Universidad de Cantabria

Facultad
de
Ciencias

CÓDIGOS CÍCLICOS. ANÁLISIS Y APLICACIONES DE CÓDIGOS BCH.

(Cyclic codes. Analysis and applications of BCH codes.)

Trabajo de Fin de Grado
para acceder al

GRADO EN MATEMÁTICAS

Autor: Lucía Iruzubieta Pinillos.

Director: Mónica Blanco Gómez.

Junio - 2025

AGRADECIMIENTOS

Évariste Galois dijo “no tengo tiempo para perder en la búsqueda de la perfección”, y ahora yo lo transcribo. Porque la vida continúa mientras hacemos matemáticas y cuándo no las hacemos. Cuándo hago matemáticas, es mi familia quién está ahí para remar en lo demás, y qué es la perfección, sino que todo funcione correctamente, que un engranaje siga girando o qué nadie se dé cuenta de que lo sigue haciendo. Gracias por hacer que todo funcione fuera de mi cabeza, y por recordarme innumerables veces que no tengo tiempo para perder en la búsqueda de la perfección. Gracias por enseñarme a ser feliz.

Richard Hamming dijo “la matemática es la herramienta fundamental para la comprensión del mundo”. Gracias Mónica por tenderme la mano para enfrentarme al último paso de la carrera, por acompañarme en este año, por la paciencia y las herramientas. Gracias porque sin saberlo, has acabado con algunos de mis miedos como joven matemática, que muchas veces me impedían dar un paso más.

Y por último, como dijo Arquímedes, “dame un punto de apoyo y moveré el mundo”. Está claro que el mundo no lo hemos movido (aún) pero Santander no sé si puede decir lo mismo. Gracias a todos y cada uno de vosotros, por ser mi familia, mis amigos, a veces mis padres y a veces mis hijos, a veces profesores, otras alumnos ; por discutir, por pedir perdón y hacerme pedirlo, por dar las gracias y por darme un abrazo cuando lo he necesitado. Jesús, Mirella, Garci, Pablo, Nuria, Asier, Darío, María, Lucía, Íñigo, Unai, Ámbar... aquí dejo vuestros nombres, como firma también de este trabajo.

Siempre he oído que las matemáticas se encuentran en cada rincón, pero allá dónde no han conseguido llegar, he encontrado el corazón de cada uno de vosotros.

RESUMEN

Este Trabajo Final de Grado se centra en el estudio de los códigos cíclicos, abordando en primer lugar su estructura y propiedades algebraicas fundamentales.

Se presenta una introducción a los códigos de Hamming, con una transición hacia los códigos Bose-Chaudhuri-Hocquenghem (BCH). Esta familia de códigos cíclicos destaca especialmente por su capacidad para detectar y corregir múltiples errores, además de por su robusta estructura algebraica.

El análisis profundiza en las propiedades algebraicas de los códigos BCH, considerando tanto su construcción como sus características y el proceso de decodificación. Se exploran los algoritmos asociados a la corrección de errores.

Finalmente, se presentan algunas aplicaciones clave de códigos BCH.

ABSTRACT

This final degree project focuses on the study of cyclic codes, first addressing their structure and fundamental algebraic properties.

An introduction to Hamming codes is presented, followed by a transition to Bose-Chaudhuri-Hocquenghem (BCH) codes. This family of cyclic codes is particularly notable for its ability to detect and correct multiple errors, as well as for its robust algebraic structure.

The analysis delves into the algebraic properties of the BCH codes, considering both their construction and characteristics, as well as the decoding process. Algorithms associated with error correction are explored.

Finally, some key applications of BCH codes are presented.

Índice

Capítulo 1

Introducción

1.1. Teoría de Códigos

La Teoría de Códigos es un área de las matemáticas cuyo principal objetivo es transmitir datos. En este esquema aparecen emisor, canal y receptor, los elementos básicos de la comunicación; además, aparecen dos procesos en los cuales se centra este área: codificar y decodificar. A grandes rasgos, codificar consiste en transformar la información para su transmisión a través del canal. La decodificación actúa de manera inversa y, a través de la señal recibida, trata de obtener la información original.

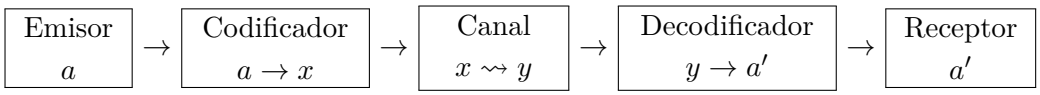

Figura 1.1: Esquema general sobre la transmisión de información

Como se puede intuir, el desarrollo de la Teoría de Códigos tuvo su auge en paralelo al progreso de la comunicación a mitad del siglo XX. A continuación, se presenta el contexto histórico en el que se desarrolla este trabajo. Conocer el origen de la teoría, las necesidades tecnológicas que impulsaron su aparición y su evolución permite comprender la motivación actual por estudiar códigos como los BCH. Se ha obtenido de [7], [1], [11].

1.1.1. Contexto histórico

Aunque el uso de sistemas de codificación se remonta a la Antigüedad, con ejemplos como los jeroglíficos egipcios (alrededor del 3300 a. C.) o el código Morse (creado en 1844 por Samuel Morse), la Teoría de Códigos como disciplina formal nace a finales de la década de 1940. Fue impulsada por necesidades tecnológicas y por los avances en la matemática aplicada. En este proceso confluyen tres ejes clave: la lógica binaria, la teoría de la probabilidad y la transmisión de señales. Por un lado, la lógica binaria, basada en los principios de la lógica aristotélica, reduce los sistemas a dos estados: 1 y 0, lo cual resulta esencial en el tratamiento algorítmico de la información. Por otro, la estadística permite modelar la información como sucesos aleatorios y medir su frecuencia o imprevisibilidad. Finalmente, la ingeniería de telecomunicaciones se enfrenta al reto de transmitir información de forma rápida, fiable y con el menor ruido posible. Con ruido nos referimos a alteraciones producidas en el canal que pueden modificar la información transmitida.

En este contexto, Claude Shannon, junto a Warren Weaver, publica en 1949 la teoría *A Mathematical Theory of Communication*, sentando las bases teóricas de la comunicación

digital. Shannon introduce el concepto de entropía de la información, una función que mide la incertidumbre asociada a una fuente de información aleatoria. Shannon demostró la existencia (aunque no la construcción explícita) de códigos capaces de alcanzar esta eficiencia óptima. Sus resultados, de carácter probabilístico, impulsaron el desarrollo de códigos eficientes. Fue David A. Huffman quien aportó el código de Huffman (1952), el primer ejemplo de esta Teoría de Shannon. Estos avances demostraron que era posible transmitir información de forma fiable en presencia de ruido si se aplicaban los códigos adecuados. En este sentido, destacan los trabajos de Richard Hamming y Marcel Golay, quienes diseñaron un tipo de códigos con estructuras algebraicas capaces de detectar cuando un mensaje transmitido no era correcto e incluso conseguir el mensaje original (corregir) sin necesidad de retransmisión. Desde entonces, la Teoría de Códigos ha evolucionado como una disciplina matemática con profundas raíces en el álgebra, la combinatoria, la teoría de la información y la probabilidad, con aplicaciones fundamentales en informática, telecomunicaciones, criptografía y compresión de datos.

Un ejemplo reciente y conocido es el código QR, creado en 1994 por la empresa japonesa Denso-Wave. Se trata de una matriz que codifica información binaria, permitiendo su rápida lectura y corrección de errores mediante códigos Reed–Solomon. Estas estructuras combinan conceptos algebraicos con técnicas de diseño robustas para entornos ruidosos.

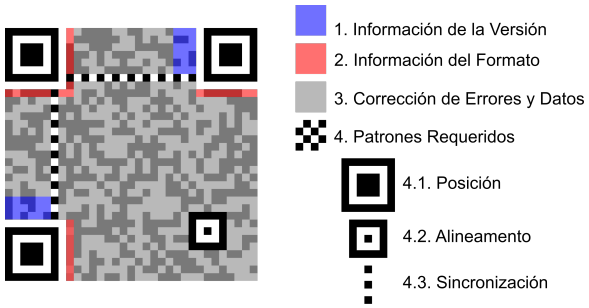

Figura 1.2: Esquema de información contenida en el código QR. Imagen de [2].

Este ejemplo ilustra la conexión entre teoría y aplicación práctica. Este trabajo se centra especialmente en los códigos BCH, pero antes conviene introducir dos tipos de códigos fundamentales en la historia de la Teoría de Códigos: los códigos lineales y los códigos cíclicos. Los primeros forman un subespacio vectorial; esto significa que cualquier combinación lineal de palabras código también pertenece al código, lo que facilita su análisis y su implementación. Dentro de esta categoría, los códigos cíclicos incorporan una propiedad adicional: si desplazamos los símbolos de un mensaje codificado en forma cíclica (por ejemplo, moviendo el último símbolo al principio), el resultado sigue siendo un mensaje válido. Esta característica permite representar los mensajes mediante polinomios, lo que abre la puerta al uso de herramientas algebraicas más potentes. Los códigos BCH son una subfamilia de códigos cíclicos descubiertos por Raj Chandra Bose (matemático indio-estadounidense) y Dwijendra Kumar Ray-Chaudhuri (físico indio) por un lado, y Alexis Hoquenghem (matemático francés) por otro.

Los trabajos relacionados con estos códigos comenzaron a desarrollarse entre 1959 y 1960. Se observó que algunos códigos binarios cíclicos contaban con propiedades más amplias que los códigos de Hamming, pudiendo corregir más de un error, gracias a su estructura algebraica. En 1960 ya se ideó el primer algoritmo de decodificación para códigos BCH binarios. Posteriormente, los resultados se generalizaron a alfabetos no binarios. Actualmente, los códigos BCH mantienen su importancia dada su fiabilidad.

1.2. Estructura y organización del trabajo

El presente Trabajo de Fin de Grado aborda el estudio de los códigos cíclicos y, de manera particular, los códigos Bose-Chaudhuri-Hocquenghem (BCH). La comprensión de estos códigos requiere una base sólida en conceptos algebraicos; por ello, este trabajo guía, desde la teoría más básica hasta las aplicaciones prácticas que demuestran el impacto tecnológico de estos códigos.

El recorrido del trabajo se inicia con el Capítulo 2, dedicado a los fundamentos algebraicos que aportan las herramientas necesarias para adentrarse en la Teoría de Códigos. Este capítulo cubrirá la teoría de cuerpos finitos, partiendo de definiciones como anillo, cuerpo, ideal o extensiones algebraicas, hasta conocer la construcción de cuerpos finitos. También se trabaja con la aritmética polinómica sobre estos cuerpos. Tendrá un gran peso en el capitulo las raíces de la unidad y los polinomios ciclotómicos. Esta base algebraica es crucial, ya que la construcción y las propiedades de los códigos cíclicos y BCH se asientan directamente sobre estas estructuras.

Posteriormente, el Capítulo 3 presentará el marco general del trabajo. Se introducirán los conceptos fundamentales de los códigos, como la distancia de Hamming y las capacidades de detección y de corrección de errores. Se presentan los códigos lineales y las herramientas para construirlos y representarlos, mediante conceptos como matriz generadora de un código o matriz de control. Aparece por primera vez un método común de decodificación a través del síndrome. El Capítulo 4 se adentrará en una clase específica de códigos lineales, los códigos cíclicos. Se estudia su relación con los ideales de anillos de polinomios, lo que simplifica enormemente el trabajo con estos códigos. Se analizarán sus generadores polinómicos y sus propiedades de codificación y decodificación. Como ejemplo paso intermedio hacia los códigos BCH, se hará una descripción específica de los códigos de Hamming.

