7. 1: Simulación 1, consola línea 229 modificación tipo de ruido. ....	83
Figura 7. 2: Simulación 1, consola línea 229 modificación valor del ruido. ....	84
Figura 7. 3: Simulación 1, respuesta para un tipo y valor de ruido determinado.....	85
Figura 7. 4: Simulación 2, interfaz gráfica.....	86
Figura 7. 5: Simulación 2, interfaz gráfica con respuesta a una entrada determinada.	87
Figura 7. 6: Simulación 3, interfaz gráfica.....	88
Figura 7. 7: Simulación 3, interfaz gráfica con respuesta a una entrada determinada.	89
Figura 7. 8: Simulación 4, interfaz gráfica.....	90
Figura 7. 9: Simulación 4, interfaz gráfica con respuesta a una entrada determinada.	91
Figura 7. 10: Simulación 5, interfaz gráfica.....	92
Figura 7. 11 Simulación 5, interfaz gráfica con respuesta a una entrada determinada. .....	93
Figura 7. 12: Simulación 6, interfaz gráfica.....	94
Figura 7. 13: Simulación 6, interfaz gráfica con respuesta a una entrada determinada. .....	95

## **CAPITULO I**

### **INTRODUCCION**

El ser humano ha estado en una búsqueda constante por satisfacer mejor su necesidad de comunicarse lo cual ha generado la creación de diversos sistemas de comunicación, por lo que a medida que la tecnología evoluciona es necesario mejorar los sistemas de transmisión. Hoy en día es importante tener satélites como medios de enlaces entre dos puntos de comunicación ya que la distancia no es excusa para estar incomunicado; todo esto por la misma tendencia natural de los seres humanos al dinamismo, por la necesidad de estar físicamente en varios sitios del mundo y poder estar comunicado o explorar un sitio totalmente lejano.

En los sistemas digitales de comunicaciones existen dos causas principales que influyen en el deterioro de la señal recibida, la primera es el ruido introducido por el propio canal de comunicaciones y la segunda es el ruido de cuantificación el cual es introducido en el transmisor y es transportado por todo el sistema hasta la salida del receptor. El ruido limita la capacidad de comunicarse, ya que de no existir podrían enviarse mensajes electrónicamente a los confines del universo, lo que nos lleva a querer proteger la información al máximo contra el ruido. En los sistemas satelitales las grandes distancias en las cuales se producen las comunicaciones generan un deterioro en la señal significativo por lo cual se implementan distintos mecanismos que nos permiten alcanzar grandes distancias para la comunicación.

La codificación del canal detecta y corrige los errores producidos como consecuencia del ruido y la distorsión introducida por el medio de propagación y/o las no linealidades en el propio sistema de transmisión. Los esquemas de detección y corrección de errores juegan un papel importante en los sistemas de comunicación actuales ya que permiten incrementar las tasas de transmisión de datos y además dan al sistema fiabilidad y

robustez, haciendo que los requisitos de potencia transmitida sean menores y reduciendo el número de retransmisiones.

El método para detectar y corregir errores consiste en incluir en los bloques de datos transmitidos bits adicionales, denominados redundancia, generando bloques de datos denominados palabras de código. Las palabras de código son suficientemente distintas unas de otras, tanto que el mensaje transmitido puede ser deducido correctamente en el receptor, incluso si algunos bits de la palabra de código se dañan durante la transmisión sobre el canal.

El código LDPC (Low Density Parity Check), Comprobación de Paridad de Baja Densidad, es un detector de errores el cual se caracteriza por tener una matriz generadora binaria que contiene un alto número de ceros y por lo tanto un número bajo de unos, ocasionando que todas las palabras de código que se puedan formar a partir de la misma matriz generadora se encuentren a la misma distancia unas de otras.

La organización internacional para la estandarización (*ISO*), en el protocolo definido como ISO/IEC 7498-1, propone utilizar un diseño estructurado para la transmisión de datos ya que de esta manera se reduce la complejidad del desarrollo, estandariza interfaces, facilita la técnica modular, asegura la interoperabilidad de la tecnología y acelera la evolución del sistema.

El diseño estructurado a utilizar en este trabajo de grado es el especificado en la ISO 22666:2007 denominado Sistema Avanzado en Órbita (*Advanced Orbiting Systems*), este estándar es empleado para transmisiones que involucren comunicación espacio-espacio, tierra-espacio o espacio-tierra.

## CAPITULO II

### FUNDAMENTO DE LA INVESTIGACION

#### *2.1 Planteamiento del problema*

El medio de transmisión constituye el canal que permite la transmisión de información entre dos terminales de un sistema, dichas transmisiones se realizan empleando ondas electromagnéticas que se propagan a través de un medio de transmisión, el cual puede ser guiado o no guiado, para el caso a evaluar consiste en un medio no guiado ya que se enfoca en la comunicación de los sistemas satelitales.

Las condiciones atmosféricas, pueden ocasionar desvanecimientos intensos y desviaciones del haz, lo que nos lleva a querer proteger la información transmitida de tal modo, que pueda ser recuperada o reconstruida en el receptor aunque una parte de ella se pierda o destruya durante el proceso de transmisión.

El ruido y la distorsión producidos en el canal pueden ser detectados y corregidos mediante la codificación del mismo, ya que la codificación del canal estudia las técnicas y procedimientos para llevar a cabo dicha tarea.

Los códigos de comprobación de paridad de baja densidad, LDPC, están diseñados para implementarse a través de canales que se encuentren limitados por la presencia de ruido y su eficiencia se acerca al límite establecido por Shannon.

Las características de dicho código nos hacen querer evaluar en este trabajo especial de grado el funcionamiento del mismo bajo el estándar AOS mediante la simulación de la codificación del canal con la finalidad de que se puedan incrementar las tasas de eficiencia en los códigos a desarrollarse en un futuro.

## ***2.2 Antecedentes***

En investigaciones y trabajos previos que sirven como apoyo se encuentran los siguientes:

- El trabajo final de grado realizado por Ferrán Mascarell Catala en la Universitat Politècnica de Valencia en España, quien diseñó e implemento en FPGA un decodificador de códigos LDPC con secuenciación por capas, dirigido por los directores Javier Valls Coquillat y María José Canet Subiela. Junio 2013. [1]
- El proyecto de grado realizado por Alix Lheritier en la Universidad de la Republica del Uruguay, quien estudió los códigos de paridad de baja densidad, dirigido por el Dr. Alfredo Viola. [2]
- El proyecto de grado realizado por María Belén Tipantuña Toapanta en la Universidad de las Fuerzas Armadas de Ecuador, quien evaluó el desempeño de los códigos de comprobación de paridad de baja densidad en transmisión de imágenes, dirigido por el Ing. Daniel Altamirano. Abril 2014. [3]

## ***2.3 Objetivos***

### **2.3.1 Objetivo General**

Simular el código LDPC del estándar AOS en sistemas de transmisión de datos satelital.

### **2.3.2 Objetivos Específicos**

- Generar el formato de trama codificado bajo el código LDPC partiendo de una fuente de información conocida.

- Evaluar la correspondencia del resultado del algoritmo a nivel de decodificación con fines de recuperar la información de la fuente.
- Analizar el efecto de fuentes ruidosas sobre la eficiencia del código LDPC.
- Desarrollar una simulación que permita visualizar la operatividad del código.
- Generar escenarios de ejemplos demostrativos donde se observe la eficiencia del código.

## **CAPITULO III**

### **MARCO TEORICO**

#### ***3.1 La Comunicación Digital***

La comunicación es aquella acción mediante la cual se transfiere información de un punto a otro a través de un medio de transmisión considerado canal, dependiendo de la información que se desea transmitir la comunicación se puede realizar analógica o digital.

En la comunicación analógica se transmiten mensajes que pertenecen a un conjunto infinito y continuo de valores lo cual hace que esta sea muy sensible a cualquier perturbación que se interponga.

En la comunicación digital los mensajes pertenecen a un conjunto finito y discreto de valores haciendo que este sea menos sensible a los ruidos que se interpongan durante la transmisión.

La comunicación digital surge por la evolución de las tecnologías en el campo de las telecomunicaciones, lo cual permite una serie de ventajas respecto a los sistemas de comunicación analógica.

Los sistemas de comunicación digital poseen un conjunto de subsistemas y mecanismos que permiten la comunicación entre el transmisor y el receptor. En la figura 3.1 podemos observar un diagrama de bloques que define el proceso para la transmisión y recepción de información.

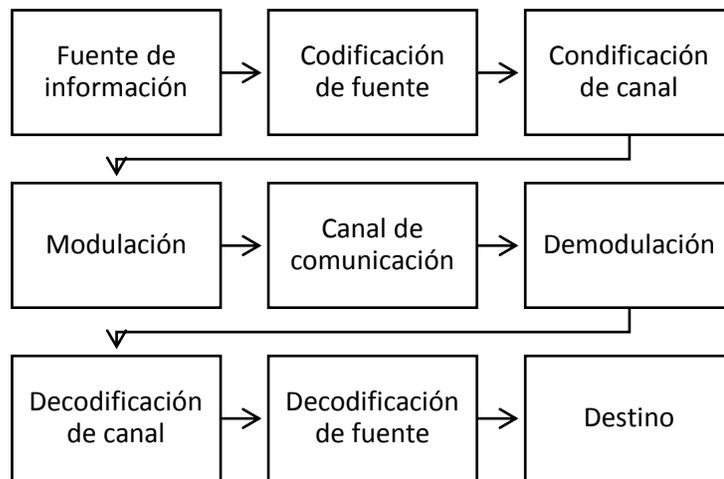


Figura 3. 1: Diagrama de bloques de un sistema de comunicación digital.

La codificación de fuente, codificación de canal y la modulación se realiza por parte del transmisor mientras que la demodulación, decodificación de canal y decodificación de fuente son procesos realizados por el receptor.

A continuación se realiza una descripción acerca de cada uno de los procesos que se realizan en un sistema de comunicación digital. [11]

### 3.1.1 Fuentes de información

Las fuentes de información pueden clasificarse como analógicas y discretas. Una fuente de información analógica es aquella que emite una o más señales de amplitud continua, en cambio una fuente de información discreta consiste en una secuencia de símbolos discreto o letras. Una fuente de información analógica puede transformarse en una fuente de información discreta a través del proceso de muestreo y cuantificación.

Conociendo el alfabeto utilizado en la fuente, la velocidad del símbolo, las probabilidades del alfabeto y la dependencia probabilística de los símbolos en una secuencia podemos construir un modelo probabilístico de la fuente de información y

definir la entropía, denotada por la letra  $H$ , y la tasa de información en la fuente, denotada por la letra  $R$ .

La entropía de una fuente se refiere al contenido de información promedio por símbolo en un mensaje largo.

La tasa de información representa el número mínimo de bits por segundo necesarios para representar la información que sale de la fuente discreta, también representa la tasa de datos promedio mínima necesaria para obtener la información desde la fuente hasta el destino.

### 3.1.2 Codificación y decodificación de fuente

#### 3.1.2.1 Codificación de fuente

El codificador de fuente convierte la secuencia de símbolos de entrada en una secuencia binaria de “0” y “1”. Esta operación puede realizarse asignando una palabra de código binario de longitud fija a cada símbolo en la secuencia de entrada, esta operación solo es eficiente si los símbolos tienen probabilidades iguales. En la mayoría de las situaciones los símbolos de una secuencia ocurren con probabilidades desiguales por lo cual el codificador de fuente toma una cadena de dos o más símbolos como un bloque y le asigna una palabra de código de longitud variable a dichos bloques.

Un codificador de fuente óptimo debe producir una velocidad de datos de salida que se aproxime a  $R$ , la tasa de información de la fuente.

#### 3.1.2.2 Decodificación de fuente

El decodificador de fuente convierte la salida binaria del decodificador de canal en una secuencia de símbolos.

Para un sistema donde se utilicen palabras de código de longitud variable el decodificador será más complejo ya que debe ser capaz de hacer frente a la pérdida de memoria cada vez mayor y la pérdida de sincronización debido a los errores de bit.

### 3.1.3 Codificación y decodificación de canal

#### 3.1.3.1 Codificación de canal

La codificación de canal es un método práctico que brinda alta confiabilidad y eficiencia de transmisión. El control de errores se puede lograr con la codificación del canal, este proceso consiste en la adición sistemática de bits adicionales a la salida del codificador de fuente, estos bits adicionales no transmiten ninguna información su función es que el receptor pueda detectar y corregir algunos de los errores en la transmisión de la información.

Existen varios métodos para realizar la codificación del canal, los cuales requieren de almacenamiento y procesamiento de datos binarios en el codificador y decodificador de canal.

#### 3.1.3.2 Decodificación de canal

El decodificador de canal recupera los bits de información de la palabra codificada, además de detectar y posiblemente corregir errores.

Los parámetros de importancia para esta etapa son básicamente el método de codificación de canal, la velocidad o eficiencia del codificador y la capacidad de control de errores.

### 3.1.4 Modulación y demodulación digital

#### 3.1.4.1 Modulación digital

El modulador recibe una corriente de bits como entrada y la convierte en una forma de onda eléctrica adecuada para la transmisión a través del canal de comunicación. Esta técnica puede utilizarse para minimizar los efectos del ruido en el canal, para proporcionar la capacidad de multiplexar muchas señales y para superar algunas limitaciones del equipo.

Este proceso requiere el uso de mayor potencia en la señal y mayor ancho de banda, los cuales son parámetros que no pueden aumentarse indefinidamente ya que se encuentran limitados por el equipo y el canal, por lo cual son usadas formas de onda de mayor duración.

#### 3.1.4.2 Demodulación digital

El demodulador realiza la extracción del mensaje de la forma de onda producida por el modulador, por lo cual el método de modulación es un parámetro importante en esta etapa.

### 3.1.5 Canal de comunicaciones

El canal de comunicación proporciona la conexión eléctrica entre la fuente y el destino, dicha conexión podría ser un cable de cobre, un cable de fibra óptica o el espacio libre. Los canales de comunicación solo tienen un ancho de banda finito por lo cual la señal portadora de información sufre frecuentemente distorsiones de amplitud y fase, además de que la potencia de la señal también disminuye debido a la atenuación del canal.

Un parámetro importante de los canales de comunicación es la relación potencia de señal a ruido, ya que una de las formas en que los efectos del ruido pueden minimizarse es aumentando la potencia de la señal. Otros parámetros importantes de los canales de comunicación son el ancho de banda utilizable, la respuesta de amplitud y fase y las propiedades del ruido en el canal.

### ***3.2 Ruido***

Se le conoce con el nombre de ruido a todas aquellas perturbaciones en un sistema de comunicaciones que provocan que la información no llegue con claridad o se distorsione.

El ruido se debe a múltiples causas tales como los componentes electrónicos, el ruido térmico de las resistencias, interferencia de señales externas, etc. Es imposible eliminar totalmente el ruido ya que los componentes electrónicos no son perfectos. Sin embargo, es posible limitar su valor de manera que la calidad de la comunicación resulte aceptable.

En la comunicación de datos, los errores que son causados por el ruido se manifiestan como bits adicionales o faltantes por lo cual es habitual introducir cierta redundancia en la codificación del mensaje, ya que la redundancia es la parte del mensaje que podría omitirse sin que se produzca pérdida de información.

A continuación una breve descripción de los tipos de ruidos. [27]

### 3.2.1 Tipos de ruido

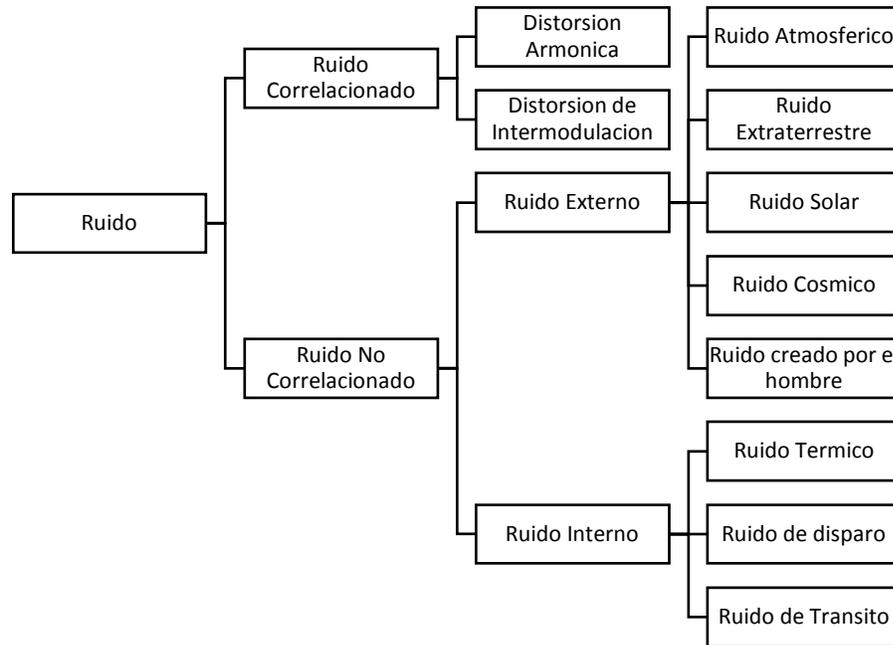


Figura 3. 2: Formas del ruido eléctrico.

**3.2.1.1 Ruido Correlacionado:** Tiene una relación directa con la señal, ya que es producido por amplificaciones no lineales de la misma., esta categoría se divide en: distorsión armónica y distorsión de intermodulación.

**3.2.1.1.1 Distorsión Armónica:** Se define como la producción de armónicos de una señal originada por una mezcla no lineal, los armónicos son múltiplos enteros de la señal original de entrada.

**3.2.1.1.2 Distorsión de Intermodulación:** Se refiere a la generación indeseable de productos cruzados que son la suma o restas de frecuencia.

**3.2.1.2 Ruido No Correlacionado:** Se encuentra presente exista una señal presente o no, este ruido se divide en dos categorías: ruido externo y ruido interno.

**3.2.1.2.1 Ruido Externo:** Se genera fuera de los sistemas de comunicaciones, este ruido se toma en cuenta solo si sus frecuencias caen dentro de la banda útil del sistema de comunicaciones.

**3.2.1.2.1.1 Ruido Atmosférico:** Es producido por la estática que se encuentra dentro de la atmosfera terrestre la cual se manifiesta habitualmente en forma de relámpagos, centellas, rayos, etc.

**3.2.1.2.1.2 Ruido Extraterrestre:** Es originado fuera de la atmosfera, principalmente en la vía láctea, otras galaxias y el sol.

**3.2.1.2.1.3 Ruido Solar:** Es producido directamente del calor del sol.

**3.2.1.2.1.4 Ruido Cósmico:** Es producido por fuentes de radiofrecuencia naturales aleatoriamente distribuidas por el universo aunque debido a la lejanía de las formaciones galácticas es de una intensidad muy baja.

**3.2.1.2.1.5 Ruido creado por el hombre:** Es producido por las chispas de automóviles, los conmutadores de motores eléctricos y equipos de conmutación de potencia.

**3.2.1.2.2 Ruido Interno:** Es la interferencia eléctrica que se genera dentro de un dispositivo los cuales pueden ser tanto pasivos (resistores y cables) como activos (diodos, transistores).

**3.2.1.2.2.1 Ruido Térmico:** Es producido por el movimiento aleatorio de los electrones en un conductor debido al calor.

**3.2.1.2.2.2 Ruido de disparo:** Es producido por la llegada aleatoria de portadoras en el elemento de salida de un dispositivo electrónico.

**3.2.1.2.2.3 Ruido de Transito:** Es producido por cualquier modificación a una corriente de portadoras conforme pasa desde la entrada hasta la salida de un dispositivo.

### 3.2.2 Ruidos del bloque imnoise de la herramienta MATLAB [24]

**3.2.2.1 Gaussian:** Este ruido es aditivo y la señal es independiente, tal como se muestra en la ecuación 3.1, donde  $g(z, y)$  es el resultado de la distorsion de la señal original  $f(z, y)$  por  $n(z, y)$ , la distribución uniforme del ruido aleatorio con media 0 y con varianza  $\sigma^2$ .

$$g(z, y) = f(z, y) + n(z, y) \quad (3.1)$$

**3.2.2.2 Salt & Pepper:** Este tipo de ruido es impulsivo y es ocasionado frecuentemente por los sensores o por la transmisión digital. Este ruido puede ser modelado con la ecuación 3.2, donde  $i(z, y)$  es el ruido impulsivo y  $d$  es la densidad del ruido.

$$g(z, y) = (1 - p)f(z, y) + di(z, y) \quad (3.2)$$

**3.2.2.3 Speckle:** Este tipo de ruido es multiplicativo y su aspecto es granulado tal como las imágenes de radares y ecografías. Este ruido puede ser modelado por la ecuación 3.3, donde  $m(z, y)$  es el ruido multiplicativo.

$$g(z, y) = f(z, y)m(z, y) \quad (3.3)$$

La señal obtenida es el resultado de la multiplicación de dos señales,  $m(z, y)$  es la distribución uniforme del ruido aleatorio con media 0 y con varianza  $\sigma^2$ .

### **3.3 Estándar AOS**

La organización internacional de normalización (ISO) plantea el estándar ISO 22666:2007 el cual especifica el estándar AOS (*Advanced Orbiting Systems*) este corresponde a un protocolo de enlace de datos especificado en el protocolo ISO/IEC 7498-1, dicho estándar es utilizado para las comunicaciones espacio-tierra, tierra-espacio y espacio-espacio en misiones espaciales, el cual proporciona funciones de transferencia de varios datos utilizando una unidad de datos de protocolo de longitud fija llamado marco de transferencia.

Este protocolo es utilizado para la sincronización de la trama con el canal, ya que cada trama contiene un encabezado que proporciona información de control del protocolo y un campo de datos de longitud fija integrada por las unidades de datos del servicio.

La instalación de un canal virtual permite que un canal físico se reparta entre múltiples flujos de datos, que como consecuencia cada uno de los canales virtuales puede tener diferentes requisitos de servicio. Cada trama transferida a través de un canal físico pertenece a uno de los canales virtuales.

La trama debe tener una longitud constante a lo largo de una misión específica ya que un cambio de longitud de la misma puede dar lugar a una pérdida de sincronización en el receptor.

