ANÁLISIS DEL DESEMPEÑO A NIVEL FÍSICO DEL DECODIFICADOR REED-SOLOMON QUE UTILIZA EL ALGORITMO EUCLIDIANO CON CAPACIDAD DE BORRADO

Juan David Guerrón Elvira
Edilberto Rivera Daza

Director: M.Sc. Harold Armando. Romo Romero

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Grupo de Nuevas Tecnologías en Telecomunicaciones-GNTT
Línea de Investigación Gestión Integrada de Redes, Servicios y Arquitectura de Telecomunicaciones
Popayán, Febrero de 2013

ANÁLISIS DEL DESEMPEÑO A NIVEL FÍSICO DEL DECODIFICADOR REED-SOLOMON QUE UTILIZA EL ALGORITMO EUCLIDIANO CON CAPACIDAD DE BORRADO

Trabajo de Grado presentado como requisito para obtener el título de Ingeniero en Electrónica y Telecomunicaciones

Juan David Guerrón Elvira
Edilberto Rivera Daza

Director: M.Sc. Harold Armando Romo Romero

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Grupo de Nuevas Tecnologías en Telecomunicaciones-GNTT
Línea de Investigación Gestión Integrada de Redes, Servicios y Arquitectura de Telecomunicaciones
Popayán, Febrero de 2013

CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ACRÓNIMOS

8-PSK	8-*Phase Shift Keying,* Modulación por Desplazamiento de 8 fases
4-QAM	4-*Quadrature Amplitude Modulation,* Modulación de Amplitud en Cuadratura de 4 estados.
16-QAM	16-*Quadrature Amplitude Modulation*, Modulación de Amplitud en Cuadratura de 16 estados.
AFM	Algoritmo Forney Modificado.
AWGN	*Additive White Gaussian Noise*, Ruido Gausiano Blanco Aditivo.
ARQ	*Automatic Retransmission Request*, Solicitud de Retransmisión Automática.
BCH	Bose Ray Chaudhuri.
BC	Buscador Chien.
BER	*Bit Error Rate,* Tasa de Error de Bit.
BPSK	*Binary Phase Shift Keying,* Modulación por Desplazamiento de Fase Binaria.
BW	*Bandwidth,* Ancho de Banda.
CD	*Compact Disc*, Disco Compacto.
CIRC	*Cross Interleaved Reed-Solomon Coding*, Codificación RS Entrecruzada.
CNPR	Constructor Nueva Palabra Recibida.
CPB	Constructor Polinomio de Borraduras.
CSM	Calculador de Síndrome Modificado.
DCB	Decodificador con Capacidad de Borrado.
DSL	*Digital Subscriber Line,* Línea de Abonado Digital.
DVB	*Digital Video Broadcasting,* Radiodifusión de Video Digital.
Eb/No	*Energy per Bit to Noise Power Spectral Density Ratio*, Relación de Energía de Bit a Densidad Espectral de Potencia de Ruido.
EPO	Estimador de Palabra Original.

ETSI	*European Telecommunications Standards Institute,* Instituto Europeo de Estándares de Telecomunicaciones.
FEC	*Forward Error Correction,* Corrección de Errores hacia Adelante.
GUI	*Graphics User Interface,* Interfaz Gráfica de Usuario.
MER	*Modulation Error Ratio,* Tasa de Error de Modulación.
MCD	Máximo Común Divisor.
PC	*Personal Computer,* Computador Personal.
RMS	*Root Mean Square,* Valor Cuadrático Medio.
RS	Reed-Solomon.
SER	*Symbol Error Rate*, Tasa de Error de Símbolo.
SNR	*Signal-To-Noise* Ratio, Relación Señal a Ruido.
WiMAX	*Worldwide Interoperability for Microwave Access,* Interoperabilidad Mundial para Acceso por Microondas.

.

RESUMEN

En este documento se presenta el análisis del desempeño a nivel físico del decodificador Reed-Solomon con capacidad de borrado utilizando el algoritmo Euclidiano. Se realiza una descripción de las técnicas de detección y corrección de errores haciendo énfasis en los códigos bloque, para así hacer el estudio de los campos de cuerpo finito o campos de Galois, herramienta esencial en la codificación Reed-Solomon. Posteriormente se realiza el estudio de la Metodología para la Simulación de Equipos de Telecomunicaciones [1] con el objetivo de obtener un modelo de simulación de un sistema de comunicación con codificación y decodificación RS con capacidad de borrado. En la parte final del documento, se muestran los resultados de las diferentes pruebas realizadas al sistema de comunicación implementado en MatLab, que permiten realizar el análisis del desempeño del decodficador RS que utiliza el algoritmo Euclidiano con capacidad de borrado frente al decodificador RS que utiliza el algoritmo Euclidiano sin capacidad de borrado.

INTRODUCCIÓN

Los sistemas de comunicación digital deben ofrecer alta confiabilidad en el proceso de transmisión de señales, debido a esto se ha introducido el concepto de codificación de canal. Uno de los parámetros para evaluar el desempeño de un sistema de comunicación digital es el valor de la Tasa de Error de Bit (BER, *Bit Error Rate*), para un valor en la Relación de Energía de Bit a Densidad Espectral de Potencia de Ruido (Eb/No, *Energy per Bit to Noise Power Spectral Density Ratio*). Así, la codificación de canal es una técnica que permite disminuir la probabilidad de error de bit en el receptor sin necesidad de incrementar la potencia de transmisión.

Las técnicas de codificación de canal consisten en transformar la secuencia de información original en una secuencia que incluyen redundancia, con el objetivo de hacerla menos susceptible a las alteraciones que puede sufrir debido a los fenómenos físicos como ruido, interferencia o multitrayecto. Estas secuencias con redundancia, permiten detectar y corregir errores en el receptor.

Existen dos técnicas para el control de errores, la Solicitud de Retransmisión Automática (ARQ[1], *Automatic Retransmission Query*) y los códigos de Corrección de Errores hacia Adelante (FEC, *Forward Error Correction).* Estos últimos realizan la detección y corrección de errores en el receptor. Esta técnica funciona mediante la adición de bits de redundancia a la secuencia de bits de información.

Teniendo en cuenta la forma como se adiciona la información redundante a la carga útil, los códigos FEC se clasifican en códigos convolucionales y códigos bloque. Los codificadores convolucionales procesan la información de forma serial y continúa. El ingreso de la información al sistema se realiza en forma serial, bit a bit. Los códigos bloque consisten en una estructura fija de datos agrupada en bloques, a la cual, se le adhiere un cierto número de bit llamados bits de redundancia.

Dentro de los códigos bloque se encuentran los códigos lineales[2], de los cuales se derivan los códigos cíclicos, caracterizados porque cada cambio cíclico[3] de una palabra código es

[1] ARQ: El receptor es capaz de reconocer o detectar la presencia de un error, pero no puede corregirlo y se limita a solicitar al transmisor la repetición del mensaje.

[2] Lineales significa que si se suman dos palabras código validas se obtiene otra palabra código válida.

[3] Los cambios cíclicos en una palabra de código es el desplazamiento de la n-tupla que conforma la palabra código, originando otra palabra de código valida.

también una palabra de código válida. Dentro de los códigos cíclicos se encuentran los códigos Bose Ray Chaudhuri (BCH)[4], siendo su principal ventaja la facilidad con que pueden ser decodificados. De estos códigos BCH se deriva el código Reed-Solomon (RS), considerado el más representativo.

Estos códigos RS pueden ser decodificados de manera simple utilizando algoritmos matemáticos para calcular un polinomio que localice las posiciones de error dentro de una palabra de código recibida, sin embargo su capacidad para encontrar errores es determinada por el tamaño del mensaje y su respectiva redundancia. Esta capacidad de corrección es mejorada utilizando un decodificador con capacidad de borrado, el cual trabaja con borraduras, cuya posición es establecida por el demodulador.

Este trabajo de grado se orienta al análisis del desempeño del decodificador RS con capacidad de borrado sobre un canal con ruido Gaussiano aditivo blanco (AWGN, *Additive White Gaussian Noise)* para ello se ha estructurado de la siguiente manera: en el capítulo 1 se presenta la definición de los códigos RS con sus características y aplicaciones, se definen los campos de Galois y se describe el proceso de codificación RS; en el capítulo 2 se muestran cada uno de los pasos necesarios para realizar la decodificación RS utilizando el algoritmo Euclidiano, se presenta la decodificación RS con capacidad de borrado, en la que se tratan aspectos similares a la decodificación RS con algunas variaciones, donde se involucran las posiciones de borradura dadas por el demodulador; posteriormente en el capítulo 3 se estudia la Metodología para la Simulación de Equipos de Telecomunicaciones [1] con el objetivo de obtener un modelo de simulación del sistema; en el capítulo 4 se presentan los resultados de las pruebas realizadas al demodulador modificado y al decodificador RS con capacidad de borrado. Posteriormente se realiza el análisis comparativo del desempeño del decodificador RS con capacidad de borrado frente al decodificador sin tal capacidad; en el capítulo 5 se presenta las conclusiones, trabajos futuros y recomendaciones.

[4] BCH: inventados en 1959 por Hocquenghem , y de forma independiente en 1960 por Bose y Ray Chaudhuri.

1. CODIFICACIÓN REED-SOLOMON

Los códigos Reed-Solomon son códigos cíclicos no binarios, subclase de los códigos Bose-Chauduri-Hocquenghem (BCH). Son códigos no binarios debido a que operan sobre símbolos de *m* bits en lugar de bits individuales. Estos códigos RS fueron diseñados en 1960 por los Ingenieros Irving S. Reed y Gustave Solomon con el fin de contribuir a la teoría de detección y corrección de errores, manteniendo la integridad de la información en las comunicaciones digitales [2].

1.1 APLICACIONES DE LOS CÓDIGOS RS

En la actualidad los códigos RS son utilizados en distintas aplicaciones entre las cuales se encuentran [3]:

- *Almacenamiento Digital de Audio*. Los códigos RS son ampliamente utilizado en los sistemas de almacenamiento digital de audio en Discos Compactos (*CD, Compact Disc*). Los sistemas de CD utilizan Codificación RS Entrecruzada (CIRC, *Cross Interleaved Reed-Solomon Coding*). Este sistema funciona con dos codificadores RS entrecruzados, el primero de ellos se utiliza para detectar las borraduras y el segundo para corregirlas, así logrando mejorar la calidad del sistema de audio de dos canales o estéreo.
- *Sistemas de comunicación en el espacio exterior*. En estos sistemas se logran grandes ganancias de codificación concatenando códigos RS con los códigos convolucionales. El ruido en el canal de trasmisión en el espacio exterior no afecta considerablemente a las ráfagas previamente codificadas con el código RS. Un ejemplo de esta aplicación se observó en la sonda espacial *VOYAGER* la cual exploraba *URANO* y *NEPTUNO, logrando* imágenes de observación espacial de alta resolución.
- *Memorias de almacenamiento*. En los sistemas de almacenamiento de información de los computadores personales se utilizan códigos RS para incrementar la confiabilidad de la información almacenada en discos duros, para su posterior utilización.
- *Sistemas de transmisión*. Los códigos RS son utilizados en la Interoperabilidad Mundial para Acceso por Microondas (WiMAX[5],*Worldwide Interoperability for Microwave Access*), Línea de Abonado Digital (DSL[6], *Digital Subscriber Line*),

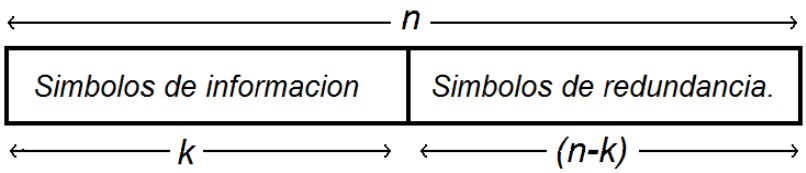
[5]Wimax: Estándar IEEE 802.16; tecnología de acceso que permite la transmisión y recepción de datos por ondas de radio frecuencia.
[6] DSL: Conjunto de tecnologías que proveen conexión digital sobre la línea de abonado.

como también en Radiodifusión de Video Digital (DVB[7], *Digital Video Broadcasting*).

1.2 CÓDIGO REED-SOLOMON

El código Reed-Solomon es un subgrupo de los códigos BCH utilizados para la detección y corrección de errores. Estos códigos funcionan a nivel de bloque en lugar de funcionar a nivel de bit, logrando corregir bloques completos de información, razón por la que son utilizados en una amplia gama de aplicaciones en transmisión digital. El proceso del codificador RS consiste en adicionar símbolos de redundancia al bloque de información. En el proceso inverso realizado por el decodificador, la palabra recibida del canal de comunicaciones, se analiza para detectar y corregir los errores que presente, logrando disminuir la BER. En la figura 1.1 se muestra un bloque codificado con RS de longitud *n* símbolos, junto a una trama de *k* símbolos de mensaje, donde *n* es mayor que *k*, de modo que la diferencia entre *k* y *n* es el número de símbolos de redundancia agregados, los cuales determinan la capacidad de corrección de errores del código $RS(n,k)$[8] [2].

Figura 1.1 Bloque RS $(\boldsymbol{n},\boldsymbol{k})$***.***



1.2.1 Características del código RS

Teniendo en cuenta que el código RS, es un subconjunto de los códigos BCH, cíclico y no binario, posee características similares a los demás códigos bloque lineal. La principal característica del código RS es que opera sobre símbolos y no bits como los demás códigos, donde la longitud de la palabra codificada n se relaciona con el número de bits por símbolo m como se muestra en la ecuación 1.1.

$$n = 2^m - 1 \tag{1.1}$$

La redundancia en un código RS es la diferencia en símbolos de la longitud de la palabra codificada y la palabra mensaje original, como se muestra en la ecuación 1.2, siendo n y k

[7] DVB: Conjunto de estándares de televisión digital.

[8] $RS(n.k)$: es la nomenclatura asociada al código Reed-Solomon con longitud de n símbolos en la palabra codificada y k símbolos de redundancia.

los parámetros de un código $RS(n,k)$, donde n *es* la longitud de la palabra codificada en símbolos y k la longitud del mensaje en símbolos.

$$h = n - k \tag{1.2}$$

Esta adición de símbolos de redundancia aumenta el ancho de banda requerido y disminuye el número de símbolos de información a transmitir. Dos de los parámetros más importantes para comparar un sistema de comunicación digital es la BER y el Ancho de Banda (BW, *Bandwidth*), parámetros que pueden obtener valores deseados a través de la manipulación de la tasa de codificación de los códigos $RS(n,k)$. Esta tasa de codificación se obtiene de la relación entre la longitud de la palabra mensaje y la palabra codificada como se muestra en la ecuación 1.3, e indica que por cada k símbolos de entrada se generan n símbolos codificados.

$$r = \frac{k}{n} \tag{1.3}$$

Se puede determinar la capacidad de corrección de errores en función de la redundancia, siendo la capacidad de corrección de error la cantidad máxima de símbolos que pueden ser corregidos denotada con la letra *t* como se muestra en la ecuación 1.4.

$$t = \frac{(n-k)}{2} \tag{1.4}$$

1.2.2 Distancia mínima

La distancia mínima de un código bloque se define como el mínimo número de posiciones en los cuales difieren dos palabras de código distintas de cero. En los códigos RS se denota con d_{min} y se encuentra en función de la capacidad de corrección de errores como se muestra en la ecuación 1.5.

$$d_{\min} = 2t + 1 \tag{1.5}$$

1.3 CAMPOS DE GALOIS

La codificación RS implica el uso de los campos de cuerpo finito o campos de Galois. Estos campos son llamados así en honor al matemático francés *Évariste Galois,* debido a que él enfocó sus estudios a la teoría de conjuntos y sus conexiones en la teoría de grupos.

Un grupo se define como un conjunto de elementos G, en el cual se ha definido una operación "**.**". Esta operación toma dos elementos del conjunto G y genera como resultado

otro elemento del conjunto, a esta propiedad se le conoce como clausura de la operación. La operación debe cumplir con las propiedades asociativa y conmutativa de las operaciones algebraicas. Además dentro del conjunto de elementos G debe existir el elemento inverso y el elemento identidad. El elemento identidad es aquel que operado con otro elemento del conjunto da como resultado el mismo elemento. El elemento inverso es aquel que operado con cualquier elemento del conjunto da como resultado el elemento identidad. La operación multiplicación en modulo p, presenta inconvenientes, ya que el elemento cero no existe dentro de un grupo debido a que este no tiene elemento inverso. Por esta razón ninguna multiplicación entre elementos debe dar como resultado cero; teniendo en cuenta esta restricción, se debe escoger un p primo [4].

Los campos de Galois se definen como grupo, cuyo conjunto está conformado por un número finito de elementos. Se denota $GF(q) = GF(p^m)$, con $m \geq 1$ y $p \geq 2$, donde m es la longitud en bits del símbolo, q el tamaño del campo y p es un número primo llamado base o módulo del campo finito. Los $GF(q)$ tienen dos operaciones básicas binarias: adición y la multiplicación. Estas dos operaciones deben cumplir con las propiedades conmutativa y asociativa.

1.3.1 Campo binario

El campo binario es el campo finito $GF(p^m)$ con $p = 2$ y $m = 1$. Los elementos de este campo son $G = \{0,1\}$. El interés por el estudio del campo binario se concreta debido a que este es el campo generador de campos de mayor tamaño $GF(q) = GF(2^m)$.

Las operaciones binarias de adición y multiplicación se definen en la tabla 1.1 [5].

Tabla 1.1 (a) Adición binaria.

+	0	1
0	0	1
1	1	0

Tabla 1.1 (b) Multiplicación binaria

*	0	1
0	0	0
1	0	1

Como se observa en la tabla 1.1a, la suma binaria se puede implementar con compuertas lógicas *XOR*. Se define de la misma manera el elemento aditivo identidad como se muestra en la ecuación 1.6.

$$A + I_{add} = I_{add} + A = A \tag{1.6}$$

Donde $I_{add} = 0$.

Se observa que la multiplicación binaria se puede implementar con compuertas lógicas *AND*. Para la multiplicación binaria también se define el elemento multiplicativo identidad como se muestra en la ecuación 1.7.

$$A * I_{mult} = I_{mult} * A = A \tag{1.7}$$

Donde $I_{mult} = 1$.

1.3.2 Polinomios primitivos y extensión del campo binario

Hasta este punto se ha dejado un amplio rango de campos finitos sin describir. Son los campos finitos de la forma $GF(q) = GF(p^m)$. Es posible construir campos de orden $GF(2^m)$ a partir del campo binario $GF(2)$. Esto es conocido como extensión del campo, donde 2 es la base prima y $GF(2)$ un subconjunto de $GF(2^m)$. Para asegurar que la extensión del campo tenga un número finito de elementos se debe cumplir con la condición indicada en la ecuación 1.8.

$$\alpha^{\left(p^m-1\right)} = \alpha^0 = 1 \tag{1.8}$$

La condición anterior señala el carácter cíclico del campo, ya que la extensión del campo solo tiene los elementos indicados en la ecuación 1.9.

$$GF\left(2^m\right) = \{0, 1, \alpha, \alpha^2, \alpha^3, \ldots, \alpha^{p^m-2}\} \tag{1.9}$$

Para generar los elementos de un campo finito se utiliza un polinomio primitivo $p(X)$ de orden m, el cual debe ser irreducible, es decir, no se puede expresar como la multiplicación de dos o más factores. Además, este $p(X)$ divide exactamente al polinomio $X^n + 1$ para $n = p^m - 1$, y no divide a ninguno de los polinomios $X^i + 1$ para $i < n$ [6].

Con $p(X)$ se generan los elementos del campo $GF(2^m)$ utilizando el elemento primitivo[9]. Como ejemplo ilustrativo a lo largo de este capítulo se empleará la construcción de un campo finito de orden $m = 4$. Para esto se empleará el polinomio primitivo indicado en la ecuación 1.10 [7].

[9] Elemento que genera el campo por medio del proceso de la multiplicación recursiva.

$$p(X) = X^4 + X + 1 \tag{1.10}$$

El paso siguiente es determinar si el polinomio indicado es primitivo, verificando si cumple las condiciones anteriormente mencionadas. Para ello, se determina si divide exactamente al polinomio $X^n + 1$, en este caso con $m = 4$ se obtiene que $n = 2^4 - 1 = 15$ por tanto $X^n + 1 = X^{15} + 1$, entonces $p(X)$ debe dividir a $X^{15} + 1$ exactamente, pero no debe dividir a $X^{14} + 1$, $X^{13} + 1$, $X^{12} + 1, \ldots, X + 1$. La división para corroborar que el polinomio $p(X)$ divide exactamente al polinomio $X^{15} + 1$, se muestra en la ecuación 1.11.

$$X^4 + X + 1 \overline{)\; X^{15} + 0X^{14} + 0X^{13} + 0X^{12} + 0X^{11} + \ldots + 1} \;\; \begin{matrix} X^{11} + X^8 + X^7 + X^5 + X^3 + X^2 + X + 1 + r(X) \\ \end{matrix}. \tag{1.11}$$

La operación de la ecuación 1.11 se describe en el anexo A, donde se puede entender el procedimiento de la división binaria entre polinomios, y comprueba que el residuo $r(X) = 0$.

Por tanto cumple la condición mencionada, ya que $X^4 + X + 1$ divide exactamente a $X^{15} + 1$. Realizando el mismo procedimiento se puede comprobar que este polinomio no divide a ninguno de los $X^i + 1$ para todos los $i = 1, 2, 3 \ldots, n - 1$.

Una vez establecido que el polinomio $p(X)$ es primitivo, se define α raíz del polinomio $p(X)$ tal que $p(\alpha) = 0$, como se muestra en la ecuación 1.12.

$$p(\alpha) = \alpha^4 + \alpha + 1 = 0 \tag{1.12}$$

$$\alpha^4 = -\alpha - 1$$

Dado que en los campos de $GF(2^m)$ se opera con aritmética binaria o modulo dos, la ecuación 1.12 se puede reescribir como la ecuación 1.13.

$$\alpha^4 = \alpha + 1 \tag{1.13}$$

Así se puede determinar los demás elementos del campo, en función de los elementos del campo menores a α^4. La forma de construcción recurrente del campo a partir de los elementos del campo menores a α^4, se muestra como ejemplo a partir de las operaciones entre elementos del campo en el anexo A.

Una forma más directa para encontrar elementos del campo $GF(2^4)$, se basa en la formula expresada en la ecuación 1.14.

$$\alpha^i = X^i \ mod \ p(X) \tag{1.14}$$

La ecuación 1.14 representa la operación *mod*[10] de la representación polinomial del elemento que se desea hallar entre el polinomio primitivo.

En la tabla 1.2 se muestran los elementos del campo $GF(2^4) = GF(16)$, donde se utiliza tres tipos de representaciones: la representación simbólica que denota el elemento del campo, la notación polinomial utilizada para operaciones aritméticas, y la notación vectorial utilizada para la codificación RS. Se aclara que la representación polinomial, en algunos casos, para efectos prácticos, suele expresarse en términos de α mientras en otras ocasiones en términos de X.

Tabla 1.2 Construcción de $GF(2^4)$.

Notación simbólica GF(16)	Representación polinomial	Representación vectorial
0	0	0000
1	1	0001
α	α	0010
α^2	α^2	0100
α^3	α^3	1000
α^4	$\alpha + 1$	0011
α^5	$\alpha^2 + \alpha$	0110
α^6	$\alpha^3 + \alpha^2$	1100
α^7	$\alpha^3 + \alpha + 1$	1011
α^8	$\alpha^2 + 1$	0101
α^9	$\alpha^3 + \alpha$	1010
α^{10}	$\alpha^2 + \alpha + 1$	0111
α^{11}	$\alpha^3 + \alpha^2 + \alpha$	1110
α^{12}	$\alpha^3 + \alpha^2 + \alpha + 1$	1111
α^{13}	$\alpha^3 + \alpha^2 + 1$	1101
α^{14}	$\alpha^3 + 1$	1001

1.3.3 Operaciones binarias para extensión de campo $GF(2^m)$

Debido a que $GF(2^m)$ es una extensión del campo binario, las operaciones binarias también pueden ser utilizadas e implementadas, recordando que las operaciones binarias del campo $GF(2^m)$ se pueden implementar con compuertas lógicas *XOR*, siendo las operaciones binarias del campo $GF(2^m)$ de fácil construcción en sistemas embebidos[11]. Debido a que los codificadores y decodificadores *RS*, se basan en la aritmética binaria de

[10] La operación *mod* da como resultado el residuo de la división entre el dividendo y el divisor.
[11] Sistemas microcontrolados diseñados para realizar tareas específicas.

campos de cuerpo finito $GF(2^m)$, es importante el desarrollo de las tablas de multiplicación y adición de los elementos del campo.

Para elaborar una tabla de adición binaria se establece una matriz *S,* donde cada elemento S_{ij} es el resultado de la suma entre elementos α^i, α^jdel campo $GF(2^m)$, así como se muestra en la ecuación 1.15.

$$S_{ij} = \alpha^i + \alpha^j \tag{1.15}$$

Una de las formas de operar la adición binaria es utilizando la representación polinomial de los elementos del campo, expresando los resultados en función de la notación simbólica de la tabla 1.2. Un ejemplo de cómo se debe operar la adición entre dos elementos del campo se da a continuación en la ecuación 1.16, teniendo en cuenta que es aritmética binaria

$$\alpha^{10} + \alpha^7 = \left(\alpha^2 + \alpha + 1\right) + \left(\alpha^3 + \alpha + 1\right) = \alpha^3 + \alpha^2 = \alpha^6 \tag{1.16}$$

Las operaciones completas para la construcción de la tabla de adición binaria se muestran en el anexo A. En la tabla 1.3 se muestra la tabla de adición binaria, la cual es simétrica con respecto a la diagonal, situándose en esta el elemento identidad.

Tabla 1.3 Suma de los elementos del campo GF(16)

+	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
1	0	α^4	α^8	α^{14}	α	α^{10}	α^{13}	α^9	α^2	α^7	α^5	α^{12}	α^{11}	α^6	α^3
α		0	α^5	α^9	1	α^2	α^{11}	α^{14}	α^{10}	α^3	α^8	α^6	α^{13}	α^{12}	α^7
α^2			0	α^6	α^{10}	α	α^3	α^{12}	1	α^{11}	α^4	α^9	α^7	α^{14}	α^{13}
α^3				0	α^7	α^{11}	α^2	α^4	α^{13}	α	α^{12}	α^5	α^{10}	α^8	1
α^4					0	α^8	α^{12}	α^3	α^5	α^{14}	α^2	α^{13}	α^6	α^{11}	α^9
α^5						0	α^9	α^{13}	α^4	α^6	1	α^3	α^{14}	α^7	α^{12}
α^6							0	α^{10}	α^{14}	α^5	α^7	A	α^4	1	α^8
α^7								0	α^{11}	1	α^6	α^8	α^2	α^5	α
α^8									0	α^{12}	α	α^7	α^9	α^3	α^6
α^9										0	α^{13}	α^2	α^8	α^{10}	α^4
α^{10}											0	α^{14}	α^3	α^9	α^{11}
α^{11}												0	1	α^4	α^{10}
α^{12}													0	α	α^5
α^{13}														0	α^2
α^{14}															0

Para realizar la tabla de multiplicación binaria del campo $GF(2^m)$ se debe crear una matriz *M,* donde cada elemento M_{ij} es el resultado de la multiplicación entre elementos α^i, α^jdel campo $GF(2^m)$, si la suma de sus exponentes es menor que $2^m - 1$. Dado que los campos de cuerpo finito $GF(2^m)$ son de característica cíclica, es posible multiplicar elementos cuya suma de exponentes es mayor a $2^m - 2$, como se observa en la ecuación 1.17, con el objetivo de expresar el resultado en uno de los elementos del campo.

$$\alpha^i . \alpha^j = \alpha^{(i+j) \bmod n} \tag{1.17}$$

Para poder entender mejor se muestra un ejemplo con dos elementos del campo $GF(16)$ de la tabla 1.2. Este cálculo se muestra en la ecuación 1.18 [6].

$$\alpha^5 . \alpha^4 = \alpha^{19 \bmod n} = \alpha^{19 \bmod 15} = \alpha^4 \tag{1.18}$$

En la tabla 1.4 se muestra la tabla de multiplicación binaria del campo $GF(16)$. Los demás resultados de las multiplicaciones de elementos se dejan como practica para el lector.

Tabla 1.4 Multiplicación de los elementos del campo GF(16).