En el Capítulo 5 el foco se dirige hacia los Códigos BCH. Esta sección está dedicada al estudio introductorio de esta importante familia de códigos. Se abordará un ejemplo concreto de su construcción para la corrección de dos errores a partir del código de Hamming $\mathcal{H}_3$ poniendo en uso los conocimientos sobre raíces en extensiones de cuerpos finitos. Se discutirán sus parámetros clave (longitud, dimensión, distancia mínima) y se presentarán los límites teóricos sobre su capacidad de corrección, a través de un nuevo parámetro: distancia de diseño. Se describirá un algoritmo simple de decodificación para códigos BCH: el Método de Peterson.

Finalmente, en el Capítulo 6 se ofrecerá una visión de la implementación y el impacto de estos códigos en el mundo real. Se presentarán dos casos de uso concretos en diversas áreas, como los sistemas de comunicación por satélite y los sistemas de telemedicina. Este capítulo busca consolidar la comprensión de la aplicabilidad práctica de los conceptos teóricos desarrollados.

Capítulo 2

Preliminares de Álgebra. Anillos y Cuerpos

Este capítulo preliminar recoge los conceptos algebraicos fundamentales para el desarrollo de la Teoría de Códigos BCH. Comenzamos con una revisión de los conceptos más básicos: anillo, ideal, cuerpo, entre otros. Seguida de un estudio del anillo R_n, esencial en códigos cíclicos. También se abordan las extensiones, cuerpos finitos, y polinomios ciclotómicos junto a raíces de la unidad, claves en la construcción de los códigos BCH. Los resultados de este capítulo son básicos, pueden encontrarse en la referencia [4], y en el Capítulo 4 de la referencia [6].

2.1. Anillos, cuerpos e ideales

Definición 2.1 (Anillo)**.** *Sean R un conjunto no vacío, $+, \cdot : R \times R \to R$ dos operaciones binarias e internas, decimos que $(R, +, \cdot)$ es un **anillo** si:*

- *$(R, +)$ es un grupo abeliano: la operación suma cumple la propiedad asociativa y conmutativa, tiene elemento neutro e inverso.*
- *$(R, \cdot)$ es un monoide conmutativo: la operación producto cumple la propiedad asociativa y tiene elemento neutro.*
- *Se cumple la propiedad distributiva: $\forall a, b, c \in R$ se tiene que $a \cdot (b + c) = a \cdot b + a \cdot c$ y $(a + b) \cdot c = a \cdot c + b \cdot c$.*

Observación 2.2. *En este texto, llamaremos anillo al anillo conmutativo y unitario. Denotaremos con 0_R al elemento neutro con la suma, y con 1_R al elemento neutro con el producto.*

Definición 2.3 (Dominio)**.** *Dado $(D, +, \cdot)$ un anillo, D es un **dominio** si el único divisor del cero en D es 0_D.*

Definición 2.4 (Unidades en el anillo)**.** *Los elementos $a \in R$ con inverso para el producto forman las **unidades en el anillo**. El conjunto de las unidades se denota por $U(R)$. $(U(R), \cdot)$ es un grupo abeliano.*

Definición 2.5 (Cuerpo)**.** *Sea $(K, +, \cdot)$ un anillo. K es **cuerpo** si $U(K) = K \setminus \{0_K\}$.*

Definición 2.6 (Cuerpo finito)**.** *Un cuerpo se dice **finito** si posee un número finito de elementos.*

Ejemplo 2.7. *El cuerpo de los racionales $\mathbb{Q}$ es un ejemplo de cuerpo infinito. $\mathbb{F}_2 = \{0, 1\}$ es un cuerpo finito.*

Definición 2.8 (Ideal)**.** *Dado un anillo $(R, +, \cdot)$, llamaremos **ideal** a todo subgrupo $(I, +)$ de $(R, +)$ tal que se verifica que: $\forall r \in R, \forall x \in I,\ r \cdot x \in I$.*

Proposición 2.9 (Test de caracterización de ideales)**.** *Sea $(R, +, \cdot)$ un anillo e I un subconjunto no vacío de R. Entonces:*

$$I \text{ ideal de } R \iff \forall x, y \in I \text{ y } \forall r \in R \text{ se tiene } \begin{cases} x - y \in I \\ r \cdot x \in I \end{cases}$$

Definición 2.10. *Sea $(R, +, \cdot)$ un anillo y S un subconjunto no vacío de R, llamamos **ideal generado por** S a la intersección de todos los ideales que contienen a S y lo denotamos por $(S) := \bigcap I_j$, siendo I_j ideal de R para todo j, y $S \subseteq I_j$ para todo j.*

Definición 2.11 (Ideal principal)**.** *Dado un anillo R y un elemento $a \in R$, se define un **ideal principal** como un ideal generado por un solo elemento a:*

$$(a) := \{r \cdot a \ :\ r \in R\} \subseteq R$$

Definición 2.12 (Dominio de Ideales Principales - DIP)**.** *Sea D un dominio, decimos que es un **DIP** cuando todo ideal de D es principal.*

2.2. Anillos de polinomios

Los anillos de polinomios, con su estructura algebraica y operaciones, facilitan la representación de los códigos lineales y su decodificación. Por el momento se definen sus propiedades algebraicas y en el próximo capítulo se verá su utilidad.

Definición 2.13 (Polinomio. Grado)**.** *Dado $(R, +, \cdot)$ un anillo, un **polinomio** con coeficientes en R y en la variable x es una expresión de la forma:*

$$p(x) = \sum_{j=0}^{n} a_j x^j = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n$$

*tal que $a_i \in R$ para todo $i = 0, ..., n$. Se denota con $R[x]$ al conjunto de todos los polinomios con coeficientes en R. El **grado del polinomio** será el mayor n tal que $a_n \neq 0$.*

Observación 2.14. *Una **función polinómica** es una función expresada mediante un polinomio, de modo que para $p \in R[x]$, $p : R \to R$, con $c \to a_0 + a_1 c + a_2 c^2 + \cdots + a_n c^n$.*

Definición 2.15 (Raíz de un polinomio)**.** *Sea $c \in R$ y $p(x)$ un polinomio tal que $p(c) = 0$, se dice que c es **raíz de p** (o una solución de la ecuación polinómica $p(x) = 0$).*

Proposición 2.16. *Dado $(R, +, \cdot)$ un anillo, $(R[x], +, \cdot)$ es un anillo cuyas operaciones están definidas de la siguiente manera. Sean $p(x), q(x) \in R[x]$ de grados n y m respectivamente:*

- *suma $+$:*

$$p(x) + q(x) = \sum_{k=0}^{max\{m,n\}} (p_k + q_k) x^k$$

- *producto* $\cdot$ *:*

$$p(x) \cdot q(x) = \sum_{k=0}^{n+m} (\sum_{i=0}^{k} p_i \cdot q_{k-i}) x^k$$

Las raíces y factorización de polinomios nos servirán para caracterizar y diseñar códigos con buenas propiedades de corrección de errores.

Definición 2.17 (Polinomio irreducible)**.** *Un polinomio es* ***irreducible*** *sobre un cuerpo K si no se puede expresar como el producto de dos polinomios en el cuerpo; es decir, dos polinomios con grado menor y con coeficientes en el cuerpo. Esto es lo mismo que decir que un polinomio no puede descomponerse en factores que no sean triviales (1 o el mismo).*

Teorema 2.18 (Identidad de Bézout)**.** *Sean $f, g \in R[x]$ dos polinomios no nulos y no unidades, y sea h su máximo común divisor. Entonces, existen $\alpha, \beta \in R[x]$ tales que*

$$h = \alpha f + \beta g$$

con $deg(\alpha) < deg(g)$, $deg(\beta) < deg(f)$.

2.2.1. Anillo R_n

K es un cuerpo cualquiera en esta sección. Este anillo nos va a permitir representar los códigos cíclicos como ideales.

Definición 2.19. *Se conoce al* ***anillo*** *R_n como el anillo que representa las clases de los residuos de $K[x]$ módulo $(x^n - 1)$:*

$$R_n \quad \cong \quad K[x] \Big/ (x^n - 1)$$

Este anillo contiene a todos los polinomios de grado menor que n y coeficientes en K.

Notación. *En lo siguiente se utiliza " $\equiv$ " para representar " $=$ (mód $x^n - 1$) ".*

Proposición 2.20. *El anillo R_n no es cuerpo para $n \geq 2$.*

Demostración. Véase que el elemento no nulo $x - 1 \in R_n$ no tiene inverso para el producto. Suponemos por reducción al absurdo que $x-1$ tiene inverso para el producto. Esto es, existe un elemento $g(x) \in K[x]$ tal que $(x-1) \cdot g(x) \equiv 1$. Sea $g(x) = g_0 + g_1 x + g_2 x^2 + ... + g_{n-1} x^{n-1}$, con $g_i \in K \ \forall i \in \{0, ..., n-1\}$. Desarrollamos el producto.

$$\begin{aligned} g(x)(x-1) &= g_0 x + g_1 x^2 + \cdots + g_{n-1} x^n - g_0 - g_1 x - g_2 x^2 - \cdots - g_{n-1} x^{n-1} \\ &\equiv g_{n-1} + g_0 x + g_1 x^2 + \cdots + g_{n-2} x^{n-1} - g_0 - g_1 x - \cdots - g_{n-1} x^{n-1} \\ &= (g_{n-1} - g_0) + (g_0 - g_1) x + \cdots + (g_{n-2} - g_{n-1}) x^{n-1} = 1 \end{aligned}$$

Por lo tanto, el sistema de ecuaciones queda como:

$$\begin{cases} g_{n-1} - g_0 = 1, \\ g_0 - g_1 = 0, \\ \vdots \\ g_{n-2} - g_{n-1} = 0 \end{cases} \iff \begin{cases} g_{n-1} - g_0 = 1, \\ g_0 = g_1 = g_2 = \cdots = g_{n-1} \end{cases}$$

Encontramos un sistema sin solución, por lo que $x - 1$ no tiene inverso. □

R_n no es un cuerpo, pero contiene a los códigos cíclicos en los que vamos a trabajar.

Corolario 2.21. *Un polinomio $f(x)$ tiene inverso con el producto en R_n si y solo si $f(x)$ y $x^n - 1$ son coprimos en $K[x]$*

Demostración. Vamos a demostrar cada implicación:

- $\Longleftarrow$: Suponemos que $f(x)$ y $x^n - 1$ son coprimos en $K[x]$. Aplicando la Identidad de Bézout (Teorema 2.18), existen dos polinomios $g(x), h(x) \in K[x]$ que cumplen

  $$f(x)g(x) + (x^n - 1)h(x) = 1$$

  Ahora, teniendo en cuenta la equivalencia módulo $x^n - 1$

  $$1 = f(x)g(x) + (x^n - 1)h(x) \equiv f(x)g(x)$$

  Por lo tanto $f(x)$ tiene inverso en R_n para el producto, y es $g(x)$.

- $\Longrightarrow$ Suponemos que $f(x)$ tiene inverso para el producto en R_n, entonces existe $g(x) \in R_n$ tal que $f(x)g(x) \equiv 1$, esto significa que para algún $h(x) \in K[x]$

  $$f(x)g(x) = 1 + (x^n - 1)h(x)$$

  Entonces $f(x)$ y $x^n - 1$ deben ser coprimos, porque si compartieran algún divisor distinto de 1, no se podría dar la anterior igualdad.

□

Proposición 2.22. *R_n es DIP*

Se puede demostrar en el área de Álgebra, pero su prueba para K finito se simplifica notablemente haciendo uso de la Teoría de Códigos (Corolario 4.11).