A continuación se explican las funciones generales del estándar AOS. [5]

#### **3.3.1 Funciones generales**

- Generación y procesamiento de información de control de protocolo para llevar a cabo la identificación de datos, detección de pérdidas y errores.
- Segmentación de unidades de datos de servicio para transferir datos de servicio de longitud variable en unidades de datos de protocolo de longitud fija.
- Multiplexación/demultiplexación y la conmutación/deconmutación para que los distintos usuarios del servicio puedan compartir un único canal físico.
- Generación y eliminación de los datos de inactividad para la transferencia de unidades de datos de protocolo a una velocidad constante.

### 3.3.2 Composición de la trama [2]

Definición	Sincronización	Encabezado Primario						Zona de Unidad de Datos		Codificación LDPC (8160,7136)
		Número de Versión	Identificador		Contador de Unidades de Datos del Canal Virtual	Campo de Señalización		Cabecera	Unidad de Datos de Flujo de Bits	
			Identificador Espacial	Identificador Canal Virtual		Campo de Repetición	Repuestos Reservados			
<b>Bit</b>	32	2	8	6	24	2	6	16	7072	1024
<b>Octeto</b>	4	6						2	884	128

Figura 3. 3: Composición trama estándar AOS.

**3.3.2.1 Sincronización:** Esta sección es esencial para el flujo de datos de telemetría ya que las siguientes acciones dependen de ella:



además de proporcionar el dominio de nomenclatura para los canales virtuales.

La convención de nomenclatura para los dominios es comunicación espacio-tierra, comunicación tierra-espacio y comunicación espacio-espacio.

**3.3.2.2.2 Identificador Canal Virtual:** Este campo se encarga de identificar el canal virtual en uso, este permite hasta 64 canales virtuales ejecutándose simultáneamente.

**3.3.2.2.3 Contador de Unidades de Datos del Canal Virtual:** El propósito de este campo es proporcionar responsabilidad individual en cada uno de los 64 canales virtuales. Este campo representa un recuento secuencial del número total de unidades de datos del canal virtual que han sido transmitidos en cada uno de los canales virtuales, se utiliza en asociación con el identificador canal virtual para así lograr mantener un contador separado para cada canal virtual.

**3.3.2.2.4 Campo de Señalización:** Este campo es utilizado para alertar al receptor de las unidades de datos del canal virtual con respecto a funciones que pueden cambiar más rápidamente de lo que puede ser manejado por la administración, proporcionar una verificación cruzada contra manual o configuraciones automatizadas, esta señalización se divide en dos sub campos:

**3.3.2.2.4.1 Campo de Repetición:** Este campo se encarga de almacenar las unidades de datos del canal virtual durante los periodos en que el

canal físico no está disponible, y así poder recuperarlos para su posterior reproducción cuando el canal se reestablezca.

Su propósito principal es el de discriminar entre unidades de datos de canal virtual en tiempo real y reproducir la transmisión en un canal virtual en particular.

La combinación de los 2 bits correspondientes para esta sección se interpreta de la siguiente manera: “00” para la transmisión es en tiempo real y “11” para cuando se está repitiendo la transmisión.

**3.3.2.2.4.2 Repuestos Reservados:** Este campo esta reservado para posibles aplicaciones futuras de señalización, por convección este campo se establece en una secuencia de 6 bits “0”.

**3.3.2.3 Zona de Unidad de Datos:** La zona de unidad de datos del canal virtual debe ser un número entero de octetos, este campo se divide en los siguientes dos sub campos:

**3.3.2.3.1 Cabecera:** La utilización de los 16 bits dentro de la cabecera es la siguiente:

**3.3.2.3.1.1 Repuesto:** Actualmente este campo no se encuentra definido por lo cual se establecen los 2 bits asignados a este campo como “00”.

**3.3.2.3.1.2 Puntero de Datos de Flujo de Bits:** Este campo de 14 bits contiene una secuencia binaria que cuando se incrementa en 1 indica que los datos transmitidos son válidos, de ser inválidos los datos transmitidos se establece una secuencia binaria de 14 bits “1”. Si los datos son parcialmente validos este campo se coloca en la misma secuencia binaria de la data valida menos 1.

**3.3.2.3.2 Unidad de Datos de Flujo de Bits:** Este campo contiene un bloque de longitud fija de un flujo de bits de datos de un usuario o una longitud fija patrón de relleno especificado por el proyecto.

**3.3.2.4 Codificación LDPC (8160,7136):** El codificador aceptara como entrada una trama 7136 bits, es decir 892 octetos que concuerden con la longitud y la dimensión de la tasa del código 255/223. Se agregaran 18 bits “0” al principio del mensaje de 7136 bits a codificar, produciendo un vector de 7154 elementos, este vector se multiplicará por la matriz generadora produciendo un vector de 8176 elementos que consta de 18 ceros de relleno, 7136 bits del mensaje y 1022 bits de paridad, de dicho vector se descartan los 18 ceros de relleno inicial y se agregaran 2 ceros al final de la palabra con lo cual se producirá una palabra de código de 8160 bits para ser enviada por el canal.

### 3.3.3 Aleatorización [7]

Este proceso es realizado por un pseudo-aleatorizador para garantizar que la palabra de código sea lo suficientemente aleatoria, esto con la finalidad de poseer un correcto funcionamiento en el receptor.

El código LDPC (8160,7136) no garantiza suficientes transiciones de símbolo por lo cual se debe utilizar un pseudo-aleatorizador después de la codificación.

El pseudo-aleatorizador generara una secuencia de datos a partir de un polinomio determinado, el cual podría ser el siguiente:

$$g(x) = x^8 + x^7 + x^5 + x^3 + 1 \quad (3.4)$$

El polinomio de la ecuación 3.4 nos indica que el tamaño del pseudo-aleatorizador es de 8 bits, el funcionamiento del mismo se expresa en la figura 3.5.

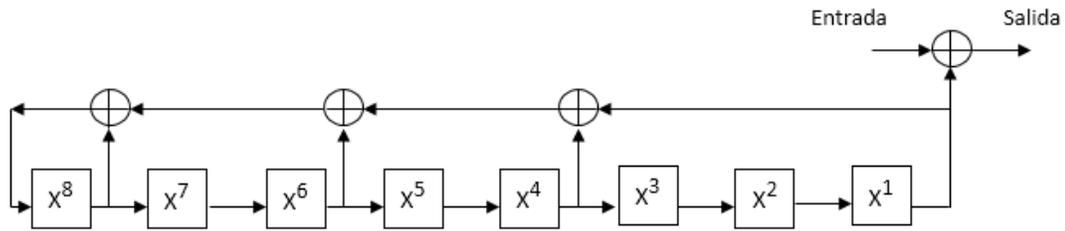


Figura 3. 5: Diagrama lógico del pseudo-aleatorizador.

El pseudo-aleatorizador comienza inicializado en uno todos los bloques, su funcionamiento detallado en la figura 3.5 nos indica que él recibe un bit en la entrada el cual se va sumando, mediante una compuerta XOR, lo que se encuentre en el bloque 3, luego ese resultado nuevamente se suma con el bloque 5 y por último ese resultado se suma con el bloque 7 este resultado va a ocupar el lugar del bloque 8 ocasionando que todos los otros bloques se corran un espacio y mandando a la salida lo que se encuentre en el bloque 1.

### 3.4 Códigos

Los sistemas de comunicaciones se encargan de enviar un mensaje desde un punto a otro mediante un canal, la transmisión confiable del mensaje depende del canal y el ruido externo que pueda interferir o distorsionar la señal, por lo cual es necesario proteger la información transmitida de tal modo que pueda ser recuperada o reconstruida en el receptor aunque una parte de ella se pierda o destruya durante el proceso de transmisión.

Los códigos de control de errores se dividen a menudo en dos categorías: códigos de bloque y códigos convolucionales. En los códigos de bloque, un bloque de  $K$  bits de información es seguido por un grupo de bits de comprobación los cuales se utilizan para verificar los bits de información en el bloque de información. En los códigos convolucionales, los bits de comprobación se entrelazan continuamente con los bits de

información, dichos bits verifican los bits de información no solo en el bloque que los precede inmediatamente, sino en otro bloque también.

A continuación se explicara los códigos de bloques lineales con la finalidad de más adelante explicar el código a usar en este trabajo especial de grado. [17]

### 3.4.1 Códigos de bloques lineales

En los códigos de bloques lineales se agregan bits adicionales a los datos transmitidos con la finalidad de detectar y corregir errores, dichos datos se segmentan en bloques de longitud fija  $K$ , los cuales pueden expresarse como una combinación lineal de los otros datos transmitidos. El codificador de canal transforma el segmento de entrada en un bloque de salida de longitud  $N$  el cual debe ser mayor que  $K$  para así proporcionar la redundancia necesaria para la corrección y detección de errores.

La tasa del código se expresa como:

$$R = \frac{K}{N} \quad (3.5)$$

Para la ecuación 3.5  $K$  es la longitud de los bloques de entrada,  $N$  es la longitud de los bloques de salida y  $R$  es la tasa del código la cual representa la fracción de información en el bloque de salida.

Si tenemos  $K$  bits en un mensaje de entrada existen  $2^K$  combinaciones de entrada. Almacenar toda esta información se logra generando una matriz  $G$  que convierte las entradas en palabras de código, dicha matriz posee  $K$  vectores de longitud  $N$ . Las palabras de código son los mensajes de salida con longitud  $N$  los cuales poseen bits asociados al mensaje de entrada.

El codificador de canal genera las palabras de código multiplicando el vector de entrada por la matriz  $G$ .

$$c = uG \quad (3.6)$$

En la ecuación 3.6  $G$  es la matriz generadora de palabras de código denotada por la letra  $c$  y  $u$  es el vector de entrada.

A partir de la matriz  $G$  se deriva una matriz de paridad  $H$  con la cual se corrigen las perdidas y los errores de transmisión de la palabra de código.

$$GH^T = 0 \quad (3.7)$$

En la ecuación 3.7  $G$  es la matriz generadora y  $H$  es la matriz de paridad, dicha igualdad debe cumplirse para que el transmisor y receptor trabajen correctamente.

Para que la palabra de código sea decodificada de manera correcta se deben cumplir las siguientes igualdades:

$$yH^T = 0 \quad (3.8)$$

$$uGH^T = 0 \quad (3.9)$$

En la ecuación 3.8  $y$  es el vector recibido el cual se multiplicara por la matriz de paridad  $H$ , esta igualdad representa que el mensaje fue recibido sin ningún error o perdida. En la ecuación 3.9  $u$  es el vector de entrada,  $G$  es la matriz generadora y  $H$  es la matriz de paridad.

Para corregir los errores en este tipo de códigos se procede a calcular la distancia Hamming, es decir el número de bits que son diferentes, entre el vector recibido en el decodificador y los  $2^K$  vectores validos del código, y se decide que se ha transmitido aquel cuya distancia de Hamming con el recibido es menor.

#### 3.4.2 Códigos de paridad de baja densidad

En 1963 Robert G. Gallager invento los códigos LDPC (*Low Density Parity Check*) los cuales son técnicas y procedimientos para proteger el canal. Al principio no se tomaron en cuenta ya que se requería de una alta capacidad de cómputo para su decodificación, en dicha época la capacidad de los ordenadores era bastante limitada por lo cual no fue

hasta la época de los 90 cuando MacKay y Nel trajeron de nuevo el uso de estos códigos y a partir de ellos fueron surgiendo distintos trabajos relacionados con el tema.

Este tipo de códigos se caracteriza por poseer una matriz generadora con un alto número de 0 y por lo tanto un número bajo de 1. Estos códigos tienen la particularidad de que se forman, a partir de la misma matriz generadora, distintas palabras de código encontrándose a la misma distancia una de otras. [16]

La codificación en los códigos LDPC es bastante similar a la de cualquier código de bloques su diferencia está en la forma en que se decodifican ya que los códigos de bloques son generalmente decodificados con algoritmos que tienden a ser cortos y diseñados algebraicamente lo cual hace la tarea menos compleja, sin embargo los códigos LDPC se decodifican de manera iterativa usando una representación gráfica de su matriz de paridad.

A continuación se realiza una descripción a cerca de los métodos de representación de los códigos LDPC. [23]

### **3.4.2.1 Representación de los códigos LDPC**

Todos los códigos de bloque pueden ser descritos de forma matricial pero en el caso de los códigos LDPC también es posible la representación gráfica, es por ello que a continuación se hace una descripción de ambos métodos de representación.

#### **3.4.2.1.1 Representación matricial**

Una matriz LDPC regular  $(i, j)$  es una matriz binaria de orden  $M \times N$  que tiene  $i$  unos en cada columna y  $j$  unos en cada fila con la condición de que:

$$N > M \quad (3.10)$$

La ecuación 3.10 es necesaria para asegurar que todas las palabras de código nulas satisfagan todas las restricciones.

A su vez se definen  $W_f$  y  $W_c$  como dos magnitudes que describen esta matriz, cada una representa la cantidad de “1” en la fila o columna respectivamente. La ecuación 3.11 y 3.12 deben cumplirse para considerar que la matriz sea de baja densidad:

$$W_f \ll M \quad (3.11)$$

$$W_c \ll N \quad (3.12)$$

La tasa del código, fracción de bits de información en la palabra de código, se expresa en la ecuación 3.13 mostrada a continuación.

$$R = \frac{K}{N} = 1 - \frac{M}{N} \quad (3.13)$$

La figura 3.6 muestra un ejemplo de una matriz LDPC regular (3,4) de orden 12x16, en la cual se puede observar que por cada columna existen 3 “1” y por cada fila 4 “1”.

1					1					1	1				
	1					1		1				1			
		1					1		1				1		
			1	1						1					1
	1					1				1				1	
		1					1				1				1
			1	1				1				1			
1						1			1				1		
			1				1			1			1		
1								1			1			1	
	1				1						1			1	
		1			1			1							1

Figura 3. 6: Matriz LDPC regular (3,4) de orden 12x16.

Por otra parte una matriz LDPC irregular es aquella en que no todas las columnas y filas tienen el mismo número de unos y por lo tanto distintos  $i$  y  $j$ , sin embargo dicha matriz sigue siendo dispersa.

#### 3.4.2.1.2 Representación gráfica

Los códigos LDPC a menudo se representan en forma gráfica ya que los algoritmos eficientes de decodificación se basan en esta representación, dicha representación es llamada gráfico de Tanner. Dicho gráfico consta de dos conjuntos de vértices: vértices para los bits de palabras de código, también llamados nodos variables, y vértices de las ecuaciones de control de paridad, también llamados nodos de chequeo. Los nodos de chequeo están asociados a las filas  $M$  de la matriz  $H$  que corresponde a cada ecuación de chequeo de paridad y los nodos variables están asociados a las columnas  $N$  y corresponden a cada bit de la palabra de código. El número de aristas en el gráfico de Tanner corresponde al número de “1” en la matriz de control de paridad.

En la figura 3.8 se muestra un ejemplo de un gráfico de Tanner de la matriz de chequeo  $H$  para la representación gráfica de la figura 3.7.

$$H = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Figura 3. 7: Matriz de paridad usada en el gráfico de Tanner de la figura 3.8.

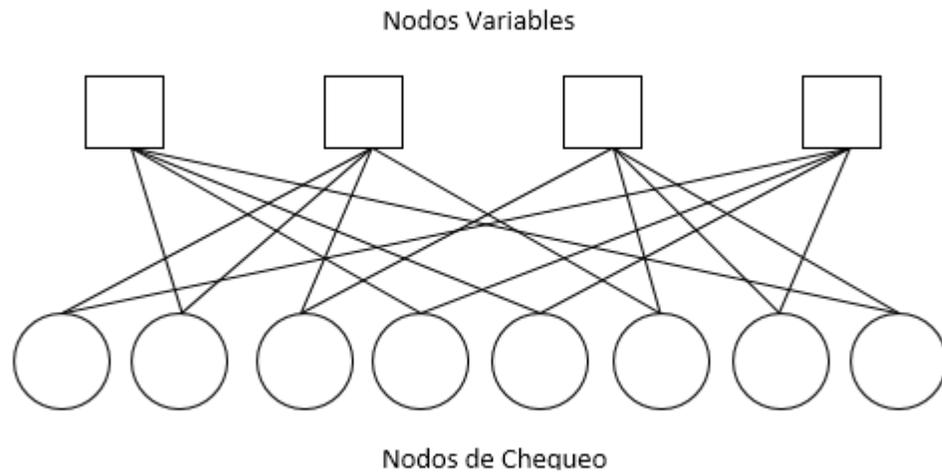


Figura 3. 8: Gráfico de Tanner de la figura 3.7.

En el ejemplo de la figura 3.8 todos los nodos variables y de chequeo tienen el mismo número de aristas por lo cual el código de chequeo de paridad es regular. Lo que se busca en este tipo de códigos es que el número de nodos por donde pasa el ciclo sea lo mayor posible ya que ofrece una mayor capacidad de detección y corrección de errores al momento de decodificar.

### 3.4.2.2 Construcción de los códigos LDPC estructurados

Los códigos LDPC generados aleatoriamente presentan una alta complejidad en su implementación por lo cual es preferible implementar códigos LDPC estructurados, estos códigos son igual de eficientes, y reducen la complejidad de codificación. Una matriz estructurada se genera dividiendo la matriz en sub-matrices cuadradas y regulares.

0	0	0	0	0	0	0	1	1	0	1	0	0	0	1
1	0	0	0	0	0	0	0	1	1	0	1	0	0	0
0	1	0	0	0	0	0	0	0	1	1	0	1	0	0

0	0	1	0	0	0	0	0	0	0	0	1	1	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	1
1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0
0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1
1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1
1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0
0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0
0	0	0	1	1	0	1	0	0	0	1	0	0	0	0	0
0	0	0	0	1	1	0	1	0	0	0	1	0	0	0	0
0	0	0	0	0	1	1	0	1	0	0	0	1	0	0	0
0	0	0	0	0	0	1	1	0	1	0	0	0	1	0	0
0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0

Figura 3. 9: Ejemplo de una sub-matriz cíclica 15x15.

En el ejemplo de la figura 3.9 podemos observar una sub-matriz cíclica de tamaño 15x15 en la cual se puede detallar que cada fila es un bit a la derecha de desplazamiento de la fila anterior. Toda sub-matriz se determina de forma única y específica por su primera fila.

#### 3.4.2.2.1 Código base LDPC (8176,7156)

La matriz de paridad del código LDPC (8176,7156) tiene un tamaño de 1022x8176 y se puede dividir en sub-matrices de tamaño 511x511 de peso unitario, por tanto el número de sub-matrices es 16 por fila y 2 por columna tal como se muestra en la figura 3.10.

$$H = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} & A_{1,5} & A_{1,6} & A_{1,7} & A_{1,8} & A_{1,9} & A_{1,10} & A_{1,11} & A_{1,12} & A_{1,13} & A_{1,14} & A_{1,15} & A_{1,16} \\ A_{2,1} & A_{2,2} & A_{2,3} & A_{2,4} & A_{2,5} & A_{2,6} & A_{2,7} & A_{2,8} & A_{2,9} & A_{2,10} & A_{2,11} & A_{2,12} & A_{2,13} & A_{2,14} & A_{2,15} & A_{2,16} \end{bmatrix}$$

Figura 3. 10: Matriz de paridad base para el código LDPC (8176,7156).

A continuación se presenta una tabla que muestra la forma en que se ubican los unos en la primera fila de cada sub-matriz cíclica. [4]

Tabla 3. 1: Especificaciones de las sub-matrices cíclicas.

Sub-matriz	Posición de los "1" en la primera fila	Posición de los "1" en la primera fila de la matriz de comprobación de paridad
$A_{1,1}$	0-176	0-176
$A_{1,2}$	12-239	523-750
$A_{1,3}$	0-352	1022-1374
$A_{1,4}$	24-431	1557-1964
$A_{1,5}$	0-392	2044-2436
$A_{1,6}$	151-409	2706-2964
$A_{1,7}$	0-351	3066-3417
$A_{1,8}$	9-359	3586-3936
$A_{1,9}$	0-307	4088-4395
$A_{1,10}$	53-329	4652-4928
$A_{1,11}$	0-207	5110-5317
$A_{1,12}$	18-281	5639-5902
$A_{1,13}$	0-399	6132-6531
$A_{1,14}$	202-457	6845-7100
$A_{1,15}$	0-247	7154-7401
$A_{1,16}$	36-261	7701-7926
$A_{2,1}$	99-471	99-471
$A_{2,2}$	130-473	641-984

$A_{2,3}$	198-435	1220-1457
$A_{2,4}$	260-478	1793-2011
$A_{2,5}$	215-420	2259-2464
$A_{2,6}$	282-481	2837-3036
$A_{2,7}$	48-396	3114-3462
$A_{2,8}$	193-445	3770-4022
$A_{2,9}$	273-430	4361-4518
$A_{2,10}$	302-451	4901-5050
$A_{2,11}$	96-379	5206-5489
$A_{2,12}$	191-386	5812-6007
$A_{2,13}$	244-467	6376-6599
$A_{2,14}$	364-470	7007-7113
$A_{2,15}$	51-382	7205-7536
$A_{2,16}$	192-414	7857-8079

Dado que en la sub-matrices solo hay 511 posiciones disponibles, las posibilidades en la primera columna van de 0 a 510. En la tercera columna de la tabla 3.1 se representa la posición absoluta de los “1” en la matriz de comprobación de paridad en la cual hay exactamente 8176 posiciones disponibles, por lo cual las posibilidades van desde 0 hasta 8175.

Un gráfico de dispersión de la matriz de comprobación de paridad se muestra en la figura 3.11 en la que cada bit “1” en la matriz está representado por un punto.