×	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
1	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
α		α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1
α^2			α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α
α^3				α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^2
α^4					α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^2	α^3
α^5						α^{10}	α^{11}	α^{12}	α^{13}	α^{14}	1	α	α^2	α^3	α^4
α^6							α^{12}	α^{13}	α^{14}	1	α	α^2	α^3	α^4	α^5
α^7								α^{14}	1	α	α^2	α^3	α^4	α^5	α^6
α^8									α	α^2	α^3	α^4	α^5	α^6	α^7
α^9										α^3	α^4	α^5	α^6	α^7	α^8
α^{10}											α^5	α^6	α^7	α^8	α^9
α^{11}												α^7	α^8	α^9	α^{10}
α^{12}													α^9	α^{10}	α^{11}
α^{13}														α^{11}	α^{12}
α^{14}															α^{13}

1.4 CODIFICACIÓN RS

El esquema general para construir una palabra código completa en un codificador $RS(n, k)$, se muestra en la ecuación 1.19.

$$C(X) = X^{n-k}M(X) + CK(X) \tag{1.19}$$

Expresada en sus componentes como se muestra en la ecuación 1.20

$$C(X) = C_{n-1}X^{n-1} + C_{n-2}X^{n-2} + \ldots + C_1X + C_0 \tag{1.20}$$

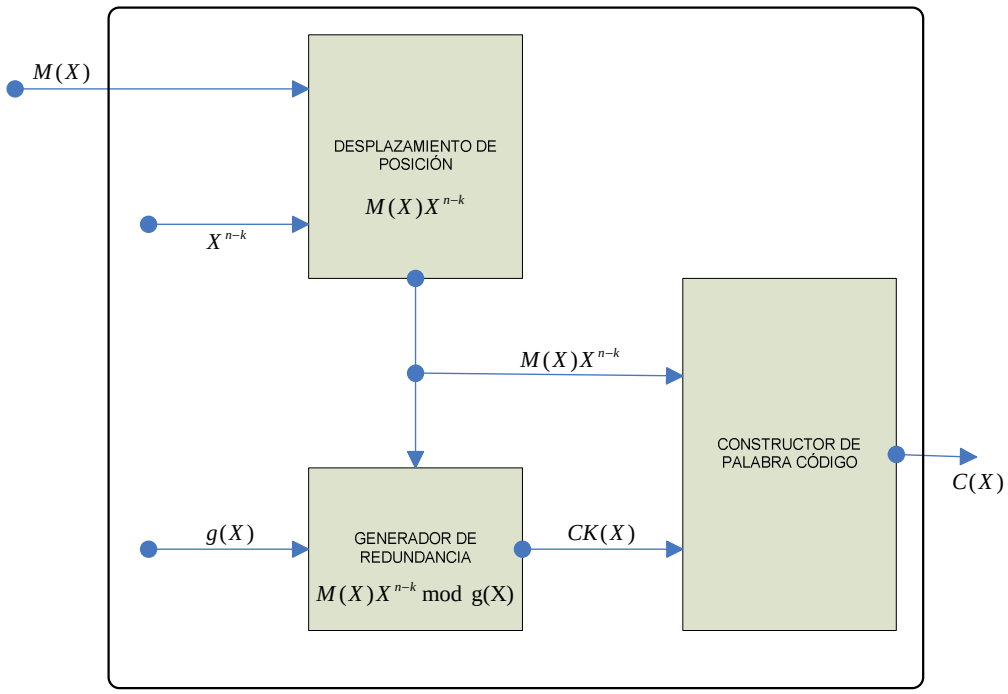
La finalidad del término X^{n-k} es dar un corrimiento a los símbolos de información para así concatenar el bloque de información con el bloque de redundancia. En la ecuación 1.19 se observa que la palabra codificada se conforma de dos partes, la primera contiene el mensaje $M(X)$ conocido como bloque información y la segunda corresponde a los símbolos añadidos conocido como bloque de redundancia.

El bloque de redundancia $CK(X)$ es el residuo de dividir $M(X)X^{n-k}$ entre el polinomio generador $g(X)$[12], como se muestra en la ecuación 1.21.

$$CK(X) = \left[M(X)X^{n-k}\right] \; mod \; g(X) \tag{1.21}$$

En la figura 1.2 se muestra el diagrama en bloques del codificador *RS*.

Figura 1.2 Diagrama en bloques del codificador $RS(n, k)$.



[12] Polinomio generador del código $RS(n, k)$.

1.4.1 Polinomio generador

Para obtener el bloque de redundancia $CK(X)$ y adicionarlo al bloque mensaje es necesario primero determinar el polinomio generador $g(X)$, el cual está determinado por la ecuación 1.22.

$$g(X) = \prod_{i=FCR}^{FCR+2t-1} (X + (\alpha^G)^i) \tag{1.22}$$

Donde FCR es la potencia de la primera raíz consecutiva en $g(X)$, $i = 1,2,\ldots,2t$ y α^G es un elemento primitivo de $p(X)$. Para el caso de estudio en cuestión $FCR = 1$, valor que se ha tomado por convención y $\alpha^G = \alpha$. Por lo tanto la ecuación 1.22 se puede expresar como se muestra en la ecuación 1.23.

$$g(X) = \prod_{i=1}^{2t} (X + \alpha^i) = (X + \alpha)(X + \alpha^2)..(X + \alpha^{2t}) \ . \tag{1.23}$$

Como ejemplo para el código $RS(15,9)$, y a partir de los elementos del campo $GF(16)$, consignados en la tabla 1.2, se tiene que la redundancia es $2t = n - k = 15 - 9 = 6$, y la capacidad de corrección de error es $t = 3$, por tanto el polinomio generador se obtiene con la ecuación 1.24.

$$g(X) = (X - \alpha^1)(X - \alpha^2)(X - \alpha^3)(X - \alpha^4)(X - \alpha^5)(X - \alpha^6) \tag{1.24}$$

Con la ayuda de las tablas de adición y multiplicación del campo $GF(16)$, el polinomio generador es reducido a la ecuación 1.25.

$$g(X) = X^6 + \alpha^{10} X^5 + \alpha^{14} X^4 + \alpha^4 X^3 + \alpha^6 X^2 + \alpha^9 X + \alpha^6 \tag{1.25}$$

1.4.2 Cálculo de una palabra de código

Para un código $RS(15,9)$, se tiene un factor de corrimiento $X^{(n-k)} = X^{(15-9)} = X^6$

Sea una palabra mensaje $M(X) = 0X^8 + 0X^7 + 0X^6 + 0X^5 + 0X^4 + 0X^3 + 0X^2 + \alpha^{11}X + 0 = \alpha^{11}X$, el polinomio se estructura tomando el último término como el término independiente, es decir, $0X^0$, el segundo término es $\alpha^{11}X$ y así sucesivamente se enumeran las potencias hasta $0X^8$. Este mensaje corresponde a la secuencia binaria [0000 0000 0000 0000 0000 0000 0000 1110 0000], cuya representación vectorial es [0 0 0 0 0 0 0 α^{11} 0]. Teniendo la palabra mensaje de prueba anteriormente mencionada y

conociendo los parámetros del código $RS(15,9)$, se puede hallar el bloque de información desplazado mostrado en la ecuación 1.26.

$$M\left(X\right)X^{n-k} = (\alpha^{11}X)(X^{6}) = \alpha^{11}X^{7} \tag{1.26}$$

El bloque de redundancia $CK(X)$ puede requerir un procedimiento más complicado debido a la operación $mod\ g(X)$, pero teniendo en cuenta las tablas de adición y multiplicación del campo $GF(16)$, este procedimiento se simplifica. El resultado de esta operación se muestra en la expresión 1.27, donde se aprecia que el bloque de redundancia tiene grado $n-k-1$.

$$\begin{gathered} M\left(X\right)X^{(n-k)} mod\ g\left(X\right) = \alpha^{11}X^{7} mod\ g(X) \\ M\left(X\right)X^{(n-k)} mod\ g\left(X\right) = \alpha^{8}X^{5} + \alpha^{10}X^{4} + \alpha^{4}X^{3} + \alpha^{14}X^{2} + \alpha^{8}X + \alpha^{12} \end{gathered} \tag{1.27}$$

Una vez obtenidos los dos bloques que conforman la palabra codificada se procede a su respectiva construcción, utilizando la ecuación 1.19. Este resultado se muestra a continuación 1.28.

$$\begin{aligned} C(X) = {} & 0X^{14} + 0X^{13} + 0X^{12} + 0X^{11} + 0X^{10} + 0X^{9} + 0X^{8} \\ & + \alpha^{11}X^{7} + \alpha^{8}X^{5} + \alpha^{10}X^{4} + \alpha^{4}X^{3} + \alpha^{14}X^{2} + \alpha^{8}X + \alpha^{12} \end{aligned} \tag{1.28}$$

2. DECODIFICACIÓN RS CON CAPACIDAD DE BORRADO

La decodificación RS se entiende como el proceso inverso a la codificación, en el cual se detectan y se corrigen errores introducidos en la palabra codificada $C(X)$, durante la transmisión en el canal de comunicaciones. La decodificación RS consiste en detectar y corregir errores utilizando algoritmos de decodificación, y la decodificación RS con capacidad de borrado en comparación a la decodificación RS, tiene la característica adicional de procesar algunos símbolos como borraduras, considerados símbolos no fiables[13] por el demodulador, de los cuales se conoce su posición en la palabra de código recibida.

2.1 DECODIFICACIÓN RS

En principio la palabra recibida en el extremo receptor $R(X)$ es la palabra originalmente codificada, la cual es alterada por un patrón de error $E(X)$ como se presenta en la ecuación 2.1.

$$R(X) = C(X) + E(X) \tag{2.1}$$

Donde el patrón de error está constituido por v errores asociados a las posiciones $X_v = \alpha^{i_v}$.

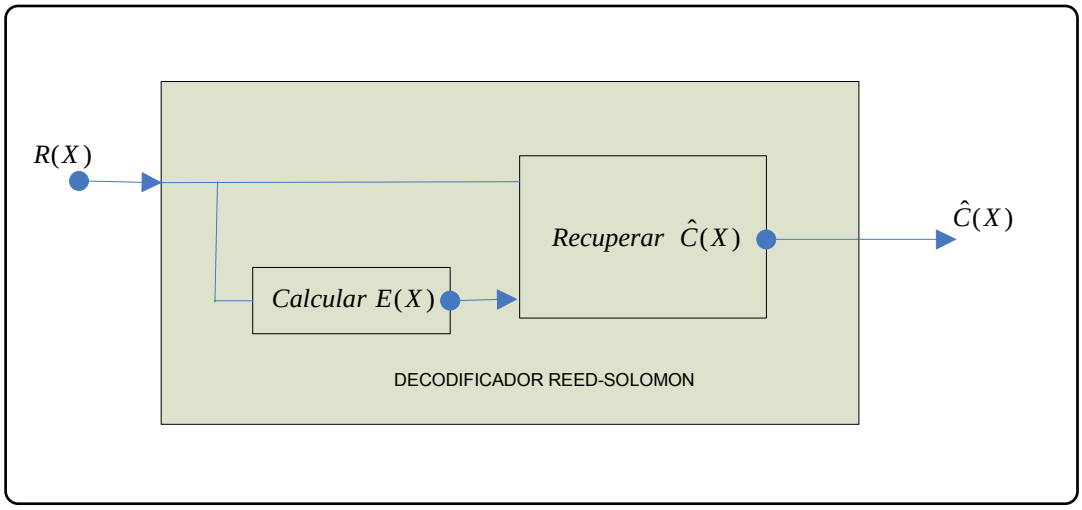
De esta manera, el proceso de decodificación consiste en encontrar las posiciones y valores de dichos errores que constituyen el patrón de error $E(X)$, para lo cual se presenta el proceso general de la decodificación RS constituido por 5 etapas principales [8]:

- Cálculo del polinomio síndrome.
- Cálculo de polinomio localizador de error.
- Cálculo de las posiciones de error con el buscador Chien.
- Cálculo de los valores de error.
- Cálculo de la palabra originalmente codificada.

En la figura 2.1 se presenta el diagrama en bloques del decodificador $RS\ (n, k)$.

[13] Símbolo del cual no se tiene certeza de su correcta demodulación.

Figura 2.1 Diagrama en bloques del decodificador $RS(n, k)$.



2.1.1 Componentes del polinomio síndrome

El polinomio síndrome permite verificar a partir de los símbolos de redundancia si una palabra recibida es válida o nó [9].

Una vez obtenida la palabra recibida $R(X)$ es necesario verificar si existen errores o nó. Teniendo en cuenta que la palabra codificada es un múltiplo del polinomio generador en un factor $Q(X)$, relacionado por la ecuación 2.2.

$$C(X) = Q(X)g(X) \tag{2.2}$$

Las raíces de la palabra recibida $R(X)$ deben ser las $2t$ raíces del polinomio generador $g(X)$. De esta forma la palabra codificada recibida evaluada en las raíces del polinomio generador deben dar como resultado cero, a menos que dicha palabra presente errores, convirtiéndola en una palabra no válida.

De esta manera, si al evaluar α^i con $i = FCR, FCR + 1, \dots, 2t + FCR - 1$ en la ecuación 2.1 el resultado de $R(\alpha^i) = 0$, se entiende que no existen errores y es válida la palabra, pero si $R(\alpha^i) \neq 0$ existen errores y dicho valor se toma como un componente de síndrome denotado por $S_i = R(\alpha^i)$, de manera que al obtener la totalidad de los componentes del polinomio síndrome se forma el polinomio síndrome con la ecuación 2.3.

$$S(X) = \sum_{i=1}^{2t} S_i X^i \tag{2.3}$$

Para ilustrar este procedimiento con un código $RS(15,9)$, se supone una palabra recibida $R(X) = X^8 + \alpha^{11}X^7 + \alpha^8X^5 + \alpha^{10}X^4 + \alpha^4X^3 + \alpha^3X^2 + \alpha^8X + \alpha^{12}$, de la cual se obtienen los siguientes componentes del polinomio síndrome y el respectivo polinomio síndrome.

para $X=\alpha$

$$S_1 = R(\alpha) = (\alpha)^8 + \alpha^{11}(\alpha)^7 + \alpha^8(\alpha)^5 + \alpha^{10}(\alpha)^4 + \alpha^4(\alpha)^3 + \alpha^3(\alpha)^2 + \alpha^8(\alpha) + \alpha^{12}$$
$$S_1 = (\alpha^8 + \alpha^3 + \alpha^{13}) + (\alpha^{14} + \alpha^7 + \alpha^5) + (\alpha^9 + \alpha^{12})$$
$$S_1 = 0 + \alpha^2 + \alpha^8$$
$$S_1 = 1$$

para $X=\alpha^2$

$$S_2 = R(\alpha^2) = (\alpha^2)^8 + \alpha^{11}(\alpha^2)^7 + \alpha^8(\alpha^2)^5 + \alpha^{10}(\alpha^2)^4 + \alpha^4(\alpha^2)^3 + \alpha^3(\alpha^2)^2 + \alpha^8(\alpha^2) + \alpha^{12}$$
$$S_2 = \alpha + \alpha^{11}\alpha^{14} + (\alpha^8\alpha^{10} + \alpha^{10}\alpha^8) + \alpha^4\alpha^6 + \alpha^3\alpha^4 + \alpha^8\alpha^2 + \alpha^{12}$$
$$S_2 = \alpha + \alpha^{10} + 0 + \alpha^{10} + \alpha^7 + \alpha^{10} + \alpha^{12}$$
$$S_2 = \alpha + \alpha^7 + \alpha^{10} + \alpha^{12}$$
$$S_2 = \alpha^{14} + \alpha^3$$
$$S_2 = 1$$

para $X=\alpha^3$

$$S_3 = R(\alpha^3) = (\alpha^3)^8 + \alpha^{11}(\alpha^3)^7 + \alpha^8(\alpha^3)^5 + \alpha^{10}(\alpha^3)^4 + \alpha^4(\alpha^3)^3 + \alpha^3(\alpha^3)^2 + \alpha^8(\alpha^3) + \alpha^{12}$$
$$S_3 = (\alpha^{12})(\alpha^{12}) + \alpha^{11}(\alpha^9.\alpha^9.\alpha^3) + \alpha^8(\alpha^{15}) + \alpha^{10}(\alpha^{12}) + \alpha^4(\alpha^9) + \alpha^3(\alpha^6) + \alpha^8(\alpha^3) + \alpha^{12}$$
$$S_3 = \alpha^9 + \alpha^2 + \alpha^8 + \alpha^7 + \alpha^{13} + \alpha^9 + \alpha^{11} + \alpha^{12}$$
$$S_3 = (\alpha^9 + \alpha^2) + (\alpha^8 + \alpha^7) + (\alpha^{13} + \alpha^9) + (\alpha^{11} + \alpha^{12})$$
$$S_3 = \alpha^{11} + \alpha^{11} + \alpha^{10} + 1$$
$$S_3 = \alpha^{10} + 1$$
$$S_3 = \alpha^5$$

para $X=\alpha^4$

$$R\left(\alpha^4\right)=(\alpha^4)^8+\alpha^{11}(\alpha^4)^7+\alpha^8\left(\alpha^4\right)^5+\alpha^{10}(\alpha^4)^4+\alpha^4(\alpha^4)^3+\alpha^3(\alpha^4)^2+\alpha^8(\alpha^4)+\alpha^{12}$$

$$S_4=\alpha^{32}+\alpha^{39}+\alpha^{28}+\alpha^{10}.\alpha^{16}+\alpha^4.\alpha^{12}+\alpha^3.\alpha^8+\alpha^8.\alpha^4+\alpha^{12}$$

$$S_4=\alpha^{30}.\alpha^2+\alpha^{30}.\alpha^9+\alpha^{15}.\alpha^{13}+\alpha^{11}.\alpha^{15}+\alpha^{15}.\alpha+\alpha^{11}+\alpha^{12}+\alpha^{12}$$

$$S_4=\alpha^2+\alpha^9+\alpha^{13}+\alpha^{11}+\alpha+\alpha^{11}+\alpha^{12}+\alpha^{12}$$

$$S_4=\alpha^2+\alpha^9+\alpha^{13}+\alpha$$

$$S_4=\alpha^{11}+\alpha^{12}$$

$$S_4=1$$

para $X=\alpha^5$

$$S_5=R\left(\alpha^5\right)=(\alpha^5)^8+\alpha^{11}(\alpha^5)^7+\alpha^8\left(\alpha^5\right)^5+\alpha^{10}(\alpha^5)^4+\alpha^4(\alpha^5)^3+\alpha^3(\alpha^5)^2+\alpha^8(\alpha^5)+\alpha^{12}$$

$$S_5=\left(\alpha^{30}.\alpha^{10}\right)+\alpha^{11}\left(\alpha^{30}.\alpha^5\right)+\alpha^8\left(\alpha^{15}.\alpha^{10}\right)+\alpha^{10}\left(\alpha^{15}.\alpha^5\right)+\alpha^4(1)+\alpha^3\left(\alpha^{10}\right)+\alpha^{13}+\alpha^{12}$$

$$S_5=\alpha^{10}+\alpha^{11}.\alpha^5+\alpha^8.\alpha^{10}+\alpha^{10}.\alpha^5+\alpha^4+\alpha^{13}+\alpha^{13}+\alpha^{12}$$

$$S_5=\left(\alpha^{10}+\alpha\right)+\left(\alpha^3+1\right)+(\alpha^4+\alpha^{12})$$

$$S_5=\alpha^8+\alpha^{14}+\alpha^6$$

$$S_5=\alpha^6+\alpha^6$$

$$S_5=0$$

para $X=\alpha^6$

$$S_6=(\alpha^6)^8+\alpha^{11}(\alpha^6)^7+\alpha^8\left(\alpha^6\right)^5+\alpha^{10}(\alpha^6)^4+\alpha^4(\alpha^6)^3+\alpha^3(\alpha^6)^2+\alpha^8(\alpha^6)+\alpha^{12}$$

$$S_6=\left(\alpha^{45}.\alpha^3\right)+\alpha^{11}\left(\alpha^{30}.\alpha^{12}\right)+\alpha^8\left(\alpha^{30}\right)+\alpha^{10}\left(\alpha^{15}.\alpha^9\right)+\alpha^4\left(\alpha^{15}\alpha^3\right)+\alpha^3\left(\alpha^{12}\right)+\alpha^{14}+\alpha^{12}$$

$$S_6=\alpha^3+\alpha^{11}\left(\alpha^{12}\right)+\alpha^8+\alpha^{10}\left(\alpha^9\right)+\alpha^4\left(\alpha^3\right)+\alpha^{15}+(\alpha^{14}+\alpha^{12})$$

$$S_6=\alpha^3+\alpha^{15}.\alpha^8+\alpha^8+\alpha^{15}.\alpha^4+\alpha^7+1+\alpha^5$$

$$S_6=\alpha^3+\alpha^8+\alpha^8+\alpha^4+\alpha^7+1+\alpha^5$$

$$S_6=\alpha^3+0+\alpha^4+\alpha^7+1+\alpha^5$$

$$S_6=\alpha^7+\alpha^7+1+\alpha^5$$

$$S_6=1+\alpha^5$$

$$S_6=\alpha^{10}$$

Por tanto

$$S_1 = 1 \qquad S_3 = \alpha^5 \qquad S_5 = 0$$
$$S_2 = 1 \qquad S_4 = 1 \qquad S_6 = \alpha^{10}$$

Donde el polinomio síndrome se observa en la ecuación 2.4.

$$S(X) = \sum_{i=1}^{2t} S_i X^i = S_1 X + S_2 X^2 + S_3 X^3 + S_4 X^4 + S_5 X^5 + S_6 X^6$$
$$S(X) = \alpha^{10} X^6 + X^4 + \alpha^5 X^3 + X^2 + X \tag{2.4}$$

2.1.2 Polinomio localizador de error

Una vez determinado el polinomio síndrome, se procede a encontrar el polinomio localizador de error $\sigma(X)$ y su recíproco conocido como $\sigma_r(X)$. De este último se obtienen las raíces llamadas números localizadores de error Zi. El grado de $\sigma_r(X)$ determina el número de símbolos errados. En la ecuación 2.5 se presenta la relación entre $\sigma_r(X)$ y $\sigma(X)$, donde v corresponde al número de errores.

$$\sigma_r(X) = X^v \sigma(X^{-1}) \tag{2.5}$$

El polinomio localizador de error se define en la ecuación 2.6.

$$\sigma(X) = \prod_{l=1}^{v} (1 - X_l X) = \sigma_v X^v + \sigma_{v-1} X^{v-1} + \ldots + \sigma_1 X + \sigma_0 \tag{2.6}$$

Donde $\sigma_0 = 1$. Por definición si $X = X_l^{-1}$ entonces $\sigma(X) = 0$; por lo tanto las raíces del polinomio localizador de error (en aritmética de campo finito), están en los recíprocos de las posiciones de error.

Existe una relación lineal entre los coeficientes del polinomio síndromes y los coeficientes del polinomio localizador de error como se observa en la ecuación 2.7 [10].

$$S_j = -\sum_{i=1}^{v} \sigma_i S_{j-i} \tag{2.7}$$

La ecuación 2.7 puede ser expresada en forma matricial, como se observa en la ecuación 2.8.

$$\begin{bmatrix} S_1 & S_2 & \vdots & S_v \\ S_2 & S_3 & \cdots & S_{v+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_v & S_{v+1} & \cdots & S_{2v-1} \end{bmatrix} \begin{bmatrix} \sigma_v \\ \sigma_{v-1} \\ \vdots \\ \sigma_1 \end{bmatrix} = - \begin{bmatrix} S_{v+1} \\ S_{v+2} \\ \vdots \\ S_{2v} \end{bmatrix} \tag{2.8}$$

La ecuación 2.8 es llamada ecuación clave, la cual se resuelve utilizando métodos de solución del algebra lineal, sin embargo es un proceso dispendioso e ineficiente para códigos con una alta capacidad de corrección. Para dar solución a este problema se cuenta con métodos matemáticos como el algoritmo Euclidiano que determinan los coeficientes del polinomio localizador de error de forma más eficiente.

2.1.3 Algoritmo Euclidiano

El algoritmo Euclidiano llamado frecuentemente como el algoritmo de Sugiyama[14] por ser quien realizó la adaptación para solucionar la ecuación clave, es un algoritmo iterativo para encontrar el Máximo Común Divisor (MCD) entre dos polinomios.

El problema de decodificación está enfocado en encontrar el polinomio evaluador de error $\Lambda(X)$ que satisfaga la ecuación 2.9.

$$\Lambda(X) = [1 + S(X)]\sigma(X) \ mod \ X^{2t+1} \tag{2.9}$$

Para esto se debe tener en cuenta que el MCD de una colección de elementos puede ser expresado como una combinación lineal de aquellos elementos. La forma extendida del algoritmo Euclidiano es entonces un medio para determinar los coeficientes de esta combinación lineal [11], de tal manera que las condiciones iniciales del proceso están dadas por $r_0(X) = X^{2t+1}$ y $r_1(X) = 1 + S(X)$ con $S(X)$ definida en la ecuación 2.3. El algoritmo termina cuando el grado de $[r_j(X)] \leq t$. La relación recursiva asegura que en algún punto dado del algoritmo, en el *j-ésimo* paso, la combinación lineal está dada por la ecuación 2.10.

$$r_j(x) = a_j(X)X^{2t+1} + b_j(X)S(X) \tag{2.10}$$

Donde el grado de $[r_j(X)] \leq t$

Entonces, $\Lambda(X) = r_j(X)$ y $\sigma(X) = b_j(X)$ (se debe notar que desde el punto de vista de la decodificación no es de interés el polinomio $a_j(X)$), cumpliéndose las ecuaciones 2.11, 2.12, 2.13 [12]:

[14] Sugiyama: En 1975 Sugiyama et al. descubrieron que se puede utilizar el algoritmo Euclidiano para decodificar los códigos de Reed-Salomón.

$$r_{j-2}(X) = q_j(X) r_{j-1}(X) + r_j(X) \qquad 0 \le grado[r_j(X)] < grado[r_{j-1}(X)] \qquad (2.11)$$

$$a_j(X) = a_{j-2}(X) - q_j(X) a_{j-1}(X) \qquad 2.12)$$

$$b_j(X) = b_{j-2}(X) - q_j(X) b_{j-1}(X) \qquad (2.13)$$

Para continuar con el ejemplo ilustrativo $RS(15,9)$ se establecen las siguientes condiciones iniciales:

$$S(X) = \alpha^{10}X^6 + X^4 + \alpha^5 X^3 + X^2 + X$$
$$r_0(X) = X^{2t+1} \qquad b_0(X) = 0$$
$$r_1(X) = 1 + S(X) \qquad b_1(X) = 1$$

Donde $S(X)$ es encontrado a partir de 2.4.

para $j = 2$

$$\begin{array}{r} \alpha^5 X \\ \alpha^{10}X^6 + X^4 + \alpha^5 X^3 + X^2 + X + 1 \overline{\big) X^7} \\ \underline{X^7 + \alpha^5 X^5 + \alpha^{10} X^4 + \alpha^5 X^3 + \alpha^5 X^2 + \alpha^5 X} \\ \alpha^5 X^5 + \alpha^{10} X^4 + \alpha^5 X^3 + \alpha^5 X^2 + \alpha^5 X \end{array}$$

$$b_j(X) = b_{j-2}(X) - q_j(X) b_{j-1}(X)$$
$$b_2(X) = 0 + \alpha^5 X\,(1)$$
$$b_2(X) = \alpha^5 X$$

para $j = 3$

$$\begin{array}{r} \alpha^5 X + \alpha^{10} \\ \alpha^5 X^5 + \alpha^{10} X^4 + \alpha^5 X^3 + \alpha^5 X^2 + \alpha^5 X \overline{\big)\ \alpha^{10} X^6 + X^4 + \alpha^5 X^3 + X^2 + X + 1} \\ \underline{\alpha^{10} X^6 + X^5 + \alpha^{10} X^4 + \alpha^{10} X^3 + \alpha^{10} X^2} \\ X^5 + \alpha^5 X^4 + X^3 + \alpha^5 X^2 + X + 1 \\ \underline{X^5 + \alpha^5 X^4 + X^3 + X^2 + 1} \\ \alpha^{10} X^2 + 1 \end{array}$$

$$b_3(X) = b_1(X) + q_3(X) b_2(X)$$
$$b_3(X) = 1 + (\alpha^5 X + \alpha^{10})(\alpha^5 X)$$
$$b_3(X) = \alpha^{10} X^2 + X + 1$$

En este punto se tiene, que $grado[r_i(X)] \leq t\ con\ t = 3$, por tanto el proceso de divisiones sucesivas termina, de manera que $\sigma(X) = \alpha^{10}X^2 + X + 1$. La ecuación 2.5 indica que el polinomio localizador reciproco es $\sigma_r(X) = X^2 + X + \alpha^{10}$, polinomio utilizado para encontrar las posiciones de error.

2.1.4 Localización de error

Una vez encontrado el $\sigma(X)$ y/o $\sigma_r(X)$ se procede a encontrar las posiciones de error X_i. Para ello hay dos algoritmos equivalentes: uno de factorización explicita, técnica que funciona bien y es rápida para códigos con una baja capacidad de corrección; y dos, el buscador Chien. El Buscador Chien encuentra las raíces de $\sigma_r(X)$, evaluando todos los componentes del campo diferentes de cero, $(1, \alpha, \alpha^2, \ldots \alpha^{p^m-2})$ en $\sigma_r(X)$, de manera que si $\sigma_r(z_i) = 0$, la raíz z_i es tomada como un número localizador de error. El $\sigma_r(X)$ es definido en la ecuación 2.14 [6].

$$\sigma_r(X) = \prod_{i=1}^{v}(X + z_i) \tag{2.14}$$

Donde v es el número de errores.