2.3. Extensiones algebraicas

Definición 2.23 (Extensión. Elemento algebraico)**.** *Sean K y F cuerpos, F es una **extensión** de K si K es un subcuerpo de F. Un elemento $\beta \in F$ es **algebraico** sobre K si es raíz de un polinomio no nulo con coeficientes en K.*

Definición 2.24 (Polinomio mínimo)**.** *Dado un cuerpo K y una extensión de cuerpos F, sea $\beta \in F$ un elemento algebraico sobre K. Llamaremos **polinomio mínimo** de β sobre K a $M(x)$ si es el único polinomio mónico, irreducible, de menor grado, con coeficientes en K, y que cumple $M(\beta) = 0$.*

Definición 2.25 (Extensión simple. Elemento primitivo)**.** *Una extensión F de K es **simple** si existe un elemento $\beta \in F \setminus K$ tal que F es el cuerpo más pequeño que contiene al cuerpo K y al elemento β. Se denota como $F = K(\beta)$. Llamaremos a β elemento **primitivo** de la extensión.*

Proposición 2.26. *Sea F una extensión de K y $M(x)$ el polinomio mínimo de β sobre K entonces*

$$\begin{array}{rccc} \psi: & K(\beta) & \to & K[x] \Big/ (M(x)) \\ & \beta & \to & x \pmod{M(x)} \end{array}$$

define un isomorfismo de cuerpos.

Proposición 2.27. *Sea $F = K(\beta)$ una extensión simple con $M(x)$ polinomio mínimo de β sobre K de grado m, entonces $K(\beta)$ es un K-espacio vectorial de dimensión m y con base $\{1, \beta, \beta^2, ..., \beta^{m-1}\}$. Además $K(\beta) = \{k_0 + k_1\beta + ... + k_{m-1}\beta^{m-1}, k_i \in K\}$. A la dimensión m se le llama **grado de la extensión** de F sobre K.*

Las extensiones algebraicas permiten ampliar un cuerpo para construir estructuras más ricas. Se verá más claro en el siguiente ejemplo.

Ejemplo 2.28. *$\mathbb{C}$ es una extensión simple de $\mathbb{R}$. $i \in \mathbb{C}$ pero $i \notin \mathbb{R}$. i es raíz del polinomio $x^2 + 1 \in \mathbb{R}[x]$, que corresponde al polinomio mínimo de i elemento algebraico sobre $\mathbb{R}$. Si representamos $\mathbb{C}$ como $\mathbb{R}(i)$ entonces $\mathbb{R}(i) := \{a + bi : a, b \in \mathbb{R}\}$.*

2.4. Raíces de la unidad y polinomios ciclotómicos en K

Las raíces de la unidad y los polinomios ciclotómicos tienen una gran importancia en la teoría de códigos. Esta sección introduce conceptos útiles y herramientas en un cuerpo K genérico.

Definición 2.29 (Orden de un elemento)**.** *En un grupo, **el orden de un elemento** es el número mínimo de veces que debe operarse un elemento consigo mismo para obtener el elemento neutro del grupo. Denotándolo con la operación multiplicativa, sea a el elemento, tenemos:*

$$a^i := a \cdot \underbrace{\ldots}_{i \text{ veces}} \cdot a$$

*Llamando al **conjunto generado por** a como $\langle a \rangle := \{a^i : i \in \mathbb{N}\}$, y denotando el orden de a como $O(a)$, se tiene*

$$O(a) = \text{mín}\{i \in \mathbb{N} : a^i = 1\} = \#\langle a \rangle$$

Definición 2.30 (Grupo cíclico)**.** *Un grupo $(G, \cdot)$ es **cíclico** si se puede generar con un solo elemento. Es decir si existe $a \in G$ tal que $G = \langle a \rangle$.*

Proposición 2.31. *Si G es finito, es cíclico si y solo sí existe $a \in G$ tal que $O(a) = \#G$.*

Observación 2.32. *En todo grupo finito, el orden de cualquier elemento divide al cardinal del grupo.*

Definición 2.33 (Característica)**.** *Sea K un cuerpo. Dado un número natural $n \in \mathbb{N}$ definamos $n \cdot 1_K$ a la cantidad siguiente:*

- *Si $n = 0$, entonces $n \cdot 1_K = 0_K$*
- *Si $n = 1$, entonces $1 \cdot 1_K = 1_K$*
- *Para $n \geq 2$, se define recursivamente $n \cdot 1_K := (n-1) \cdot 1_K + 1_K$*

Si existe un número entero $n \in \mathbb{N}$, $n \neq 0$ tal que $n \cdot 1_k = 0$, se tendrá que el número natural:

$$p := \text{mín}\{n \in \mathbb{N}, n \neq 0, n \cdot 1_K = 0\} = \text{mín}\{n \in \mathbb{N} : n \neq 0, n \cdot x = 0, \forall x \in K\}$$

*es la **característica del cuerpo K**.*
*Si no existe $n \in \mathbb{N}$ bajo las condiciones anteriores, diremos que la **característica del cuerpo K** es 0. Denotamos la característica de K como $char(K)$.*

Observación 2.34. *Si $char(K) = 0$, entonces K es infinito. No ocurre al revés. Existen cuerpos infinitos con característica no nula. Por ejemplo $(\mathbb{Z}/2\mathbb{Z})[x]$ es infinito pero $char((\mathbb{Z}/2\mathbb{Z})[x]) = char(\mathbb{Z}/2\mathbb{Z}) = 2$.*

Definición 2.35 (Raíces n-ésimas de la unidad)**.** *Dado $n \in \mathbb{N}$ y K un cuerpo, se define el polinomio*

$$x^n - 1 \in K[x]$$

*como el polinomio mónico cuyas raíces son las **raíces n-ésimas de la unidad**.*

Observación 2.36. *Las raíces n-ésimas de la unidad no tienen porque pertenecer al cuerpo K. En concreto pertenecen a una extensión algebraica F de K.*

Definición 2.37 (Raíces n-ésimas primitivas)**.** *Entre las raíces n-ésimas de la unidad, una raíz es **primitiva** si tiene orden exactamente n.*

Observación 2.38. *No siempre existen raíces primitivas de la unidad en todo cuerpo. Por ejemplo, sea $K = \mathbb{F}_2$, $x^4 - 1$ no tiene raíces primitivas 4-ésimas en el cuerpo $\mathbb{F}_2$. En este caso, $x^4 - 1 = (x + 1)^4$ y su única raíz es 1, raíz cuádruple.*

Proposición 2.39. *Sea K un cuerpo, y $x^n - 1 \in K[x]$, si $char(K)$ no divide a n entonces todas las raíces son distintas y además existen raíces primitivas n-ésimas de la unidad.*

Notación. *Denotaremos a G_n como el conjunto de todas las raíces n-ésimas de la unidad y con $P_n \subseteq G_n$ al conjunto formado por las raíces n-ésimas primitivas de la unidad.*

Proposición 2.40. *Si $P_n \neq \emptyset$ entonces $(G_n, \cdot)$ es un grupo cíclico de orden n generado por cualquier elemento de P_n.*

Ejemplo 2.41. *Consideramos el cuerpo de los reales $\mathbb{R}$. Las n-ésimas raíces de la unidad serán*

$$G_n = \left\{ e^{\frac{2\pi i k}{n}} | k = 0, ..., n-1 \right\} \subseteq \mathbb{C}$$

Siendo $e^{2\pi i k} = 1$, con $k \in \mathbb{N}$. En concreto, para $n = 1$, se tiene que la única raíz de $x - 1$ es 1.
En otro caso, por ejemplo para $n = 4$, $x^4 - 1 \in \mathbb{R}[x]$ tiene 4 raíces que son $G_4 = \left\{ 1 = e^{2\pi i}, e^{\frac{\pi i}{2}}, e^{\pi i}, e^{\frac{3\pi i}{2}} \right\} \subseteq \mathbb{C}$, donde tan solo dos son raíces 4-ésimas primitivas de la unidad $P_4 = \left\{ e^{\frac{\pi i}{2}}, e^{\frac{3\pi i}{2}} \right\} \subseteq \mathbb{C}$. Se puede comprobar rápidamente que cada elemento de P_4 genera el grupo G_4.

Definición 2.42 (Polinomio ciclotómico)**.** *Consideramos el **n-ésimo polinomio ciclotómico** como:*

$$\Phi_n(x) = \prod_{\alpha \in P_n} (x - \alpha)$$

El polinomio cuyas raíces son exactamente los elementos de P_n.

Proposición 2.43. *Sea $n \in \mathbb{N}$ y K un cuerpo cuya característica no divida a n:*

$$x^n - 1 = \prod_{d|n} \Phi_d(x)$$

y $\Phi_d(x) \in K[x]$ $\forall d \in \mathbb{N}$ divisor de n.

Veamos algunos ejemplos de polinomios ciclotómicos.

Ejemplo 2.44. $n = 1 : x^1 - 1 = x - 1 = \Phi_1(x)$
$n = 2 : x^2 - 1 = \Phi_1(x)\Phi_2(x)$ *entonces* $\Phi_2(x) = \frac{x^2-1}{\Phi_1(x)} = x + 1$
$n = 3 : x^3 - 1 = \Phi_1(x)\Phi_3(x)$ *entonces* $\Phi_3(x) = \frac{x^3-1}{\Phi_1(x)} = x^2 + x + 1$
$n = 6 : x^6 - 1 = \Phi_1(x)\Phi_2(x)\Phi_3(x)\Phi_6(x)$ *entonces* $\Phi_6(x) = \frac{x^6-1}{\Phi_1(x)\Phi_2(x)\Phi_3(x)} = x^2 - x + 1$

Observación 2.45. *Para un p primo, el polinomio* $x^p - 1$ *cumple que:*

$$x^p - 1 = \Phi_1(x)\Phi_p(x) \text{ siendo } \Phi_p(x) = \frac{x^p - 1}{\Phi_1(x)} = \frac{x^p - 1}{x - 1} = x^{p-1} + x^{p-2} + ... + x + 1$$

2.5. Cuerpos finitos. Grupos cíclicos. Elementos primitivos

Los cuerpos finitos y sus propiedades serán útiles para trabajar con códigos finitos.

Teorema 2.46.

- *Sea K un cuerpo finito, existe un primo p tal que* $\text{char}(K) = p$. *Además existe un* $m \in \mathbb{N}$ *tal que* $|K| = p^m$.
- *Sea* $q = p^m$, *con* $m \in \mathbb{N}$ *y p primo, existe un único cuerpo finito (salvo isomorfismos) con exactamente q elementos, denotado por* $\mathbb{F}_q$. *En particular, si q es primo,* $\mathbb{F}_q$ *es isomorfo a* $\mathbb{Z}/q\mathbb{Z}$.

A partir de aquí, $\mathbb{F}_q$ representa a un cuerpo finito de q elementos, siendo $q = p^m$ una potencia de primo. El caso de los números primos se encuentra dentro de este, siendo q primo cuando $m = 1$. Cuando aparezca p siempre será referido a un número primo. Vistos estos resultados, nos podemos preguntar que forma tendrán los cuerpos finitos $\mathbb{F}_q$ con q no primo.

Observación 2.47. *El conjunto* $\mathbb{Z}/p^m\mathbb{Z}$ *no es un cuerpo para* $m > 1$. *En efecto, existen elementos* $a = p$ *y* $b = p^{m-1}$ *en* $\mathbb{Z}/p^m\mathbb{Z}$ *tales que* $a, b \neq 0$, *pero* $a \cdot b \equiv 0 \pmod{p^m}$.

Parece que no se pueden describir con facilidad, y que debemos construirlos. Haremos uso de la Proposición 2.26.