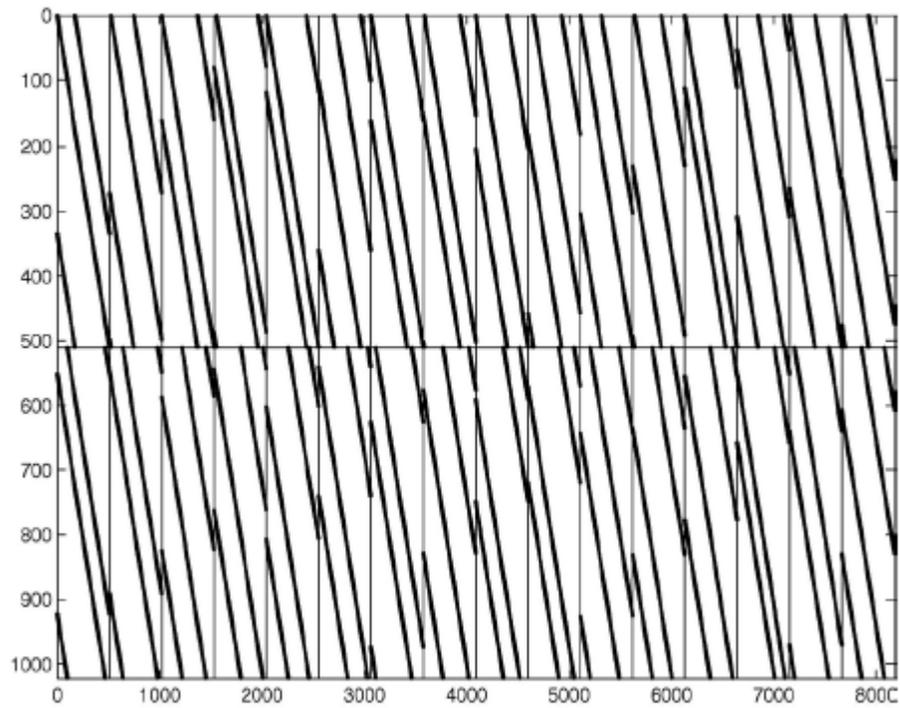


Figura 3. 11: Gráfico de dispersión de la matriz de comprobación de paridad.

La matriz generadora  $G$  se obtiene a partir de la matriz de paridad y para el código (8176,7156) consiste en una matriz de tamaño 7154x8176 compuesta por una matriz identidad de tamaño 7154x7154 y al final dos columnas compuestas de 14 sub-matrices circulantes de tamaño 511x511, tal como se muestra en la figura 3.12. Dichas sub-matrices son de alta densidad y se obtienen aplicando el método de eliminación de Gauss-Jordan a la matriz de paridad.

$$G = \begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{1,1} & B_{1,2} \\ 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{2,1} & B_{2,2} \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{3,1} & B_{3,2} \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{4,1} & B_{3,2} \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{5,1} & B_{3,2} \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{6,1} & B_{3,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{7,1} & B_{3,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & B_{8,1} & B_{3,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & B_{9,1} & B_{3,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & B_{10,1} & B_{3,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & B_{11,1} & B_{3,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & B_{12,1} & B_{3,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & B_{13,1} & B_{13,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & B_{14,1} & B_{14,2} \end{bmatrix}$$

Figura 3. 12: Matriz generadora para el código LDPC (8176,7156).

Tabla 3. 2: Especificaciones de las sub-matrices de la matriz generadora.

Sub-matriz	Primera fila
$B_{1,1}$	55BF56CC55283DFEEFEA8C8CFF04E1EBD9067710988 E25048D67525426939E2068D2DC6FCD2F822BEB6BD96C8A7 6F4932AAE9BC53AD20A2A9C86BB461E43759C
$B_{1,2}$	6855AE08698A50AA3051768793DC238544AF3FE987391 021AAF6383A6503409C3CE971A80B3ECE12363EE809A01D9 1204F1811123EAB867D3E40E8C652585D28
$B_{2,1}$	62B21CF0AEE0649FA67B7D0EA6551C1CD194CA77501 E0FCF8C85867B9CF679C18BCF7939E10F8550661848A4E0A9 E9EDB7DAB9EDABA18C168C8E28AACDDEAB1E
$B_{2,2}$	64B71F486AD57125660C4512247B229F0017BA649C6C1 1148FB00B70808286F1A9790748D296A593FA4FD2C6D7AAF 7750F0C71B31AEE5B400C7F5D73AAF00710
$B_{3,1}$	681A8E51420BD8294ECE13E491D618083FFBBA830DB 5FAF330209877D801F92B5E07117C57E75F6F0D873B3E520F2 1EAFD78C1612C6228111A369D5790F5929A
$B_{3,2}$	04DF1DD77F1C20C1FB570D7DD7A1219EAECEA4B287 7282651B0FFE713DF338A63263BC0E324A87E2DC1AD64C9F 10AAA585ED6905946EE167A73CF04AD2AF9218
$B_{4,1}$	35951FEE6F20C902296C9488003345E6C5526C55192304 54C556B8A04FC0DC642D682D94B4594B5197037DF15B5817 B26F16D0A3302C09383412822F6D2B234E

$B_{4,2}$	7681CF7F278380E28F1262B22F40BF3405BFB92311A8A 34D084C086464777431DBFDDDD2E82A2E6742BAD6533B51B2 BDEE0377E9F6E63DCA0B0F1DF97E73D5CD8
$B_{5,1}$	188157AE41830744BAE0ADA6295E08B79A44081E111F 69BBE7831D07BEEBF76232E065F752D4F218D39B6C5BF20A E5B8FF172A7F1F680E6BF5AAC3C4343736C2
$B_{5,2}$	5D80A6007C175B5C0DD88A442440E2C29C6A136BBC E0D95A58A83B48CA0E7474E9476C92E33D164BFF943A61CE 1031DFF441B0B175209B498394F4794644392E
$B_{6,1}$	60CD1F1C282A1612657E8C7C1420332CA245C0756F78 744C807966C3E1326438878BD2CCC83388415A612705AB192 B3512EEF0D95248F7B73E5B0F412BF76DB4
$B_{6,2}$	434B697B98C9F3E48502C8DBD891D0A0386996146DE BEF11D4B833033E05EDC28F808F25E8F314135E6675B7608B 66F7FF3392308242930025DDC4BB65CD7B6E
$B_{7,1}$	766855125CFDC804DAF8DBE3660E8686420230ED4E04 9DF11D82E357C54FE256EA01F5681D95544C7A1E32B7C30A 8E6CF5D0869E754FFDE6AEFA6D7BE8F1B148
$B_{7,2}$	222975D325A487FE560A6D146311578D9C5501D28BC0 A1FB48C9BDA173E869133A3AA9506C42AE9F466E85611FC5 F8F74E439638D66D2F00C682987A96D8887C
$B_{8,1}$	14B5F98E8D55FC8E9B4EE453C6963E052147A857AC1E 08675D99A308E7269FAC5600D7B155DE8CB1BAC786F45B46 B523073692DE745FDF10724DDA38FD093B1C
$B_{8,2}$	1B71AFFB8117BCF8B5D002A99FEEA49503C0359B056 963FE5271140E626F6F8FCE9F29B37047F9CA89EBCE760405 C6277F329065DF21AB3B779AB3E8C8955400
$B_{9,1}$	0008B4E899E5F7E692BDCE69CE3FAD997183CFAEB27 85D0C3D9CAE510316D4BD65A2A06CBA7F4E4C4A80839AC A81012343648EEA8DBBA2464A68E115AB3F4034
$B_{9,2}$	5B7FE6808A10EA42FEF0ED9B41920F82023085C106FB BC1F56B567A14257021BC5FDA60CBA05B08FAD6DC3B0410 295884C7CCDE0E56347D649DE6DDCEEB0C95E
$B_{10,1}$	5E9B2B33EF82D0E64AA2226D6A0ADCD179D5932EE1 CF401B336449D0FF775754CA56650716E61A43F963D59865C7 F017F53830514306649822CAA72C152F6EB2
$B_{10,2}$	2CD8140C8A37DE0D0261259F63AA2A420A8F81FECB6 61DBA5C62DF6C817B4A61D2BC1F068A50DFD0EA8FE1BD3 87601062E2276A4987A19A70B460C54F215E184
$B_{11,1}$	06F1FF249192F2EAF063488E267EEE994E7760995C4FA 6FFA0E4241825A7F5B65C74FB16AC4C891BC008D33AD4FF 97523EE5BD14126916E0502FF2F8E4A07FC2
$B_{11,2}$	65287840D00243278F41CE1156D1868F24E02F91D3A18 86ACE906CE741662B40B4EFD90F76C1ADD884D920AFA8 B3427EEB84A759FA02E00635743F50B942F0
$B_{12,1}$	4109DA2A24E41B1F375645229981D4B7E88C36A12DA B64E91C764CC43CCEC188EC8C5855C8FF488BB91003602BE F43DBEC4A621048906A2CDC5DBD4103431DB8

$B_{12,2}$	2185E3BC7076BA51AAD6B199C8C60BCD70E8245B874 927136E6D8DD527DF0693DC10A1C8E51B5BE93FF7538FA13 8B335738F4315361ABF8C73BF40593AE22BE4
$B_{13,1}$	228845775A262505B47288E065B23B4A6D78AFBDDDB23 56B392C692EF56A35AB4AA27767DE72F058C6484457C95A8 CCDD0EF225ABA56B7657B7F0E947DC17F972
$B_{13,2}$	2630C6F79878E50CF5ABD353A6ED80BEACC7169179E A57435E44411BC7D566136DFA983019F3443DE8E4C60940BC 4E31DCEAD514D755AF95A622585D69572692
$B_{14,1}$	7273E8342918E097B1C1F5FEF32A150AEF5E11184782B 5BD5A1D8071E94578B0AC722D7BF49E8C78D391294371FFB A7B88FABF8CC03A62B940CE60D669DFB7B6
$B_{14,2}$	087EA12042793307045B283D7305E93D8F74725034E77 D25D3FF043ADC5F8B5B186DB70A968A816835EFB575952E AE7EA4E76DF0D5F097590E1A2A978025573E

En la tabla 3.2 los números en la segunda columna representan la primera fila de cada sub-matriz en representación hexadecimal. Puesto que hay solamente 511 posiciones disponibles el bit a la izquierda se rellena con “0” para permitir un número hexadecimal de 128 dígitos. [4]

#### 3.4.2.2.2 Código LDPC (8160,7136)

Este código es una versión del código LDPC (8176,7156) acortado. Esto se logra mediante la codificación de los bits de relleno virtual, es decir puede usarse el codificador del código LDPC (8176,7156) pero no transmite los bits de relleno virtual por lo que la longitud total de la palabra de código se convierte en 8158, esto nos lleva a agregar dos bits de relleno al final de la palabra de código para conseguir la dimensión de código de (8160,7136) bits o (1020,892) octetos o (255,223) palabras de 32 bits. [4]

La palabra de código abreviada se muestra en la figura 3.13.

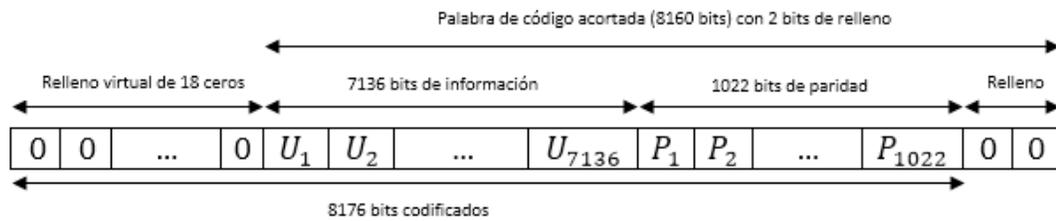


Figura 3. 13: Palabra de código abreviada.

### 3.4.2.3 Decodificadores LDPC

Las técnicas basadas en la decodificación iterativa dan excelentes resultados, por lo cual la implementación de algoritmos para la decodificación de los códigos LDPC es lo más usado. Este proceso consiste en el intercambio de mensajes en forma iterativa entre los nodos de chequeo y los nodos variables, a través de las conexiones propuestas en el gráfico de Tanner. Los mensajes se inicializan con el mensaje recibido, pasan a la parte iterativa donde primero se actualizan los nodos de chequeo usando los mensajes generados por los nodos variables y luego se actualizan los nodos variables procesando el mensaje recibido y los mensajes provenientes de los nodos de chequeo. El proceso de iteración termina cuando se cumple el máximo de iteraciones o la detección de una palabra de código correcta.

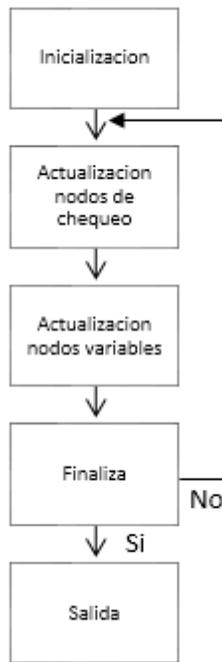


Figura 3. 14: Diagrama de flujo de los algoritmos de decodificación.

A continuación se hace una breve descripción sobre algunos algoritmos de decodificación: [9]

#### 3.4.2.3.1 Sum-Product

Este algoritmo posee una capacidad correctora muy buena, mas sin embargo su complejidad computacional es elevada debido a que hace uso de funciones no lineales, a pesar de ello este algoritmo es ampliamente utilizado.

En el gráfico de Tanner se visualizan las relaciones entre los nodos de chequeo y los nodos variables, este grafico es construido a partir de la matriz de paridad. Los nodos variables son denotados por  $d_j$  y los nodos de chequeo por  $h_i$ , la conexión entre los nodos existirá siempre y cuando el elemento en la matriz de paridad sea uno, es decir  $H_{ij} = 1$ .

Cada nodo variable envía al nodo de chequeo la información probabilística, denotada por  $Q_{ij}^a$ , basada en la información proporcionada por los otros nodos de chequeo relacionados con ese nodo variable. A su vez, cada nodo de chequeo envía la información, denotada por  $R_{ij}^a$ , a cada nodo variable, la cual es calculada con la información dada por los otros nodos variables.

Los valores de los coeficientes  $R_{ij}^0$  y  $R_{ij}^1$  son determinados como una función de los valores de los coeficientes  $Q_{ij}^0$  y  $Q_{ij}^1$  teniendo en cuenta todas las combinaciones de los bits código que satisfacen la ecuación de paridad relacionada con ese cálculo.

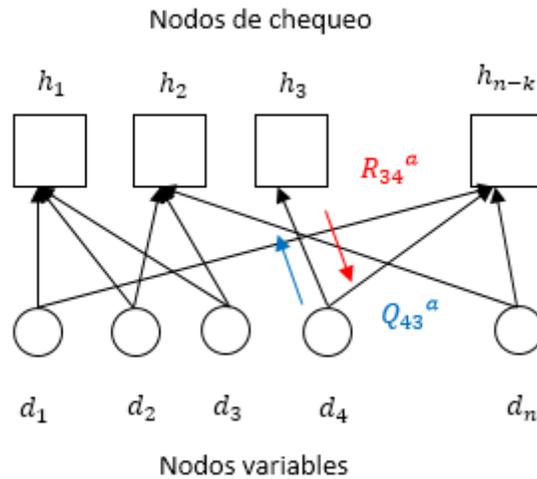


Figura 3. 15: Gráfico de Tanner con ubicación de coeficientes.

#### 3.4.2.3.1.1 Etapas del algoritmo:

1. Inicialización: los valores de  $Q_{ij}^a$  se ajustan en un principio a la probabilidad a priori  $f_j^a$  de los símbolos, que es la probabilidad de que el  $j$  ésimo símbolo sea  $a$ .
2. Paso horizontal: se calcula la ecuación 3.14 para cada  $(i, j)$ .

$$\delta Q_{ij} = Q_{ij}^0 - Q_{ij}^1 \quad (3.14)$$

$$\delta R_{ij} = \prod_{j \in N(i) \setminus j} \delta Q_{ij} \quad (3.15)$$

En la ecuación 3.15  $N(i)$  representa el conjunto de subíndices de todos los nodos variables  $d_j$  que participan del nodo de chequeo  $h_i$ , mientras que  $N(i) \setminus j$  indica la exclusión del nodo  $j$  de ese conjunto. Con el valor  $\delta R_{ij}$  hallado se obtiene el coeficiente  $R_{ij}$  con las siguientes ecuaciones:

$$R_{ij}^0 = \frac{1}{2}(1 + \delta R_{ij}) \quad (3.16)$$

$$R_{ij}^1 = \frac{1}{2}(1 - \delta R_{ij}) \quad (3.17)$$

3. Paso vertical:

$$C_{ij}^a = \alpha_{ij} * f_j^a * \prod_{i \in M(j) \setminus i} R_{ij}^a \quad (3.18)$$

En la ecuación 3.18 se estima el coeficiente  $Q_{ij}^a$  para cada  $(i, j)$ .  $M(j)$  representa el conjunto de subíndices de todos los nodos de chequeo  $h_i$  que participan del nodo variable  $d_j$ , mientras que  $M(j) \setminus i$  indica la exclusión del nodo  $i$  de ese conjunto. La constante  $\alpha_{ij}$  se elige de forma tal que cumpla con la condición  $Q_{ij}^0 + Q_{ij}^1 = 1$ .

4. Estimación a posteriori:

$$Q_j^a = \alpha_j * f_j^a * \prod_{i \in M(j)} R_{ij}^a \quad (3.19)$$

En la ecuación 3.19 para cada  $j$  se realiza una estimación a posteriori, donde de forma similar  $\alpha_j$  se elige de forma tal que cumpla con la condición  $Q_j^0 + Q_j^1 = 1$ .

Estos valores son utilizados para obtener un valor tentativo para cada símbolo  $d_j$ , usando  $d_j = \max(Q_j^a)$ , tal como se muestra en la ecuación 3.20.

$$d_j = \begin{cases} 0 & \text{si } Q_j^0 > Q_j^1 \\ 1 & \end{cases} \quad (3.20)$$

Si la corrección fue la correcta se detiene el algoritmo, sino, se repite a partir del paso horizontal hasta alcanzar el número máximo de iteraciones.

#### 3.4.2.3.2 Min-Sum

Este algoritmo es una variación del algoritmo Sum-Product donde  $d_j$  se aproxima al cálculo del valor mínimo de  $Q_j^a$ . Esta aproximación simplifica el algoritmo debido a que la complejidad del cálculo de los mensajes que envían los nodos de chequeo es menor, sin embargo, la aproximación añade un error que empeora las prestaciones del algoritmo.

## **CAPITULO IV**

### **METODOLOGIA**

En este capítulo se describe la metodología de las fases de ejecución del presente trabajo de grado:

#### ***4.1 Estudio Documental***

Durante esta fase se realizó una revisión bibliográfica y se recopiló información y documentación necesaria para realizar los estudios propuestos, usando el material disponible en Internet, la Biblioteca Central de la Universidad Central de Venezuela, la Biblioteca de la Escuela de Ingeniería Eléctrica de la misma casa de estudios, entre otras bibliotecas de las más reconocidas en el país ya sea en formato físico o digital. Se buscó obtener los conocimientos base para empezar a desarrollar este proyecto, tomando en cuenta las recomendaciones de la CCSDS, ISO, IEEE, así como cualquier otra norma, comité o ente regulador que debió ser considerado.

#### ***4.2 Pruebas Prácticas***

Se generó el formato de la trama propuesta por el estándar AOS, partiendo de una fuente de información conocida y de los valores suministrados en la tabla 4.1 para cada uno de los campos que conforman la trama, lo que nos permitió realizar la simulación 1 de la codificación de la trama con el código LDPC y simular el envío de la misma por varias fuentes ruidosas, esto se llevó a cabo utilizando la figura 4.1 como fuente de información conocida, y los valores de la tabla 4.2, 4.3 y 4.4 en la herramienta

MATLAB para usar el bloque imnoise en tres distintas modalidades de ruido y evaluar el efecto de las distintas fuentes ruidosas.

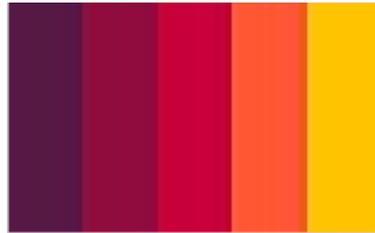


Figura 4. 1: Fuente de información para la simulación 1 de la trama del estándar AOS codificada por el código LDPC.

Tabla 4. 1: Valores de los campos de la trama para la simulación 1.

Campo	Valor
Sincronización	1ACFFC1D
Numero de Versión	01
Identificador Nave Espacial	EA
Identificador Canal Virtual	000001
Contador de Unidades de Datos del Canal Virtual	-
Campo de Repetición	00
Repuestos Reservados	000000
Cabecera	3FFF
Unidad de Datos de Flujo de Bits	-

En la tabla 4.1 no aparecen los valores de los campos Contador de Unidades de Datos del Canal Virtual y Unidad de Datos de Flujo de Bits ya que estos no tienen un valor fijo, el primer campo fue llenado por un conteo de la cantidad de tramas que conforman la información enviada y el segundo campo por la información a enviar.

Tabla 4. 2: Valores de media y varianza suministrados a la herramienta MATLAB para hacer uso del ruido gaussiano en la simulación 1.

Varianza [ $\sigma^2$ ]	Media
0,2	0
0,225	0
0,25	0
0,275	0
0,3	0
0,325	0
0,35	0
0,4	0
0,5	0
0,6	0

Tabla 4. 3: Valores suministrados a la herramienta MATLAB para hacer uso del ruido salt & pepper en la simulación 1.

Densidad [ $d$ ]
0,005
0,01
0,02
0,05
0,1
0,2
0,3
0,4
0,5
0,6

Tabla 4. 4: Valores suministrados a la herramienta MATLAB para hacer uso del ruido speckle en la simulación 1.

Varianza [ $\sigma^2$ ]
0,08
0,085
0,09
0,1
0,2
0,3
0,4
0,5
0,6
0,7

Las matriz generadora y de paridad usada en la simulación 1 fueron las propuestas por el recomendación de la CCSDS 131.1-O-2, Low Density Parity Check Codes For Use In Near-Earth And Deep Space Applications. Orange Book. Issue 2. Septiembre 2007.

Se realizó la simulación 2, 3 y 4 donde se evaluó la correspondencia del resultado del algoritmo a nivel de decodificación con fines de recuperar la información de la fuente.

La simulación 2 fue realizada transmitiendo una o dos letras y junto con el grafico de Tanner formado por la matriz de paridad se realizó la correcta codificación y decodificación de la letra enviada. Se usó como fuente de información letras del alfabeto hexadecimal por lo cual las letras o combinaciones de dos letras aceptadas en dicha simulación se encuentran en la tabla 4.5, y las siguientes matrices generadoras y de paridad dependiendo de la cantidad de letras enviadas.

Tabla 4. 5: Fuentes de información para la simulación 2 de transmisión de letras codificados y decodificados con el código LDPC.