Una vez obtenidos los z_i el siguiente paso es evaluarlos en la ecuación 2.15 que relaciona la posición del error, los números localizadores de error y el grado del elemento primitivo, que para efectos de simplificación en los cálculos ha sido tomado desde el principio como uno $(G = 1)$[6].

$$X_i = X^{\left(\frac{\log_\alpha z_i}{G}\right)} = X^{(\log_\alpha z_i)} \tag{2.15}$$

A continuación se presenta las raíces de $\sigma_r(X)$ del ejemplo ilustrativo $RS(15, 9)$.

$$\sigma_r(X) = X^2 + X + \alpha^{10}$$
$$\sigma_r(1) = (1)^2 + (1) + \alpha^{10} = \alpha^{10}$$
$$\sigma_r(\alpha) = (\alpha)^2 + (\alpha) + \alpha^{10} = \alpha^5 + \alpha^{10} = 1$$
$$\sigma_r(\alpha^2) = (\alpha^2)^2 + (\alpha^2) + \alpha^{10} = \alpha^4 + \alpha^2 + \alpha^{10} = \alpha^{10} + \alpha^{10} = 0$$
$$\sigma_r(\alpha^3) = (\alpha^3)^2 + (\alpha^3) + \alpha^{10} = \alpha^6 + \alpha^3 + \alpha^{10} = \alpha^2 + \alpha^{10} = \alpha^4$$
$$\sigma_r(\alpha^4) = (\alpha^4)^2 + (\alpha^4) + \alpha^{10} = \alpha^8 + \alpha^4 + \alpha^{10} = \alpha^5 + \alpha^{10} = 1$$
$$\sigma_r(\alpha^5) = (\alpha^5)^2 + (\alpha^5) + \alpha^{10} = \alpha^{10} + \alpha^5 + \alpha^{10} = \alpha^5$$
$$\sigma_r(\alpha^6) = (\alpha^6)^2 + (\alpha^6) + \alpha^{10} = \alpha^{12} + \alpha^6 + \alpha^{10} = \alpha^4 + \alpha^{10} = \alpha^2$$

$$\sigma_r\left(\alpha^7\right)=(\alpha^7)^2+\left(\alpha^7\right)+\alpha^{10}=\alpha^{14}+\alpha^7+\alpha^{10}=\alpha+\alpha^{10}=\alpha^8$$

$$\sigma_r\left(\alpha^8\right)=(\alpha^8)^2+\left(\alpha^8\right)+\alpha^{10}=\alpha+\alpha^8+\alpha^{10}=\alpha^{10}+\alpha^{10}=0$$

$$\sigma_r\left(\alpha^9\right)=(\alpha^9)^2+\left(\alpha^9\right)+\alpha^{10}=\alpha^{15}.\alpha^3+\alpha^9+\alpha^{10}=\alpha^3+\alpha^9+\alpha^{10}=\alpha+\alpha^{10}=\alpha^8$$

$$\sigma_r\left(\alpha^{10}\right)=(\alpha^{10})^2+\left(\alpha^{10}\right)+\alpha^{10}=\alpha^{15}.\alpha^5+\alpha^{10}+\alpha^{10}=\alpha^5$$

$$\sigma_r\left(\alpha^{11}\right)=(\alpha^{11})^2+\left(\alpha^{11}\right)+\alpha^{10}=\alpha^{15}.\alpha^7+\alpha^{11}+\alpha^{10}=\alpha^7+\alpha^{11}+\alpha^{10}=\alpha^8+\alpha^{10}=\alpha$$

$$\sigma_r\left(\alpha^{12}\right)=(\alpha^{12})^2+\left(\alpha^{12}\right)+\alpha^{10}=\alpha^{15}.\alpha^9+\alpha^{12}+\alpha^{10}=\alpha^9+\alpha^{12}+\alpha^{10}=\alpha^8+\alpha^{10}=\alpha$$

$$\sigma_r\left(\alpha^{13}\right)=(\alpha^{13})^2+\left(\alpha^{13}\right)+\alpha^{10}=\alpha^{15}.\alpha^{11}+\alpha^{13}+\alpha^{10}=\alpha^{11}+\alpha^{13}+\alpha^{10}=\alpha^4+\alpha^{10}=\alpha^2$$

$$\sigma_r\left(\alpha^{14}\right)=(\alpha^{14})^2+\left(\alpha^{14}\right)+\alpha^{10}=\alpha^{15}.\alpha^{13}+\alpha^{14}+\alpha^{10}=\alpha^{13}+\alpha^{14}+\alpha^{10}=\alpha^2+\alpha^{10}=\alpha^4$$

Se debe notar que $\sigma_r(0)=\sigma_r(\alpha^{-\infty})$ nunca se calcula, debido a que no existe un $X^{-\infty}=0$ lo cual indica una posición nula [6]. Por tanto los números localizadores de error son $z_1=\alpha^2$ y $z_2=\alpha^8$, dando como resultado $X_1=X^2$ y $X_2=X^8$ como posiciones de error. Es importante notar que dado que $G=1$, los exponentes de los elementos del campo son iguales a los exponentes de las posiciones de error, pero si $G\neq 1$, el resultado se ve alterado por el factor $1/G$ en el exponente de la ecuación 2.15, debido a que el elemento primitivo es distinto de α.

2.1.5 Cálculo de los valores de error

Una vez conocidas las posiciones de error, se encuentra los respectivos valores de error de dichas posiciones. Este proceso se realiza utilizando el algoritmo de Forney como se observa en la ecuación 2.16, donde $\sigma'(X)$ es definida como la derivada formal para $\sigma(X)$ y $\Lambda(X)$ es el último residuo obtenido del algoritmo Euclidiano al cumplirse que el grado de $[r_j(X)]\leq t$ y [12].

$$E_i=\frac{(z_i)^{2-FCR}\Lambda(z_i^{-1})}{\sigma'(z_i^{-1})}\ . \qquad (2.16)$$

Retomando el ejercicio $RS(15,9)$ se tiene que

$$\Lambda(X)=\alpha^{10}X^2+1$$

$$\sigma(X)=\alpha^{10}X^2+X+1$$

$$\sigma'(X) = 2\alpha^{10}X + 1 = [\alpha^{10} + \alpha^{10}]X + 1$$

$$\sigma'(X) = 1$$

$$con\ FCR = 1$$

$$E_i = \frac{(z_i)^{2-FCR}\Lambda(z_i^{-1})}{\sigma'(z_i^{-1})} = \frac{z_i[\alpha^{10}(z_i^{-1})^2 + 1]}{1} = \alpha^{10}z_i^{-1} + z_i$$

$$E_2 = \alpha^{10}(\alpha^2)^{-1} + \alpha^2 = \alpha^8 + \alpha^2 = 1$$

$$E_2 = \alpha^{10}(\alpha^8)^{-1} + \alpha^8 = \alpha^2 + \alpha^8 = 1$$

2.1.6 Palabra decodificada

Una vez son obtenidos los valores de X_i y E_i se puede construir patrón de error $E(X)$. Dicha estimación del ruido es relacionada con los valores de X_i y E_i como se observa en la ecuación 2.17.

$$E(X) = \sum_{i=1}^{v} E_i X_i \tag{2.17}$$

La palabra recibida es una función de $C(X)$ y $E(X)$ como se observa en 2.1, de manera que es posible calcular $C(X)$, Sin embargo, no siempre es posible obtener la palabra codificada originalmente transmitida, por lo cual es más conveniente referirse a la palabra recuperada como estimación de la palabra transmitida $\hat{C}(X)$ y no directamente como el polinomio $C(X)$ que se originó en el codificador. Esta estimación de la palabra transmitida es calculada con la ecuación 2.18 [13].

$$\hat{C}(X) = R(X) + E(X) \tag{2.18}$$

En la ecuación 2.19 se muestran la suma de forma polinomial.

$$\begin{gathered}\hat{C}(X) = (R_{n-1} + E_{n-1})X^{n-1} + (R_{n-2} + E_{n-2})X^{n-2} + \ldots + (R_1 + E_1)X + (R_0 + E_0) \\ \hat{C}(X) = \hat{C}_{n-1}X^{n-1} + \hat{C}_{n-2}X^{n-2} + \ldots + \hat{C}_1X + \hat{C}_0\end{gathered} \tag{2.19}$$

Una vez obtenida la estimación de la palabra codificada, se observa que su resultado es similar a la palabra codificada mostrada en 1.20, la cual es la palabra originalmente transmitida, como se muestra en la ecuación 2.20.

$$\hat{C}(X) = C(X) \tag{2.20}$$

Donde se concluye que la palabra mensaje y la redundancia recuperada son equivalentes a la palabra mensaje y redundancia transmitidas, como se observa en las ecuaciones 2.21 y 2.22 respectivamente.

$$\hat{M}(X) = M(X) \tag{2.21}$$

$$C\hat{K}(X) = CK(X) \tag{2.22}$$

Donde $\hat{M}(X)$ es presentada en su forma polinomial en la ecuación 2.23.

$$\hat{M}(X) = \hat{C}_{n-1}X^{k-1} + \hat{C}_{n-2}X^{k-2} + \ldots + \hat{C}_{n-k+1}X + \hat{C}_{n-k} \tag{2.23}$$

Una vez finalizado los procesos matemáticos de decodificación, se procede a concluir el ejemplo $R(15,9)$, tratado durante esta primer parte del capítulo 2.

$$E(X) = \sum_{i=1}^{2} E_i X_i$$

$$E(X) = X^2 + X^8$$

$$\hat{C}(X) = R(X) + E(X)$$

$$\hat{C}(X) = X^8 + \alpha^{11}X^7 + \alpha^8 X^5 + \alpha^{10}X^4 + \alpha^4 X^3 + \alpha^3 X^2 + \alpha^8 X + \alpha^{12} + [X^2 + X^8]$$

$$\hat{C}(X) = \alpha^{11}X^7 + \alpha^8 X^5 + \alpha^{10}X^4 + \alpha^4 X^3 + \alpha^{14} X^2 + \alpha^8 X + \alpha^{12}$$

Donde efectivamente el polinomio $\hat{C}(X)$ es el valor de la palabra codificada transmitida.

2.2 DECODIFICACIÓN CON CAPACIDAD DE BORRADO

Cuando en un sistema de comunicación digital, bajo un esquema de modulación M-aria, algunos símbolos recibidos por el demodulador son no fiables, los símbolos de código involucrados en el símbolo de modulación son tratados como símbolos de borrado.

Una borradura o símbolo de borrado es definido como un símbolo de error cuya localización de borradura Y_f es conocida con un alto grado de certeza por parte del demodulador. De esta manera el proceso de decodificación con capacidad de borrado, de forma general consiste en encontrar los valores de las borraduras, las posiciones de error y sus respectivos valores. Se debe tener en cuenta que no se trata de eliminar símbolos, se trata de insertar valores de $GF(q)$ en todas las posiciones que han sido indicadas como borraduras [14], formando una nueva palabra recibida para realizar cálculos que permitan

establecer los valores de los símbolos de la palabra originalmente transmitida. Para el caso de estudio el valor cero es tomado para ser evaluado en las posiciones marcadas como borraduras. En la sección 2.2.1 se presenta el criterio que utiliza el demodulador para marcar las posiciones poco fiables como borraduras.

2.2.1 Tasa de error de modulación

La Tasa de Error de Modulación (MER, *Modulation Error Ratio*) es una medida de la relación señal a ruido (SNR) en una señal modulada digitalmente que proporciona una información más precisa de la capacidad del receptor para demodular.

En una señal de N símbolos recibidos, son capturadas las coordenadas (I_j, Q_j) que representan las posiciones de los símbolos en un diagrama de constelación, donde $(\bar{I}_j, \bar{Q}_j)$ son las coordenadas del vector que representa la posición ideal de un símbolo y $(\delta I_j, \delta Q_j)$ las coordenadas del vector error, el cual es definido como la distancia desde la posición ideal a la posición real del símbolo recibido. De esta manera el vector recibido (I_j, Q_j) es la suma entre el vector ideal $(\bar{I}_j, \bar{Q}_j)$ y el vector de error $(\delta I_j, \delta Q_j)$ [15].

La suma de los cuadrados de las magnitudes de los vectores ideales del símbolo es dividida entre la suma de los cuadrados de las magnitudes de los vectores error. El resultado es expresado como la relación de potencia en unidades logarítmicas en la ecuación 2.24.

$$MER_{[dB]} = 10 * \log_{10}\left(\frac{\sum_{j=1}^{n}(\bar{I}_j^{\,2} + \bar{Q}_j^2)}{\sum_{j=1}^{n}(\delta I_j^2 + \delta Q_j^2)}\right) = 10 * \log_{10}\left[\frac{\sum_{j=1}^{n}(\bar{I}_j^{\,2} + \bar{Q}_j^2)}{\sum_{j=1}^{n}[(I_j - \bar{I}_j)^2 + (Q_j - \bar{Q}_j)^2]}\right] \quad (2.24)$$

Donde I_j, es el componente I del j-ésimo símbolo recibido.

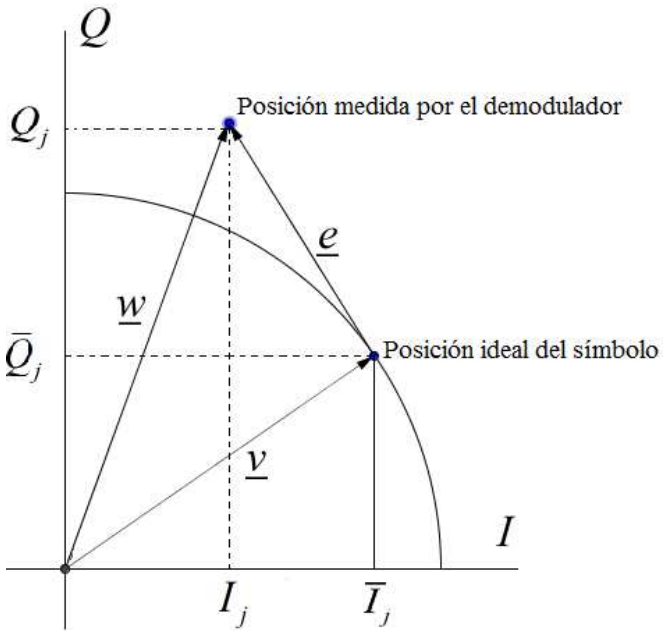
Q_j, es el componente Q del j-ésimo símbolo recibido.

$\bar{I}_j$, es el componente $\bar{I}$ ideal del j-ésimo símbolo recibido.

$\bar{Q}_j$, es el componente $\bar{Q}$ ideal del j-ésimo símbolo recibido.

Como se observa en la figura 2.2, la ubicación real del símbolo es dada por $\underline{w}$, mientras que la posición ideal del símbolo está dado por $\underline{v}$. Por lo tanto, el vector de error resultante medido entre la posición real e ideal del símbolo es $\underline{e} = \underline{w} - \underline{v}$ [16].

Figura 2.2 Representación gráfica de los vectores ideal, real y de error.



Donde $\underline{v}$ es el vector ideal del símbolo.

$\underline{w}$ es la medida del vector símbolo.

$\underline{w} - \underline{v}$ es la magnitud de error.

θ es el error de fase .

La importancia de la capacidad de borrado se debe al incremento sobre la capacidad de corrección de errores hasta el límite superior $n - k = 2t$ símbolos de error, si estos se toman como borraduras. Sin embargo, tiene algunos inconvenientes como el aumento en la carga computacional y la imprecisión del modulador para determinar correctamente el estado de borrado.

Un decodificador que corrige borraduras puede corregir todos los v errores y f borraduras si cumple con la desigualdad mostrada en 2.25

$$2v + f \leq 2t \tag{2.25}$$

A manera de ejemplo con $t = 3$ se tiene que $2v + f \leq 6$ por tanto las posibles combinaciones entre los símbolos de borradura y símbolos de error son [6]:

Con $v = 0$ entonces $0 + f \leq 6$ luego las posibilidades de $f = 6,5,4,3,2,1$ ó 0

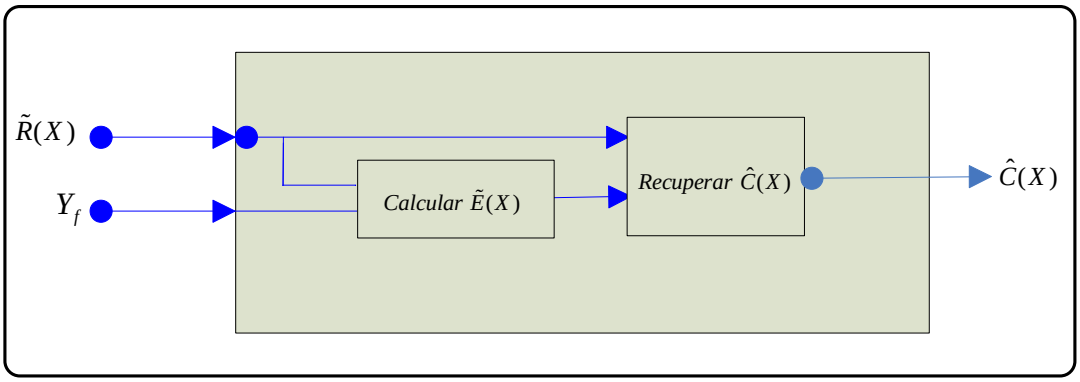
Con $v = 1$ entonces $2 + f \leq 6$ luego las posibilidades de $f = 4,3,2,1$ ó 0

Con $v = 2$ entonces $4 + f \leq 6$ luego las posibilidades de $f = 2,1$ ó 0

Con $v = 3$ entonces $6 + f \leq 6$ luego las posibilidades de $f = 0$

El diagrama en bloques del decodificador $RS(n,k)$ con capacidad de borrado es presentado en la figura 2.3, donde $\tilde{R}(X)$ es la palabra recibida, Y_f las borraduras, $\tilde{E}(X)$ el patrón de error-borradura y $\hat{C}(X)$ es la estimación de la palabra transmitida. El símbolo ~ hace referencia a términos involucrados en el proceso con capacidad de borrado, con el propósito de diferenciar los términos utilizados en el proceso de decodificación sin capacidad de borrado.

Figura 2.3 Diagrama en bloques del decodificador $RS(n,k)$ con capacidad de borrado.



Para entender el proceso de la decodificación con capacidad de borrado se supone una palabra recibida con v errores y f borraduras. Los errores ocurren en las posiciones $i_1, i_2, \ldots, i_v$, mientras que las borraduras ocurren en las posiciones $j_1, j_2, \ldots, j_f$. Las posiciones para los errores y las borraduras son designadas utilizando las posiciones de error $X_1 = \alpha^{i_1}, X_2 = \alpha^{i_2}, \ldots X_v = \alpha^{i_v}$ y las posiciones de borradura como $Y_1 = \alpha^{j_1}, Y_2 = \alpha^{j_2}, \ldots Y_f = \alpha^{j_f}$.

La primera tarea es determinar las posiciones de error, la segunda tarea es encontrar los valores de error $\{E_i\}$ asociados a las ubicaciones de error y los valores asociados con las posiciones de las borraduras previamente dadas por el demodulador.

2.2.2 Polinomio síndrome modificado

El proceso de decodificación con capacidad de borrado, es similar al proceso para la decodificación sin capacidad de borrado con algunas alteraciones. Se define un polinomio evaluador de error-borradura error $W(X)$ como se muestra en la ecuación 2.26 [11].

$$W(X) = [1 + \Xi(X)]\sigma(X) \ mod \ X^{2t+1} \tag{2.26}$$

Donde el polinomio síndrome modificado $\Xi(X)$ es definido como se muestra en la ecuación 2.27.

$$1+\Xi(X)=\tau(X)[1+S(X)]\ mod\ X^{2t+1} \tag{2.27}$$

El término $S(X)$ se obtiene por medio de la ecuación 2.4 y es constituido por los componentes del polinomio síndrome que se obtienen al evaluar las $2t$ raíces del polinomio generador $g(X)$, en la palabra recibida $\tilde{R}(X)$, cuyas f posiciones de borradura han sido evaluadas en cero. Dicha evaluación da como resultado la nueva palabra recibida $\tilde{R}_{f=0}(X)$, con la que se continúa el proceso de decodificación.

2.2.3 Polinomio localizador de borraduras

El polinomio está constituido por las posiciones de borradura como se observa en la ecuación 2.28, donde las borraduras son de la forma $Y_f = \alpha^{j_f}$.

$$\tau(X)=\prod_{l=1}^{f}(1-Y_l X) \tag{2.28}$$

2.2.4 Algoritmo Euclidiano

Nuevamente como en el caso de decodificación sin capacidad de borrado, el problema de decodificación se enfoca en encontrar el polinomio $W(X)$ que satisface la ecuación 2.26. Las condiciones iniciales para operar el algoritmo Euclidiano son $b_o(X)=0, b_1(X)=1$, $a_o(X)=0$, $a_1(X)=1$, $r_0(X)=X^{2t+1}$ y $r_1(X)=1+\Xi(X)$.

La condición de parada de las iteraciones del algoritmo Euclidiano depende del número total de las borraduras y de la capacidad de corrección t. Esta condición se presenta en la desigualdad 2.29 [11].

$$\text{grado}[\mathrm{r}_j(\mathrm{X})] \le \begin{cases} \mathrm{t}+\dfrac{f}{2}, & si \quad f \text{ es par} \\ \mathrm{t}+\dfrac{f-1}{2}, & si \quad f \text{ es impar} \end{cases} \tag{2.29}$$

En el momento en que el algoritmo Euclidiano termina sus iteraciones se tiene que $W(X)=r_j(X)$ y $b_j(X)=\sigma_i(X)$.

En este punto el procedimiento para encontrar las posiciones de error es similar al proceso sin capacidad de borrado, donde se debe encontrar el reciproco del polinomio localizador de error $\sigma_r(X)$ por medio de la ecuación 2.4. Los números localizadores z_i son obtenidos con la ayuda del buscador Chien, para finalmente encontrar las posiciones de error X_i.

2.2.5 Cálculo de los valores de error

Una vez se tiene el polinomio localizador de error $\sigma(X)$ se combina con el polinomio localizador de borradura para obtener el polinomio localizador de error-borradura $\Psi(X)$. Este polinomio se calcula con la ecuación 2.30.

$$\Psi(X) = \sigma(X)\tau(X) \tag{2.30}$$

Se utiliza una versión modificada del algoritmo Forney para encontrar los valores de error y borradura como se observa en las ecuaciones 2.31 y 2.32 respectivamente.

$$E_i = \frac{(z_i)^{2\text{-}FCR} W(z_i^{-1})}{\Psi'(z_i^{-1})} \tag{2.31}$$

$$F_i = \frac{(Y_i)^{2\text{-}FCR} W(Y_i^{-1})}{\Psi'(Y_i^{-1})} \tag{2.32}$$

El patrón de error-borradura $\tilde{E}(X)$, es calculado con la ecuación 2.33, el cual es la suma de todas las borraduras y errores en sus respectivas posiciones. Obtenido el patrón de error-borradura se suma a la palabra recibida con las f borraduras evaluadas en cero $\tilde{R}_{f=0}(X)$, obteniendo la estimación de la palabra codificada $\hat{C}(X)$, mostrada en la ecuación 2.34.

$$\tilde{E}(X) = \sum_{i=1}^{v} E_i X_i + \sum_{i=1}^{f} F_i Y_i \tag{2.33}$$

$$\hat{C}(X) = \tilde{R}_{f=0}(X) + \tilde{E}(X) \tag{2.34}$$

2.2.6 Ejemplo ilustrativo decodificación con capacidad de borrado

Para explicar cómo el código RS puede aumentar su capacidad de corrección de errores usando borraduras, se asume que el demodulador indica las posiciones X^2 y X^7 como símbolos no fiables y por tanto estas son consideradas por el decodificador como

borraduras. Teniendo en cuenta esta condición y asumiendo dos errores en las posiciones X^8 y X se retoma el ejemplo $RS\ (15, 9)$.

Considerando el mismo mensaje transmitido se tiene

$$M(X) = 0X^8 + 0X^7 + 0X^6 + 0X^5 + 0X^4 + 0X^3 + 0X^2 + \alpha^{11}X + 0 = \alpha^{11}X$$
$$C(X) = \alpha^{11}X^7 + \alpha^8 X^5 + \alpha^{10}X^4 + \alpha^4 X^3 + \alpha^{14}X^2 + \alpha^8 X + \alpha^{12}$$

Y con la palabra recibida $\tilde{R}(X)$ dada por

$$\tilde{R}(X) = \underline{X^8} + \underline{\alpha^7 X^7} + \alpha^8 X^5 + \alpha^{10}X^4 + \alpha^4 X^3 + \underline{\alpha^3 X^2} + \underline{\alpha^9 X} + \alpha^{12}$$

ERROR ↵ ↓ BORRADURA ↵ ↓

BORRADURA ERROR

Se procede con la decodificación.

1. Se evalúan los valores de borradura $Y_1 = \alpha$ y $Y_2 = \alpha^8$ en la ecuación (2.28) y se obtiene el polinomio localizador de borradura.

$$\tau(X) = \alpha^9 X^2 + \alpha^{12} X + 1$$

2. Se calcula la nueva palabra recibida y se procede a evaluar las $2t = 6$ raíces del polinomio generador $g(X)$.

$$\tilde{R}(X) = X^8 + fX^7 + \alpha^8 X^5 + \alpha^{10}X^4 + \alpha^4 X^3 + fX^2 + \alpha^9 X + \alpha^{12}$$
$$\tilde{R}_{f=0}(X) = X^8 + \alpha^8 X^5 + \alpha^{10}X^4 + \alpha^4 X^3 + \alpha^9 X + \alpha^{12}$$

para $X = \alpha$

$$S_1 = \tilde{R}_{f=0}(\alpha) = (\alpha)^8 + \alpha^8(\alpha)^5 + \alpha^{10}(\alpha)^4 + \alpha^4(\alpha)^3 + \alpha^9(\alpha) + \alpha^{12}$$
$$S_1 = (\alpha^8 + \alpha^{13}) + (\alpha^{14} + \alpha^7) + (\alpha^{10}\alpha^{12})$$
$$S_1 = \alpha^3 + \alpha + \alpha^3$$
$$S_1 = \alpha$$

para $X = \alpha^2$

$$S_2 = \tilde{R}_{f=0}(\alpha^2) = (\alpha^2)^8 + \alpha^8(\alpha^2)^5 + \alpha^{10}(\alpha^2)^4 + \alpha^4(\alpha^2)^3 + \alpha^9(\alpha^2) + \alpha^{12}$$
$$S_2 = \alpha + \alpha^3 + \alpha^3 + \alpha^{10} + \alpha^{11} + \alpha^{12}$$
$$S_2 = \alpha^8 + 1$$
$$S_2 = \alpha^2$$

para $X=\alpha^3$

$$S_3 = \tilde{R}_{f=0}(\alpha^3) = (\alpha^3)^8 + \alpha^8(\alpha^3)^5 + \alpha^{10}(\alpha^3)^4 + \alpha^4(\alpha^3)^3 + \alpha^9(\alpha^3) + \alpha^{12}$$
$$S_3 = \alpha^9 + \alpha^8 + \alpha^7 + \alpha^{13} + \alpha^{12} + \alpha^{12}$$
$$S_3 = \alpha^{14}$$

para $X=\alpha^4$

$$S_4 = \tilde{R}_{f=0}(\alpha^4) = (\alpha^4)^8 + \alpha^8(\alpha^4)^5 + \alpha^{10}(\alpha^4)^4 + \alpha^4(\alpha^4)^3 + \alpha^9(\alpha^4) + \alpha^{12}$$
$$S_4 = \alpha^2 + \alpha^{13} + \alpha^{11} + \alpha + \alpha^{13} + \alpha^{12}$$
$$S_4 = \alpha^{10}$$

para $X=\alpha^5$

$$S_5 = \tilde{R}_{f=0}(\alpha^5) = (\alpha^5)^8 + \alpha^8(\alpha^5)^5 + \alpha^{10}(\alpha^5)^4 + \alpha^4(\alpha^5)^3 + \alpha^9(\alpha^5) + \alpha^{12}$$
$$S_5 = \alpha^{10} + \alpha^3 + 1 + \alpha^4 + \alpha^{14} + \alpha^{12}$$
$$S_5 = \alpha^{12} + \alpha^{14} + \alpha^{12} + 1 + \alpha^4$$
$$S_5 = \alpha^{14} + 1 + \alpha^4$$
$$S_5 = \alpha^{14} + \alpha$$
$$S_5 = \alpha^7$$

para $X=\alpha^6$

$$S_6 = \tilde{R}_{f=0}(\alpha^6) = (\alpha^6)^8 + \alpha^8(\alpha^6)^5 + \alpha^{10}(\alpha^6)^4 + \alpha^4(\alpha^6)^3 + \alpha^9(\alpha^6) + \alpha^{12}$$
$$S_6 = \alpha^3 + \alpha^8 + \alpha^4 + \alpha^7 + 1 + \alpha^{12}$$
$$S_6 = \alpha^{13} + \alpha^3 + \alpha^{11}$$
$$S_6 = \alpha^7$$

Por tanto los componentes del polinomio síndrome son:

$$S_1 = \alpha \qquad S_2 = \alpha^2 \qquad S_3 = \alpha^{14}$$
$$S_4 = \alpha^{10} \qquad S_5 = \alpha^7 \qquad S_6 = \alpha^7$$

3. Se evalúan los componentes del polinomio síndrome en la ecuación 2.3 obteniendo:

$$S(X) = \alpha X + \alpha^2 X^2 + \alpha^{14} X^3 + \alpha^{10} X^4 + \alpha^7 X^5 + \alpha^7 X^6$$