Definición 2.48 (Construcción de $\mathbb{F}_{p^m}$)**.** *El cuerpo* $\mathbb{F}_{p^m}$ *se construye como el anillo cociente entre el anillo de polinomios* $\mathbb{F}_p[x]$ *sobre el ideal generado por un polinomio irreducible* $f(x) \in \mathbb{F}_p[x]$ *de grado m.*

$$\mathbb{F}_{p^m} \cong \mathbb{F}_p[x] \Big/ (f(x))$$

Observación 2.49. *Teniendo en cuenta la Proposición 2.26 y la Proposición 2.27,* $\mathbb{F}_{p^m} \cong \mathbb{F}_p[x] \Big/ (f(x)) \cong \mathbb{F}_p(\beta)$, *siendo* $\beta \in \mathbb{F}_{p^m}$ *un elemento algebraico sobre* $\mathbb{F}_p$ *con polinomio mínimo f.*

Ejemplo 2.50.

- *Vamos a construir el cuerpo* $\mathbb{F}_8$.

  Siendo $\beta \in \mathbb{F}_8$ *un elemento algebraico sobre* $\mathbb{F}_2$ *y* $f(x) = x^3 + x + 1 \in \mathbb{F}_2[x]$ *su polinomio mínimo tal que* $\beta^3 = \beta + 1$, *los elementos del cuerpo son:*

  $$\mathbb{F}_8 = \{0, 1, \beta, \beta^2, \beta + 1, \beta^2 + \beta, \beta^2 + \beta + 1, \beta^2 + 1\}$$

  Se puede representar este cuerpo de la misma forma como $\mathbb{F}_2[x] \Big/ (f(x))$. *Todos los elementos distintos de cero en el cuerpo tienen inverso para el producto:*

- $x \cdot (x^2+1) \equiv 1 \pmod{x^3+x+1}$
- $x^2 \cdot (x^2+x+1) \equiv 1 \pmod{x^3+x+1}$
- $(1+x) \cdot (x^2+x) \equiv 1 \pmod{x^3+x+1}$

Las unidades son $U(\mathbb{F}_8) = \{1, \beta, \beta^2, \beta+1, \beta^2+\beta, \beta^2+\beta+1, \beta^2+1\} = \mathbb{F}_8 \setminus \{0\}$*, es decir, efectivamente es cuerpo. (Véase la Tabla 5.1).*

- *El conjunto* $\mathbb{Z}/8\mathbb{Z}$ *no es un cuerpo. Sus elementos son:*

$$\mathbb{Z}/8\mathbb{Z} = \{0, 1, 2, 3, 4, 5, 6, 7\}$$

pero no todos son unidades. Por ejemplo no existe ningún elemento $b \in \mathbb{Z}/8\mathbb{Z}$ *tal que* $2 \cdot b = 1$*, esto es porque* $mcd(2, 8) = 2 \neq 1$*, por ello* $2 \notin U(\mathbb{Z}/8\mathbb{Z})$*.*

2.6. Raíces de la unidad y polinomios ciclotómicos en K finito

Vayamos al caso de cuerpos finitos.

Teorema 2.51 (Corolario 3, p.96 de [6]. Teorema de Fermat)**.** *Para todo elemento* β *en un cuerpo* $\mathbb{F}_{p^m}$*, siendo* p *un número primo y* m *un natural, se cumple que:*

$$\beta^{p^m} = \beta$$

o lo que es lo mismo, β *es una raíz de:*

$$x^{p^m} - x$$

El orden de β *es divisor de* $p^m - 1$*.*

Observación 2.52 (Elemento primitivo)**.** *Bajo las condiciones anteriores, si el orden de* β *es exactamente* $p^m - 1$*, diremos que* β *es un* ***elemento primitivo*** *de* $\mathbb{F}_{p^m}$*. Sea* K *un cuerpo finito, todo elemento generador del grupo multiplicativo* $U(K)$ *es* ***elemento primitivo*** *de* K*.*

Observación 2.53. *Dado* $\beta \in \mathbb{F}_{p^m} \setminus \{0\}$*,* β *es raíz de* $x^n - 1$ *con* $n = p^m - 1$*. En particular* β *es raíz* n*-ésima de la unidad y, si es primitivo de* $\mathbb{F}_{p^m}$*, es raíz primitiva* n*-ésima.*

Ejemplo 2.54. *(Véase Ejemplo 2.50, Definiciones 2.29, 2.30 y la Proposición 2.31).* $(\mathbb{F}_8 \setminus \{0\}, \cdot)$ *es un grupo cíclico. Sea* $\beta \in \mathbb{F}_8$ *un elemento primitivo del cuerpo, es de orden* $7 = 2^3 - 1$*. Como* 7 *es primo, todos* $\beta \in \mathbb{F}_8 \setminus \{0, 1\}$ *tienen orden 7 y pueden generar* $\mathbb{F}_8 \setminus \{0\}$*, es decir son primitivos.*

En el próximo ejemplo se verá una situación diferente, donde no todos los elementos son primitivos.

Ejemplo 2.55 (Cuerpo $\mathbb{F}_{16} = \mathbb{F}_{2^4}$)**.** *En el cuerpo* $\mathbb{F}_{16}$ *existen elementos de orden* $2^4 - 1 = 15$ *que son primitivos, pero también hay elementos de orden* 3 *y* 5 *divisores de* 15*, que no son primitivos.*

$$\mathbb{F}_{16} \quad \cong \quad \mathbb{F}_2[x] \Big/ (x^4 + x + 1)$$

Los elementos del cuerpo son $\mathbb{F}_{16} = \{0, 1, \beta, \beta^2, \beta^3, 1+\beta, \beta+\beta^2, \beta^2+\beta^3, 1+\beta+\beta^3, 1+\beta^2, \beta+\beta^3, 1+\beta+\beta^2, \beta+\beta^2+\beta^3, 1+\beta+\beta^2+\beta^3, 1+\beta^2+\beta^3, 1+\beta^3\}$*. Tenemos* β *como un elemento primitivo,* $O(\beta) = 15$*. Otro elemento primitivo sería* β^2*, pero existen elementos que no lo*

son porque no generan todo el cuerpo. Por ejemplo β^3 *tiene orden* 5*, porque* $(\beta^3)^5 = 1$*. Otro ejemplo puede ser el elemento* $\beta + \beta^2$*, que cumple* $(\beta + \beta^2)^3 = \beta^3 + \beta^4 + \beta^5 + \beta^6 = \beta^3 + \beta + 1 + \beta^2 + \beta + \beta^3 + \beta^2 = 1$*. Por lo tanto* $O(\beta + \beta^2) = 3$*.*

Los siguientes resultados con los que vamos a trabajar tienen como objetivo factorizar $x^n - 1$ en $\mathbb{F}_p[x]$.

Proposición 2.56. *Sea* $\beta \in \mathbb{F}_q$ *un elemento primitivo del cuerpo,* β *es algebraico sobre* $\mathbb{F}_p$ *con* $M(x) \in \mathbb{F}_p[x]$ *su polinomio mínimo, contamos con las siguientes propiedades:*

- *Si* $f(x) \in \mathbb{F}_p[x]$ *cumple* $f(\beta) = 0$*, entonces* $M(x)|f(x)$*.*
- $M(x)|x^{p^m} - x$
- *El polinomio mínimo de un elemento primitivo de* $\mathbb{F}_p$ *tiene grado* m*.*

Ahora nos interesa saber cómo factorizar $\Phi_d(x) \in \mathbb{F}_p[x]$ en polinomios irreducibles, puesto que el polinomio mínimo $M(x)$ de β raíz primitiva n-ésima de la unidad será un factor de $\Phi_n(x)$. Para ello es necesario definir la función de Euler.

Definición 2.57 (Función de Euler)**.** *Definimos la* ***función de Euler*** *para un* $n \in \mathbb{N}$ *como*

$$\varphi(n) = \#\{x \in \mathbb{N}_{\geq 1} : 0 < x \leq n, \operatorname{mcd}(x, n) = 1\}$$

Proposición 2.58. *Sea* $x^n - 1 \in \mathbb{F}_p[x]$ *y sea* $m = O(p)$ *en* $(\mathbb{Z}/n\mathbb{Z}, \cdot)$*, entonces* $\Phi_n(x) \in \mathbb{F}_p[x]$ *factoriza en* $\frac{\varphi(n)}{m}$ *factores irreducibles de grado* m*.*

Se van a desarrollar cuatro ejemplos ilustrativos de cuatro casos para $n = 3, 7, 8, 15$, donde trabajaremos con las factorizaciones y elementos primitivos de $x^n - 1$ con $n = p^m - 1$, que pertenecen al cuerpo $\mathbb{F}_{p^m}$. Se recomienda prestar mayor atención a los casos $n = 7$ y $n = 15$ porque se volverá a ellos en el Capítulo 5 para construir los primeros códigos BCH.

Observación 2.59. *Para trabajar en los próximos capítulos, nos interesa que una base del cuerpo* $\mathbb{F}_q$ *como espacio vectorial esté dada por un elemento primitivo, así la base sea* $\{1, \beta, \beta^2, ..., \beta^{m-1}\}$ *y entonces* $\mathbb{F}_q = \{0, 1, \beta, ..., \beta^{n-1}\}$*. Por ello, en este trabajo, escogeremos* $f(x)$ *para generar el cuerpo como un anillo cociente, siempre y cuando* $f(x)|\Phi_n(x)$*, para que sus raíces sean primitivas.*

Ejemplo 2.60 ($n = 3$)**.** *Sea el cuerpo* $\mathbb{F}_4$ *donde* $p = m = 2$ *y* $n = 3 = 2^2 - 1$*. Sea* $\beta \in \mathbb{F}_4 \setminus \{0\}$ *raíz de* $x^3 - 1$*, su polinomio mínimo en* $\mathbb{F}_2[x]$ *es un factor irreducible de* $x^3 - 1$*. En concreto*

$$x^3 - 1 = \Phi_1(x)\Phi_3(x)$$

Si tomamos $\Phi_1(x) = x - 1$*, la raíz del polinomio es* 1*. Si tomamos* β *como raíz de* $\Phi_3(x) = x^2 + x + 1$*,* β *es un elemento primitivo y ese es su polinomio mínimo, ya que* $\Phi_3(x)$ *factoriza en* $\mathbb{F}_2[x]$ *en* $\frac{\varphi(3)}{2} = 1$ *polinomio de grado* 2*. Es decir es irreducible. Las raíces de* $x^3 - 1$ *pertenecen al cuerpo* $\mathbb{F}_{2^2} = \mathbb{F}_4 \cong \mathbb{F}_2[x] \Big/ (x^2 + x + 1) = \{0, 1, x, x + 1\}$

Ejemplo 2.61 ($n = 7$)**.** *Trabajamos en este caso con las raíces de* $x^7 - 1$ *que pertenecen al cuerpo* $\mathbb{F}_8$*. Se tiene* $p = 2$*,* $m = 3$*,* $n = 2^3 - 1$*. Sea* $\beta \in \mathbb{F}_8 \setminus \{0\}$ *una raíz de* $x^7 - 1$*, si* $\beta \neq 1$*,* β *tiene orden* 7 *por ser primo y es un elemento primitivo. Su polinomio mínimo será*

un factor ireducible de $\Phi_7(x)$ *que factoriza en* $\mathbb{F}_2[x]$ *en* $\frac{\varphi(7)}{3} = \frac{6}{3} = 2$ *factores irreducibles de grado* 3*. Vamos a calcularlos:*

$$x^7 - 1 = \Phi_1(x)\Phi_7(x) \Longrightarrow \Phi_7(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$$