	A	B	C	D	E	F
A	AA	AB	AC	AD	AE	AF
B	BA	BB	BC	BD	BE	BF
C	CA	CB	CC	CD	CE	CF
D	DA	DB	DC	DD	DE	DF
E	EA	EB	EC	ED	EE	EF
F	FA	FB	FC	FD	FE	FF

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 4. 2: Matriz de paridad para la transmisión de una letra, simulación 2.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Figura 4. 3: Matriz generadora para la transmisión de una letra, simulación 2.

$$C1 + C4 + C5 = 0 \quad (4.1)$$

$$C1 + C2 + C6 = 0 \quad (4.2)$$

$$C2 + C3 + C7 = 0 \quad (4.3)$$

$$C3 + C4 + C8 = 0 \quad (4.4)$$

Las ecuaciones 4.1, 4.2, 4.3 y 4.4 son las ecuaciones de chequeo de paridad propuestas por la figura 4.2.

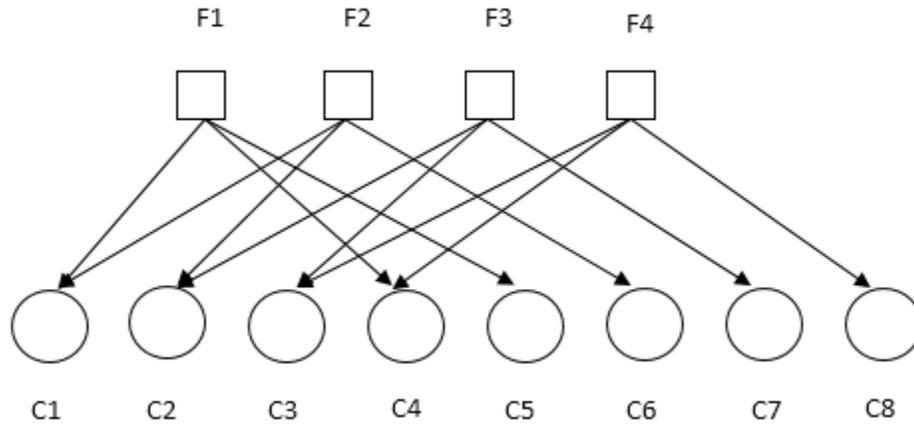


Figura 4. 4: Gráfico de Tanner de la figura 4.2, simulación 2.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 4. 5: Matriz de paridad para la transmisión de dos letras, simulación 2.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Figura 4. 6: Matriz generadora para la transmisión de dos letras, simulación 2.

$$C1 + C6 + C7 + C8 + C9 = 0 \quad (4.5)$$

$$C1 + C2 + C7 + C8 + C10 = 0 \quad (4.6)$$

$$C1 + C2 + C3 + C8 + C11 = 0 \quad (4.7)$$

$$C2 + C3 + C4 + C8 + C12 = 0 \quad (4.8)$$

$$C3 + C4 + C5 + C8 + C13 = 0 \quad (4.9)$$

$$C4 + C5 + C6 + C8 + C14 = 0 \quad (4.10)$$

Las ecuaciones 4.5, 4.6, 4.7, 4.8, 4.9 Y 4.10 son las ecuaciones de chequeo de paridad propuestas por la figura 4.5.

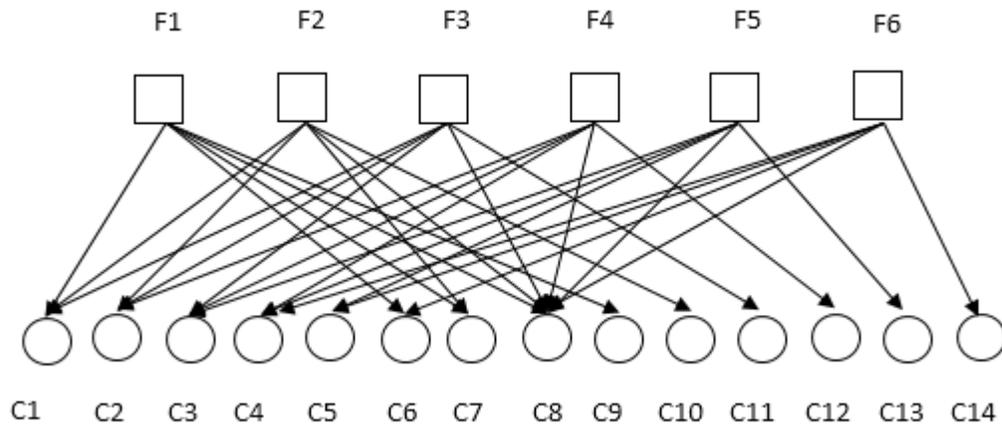


Figura 4. 7: Gráfico de Tanner de la figura 4.5, simulación 2.

Para la simulación 3 se partió de una imagen, figura 4.8, la cual fue transmitida por varias tramas de 8 bits compuestas solamente de los bits de datos, estas fueron sometidas a ruido gaussiano y luego se decodificaron las tramas recibidas. En esta simulación se usaron las matrices de paridad y generadora de las figuras 4.5 y 4.6, respectivamente.



Figura 4. 8: Fuente de información de la simulación 3 de transmisión de una imagen.

En la simulación 4 se permite el ingreso de cualquier tipo de mensaje por lo cual se ingresó el mensaje “*La ciencia es el alma de la prosperidad de las naciones y la fuente de vida de todo progreso*”, se codificó, se sometió a ruido gaussiano y luego se realizó el proceso de decodificación, las letras y el espacio entre ellas están definidos por la tabla 4.6. A continuación también se muestran las matrices de paridad y generadora junto con su gráfico de Tanner usados en esta simulación.

Tabla 4. 6: Alfabeto de las letras y el espacio entre ellas en la simulación 4.

		Alfabeto
Espacio		000000
Letra		
A	A	000001
B	B	000010
C	C	000011
D	D	000100
E	E	000101
F	F	000110
G	G	000111
H	H	001000
I	i	001001

J	j	001010
K	k	001011
L	l	001100
M	m	001101
N	n	001110
O	o	001111
P	p	010000
Q	q	010001
R	r	010010
S	s	010011
T	t	010100
U	u	010101
V	v	010110
W	w	010111
X	x	011000
Y	y	011001
Z	z	011010

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 4. 9: Matriz de paridad de la simulación 4 de envío de un mensaje.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Figura 4. 10: Matriz generadora de la simulación 4 de envío de un mensaje.

$$C1 + C2 + C3 + C7 = 0 \quad (4.11)$$

$$C2 + C3 + C4 + C8 = 0 \quad (4.12)$$

$$C3 + C4 + C5 + C9 = 0 \quad (4.13)$$

$$C4 + C5 + C6 + C10 = 0 \quad (4.14)$$

$$C1 + C5 + C6 + C11 = 0 \quad (4.15)$$

$$C1 + C2 + C6 + C12 = 0 \quad (4.16)$$

Las ecuaciones 4.11, 4.12, 4.13, 4.14, 4.15 y 4.16 son las ecuaciones de chequeo de paridad propuestas por la figura 4.9.

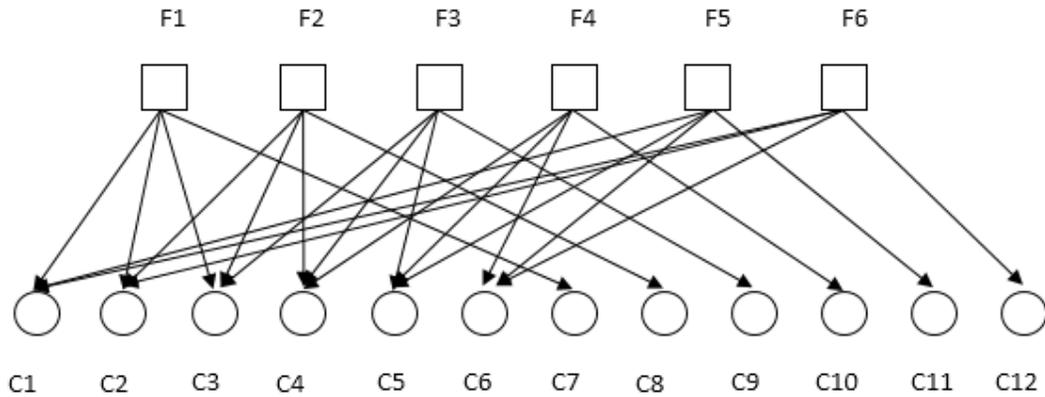


Figura 4. 11: Gráfico de Tanner de la figura 4.9, simulación 4.

También se realizó la simulación 5 y 6 de distintos bloques de aleatorización para evaluar la aleatorización que propone dicho esquema a las tramas luego de ser codificados y antes de ser moduladas.

Tabla 4. 7: Trama utilizada para la simulación 5 y 6 del bloque de aleatorización.

0	1	1	1	1	0	1	0	1	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$$g_1(x) = x^{15} + x^{14} + 1 \quad (4.17)$$

$$g_1(x) = x^8 + x^7 + x^5 + x^3 + 1 \quad (4.18)$$

Con las ecuaciones 4.17 y 4.18 se construyeron los bloques de aleatorización de las figura 4.12 y 4.13.

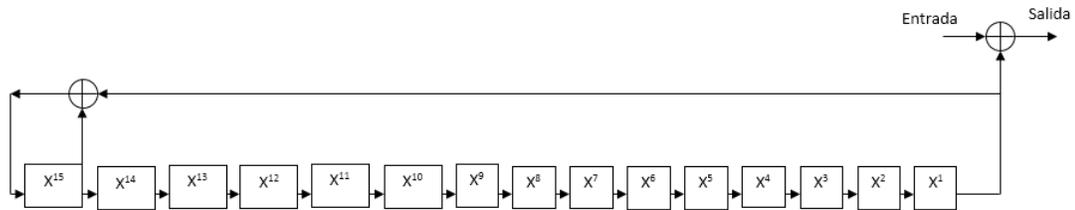


Figura 4. 12: Bloque de aleatorización de la ecuación 4.17, simulación 5.

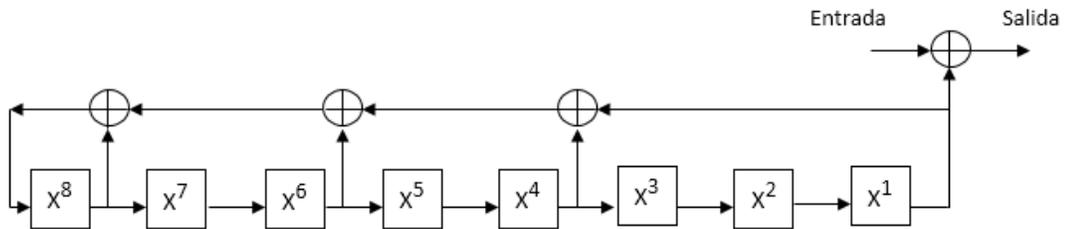


Figura 4. 13: Bloque de aleatorización de la ecuación 4.18, simulación 6.

## CAPITULO V

### RESULTADOS

#### *5.1 Simulación 1. Trama del estándar AOS codificada por el código LDPC.*

En esta simulación se comenzó por armar la trama con el formato del estándar AOS con los valores suministrados de la tabla 3 y rellenando el campo de la Unidad de Datos de Flujo de Bits con los bits de la fuente de información. A medida que dicho campo llegaba almacenar 7072 bits, se sumaba uno al Contador de Unidades de Datos del Canal Virtual. Como el Identificador de Canal Virtual indica la cámara por la cual es realizada la captura de la imagen, de ser una imagen transmitida desde el espacio, el receptor sabe las dimensiones de la imagen a recibir por lo cual esta información de la imagen no es transmitida.

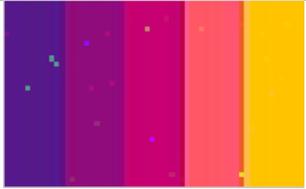
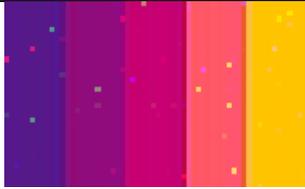
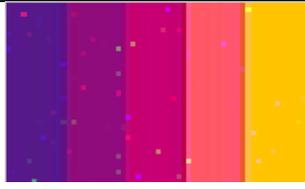
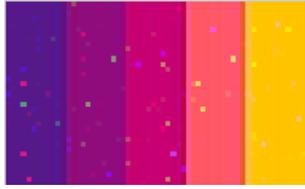
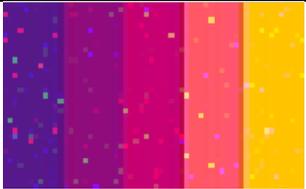
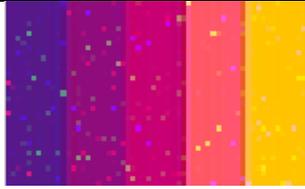
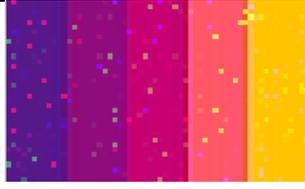
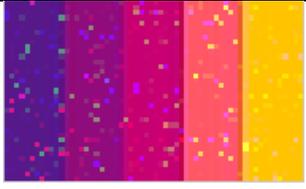
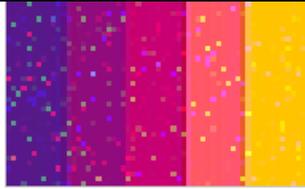
Se formaron 8 vectores de 7136 bits con la trama AOS y la fuente de información, a estos 8 vectores se le agregaron 18 bits 0 de relleno virtual al comienzo de la trama para así poder hacer uso de la matriz generadora recomendada por la CCSDS. Luego de formar las palabras de código mediante la multiplicación de las tramas y la matriz generadora se obtuvieron 8 vectores de 8176 bits cada uno, es decir la matriz generadora les agregó 1022 bits de paridad a cada una de las tramas. A esos 8 vectores se les extrajeron los 18 bits agregados al comienzo de la misma y se le agregaron 2 bits 0 al final de cada una de ellas.

Al realizar los pasos indicados se generaron 8 tramas de 8160 bits, es decir que a las tramas originales se les agregaron 1024 bits de paridad, estos bits de redundancia agregados en la trama ayudó a verificar el comportamiento de la misma al ser sometida a diferentes fuentes ruidosas.

En la tabla 5.1 se tienen las degradaciones obtenidas de la imagen conocida como fuente de información para la fuente ruidosas de tipo gaussiana, dicha simulación se

corrió tres veces por valor de ruido introducido para así calcular un promedio de bits errados en las tramas transmitidas, dichos promedios se encuentran reflejados en la tabla 5.2.

Tabla 5. 1: Degradación de la imagen obtenida luego de ser sometida a distintos valores para una fuente de ruido tipo gaussiana.

$\sigma^2$	Muestra 1	Muestra 2	Muestra 3
0,2			
0,225			
0,25			
0,275			
0,3			

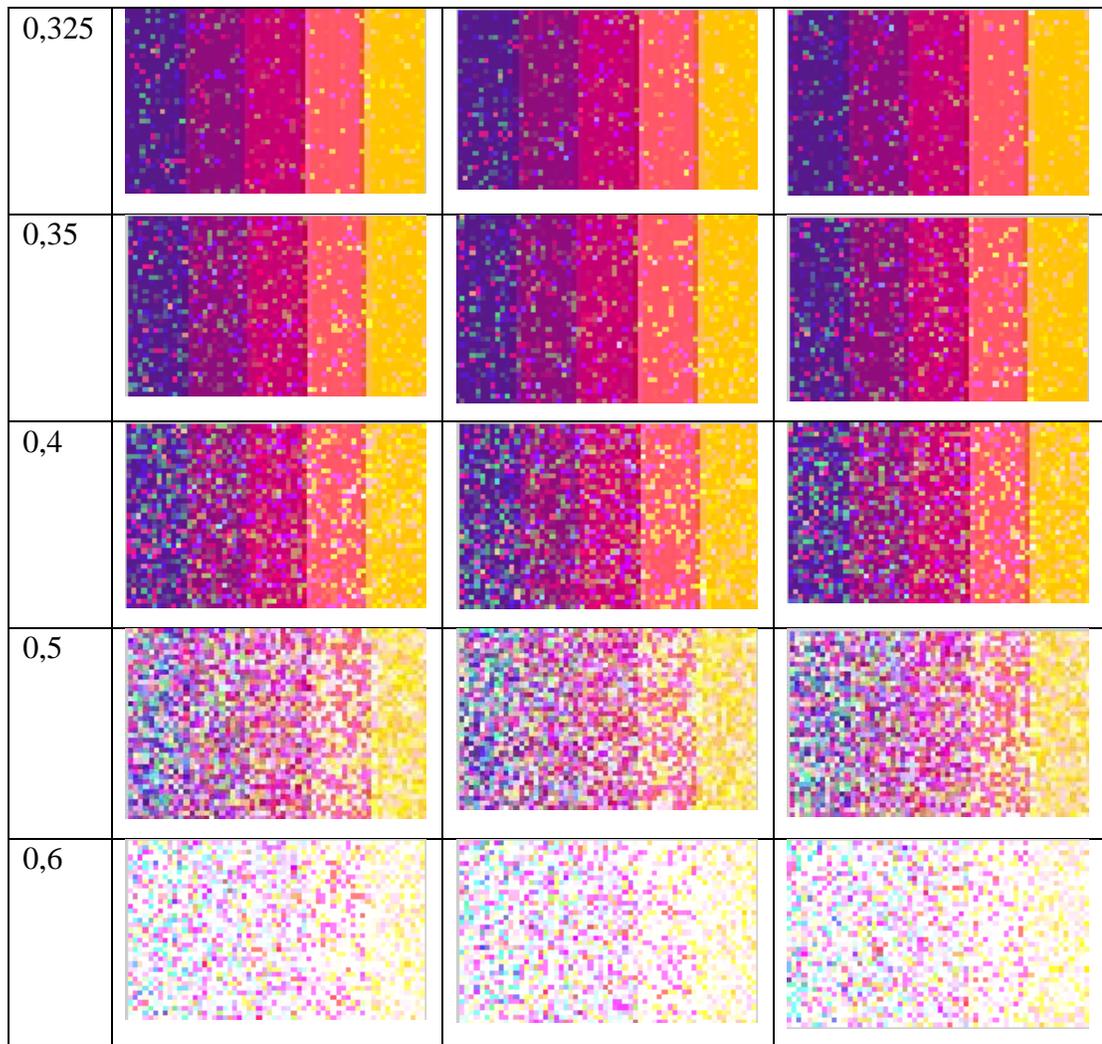


Tabla 5. 2: Cantidad de bits errados en cada una de las muestras de la imagen afectada por distintos valores de ruido tipo gaussiano.

$\sigma^2$	Muestra 1	Muestra 2	Muestra 3	Promedio
0,2	42	55	55	51
0,225	115	98	111	108
0,25	225	213	229	223
0,275	428	495	466	463
0,3	808	789	873	823
0,325	1431	1467	1408	1436

0,35	2463	2371	2427	2421
0,4	5760	5717	5789	5756
0,5	18002	18125	17991	18040
0,6	30591	30541	30528	30554

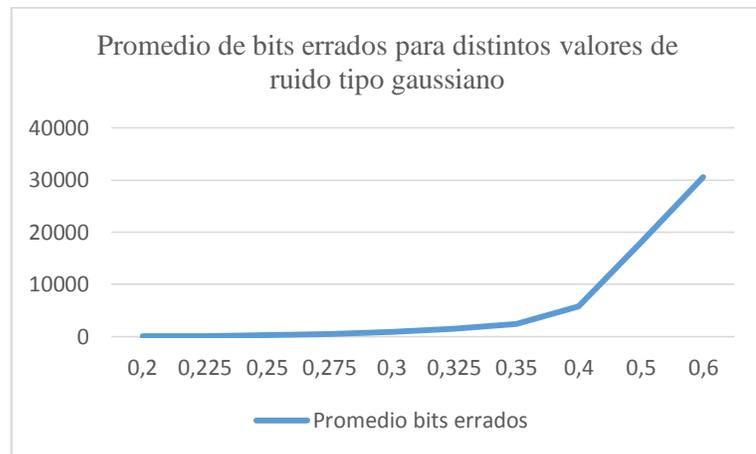


Figura 5. 1: Promedio de bits errados para distintos valores de ruido tipo gaussiano.

Cuando las dimensiones de la imagen enviada son grandes, la probabilidad de que se observe el error recibido es muy pequeña por lo cual es a partir de altos valores de fuentes ruidosas que se empieza a ver muy afectada la imagen, esto se puede observar en la figura 5.1.

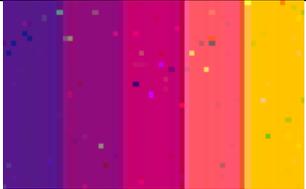
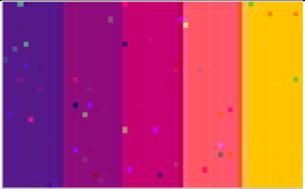
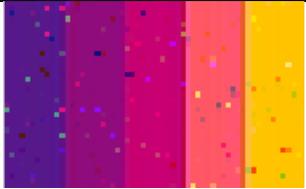
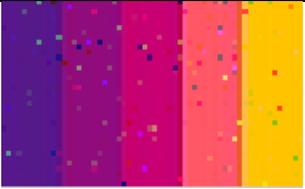
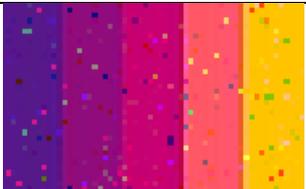
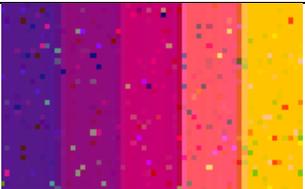
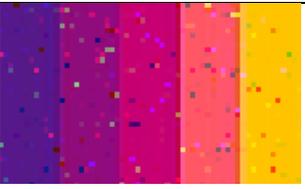
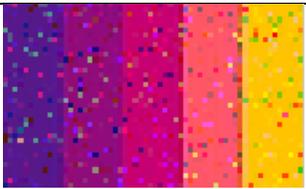
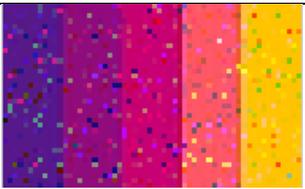
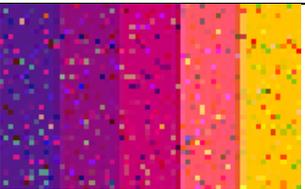
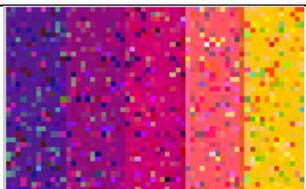
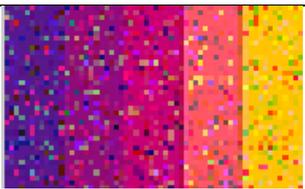
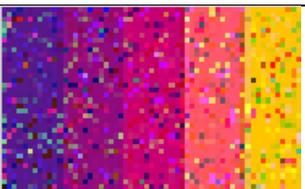
Este tipo de ruido produce pequeñas variaciones en las imágenes afectándolas completamente y ocasionando que la intensidad de todos los pixeles se vea alterada, suele ser ocasionado por los componentes electrónicos.