4. Se calcula el polinomio síndrome modificado $\Xi(X)$ con la ecuación 2.27. Entonces se calcula $\tau(X)[1+S(X)]$.

$$1+\alpha X+\alpha^2X^2+\alpha^{14}X^3+\alpha^{10}X^4+\alpha^7X^5+\alpha^7X^6$$
$$\underline{1+\alpha^{12}X+\alpha^9X^2}$$
$$1+\alpha X+\alpha^2X^2+\alpha^{14}X^3+\alpha^{10}X^4+\alpha^7X^5+\alpha^7X^6$$
$$\alpha^{12}X+\alpha^{13}X^2+\alpha^{14}X^3+\alpha^{11}X^4+\alpha^7X^5+\alpha^4X^6+\alpha^4X^7$$
$$\underline{\alpha^9X^2+\alpha^{10}X^3+\alpha^{11}X^4+\alpha^8X^5+\alpha^4X^6+\alpha X^7+\alpha X^8}$$
$$1+\alpha^{13}X+\alpha^4X^2+\alpha^{10}X^3+\alpha^{10}X^4+\alpha^8X^5+\alpha^7X^6+X^7+\alpha X^8$$

Ahora $\tau(X)[1+S(X)] \bmod X^{2t+1}$

$$\begin{array}{r|l} & \alpha X+1 \\ \hline X^7 & \alpha X^8+X^7+\alpha^7X^6+\alpha^8X^5+\alpha^{10}X^4+\alpha^{10}X^3+\alpha^4X^2+\alpha^{13}X+1 \\ & \underline{\alpha X^8} \\ & X^7+\alpha^7X^6+\alpha^8X^5+\alpha^{10}X^4+\alpha^{10}X^3+\alpha^4X^2+\alpha^{13}X+1 \\ & \underline{X^7} \\ & \alpha^7X^6+\alpha^8X^5+\alpha^{10}X^4+\alpha^{10}X^3+\alpha^4X^2+\alpha^{13}X+1 \end{array}$$

Por tanto $\Xi(X) = \alpha^7X^6 + \alpha^8X^5 + \alpha^{10}X^4 + \alpha^{10}X^3 + \alpha^4X^2 + \alpha^{13}X$

5. Se realiza el procedimiento del Algoritmo Euclidiano bajo las siguientes condiciones iníciales.

$b_0(X) = 0$
$b_1(X) = 1$
$r_1 = \Xi(X) + 1$
$r_0 = X^{2t+1}$

condicion de parada

$$grado[r_j(X)] \le \left\{ \begin{array}{ll} t+\frac{f}{2}, & si \quad f \text{ es par} \\ t+\frac{f-1}{2}, & si \quad f \text{ es impar} \end{array} \right\}$$

Como $f = 2$ *entonces el* $grado[r_i(X)] \le 4$

para $j=2$

$$\begin{array}{r|l} & \alpha^8 X+\alpha^9 \\ \hline \alpha^7 X^6+\alpha^8 X^5+\alpha^{10} X^4+\alpha^{10} X^3+\alpha^4 X^2+\alpha^{13} X+1 & X^7 \\ & X^7+\alpha X^6+\alpha^3 X^5+\alpha^3 X^4+\alpha^{12} X^3+\alpha^6 X^2+\alpha^8 X \\ \hline & \alpha X^6+\alpha^3 X^5+\alpha^3 X^4+\alpha^{12} X^3+\alpha^6 X^2+\alpha^8 X \\ & \alpha X^6+\alpha^2 X^5+\alpha^4 X^4+\alpha^4 X^3+\alpha^{13} X^2+\alpha^7 X+\alpha^9 \\ \hline & \alpha^6 X^5+\alpha^7 X^4+\alpha^6 X^3+X^2+\alpha^{11} X+\alpha^9 \end{array}$$

$b_2(X)=\alpha^8 X+\alpha^9$

para $j=3$

$$\begin{array}{r|l} & \alpha X \\ \hline \alpha^6 X^5+\alpha^7 X^4+\alpha^6 X^3+X^2+\alpha^{11} X+\alpha^9 & \alpha^7 X^6+\alpha^8 X^5+\alpha^{10} X^4+\alpha^{10} X^3+\alpha^4 X^2+\alpha^{13} X+1 \\ & \alpha^7 X^6+\alpha^8 X^5+\alpha^7 X^4+\alpha X^3+\alpha^{12} X^2+\alpha^{10} X \\ \hline & \alpha^6 X^4+\alpha^8 X^3+\alpha^6 X^2+\alpha^9 X+1 \end{array}$$

$b_3(X)=\alpha^9 X^2+\alpha^{10} X+1$

En este punto el $grado\,[r_j(X)] \leq 4$, de manera que hasta aquí se realizan las iteraciones del algoritmo Euclidiano y el polinomio localizador de error con capacidad de borrado es $\sigma(X)=\alpha^9 X^2+\alpha^{10} X+1$. Por tanto $\sigma_r(X)=X^2+\alpha^{10} X+\alpha^9$ y $W(X)=\alpha^6 X^4+\alpha^8 X^3+\alpha^6 X^2+\alpha^9 X+1$.

6. Se evalúan en $\sigma_r(X)$ los componentes del campo diferentes de cero $\left(1,\alpha,\alpha^2,\ldots\alpha^{p^m-2}\right)$, encontrando que los valores para los cuales la expresión se hace cero son α y α^8. Dado que el elemento primitivo es α, las raíces $z_1=\alpha$ y $z_2=\alpha^8$ son las posiciones de los errores en X y X^8 respectivamente.

$$\sigma_r(\alpha)=(\alpha)^2+\alpha^{10}.\alpha+\alpha^9=\alpha^9+\alpha^9=0$$
$$\sigma_r(\alpha^2)=(\alpha^2)^2+\alpha^{10}(\alpha^2)+\alpha^9=\alpha^4+\alpha^{12}+\alpha^9=\alpha^5$$
$$\sigma_r(\alpha^3)=(\alpha^3)^2+\alpha^{10}(\alpha^3)+\alpha^9=\alpha^6+\alpha^{13}+\alpha^9=\alpha^7$$
$$\sigma_r(\alpha^4)=(\alpha^4)^2+\alpha^{10}(\alpha^4)+\alpha^9=\alpha^8+\alpha^{14}+\alpha^9=\alpha^5$$
$$\sigma_r(\alpha^5)=(\alpha^5)^2+\alpha^{10}(\alpha^5)+\alpha^9=\alpha^{10}+1+\alpha^9=\alpha^6$$
$$\sigma_r(\alpha^6)=(\alpha^6)^2+\alpha^{10}(\alpha^6)+\alpha^9=\alpha^{12}+\alpha=\alpha^{13}$$
$$\sigma_r(\alpha^7)=(\alpha^7)^2+\alpha^{10}(\alpha^7)+\alpha^9=\alpha^{14}+\alpha^2+\alpha^9=\alpha^{10}$$
$$\sigma_r(\alpha^8)=(\alpha^8)^2+\alpha^{10}(\alpha^8)+\alpha^9=\alpha+\alpha^3+\alpha^9=0$$

$$\sigma_r\left(\alpha^9\right)=(\alpha^9)^2+\alpha^{10}(\alpha^9)+\alpha^9=\alpha^3+\alpha^4+\alpha^9=1$$
$$\sigma_r\left(\alpha^{10}\right)=(\alpha^{10})^2+\alpha^{10}(\alpha^{10})+\alpha^9=\alpha^5+\alpha^5+\alpha^9=\alpha^9$$
$$\sigma_r\left(\alpha^{11}\right)=(\alpha^{11})^2+\alpha^{10}(\alpha^{11})+\alpha^9=\alpha^7+\alpha^6+\alpha^9=\alpha^{13}$$
$$\sigma_r\left(\alpha^{12}\right)=(\alpha^{12})^2+\alpha^{10}(\alpha^{12})+\alpha^9=\alpha^9+\alpha^7+\alpha^9=\alpha^7$$
$$\sigma_r\left(\alpha^{13}\right)=(\alpha^{13})^2+\alpha^{10}(\alpha^{13})+\alpha^9=\alpha^{11}+\alpha^8+\alpha^9=1$$
$$\sigma_r\left(\alpha^{14}\right)=(\alpha^{14})^2+\alpha^{10}(\alpha^{14})+\alpha^9=\alpha^{13}+\alpha^9+\alpha^9=\alpha^{13}$$

7. Se evalúan z_1 y z_2 en la ecuación 2.15 y se obtienen las posiciones donde se encuentran los errores, las cuales son $X_1=X$, $X_2=X^8$. Una vez obtenidas las posiciones de error y conociendo las posiciones de borradura $Y_1=\alpha^2$, $Y_2=\alpha^7$, se calculan los valores de error (E_i) y borradura (F_i), utilizando el polinomio error-borradura $\Psi(X)$ y el último residuo obtenido con el algoritmo Euclidiano.

Polinomio error-borradura

$$\Psi(X)=\sigma(X)\tau(X)$$
$$\alpha^9X^2+\alpha^{12}X+1$$
$$\alpha^9X^2+\alpha^{10}X+1$$
$$\alpha^3X^4+\alpha^4X^3+\alpha^9X^2$$
$$\alpha^6X^3+\alpha^7X^2+\alpha^{12}X$$
$$\alpha^9X^2+\alpha^{10}X+1$$
$$\alpha^3X^4+\alpha^{12}X^3+\alpha^7X^2+\alpha^3X+1$$

$$\Psi'(X)=4\alpha^3X^3+3\alpha^{12}X^2+2\alpha^7X+\alpha^3$$
$$\Psi'(X)=\alpha^{12}X^2+\alpha^3$$

Con $W(X)=\alpha^6X^4+\alpha^8X^3+\alpha^6X^2+\alpha^9X+1$ como último residuo del algoritmo Euclidiano. Estos resultados son evaluados en las ecuaciones 2.31 y 2.32 para así encontrar los valores de los errores y las borraduras respectivamente.

$$E_i=\frac{(z_i)W(z_i^{-1})}{\Psi'(z_i^{-1})}$$
$$E_i=\frac{z_i[\alpha^6(z_i^{-1})^4+\alpha^8(z_i^{-1})^3+\alpha^6(z_i^{-1})^2+\alpha^9(z_i^{-1})+1]}{\alpha^{12}(z_i^{-1})^2+\alpha^3}$$

$$E_i = \frac{\alpha^6 z_i^{-3} + \alpha^8 z_i^{-3} + \alpha^6 z_i^{-1} + \alpha^9 + z_i}{\alpha^{12} z_i^{-2} + \alpha^3}$$

$$con\ z_1 = \alpha\ se\ obtiene\ que$$

$$E_1 = \frac{\alpha^6 \alpha^{-3} + \alpha^8 \alpha^{-3} + \alpha^6 \alpha^{-1} + \alpha^9 + \alpha}{\alpha^{12} \alpha^{-2} + \alpha^3}$$

$$E_1 = \alpha^{12}\ y\ para\ z_2 = \alpha^8 se\ obtine\ E_2 = 1$$

Se procede a encontrar los valores de borraduras.

$$F_i = \frac{(Y_i) W(Y_i^{-1})}{\Psi'(Y_i^{-1})}$$

$$F_i = \frac{Y_k[\alpha^6 (Y_i^{-1})^4 + \alpha^8 (Y_i^{-1})^3 + \alpha^6 (Y_i^{-1})^2 + \alpha^9 (Y_i^{-1}) + 1]}{\alpha^{12} (Y_i^{-1})^2 + \alpha^3}$$

$$F_i = \frac{\alpha^6 Y_i^{-3} + \alpha^8 Y_i^{-3} + \alpha^6 Y_i^{-1} + \alpha^9 + Y_i}{\alpha^{12} Y_i^{-2} + \alpha^3}$$

Para las borraduras en las posiciones $Y_1 = \alpha^2$ y $Y_2 = \alpha^7$ se obtienen los siguientes valores

$$F_1 = \alpha^{14}$$

$$F_2 = \alpha^{11}$$

8. Se evalúan los valores de error y de borradura en sus respectivas posiciones en la ecuación 2.33, se obtiene el patrón de error-borradura $\tilde{E}(X)$:

$$\tilde{E}(X) = \alpha^{12} X + \alpha^{14} X^2 + \alpha^{11} X^7 + X^8$$

Por tanto la palabra codificada es obtenida, al adicionar el patrón de error-borradura, a la palabra nueva recibida $\tilde{R}_{f=0}(X)$.

$$\hat{C}(X) = \tilde{R}_{f=0}(X) + \tilde{E}(X)$$

$$\hat{C}(X) = [X^8 + \alpha^8 X^5 + \alpha^{10} X^4 + \alpha^4 X^3 + \alpha^9 X + \alpha^{12}] + [\alpha^{12} X + \alpha^{14} X^2 + \alpha^{11} X^7 + X^8]$$

$$\hat{C}(X) = \alpha^{11} X^7 + \alpha^8 X^5 + \alpha^{10} X^4 + \alpha^4 X^3 + \alpha^{14} X^2 + \alpha^8 X + \alpha^{12}$$

La cual es la palabra inicialmente codificada.

3. MODELO DE SIMULACIÓN DEL DECODIFICADOR REED-SOLOMON QUE UTILIZA EL ALGORITMO EUCLIDIANO CON CAPACIDAD DE BORRADO

Un modelo de simulación es la representación del comportamiento de un sistema para entender mejor el funcionamiento de dicho sistema, donde el modelo es constituido por elementos tales como procesos, relaciones funcionales y variables. La manera como las variables son manejadas determinan los diferentes modelos de simulación. De esta manera los modelos de simulación son clasificados en continuos o discretos según cambien las variables en el tiempo; en determinísticos o estocásticos[15] según sea el comportamiento de las variables; y en estáticos o dinámicos según su comportamiento en el tiempo. Este capítulo está elaborado siguiendo las recomendaciones del documento Metodología para la Simulación de Equipos de Telecomunicaciones [1], y el proceso para la simulación del Decodificador RS con capacidad utilizando el algoritmo Euclidiano, el cual presenta eventos discretos, dinámicos y estocásticos.

Teniendo en cuenta que la ingeniería del software define tres fases en el planteamiento y solución de un problema, siendo estas: la fase de definición, la fase de desarrollo y la fase de cambios orientados a la depuración del modelo. Las etapas para cumplir las tres fases son:

- Formulación del problema y plan de estudio.
- Recolección y procesamiento de datos.
- Formulación de un modelo de simulación.
- Evaluación del modelo y los parámetros estimados.

3.1 FORMULACIÓN DEL PROBLEMA Y PLAN DE ESTUDIO

El propósito de la formulación del problema es plantear objetivos que den origen a la construcción de un modelo de simulación. Estos objetivos deben arrojar elementos necesarios para establecer los comportamientos internos del sistema. Con el fin de obtener una formulación correcta de dichos objetivos se tienen en cuenta los siguientes aspectos:

- Adquisición de información sobre el funcionamiento del sistema.

[15] Una variable o señal estocástica no puede ser determinada completamente en función del tiempo, y debe ser modelada probabilísticamente.

- Identificación de los propósitos de la simulación.
- Formulación de los objetivos.

3.1.1 Adquisición de información sobre el funcionamiento del sistema

3.1.1.1 Tipo de señales que se van a procesar

Un generador de fuente entrega como señal de salida una secuencia de números aleatorios que constituyen el mensaje a transmitir. Dicho bloque es la señal de entrada al codificador RS, el cual entrega un vector que representa el mensaje codificado como señal de salida. Esta señal es tomada por el modulador como señal de entrada y entrega un vector de números complejos que representa la señal modulada.

El canal de transmisión recibe el vector de números complejos y entrega como señal de salida el vector alterado debido al ruido proporcionado por el canal de transmisión.

El vector de números complejos afectado por el ruido Gaussiano aditivo blanco constituye la señal de entrada al demodulador, el cual entrega como señal de salida un vector que representa el mensaje recuperado. Esta palabra recibida es la señal de entrada al decodificador y este a su vez entrega como señal de salida una secuencia de números que representa la palabra codificada inicialmente transmitida.

3.1.1.2 Pasos a seguir en el procesamiento de las señales

Inicialmente una palabra mensaje $M(X)$ originada en la fuente de información, ingresa al codificador RS. La señal resultante es la palabra mensaje desplazada en un factor de $X^{(n-k)}$ a la cual se le añade un bloque de bits de redundancia obtenidos por la operación *mod* entre el mensaje desplazado y el polinomio generador $g(X)$.

Una vez obtenida esta señal codificada se realiza el proceso de modulación digital, para lo cual se consideran los esquemas de modulación BPSK, 4-QAM, 8-PSK, 16-QAM.

La señal modulada se ve afectada por el ruido durante la transmisión sobre el canal AWGN y es entregada al receptor donde es demodulada teniendo en cuenta el esquema de modulación establecidos en la transmisión. Esta señal pasa al decodificador donde se detectan y corrigen errores, obteniendo la palabra codificada transmitida. Este proceso puede variar si la capacidad de borrado se encuentra activa o no. Los pasos para la decodificación sin capacidad de borrado son: hallar el polinomio síndrome, encontrar el polinomio localizador de error, encontrar las posiciones de error con sus respectivos valores y finalmente obtener la palabra codificada transmitida. Para la decodificación con capacidad de borrado el demodulador debe marcar las posiciones de borradura al decodificador. El número de pasos a realizar es el mismo, diferenciándose en los

procedimientos internos y matemáticos: hallar el polinomio síndrome, encontrar los polinomios localizador de error y localizador de borradura, encontrar las posiciones de error y borradura, asociado a sus respectivos valores, calcular el patrón de error-borradura.

3.1.1.3 Factores que serán evaluados durante la simulación

CODIFICADOR

- Longitud de palabra codificada……..$n = p^m - 1$
- Longitud el mensaje……..k
- Tasa de codificación……..$r = k/n$
- Tiempo de ejecución del codificador.

CANAL

- Relación de energía de bit a densidad espectral de potencia de ruido …E_b/N_0

DECODIFICADOR

- Longitud de palabra codificada……..$n = p^m - 1$
- Longitud del mensaje……..k
- Tasa de codificación……..$r = k/n$
- Tasa de Error de Bit (BER, *Bit Error Rate*).
- Tasa de error de símbolo (SER, *Symbol Error Rate*).
- Tiempo de respuesta.
- Ganancia de codificación.

3.1.2 Identificación de los propósitos de la simulación

Se busca en un ambiente de simulación, evaluar el desempeño a nivel físico del decodificador RS utilizando el algoritmo Euclidiano con capacidad de borrado, para un sistema de banda base, y un modelo de canal AWGN. Para este propósito se utiliza la Metodología para la Simulación de Equipos de Telecomunicaciones.

3.1.3 Formulación de los objetivos

- Generar el modelo de simulación del decodificador RS que utiliza el algoritmo Euclidiano con capacidad de borrado mediante la adaptación y aplicación de la Metodología para la Simulación de Equipos de Telecomunicaciones [1].

- Evaluar el desempeño a nivel físico[16] del decodificador RS que utiliza el algoritmo Euclidiano con capacidad de borrado y comparar su desempeño con respecto al decodificador RS que utiliza el algoritmo Euclidiano sin capacidad de borrado en un canal AWGN.

3.2 RECOLECCIÓN Y PROCESAMIENTO DE DATOS

Es necesario identificar diferentes elementos relacionados entre sí que dan claridad en la recolección y procesamiento de los datos. Estos elementos son identificados uno a uno como se muestra a continuación.

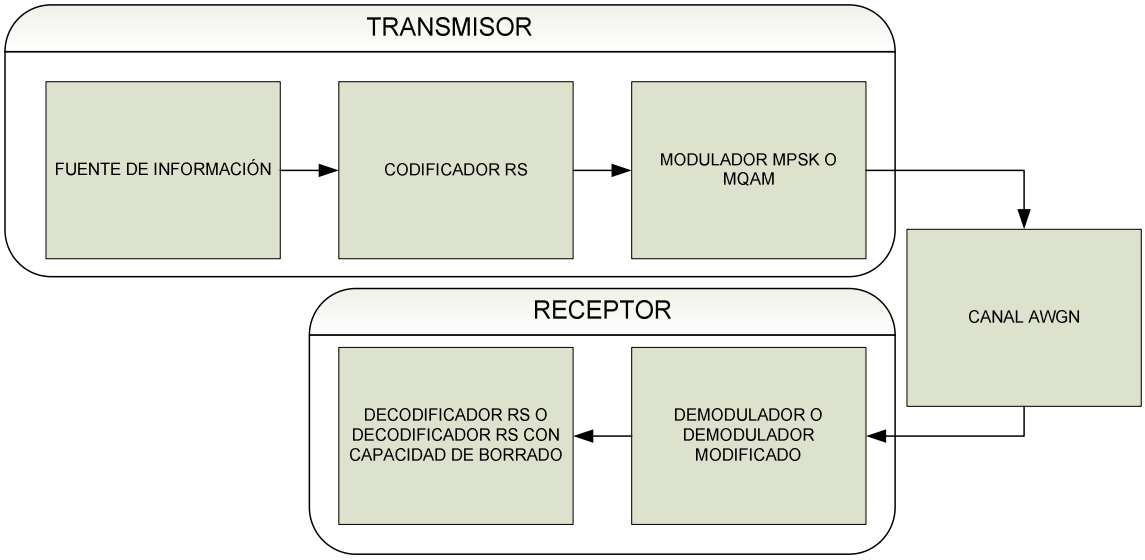
- Identificación de clases y objetos.
- Identificación de estructuras.
- Definición de atributos.
- Definición de servicios.
- Notación en la carta de especificación.

3.2.1 Identificación de clases y objetos

Un sistema de comunicaciones consta básicamente del Transmisor, Canal y Receptor. El transmisor se compone de la fuente de información, Codificador de canal RS, modulador con esquemas de modulación M-PSK y M-QAM. El receptor está compuesto por el demodulador con los esquemas de modulación M-PSK y M-QAM, demodulador modificado, el decodificador RS y decodificador RS con capacidad de borrado. Estos bloques pueden ser considerados como independientes y por tanto utilizados como componentes individuales del sistema.

[16] Parámetros a analizar el desempeño a nivel físico son BER, SER, ganancia de codificación, tiempo de respuesta.

Figura 3.1 Diagrama en bloques sistema de comunicacion.



3.2.2 Identificación de estructuras

Considerando las clases mencionadas anteriormente se presentan las estructuras que permiten manejar la complejidad del sistema.

- Fuente de información

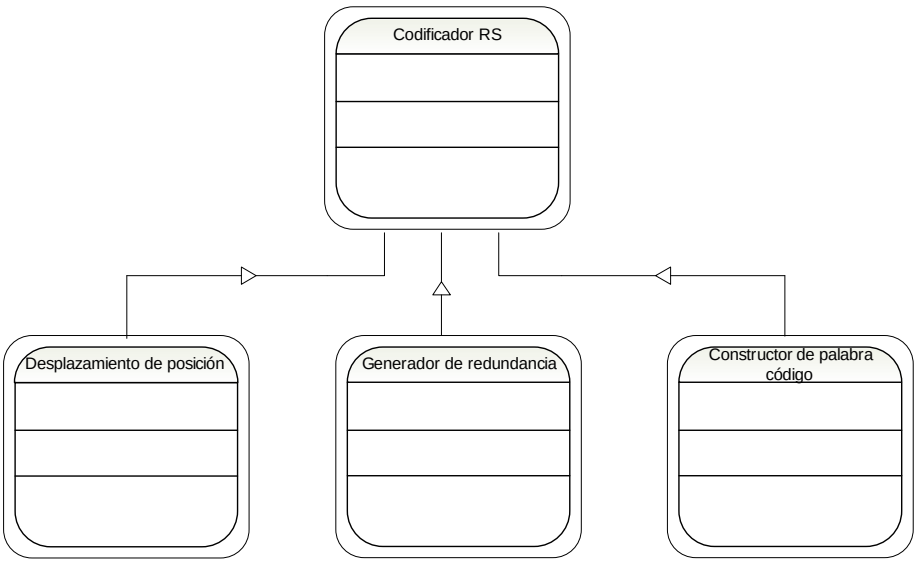
Figura 3.2 Estructura de la fuente de información.

- Codificador de canal RS

  La estructura que mejor define un decodificador de canal es la estructura Todo-Parte, como se observa en la figura 3.3, debido a que esta permite visualizar las partes que conforman el objeto codificador, describiendo una relación con los

componentes: desplazamiento de posición, generador de redundancia y constructor de palabra codificada.

Figura 3.3 Estructura del codificador.



- Modulador digital

  La estructura que modela el comportamiento del demodulador es la estructura General-Específica. Así el modulador digital posee características generales que se heredan a las clases específicas modulador M-PSK y modulador M-QAM. La estructura del demodulador digital se muestra en la figura 3.4

- Canal de transmisión

  El canal de transmisión es descrito por la estructura Todo-Parte, en donde se aprecia que el objeto generador de ruido AWGN hace parte del canal de transmisión como se observa en la figura 3.5.

Figura 3.4 Estructura del modulador digital.

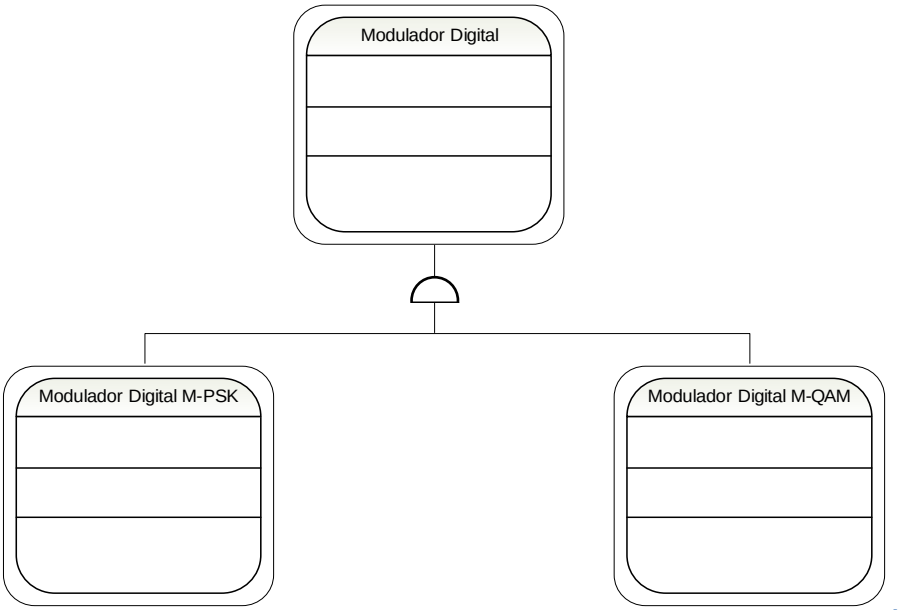
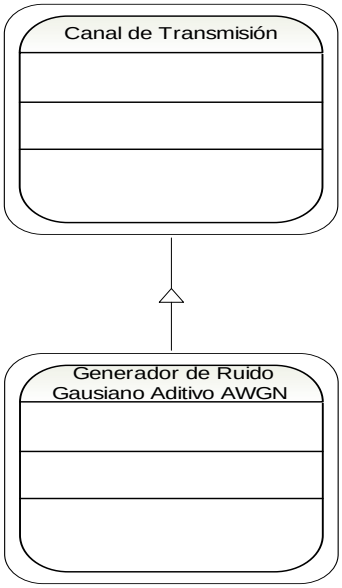


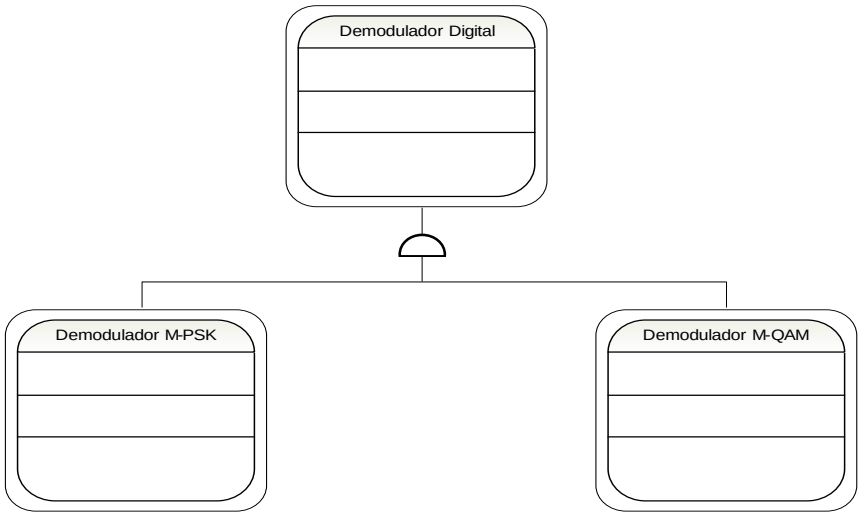
Figura 3.5 Estructura del canal de transmisión.



- Demodulador digital

  Al igual que en el caso del modulador este es descrito por la estructura General-Específico mostrada en la figura 3.6, donde el demodulador digital es la clase general, de la cual se desprenden las clases especificas M-PSK y M-QAM respectivamente.