Factorizamos en $\mathbb{F}_2[x]$ *y obtenemos* $\Phi_7(x) = (x^3 + x^2 + 1)(x^3 + x + 1)$ *irreducibles. Podemos encontrar las raíces de* $x^7 - 1$ *en* $\mathbb{F}_8 \cong \mathbb{F}_2[x] \Big/ (x^3 + x + 1)$*. Ya se han descrito los elementos del cuerpo en otros apartados. (Véase el Ejemplo 2.50 o la Tabla 5.1).*

Ejemplo 2.62 ($n = 8$)**.** *Veamos ahora la factorización y las raíces de* $x^8 - 1$ *en* $\mathbb{F}_3[x]$*. Sea* $p = 3$*,* $m = 2$*, entonces* $n = 3^2 - 1$*. Las raíces del polinomio pertenecen al cuerpo* $\mathbb{F}_9$*. Veámoslo. Sea* $\beta \in \mathbb{F}_9 \setminus \{0\}$ *raíz de* $x^8 - 1 \in \mathbb{F}_3[x]$*, si* $\beta \neq 1$*,* β *es raíz de* $\Phi_2(x) = x + 1$*,* $\Phi_4(x) = x^2 + 1$ *o* $\Phi_8(x)$*. Calculamos* $\Phi_8(x)$*:*

$$\Phi_8(x) = \frac{x^8 - 1}{\Phi_1(x)\Phi_2(x)\Phi_4(x)} = x^4 + 1 \in \mathbb{F}_3[x]$$

Se puede factorizar este polinomio en $\mathbb{F}_3[x]$ *en* $\frac{\varphi(8)}{2} = 2$ *polinomios de grado* 2*. Los calculamos y obtenemos* $x^2 + x + 2$*,* $x^2 + 2x + 2$*. Podemos generar* $\mathbb{F}_9$ *como* $\mathbb{F}_3[x] \Big/ (m(x))$*, siendo* $m(x)$ *cualquiera de los* 2 *factores irreducibles de* $\Phi_8(x)$ *y* β *será una raíz de* $m(x)$*. Por ejemplo, si tomamos* $m(x) = x^2 + x + 2$*, y* β *raíz, la otra raíz corresponde a* $2\beta + 2$*. Tanto* β *como* $2 + 2\beta$ *son elementos primitivos del cuerpo.* $\mathbb{F}_9 = \{0, 1, 2, \beta, 2\beta, 1 + \beta, 2 + \beta, 1 + 2\beta, 2 + 2\beta\}$*, siendo* $\beta^2 = 2\beta + 1$*. Las raíces de* $x^2 + 2x + 2$ *también son elementos primitivos del cuerpo.*

Ejemplo 2.63 ($n = 15$)**.** *Por último, trabajaremos con las raíces de* $x^{15} - 1$ *en el cuerpo* $\mathbb{F}_{16}$*. En este caso* $p = 2$*,* $m = 4$*, y* $n = 2^4 - 1$*. Sea* $\beta \in \mathbb{F}_{16} \setminus \{0\}$ *raíz de* $x^{15} - 1$*. Si* $\beta \neq 1$*,* β *debe ser raíz de* $\Phi_3(x)$*,* $\Phi_5(x)$ *y* $\Phi_{15}(x)$*. Calculamos los que no conocemos:*

- $\Phi_5(x) = x^4 + x^3 + x^2 + x + 1$ *(aplicando la Observación 2.45 por ser* 5 *primo).*
- $\Phi_{15}(x) = \frac{x^{15}-1}{\Phi_1(x)\Phi_3(x)\Phi_5(x)} = x^8 + x^7 + x^5 + x^4 + x^3 + x + 1$

El polinomio $\Phi_{15}(x)$ *se puede factorizar en* $\frac{\varphi(15)}{4} = 2$ *factores irreducibles de grado 4 en* $\mathbb{F}_2[x]$*. Los calculamos y obtenemos:* $x^4 + x + 1$*,* $x^4 + x^3 + 1$*. Las raíces de* $x^{15} - 1$ *pertenecen a* $\mathbb{F}_{16} \cong \mathbb{F}_2[x] \Big/ (m(x))$ *pudiendo ser* $m(x)$*:* $x^4 + x + 1$ *o* $x^4 + x^3 + 1$*. (Véase el Ejemplo 2.55).*

Se ha profundizado en estos ejemplos porque es importante conocer el polinomio $x^n - 1$. Se verá su utilidad para trabajar con la estructura algebraica de los códigos cíclicos que se trabajarán más adelante.

Capítulo 3

Preliminares de códigos. Códigos Lineales

Antes de adentrarnos en los códigos cíclicos, expondremos en este apartado algunos conceptos básicos de Teoría de Códigos, obtenidos de la Bibliografía [6].

Definición 3.1. *Un **alfabeto** es un conjunto finito $\mathcal{A} = \{a_1, a_2, ..., a_q\}$, donde los elementos a_i son las **letras o símbolos** del alfabeto. A las sucesiones finitas de elementos de $\mathcal{A}$ se les llama **palabras** o vectores.*

Definición 3.2. *Una palabra tiene **longitud** n cuando es un elemento del producto cartesiano $\mathcal{A} \times ... \times \mathcal{A} = \mathcal{A}^n$. El conjunto de todas las palabras sobre el alfabeto $\mathcal{A}$, las denotaremos como $\mathcal{A}^* = \mathcal{A} \cup \mathcal{A}^2 \cup \mathcal{A}^3 \cup ... \cup \mathcal{A}^n \cup ...$*

Definición 3.3 (Código)**.** *Un **código** $\mathcal{C}$ es un subconjunto de $\mathcal{A}^*$.*

Estos subconjuntos siguen un conjunto de reglas y convenios, cuyo objetivo es representar o convertir los datos. Su uso se basa en el procesamiento, almacenamiento y transmisión de información. En lo general trabajaremos con la familia de códigos que se describe a continuación.

Definición 3.4. *Llamamos **código bloque de longitud** n, a un código $\mathcal{C}$ donde todas las palabras tienen la misma longitud n, $\mathcal{C} \subseteq \mathcal{A}^n$.*

En este trabajo, nos referiremos a un código bloque simplemente como código. El alfabeto $\mathcal{A}$ se corresponderá con el conjunto de elementos del cuerpo finito $\mathbb{F}_q$, siendo $q = p^m$ con p un número primo y $m \in \mathbb{N}_{\geq 1}$. Esta notación se mantendrá en la teoría. En los ejemplos será habitual trabajar con códigos binarios, es decir con el alfabeto $\mathcal{A} = \{0, 1\} = \mathbb{F}_2$.

Definición 3.5. *Sea $x = (x_1, x_2, ..., x_n)$ una palabra de un código $\mathcal{C}$ que se transmite por el canal, y sea $y = (y_1, y_2, ..., y_n)$ la palabra recibida. Se dice que **se ha cometido un error** en la posición j si $x_j \neq y_j$.*

Ejemplo 3.6. *Se utilizarán estos códigos para ilustrar diversos conceptos y propiedades:*

a) $\mathcal{C}_1 = \{(0,0,0,0,0,0), (0,0,1,1,1,0), (0,1,0,1,0,1), (0,1,1,0,1,1), (1,0,0,0,1,1), (1,0,1,1,0,1), (1,1,0,1,1,0), (1,1,1,0,0,0)\} \subseteq \mathbb{F}_2^6$ *es un código bloque de longitud* 6. *Por ejemplo, la palabra* $(1,0,0,0,0,0)$ *tiene longitud 6 y sus elementos están en el alfabeto* $\mathbb{F}_2$ *pero no pertenece al código.*

b) $\mathcal{C}_2 = \{(0,0,0),(0,1,1),(1,0,1),(1,1,0)\} \subseteq \mathbb{F}_2^3$ *es un código bloque de longitud* 3*. Por ejemplo,* $(1,0,1)$ *es una palabra del código. Sin embargo,* $(1,0,0)$ *no pertenece al código, por lo que si recibimos esta palabra, necesariamente se ha producido un error en la transmisión.*

c) ***Código no binario:*** $\mathcal{C}_3 = \{(0,2,1,2,1,2,0),(2,0,1,1,0,1,1),(1,1,0,0,1,0,1),$ $(2,1,2,0,2,2,0),(0,1,0,2,2,1,2)\} \subseteq \mathbb{F}_3^7$ *es un código bloque de longitud* 7*, con palabras sobre el alfabeto* $\mathcal{A} = \{0,1,2\}$*. Por ejemplo, la palabra* $u = (2,1,1,0,2,1,0)$ *no está en el código.*

3.1. Códigos detectores y correctores de errores

La localización del error en la transmisión y su posible corrección encierra uno de los principales objetivos de la Teoría de Códigos. Se diferencian dos grandes clases de códigos: **detectores de errores**, y **detectores y correctores de errores**; introducidos por Richard Hamming en 1950, en la publicación de un artículo fundamental [3].
Se dice que un código detecta errores si, al transmitir una palabra por un canal se produce algún error y la palabra recibida no pertenece al código. La perturbación se puede producir por ejemplo a causa de un canal ruidoso. Un código corrige errores si además de detectarlos, es capaz de encontrar la posición del error y descubrir el mensaje original.
Es usual que los códigos **detectores de errores** transmitan además de la información, una cantidad extra redundante que permita la detección de la existencia de errores.

Ejemplo 3.7. *El pin de las tarjetas bancarias electrónicas es evaluado como una palabra de* 4 *letras. Si se introduce de manera incorrecta, el banco detecta que ha ocurrido un error, pero no conoce la posición del error.*

Los códigos **detectores y correctores de errores** deben incluir la suficiente información redundante, que permita al receptor recuperar el mensaje. Existirá una cota para la cantidad de errores producidos en la transmisión.

Ejemplo 3.8. *a) Los CDs pueden recuperar los datos grabados de esta manera, es por eso que un disco rallado puede sonar correctamente a pesar de estar sucio o dañado.*

b) Otro importante ejemplo para la recuperación de los datos es la letra del DNI, donde se utiliza la congruencia módulo 23 en el número del DNI, y asignándole una letra con una tabla de equivalencias. Gracias a este mecanismo se pueden detectar errores que pueden venir de un fraude interesado o un simple error manual. Se podrá corregir el error sólo si se sabe la posición de éste.

Una herramienta sencilla de localización y corrección de errores es el uso de códigos de repetición. Esta codificación consiste en repetir cada símbolo del mensaje original un número k determinado de veces. Esa redundancia permite la detección y corrección de errores de la transmisión.

Ejemplo 3.9. *Un ejemplo sencillo.*
Con $k = 7$*, se recibe la palabra* $y = (1000100)$*. La opción más probable es que la palabra enviada haya sido* $x = (0000000)$*, es decir, se han cometido* 2 *errores en la transmisión. La otra opción sería que la palabra enviada corresponda a* $x = (1111111)$ *en cuyo caso se habrían cometido* 5 *errores.*

Será una ventaja en el código que las palabras que lo forman sean lo más distintas posible, para impedir que las perturbaciones en el canal transformen unas en otras. Esta idea es la base de los códigos correctores. La diferencia entre palabras se mide con la distancia de Hamming.

Definición 3.10 (Distancia de Hamming)**.** *La **distancia** entre dos palabras $x = (x_1, ..., x_n)$ e $y = (y_1, ..., y_n)$, $x, y \in \mathcal{A}^n$, es el número de elementos diferentes en cada palabra que ocupan la misma posición. Lo denotamos como:*

$$d(x, y) = |\{i : 1 \leq i \leq n, x_i \neq y_i\}|$$

Si se emite una palabra $x \in \mathcal{C}$ y se recibe y, suponiendo que se han cometido el menor número de errores, podemos decodificarla por la palabra de $\mathcal{C}$ a una menor distancia de Hamming de y.