El estándar AOS introduce cierta redundancia además de la introducida por los bits de paridad lo que también protege mejor la información enviada, por lo cual de ser empleado el algoritmo de detección y corrección de errores en el receptor, los errores en la imagen para bajos niveles de ruido podrían estar muy cerca de ser completamente

corregidos, además que los bits afectados por el ruido para los valores de varianza menores que 0,4 son solo alguno de los bits suministrados por la fuente de información.

En la tabla 5.3 se tiene las degradaciones obtenidas de la imagen conocida como fuente de información para la fuente ruidosas de tipo salt & pepper, dicha simulación se corrió tres veces por valor de ruido introducido para así calcular un promedio de bits errados en las tramas transmitidas, dichos promedios se encuentran reflejados en la tabla 5.4.

Tabla 5. 3: Degradación de la imagen obtenida luego de ser sometida a distintos valores para una fuente de ruido tipo salt & pepper.

$d$			
0,005			
0,01			
0,02			
0,05			
0,1			

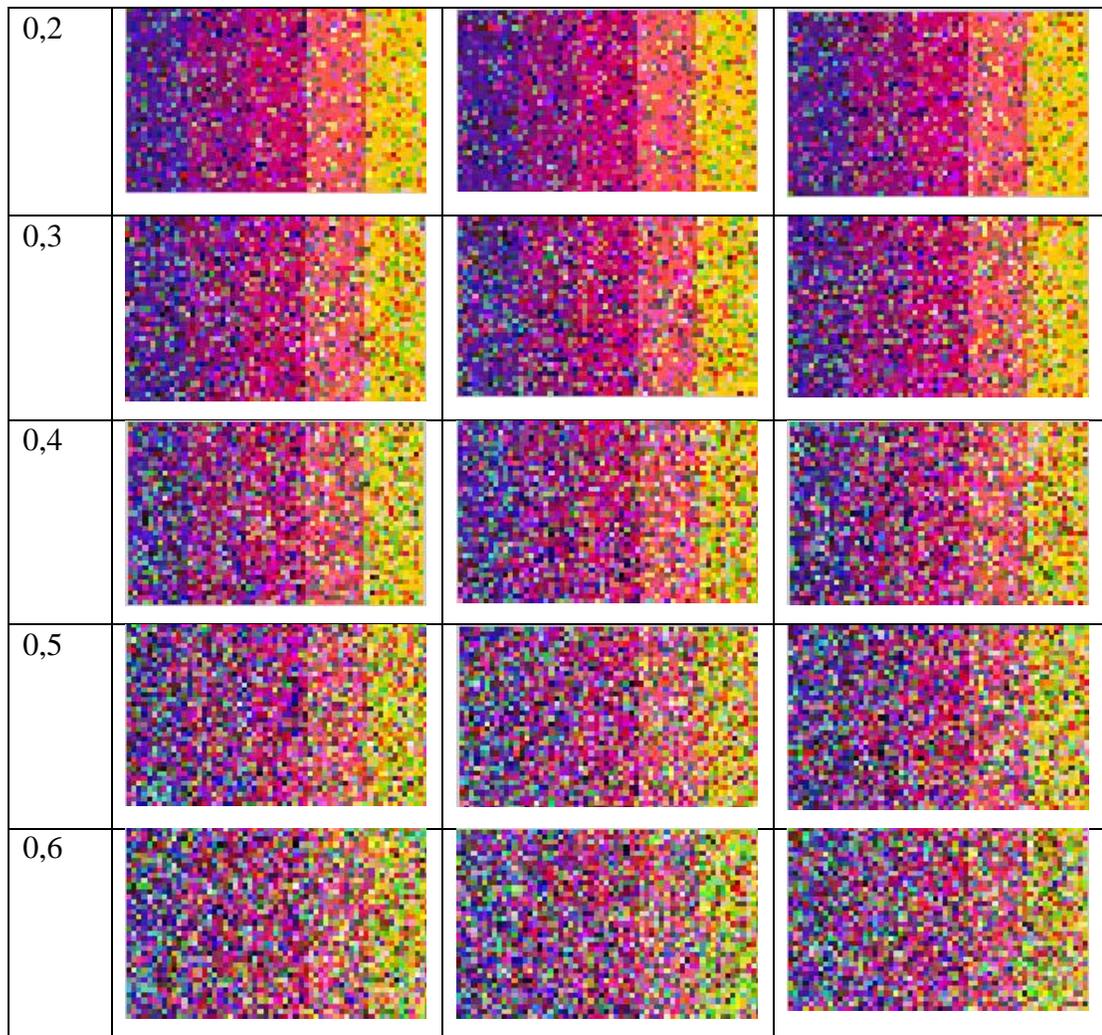


Tabla 5. 4: Cantidad de bits errados en cada una de las muestras de la imagen afectada por distintos valores de ruido tipo salt & pepper.

$d$	Muestra 1	Muestra 2	Muestra 3	Promedio
0,005	160	158	168	162
0,01	321	330	337	330
0,02	638	697	617	651
0,05	1555	1533	1749	1613
0,1	3250	3266	3237	3251
0,2	6491	6383	6487	6454

0,3	9736	9935	9858	9843
0,4	13065	12281	13016	12788
0,5	16357	16299	16336	16331
0,6	19494	19671	19464	19543

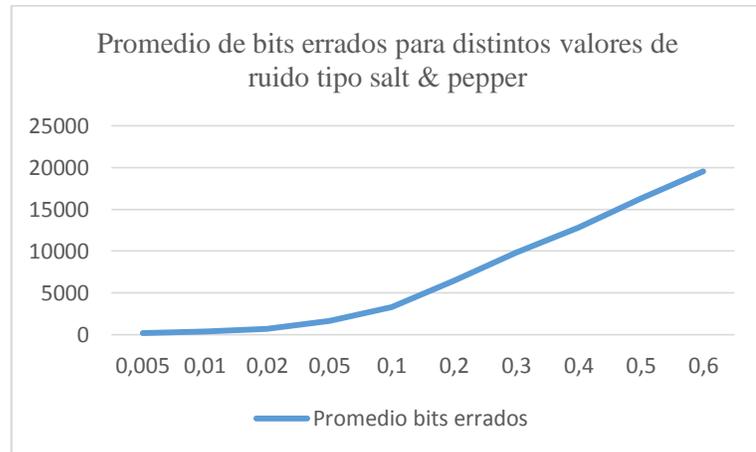


Figura 5. 2: Promedio de bits errados para distintos valores de ruido tipo salt & pepper.

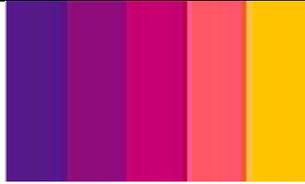
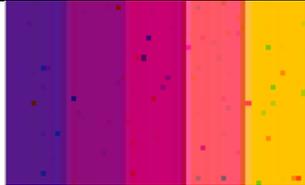
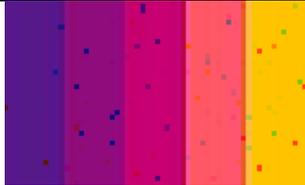
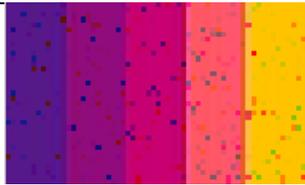
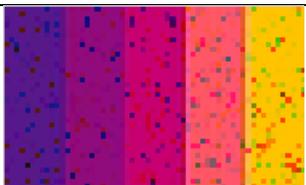
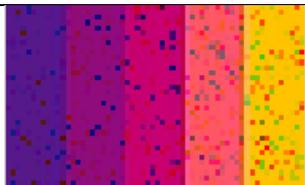
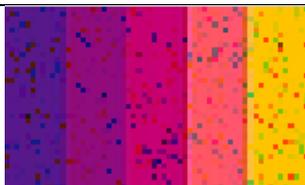
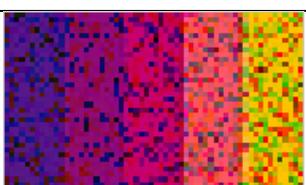
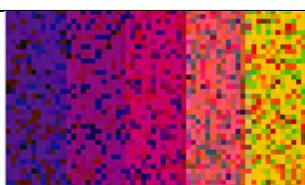
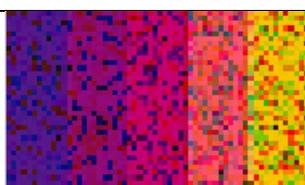
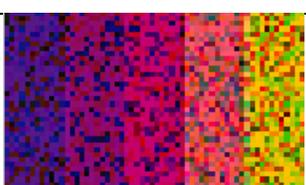
Para este tipo de ruido el valor que toma la señal no tiene relación con el valor real sino que toma valores muy altos o muy bajos, es decir toma el valor máximo (sal) o el mínimo (pimienta), por lo cual en la figura 5.2 se puede observar que la información se ve afectada a muy bajos valores de densidad y a medida que se incrementa la información se distorsiona notablemente.

Este tipo de ruido es ocasionado por la transmisión digital por lo cual es importante emplear el algoritmo de detección y corrección de errores para así poder reducir el efecto de la misma sobre la trama, aunque también pueden ser disminuidos haciendo uso de un filtro pertinente.

En la tabla 5.5 se tiene las degradaciones obtenidas de la imagen conocida como fuente de información para la fuente ruidosas de tipo speckle, dicha simulación se corrió tres

veces por valor de ruido introducido para así calcular un promedio de bits errados en las tramas transmitidas, dichos promedios se encuentran reflejados en la tabla 5.6.

Tabla 5. 5: Degradación de la imagen obtenida luego de ser sometida a distintos valores para una fuente de ruido tipo speckle.

$\sigma^2$			
0,08			
0,085			
0,09			
0,1			
0,2			
0,3			

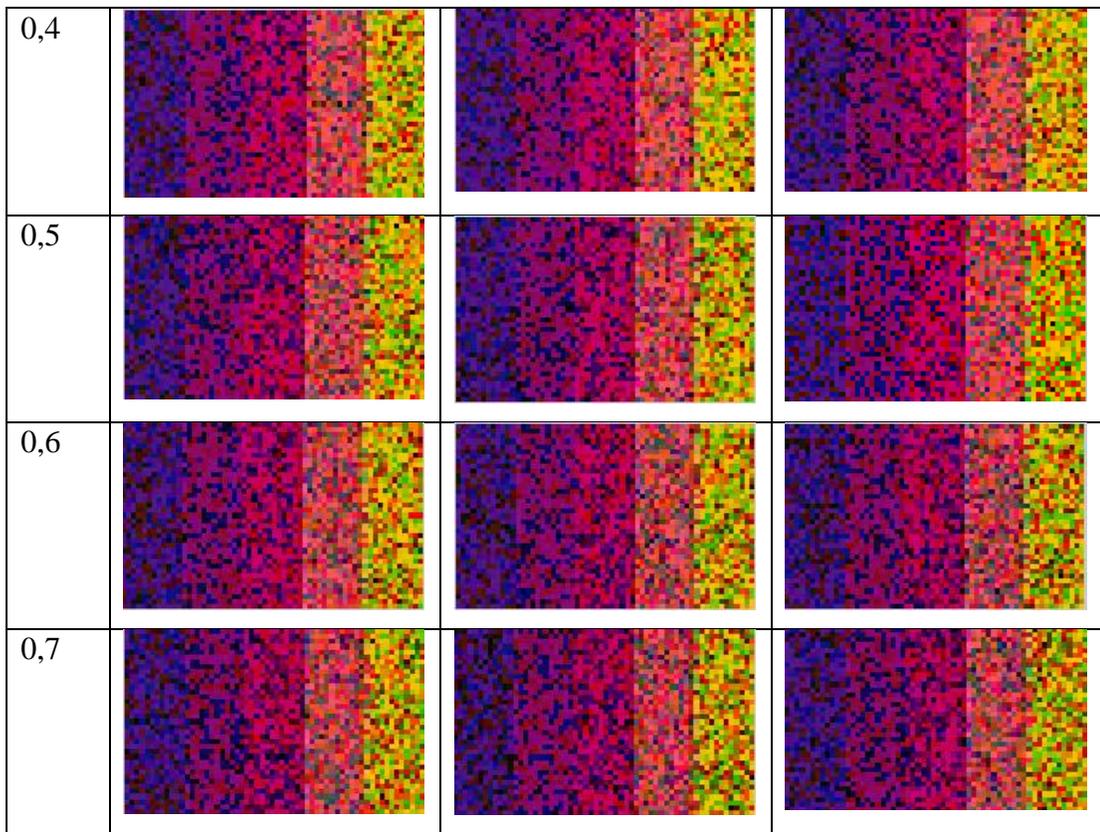


Tabla 5. 6: Cantidad de bits errados en cada una de las muestras de la imagen afectada por distintos valores de ruido tipo speckle.

$\sigma^2$	Muestra 1	Muestra 2	Muestra 3	Promedio
0,08	0	0	0	0
0,085	146	136	145	143
0,09	547	573	526	549
0,1	1248	1290	1193	1244
0,2	5235	5192	5127	5185
0,3	6885	6846	6814	6849
0,4	7928	7709	7875	7838
0,5	8569	8658	8549	8592
0,6	8954	9109	9195	9086
0,7	9375	9511	9402	9430

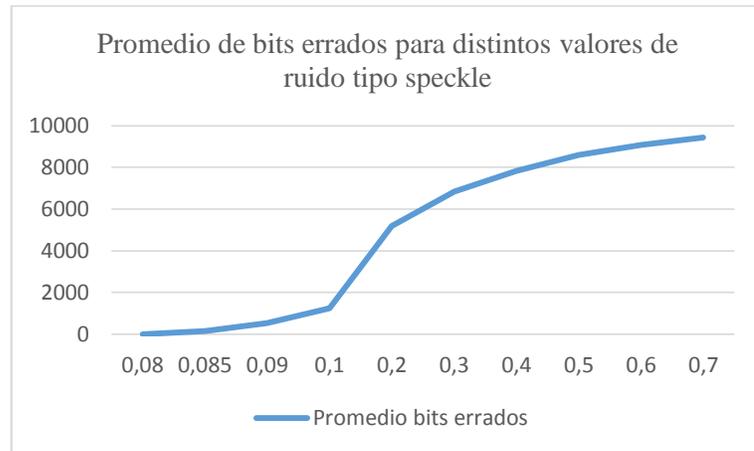


Figura 5. 3: Promedio de bits errados para distintos valores de ruido tipo speckle.

Este ruido es el patrón de interferencia que se forma cuando una imagen se obtiene a partir de un medio con una radiación, el cual degrada significativamente la calidad de la imagen, aumentando de esta forma la dificultad de discriminar detalles finos en las imágenes. La información recibida se ve muy afectada a bajos valores de ruido tal como se observa en la figura 5.3, por lo cual es importante recordar nuevamente el uso de algoritmos para la detección y corrección de errores en la trama ya que como la transmisión de este tipo de tramas se lleva a cabo en el espacio la información se encuentra expuesta a este tipo de ruidos y es importante que no afecten la trama enviada y el mensaje pueda ser interpretado correctamente en el receptor.

### ***5.2 Simulación 2. Transmisión de letras codificadas y decodificadas con el código LDPC.***

En la tabla 5.7 se observa la evolución de la letra o letras enviadas en la simulación 2, dicha letra o letras fueron codificadas con la matriz generadora correspondiente a su longitud. Las palabras de código formadas se les modificó un bit aleatorio y esta es la

ráfaga que se denomina como recibida. La palabra de código recibida es decodificada con su matriz de paridad y ecuaciones de chequeo de paridad correspondientes, luego de este proceso lo llamamos palabra de código corregida.

Tabla 5. 7: Resultados obtenidos para la transmisión de una o dos letras.

		Enviado	Recibido	Corregido
	Letras	A	8	A
	Palabra Código	1010111	10001111	10101111
	Letras	D	D	D
	Palabra Código	11010011	11010001	11010011
	Letras	EC	CC	EC
	Palabra Código	1101100001000	11001100001000	11101100001000
	Letras	BF	3F	BD
	Palabra Código	10111111011100	00111111011100	10111101011100

Para el primer resultado de la tabla 5.7 fue enviada la letra A por lo cual la matriz generadora utilizada es la de la figura 4.3 y junto con las ecuaciones de chequeo de paridad 4.1, 4.2, 4.3 y 4.4 se calcularon los nodos de chequeo y luego los nodos variables, dichos resultados arrojados por la simulación se encuentran en la figura 5.4.

```

Chequeo de paridad recibido:
F1 -> C1+C4+C5=0
F2 -> C1+C2+C6=0
F3 -> C2+C3+C7=1
F4 -> C3+C4+C8=1
Correccion de paridad:
C1 -> F1=0 F2=0
C2 -> F2=0 F3=1
C3 -> F3=1 F4=1
C4 -> F1=0 F4=1
C5 -> F1=0 F2=0 F3=1 F4=1
C6 -> F1=0 F2=0 F3=1 F4=1
C7 -> F1=0 F2=0 F3=1 F4=1
C8 -> F1=0 F2=0 F3=1 F4=1

```

Figura 5. 4: Resultados de los nodos de chequeo y nodos variables para la letra enviada A.

En la figura 5.4 se puede observar que a lo se llamó chequeo de paridad es el cálculo de los nodos variables calculados con la palabra de código recibida, allí podemos ver que el nodo 3 y 4 existe un error por lo cual se debe detectar cual es el bit errado entre ellos dos. El siguiente paso es calcular los nodos de chequeo, en la figura se denomina corrección de paridad, determinando que el bit errado en el sistema es aquel bit que posea todos sus nodos variables con valor 1. Para el caso de la figura 5.4 el bit errado es el 3 el cual es modificado y podemos ver en la tabla 5.7 que luego de dicha modificación la letra corregida es la enviada.

En el segundo resultado de la tabla 5.7 fue enviada la letra D por lo cual la matriz generadora utilizada es la misma del caso anterior al igual que sus ecuaciones de chequeo de paridad, conociendo esta información se calcularon los nodos de chequeo y luego los nodos variables, dichos resultados arrojados por la simulación se encuentran en la figura 5.5.

```

Chequeo de paridad recibido:
F1 -> C1+C4+C5=0
F2 -> C1+C2+C6=0
F3 -> C2+C3+C7=1
F4 -> C3+C4+C8=0
Correccion de paridad:
C1 -> F1=0 F2=0
C2 -> F2=0 F3=1
C3 -> F3=1 F4=0
C4 -> F1=0 F4=0
C5 -> F1=0 F2=0 F3=1 F4=0
C6 -> F1=0 F2=0 F3=1 F4=0
C7 -> F1=0 F2=0 F3=1 F4=0
C8 -> F1=0 F2=0 F3=1 F4=0

```

Figura 5. 5: Resultados de los nodos de chequeo y nodos variables para la letra enviada D.

En la figura 5.5 se puede observar que a lo que se llamó chequeo de paridad es el cálculo de los nodos variables determinados con la palabra de código recibida, allí se puede ver que el nodo 3 existe un error por lo cual se debe detectar cual es el bit errado entre ellos dos. El siguiente paso es calcular los nodos de chequeo, en la figura se denomina corrección de paridad, determinando que el bit errado en el sistema es aquel bit que posea todos sus nodos variables con valor 1. Para el caso de la figura 5.5 no existe ningún bit errado en la información recibida pero si en los bits de paridad para este caso se toma una condición diferente que cuando el bit errado se encuentra en la información, es decir se determinó que cuando el nodo variable que este incluido en el nodo de chequeo a comprobar sea 1 y solo si todos los demás nodos variables que no conforman el nodo de chequeo son 0 ese bit será el errado. Para el caso de la figura 5.5 el nodo de chequeo 7 cumple con la condición indicada por lo cual este es el bit modificado, lo cual no afecta en la corrección del mensaje enviado ya que en los bits de información no existía ningún error.

En el tercer resultado de la tabla 5.7 fueron enviadas las letras EC por lo cual la matriz generadora utilizada es la de la figura 4.6 y junto con las ecuaciones de chequeo de

paridad 4.5, 4.6, 4.7, 4.8, 4.9 y 4.10 se calcularon los nodos de chequeo y luego los nodos variables, dichos resultados arrojados por la simulación se encuentran en la figura 5.6.

```

Chequeo de paridad recibido:
F1 -> C1+C6+C7+C8+C9=0
F2 -> C1+C2+C7+C8+C10=0
F3 -> C1+C2+C3+C8+C11=1
F4 -> C2+C3+C4+C8+C12=1
F5 -> C3+C4+C5+C8+C13=1
F6 -> C4+C5+C6+C8+C14=0
Correccion de paridad:
C1 -> F1=0 F2=0 F3=1
C2 -> F2=0 F3=1 F4=1
C3 -> F3=1 F4=1 F5=1
C4 -> F4=1 F5=1 F6=0
C5 -> F5=1 F6=0
C6 -> F1=0 F6=0
C7 -> F1=0 F2=0
C8 -> F1=0 F2=0 F3=1 F4=1 F5=1 F6=0
C9 -> F1=0 F2=0 F3=1 F4=1 F5=1 F6=0
C10 -> F1=0 F2=0 F3=1 F4=1 F5=1 F6=0
C11 -> F1=0 F2=0 F3=1 F4=1 F5=1 F6=0
C12 -> F1=0 F2=0 F3=1 F4=1 F5=1 F6=0
C13 -> F1=0 F2=0 F3=1 F4=1 F5=1 F6=0
C14 -> F1=0 F2=0 F3=1 F4=1 F5=1 F6=0

```

Figura 5. 6: Resultados de los nodos de chequeo y nodos variables para la letra enviada EC.

En la figura 5.6 se puede observar que a lo que se llamó chequeo de paridad es el cálculo de los nodos variables calculados con la palabra de código recibida, allí se puede ver que el nodo 3, 4 y 5 existe un error por lo cual se debe detectar cual es el bit errado entre ellos tres. El siguiente paso es calcular los nodos de chequeo, en la figura se denomina corrección de paridad, determinando que el bit errado en el sistema es aquel bit que posea todos sus nodos variables con valor 1. Para el caso de la figura 5.6

el bit errado es el 3 el cual es modificado y podemos ver en la tabla 5.7 que luego de dicha modificación la letra corregida es la enviada.