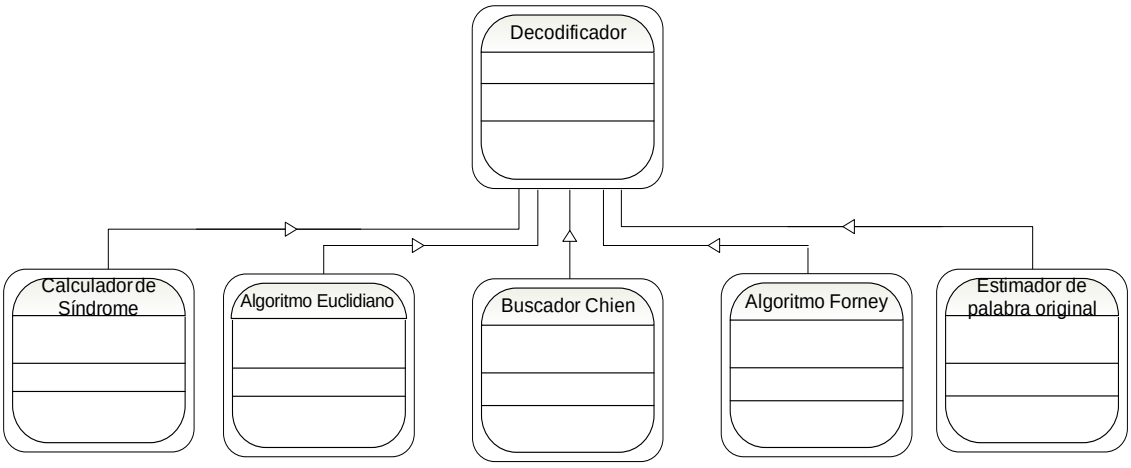
Figura 3.6 Estructura del demodulador digital.



- Decodificador RS con el algoritmo Euclidiano

Para la descripción del decodificador RS con el algoritmo Euclidiano, se presenta en la figura 3.7 la estructura Todo-Parte, en la cual se aprecia los componentes que hacen parte del decodificador, siendo estos: calculador de síndrome, algoritmo Euclidiano, buscador Chien, algoritmo Forney, estimador de palabra original.

Figura 3.7 Estructura del decodificador RS.

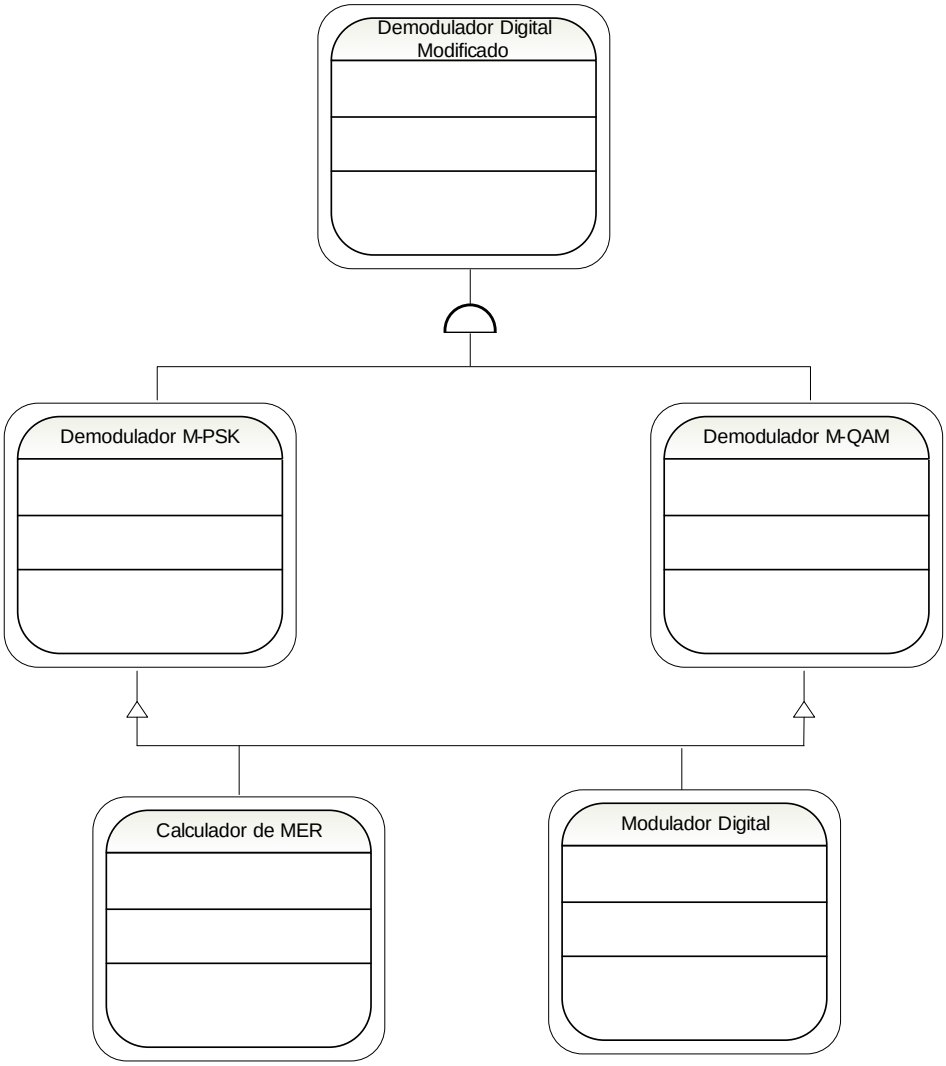


A continuación se presenta el procedimiento para el demodulador modificado y el decodificador RS con capacidad de borrado

- Demodulador digital modificado

El demodulador modificado es descrito por una estructura mixta, donde se observa que conserva la estructura General-Específico del demodulador digital visto anteriormente, al cual se le adiciona el componente MER. La estructura del demodulador digital modificado se muestra en la figura 3.8.

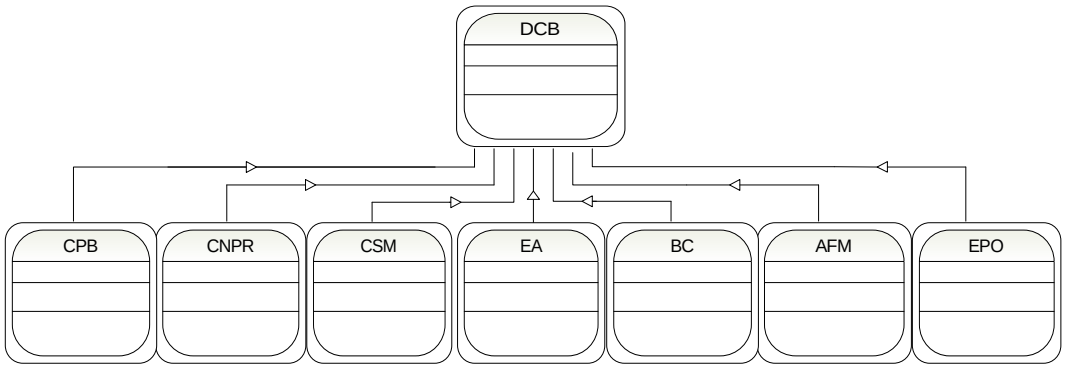
Figura 3.8 Demodulador digital modificado.



- Decodificador con capacidad de borrado

El decodificador con capacidad de borrado, al igual que el decodificador sin capacidad de borrado es descrito por la estructura Todo-Parte, la cual describe el objeto decodificador con capacidad de borrado (DCB) como el todo y las siguientes partes como los componentes que lo conforman: constructor polinomio de borraduras (CPB),constructor nueva palabra recibida (CNPR), calculador de síndrome modificado (CSM), algoritmo Euclidiano (EA), buscador Chien (BC), algoritmo Forney modificado (AFM), estimador de palabra original (EPO). Esta estructura es mostrada en la figura 3.9.

Figura 3.9 Estructura del decodificador RS con capacidad de borrado.



3.2.3 Definición de atributos

Los atributos de una clase o estructura hacen referencia a las características para la cual cada objeto de una clase tiene su propio valor [1]. A continuación son descritos los atributos de las estructuras mencionadas en la sección 3.2.2.

- Fuente de información

  - **Nombre**: conjunto finito de símbolos.
    Descripción: conjunto finito de símbolos generados de forma aleatoria.

  - **Nombre**: longitud de la palabra mensaje.
    Descripción: número de símbolos que constituyen la secuencia del mensaje.

- Codificador RS

  - **Nombre**: polinomio primitivo $p(X)$.
    Descripción: polinomio base con el que se construye el conjunto de símbolos del campo de Galois.

- **Nombre**: longitud de símbolo m.
  Descripción: determina el número de m bits de cada símbolo.

- **Nombre**: longitud de mensaje k.
  Descripción: número de k símbolos del campo de Galois a transmitir.

- **Nombre**: longitud de palabra codificada n.
  Descripción: longitud de la palabra codificada en símbolos, constituido por el bloque mensaje k y el bloque de redundancia CK.

- **Nombre**: primera raíz consecutiva FCR.
  Descripción: determina la potencia de la primera raíz consecutiva del polinomio generador.

- **Nombre**: polinomio generador $g(X)$.
  Descripción: polinomio de grado $n-k$ utilizado para generar el bloque de redundancia que se adiciona a la palabra mensaje.

➢ Modulador digital

- **Nombre**: esquema de modulación.
  Descripción: técnicas utilizadas para modular la señal de entrada. Los esquemas utilizados son PSK y QAM.

- **Nombre**: orden de modulación.
  Descripción: número de símbolos mapeados por el esquema de modulación respectivo.

➢ Canal de transmisión

- **Nombre**: E_b/N_0.
  Descripción: relación de energía de bit a densidad espectral de potencia de ruido.

➢ Demodulador digital y demodulador digital modificado

- **Nombre**: esquema de modulación.
  Descripción: técnica para realizar la demodulación de la señal entrante. Los esquemas de demodulación para el modelo de simulación son M-QAM y M-PSK.

- **Nombre**: orden de modulación.
  Descripción: número de símbolos mapeados en el esquema de modulación respectivo.

➢ Decodificador RS y decodificador con capacidad de borrado

- **Nombre**: polinomio primitivo $p(X)$
  Descripción: polinomio base con el que se construye el conjunto de símbolos del campo de Galois.
- **Nombre**: longitud de mensaje k.
  Descripción: número de símbolos del campo de Galois recibidos.

- **Nombre**: longitud de la palabra codificada n.
  Descripción: longitud de la palabra codificada en símbolos, constituido por el bloque mensaje k y el bloque de redundancia.

- **Nombre**: primera raíz consecutiva FCR.
  Descripción: determina la potencia de la primera raíz consecutiva del polinomio generador.

- **Nombre**: polinomio generador $g(X)$.
  Descripción: polinomio de grado $n-k$ utilizado para generar el bloque de redundancia que se adiciona a la palabra mensaje.

3.2.4 Definición de servicios

A continuación se presentan los servicios de los objetos anteriormente tratados, los cuales describen el comportamiento de cada uno de los objetos.

➢ Fuente de información

- **Nombre:** fijar tamaño de símbolo.
  Descripción: determina el tamaño de cada símbolo en bits utilizando el grado m del polinomio primitivo.

- **Nombre:** fijar longitud de palabra mensaje.
  Descripción: establece el número de símbolos que conforman la palabra mensaje.

- Codificador RS

  - **Nombre**: determinar el polinomio primitivo $p(X)$.
    Descripción: establece el conjunto de símbolos que constituyen el campo de Galois, es decir el lenguaje en el cual se van a entender el codificador con el decodificador.

  - **Nombre**: longitud de símbolo m
    Descripción: determina el tamaño de los símbolos

  - **Nombre**: establecer la longitud del mensaje k.
    Descripción: determina el número de símbolos del campo de Galois a transmitir.

  - **Nombre**: establecer longitud del bloque n.
    Descripción: este servicio asigna la longitud de la palabra codificada en símbolos, constituido por el bloque mensaje k y el bloque de redundancia.

  - **Nombre**: calcular la capacidad de corrección t.
    Descripción: operando los parámetros k y n determina la cantidad máxima de símbolos que pueden ser corregidos.

  - **Nombre**: fijar la primera raíz consecutiva FCR.
    Descripción: establece el valor de FCR para la construcción del polinomio generador.

  - **Nombre**: calcular el polinomio generador $g(X)$.
    Descripción: construcción del polinomio generador, el cual es utilizado para la calcular la redundancia.

  - **Nombre**: correr mensaje.
    Descripción: multiplicación de la palabra mensaje con un polinomio de la forma X^{n-k}.

  - **Nombre**: calcular de la redundancia $CK(X)$.
    Descripción: realiza la función mod entre el mensaje $M(X)$ con corrimiento de grado y el polinomio generador $g(X)$.

  - **Nombre**: generación de la palabra codificada $C(X)$.
    Descripción: realiza la concatenación entre el mensaje con corrimiento y el bloque de redundancia $CK(X)$.

- Modulador digital

  - **Nombre**: seleccionar el esquema de modulación.
    Descripción: selecciona el esquema de modulación a utilizar.

  - **Nombre**: determinar el orden de modulación.
    Descripción: fija el valor del orden de modulación M considerado para cada una de los esquemas de modulación.

- Canal de transmisión

  - **Nombre:** ingresar E_b/N_0.
    Descripción: permite ingresar el valor de E_b/N_0 .

- Demodulador digital

  - **Nombre:** seleccionar el esquema de demodulación.
    Descripción: selecciona el esquema de demodulación a utilizar según se haya establecido en el proceso de modulación.

  - **Nombre:** determinar el orden de demodulación.
    Descripción: fija el valor del orden de demodulación M considerado para cada una de los esquemas de demodulación.

- Demodulador digital modificado

  - **Nombre:** seleccionar el esquema de demodulación.
    Descripción: selecciona el esquema de demodulación a utilizar según se haya establecido en el proceso de modulación.

  - **Nombre:** determinar el orden de modulación.
    Descripción: fija el valor del orden de demodulación M considerado para cada una de los esquemas de demodulación.

  - **Nombre:** calcular MER
    Descripción: calcula el valor de la tasa de error de modulación MER, comparando la señal de salida del canal AWGN y los símbolos ideales interpretados en la demodulación de la señal recibida $R(X)$.

- Decodificador RS

  - **Nombre**: fijar la longitud del mensaje k.
    Descripción: determina el tamaño del mensaje constituido por un número fijo de símbolos que hacen parte del campo de Galois.

  - **Nombre**: fijar longitud de la palabra codificada n.
    Descripción: determina el tamaño de palabra codificada en un número finito de símbolos, constituido por el bloque mensaje k y el bloque de redundancia.

  - **Nombre**: calcular la capacidad de corrección t.
    Descripción: operando los parámetros k y n determina la cantidad máxima de símbolos que pueden ser corregidos.

  - **Nombre**: fijar la primera raíz consecutiva FCR.
    Descripción: establece el valor de FCR para hallar las raíces del polinomio generador.

  - **Nombre**: calcular síndrome $S(X)$.
    Descripción: con la ecuación 2.3 se forma el síndrome. Se obtienen los componentes del síndrome al evaluar las $2t$ raíces del polinomio generador $g(X)$ en la palabra recibida.

  - **Nombre**: calcular polinomio localizador de error $\sigma(X)$.
    Descripción: se ejecuta el algoritmo Euclidiano teniendo en cuenta las condiciones iniciales y la condición de parada, la cual es dada por la capacidad de corrección t.

  - **Nombre**: localizar posiciones de error.
    Descripción: se ejecuta el algoritmo Chien con el objetivo de encontrar las posiciones de error.

  - **Nombre**: calcular valores de error.
    Descripción: se calcula los valores de error utilizando el algoritmo Forney.

  - **Nombre**: calcular la estimación de la palabra transmitida $\hat{C}(X)$.
    Descripción: se realiza la adición de la palabra recibida y el patrón de error.

- Decodificador RS con capacidad de borrado

  - **Nombre**: fijar la longitud del mensaje k.
    Descripción: determina el tamaño del mensaje constituido por un número fijo de símbolos que hacen parte del campo de Galois.

  - **Nombre**: fijar longitud de la palabra codificada n.
    Descripción: determina el tamaño de palabra codificada en un número finito de símbolos, constituido por el bloque mensaje k y el bloque de redundancia.

  - **Nombre**: calcular la capacidad de corrección t.
    Descripción: operando los parámetros k y n determina la cantidad máxima de símbolos que pueden ser corregidos.

  - **Nombre**: fijar la primera raíz consecutiva FCR.
    Descripción: establece el valor de FCR para hallar las raíces del polinomio generador.

  - **Nombre**: calcular síndrome $S(X)$.
    Descripción: utilizando la ecuación 2.3 se forma el síndrome. Se obtienen los componentes del síndrome al evaluar las $2t$ raíces del polinomio generador $g(X)$ en la palabra recibida, para la cual se asumen ceros los valores de las posiciones marcadas como borradura.

  - **Nombre**: calcular polinomio localizador de borradura $\tau(X)$.
    Descripción: teniendo las posiciones de borradura se calcula $\tau(X)$ utilizando la ecuación 2.28.

  - **Nombre**: calcular síndrome modificado $\Xi(X)$.
    Descripción: resultado de operar el síndrome $S(X)$ y el polinomio localizador borradura $\tau(X)$, utilizando la ecuación 2.27.

  - **Nombre**: calcular polinomio localizador de error $\sigma(X)$.
    Descripción: se ejecuta el algoritmo Euclidiano teniendo en cuenta las condiciones iniciales y la condición de parada, la cual es dada por la capacidad de corrección t y un cociente que relaciona el número total de borraduras teniendo en cuenta si el número de borraduras es par o impar.

  - **Nombre**: localizar posiciones de error.
  - **Descripción**: se ejecuta el algoritmo Chien, el cual consiste en evaluar los componentes de 1 hasta α^{n-1} en el polinomio localizador de error para obtener

sus raíces, las cuales son evaluadas en la ecuación 2.15 y así determinar las posiciones de error.

- **Nombre**: encontrar los valores de error y de borradura.
  Descripción: establecido el FCR y conociendo las posiciones de error y las de borradura se hallan los valores de error y de las borraduras utilizando el algoritmo Forney modificado, el cual involucra el polinomio localizador error-borradura $\Psi(X)$, siendo este el producto del polinomio localizador de error $\sigma(X)$ y el polinomio localizador de borradura $\tau(X)$.

- **Nombre**: calcular la estimación de la palabra transmitida $\hat{C}(X)$.
  Descripción: consiste en adicionar el patrón de error-borradura $\tilde{E}(X)$, formado por los valores de error y de borradura en sus respectivas posiciones, a la nueva palabra recibida $\tilde{R}_{f=0}(X)$, obteniendo así la estimación de la palabra codificada $\hat{C}(X)$.

3.2.5 Notación en carta de especificación

En esta sección se muestra la notación en carta de especificación de la fuente de información, codificador RS, modulador digital, canal de transmisión, demodulador digital, decodificador RS y decodificador RS con capacidad de borrado, las cuales van desde la figura 3.10 hasta la figura 3.15.

Figura 3.10 Carta de especificación de la fuente de información.

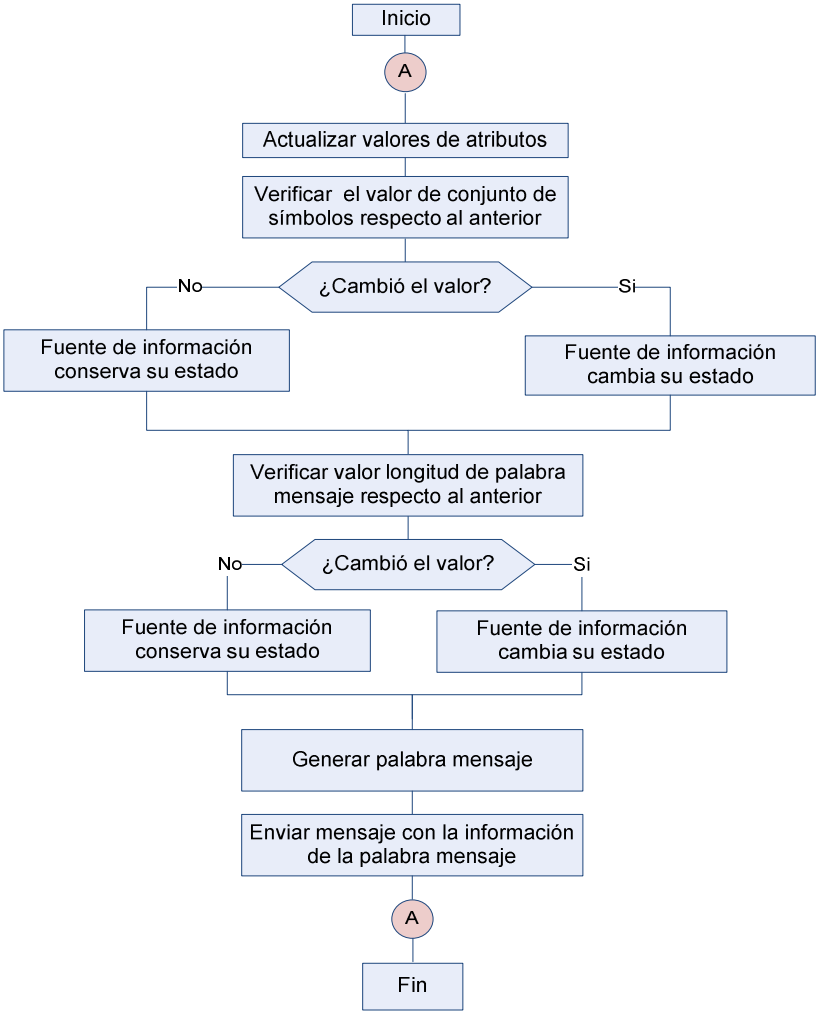


Figura 3.11 Carta de especificación del codificador RS.

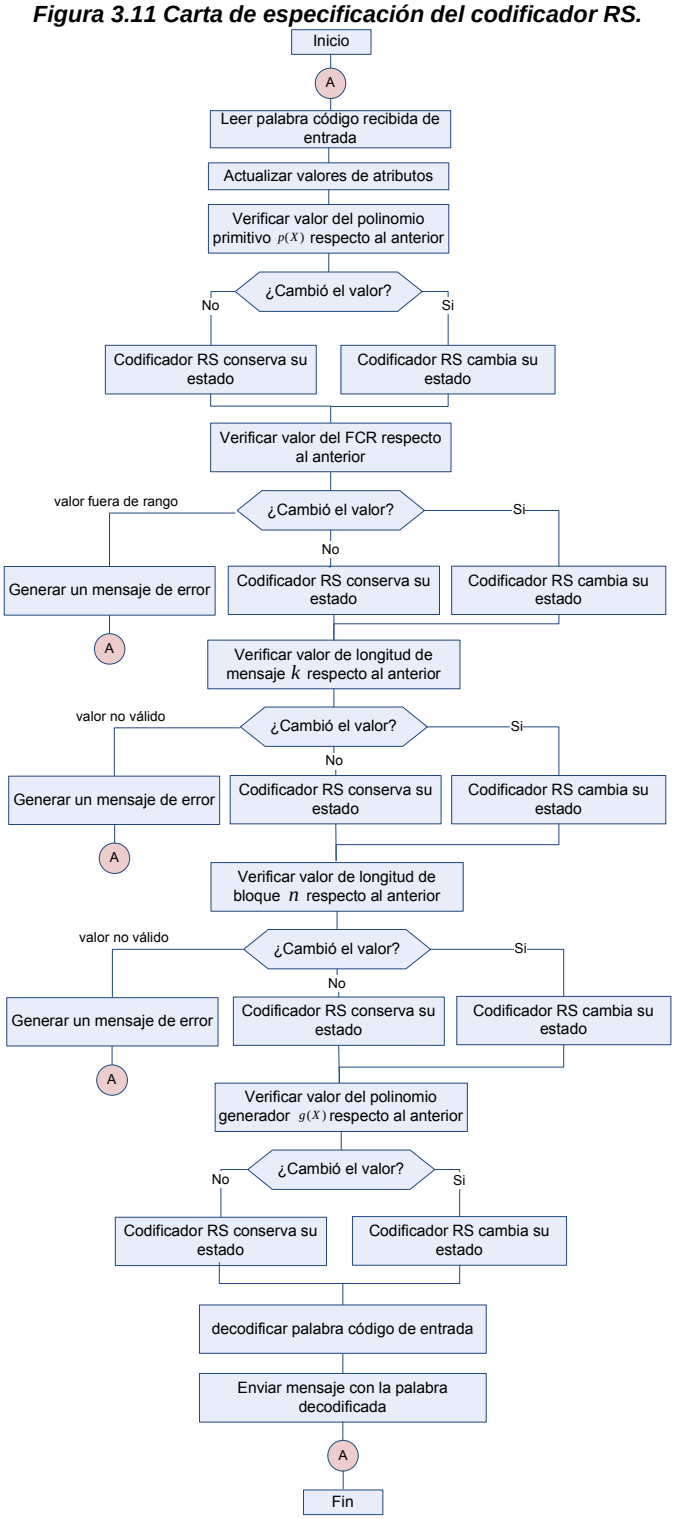


Figura 3.12 Carta de especificación del modulador digital.

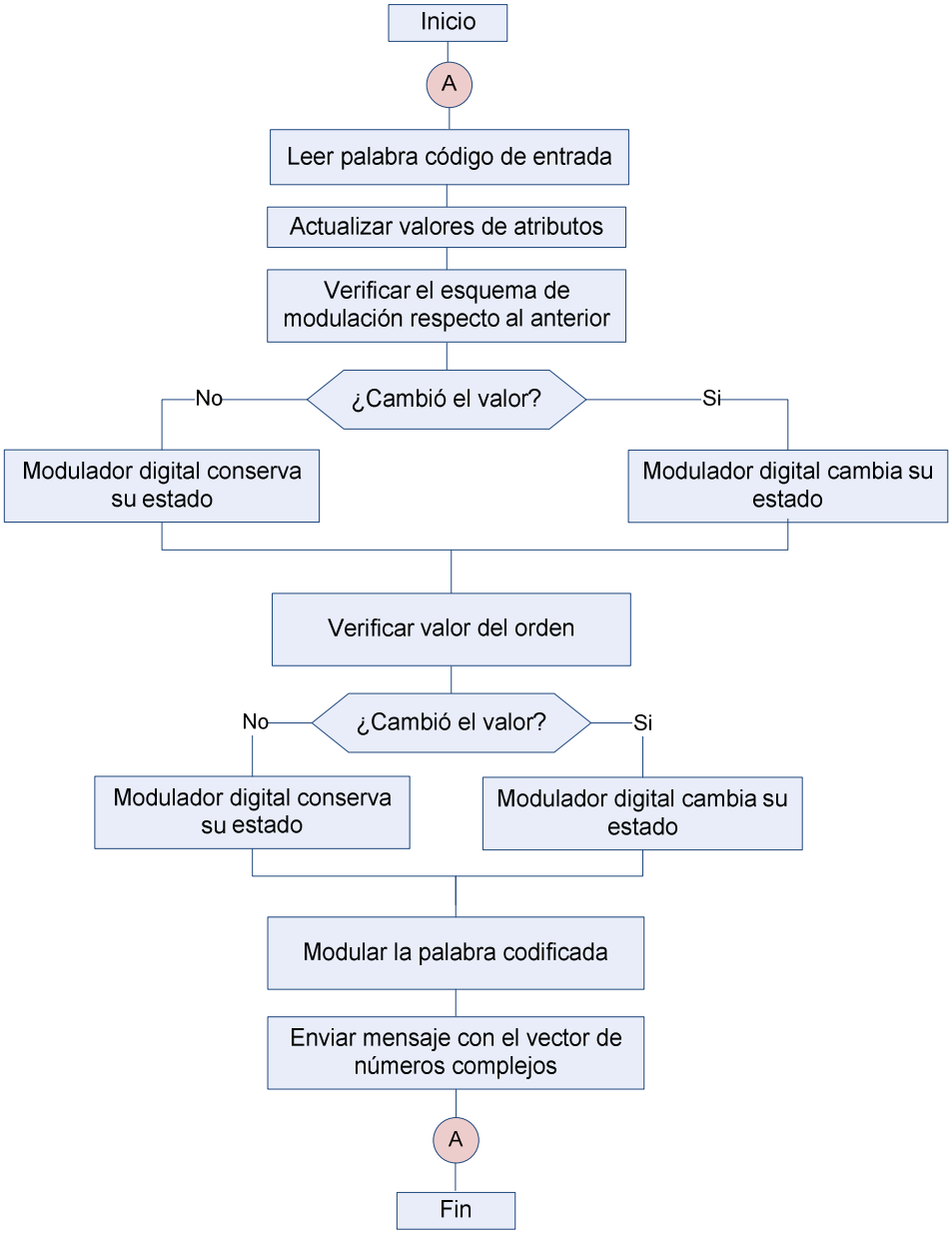


Figura 3.13 Carta de especificación del canal de comunicación.