Ejemplo 3.11. *Tomamos el código $\mathcal{C}_3$ del Ejemplo 3.6 c).*

$$\mathcal{C}_3 = \{v_1 = (0212120), v_2 = (2011011), v_3 = (1100101), v_4 = (2120220), v_5 = (0102212)\}$$

Si recibimos la palabra $u = (2, 1, 1, 0, 2, 1, 0)$, que como hemos dicho no está en el código, y comparamos con las palabras del código:
$d(u, v_1) = 5, d(u, v_2) = 4, d(u, v_3) = 5, d(u, v_4) = 2, d(u, v_5) = 4$.
La palabra más cercana a u es v_4, entonces decodificamos la palabra como v_4 suponiendo que el error cometido es el menor posible.

Podemos encontrar problemas cuando hay más de una palabra a la misma distancia.

Definición 3.12 (Detección y corrección de errores)**.**

- *Se dice que un código $\mathcal{C}$ **detecta** s **errores** si al recibirse $y \in \mathbb{F}_q^n$ con a lo sumo s errores, el receptor es capaz de determinar si ha habido errores o no en la transmisión. Es decir, si $y \in \mathcal{C}$ o $y \notin \mathcal{C}$.*
- *Un código $\mathcal{C}$ **corrige** s **errores** si al recibirse $y \in \mathbb{F}_q^n$ con a lo sumo s errores, el receptor puede determinar unequívocamente cuál fue la palabra enviada $x \in \mathcal{C}$. Al máximo s tal que $\mathcal{C}$ corrige s errores lo denominamos **capacidad correctora del código**.*

Definición 3.13 (Distancia mínima)**.** *Dado un código bloque $\mathcal{C}$ definimos la **distancia mínima del código** como la mínima distancia entre dos palabras del código.*

$$d(\mathcal{C}) = \min\{d(x, y) | x, y \in \mathcal{C}, x \neq y\}$$

Ejemplo 3.14. *Volviendo al Ejemplo 3.11. Calculamos la distancia de Hamming entre las palabras del código:*
$d(v_1, v_2) = 6, d(v_1, v_3) = 6, d(v_1, v_4) = 5, d(v_1, v_5) = 5, d(v_2, v_3) = 6, d(v_1, v_4) = 6, d(v_2, v_5) = 6, d(v_3, v_4) = 5, d(v_3, v_5) = 5, d(v_4, v_5) = 5$
La distancia mínima del código es: $d(\mathcal{C}) = 5$.

Teorema 3.15. *Sea $\mathcal{C}$ un código bloque con distancia mínima d se tiene que:*

- *$\mathcal{C}$ detecta hasta d-1 errores.*
- *$\mathcal{C}$ corrige hasta $\left\lfloor \frac{d-1}{2} \right\rfloor$ errores.*

Ejemplo 3.16. *Volviendo al código $\mathcal{C}_1$ (Ejemplo 3.6), su distancia mínima es 3 (se calcula de forma sencilla más adelante). Este código tiene la capacidad de detectar hasta 2 errores y de corregir 1 error.*

3.2. Códigos lineales

De manera más específica, nos adentraremos en las propiedades básicas de los códigos lineales y su interés en la corrección de errores. Además comenzaremos a observar la estrecha relación de la Teoría de Códigos con el área de Álgebra. Desde aquí se trabaja con códigos finitos cuyo alfabeto corresponde a un cuerpo finito $\mathbb{F}_q$ con $q = p^m$ con p primo y $m \in \mathbb{N}$.

Definición 3.17 (Código lineal)**.** *Llamaremos* ***código lineal*** *a un código $\mathcal{C}$ si es un subespacio lineal de $\mathbb{F}_q^n$.*

Podemos definir una aplicación lineal e inyectiva

$$f : \mathbb{F}_q^k \to \mathbb{F}_q^n$$

donde $\mathbb{F}_q^k$ sea la fuente y f defina la codificación de la fuente. Obtenemos como imagen de la aplicación $f\left(\mathbb{F}_q^k\right) \subseteq \mathbb{F}_q^n$ un subespacio vectorial de dimensión k al que llamaremos código lineal $\mathcal{C}$. Su cardinal será siempre q^k, si el alfabeto tiene q elementos.

Notación. *Definimos un $[n, k, d]-$código lineal como $\mathcal{C}$ un subespacio vectorial de $\mathbb{F}_q^n$ de dimensión k y distancia mínima d.*

Ejemplo 3.18. *Revisando los códigos ya definidos en el ejemplo 3.6:*

a) $\mathcal{C}_1 \subseteq \mathbb{F}_2^6$ es $[6, 3, 3]$-código lineal. $q = 2$, es un código binario. $n = 6$, sus palabras tienen longitud 6, la dimensión del subespacio lineal es 3, y la distancia mínima es 3.

b) $\mathcal{C}_2 \subseteq \mathbb{F}_2^3$ es $[3, 2, 2]$-código lineal. Sus parámetros son $n = 3, q = 2, d = 2$.

c) El código no binario $\mathcal{C}_3 \subseteq \mathbb{F}_3^7$ no es un código lineal porque no contiene al vector $(0, 0, 0, 0, 0, 0, 0)$, y por lo tanto no es un subespacio lineal.

Dado que toda aplicación es representada por una matriz, intuitivamente se puede pensar que existe una matriz que representa el código.

Definición 3.19 (Matriz generadora del código)**.** *Dado $\mathcal{C}$ un código lineal, la* ***matriz G generadora del código****, será la matriz de la aplicación biyectiva $f : \mathbb{F}_q^k \to \mathcal{C} \subseteq \mathbb{F}_q^n$. Podemos interpretarlo como: G una matriz $k \times n$, con filas formando una base de $\mathcal{C}$.*

La matriz G nos proporciona una codificación de la fuente $\mathbb{F}_q^k$, ya que podemos generar $\mathcal{C}$ operando con la matriz de la aplicación inyectiva G. Es decir:

$$\mathcal{C} = \left\{uG \ : \ u \in \mathbb{F}_q^k\right\}$$

El esquema de transmisión sería el siguiente:

$$\begin{array}{ccc} u & \longrightarrow & uG \\ \text{mensaje} & & \text{codificación} \end{array}$$

De esta manera simplificamos la codificación y disminuimos el almacenamiento, utilizando herramientas algebraicas.

Notación. *Dada la matriz G, denotamos con G_i a la fila i de la matriz G.*

Ejemplo 3.20. *En el código $\mathcal{C}_1$, la matriz generadora es:*

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

Obtenemos las 8 palabras del código (descritas en Ejemplo 1.1) con:

$$u_1 \cdot G_1 + u_2 \cdot G_2 + u_3 \cdot G_3 \text{ con } u_1, u_2, u_3 \in \{0, 1\}$$

Suponemos que tenemos el mensaje $u = (1, 0, 1) \in \mathbb{F}_2^3$*, en este caso obtenemos la palabra del código:* $(1, 0, 1, 1, 0, 1) \in \mathbb{F}_2^6$.

Se ha comprobado que G define unas ecuaciones paramétricas del código. Se define ahora la matriz de control del código, con la que se obtienen las ecuaciones implícitas del subespacio $\mathcal{C}$.

Definición 3.21 (Matriz de control)**.** *Sea* $\mathcal{C} \subseteq \mathbb{F}_q^n$ *un* $[n, k, d]-$*código lineal, definimos la matriz H como la* ***matriz de control del código*** *si:*

$$x = (x_1, ..., x_n) \in \mathcal{C} \iff Hx^t = (0, ..., 0)^t \in \mathbb{F}_q^{n-k}$$

La matriz H tiene tamaño $(n - k) \times n$ *y rango* $n - k$*.*

Ejemplo 3.22. *Matriz de control o "parity check matrix" H.*

a) Volviendo a $\mathcal{C}_1 \subseteq \mathbb{F}_2^6$*, y su matriz de control H, tenemos:*

$$H \cdot x^t = 0 \in \mathbb{F}_2^6 \iff \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \iff \begin{cases} x_2 + x_3 + x_4 = 0 \\ x_1 + x_3 + x_5 = 0 \\ x_1 + x_2 + x_6 = 0 \end{cases}$$

Buscamos todas las palabras que cumplan las ecuaciones:
Si $x_1 = 0$ *entonces:*

$$\begin{cases} x_3 + x_5 = 0 \\ x_2 + x_6 = 0 \\ x_2 + x_3 + x_4 = 0 \end{cases} \implies \begin{cases} x_3 = x_5 \\ x_2 = x_6 \\ x_2 = x_3 + x_4 \end{cases}$$

Bajo estas premisas obtenemos las palabras: (011011), (010101), (000000), (001110)*.*
Si $x_1 = 1$ *entonces:*

$$\begin{cases} x_2 + x_3 + x_4 = 0 \\ 1 + x_3 + x_5 = 0 \\ 1 + x_2 + x_6 = 0 \end{cases} \implies \begin{cases} x_2 + x_3 + x_4 = 0 \\ x_3 \neq x_5 \\ x_2 \neq x_6 \end{cases}$$

Obtenemos las palabras (110110), (111000), (101101), (100011)*. Se puede comprobar que estas palabras corresponden con el código* $\mathcal{C}_1$ *del Ejemplo 3.6.*

b) Sea $\mathcal{C}_4 \subseteq \mathbb{F}_2^5$*„ con matriz de control H, tenemos el siguiente ejemplo:*

$$H \cdot x^t = 0 \iff \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \iff x_1 + x_i = 0 \forall i \in 2, ..., 5$$

Como estamos en un alfabeto binario, tenemos:

$$x_1 = x_2 = x_3 = x_4 = x_5$$

Por lo que las dos palabras del código son $(1, 1, 1, 1, 1)$ *y* $(0, 0, 0, 0, 0)$.

Proposición 3.23. *Si G y H son las matrices generadora y de control de un código lineal $\mathcal{C}$, entonces se cumple $GH^t = 0$.*

En códigos lineales, se define un nuevo término para trabajar con la distancia $d(\mathcal{C})$, el peso.

Definición 3.24 (Peso)**.** *El **peso** de un vector $x = (x_1, ..., x_n) \in \mathbb{F}_q^n$ es el número de elementos de x diferentes de 0. Lo denotamos como $w(x)$.*

$$w(x) = d(x, 0) = |\{i : 1 \leq i \leq n, x_i \neq 0\}|$$

Observación 3.25. *Es claro de esto que: $d(x, y) = w(x - y)$ $\forall x, y \in \mathbb{F}_q^n$.*

Proposición 3.26. *Dado un código $\mathcal{C}$, la distancia mínima del código, que denotamos como $d(\mathcal{C})$ será el menor de los pesos de las palabras no nulas del código, es decir:*

$$d(\mathcal{C}) = \min_{x \in \mathcal{C}, x \neq 0} w(x)$$

Proposición 3.27. *La distancia mínima d coincide con el menor número de columnas linealmente dependientes de H.*

Ejemplo 3.28. *Obtener la distancia mínima del código $\mathcal{C}_1$. Podemos evaluar el peso de las 7 palabras no nulas del código.*
$w(0, 0, 1, 1, 1, 0) = w(0, 1, 0, 1, 0, 1) = w(1, 0, 0, 0, 1, 1) = w(1, 1, 1, 0, 0, 0) = 3$
$w(0, 1, 1, 0, 1, 1) = w(1, 0, 1, 1, 0, 1) = w(1, 1, 0, 1, 1, 0) = 4$
La distancia mínima d del código es 3. De la misma forma, mirando la matriz de control H del Ejemplo 3.22 todas las columnas son independientes dos a dos. Sin embargo, la primera, la quinta y la sexta son dependientes.

La base de un espacio vectorial no es única, por lo tanto, la matriz generadora de un código lineal tampoco lo es. Entonces, cómo saber si dos matrices generan el mismo código, ó cómo saber si dos códigos son equivalentes.