Para el último resultado de la tabla 5.7 fueron enviadas las letras BF por lo cual la matriz generadora utilizada es la misma del caso anterior al igual que sus ecuaciones de chequeo de paridad, conociendo esta información se calcularon los nodos de chequeo y luego los nodos variables, dichos resultados arrojados por la simulación se encuentran en la figura 5.7.

```
Chequeo de paridad recibido:
F1 -> C1+C6+C7+C8+C9=1
F2 -> C1+C2+C7+C8+C10=1
F3 -> C1+C2+C3+C8+C11=1
F4 -> C2+C3+C4+C8+C12=0
F5 -> C3+C4+C5+C8+C13=0
F6 -> C4+C5+C6+C8+C14=0
Correccion de paridad:
C1 -> F1=1 F2=1 F3=1
C2 -> F2=1 F3=1 F4=0
C3 -> F3=1 F4=0 F5=0
C4 -> F4=0 F5=0 F6=0
C5 -> F5=0 F6=0
C6 -> F1=1 F6=0
C7 -> F1=1 F2=1
C8 -> F1=1 F2=1 F3=1 F4=0 F5=0 F6=0
C9 -> F1=1 F2=1 F3=1 F4=0 F5=0 F6=0
C10 -> F1=1 F2=1 F3=1 F4=0 F5=0 F6=0
C11 -> F1=1 F2=1 F3=1 F4=0 F5=0 F6=0
C12 -> F1=1 F2=1 F3=1 F4=0 F5=0 F6=0
C13 -> F1=1 F2=1 F3=1 F4=0 F5=0 F6=0
C14 -> F1=1 F2=1 F3=1 F4=0 F5=0 F6=0
```

Figura 5. 7: Resultados de los nodos de chequeo y nodos variables para la letra enviada BF.

En la figura 5.7 se puede observar que a lo que se llamó chequeo de paridad es el cálculo de los nodos variables calculados con la palabra de código recibida, allí se puede ver que el nodo 1, 2 y 3 existe un error por lo cual se debe detectar cual es el bit

errado entre ellos tres. El siguiente paso es calcular los nodos de chequeo, en la figura se denomina corrección de paridad, determinando que el bit errado en el sistema es aquel bit que posea todos sus nodos variables con valor 1. Para el caso de la figura 5.7 se poseen dos bits errados, el bit errado 3 y 7, los cuales son modificados y se puede ver en la tabla 16 que luego de dicha modificación la letra corregida no es la enviada, esto se debe a que algunas veces el proceso de detección y corrección de errores puede introducir errores en el proceso. Esta equivocación del proceso ocurre mayormente en matrices de paridad irregulares, ya que como vemos en este caso la cantidad de nodos variables que relacionan al nodo de chequeo 7 son menos que los nodos variables que relacionan al nodo de chequeo 3.

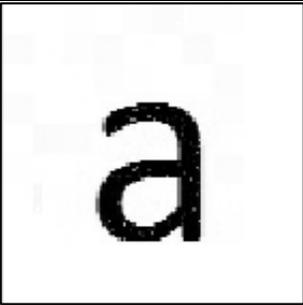
### ***5.3 Simulación 3. Transmisión de imagen codificada y decodificada con el código LDPC.***

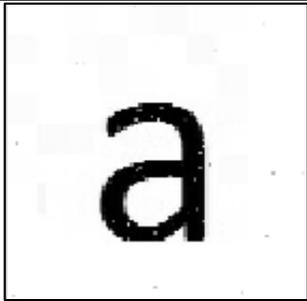
La imagen utilizada en esta simulación esta en formato JPEG y es de tamaño 100x100, es decir que posee 10000 pixeles los cuales se representan por números decimales. Para la simulación cada uno de esos pixeles se pasó a binario y se codificaron con la matriz generadora de la figura 4.6, esas palabras de código obtenidas fueron sometidas a distintos valores de ruido introducidos en la simulación.

A las palabras de código recibidas se les calcularon los nodos variables con las ecuaciones 4.5, 4.6, 4.7, 4.8, 4.9 y 4.10, dichas ecuaciones vienen dadas por la matriz de paridad de la figura 4.5, luego se realizó la comprobación de los nodos de chequeo, esto se llevó a cabo modificando el bit que tuviera todos los nodos variables con valor 1. Este procedimiento se llevó a cabo 10 veces, es decir que luego de modificar algún bit de la palabra de código se volvieron a calcular los nodos variables y se realizó nuevamente la comprobación de los nodos de chequeo, esto con la finalidad de poder detectar y corregir la mayor cantidad de errores posibles.

En la tabla que se muestra a continuación se puede apreciar las imágenes recibidas con distintos valores de ruido gaussiano introducidos en la simulación y las imágenes luego de detectar y corregir la mayor cantidad de errores posibles.

Tabla 5. 8: Resultados de la imagen enviada para distintos valores de ruido gaussiano.

$\sigma^2$	Resultados			
0,2	Mensaje Recibido			
	Mensaje Corregido			
	Bits Errados		Recibidos	80
			Corregidos	14
0,3	Mensaje Recibido			

	Mensaje Corregido		
	Bits Errados	Recibidos	1391
		Corregidos	143
0,4	Mensaje Recibido		
	Mensaje Corregido		
	Bits Errados	Recibidos	9366
		Corregidos	-427

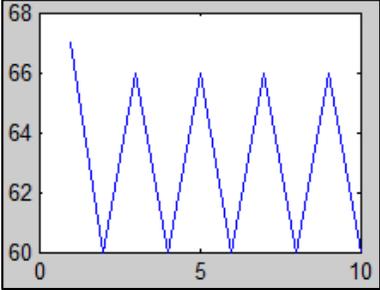
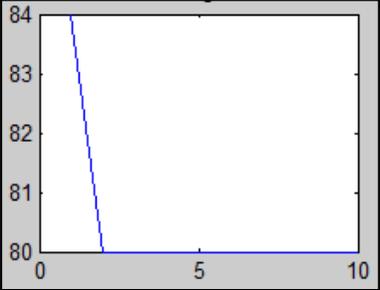
Para el primer resultado de la tabla 5.8 observamos que la imagen recibida llego con 80 errores de los cuales fueron corregidos 14, a pesar de que los errores no fueron corregidos a su totalidad en la imagen no se aprecia ninguna pérdida significativa, esto puede deberse a que la mayoría de los errores pueden encontrarse en los bits de paridad y estos al ser afectados no representan una perdida en la información y/o que la dimensión de la imagen posee tantos bits que la perdida de varios de ellos no es significativa para la misma.

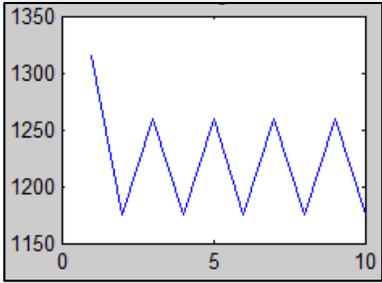
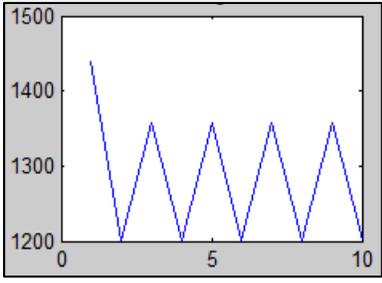
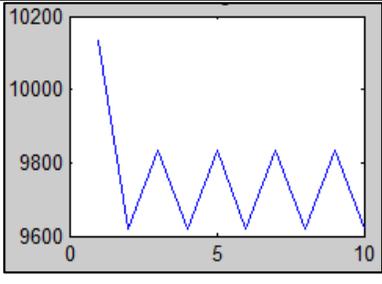
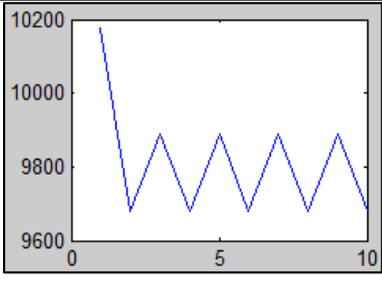
En el segundo resultado se recibieron 1391 errores de los cuales 143 fueron corregidos, para este valor de ruido introducido ya se empieza a notar la perdida de información, ya que se visualiza en la letra de la imagen ciertos puntos blancos. Esto puede corregirse introduciendo mayor cantidad de bits de paridad en el sistema logrando que a pesar de que la fuente de ruido afecte mayor cantidad de bits los que sean afectados no siempre sean los de información.

En el último valor de ruido introducido en la simulación se recibieron 9366 errores de los cuales ninguno fue corregido, es por lo cual en la tabla aparece un -427 lo cual indica que el sistema en vez de corregir errores introdujo 427 errores más, ocurre esto porque las matrices utilizadas en esta simulación no tienen la capacidad de corregir tantos errores producidos ocasionando que en el intento de detectar y corregir errores la simulación introduzca más de ellos.

A continuación se presenta una tabla la cual demuestra el comportamiento de la simulación en el proceso de decodificación.

Tabla 5. 9: Comportamiento de 10 iteraciones en la simulación 3 para distintos valores de ruido gaussiano.

$\sigma^2$		Muestra	
		1	2
0,2	Bits errados recibidos	69	91
	Bits errados en 10 iteraciones		

0,3	Bits errados recibidos	1324	1379
	Bits errados en 10 iteraciones		
0,4	Bits errados recibidos	9225	9306
	Bits errados en 10 iteraciones		

Para el primero valor de ruido gaussiano introducido en la tabla 5.9 podemos ver que existen dos variantes en el comportamiento de la decodificación. La primera opción es que corrija cierta cantidad de errores recibidos pero luego en el proceso de continuar corrigiendo errores quede en un ciclo en el cual cuando corrija ciertos errores introducirá otros errores, para solucionar este problema se emplean algoritmos que requieren una mayor cantidad de procesamiento ya que con ellos la simulación no solo realizaría los cálculos y cambios indicados si no que con el cálculo de otros parámetros este podría tomar una decisión la cual podría ser la opción más acertada para detectar y corregir la mayor cantidad de bits erróneos posibles. La segunda opción es que corrija cierta cantidad de errores hasta un punto en el que la simulación no detecte más errores.

En el segundo valor de ruido gaussiano introducido en las dos muestras se obtuvo la primera opción indicada anteriormente. Para el tercer valor de ruido gaussiano introducido se obtiene de igual manera la primera opción pero las cantidades de bits

errados en la imagen recibida son menores que los valores por donde oscila la detección y corrección de los mismo, esto ocurre por lo que se indicó anteriormente de que se necesita un algoritmo más eficiente para la detección y corrección de errores en estos casos.

**5.4 Simulación 4. Transmisión de mensajes codificados y decodificados con el código LDPC.**

En esta simulación la transmisión del mensaje se realizó letra por letra, es decir que cada una de las letras fue codificada con la matriz generadora de la figura 4.10, se les introdujo ruido a las palabras de código enviadas y dependiendo del valor del ruido ingresado en la simulación se recibieron los distintos mensajes que se encuentran en la tabla 5.10.

Al recibirse los mensajes la simulación realizó el cálculo de los nodos variables planteados en las ecuaciones 4.11, 4.12, 4.13, 4.14, 4.15 y 4.16 con esto se chequearon los nodos de chequeo, este chequeo se llevaba a cabo modificando el bit que tuviera todos los nodos variables con valor 1. Este procedimiento se realizó 3 veces, es decir que después de realizar el chequeo de los nodos de chequeo y modificar los bits correspondientes se volvían a calcular las ecuaciones de los nodos variables con los bits modificados.

En la tabla 5.10 se puede apreciar la cantidad de bits corregidos en el mensaje dependiendo del valor del ruido ingresado en la simulación.

Tabla 5. 10: Resultados del mensaje enviado para distintos valores de ruido gaussiano.

$\sigma^2$	Resultados	
0,2	Mensaje Recibido	la ciencia es el alma de a prosperidad de aw naciones y la fuente de vidahde todo progreso

	Mensaje Corregido	la ciencia es el alma de la prosperidad de las naciones y la fuente de vida de todo progreso		
	Bits Errados		Recibidos	4
			Corregidos	4
0,3	Mensaje Recibido	la ciencia es el alma de la prosperidad de las naciones y la fuente de vida de todo progreso		
	Mensaje Corregido	la ciencia es el alma de la prosperidad de las naciones y la fuente de vida de todo progreso		
	Bits Errados		Recibidos	9
			Corregidos	9
0,4	Mensaje Recibido	lakgmmoci es q mac eilc q owxmrnted u cs nacyoous y lapfu te fepvydc fe toda rogreso		
	Mensaje Corregido	lakcimncia es a mac enla prosperidad de las naciones y la fuente de vida de todo progreso		
	Bits Errados		Recibidos	102
			Corregidos	40

Los mensajes tienen menor probabilidad de verse afectados por el ruido en su transmisión, ya que como fueron enviados letra por letra poseen una gran cantidad de bits de paridad que permiten la correcta detección y corrección de errores esto se puede apreciar en la tabla 5.10 ya que para los dos primeros valores de ruido ingresados la palabra fue recuperada en su totalidad, caso contrario para el último valor de ruido ingresado donde el código solo corrigió 40 de los 120 errores recibidos. Esto podría solucionarse con una matriz que introduzca mayor redundancia y así los bits de información no se vean tan afectados por los altos valores de ruido introducido en el sistema.

### **5.5 Simulación 5 y 6. Bloque de aleatorización.**

En esta simulación se obtuvieron las tablas 5.11 y 5.12 para la simulación 5 y 6 de los distintos bloques de aleatorización. Este bloque debe ser diseñado sabiendo el tipo de modulación que se desea emplear en el sistema de comunicaciones, ya que este bloque nos dará las transiciones de bits necesarias para el correcto funcionamiento del reloj en el receptor.

Tabla 5. 11: Bloque de entrada y de salida para la simulación 5 del bloque de aleatorización de la figura 4.12.

Entrada	0	1	1	1	1	0	1	0	1	0	0	0	0	0	0	1
Salida	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

La figura 5.8 muestra las transiciones de bits de la entrada modulada con QPSK en la cual se visualiza un gran desequilibrio en el flujo de datos lo que ocasionaría una degradación de la tasa de bits errados en el proceso de detección de símbolos.

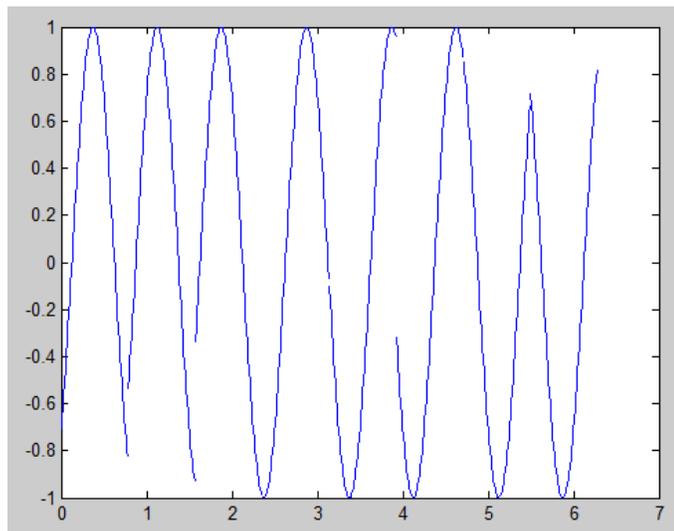


Figura 5. 8: Modulación QPSK de la entrada en el bloque de aleatorización.

La figura 5.9 muestra las transiciones de bits de la salida de la tabla 5.11 modulada con QPSK, en dicha figura se puede apreciar que el flujo de datos presenta un mayor equilibrio.

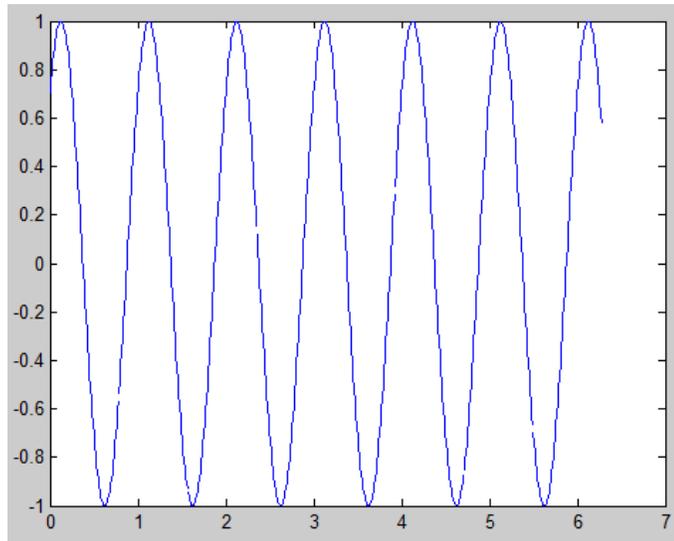


Figura 5. 9: Modulación QPSK de la salida de la simulación 5 en el bloque de aleatorización.

Para la simulación 6 se empleó un bloque de aleatorización distinto con la finalidad de igual demostrar que estos bloques proporcionan un mayor equilibrio en el flujo de datos.

Tabla 5. 12: Bloque de entrada y de salida para la simulación 6 del bloque de aleatorización de la figura 4.13.

Entrada	0	1	1	1	1	0	1	0	1	0	0	0	0	0	0	1
Salida	1	1	1	1	1	1	1	1	1	0	0	1	1	0	1	1

La entrada de la tabla 5.12 es la misma que la tabla 5.11, es decir que la figura 5.8 es la correspondiente a la entrada de la tabla 5.12 modulada con QPSK. La figura 5.10 muestra las transiciones de bits de la salida de la tabla 5.12 modulada con QPSK, en dicha figura se puede apreciar que el flujo de datos presenta un mayor equilibrio al principio de la misma, esto proporciona un mejor funcionamiento en el reloj del receptor.

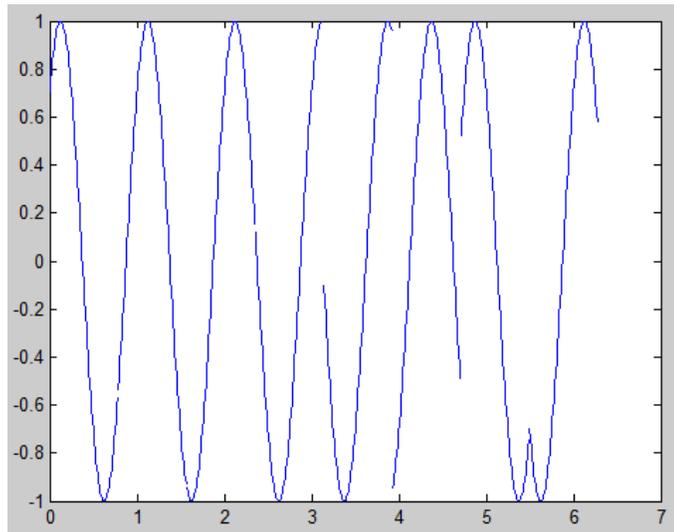


Figura 5. 10: Modulación QPSK de la salida de la simulación 6 en el bloque de aleatorización.

Entonces si se comparan los dos bloques de la simulación 5 y 6 se puede decir que a pesar de que sean bloques distintos los dos tienen la misma finalidad, proporcionar al principio del mensaje a enviar un flujo de datos equilibrado.

Si se considera que:

- Los sistemas de recuperación de reloj de símbolos normalmente extraen la frecuencia del reloj a partir de las transiciones de símbolos recibidas.

- Un gran desequilibrio entre unos y ceros en el flujo de datos podría dar como resultado una degradación de la tasa de bits errados en el proceso de detección de símbolos.
- Las formas de onda NRZ, las cuales son ampliamente utilizadas en estos sistemas de modulación, requieren suficientes transiciones de símbolos para la recuperación del reloj.

Se puede decir que el pseudo-aleatorizador mejorará la estabilidad del sistema.

## CAPITULO VI

### CONCLUSIONES Y RECOMENDACIONES

Con los resultados obtenidos en el capítulo V se pueden obtener una serie de conclusiones y recomendaciones sobre la codificación de canal bajo el código LDPC y el estándar AOS descritas a continuación.

#### *6.1 Conclusiones*

Realizar la transmisión de fuentes de información estructuradas bajo el estándar AOS disminuye la complejidad del sistema y mejora la calidad significativamente de los datos recibidos, ya que a medida de que los datos transmitidos son de mayor dimensión se requieren matrices de paridad de mayor tamaño para su codificación mediante el código LDPC, ocasionando que la dispersión de la matriz de paridad y la cantidad de bits de redundancia introducidos sea mayor esto nos lleva a, que a valores significativos de ruido la mayoría de esos bits sean los que se vean afectados los cuales pueden ser omitidos sin que se produzca una pérdida de información significativa.

Al momento de la detección y corrección de errores al usar matrices irregulares se pueden generar errores en el proceso de decodificación, ya que las relaciones entre nodos variables y nodos de chequeo no son en la misma proporción lo cual ocasiono que al momento de la simulación al querer corregir un error surgiera otro.

Se demostró que a medida que los bits de paridad son menores que la longitud del bloque de entrada, las palabras de código generadas se ven mucho más afectadas por lo cual es más difícil la detección y corrección de errores. En algunos caso se evidenció que era tanto el ruido suministrado que la simulación en vez de corregir errores incluía errores en el mensaje final.

La transmisión de imágenes de grandes dimensiones conlleva a que si son sometidas a bajos o medianos niveles de ruido y ocurren pérdidas en la información, dicha pérdida no es perceptible al ojo humano ya que la cantidad de píxeles que conforma la imagen es grande.

Se requiere un algoritmo para la decodificación más eficiente de manera que tome otras consideramos en el proceso. En este trabajo se utilizó un algoritmo de decodificación sencillo debido al alto requerimiento de almacenamiento y procesamiento demandado por los algoritmos recomendados para la decodificación de los códigos LDPC.

Se pudo observar que el modelo iterativo de decodificación de códigos LDPC implementado mejora levemente los resultados lo cual no lo convierte en un factor que mejore significativamente el desempeño, por el contrario aumenta la memoria y el tiempo requerido por el decodificador LDPC.