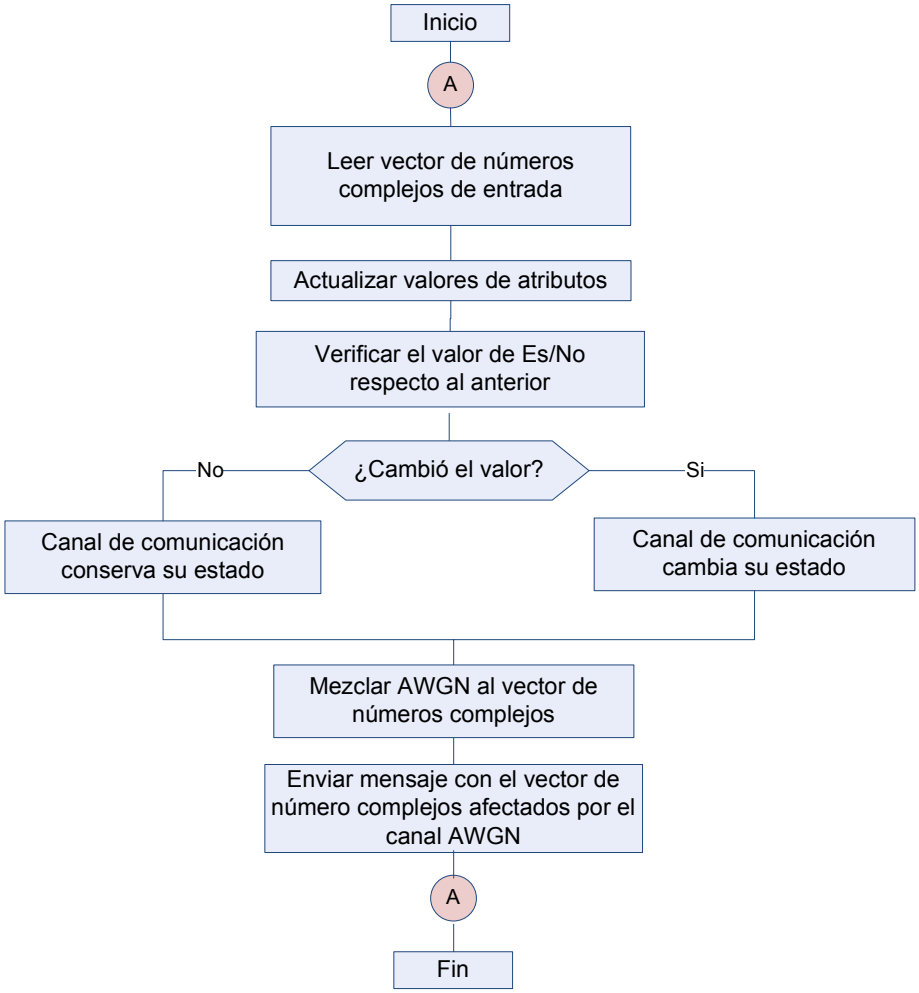


Figura 3.14 Carta de especificación del demodulador digital y demodulador digital modificado.

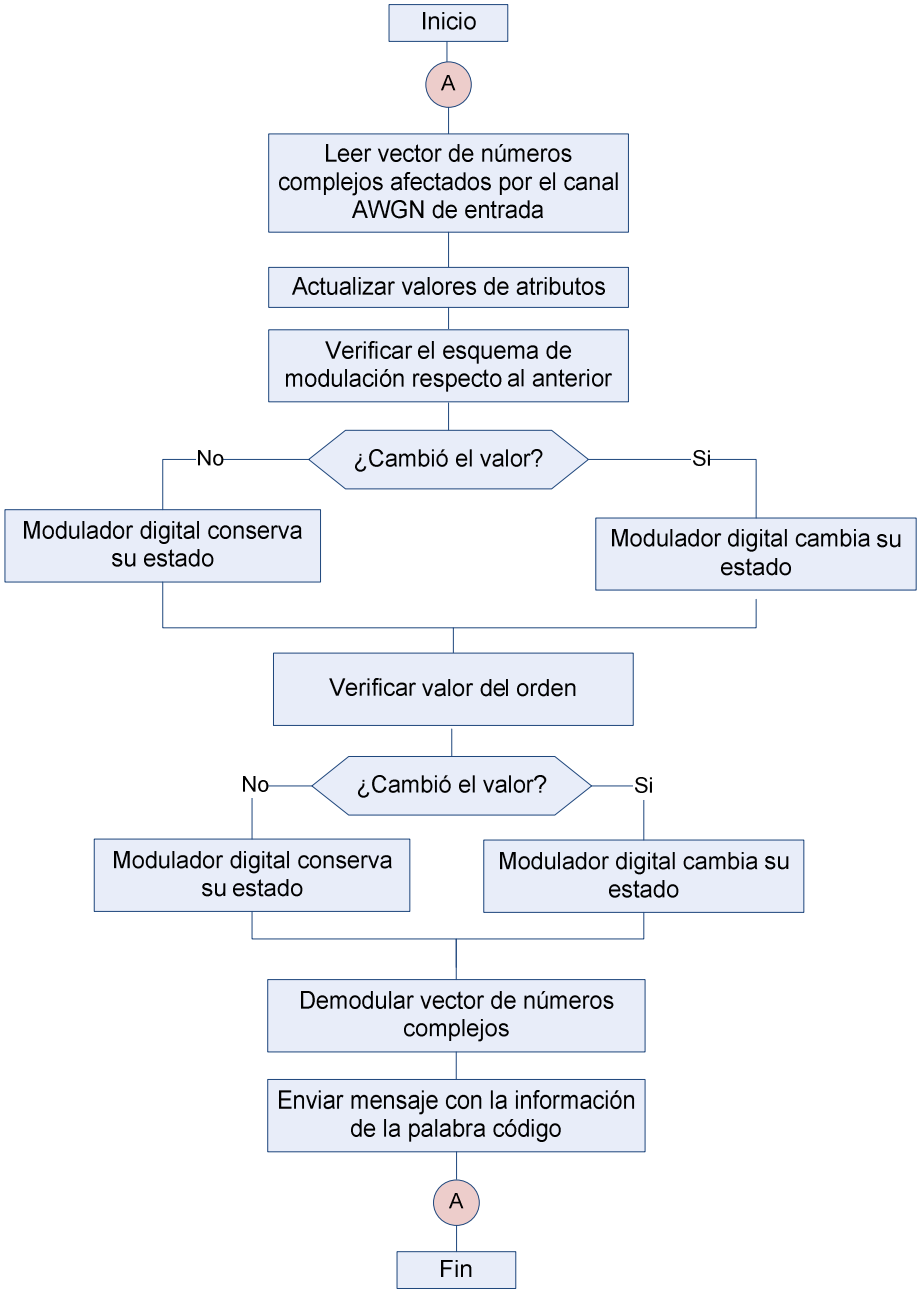
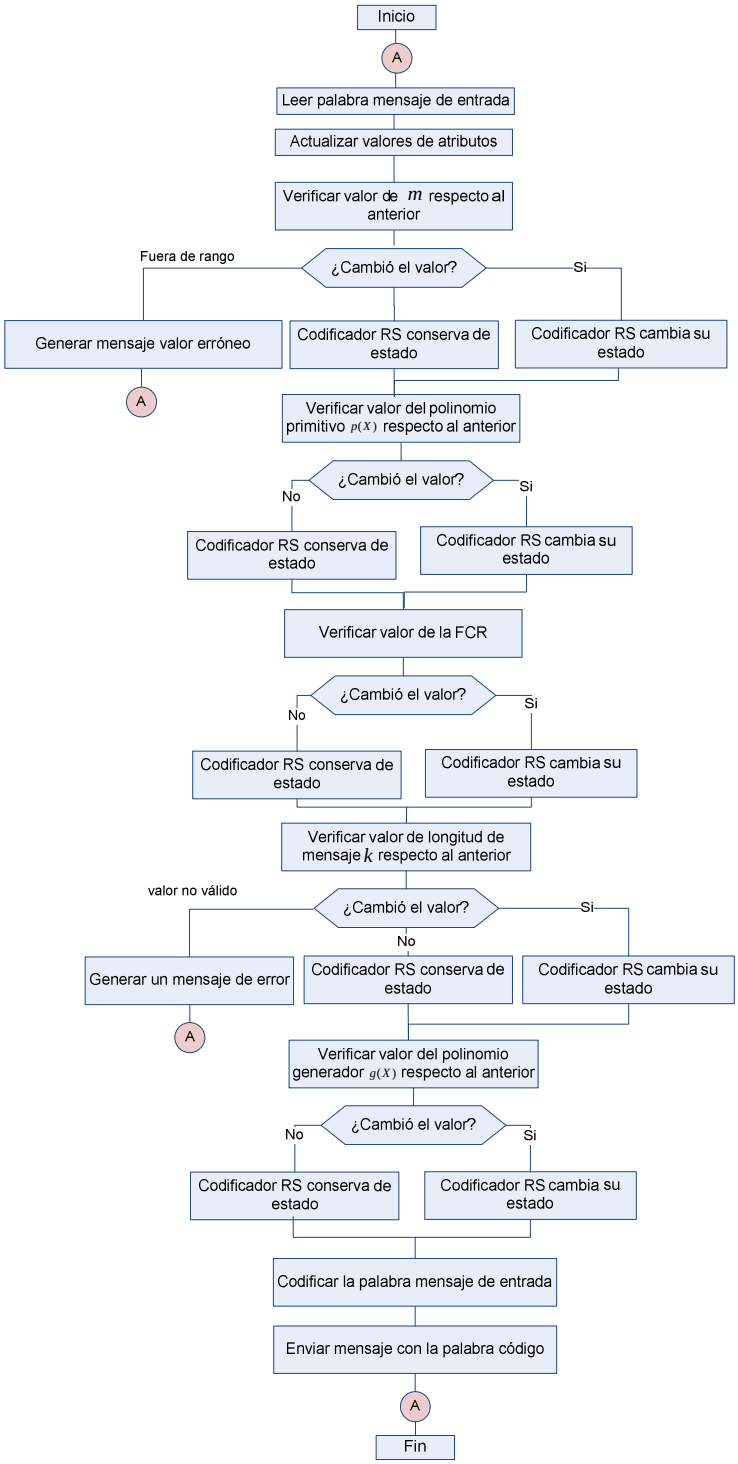


Figura 3.15 Carta de especificación del decodificador RS y decodificador RS con capacidad de borrado



3.3 MODELO DE SIMULACIÓN

Se presenta un modelo de simulación realizando la abstracción del sistema real, conservando la claridad del planteamiento inicial e ignorando aspectos poco relevantes, dando énfasis a aquellos aspectos de mayor importancia, logrando reducir la complejidad del sistema.

Para llevar a cabo esta tarea, el sistema es dividido en tres grandes bloques: transmisor, canal de comunicación y receptor, facilitando su estudio y reduciendo su complejidad.

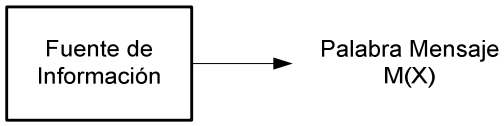
3.3.1 Modelo de simulación del transmisor

El modelo de simulación del transmisor se encuentra compuesto por los objetos: fuente de información, codificador de canal RS y modulador digital. En las secciones 3.3.1.1, 3.3.1.2 y 3.3.1.3 se describe la caracterización de los objetos del modelo de simulación del transmisor.

3.3.1.1 Caracterización de la fuente de información

En la figura 3.16 se muestra el diagrama en bloque fuente de información con su respectiva señal.

Figura 3.16 Fuente de información



Señales de entrada: ninguna.

Señales de salida: palabra mensaje.

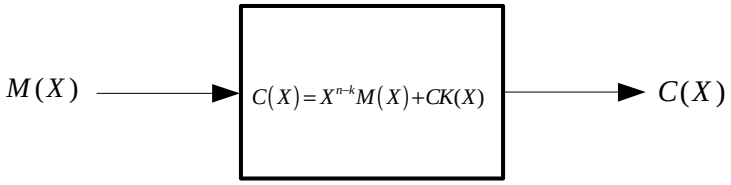
Variables de entrada: tamaño de palabra mensaje.

Señales y variables propias de procesos internos: ninguna.

3.3.1.2 Caracterización del codificador RS

En la figura 3.17 se muestra el diagrama en bloque del codificador $RS(n,k)$ con sus señales de entrada y salida

Figura 3.17 Codificador $RS(n,k)$



Señales de entrada: palabra mensaje $M(X)$.

Señales de salida: palabra codificada $C(X)$.

Variables de entrada: las variables que determina el comportamiento del objeto codificador RS son:

- Polinomio primitivo $p(X)$.
- Longitud de símbolo m.
- Tamaño del campo q.
- Primera raíz consecutiva FCR.
- Polinomio generador $g(X)$.
- Longitud del mensaje k.
- Longitud de la palabra codificada n.

Variables de salida: ninguna.

Señales y variables propias de procesos internos: los procesos internos del codificador RS son: desplazamiento de posición, generador de redundancia y constructor de palabra codificada. Estos procesos se describen a continuación.

- **Desplazamiento de posición**

  Señales de entrada:

  - Palabra mensaje $M(X)$.
  - Factor de desplazamiento X^{2t}.

  Señales de salida: mensaje con corrimiento de grado en X^{2t} posiciones.

  Variable de entrada: ninguna.

Descripción matemática: consiste en la multiplicación de dos polinomios que dan como resultado un corrimiento o cambio de grado del mensaje $M(X)$, en un factor de X^{n-k}.

- **Generador de redundancia**

Señales de entrada:

- Polinomio generador $g(X)$.
- Mensaje con corrimiento de grado en X^{2t} posiciones.

Variables de salida: bloque de símbolos de redundancia $CK(X)$.

Variable de entrada: ninguna.

Variables de salida: ninguna.

Descripción matemática: el bloque de redundancia $CK(X)$ se obtiene realizando la función mod entre el polinomio generador $g(X)$ y la palabra mensaje sometida a un desplazamiento de grado $M(X)$ como se trató en la ecuación 1.20. El polinomio generador $g(X)$ es construido teniendo en cuenta la primera raíz consecutiva FCR y la longitud de la redundancia $n-k$ como se observa en la ecuación 1.21.

- **Constructor de palabra codificada**

Señales de entrada:

- Mensaje con desplazamiento de grado en X^{2t} posiciones.
- Bloque de símbolos de redundancia $CK(X)$.

Señales de salida: palabra codificada $C(X)$

Variables de entrada: ninguna.

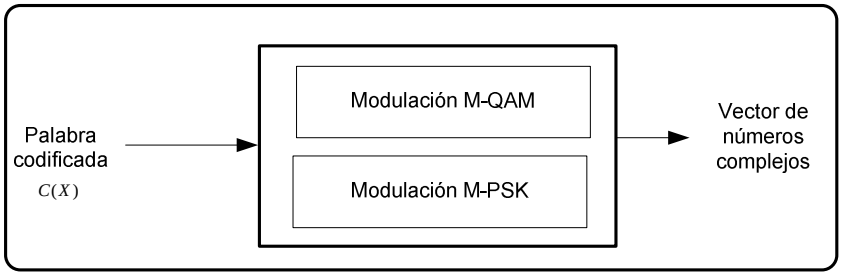
Variables de salida: ninguna.

Descripción matemática: se adiciona a la palabra mensaje $M(X)$ el bloque de redundancia $CK(X)$, dando como resultado un código sistemático, donde el mensaje se mantiene en su forma original y los símbolos de redundancia se añaden al mensaje [17].

3.3.1.3 Caracterización del modulador digital

En la figura 3.18 se presenta la caracterización del modelo de este objeto, el cual depende del esquema de modulación a utilizar, siendo estos M-PSK Y M-QAM.

Figura 3.18 Modulador digital



Señales de entrada: palabra codificada $C(X)$.

Señales de salida: vector de números complejos que representan los símbolos modulados en el esquema elegido, M-PSK o M-QAM.

Variables de entrada:

- Orden de modulación M.
- Esquema de modulación M-PSK o M-QAM.

Variables de salida: ninguna.

Descripción matemática: consiste en un vector de números complejos resultado de la modulación en M-PSK o M-QAM que describen la palabra codificada transmitida. Para el caso de M-PSK se trabajan con un orden de modulación de 2 y 8, mientras que para M-QAM el orden de modulación es de 4 y 16.

3.3.2 Modelo de simulación del canal de transmisión

La caracterización de las señales de entrada, las señales de salida, las variables de entrada y las variables de salida se describen para el canal de transmisión, mostrado en la figura 3.19.

Figura 3.19 Canal de transmisión

Señales de entrada: vector de números complejos que corresponde a la señal modulada en banda base, según sea el esquema de modulación M-PSK o M-QAM elegido.

Señales de salida: vector de números complejos de la señal transmitida alterado, por las características del ruido Gaussiano blanco aditivo.

Variables de entrada: variaciones del factor E_b/N_0.

Variables de salida: ninguna.

Descripción matemática: simula un canal AWGN, el cual presenta un ruido Gaussiano aditivo blanco que deteriora la señal trasmitida. El ruido Gaussiano aditivo blanco es producido por las corrientes eléctricas, agitaciones térmicas de los elementos resistivos o por interferencia de señales externas [18].

3.3.3 Modelo de simulación del receptor

El modelo de simulación del receptor está compuesto por los objetos: demodulador digital, demodulador digital modificado, decodificador RS y decodificador RS con capacidad de borrado.

3.3.3.1 Caracterización demodulador digital

Señales de entrada: vector de números complejos de la señal transmitida mezclada con el ruido Gaussiano blanco aditivo.

Señales de salida: palabra recibida $R(X)$.

Variables de entrada:

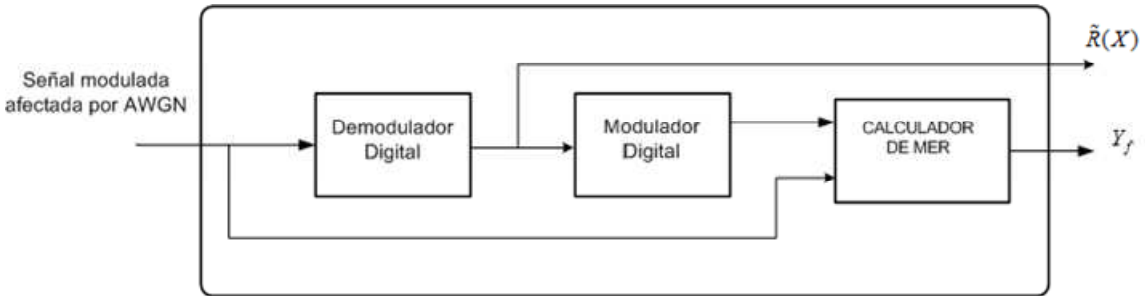
- Orden de modulación M.
- Esquema de demodulación, M-QAM o M-PSK.

Descripción matemática: toma el vector de números complejos alterado por el canal AWGN y realiza el proceso inverso a la modulación. En este proceso se debe tener en cuenta el esquema de modulación utilizado en el transmisor.

3.3.3.1 Caracterización demodulador digital modificado

En la figura 3.20 se muestra la estructura interna y caracterización del demodulador modificado.

Figura 3.20 Demodulador digital modificado



Señales de entrada: vector de números complejos de la señal transmitida mezclada con el ruido Gaussiano blanco aditivo.

Señales de salida:

- Palabra recibida $\tilde{R}(X)$.
- Polinomio con las posiciones marcadas como poco fiables Y_f .

Variables de entrada:

- Orden de modulación M
- Esquema de demodulación, M-QAM o M-PSK

Señales y variables propias de procesos internos: la caracterización del demodulador digital se referencia en la sección 3.3.3.1. La señal de entrada del modulador interno es la palabra recibida $\tilde{R}(X)$ y tiene como señal de salida un vector de números complejos que representan la palabra recibida modulada. La caracterización del componente calculador MER es mostrada a continuación.

- **Calculador de MER**

  Señales de entrada:

  - Vector de números complejos que provienen del modulador interno

- Vector de números complejos de la señal transmitida alterado, por las características del ruido Gaussiano blanco aditivo.

Señal de salida: polinomio con las posiciones marcadas como poco fiables Y_f .

Variable de entrada: variación MER.

Descripción matemática: consiste en marcar las posiciones no fiables o borraduras de la palabra recibida utilizando como criterio de decisión el MER, el cual es un criterio de desempeño de los esquemas de modulación [19]. En el estándar de los sistemas DVB del Instituto Europeo de Estándares de Telecomunicaciones (ETSI, *European Telecommunications Standards Institute),* define que la MER mínima aceptable debe ser 27 dB a la entrada del Receptor de Televisión Digital o Decodificador (STP,*Set Top Box*) [15].

3.3.3.2 Caracterización del decodificador RS

Señales de entrada: palabra recibida $R(X)$, la cual corresponde a la palabra codificada $C(X)$, alterada por un patrón de error $E(X)$ suministrado por el canal de transmisión.

Señales de salida: estimación de la palabra codificada $\hat{C}(X)$.

Variables de entrada: las variables de entrada corresponden a las características propias del decodificador RS.

- Polinomio primitivo $p(X)$.
- Primera raíz consecutiva FCR.
- Longitud del mensaje k.
- Longitud de la palabra codificada n.

Señales y variables propias de procesos internos: los objetos que permiten el funcionamiento del decodificador RS y constituyen su estructura interna son: calculador de síndrome, algoritmo Euclidiano, buscador Chien, algoritmo Forney, estimador de palabra codificada.

- **Calculador de síndrome**

  Señales de entrada: palabra recibida $R(X)$.

  Señales de salida: polinomio de elementos del $GF(2^m)$, conformado por los componentes de síndrome encontrados de la palabra recibida $R(X)$.

Variables de entrada: primera raíz consecutiva FCR.

Descripción matemática: evalúa las $2t$ raíces del polinomio generador en la palabra recibida, con el fin de verificar la existencia o no de errores en ella. Si el resultado es cero indica que no existen errores, de lo contrario los resultados son tomados como componentes que constituyen el síndrome obtenido con la ecuación 2.3.

- **Algoritmo Euclidiano**

Señales de entrada: polinomio de elementos del $GF(2^m)$, conformado por los componentes de síndrome encontrados de la palabra recibida $R(X)$.

Señales de salida: polinomio localizador de error $\sigma(X)$ y el polinomio evaluador de error $\Omega(X)$.

Variables de entrada: capacidad de corrección t.

Descripción matemática: algoritmo iterativo para encontrar el máximo común divisor (MCD) entre un par de polinomios. Consiste en una serie de divisiones sucesivas hasta que el grado $[r_j(X)] \leq t$, es decir, cuando el grado del polinomio residuo sea menor o igual a la capacidad de corrección. Las condiciones iniciales son: $r_0(X) = X^{2t+1}$, $r_1(X) = 1 + S(X)$, $b_0(X) = 1$ y $a_0(X) = 0$. Inicialmente se divide $r_1(X)$ entre $r_0(X)$ y se evalúa la condición $[r_j(X)] \leq t$, de cumplirse la condición cesan las iteraciones, el último residuo es tomado como el polinomio evaluador de error $\Omega(X)$ y el polinomio localizador de error es encontrado con la ecuación 2.13, de lo contrario, el residuo pasa a ser el divisor, el divisor a ser el dividendo y nuevamente se realiza la división.

- **Buscador Chien**

Señales de entrada: polinomio localizador de error reciproco $\sigma_r(X)$.

Señales de salida: polinomio que contiene las posiciones de error.

Variable de entrada: elementos que conforman el campo distinto de cero.

Descripción matemática: consiste en evaluar los elementos del campo distintos de cero en el polinomio localizador de error reciproco $\sigma_r(X)$, de manera que si $\sigma_r(z_i) = 0$, la raíz z_i es tomada como un número localizador de error, valor que es evaluado en la ecuación 2.15 para encontrar las posiciones de error X_i.

- **Algoritmo Forney**

  Señales de entrada:

  - Polinomio que contiene los números localizadores de error z_i.
  - Polinomio localizador de error $\sigma(X)$.
  - Polinomio evaluador de error $\Omega(X)$.

  Señales de salida: polinomio con los valores de error que corresponden a las posiciones encontradas por el buscador Chien.

  Variables de entrada: primera raíz consecutiva FCR.

  Descripción matemática: conociendo la primera raíz consecutiva FCR, las posiciones de error, el polinomio localizador de error $\sigma(X)$ y el polinomio evaluador de error $\Lambda(X)$, se pueden encontrar los valores de error de las posiciones de error utilizando la ecuación 2.16, donde $\sigma'(X)$ es la derivada formal del polinomio localizador de error.

- **Estimador de palabra codificada**

  Señales de entrada: como señales de entrada se tienen dos polinomios que constituyen el patrón de error adicionado por el canal a la señal deseada y el polinomio con la palabra entregada por el demodulador.

  - Polinomio que contiene las posiciones de error.
  - Polinomio con los valores de error que corresponden a las posiciones encontradas por el buscador Chien.
  - Palabra recibida $R(X)$.

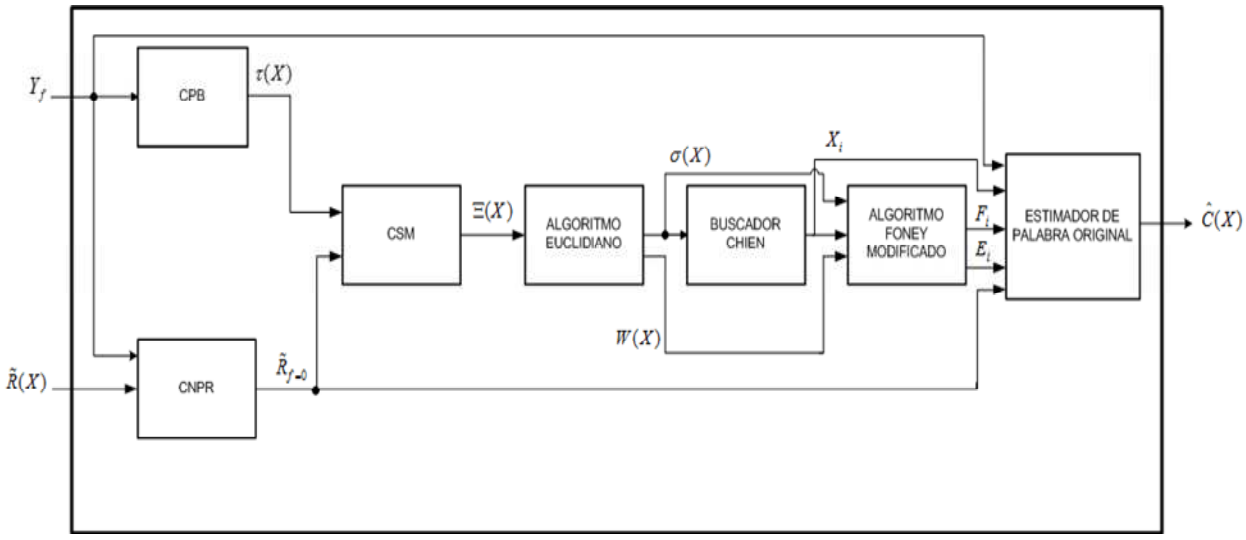
  Señales de salida: estimación de la palabra codificada $\hat{C}(X)$.

  Descripción matemática: consta de dos procesos, el primero es construir el polinomio patrón de error $\hat{E}(X)$, formado por la suma de todos los valores de error multiplicados por su respectiva posición de error utilizando la ecuación 2.17, y como segundo paso se adhiere el patrón de error a la palabra recibida entregada por el demodulador, de manera que al operar estos dos polinomios posición a posición, basado en la aritmética de Galois dé como resultado la estimación de la palabra codificada originalmente transmitida como en la ecuación 2.18.

3.3.3.3 Caracterización del decodificador RS con capacidad de borrado

La caracterización de los objetos del decodificador RS con capacidad de borrado son similares en gran parte al decodificador RS estudiado en la sección 3.3.3.2, sin embargo la capacidad de borrado en un decodificador introduce modificaciones en los procesos como también nuevas señales y variables tanto de entrada como de salida a continuación mostradas. En la figura 3.21 se muestra estructura interna del decodificador $RS(n,k)$ con capacidad de borrado.

Figura 3.21 Decodificador $RS(n,k)$ con capacidad de borrado



Señales de entrada:

- Palabra recibida $\tilde{R}(X)$.
- Polinomio con las posiciones marcadas como poco fiables Y_f .

Señales de salida: estimación de la palabra codificada transmitida $\hat{C}(X)$.

Variables de entrada: las variables de entrada corresponden a las características propias del decodificador RS con capacidad de borrado.

- Polinomio primitivo $p(X)$.
- Primera raíz consecutiva FCR.
- Longitud de símbolos del mensaje k.
- Longitud de la palabra codificada n.

Señales y variables propias de procesos internos: los objetos que permiten el funcionamiento del decodificador RS con capacidad de borrado y constituyen su estructura interna son: constructor polinomio de borraduras (CPB), constructor nueva palabra recibida (CNPR), calculador de síndrome modificado (CSM), algoritmo Euclidiano, buscador Chien, algoritmo Forney modificado (AFM), estimador de palabra original (EPO).

- **Constructor polinomio localizador de borraduras**

  Señales de entrada: vector que contiene con las posiciones marcadas como poco fiables Y_f .

  Señales de salida: polinomio formado por las borraduras $\tau(X)$.
  Variables de entrada: número de borraduras f.

  Descripción matemática: se realiza la productoria con el total de las borraduras, evaluando sus posiciones de la forma $Y_f = \alpha^{j_f}$ en la ecuación 2.27.

- **Constructor de nueva palabra recibida**

  Señales de entrada:

  - Palabra recibida $\tilde{R}(X)$.
  - vector que contiene con las posiciones marcadas como poco fiables Y_f .

  Señales de salida: nueva palabra recibida $\tilde{R}_{f=0}(X)$.

  Variables de entrada: ninguna.

  Descripción matemática: se trata de evaluar valor cero en las posiciones de la palabra recibida $\tilde{R}(X)$ que el demodulador identifica como borradura, formando la nueva palabra recibida $\tilde{R}_{f=0}(X)$

- **Calculador del síndrome modificado**

  Señales de entrada:

  - Nueva palabra recibida $\tilde{R}_{f=0}(X)$.
  - Polinomio localizador de borradura $\tau(X)$.