Definición 3.29 (Códigos equivalentes)**.** *Se dice que dos **códigos son equivalentes** si uno se puede obtener del otro mediante alguna permutación de las coordenadas en $\mathbb{F}_q^n$.*

Proposición 3.30. *Dada una matriz de control H que representa un código $\mathcal{C}$, y una matriz de control H' que representa al código $\mathcal{C}'$. Los códigos $\mathcal{C}$ y $\mathcal{C}'$ son códigos equivalentes si H y H' tienen las mismas columnas, aunque en distinto orden.*

3.2.1. Decodificación por síndrome

Vamos a introducir el concepto de síndrome. Dado que los códigos lineales permiten detectar y corregir errores gracias a su estructura algebraica, esta es una herramienta fundamental en este proceso. Utilizando la matriz de control del código, el síndrome permite identificar si una palabra recibida pertenece al código y, en caso contrario, proporciona información crucial sobre el error ocurrido, facilitando su corrección.

Definición 3.31 (Síndrome)**.** *Sea $\mathcal{C}$ un código lineal con matriz de control H y sea $y \in \mathbb{F}_q^n$. Se llama **síndrome** de y al vector $S(y) = Hy^t \in \mathbb{F}_q^{n-k}$.*

Proposición 3.32. *Propiedades del síndrome:*

- *Por definición de H: $x \in \mathcal{C}$ si y sólo si $S(x) = (0, ..., 0)^t \in \mathbb{F}_q^{n-k}$*
- *Si se transmite una palabra $x \in \mathcal{C}$, se recibe una palabra $y \notin \mathcal{C}$, se tiene $y = x + e$ con e el vector de error. Entonces $S(y) = S(x + e) = S(x) + S(e) = S(e) \in \mathbb{F}_q^{n-k}$*

Observación 3.33. *Si no ocurren errores en la transmisión, $S(y)$ debe ser 0.*

La observación anterior no se da al revés, es decir, aunque el síndrome de una palabra sea 0, pueden haber ocurrido errores. Aunque, teniendo en cuenta el primer punto de la Proposición 3.32, la palabra debe pertenecer al código. Esto significa que hemos obtenido otra palabra del código, diferente de la que se pretendía comunicar.

Es por ello que el síndrome es útil para identificar la aparición de algún error en la transmisión. Además, a través de e podemos localizar esos errores y corregirlos. Veamos un ejemplo:

Ejemplo 3.34. *Consideramos de nuevo el código $\mathcal{C}_1$.*
Sea $y = (110100) \in \mathbb{F}_2^6$, el síndrome de y es:

$$S(y) = Hy^t = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Esto quiere decir que la palabra y no pertenece a $\mathcal{C}_1$, y por lo tanto, se ha cometido un error en la transmisión, este error depende del vector $e \in \mathbb{F}_2^6$.
Si $y = x + e$ vamos a localizar el vector e.

$$S(y) = S(e) \iff \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \iff \begin{cases} e_2 + e_3 + e_4 = 0 \\ e_1 + e_3 + e_5 = 1 \\ e_1 + e_2 + e_6 = 0 \end{cases}$$

Como estamos en el alfabeto $\mathbb{F}_2$, $e_i \in \{0, 1\}$, y el rango de la matriz H es 3. Este sistema tiene $2^k = 2^3 = 8$ soluciones posibles:

$$\begin{array}{cccc} (101111) & (111010) & (001100) & (011001) \\ (110100) & (100001) & (000010) & (010111) \end{array}$$

Existen varias soluciones posibles del sistema, pero solo una es la más probable. Buscamos aquella con menor distancia de Hamming, por lo que elegimos el vector de menor peso: $e = (000010)$*. Este vector indica un error en* y *en la posición* 5*. Observando los elementos del código, concluimos que la palabra más cercana a* y *es* $x = (110110) \in \mathcal{C}_1$*, donde* $d(x, y) = 1$*.*

3.2.2. Códigos de Hamming

Se utilizan más adelante códigos de Hamming (en concreto binarios) para introducir los códigos BCH. Por eso en esta sección se hace una breve introducción de esta familia de códigos lineales.
Los códigos de Hamming son una importante familia de códigos lineales, dada su fácil codificación y decodificación. Tienen capacidad para detectar dos errores y corregir un solo error.

Definición 3.35. *Un* ***código de Hamming binario*** *es un* $[n = 2^m - 1, k = 2^m - 1 - m, d = 3]-$*código lineal. Lo denotaremos como* $\mathcal{H}_m$ *con* $m \geq 2$*. Las columnas de la matriz de control* H *están determinadas por todos los vectores binarios de longitud* m *distintos de* 0*.*

Ejemplo 3.36. $\mathcal{H}_3$ *es un* ***código de Hamming binario*** *con* $m = 3$ *y parámetros* $[7, 4, 3]$*. Su matriz de control* H *es:*

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

Donde podemos encontrar todas las $3-$*tuplas binarias distintas de* 0*.*

Observación 3.37. *El código de Hamming* $\mathcal{H}_m$ *viene representado por sus parámetros y una matriz* H *matriz de control del código. Tal y como hemos descrito en su construcción, las columnas de* H *son todas las* m*-tuplas distintas de* 0*. El código de Hamming* $\mathcal{H}_m$ *es único como código lineal, módulo permutación de sus columnas. Es decir si permutamos las columnas de* H *obtenemos códigos equivalentes. (Véase la Definición 3.29).*

Capítulo 4

Códigos cíclicos

Recordemos que se está trabajando con códigos cuyo alfabeto es un cuerpo finito.
Los códigos cíclicos son unos de los más estudiados de la Teoría de Códigos, debido a su fácil decodificación. Una importante familia de códigos cíclicos son los códigos BCH, en los cuales se adentrará este texto más adelante.
Primero necesitamos conocer la siguiente definición:

Definición 4.1 (Permutación. Permutación cíclica)**.** *Sea $X = \{1, 2, ..., n\}$, y σ una aplicación biyectiva $\sigma : X \to X$ que llamaremos* ***permutación****. Sean $i_1, i_2, ..., i_l$ elementos de X distintos dos a dos, llamaremos a σ* ***permutación cíclica*** *de longitud l si cumple*

$$\sigma(i_1) = i_2, \sigma(i_2) = i_3, ..., \sigma(i_l) = i_1$$

y deja fijos los elementos $X \setminus \{i_1, i_2, ..., i_l\}$.

A partir de aquí nos referiremos con permutación cíclica a una permutación cíclica de longitud n.

Definición 4.2 (Código cíclico)**.** *Un código $\mathcal{C} \subseteq \mathbb{F}_q^n$, es* ***cíclico*** *si es lineal y además si dado un elemento $c = (c_0, ..., c_{n-1}) \in \mathcal{C}$ entonces $\hat{c} = (c_1, ..., c_{n-1}, c_0) \in \mathcal{C}$, siendo $\hat{c}$ una permutación cíclica de c. Es decir, si toda permutación cíclica de una palabra del código pertenece también al código.*

Ejemplo 4.3.
- *Dado el Código $\mathcal{C}_5 = \{a = (000), b = (110), c = (101), d = (011)\} \subseteq \mathbb{F}_2^3$. Cualquier permutación cíclica de sus elementos está en el código.*
- *Los códigos de Hamming son códigos cíclicos.*

Haciendo uso del álgebra, siendo $c = (c_0, ..., c_{n-1})$ un elemento de $\mathbb{F}_q^n$, podemos identificar a c con el polinomio $c(x) = c_0 + c_1x + ... + c_{n-1}x^{n-1} \in \mathbb{F}_q[x]$.

Notación. *Denotaremos con c tanto a una palabra del código $\mathcal{C} \subseteq \mathbb{F}_q^n$ como a un polinomio en $\mathbb{F}_q[x]$ con grado menor que n.*

Ejemplo 4.4. *Dado el Código $\mathcal{C}_5 = \{(000), (110), (101), (011)\} \subseteq \mathbb{F}_2^3$ podemos asociar cada palabra con un polinomio de la siguiente manera:*

$$\begin{aligned} (000) &\longleftrightarrow 0 \\ (110) &\longleftrightarrow 1 + x \\ (101) &\longleftrightarrow 1 + x^2 \\ (011) &\longleftrightarrow x + x^2 \end{aligned}$$

Podemos considerar un código cíclico contenido en $\mathbb{F}_q^n$ como un ideal del anillo $R_n \cong \mathbb{F}_q[x]\Big/(x^n-1)$. Vamos a formalizarlo con una proposición. Si es necesario, se recomienda revisar los resultados correspondientes de la Sección 2.2.1.

Observación 4.5. *Dado $c(x) \in R_n$, si multiplicamos por x obtenemos una permutación cíclica de longitud n de la palabra c:*

$$x \cdot c(x) = c_0x + c_1x^2 + ... + c_{n-1}x^n \equiv c_{n-1} + c_0x + ... + c_{n-2}x^{n-1}$$

Teniendo en cuenta que $x^n \equiv 1$ en R_n.

Proposición 4.6. *Sea $\mathcal{C} \subseteq \mathbb{F}_q^n$ un código lineal, es cíclico si y solo sí es ideal de R_n.*

Demostración. Demostramos cada implicación:

- $\Longrightarrow$: Suponemos que $\mathcal{C} \subseteq R_n$ es un código cíclico de longitud n. Contamos con sus propiedades lineales y cíclicas: $\mathcal{C} \neq \emptyset$ y si $c(x) \in \mathcal{C}$ entonces $xc(x) \in \mathcal{C}$.
  (Utilizamos la Proposición 2.9). Dados dos elementos $c, \hat{c} \in \mathcal{C}$ y $r \in R_n$:
  - $c(x) - \hat{c}(x) \in \mathcal{C}$ por ser subespacio lineal (suma y producto por escalar -1).
  - $r(x) \cdot c(x) \in \mathcal{C}$. Cuando se multiplica por un escalar (coeficientes de $r(x)$) por linealidad, y cuando se multiplica por alguna potencia de x por permutación cíclica.

  Entonces $\mathcal{C}$ es ideal de R_n.
- $\Longleftarrow$: Suponemos $\mathcal{C}$ ideal de R_n. Por lo tanto, $\mathcal{C} \subseteq R_n$ y sus elementos tienen grado menor que n. Si $c(x) \in \mathcal{C}$ entonces $x \cdot c(x) \in \mathcal{C}$ por ideal. Esto corresponde a una permutación cíclica. Por ello, $\mathcal{C}$ es un código cíclico.

□

Ejemplo 4.7. *Sea el código $\mathcal{C}_5 \subseteq \mathbb{F}_2^3$ un código cíclico (véase el Ejemplo 4.4), veamos que es un ideal en $R_3 = \mathbb{F}_2[x]\Big/(x^3-1)$. Los polinomios representados en el Ejemplo 4.4 tienen grado menor que 3, por lo tanto pertenecen a R_3. Ahora aplicando el Test de Caracterización (Proposición 2.9):*
Si sumamos cualquiera de las palabras del código, obtenemos otra palabra del código. Es simple comprobarlo con los elementos de $\mathcal{C}_3$.
Veamos el producto de un elemento $c(x) \in \mathcal{C}_3$ por un elemento cualquiera $r(x) = r_0 + r_1x + r_2x^2 \in R_3$, con $r_i \in \mathbb{F}_2$: $c(x) \cdot r(x) = c(x) \cdot r_0 + c(x) \cdot r_1 \cdot x + c(x) \cdot r_2 \cdot x^2$ El producto $c(x) \cdot r_i$ solo puede ser $0 \in \mathcal{C}_3$ (si $r_i = 0$) o $c(x) \in \mathcal{C}_3$ (si $r_i = 1$). El resto es trivial por ser permutaciones cíclicas de $\mathcal{C}_3$.