La trama del estándar AOS necesita del uso de bloques de aleatorización para proporcionar un equilibrio en el flujo de datos antes de ser modulado, ya que su trama no posee dicha estabilidad y esto es perjudicial para el funcionamiento del receptor.

## ***6.2 Recomendaciones***

Realizar la decodificación mediante algoritmos en matrices de paridad de gran tamaño las cuales poseen un mayor número de columnas linealmente independientes para demostrar que este tipo de códigos tienen una mayor capacidad de corrección de errores.

Implementar un algoritmo distinto al aplicado en este trabajo en distintos casos demostrativos con la finalidad de poder observar el comportamiento del código.

## CAPITULO VII

### MATERIALES DE REFERENCIA

#### *7.1 Referencias bibliográficas*

[1] COMMITTEE FOR SPACE DATA SYSTEMS (2012) **TM Synchronization and Channel Coding – Summary of Concept and Rationale**. CCSDS 130.1-G-2. Green Book. Issue 2.

[2] COMMITTEE FOR SPACE DATA SYSTEMS (2003) **TM Synchronization and Channel Coding**. CCSDS 130.0-B-1. Blue Book. Issue 1.

[3] COMMITTEE FOR SPACE DATA SYSTEMS (2003) **Space Packet Protocol**. CCSDS 133.0-B-1. Blue Book. Issue 1.

[4] COMMITTEE FOR SPACE DATA SYSTEMS (2007) **Low Density Parity Check Codes for Use In Near-Earth and Deep Space Applications**. CCSDS 131.1-O-2. Orange Book. Issue 2.

[5] COMMITTEE FOR SPACE DATA SYSTEMS (2006) **AOS Space Data Link Protocol**. CCSDS 732.0-B-2. Blue Book. Issue 2.

[6] COMMITTEE FOR SPACE DATA SYSTEMS (2012) **Short Block Length LDPC Codes for TC Synchronization and Channel Coding**. CCSDS 231.0-O-x.x. Orange Book. Issue 2.

[7] COMMITTEE FOR SPACE DATA SYSTEMS (2001) **Advanced Orbiting Systems, Networks and Data Links: Architectural Specification**. CCSDS 701.0-B-3. Blue Book. Issue 3.

[8] GODDARD SPACE FLIGHT CENTER GREENBELT (2008) **Low Density Parity Check Code for Rate 7/8**. MD 20771.

- [9] ANGARITA PRECIADO, F. (2013) **Diseño de Decodificadores de Altas Prestaciones para Códigos LDPC**. Universitat Politècnica De València. Tesis Doctoral.
- [10] EUROPEAN COOPERATION FOR SPACE STANDARDIZATION (2007) **Space Engineering. Space Data Links – Telemetry Synchronization and Channel Coding**. ECSS-E-50-01A.
- [11] SAM SHANMUGAN, K. (1979) **Digital and Analog Communication Systems**.
- [12] PEREZ VEGA, C. **Codificación de Canal y Modulación**.
- [13] SANDOVAL RUIZ, C., FEDON, A. (2008) **Descripción Modular de un Esquema de Codificación Concatenado para Corrección de Errores con Programación de Hardware**. Revista chilena de ingeniería, vol. 16 N° 2.
- [14] KAZZ, G., GREENBERG, E., BURLEIGH, S. **Replacing the CCSDS Telecommand Protocol with the Next Generation Uplink (NGU)**. Jet Propulsion Laboratory, California Institute of Technology.
- [15] DE COLA, T., ERNEST, H., MARCHESE, M. (2007) **Performance Analysis of CCSDS File Delivery Protocol and Erasure Coding Techniques in Deep Space Environments**.
- [16] GALLAGER, R. (1963) **Low Density Parity-Check Codes**.
- [17] JOHNSON, S. **Introducing Low-Density Parity-Check Codes**. School of Electrical Engineering and Computer Science, the University of Newcastle.
- [18] FERNANDEZ, W., GARRIDO, C., OLIVARES, H. (2009) **Análisis y Estudio del Desempeño del Código Chequeo de Paridad de Baja Densidad Irregular en un Canal de Línea Eléctrica de Baja Tensión**. Revista chilena de ingeniería, vol. 17 N° 9.
- [19] DECHENE, D., PEETS, K. (2006) **Simulated Performance of Low-Density Parity-Check Codes**. Faculty of Engineering, Lakehead University.

- [20] WOLF, J. (2010) **An Introduction to Error Correcting Codes Part 3.** ECE 154 C.
- [21] VASQUEZ ARRANZ, J. (2012) **Algoritmo de Encadenamiento Jerárquico para WSN Extensas Basado en Codificación de Red y LDPC.** Escuela Técnica Superior de Ingeniería de Telecomunicación, Universidad Rey Juan Carlos.
- [22] DE FEZ LAVA, I. (2010) **Implementación y evaluación de la codificación para la transmisión de ficheros en entornos unidireccionales.**
- [23] ALAFANG MALEMA, G. (2007) **Low-Density Parity-Check Codes: Construction and Implementation.** School of Electrical and Electronic Engineering, Faculty of Engineering, Computer and Mathematical Sciences, the University of Adelaide.
- [24] DE VES CUENCA, E. **MATLAB: Introducción al Procesamiento de Imágenes.**
- [25] ABRAMSON, N. (1981) **Teoría de la Información y Codificación.**
- [26] LEINER, B. (2005) **LDPC Codes- a brief Tutorial.**
- [27] BUEMI, M. (2014) **Ruidos y Filtros Segmentación.** Universidad de Buenos Aires.

## ***7.2 Apéndice***

### **7.2.1 Acrónimos**

AOS: Advanced Orbiting Systems (Sistemas Avanzados en Orbita).

CCSDS: Consultative Committee for Space Data Systems (Comité Consultivo para Sistemas de Datos Espaciales).

IEEE: Institute of Electrical and Electronics Engineers (Instituto de Ingenieros Eléctricos y Electrónicos).

IEC: International Electrotechnical Commission (Comisión Electrotécnica Internacional).

ISO: International Organization for Standardization (Organización Internacional de Normalización).

JPEG: Joint Photographic Experts Group (Grupo Conjunto de Expertos en Fotografía).

LDPC: Low Density Parity Check (Comprobación de Paridad de Baja Densidad).

MATLAB: Matrix Laboratory (Laboratorio de Matrices).

NRZ: Non Return to Zero (No Devolver a Cero).

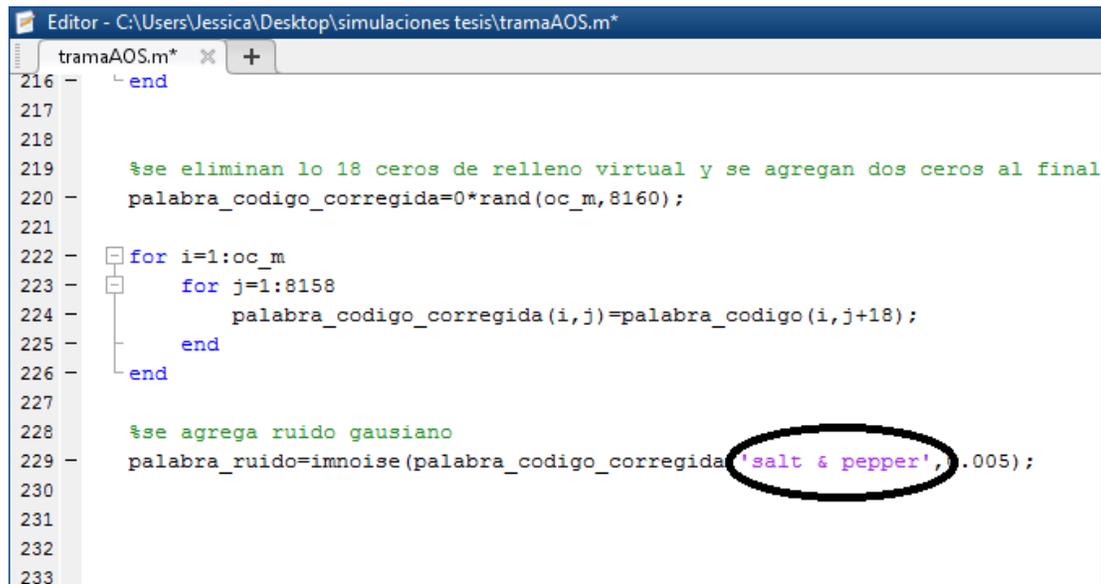
QPSK: Quadrature Phase Shift Keying (Desplazamiento de Fase en Cuadratura).

## 7.3 Anexos

### 7.3.1 Funcionamiento de la simulación 1

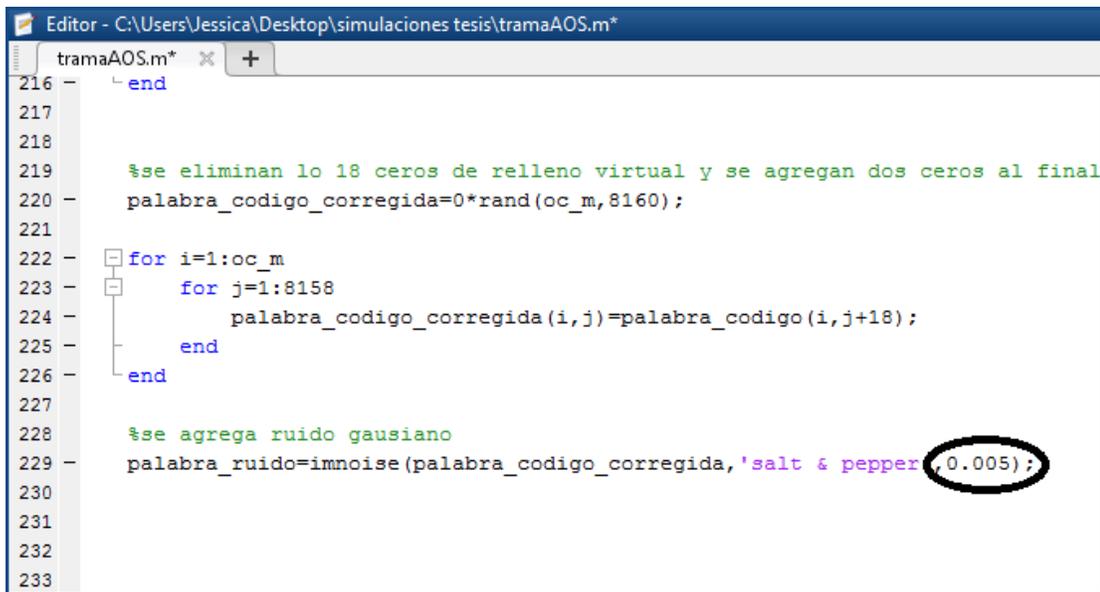
En la simulación 1 no se realizó una interfaz gráfica ya que el manejo de dicha simulación ronda entre los 5 a 8 minutos para ejecutarse por lo cual las pruebas se realizaron en la consola de MATLAB.

Dependiendo del tipo de ruido a evaluar en la imagen nos dirigimos a la línea 229 de la simulación llamada tramaAOS.m y allí como se indica en la figura 7.1 colocamos el tipo de ruido los cuales pueden ser gaussian, salt & pepper o speckle. De igual manera en la misma línea 229 se encuentra el campo donde se debe ingresar el valor de la varianza del ruido a evaluar, dicho campo se encuentra señalado en la figura 7.2.



```
Editor - C:\Users\Jessica\Desktop\simulaciones tesis\tramaAOS.m*
tramaAOS.m* x +
216 - end
217
218
219 %se eliminan los 18 ceros de relleno virtual y se agregan dos ceros al final
220 palabra_codigo_corregida=0*rand(oc_m,8160);
221
222 for i=1:oc_m
223     for j=1:8158
224         palabra_codigo_corregida(i,j)=palabra_codigo(i,j+18);
225     end
226 end
227
228 %se agrega ruido gaussiano
229 palabra_ruido=imnoise(palabra_codigo_corregida,'salt & pepper',.005);
230
231
232
233
```

Figura 7. 1: Simulación 1, consola línea 229 modificación tipo de ruido.



```
216 -   end
217
218
219   %se eliminan lo 18 ceros de relleno virtual y se agregan dos ceros al final
220   palabra_codigo_corregida=0*rand(oc_m,8160);
221
222   for i=1:oc_m
223       for j=1:8158
224           palabra_codigo_corregida(i,j)=palabra_codigo(i,j+18);
225       end
226   end
227
228   %se agrega ruido gaussiano
229   palabra_ruido=imnoise(palabra_codigo_corregida,'salt & pepper',0.005);
230
231
232
233
```

Figura 7. 2: Simulación 1, consola línea 229 modificación valor del ruido.

Una vez modificados los valores indicados en las figuras 7.1 y 7.2 podemos proceder a darle al botón de RUN de la herramienta MATLAB esperar unos minutos y en pantalla nos aparecerá la imagen antes y después de ser sometida al ruido, las cuales aparecen con los nombres de imagen enviada e imagen recibida respectivamente, tal como se observa en la figura 7.3.

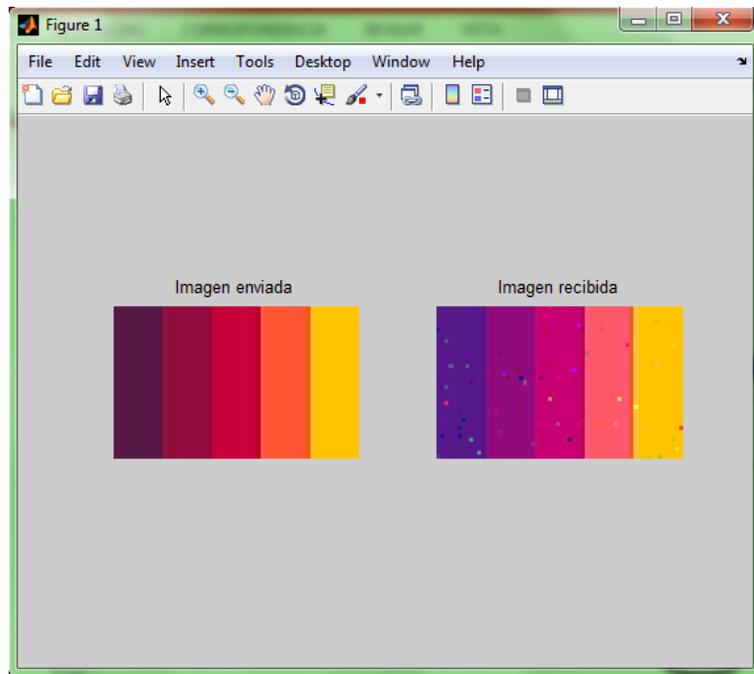


Figura 7. 3: Simulación 1, respuesta para un tipo y valor de ruido determinado.

### 7.3.2 Funcionamiento de la interfaz gráfica de la simulación 2

La simulación 2 la podemos encontrar con el nombre de s1 tanto para el código como para la figura 7.4, en dicha figura podemos apreciar que el único campo a ingresar por el usuario es la letra o letras a la cual se le van a realizar las pruebas. Luego de ingresar la letra o letras se presiona el botón calcular para hacer funcionar la simulación.

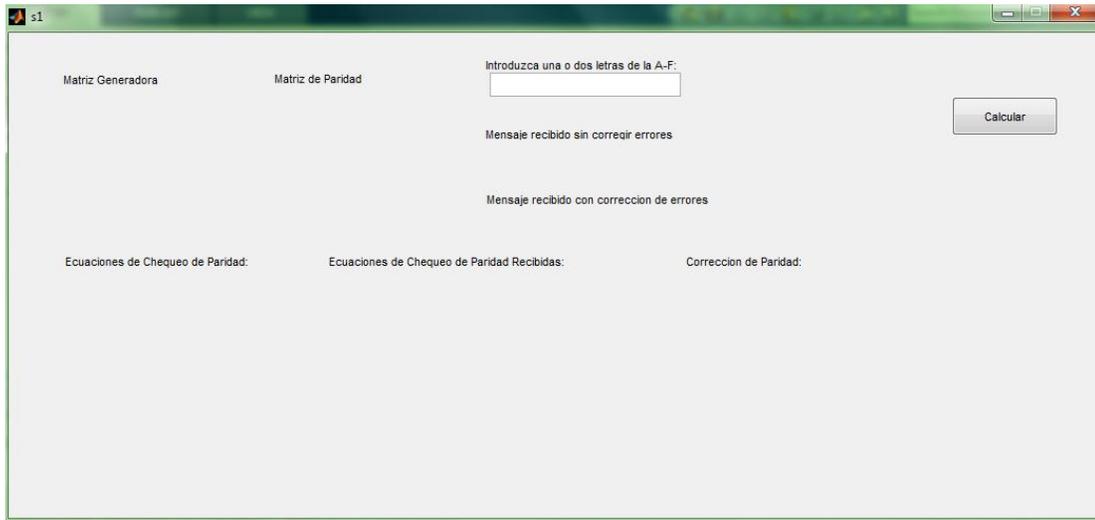


Figura 7. 4: Simulación 2, interfaz gráfica.

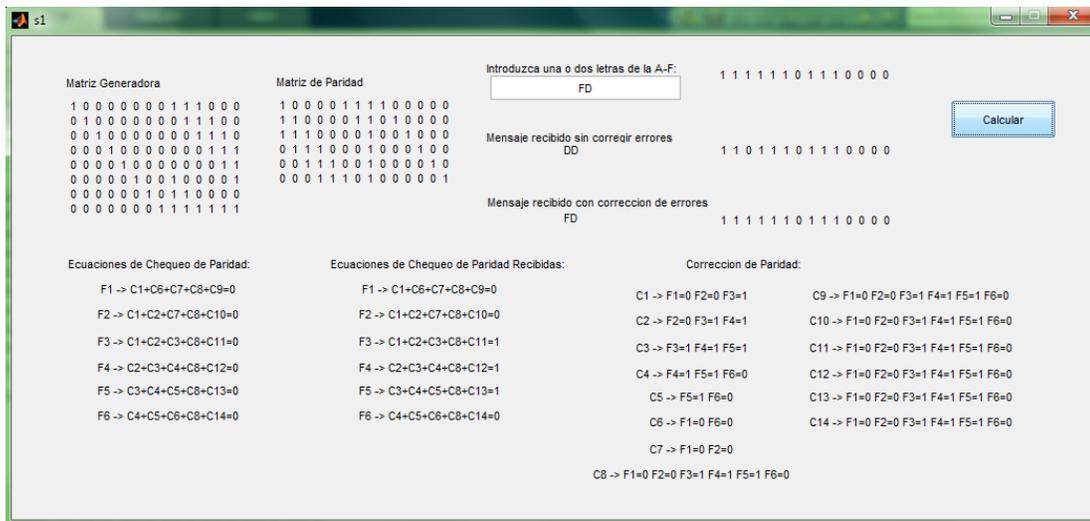


Figura 7. 5: Simulación 2, interfaz gráfica con respuesta a una entrada determinada.

En la figura 7.5 podemos apreciar que las letras ingresadas fueron “FD”, los primeros 8 bits que se encuentran al lado del cuadro donde el usuario ingresa las letras a evaluar son multiplicados por la matriz generadora lo que nos proporciona los 6 bits de paridad siguientes a los primeros 8 bits. La matriz de paridad nos indica las ecuaciones de chequeo de paridad que allí se encuentran, estas se calculan con los bits del mensaje recibido sin corregir errores, se evalúa cuáles de los nodos de chequeo poseen más probabilidad de estar erróneos y se corrigen, los bits ya corregidos se muestran en el mensaje recibido con corrección de errores.

### 7.3.3 Funcionamiento de la interfaz gráfica de la simulación 3

La simulación 3 la podemos encontrar con el nombre de s2 tanto para el código como para la figura 7.6, en dicha figura podemos apreciar que el usuario debe ingresar valor de la varianza a usar ya que en dicha simulación se utiliza el bloque de ruido gaussiano, además de indicar el número de iteraciones, es decir la cantidad de veces que se desea que la simulación repita el procedimiento de detectar y corregir errores. Luego se presiona el botón calcular para hacer correr la simulación y evaluar los resultados.

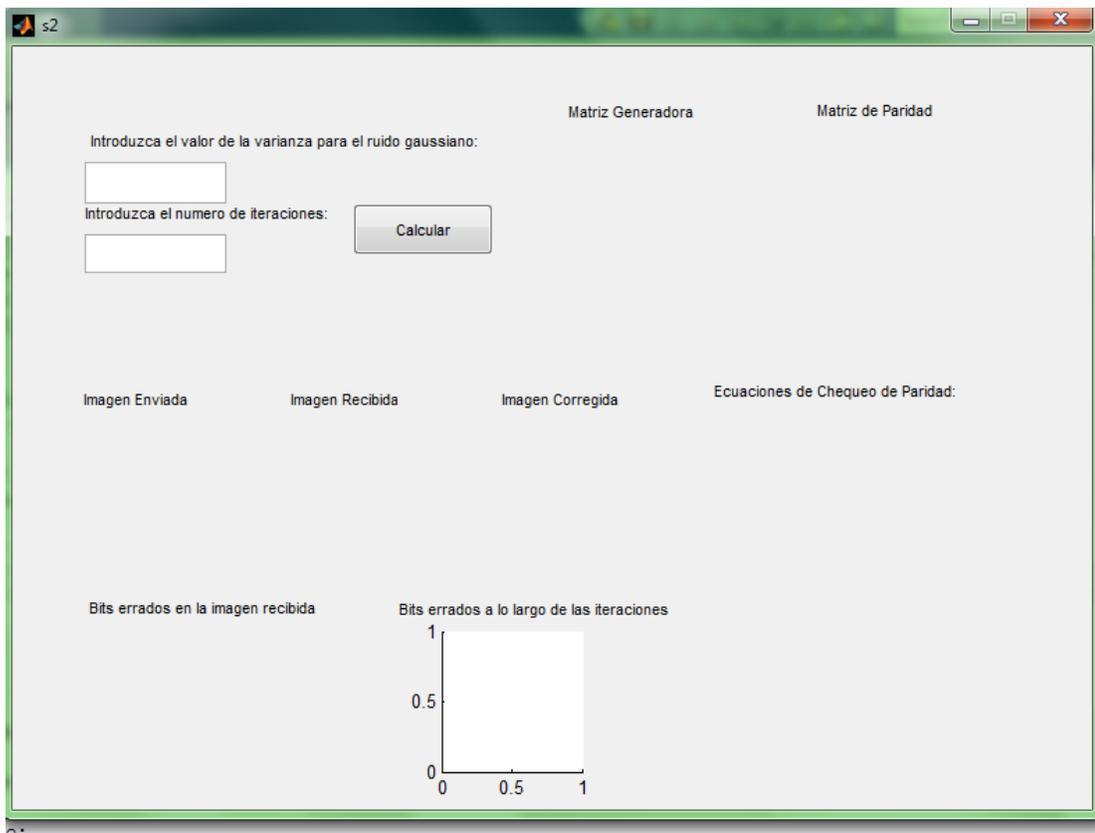


Figura 7. 6: Simulación 3, interfaz gráfica.