  Señales de salida: polinomio que representa el síndrome modificado $\Xi(X)$.

  Variables de entrada: primera raíz consecutiva FCR.

Descripción matemática: una vez obtenida la nueva palabra recibida, se evalúa en ella las $2t$ raíces del polinomio generador en la palabra recibida, y los resultados son tomados como componentes que constituyen el síndrome obtenido utilizando la ecuación 2.3. Este resultado es operado con el polinomio localizador de borraduras $\tau(X)$ con la ecuación 2.27.

- **Algoritmo Euclidiano**

Señales de entrada: polinomio que representa el síndrome modificado $\Xi(X)$.

Señales de salida: polinomio localizador de error $\sigma(X)$ y el polinomio evaluador de error-borradura $W(X)$ para codificación con capacidad de borrado, el cual es el último residuo obtenido cuando se cumple la condición en la que paran las iteraciones del algoritmo.

Variables de entrada: corresponde a la condición con la cual paran las iteraciones del algoritmo Euclidiano.

- Capacidad de corrección errores t.
- Número de borraduras f.

Descripción matemática: el proceso realizador por el algoritmo Euclidiano cuando se trabaja con capacidad de borrado, es similar al realizado cuando no se trabaja con capacidad de borrado, considerando las siguientes salvedades:

En la ecuación 3.1 se muestran las condiciones iniciales del algoritmo euclidiano.

$$r_0(X) = X^{2t+1},\ r_1(X) = 1 + \Xi(X),\ b_0(X) = 1 \text{ y } a_0(X) = 0 \qquad (3.1)$$

En la ecuación 3.2 se muestran las condiciones de parada del algoritmo Euclidiano

$$\text{grado}[\mathrm{r}_i(\mathrm{X})] \leq \begin{Bmatrix} \mathrm{t}+\dfrac{f}{2}, & si \quad f \text{ es par} \\ \mathrm{t}+\dfrac{f-1}{2}, & si \quad f \text{ es impar} \end{Bmatrix} \qquad (3.2)$$

De manera que se realizan las divisiones sucesivas hasta cumplir la condición de parada, momento en el que último residuo es tomado como el polinomio evaluador $W(X)$ y el polinomio localizador de error es encontrado con la ecuación 2.13.

- **Buscador Chien**

Señales de entrada: polinomio localizador de error reciproco $\sigma_r(X)$.

Señales de salida: polinomio que contiene las posiciones de error.

Variable de entrada: elementos que conforman el campo distinto de cero.

Descripción matemática: esta descripción es idéntica a la presentada cuando no se trabaja con capacidad de borrado.

- **Algoritmo Forney modificado**

Señales de entrada:

- Polinomio que contiene los números localizadores de error z_i.
- Polinomio localizador de error $\sigma(X)$.
- Polinomio evaluador de error-borradura $W(X)$.
- vector que contiene con las posiciones marcadas como poco fiables Y_f .

Señales de salida: polinomio con los valores de error que corresponden a las posiciones encontradas por el buscador Chien.

Variables de entrada: primera raíz consecutiva FCR.

Descripción matemática: conociendo el polinomio localizador de borradura y el polinomio localizador de error se crea el polinomio localizador de error-borradura $\Psi(X)$, el cual junto a la primera raíz consecutiva FCR, el polinomio evaluador de error-borradura $W(X)$, las posiciones de error y borradura son utilizadas en la ecuación 2.31 y 2.32 para encontrar los valores de error y de borradura respectivamente.

- **Estimador de palabra codificada**

Señales de entrada: como señales de entrada se tienen dos polinomios que constituyen el patrón de error adicionado por el canal a la señal deseada y un polinomio con la nueva palabra recibida.

- Polinomio que contiene las posiciones de error.
- Polinomio que contiene las posiciones de borradura.
- Polinomio con los valores de error que corresponden a las posiciones encontradas por el buscador Chien.
- Polinomio con los valores de las borraduras que corresponden a las posiciones dadas por el demodulador modificado.
- Nueva palabra recibida $\tilde{R}_{f=0}(X)$.

Señales de salida: estimación de la palabra codificada transmitida $\hat{C}(X)$.

Descripción matemática: consta de dos procesos, el primero de ellos es construir el polinomio que representa el patrón de error-borradura $\tilde{E}(X)$, formado por la sumatoria de todos los valores de error en su respectiva posición de error y la sumatoria de todos los valores de borradura en su respectiva posición de borradura. Este proceso se calcula con la ecuación 2.33. Como segundo proceso, se adhiere el patrón de error-borradura a la palabra recibida $\tilde{R}_{f=0}(X)$ entregada por el demodulador, de manera que al operar estos dos polinomios posición a posición, basado en la aritmética de Galois, dé como resultado la estimación de la palabra codificada originalmente transmitida establecida en 2.34.

3.4 EVALUACIÓN DEL MODELO Y LOS PARAMETROS ESTIMADOS

Con el propósito de ajustar valores, límites y rangos de las variables de los diferentes procesos, se realizan pruebas de ensayo y error constatando la validez del análisis presentado. Dentro de los factores a tener en cuenta en la evaluación del modelo se encuentran:

- **Verificación de la correcta descripción de variables**

  Examinando el direccionamiento y almacenamiento de las diversas variables de entrada, salida y variables intermedias involucradas en los procesos del modelo, se eliminan variables redundantes del modelo se simulación.

- **Complejidad resultante para cada objeto**

  El sistema ha sido dividido en objetos claramente definidos que permiten un buen entendimiento del comportamiento del sistema. De igual forma aquellos objetos de mayor complejidad se descomponen en objetos que describen procesos internos necesarios para proporcionar una respuesta específica. Estos objetos que hacen parte de otros objetos son estudiados de forma individual, los cuales son debidamente descritos por medio de atributos y procesos dando como resultado disminución en la complejidad de los objetos.

- **Evaluación de los objetivos**

  Teniendo en cuenta el desarrollo del modelo de simulación planteado, donde se observa la descripción de factores que permiten hacer una abstracción del sistema real y realizar una evaluación de su comportamiento, se da por cumplido el primer objetivo que consiste en generar el modelo de simulación del decodificador RS con capacidad de borrado.

En el capítulo 4, para dar cumplimiento al segundo objetivo es necesario el desarrollo de la etapa de implementación del modelo, se diseñan un plan de pruebas de manera que permita evaluar el comportamiento del decodificador RS con capacidad de borrado, siendo posible la recolección, interpretación y comparación de resultados.

4. ANÁLISIS DEL DESEMPEÑO DEL DECODIFICADOR REED-SOLOMON QUE UTILIZA EL ALGORITMO EUCLIDIANO CON CAPACIDAD DE BORRADO

El análisis y plan de pruebas del desempeño del decodificador RS que utiliza el algoritmo Euclidiano con capacidad de borrado se diseñó con base en la Metodología de Simulación de Equipos de Telecomunicaciones [1]. Esta simulación se desarrolló con la herramienta software MATLAB® Release 2010, debido a que provee funciones nativas utilizadas en la aritmética de campos de Galois y a su gran capacidad de procesamiento de señales de comunicaciones.

El plan de pruebas para la evaluación y análisis del desempeño del decodificador con capacidad de borrado se realizan en tres etapas.

- Análisis individual al subsistema demodulador modificado: etapa que consiste en las pruebas para observar el comportamiento del demodulador modificado diseñado, comprando su función de detectar posiciones de borradura.
- Validación del modelo del decodificador RS con capacidad de borrado: etapa que consiste en las pruebas al decodificador RS con capacidad de borrado en un ambiente de errores y borraduras controladas, con el objetivo de estudiar el comportamiento variando la cantidad de errores y borraduras que son entregadas al decodificador.
- Análisis de resultados de la simulación del modelo del decodificador RS con capacidad de borrado: etapa que consiste en las pruebas al decodificador RS con capacidad de borrado y analizar las curvas de desempeño, en distinto escenarios de simulación.

4.1 ANÁLISIS INDIVIDUAL DEL SUBSISTEMA DEMODULADOR MODIFICADO

Para el análisis individual del demodulador modificado se realizó el diseño de un demodulador que marca las posiciones de borrado según el parámetro de la variación de la MER descrito en la sección 2.2.1, el cual es un criterio de desempeño de los moduladores en esquemas de modulación avanzadas como se vio en la descripción matemática del demodulador modificado.

La MER definida en la ecuación 2.24, establece una relación entre la sumatoria de todos símbolos ideales recibidos y la sumatoria de la magnitud de todos los vectores error

recibidos[17]. Así mismo se puede definir una tasa de error de modulación para el k-*esimo* símbolo de modulación MER_k, como en la ecuación 4.1.

$$MER_k = 10 * \log_{10} \left(\frac{\frac{1}{N} \sum_{n=1}^{N} (\bar{I}_k^{\,2} + \bar{Q}_k^2)}{(\delta I_k^2 + \delta Q_k^2)} \right) dB \tag{4.1}$$

La cual puede ser utilizada para observar que tan distante se encuentra un símbolo recibido del símbolo ideal demodulado.

Si se define la magnitud del vector error como en la ecuación 4.2.

$$\left| e_j \right| = \sqrt{(\delta I_j^2 + \delta Q_j^2)} \tag{4.2}$$

El doble de la magnitud del vector error sería de la forma de la ecuación 4.3.

$$2\left| e_j \right| = \sqrt{4(\delta I_j^2 + \delta Q_j^2)} \tag{4.3}$$

Así, reemplazando la ecuación 4.3 en la ecuación 4.1 se obtiene la expresión de MER para un símbolo alejado el doble de la distancia, se obtiene la ecuación 4.4.

$$MER_j\Big|_{2d} = 10 * \log_{10} \left(\frac{\frac{1}{N} \sum_{n=1}^{N} (\bar{I}_j^{\,2} + \bar{Q}_j^2)}{(\delta I_j^2 + \delta Q_j^2)} \right) dB - 6dB \tag{4.4}$$

Realizando el análisis análogo para un símbolo que se encuentre a la mitad de la distancia se obtiene la ecuación 4.5.

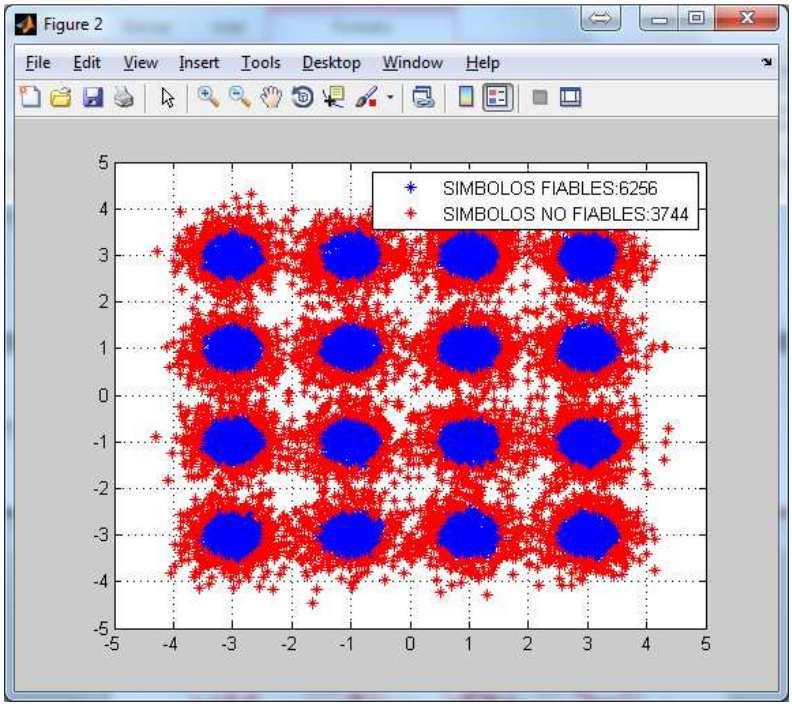
$$MER_j\Big|_{\frac{d}{2}} = 10 * \log_{10} \left(\frac{\frac{1}{N} \sum_{n=1}^{N} (\bar{I}_j^{\,2} + \bar{Q}_j^2)}{(\delta I_j^2 + \delta Q_j^2)} \right) dB + 6dB \tag{4.5}$$

La medición del MER como parámetro de desempeño de los demoduladores digitales, puede asimilarse a regiones de decisión en las constelaciones de los diferentes esquemas de modulación. Para observar el desempeño del demodulador modificado se generó una trama de 10000 símbolos, los cuales fueron modulados utilizando diferentes esquemas y expuestos a ruido blanco Gaussiano aditivo. Posterior a este proceso se

[17] Cada vector error es la diferencia entre el símbolo ideal recibido y el símbolo recibido.

calculó la MER del demodulador, seguido del cálculo del valor de MER_k para cada uno de los 10000 símbolos modulados. La figura 4.1 muestra un diagrama de constelación 16-QAM con SNR=15 dB

Figura 4.1 Diagrama de constelación para 16-QAM con SNR=15 dB.



Como se aprecia en la figura 4.1, la visualización y clasificación de los símbolos fiables y no fiables se establece al comparar el valor de MER_k para cada símbolo y el valor de referencia en función de la MER. El tamaño de las regiones de fiabilidad y de no fiabilidad están en función del SNR y de variación del MER. Según estos valores se definen los símbolos fiables y los símbolos no fiables dependiendo de la dispersión con respecto al punto teórico donde se encuentren. La interfaz gráfica del modelo de simulación del demodulador modificado se describe en el apéndice A.1.

El comportamiento y valor de la MER cambia en función de la SNR para diferentes esquemas de modulación. En la tabla 4.1 se consignan la variación de la MER respecto a valores de SNR entre 0 a 30 dB, para los esquemas de modulación 8-PSK, 4-QAM, 16-QAM.

Tabla 4.1 Valor de MER para distintos esquemas de modulación.

SNR (dB)	4-QAM (dB)	8-PSK (dB)	16-QAM (dB)
0	1,74	2,97	4,94
1	2,55	3,84	5,84
2	3,24	4,68	6,70
3	3,97	5,42	7,48
4	4,74	6,19	8,21
5	5,50	6,95	8,87
6	6,31	7,64	9,46
7	7,16	8,35	10,04
8	8,09	9,07	10,54
9	9,00	9,79	11,03
10	10,02	10,58	11,56
11	11,01	11,37	12,15
12	12,00	12,21	12,76
13	13,01	13,12	13,47
14	13,99	14,05	14,25
15	14,99	15,03	15,14
16	16,00	16,02	16,04
17	17,02	17,01	17,01
18	18,00	18,02	18,00
19	18,98	18,99	19,01
20	20,00	19,99	20,02
21	21,01	21,01	21,00
22	21,98	21,99	22,00
23	22,97	22,99	23,01
24	24,02	24,01	24,00
25	24,99	24,98	25,01
26	26,02	26,01	26,01
27	26,98	26,99	27,00
28	28,01	28,00	28,00
29	28,99	29,02	29,00
30	30,00	30,00	29,99

En la tabla 4.1 se observa que la relación entre MER y SNR tiende a 1 a medida que el SNR crece. Este comportamiento se aprecia mejor para valores mayores o iguales a 12 dB, mientras que para valores bajos de SNR la relación varía según el esquema de

modulación. En la tabla 4.2 se muestran los resultados de la cantidad de símbolos fiables y los símbolos no fiables, para un valor fijo del parámetro "variación MER", con variaciones del valor SNR y un esquema de modulación dado. El parámetro "variación MER" es la diferencia entre la MER_k de un símbolo y la MER del demodulador.

Tabla 4.2 Datos de número de símbolos fiables y símbolos no fiables en función de SNR.

ESQUEMA DE MODULACION: 16-QAM			
VARIACIÓN MER:6 dB			
SNR (dB)	SIMBOLOS FIABLES	SIMBOLOS NO FIABLES	MER (dB)
0	4233	5767	4,92
1	3542	6458	5,86
2	3061	6939	6,64
3	2718	7282	7,52
4	2357	7643	8,18
5	2093	7907	8,86
6	1866	8134	9,49
7	1750	8250	9,99
8	1662	8338	10,55
9	1697	8303	11,05
10	1704	8296	11,59
11	1826	8174	12,13
12	1919	8081	12,74
13	1936	8064	13,39
14	2086	7914	14,25
15	2138	7862	15,14
16	2168	7832	16,01
17	2166	7834	17,03
18	2224	7776	17,94
19	2241	7759	18,99
20	2208	7792	20,00
21	2221	7779	20,97
22	2296	7704	21,97
23	2202	7798	22,97
24	2210	7790	24,01
25	2249	7751	25,06
26	2228	7772	26,06
27	2298	7702	26,98
28	2198	7802	27,98
29	2198	7802	29,01
30	2228	7772	30,00

En la tabla 4.2 se muestra un ejemplo de las tablas de datos, donde se observa como varía la cantidad de símbolos no fiables en función de variación del MER. En esta tabla se observa que la variación en la cantidad de símbolos fiables y no fiables se vuelve estable a medida que el SNR aumenta. En el apéndice B se encuentran consignados los datos del número de símbolos fiables y símbolos no fiables en función de SNR para variaciones de la MER para diferentes esquemas de modulación.

Para un esquema de modulación 16-QAM, disminuyendo el valor del parámetro "variación MER" desde 9 dB hasta -9 dB, la cantidad de símbolos no fiables también decrece.

Tabla 4.3 Símbolos fiables y símbolos no fiables vs variación MER en 16-QAM.

ESQUEMA DE MODULACION :16-QAM				
VARIACIÓN MER (dB)	SIMBOLOS FIABLES	SIMBOLOS NO FIABLES	TOTAL SIMBOLOS	% SIMBOLOS NO FIABLES
9	1174	8826	10000	88,26
6	2256	7744	10000	77,44
5	2786	7214	10000	72,14
4	3406	6594	10000	65,94
3	4098	5902	10000	59,02
2	4862	5138	10000	51,38
1	5679	4321	10000	43,21
0	6523	3477	10000	34,77
-1	7339	2661	10000	26,61
-2	8095	1905	10000	19,05
-3	8718	1282	10000	12,82
-4	9191	809	10000	8,09
-5	9539	461	10000	4,61
-6	9753	247	10000	2,47
-9	9974	26	10000	0,26

La tabla 4.3 indica que la cantidad de símbolos no fiables se hace pequeña cuando se reduce el valor de la MER mediante la variación negativa de esta, debido a que se extiende la región de fiabilidad como lo indica la ecuacion 4.4.

En la tabla 4.4 y 4.5 se muestran los datos de porcentaje de símbolos no fiables obtenidos para los esquemas de modulación 4-QAM, y 8-PSK respectivamente; y un valor de SNR=15 dB.

Tabla 4.4 Símbolos fiables y símbolos no fiables vs variación MER en 4-QAM.

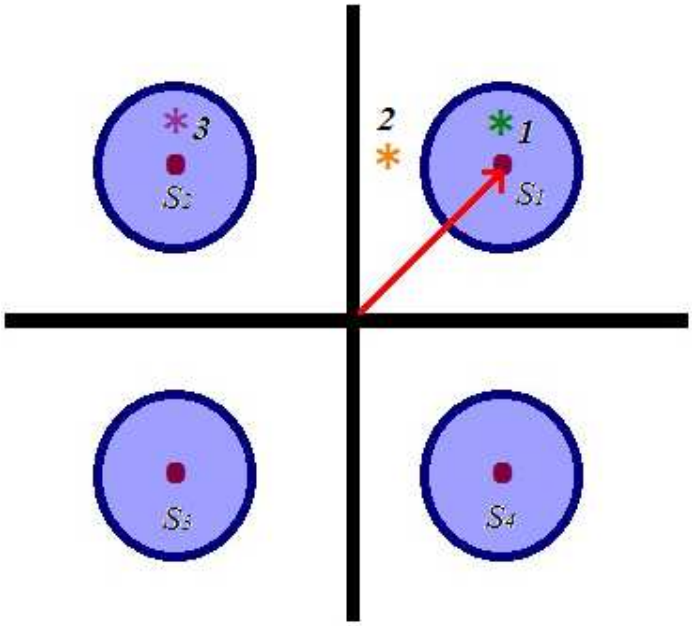
ESQUEMA DE MODULACION : 4-QAM				
VARIACIÓN MER (dB)	SIMBOLOS FIABLES	SIMBOLOS NO FIABLES	TOTAL SIMBOLOS	% SIMBOLOS NO FIABLES
9	1141	8858	10000	88,58
6	2176	7824,	10000	78,24
5	2654	7346	10000	73,46
4	3218	6782	10000	67,82
3	3881	6119	10000	61,19
2	4616	5383	10000	53,83
1	5461	4538	10000	45,38
0	6319	3680	10000	36,80
-1	7174	2826	10000	28,26
-2	7983	2017	10000	20,17
-3	8671	1329	10000	13,29
-4	9208	792	10000	7,92
-5	9585	415	10000	4,15
-6	9811	189	10000	1,89
-9	9996	4	10000	0,04

Tabla 4.5 Símbolos fiables y símbolos no fiables vs variación MER en 8-PSK.

ESQUEMA DE MODULACION : 8-PSK				
VARIACIÓN MER (dB)	SIMBOLOS FIABLES	SIMBOLOS NO FIABLES	TOTAL SIMBOLOS	% SIMBOLOS NO FIABLES
6	2219	7781	10000	77,81
5	2729	7271	10000	72,71
4	3304	6696	10000	66,96
3	3982	6018	10000	60,18
2	4740	5260	10000	52,60
1	5553	4447	10000	44,47
0	6398	3602	10000	36,02
-1	7218	2782	10000	27,82
-2	8002	1998	10000	19,98
-3	8667	1333	10000	13,33
-4	9190	810	10000	8,10
-5	9555	445	10000	4,45
-6	9788	212	10000	2,12
-9	9990	9	10000	0,09

Se debe tener en cuenta como el demodulador asigna los rótulos a los símbolos, ya que los símbolos no fiables no son equivalentes a símbolos errados, dado que el demodulador marca los símbolos como no fiables con base en la MER calculada respecto a los símbolos ideales demodulados, y no con base en la MER calculada respecto a los símbolos ideales modulados. En la gráfica 4.2, se observa la constelación del esquema de modulación 4-QAM con 3 posibles eventos al demodular un símbolo S_1. El primer posible evento, rotulado con el número 1 y visualizado con un símbolo de color verde, representa un símbolo fiable y acertado, ya que el demodulador lo interpreta como símbolo S_1. El segundo posible evento es rotulado con el número 2 y visualizado con un símbolo de color amarillo, representa un símbolo no fiable. El último posible evento, rotulado con el número 3 y visualizado con un símbolo de color rojo, representa un símbolo fiable y errado, debido a que es demodulado como un símbolo S_2.

Figura 4.2 Diagrama de constelación para 4-QAM con tres posibles eventos al demodular un símbolo S_1.



La fiabilidad de un símbolo es determinada por el demodulador modificado en función de la variación del MER. Esta fiabilidad se da a conocer al decodificador RS con capacidad de borrado. La detección y corrección de los errores es únicamente responsabilidad del decodificador y no del demodulador modificado.

4.2 VALIDACIÓN DEL MODELO DEL DECODIFICADOR RS CON CAPACIDAD DE BORRADO

Para analizar el comportamiento del modelo del decodificador RS con capacidad de borrado, se desarrolló una GUI que permite codificar una palabra mensaje según los parámetros del código $RS(n,k)$, además permite ingresar los valores y posiciones de los errores y borraduras deseadas. En el apéndice A.2 se explica detalladamente el funcionamiento de la GUI del validador.

El validador del modelo del decodificador RS con capacidad de borrado utiliza los algoritmos conceptualizados en la sección 2.2.

En la tabla 4.6 se muestra una tabla de datos obtenidos en el validador del modelo de simulación. En ella se visualiza el resultado de los cálculos de decodificación paso a paso para un código $RS(n,k)$.

Tabla 4.6 Tabla de datos del algoritmo de decodificación del código RS(15,9).

$RS(15,9)$	
NUMERO DE ERRORES (v)	2
NUMERO DE BORRADURAS (f)	2
CAPADIDAD DE CORRECIÓN $(2t)$	6
MENSAJE	1 2 3 4 5 6 7 8 9
VECTOR ERROR	0 0 0 0 1 0 0 0 0 0 14 0 0 0 0
VECTOR DE BORRADURAS	0 0 1 0 0 0 0 0 0 0 0 0 0 1 0
PALABRA CODIFICADA	1 2 3 4 5 6 7 8 9 2 1 3 12 15 11
PALABRA RECIBIDA	1 2 3 4 1 6 7 8 9 2 14 3 12 15 11
PALABRA RECIBIDA SIN BORRADURAS	1 2 0 4 1 6 7 8 9 2 14 3 12 0 11
POLINOMIO LOCALIZADOR DE BORRADURAS	13 13 1
SINDROME	2 8 11 8 9 2 1
SINDROME MODIFICADO	6 12 6 14 13 15 1
POLINOMIO LOCALIZADOR DE ERROR	11 7 5
POLINOMIO EVALUADOR	7 5 12 1 5
POLINOMIO LOCALIZADOR ERROR-BORRADURA	6 3 2 11 5
PATRON DE ERROR-BORRADURA	0 0 3 0 4 0 0 0 0 0 15 0 0 15 0
PALABRA DECODIFICADA	1 2 3 4 5 6 7 8 9 2 1 3 12 15 11
SER	0
TIEMPO DE DECODIFICACION (Segundos)	0,05339

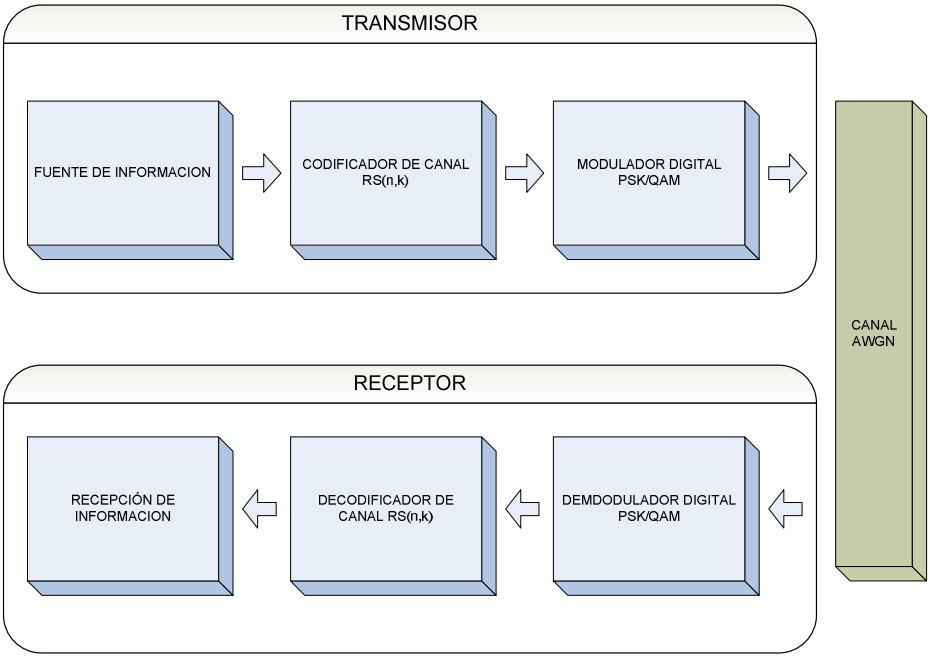
La palabra codificada se genera con funciones nativas de MATLAB® según la ecuación 1.19. La palabra recibida corresponde a la construcción del bloque de la ecuación 2.1. La palabra recibida sin borraduras está conformada por la palabra recibida y evaluando en cero las posiciones que se marcaron como borradura. El polinomio localizador de borraduras se construye con el arreglo que marca las posiciones de borradura utilizando la ecuación 2.28. El síndrome y síndrome modificado son el subproducto de las ecuaciones 2.3 y 2.27 respectivamente. El polinomio localizador de error y el polinomio evaluador son los resultados de la ejecución del algoritmo Euclidiano explicado en la sección 2.2.4. El polinomio localizador error-borradura se calcula como se muestra en la ecuación 2.30. Conociendo las posiciones y valores de todos los errores y borraduras se puede construir el patrón de error como lo indica la ecuación 2.33. Finalmente se decodifica la palabra realizando la adición entre la palabra recibida y el patrón de error.

La capacidad de corrección de un código $RS(n,k)$ está relacionada con la cantidad de errores y de borraduras, mediante la ecuación 2.25. Los resultados indican que siempre que se cumpla esta condición, el desempeño del decodificador es óptimo, decodificando cualquier combinación de errores y borraduras. El decodificador sólo tiene conocimiento de los símbolos no fiables que le marca el demodulador. A partir de ese momento, esos símbolos no fiables son denominados borraduras por el decodificador. Si únicamente la cantidad de borraduras sobrepasa la capacidad de corrección, el decodificador con capacidad de borrado se comporta como un decodificador sin capacidad de borrado. Si la combinación de errores y borraduras sobrepasa la capacidad de corrección del código $RS(n,k)$, el código se comporta de forma imprecisa, generando más errores a la palabra decodificada.

4.3 ANÁLISIS DE RESULTADOS DE LA SIMULACIÓN DEL MODELO DEL DECODIFICADOR RS CON CAPACIDAD DE BORRADO

La GUI del modelo de simulación del decodificador RS con capacidad de borrado se explica detalladamente en el apéndice C. Utilizando esta interfaz, se simula un sistema de comunicación en banda base que, genera ráfagas de información en bloques que son codificadas y moduladas según los parámetros de configuración. En la figura 4.3 se muestra el diagrama en bloques general del sistema de comunicación simulado.

Figura 4.3 Diagrama en bloques de un sistema de comunicación digital



Entender el comportamiento del demodulador modificado y del decodificador, validados con los modelos de simulación independientes, dan forma a las pruebas para el análisis del desempeño del decodificador RS con capacidad de borrado. El desempeño del decodificador $RS(n,k)$ se ve afectado por la cantidad de borraduras que son entregadas por el demodulador. El demodulador a su vez, entrega las posiciones de los símbolos no fiables en función de la variación de la MER. Esto significa que la variación de la MER es un criterio que afecta el desempeño del decodificador RS con capacidad de borrado. Si la variación de la MER es positiva, la cantidad de borraduras se aumenta; y si la MER es negativa la cantidad de borraduras disminuye.

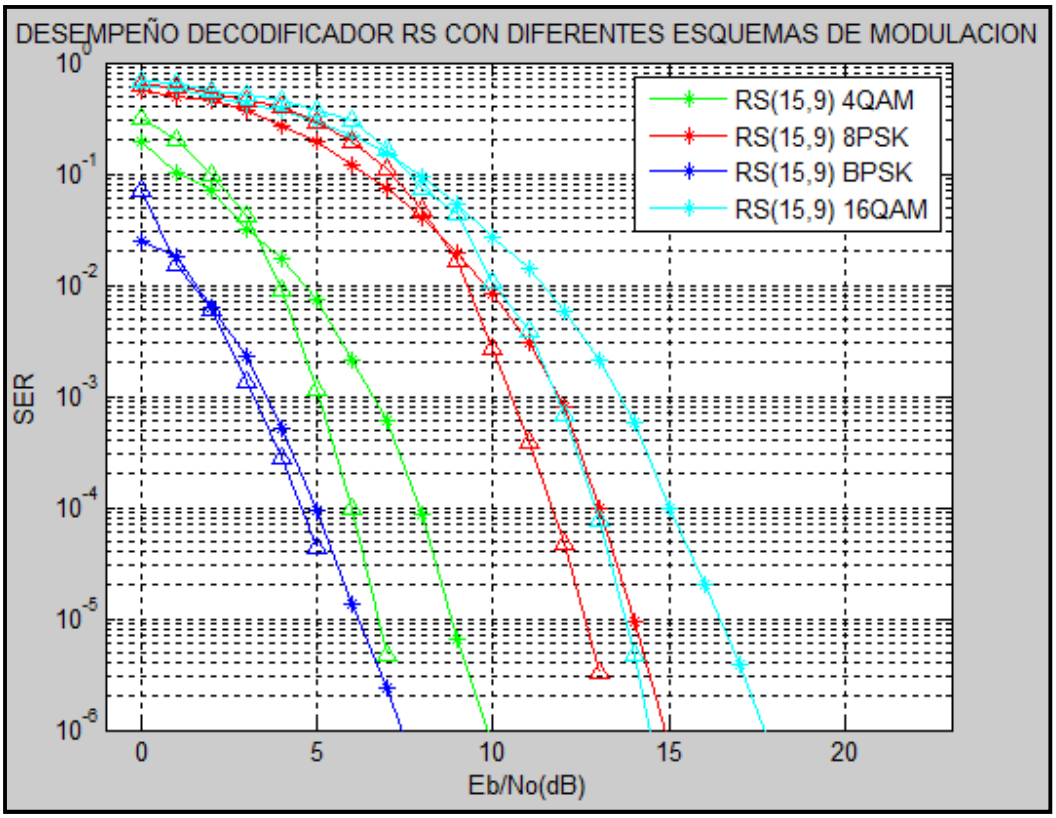
4.3.1 Análisis del desempeño del decodificador RS con capacidad de borrado para distintos esquemas de modulación

Este análisis se realizó con el objetivo de analizar el desempeño del decodificador RS con capacidad de borrado y sin capacidad de borrado para cuatro diferentes esquemas de modulación: BPSK, 4-QAM, 8-PSK, 16-QAM.

En la figura 4.4 se observan las curvas de desempeño del decodificador $RS(15,9)$ con capacidad de borrado y sin capacidad de borrado para diferentes esquemas de modulación. Las curvas de desempeño del decodificador sin capacidad de borrado se

dibujan con un triángulo, a diferencia de las curvas de desempeño del decodificador sin capacidad de borrado que se dibujan con un asterisco. Las curvas de desempeño se realizaron con un valor del parámetro “variación MER” de -3 dB. De esta figura se deduce que el esquema de modulación que necesita una menor relación Eb/No para alcanzar una SER de 10^{-6} es BPSK con capacidad de borrado.

Figura 4.4 Curvas de desempeño del código $RS(15,9)$ sin capacidad de borrado para BPSK, 4-QAM, 8-PSK y 16-QAM.



En la tabla 4.7, se tabulan los resultados de la Eb/No necesaria para alcanzar una SER=10^{-6} para los esquemas de modulación BPSK, 4-QAM, 8-PSK, y 16-QAM con capacidad de borrado y sin capacidad de borrado. Además, se colocan los valores de la ganancia de codificación que se obtienen al comparar el decodificador RS sin capacidad de borrado y el decodificador RS con capacidad de borrado.

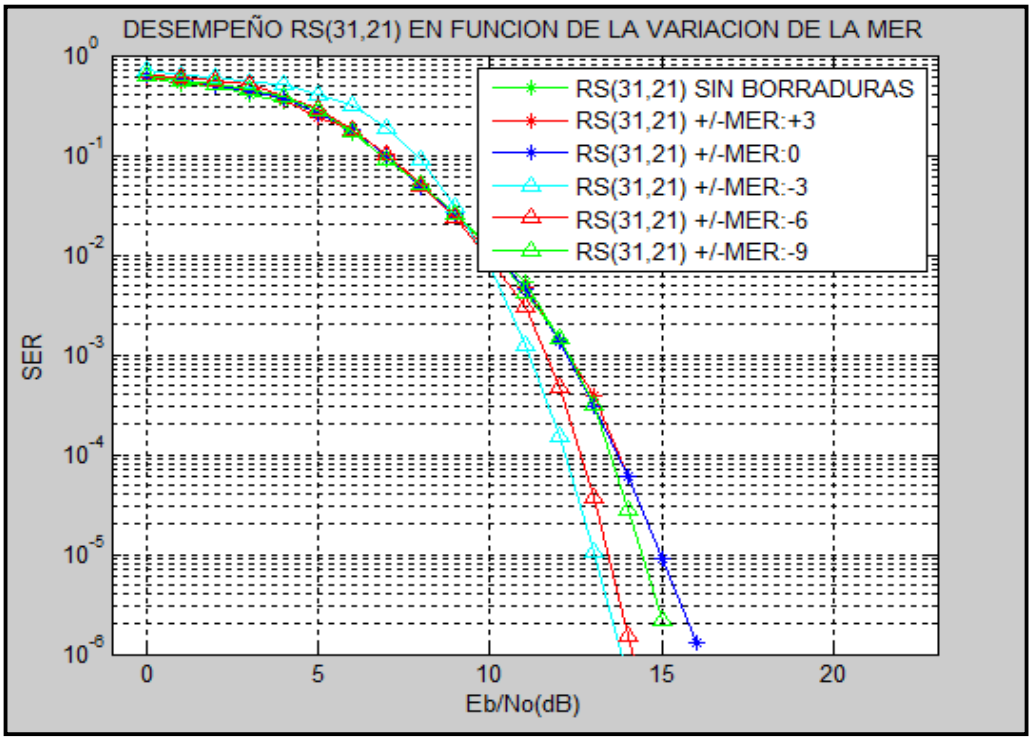
Tabla 4.7 Valores de Eb/No para SER=10^{-6} en BPSK, 8-PSK, 4QAM, 16-QAM.

ESQUEMA DE MODULACION	Eb/No a 10^{-6} en la SER CON BORRADURAS	Eb/No a 10^{-6} en la SER SIN BORRADURAS	GANANCIA DE CODIFICACIÓN[18]
BPSK	7,41 dB	7 dB	0,41 dB
4-QAM	9,86 dB	7,4 dB	2,46 dB
8-PSK	15 dB	13 dB	2dB
16-QAM	17,7 dB	13,8 dB	3,9 dB

4.3.2 Análisis del desempeño del decodificador RS con capacidad de borrado en función de la cantidad de borraduras

En la figura 4.5 se presenta las curvas de desempeño del decodificador RS con capacidad de borrado para un código $RS(31,21)$, con un esquema de modulación 16-QAM y distintos valores en la variación de la MER. En ella se observa que el desempeño del decodificador RS con capacidad de borrado se asemeja al desempeño del decodificador RS sin capacidad de borrado. Los datos de esta simulación son consignados en la tabla 4.8.

Figura 4.5 Curvas de desempeño del decodificador RS con capacidad de borrado en función del parámetro variación de la MER en 16-QAM.



[18] Ganancia de codificación de la decodificación con capacidad de borrado respecto a la decodificación sin capacidad de borrado para cada uno de los esquemas de modulación.

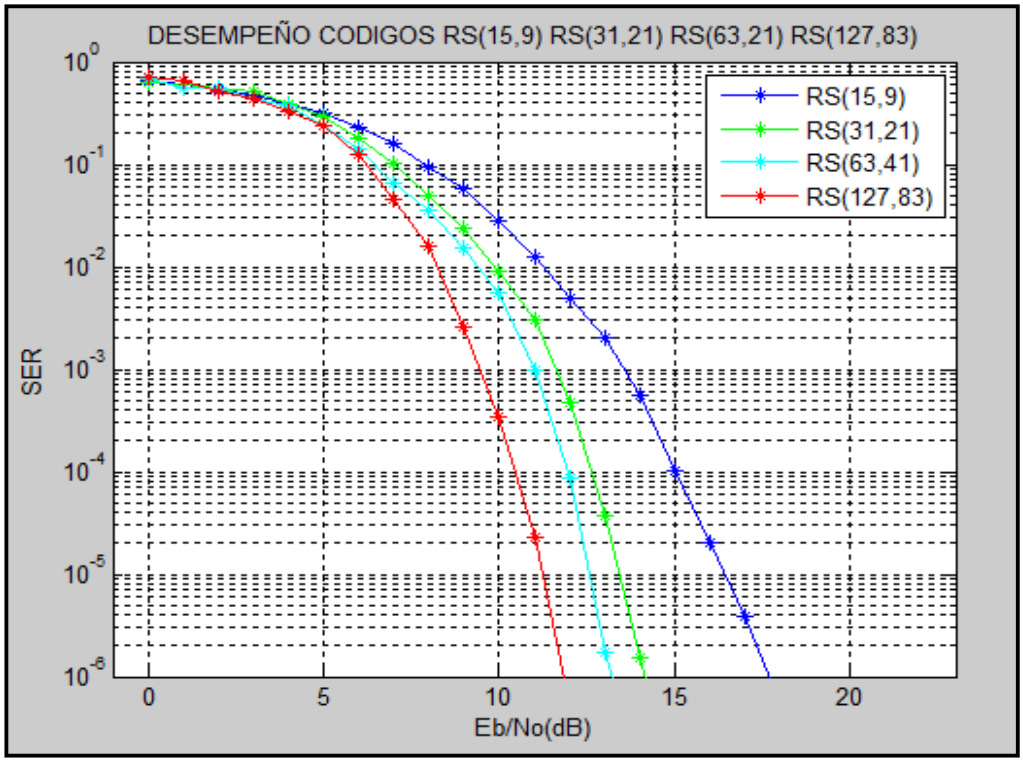
Tabla 4.8 Datos de desempeño del decodificador RS con capacidad de borrado en función de la variación de la MER para 16-QAM

VARIACION DE MER (dB)	Eb/No para alcanzar 10^{-6} en SER (dB)	GANANCIA DE CODIFICACION (dB)
+3	16.1	0,02
0	16.08	0,04
-3	13,79	2,33
-6	13.80	2,32
-9	15.2	0,92

De la tabla 4.8 se deduce que el valor óptimo de variación de la MER, el cual produce una ganancia de codificación mayor es -3 dB. A medida que va creciendo los valores negativos de 0 a -3 dB, mejora el desempeño del decodificador, si los valores negativos siguen creciendo, el desempeño disminuye. En los valores positivos de 0 a 3 dB, las curvas de desempeño se superponen a la curva de desempeño del decodificador RS sin capacidad de borrado,

4.3.3 Análisis del desempeño del decodificador RS con capacidad de borrado para distintas longitudes de código $RS(n,k)$.

Figura 4.6 Curvas de desempeño del decodificador RS con capacidad de borrado para distintas longitudes de palabra codificada.



Con el objetivo de observar el desempeño del decodificador RS con capacidad de borrado, se establece el esquema de modulación a 16-QAM, el parámetro variación de la MER en -3 dB, la tasa de codificación del código $RS(127,83)$ es 0.65, la del código $RS(63,41)$ es 0.65, la del código $RS(31,21)$ es 0.67 y la del código $RS(15,9)$ es 0.6. En la figura 4.6 se observa la propiedad de detección y corrección de códigos grandes. Como la longitud del código RS se calcula por el número de bits que contiene un símbolo, a mayor longitud de código RS, se mejora la SER contra la Eb/No, debido a que un código RS de mayor longitud, tiene mayor capacidad de información en sus símbolos. Esta es una de las principales ventajas de un código RS sobre los códigos binarios. El inconveniente de aumentar la longitud del código, es el aumento también en el ancho de banda que se requiere para enviar la información de la palabra codificada.

Tabla 4.9 Tiempos decodificación de los códigos

$RS(n,k)$	Tiempo de Decodificación (Segundos)
$RS(15,9)$	0,0022
$RS(31,21)$	0,455
$RS(63,41)$	0,573
$RS(127,83)$	13,48

El tiempo de decodificación en el modelo de simulación del decodificador RS con capacidad de borrado aumenta medida que aumenta la longitud de la palabra código. Esto se debe a que los polinomios generadores que se utilizan en la codificación y decodificación son de orden mayor. Como se observa en la tabla 4.9, los tiempos de decodificación no representarían una ventaja del decodificador RS que utiliza la capacidad de borrado en ambientes simulados; en el caso del decodificador $RS(127,83)$ el tiempo de decodificación fue $13,48$ segundos por palabra recibida de 127 símbolos.

4.3.4 Análisis del desempeño del decodificador RS que utiliza el algoritmo Euclidiano con capacidad de borrado variando la tasa de codificación

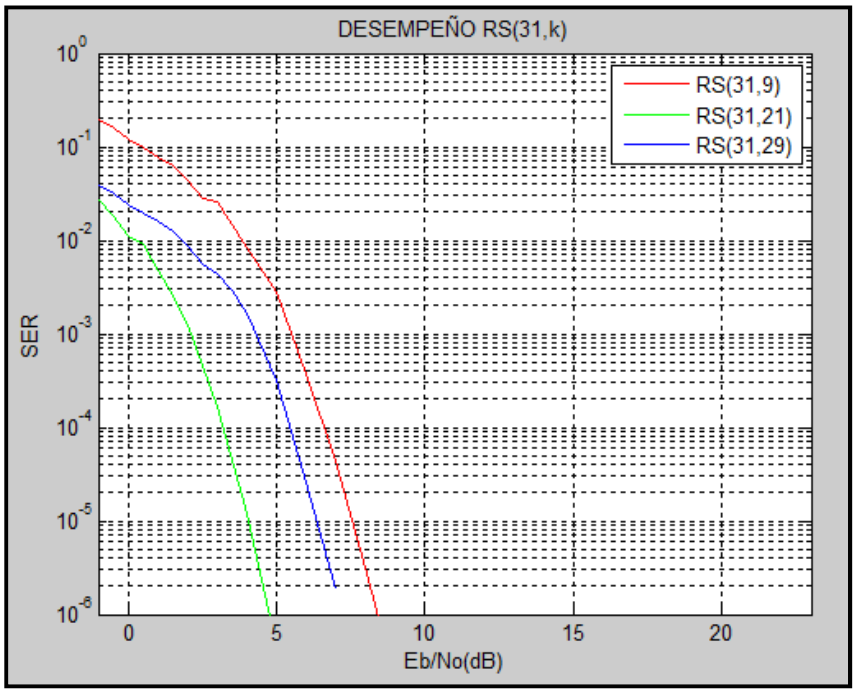
Para analizar el decodificador RS, se escoge una longitud de código $n = 31$, y esquema de modulación BPSK, variando la longitud de la palabra mensaje, y fijando el parámetro variación de la MER en -3 dB se obtienen distintos valores de redundancia y tasa de codificación como se muestra en la tabla 4.10.

Tabla 4.10 Tasas de codificación y redundancia de los códigos $RS(31,k)$.

$RS(n,k)$	Tasa de codificación k/n	Redundancia
$RS(31,9)$	0.29	22
$RS(31,21)$	0.67	10
$RS(31,29)$	0.93	2

En la figura 4.7 se observan las curvas de desempeño del decodificador $RS(31{,}21)$. Estas curvas de desempeño se construyen con la GUI del modelo de simulación desarrollado.

Figura 4.7 Curvas de desempeño del decodificador RS con capacidad de borrado para distintas tasas de codificación.

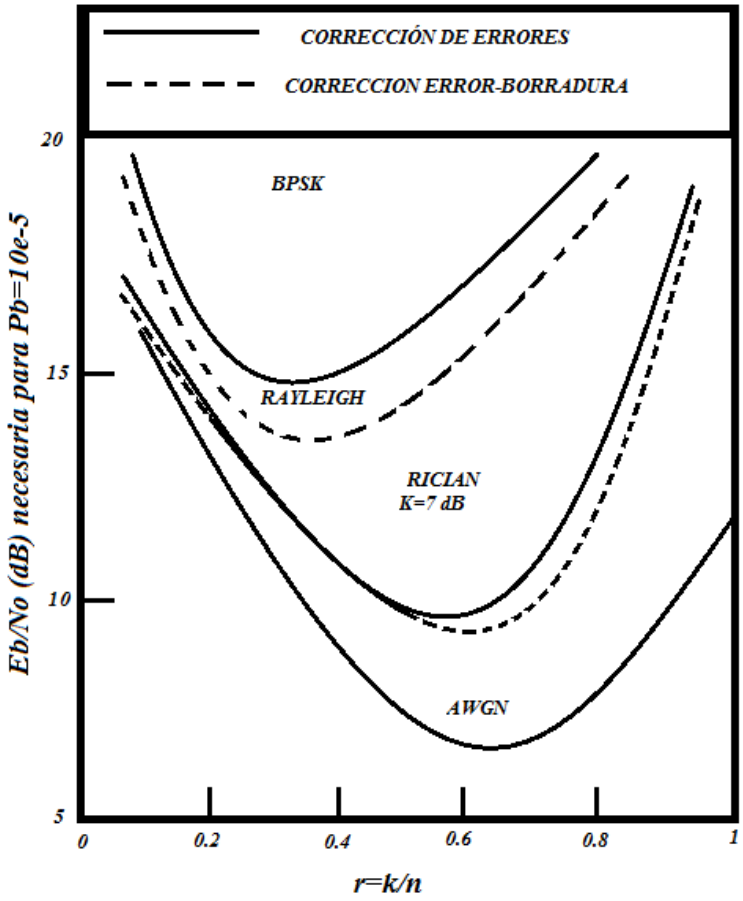


En la tabla 4.11 se observa la Eb/No que necesita un código para alcanzar un valor de SER=10^{-5}. La figura 4.8 confirma los datos y tabulados en la tabla 4.11.

Tabla 4.11 Datos de simulación para decodificador $RS(31,k)$ con capacidad de borrado.

$RS(n,k)$	Tasa de codificación k/n	Redundancia	Eb/No (dB) en SER=10^{-5}
$RS(31{,}9)$	0,29	22	7,5
$RS(31{,}21)$	0,67	10	4
$RS(31{,}29)$	0,93	2	6,2

Figura 4.8 Desempeño del código $RS(31,k)$ en función de la tasa de codificación en BPSK [20].



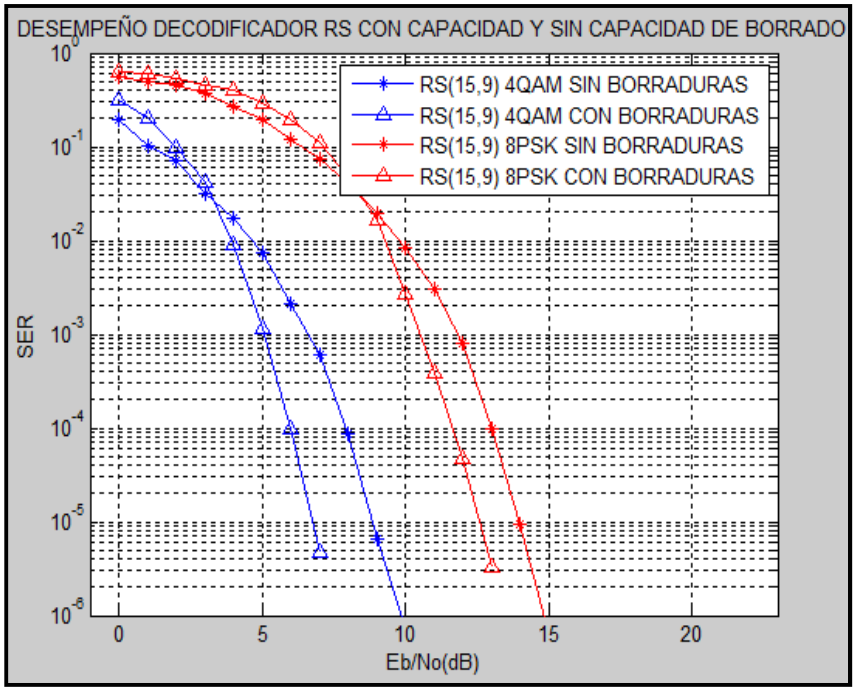
El interés de este trabajo es el canal AWGN, y de la figura 4.8 se observa que el desempeño del decodificador RS con capacidad de borrado es similar al decodificador RS sin capacidad de borrado en un canal AWGN, lo cual es corroborado en la tabla 4.7, donde se observa que la ganancia de codificación que se obtiene al utilizar el decodificador RS con capacidad de borrado en BPSK es pequeña.

4.3.5 Análisis del desempeño del decodificador RS con capacidad de borrado y sin capacidad de borrado en un canal AWGN

Con base en el análisis de los escenarios de simulación se observó que los comportamientos de las curvas de desempeño en función de los parámetros que se variaron para el análisis comparativo de los decodificadores con capacidad de borrado frente a los decodificadores sin capacidad de borrado, son similares. Es por esto que la capacidad de borrado en un decodificador RS en un sistema de comunicación modelado en el canal AWGN en el demodulador modificado no representan una mejora significativa. En la figura 4.9, se presentan las curvas de desempeño de los decodificadores $RS(15,9)$

con capacidad de borrado y sin capacidad de borrado en los esquemas de modulación 4-QAM y 8-PSK y con un valor del parámetro “variación MER” de -3 dB.

Figura 4.9 Comparación del desempeño del decodificador $RS(15,9)$ con capacidad de borrado y sin capacidad de borrado



5. CONCLUSIONES TRABAJOS FUTUROS Y RECOMENDACIÓNES

5.1 CONCLUSIONES

- La adaptación de la metodología de simulación de equipos de telecomunicaciones [1] es de gran ayuda en la construcción y desarrollo de un modelo de simulación de un sistema de comunicaciones digitales como el que se diseñó.

- El algoritmo Euclidiano utilizado como paso de decodificación en los códigos $RS(n,k)$ para la detección y corrección de errores, es sencillo de implementar para códigos de longitud de palabra extensa.

- El comportamiento del modelo de simulación del decodificador RS que utiliza el algoritmo Euclidiano con capacidad de borrado, se asemeja al modelo teórico estudiado. Se observa la decodificación RS sin capacidad de borrado, como un caso especial de la decodificación RS con capacidad de borrado.

- El criterio escogido para la marcación de las borraduras, da confiabilidad en el funcionamiento en los esquemas de modulación en cuadratura. La comparación de la MER_k de símbolo y la MER del demodulador, generan las regiones de fiabilidad y no fiabilidad en la constelación, útiles para la marcación de las borraduras. Así se logra deducir que la variación de la MER es un criterio que afecta el desempeño del decodificador RS con capacidad de borrado.

- El demodulador modificado diseñado cumple con la funcionalidad propuesta para incluir la propiedad de marcación de borraduras La cantidad de posiciones de borradura está en función del valor SNR y del valor del parámetro “variación MER”.

- El decodificador RS con capacidad de borrado presenta ganancia de codificación respecto al decodificador RS sin capacidad de borrado, en función del valor del parámetro “variación MER”.

- El comportamiento de las curvas de desempeño del decodificador RS con capacidad de borrado es similar al comportamiento de las curvas de desempeño del decodificador RS sin capacidad de borrado cuando se varían los esquemas de modulación y los parámetros del código, tales como: longitud de la palabra código, redundancia.

5.2 TRABAJOS FUTUROS

- Análisis del desempeño a nivel físico del decodificador RS que utiliza algoritmo Euclidiano con capacidad de borrado, utilizando modelos de canal Rayleigh y Rician, comparándolos con el modelo de canal AWGN mostrados en el presente trabajo de grado.

- Análisis del desempeño a nivel físico del decodificador RS que utiliza el algoritmo Euclidiano con capacidad de borrado, modificando el criterio de marcación de posición de borraduras, y compararlo con el presente trabajo de grado.

- Implementación del decodificador RS que utiliza el algoritmo Euclidiano con capacidad de borrado a nivel hardware concatenado con códigos convolucionales.

5.3 RECOMENDACIONES

- El desarrollo completo de los modelos de simulación utilizando la Metodología de Simulación de Equipos de Telecomunicaciones; para la adecuada construcción de los modelos.

- El estudio e implementación de sistemas en tiempo real, compilados en lenguaje de bajo nivel, con el objetivo de disminuir los tiempos de simulación de los sistemas de comunicación digital que utilicen códigos RS con capacidad de borrado.

BIBLIOGRAFÍA

[1] C. L. Muñoz, J. R. Muñoz, "Metodología para la simulación de equipos de equipos de telecomunicaciones", Universidad del Cauca, 1995

[2] S Haykin, “Sistemas de Comunicaciones”,Editorial Limusa-Wiley, 2002.

[3] S. Wicker, V. Bhargava, V. British, "An introduction to Reed-Solomon Codes".

[4] S. Wicker. “Error Control Systems for Digital Communications and Storage”. Prentice hall. 1995.

[5] P. Dubreil, M. Dubreil “Lecciones de Algebra Moderna”, Editorial Reverté S.A, 2da Edición 1975

[6] W. A. Geisel, “Tutorial on Reed-Solomon Error Correction Coding,” National Aeronautics and Space Administration, 1990.

[7] Create Galois Field Array, Matlab, [en linea]. Disponible: http://www.mathworks.com/help/comm/ref/gf.html, [consultado Febrero 24, 2013].

[8] J. Silvestre, “Reed Solomon Codec”, Elektrobit, January 2001.

[9] B. Sklar. “Digital Communications: Fundamentals and Applications”. Prentice Hall. 2da ed. 2001.

[10] T. K. Moon, “Error CorretionCoding, Mathematical Methods and Algorithms”, Wiley & Sons. 2005.

[11] S. Wicker. “Error Control Systems for Digital Communications and Storage”. Prentice hall. 1995.

[12] R. H. Zaragoza, " The Art of Error Correcting Coding", Sony Computer Science Laboratories Inc JAP, Jhon wiley and sons.

[13] R. E. Castillo, O. F. Muñoz, “Análisis y Evaluación del Desempeño del Decodificador Reed-Solomon con el Algoritmo Euclidiano”. Universidad del Cauca. Trabajo de grado en desarrollo. Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca.

[14] S. lin, D. Costello, "Error Control Coding", 2da edición, 2004.

[15] ETSI, European Telecomunications Standards Institute, *Digital Video Broadcasting y Measurement guidelines for DVB systems,* 1997. ETR 290.

[16] "Modulation Error Ratio (MER) and Error Vector Magnitude (EVM)", National Instruments [en linea] http://www.ni.com/white-paper/3652/en, [consultado 6 de enero 2013].

[17] C. Perez, J. Sainz, “Fundamentos de Telvision Analogica y Digital, Universidad de Cantabria. 2003.

[18] E. Garcia "Frequently Asked Questions About Noise". High Frequency Electronics. February 2008 summit technical Media, LLC. [en linea] http://www.highfrequencyelectronics.com/Archives/Feb08/HFE0208_Noisewavepdf . [consultado diciembre, 2012].

[19] R. Schmogrow, B. Nebendahl, M. Winter, A. Josten, D. Hillerkuss, S. Koenig, J. Meyer, M.Dreschmann, M. Huebner, C. Koos, J. Becker, W. Freude, and J. Leuthold, "Error Vector Magnitude as a Performance Measure for Advanced Modulation Formats-autores," *IEEE Photonics Technology Letters*, vol. 24, no.1, January 1, 2012.

[20] B. Sklar. “Reed-Solomon Codes”. Publicación. [En Línea]. Disponible: http://ptgmedia.pearsoncmg.com/images/art_sklar7_reed-solomon/elementLinks/art_sklar7_reed-solomon.pdf. [Consultado Enero 10, 2013].