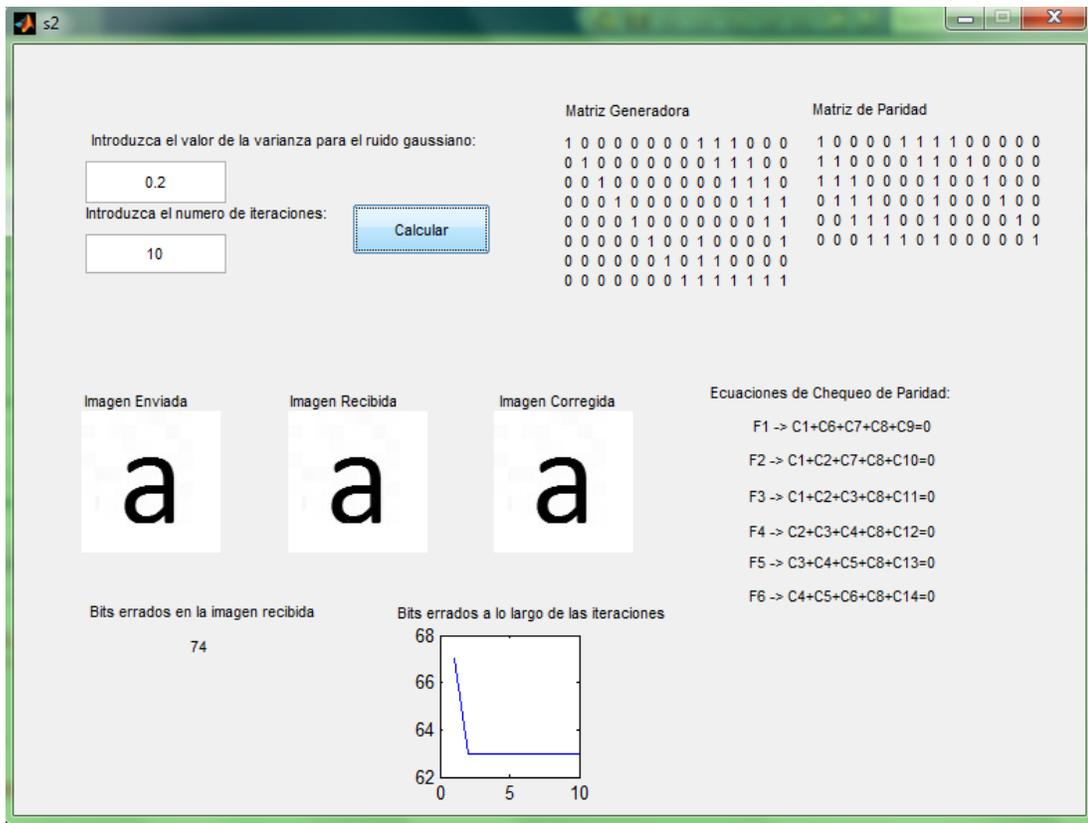


Figura 7. 7: Simulación 3, interfaz gráfica con respuesta a una entrada determinada.

En la figura 7.7 podemos apreciar que luego de correr la simulación aparece la matriz generadora y de paridad utilizadas en la simulación y las ecuaciones de chequeo de paridad generadas por la misma. Para la evaluación de los resultados obtenidos se visualiza la imagen enviada, recibida la cual es la que fue sometida a ruido gaussiano y la imagen corregida la cual es la imagen luego de aplicar el proceso de detección y corrección de errores. Además de las imágenes en sus distintas etapas de la simulación se visualiza la cantidad de bits errados en la imagen recibida y una gráfica que enseña la cantidad de bits errados a lo largo de las iteraciones.

### 7.3.4 Funcionamiento de la interfaz gráfica de la simulación 4

La simulación 4 la podemos encontrar con el nombre de s3 tanto para el código como para la figura 7.8, en dicha figura podemos apreciar que el usuario debe introducir una oración, el valor de la varianza a usar, ya que en dicha simulación se utiliza el bloque de ruido gaussiano, además de indicar el número de iteraciones, es decir la cantidad de veces que se desea que la simulación repita el procedimiento de detectar y corregir errores. Luego se presiona el botón calcular para hacer correr la simulación y evaluar los resultados.

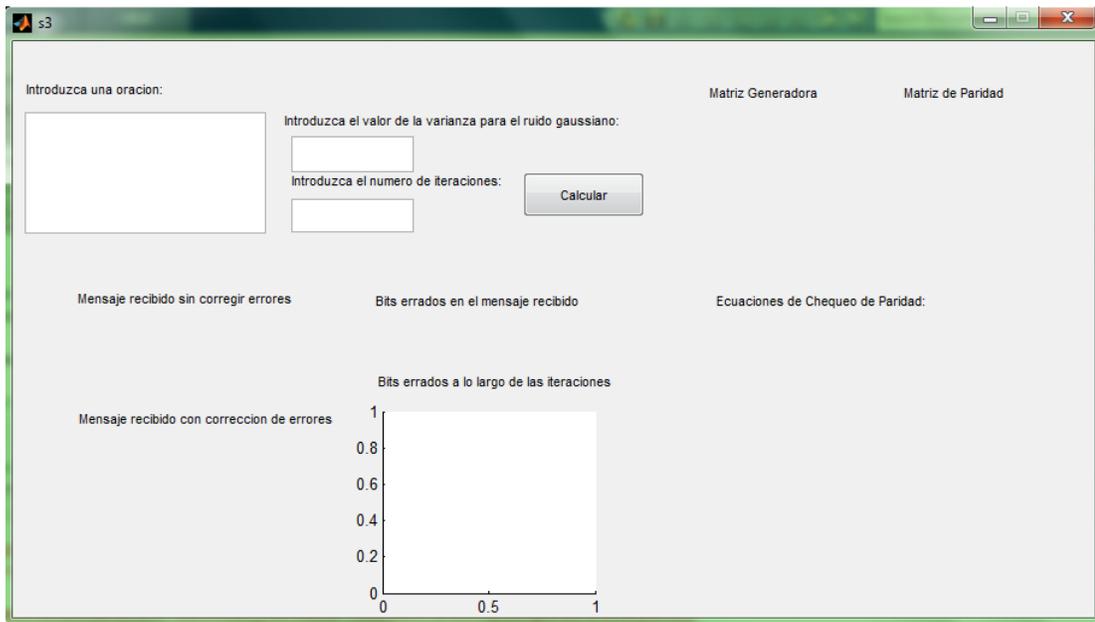


Figura 7. 8: Simulación 4, interfaz gráfica.

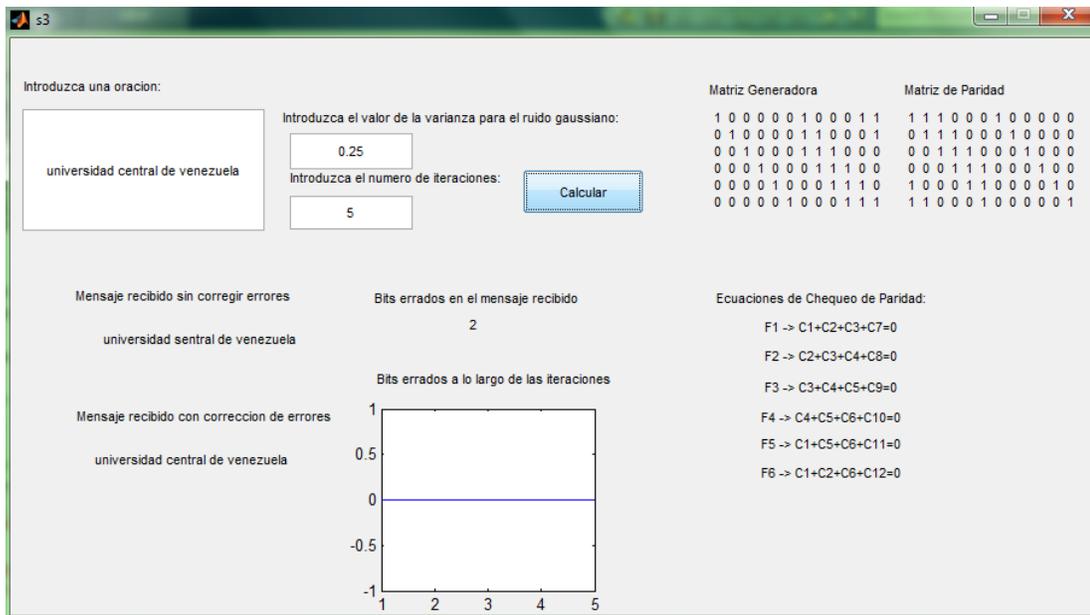


Figura 7. 9: Simulación 4, interfaz gráfica con respuesta a una entrada determinada.

En la figura 7.9 podemos apreciar que luego de correr la simulación aparece la matriz generadora y de paridad utilizadas en la simulación y las ecuaciones de chequeo de paridad generadas por la misma. Para la evaluación de los resultados obtenidos se visualiza el mensaje recibido sin corregir errores y el mensaje recibido con corrección de errores en los cuales puede notarse el error que se produjo en el envío y como luego de la detección y corrección de errores pudo ser arreglado. Además de los mensajes en sus distintas etapas de la simulación se visualiza la cantidad de bits errados en el mensaje recibido y una gráfica que enseña la cantidad de bits errados a lo largo de las iteraciones.

### 7.3.5 Funcionamiento de la interfaz gráfica de la simulación 5

La simulación 5 la podemos encontrar con el nombre de randomizations5 tanto para el código como para la figura 7.10, en dicha figura podemos apreciar que el usuario debe introducir una trama de bits para que el bloque de aleatorización de esta simulación pueda usarlo. Luego se presiona el botón calcular para hacer correr la simulación y evaluar los resultados.

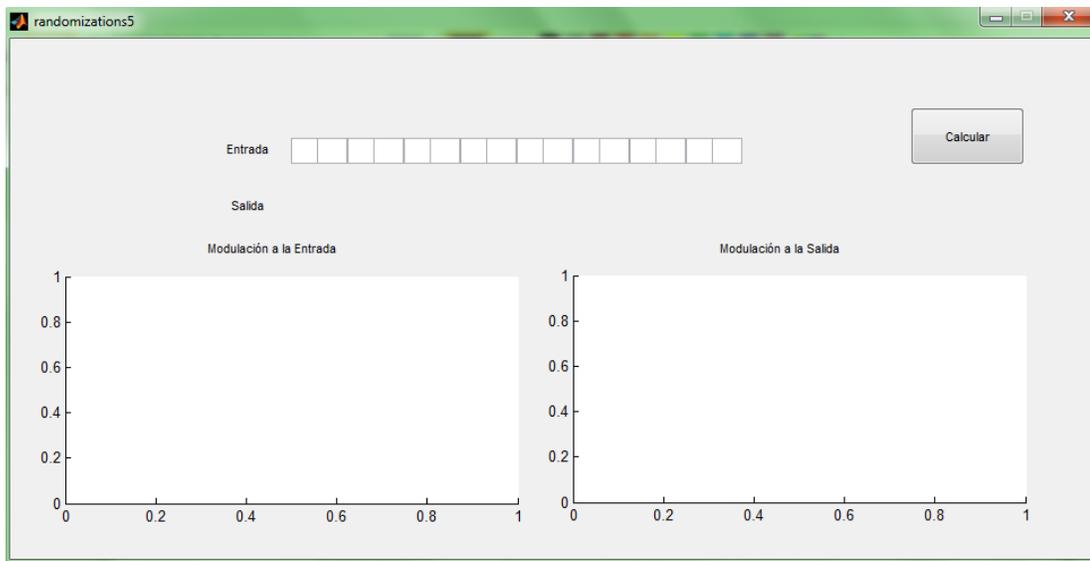


Figura 7. 10: Simulación 5, interfaz gráfica.

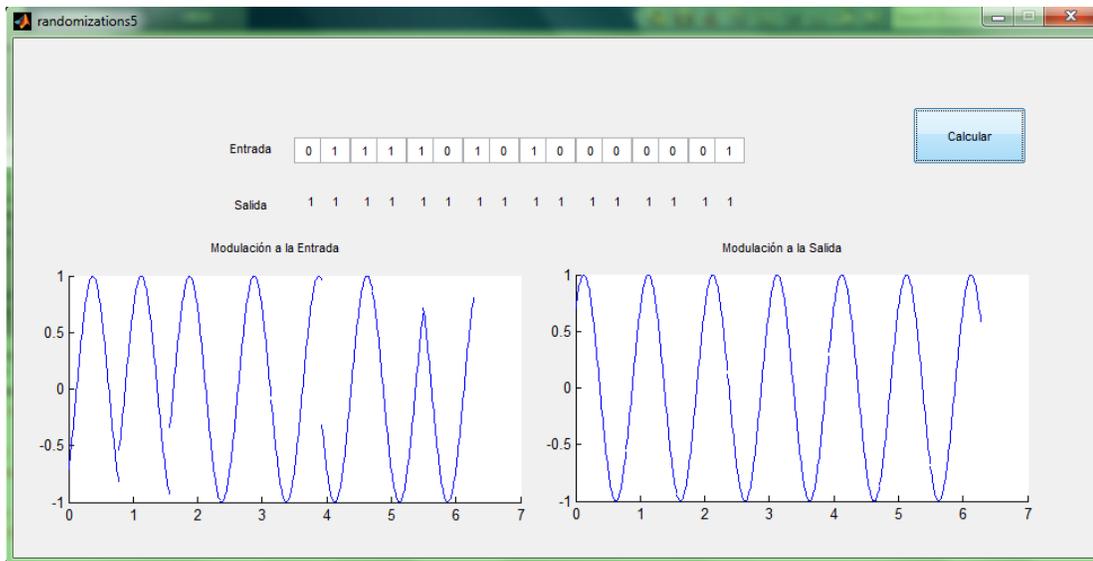


Figura 7. 11 Simulación 5, interfaz gráfica con respuesta a una entrada determinada.

En la figura 7.11 podemos apreciar cómo queda la trama de bits ingresada por el usuario luego de ser procesada por el bloque de aleatorización. Además de visualizar la trama se pueden apreciar dos graficas las cuales son la trama ingresada por el usuario y la trama luego de ser sometida al bloque de aleatorización las dos moduladas con una forma de onda QPSK.

### 7.3.6 Funcionamiento de la interfaz gráfica de la simulación 6

La simulación 6 la podemos encontrar con el nombre de randomizations6 tanto para el código como para la figura 7.12, en dicha figura podemos apreciar que el usuario debe introducir una trama de bits para que el bloque de aleatorización de esta simulación pueda usarlo. Luego se presiona el botón calcular para hacer correr la simulación y evaluar los resultados.

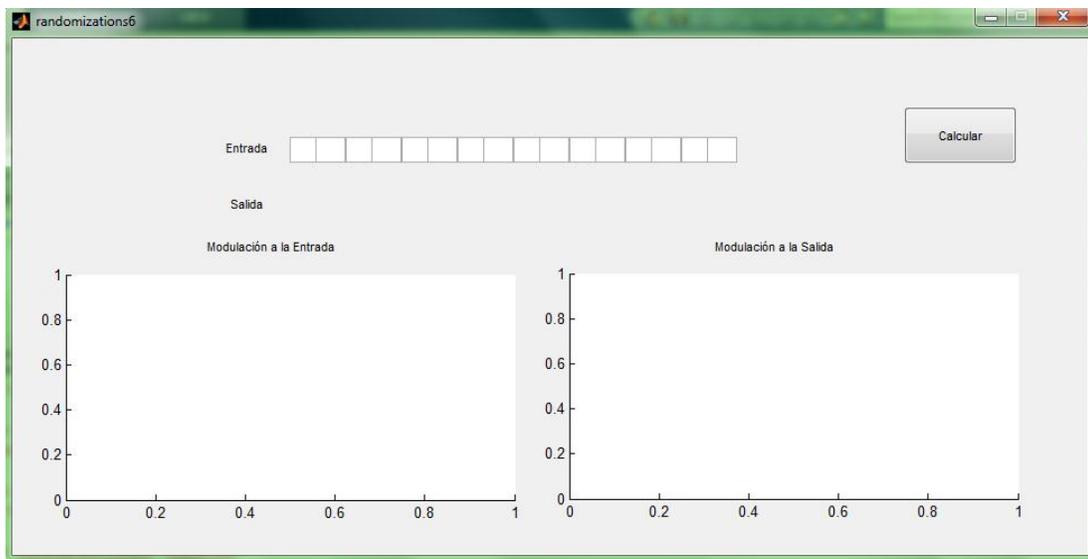


Figura 7. 12: Simulación 6, interfaz gráfica.

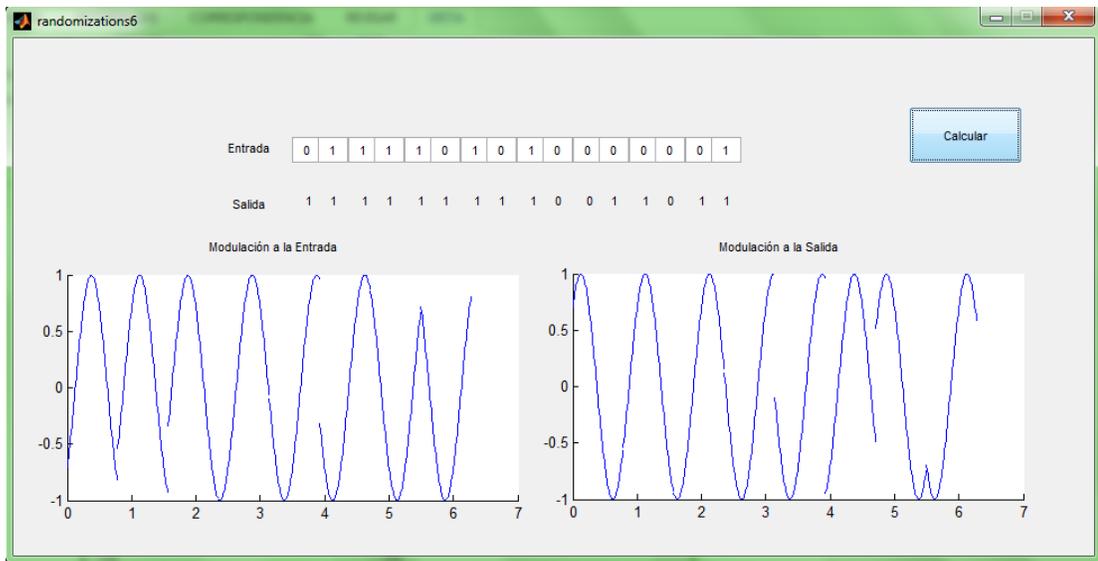


Figura 7. 13: Simulación 6, interfaz gráfica con respuesta a una entrada determinada.

En la figura 7.13 podemos apreciar cómo queda la trama de bits ingresada por el usuario luego de ser procesada por el bloque de aleatorización. Además de visualizar la trama se pueden apreciar dos graficas las cuales son la trama ingresada por el usuario y la trama luego de ser sometida al bloque de aleatorización las dos moduladas con una forma de onda QPSK.

## 7.4 Glosario

**Canal Virtual:** Un mecanismo por el cual un canal físico puede ser compartido por diferentes tipos de usuarios creando múltiples rutas “virtuales” aparentemente paralelas a través del canal.

**Canal Físico:** El espacio-espacio o espacio-tierra medio de transmisión.

**Capa:** Una organización funcional por la cual un sistema distribuido complejo puede dividirse en módulos de servicio relativamente simples.

**Codificación de Bloques:** Transformación de una a una de las secuencia de longitud  $k$  de elementos de un alfabeto fuente en secuencias de longitud  $n$  de elementos de una alfabeto de código,  $n > k$ .

**Encabezado:** Una etiqueta estándar que identifica una estructura de comunicaciones de datos estándar.

**Forma de Onda de Modulación:** Una manera de representar los bits de datos (“1” Y “0”) mediante una forma de onda en particular.

**Misión:** Una actividad en el espacio bajo el control de un proyecto.

**NRZ-L:** Representaciones de formato de datos en las que un “1” de los datos está representado por uno de dos niveles, y un cero de datos se representa por el otro nivel.

**NRZ-M:** Representaciones en formato de datos en las que un “1” de los datos está representado por un cambio de nivel y un “0” de los datos se representa por ningún cambio en el nivel.

**Octeto:** Una palabra de ocho bits.

**Palabra de Código:** En un código de bloque, una de las secuencias de longitud  $n$  en el rango de la transformación uno a uno (véase codificación de bloques).

**Paquete:** Estructura de datos delimitada y de longitud variable.

**Protocolo:** Un conjunto de procedimientos y sus convenciones de formato de habilitación que definen el intercambio ordenado de información entre entidades dentro de una capa dada del sistema de transmisión.

**Señalización:** Conjunto de funciones y servicios prestados dentro de una red para dar formato e intercambiar información de gestión entre elementos de gestión.

**Tasa de Código:** La relación promedio del número de dígitos binarios en la entrada de un codificador con el número de dígitos binarios en su salida.

**Telemetría:** Término utilizado para caracterizar la generación de conjuntos de datos de medición de misión espacial más o menos continuos y predecibles a velocidades y volúmenes de datos que pueden ser extremadamente altos y que tienen una gran interacción con los recursos de comunicación generales.

**Trama:** Unidad de datos de protocolo orientada a la comunicación que facilita la transferencia de unidades de datos de protocolo orientadas a la aplicación a través del enlace espacio-tierra.

**Usuario:** Un proceso humano o inteligente de la máquina que dirige y analiza el progreso de una misión espacial.

**XOR:** Es una puerta lógica digital que implementa el o exclusivo; es decir, una salida verdadera resulta si una, y solo una de las entradas a la puerta es verdadera. Si ambas entradas son falsas o ambas son verdaderas, resulta una salida falsa.

**Zona de Unidades de Datos:** Es la parte en la que pueden colocarse unidades de datos para su transmisión.