La estructura algebraica de un código cíclico como ideal de este particular anillo motiva el enunciado de algunas propiedades útiles en el siguiente teorema.

Teorema 4.8 (Teorema 1 del Capítulo 7 en [6])**.** *Sea $\mathcal{C} \neq (0)$ ideal de R_n, es decir un código cíclico de orden n, se tiene:*

I. *Hay un único polinomio mónico $g(x)$ de grado mínimo r en $\mathcal{C}$*

II. $\mathcal{C} = (g(x))$

III. $g(x) | x^n - 1$

IV. *Todo $c(x) \in \mathcal{C}$ se puede escribir de forma única como $c(x) = f(x)g(x)$, con $f(x) \in \mathbb{F}_q[x]$ de grado menor que $n - r$*

V. *Si $g(x) = g_0 + g_1x + ... + g_rx^r$, entonces G es matriz generadora de $\mathcal{C}$*

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & ... & g_r & & 0 \\ & g_0 & g_1 & ... & g_{r-1} & g_r & \\ & & ... & & & & \\ 0 & & & g_0 & ... & ... & g_r \end{pmatrix} = \begin{pmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{n-r-1}g(x) \end{pmatrix}$$

VI. $dim(\mathcal{C}) = n - r$

Demostración. I. Suponemos que existen dos polinomios mónicos $f(x)$ y $g(x)$ en $\mathcal{C}$, con $f \neq g$ de grado mínimo $r \in \mathbb{N}$. Dado que $f(x) - g(x) \in \mathcal{C}$ y ambos son mónicos, su grado será menor que r. Llegamos a una contradicción, ya que el de grado mínimo es único.

II. Dado $c(x) \in \mathcal{C}$ un elemento cualquiera del código, sabemos por (I) $deg(g(x)) \leq deg(c(x))$ y $q(x), r(x)$ cociente y resto de la división euclidea de c entre g expresamos

$$c(x) = q(x) \cdot g(x) + r(x)$$

Donde $deg(r(x)) < r$ y $deg(q) < n - r$. Por otro lado, siendo $\mathcal{C}$ un código cíclico, por tanto ideal, $r(x) = c(x) - q(x) \cdot g(x) \in \mathcal{C}$.
Como r es el grado mínimo, y el grado de $r(x)$ es menor estrictamente que r, concluimos que $r(x) = 0$, entonces $c(x) = q(x) \cdot g(x)$, y por tanto $c(x) \in (g(x))$.

III. Si dividimos $x^n - 1$ entre $g(x)$ obtenemos $q(x)$ y $r(x)$ dos polinomios en $\mathbb{F}_q[x]$ tales que $x^n - 1 = q(x)g(x) + r(x)$ con $gr(r(x)) < r$. Como $q(x)g(x) \in \mathcal{C}$ módulo $x^n - 1$ y $r(x) \equiv -q(x)g(x)$ módulo $x^n - 1$, suponemos $r(x) \neq 0$ en ese caso $deg(r(x)) < r$ donde obtenemos un absurdo porque r es el grado mínimo. Entonces $r(x) = 0$ entonces $x^n - 1 = q(x)g(x)$ entonces $g(x)$ divide a $x^n - 1$.

IV. Por (II) con $g(x) \in R_n$ queda determinada la existencia de un $f(x) \in \mathbb{F}_q[x]$ tal que $c(x) = f(x)g(x)$ con $deg(f) < n - r$ (por tener $deg(g) = r$ y $deg(c) < n$). Ahora, suponemos por reducción al absurdo que existen dos elementos $f(x), h(x) \in K[x]$, $deg(f) = deg(h) < n - r$ tales que $c(x) = f(x)g(x)$ y $c(x) = h(x)g(x)$. Entonces:

$$f(x)g(x) = h(x)g(x) \Longrightarrow (f(x) - h(x))\, g(x) = 0$$

Como $g(x) \neq 0$, entonces $f(x) - h(x) = 0$, esto significa que $f(x) = h(x)$, donde llegamos a un absurdo.

V. Demostramos que G es una matriz generadora del código a partir de la Definición 3.19. Dado $c(x) \in \mathcal{C}$, por (IV), existe $f(x)$ con $deg\,(f(x)) < n - r$ tal que $c(x) = f(x)g(x)$. Si desarrollamos la expresión

$$c(x) = f(x)g(x) = f_0g(x) + f_1xg(x) + ... + f_{n-r-1}x^{n-r-1}g(x)$$

encontramos una combinación lineal de las filas de G, que es única por (IV), entonces $\{g(x), xg(x), ..., x^{n-r-1}g(x)\}$ es una base del código $\mathcal{C}$, y por tanto G, es matriz generadora de $\mathcal{C}$.

VI. La base de $\mathcal{C}$ tiene $n-r$ elementos linealmente independientes, lo que determina la dimensión de $\mathcal{C}$ como subespacio vectorial.

□

Definición 4.9 (Polinomio generador)**.** *El polinomio $g(x)$ que cumpla las propiedades anteriores será el* ***polinomio generador del código****.*

Notación. *Sea n la longitud de las palabras del código, k la dimensión del código y r el grado mínimo (o grado del polinomio generador), se relacionan con $r = n-k$. En concreto r y el parámetro m (que aparece en las secciones referidas a Códigos de Hamming) representan el mismo valor. Lo podemos llamar la co-dimensión del código.*

Observación 4.10. *Dado $f(x) \in \mathbb{F}_q[x]$ y $g(x) \in \mathcal{C}$ polinomio generador, la transmisión de información en el código se representa de la siguiente manera:*

$$\begin{array}{ccc} f(x) & \longrightarrow & f(x)g(x) \\ \textit{mensaje} & & \textit{palabra código} \end{array}$$

Corolario 4.11. *Todo ideal de R_n es ideal principal.*

Demostración. Consecuencia del punto (II) del Teorema 4.8. Si $\mathcal{C}$ es un código cíclico, es un ideal de R_n (Proposición 4.6), y $\mathcal{C} = \langle g(x) \rangle$ es principal. □

Corolario 4.12. *Todo $g(x)$ divisor de $x^n - 1$ determina un código cíclico.*

Ejemplo 4.13. *Consideremos un código cíclico $\mathcal{C} \subseteq \mathbb{F}_2^7$ definido por su polinomio generador $g(x)$ divisor de $x^n - 1$. Tomamos el polinomio generador*

$$g(x) = x^3 + x + 1, \frac{x^7 - 1}{g(x)} = x^4 + x^2 + x + 1$$

El polinomio divide a $x^7 - 1$, lo que garantiza que genera un código de longitud $n = 7$. $g(x)$ tiene grado $r = 3$. El código tendrá dimensión $k = n - r = 4$.
La matriz generadora del código se construye por las permutaciones cíclicas de $g(x)$, por lo tanto la matriz generadora del código es:

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

Una vez vista la relación entre la matriz generadora y el polinomio generador, es inevitable pensar en la existencia de un poliomio de control, relacionado con la matriz de control.

Definición 4.14 (Polinomio de control.)**.** *Sea $\mathcal{C}$ un código cíclico con polinomio generador $g(x)$, llamaremos $h(x)$ al* ***polinomio de control*** *definido como:*

$$h(x) = (x^n - 1)/g(x)$$

Observación 4.15. *Si el grado de $g(x)$ es r, el polinomio de control $h(x)$ tiene grado $n-r$.*

Ejemplo 4.16. *En el Ejemplo 4.13 el polinomio de control del código es*

$$h(x) = x^4 + x^2 + x + 1$$

Teorema 4.17. *Sea $\mathcal{C}$ un código cíclico, y $h(x) = h_0 + h_1x + ... + h_{n-r}x^{n-r}$ su polinomio de control, con las características anteriores, entonces:*

I. *Sea $c \in R_n$,*

$$c \in \mathcal{C} \iff c(x)h(x) \equiv 0$$

II. *Sea $c \in R_n$*

$$c \in \mathcal{C} \iff Hc^t = 0 \in \mathbb{F}_q^r$$

donde H es la siguiente matriz de control de $\mathcal{C}$

$$H = \begin{pmatrix} & & & h_{n-r} & ... & h_2 & h_1 & h_0 \\ & & h_{n-r} & ... & h_2 & h_1 & h_0 & \\ & & & & & & & \\ h_{n-r} & ... & h_2 & h_1 & h_0 & & & \end{pmatrix} = \begin{pmatrix} h(x) \\ xh(x) \\ \vdots \\ x^{r-1}h(x) \end{pmatrix}$$

Demostración. I. Veamos las implicaciones por separado:

($\Longrightarrow$): Tengamos en cuenta el Teorema 4.8, que nos dice que podemos expresar cualquier $c \in \mathcal{C}$ como $c(x) = f(x)g(x)$, siendo $g(x)$ el polinomio generador y $deg(f(x)) < n - r$.

$$c(x)h(x) = f(x)g(x)h(x) = f(x)g(x)\tfrac{x^n-1}{g(x)} = f(x)(x^n - 1) \equiv 0 \Longrightarrow c(x)h(x) \equiv 0$$

($\Longleftarrow$): Sea $c \in R_n$ y $c(x)h(x) \equiv 0$. Sean $q(x)$, $r(x)$ cociente y resto de la división euclídea de $c(x)$ entre $g(x)$

$$c(x) = q(x)g(x) + r(x) \text{ con } deg(r(x)) < r \Longrightarrow$$
$$c(x)h(x) = q(x)g(x)h(x) + r(x)h(x) = q(x)(x^n - 1) + r(x)h(x) = 0$$

Está claro que $q(x)(x^n - 1) \equiv 0$, por ello $r(x)h(x) \equiv 0$.Teniendo en cuenta que $deg(r(x)) < r$ y $deg(h(x)) = n-r$, es directo observar que $deg(r(x)h(x)) < n-r+r = n$, y $r(x)h(x) \equiv 0$, entonces $r(x) = 0$ y $c(x) = q(x)g(x) \in \mathcal{C}$.

II. Veamos cada implicación.

($\Longrightarrow$): Sea $c \in \mathcal{C}$ por (I) tenemos que $h(x)c(x) \equiv 0$. Existe f tal que $h(x)c(x) = f(x)(x^n - 1)$, y $deg(h(x)) = n - r$ entonces $deg(h(x)c(x)) < n + n - r = 2n - r$, y teniendo en cuenta que $deg(x^n - 1) = n$, se tiene $deg(f(x)) < n - r$. Esto implica que todos los coeficientes de $f(x)(x^n - 1)$ desde el término x^{n-r} hasta x^{n-1} son 0.
El coeficiente de $h(x)c(x)$ para el término x^j será

$$\sum_{i=0}^{j} h_i c_{j-i} = 0$$

con $j = 0, ..., n-1$. Así obtenemos las siguientes r ecuaciones:

$$\begin{aligned} h_{n-r}c_{r-1} + h_{n-r-1}c_r + ... + h_0 c_{n-1} = 0 \qquad & \text{Coeficiente grado } n-1 \\ h_{n-r}c_{r-2} + h_{n-r-1}c_{r-1} + ... + h_0 c_{n-2} = 0 \qquad & \text{Coeficiente grado } n-2 \\ \vdots & \\ h_{n-r}c_0 + h_{n-r-1}c_1 + ... + h_0 c_{n-r} = 0 \qquad & \text{Coeficiente grado } n-r \end{aligned}$$

Si escribimos matricialmente:

$$\begin{pmatrix} 0 & ... & 0 & h_{n-r} & ... & h_0 \\ 0 & ... & h_{n-r} & ... & h_0 & 0 \\ & & & & & \\ h_{n-r} & ... & h_0 & 0 & ... & 0 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ ... \\ c_{n-1} \end{pmatrix} = Hc^t = 0$$

Con H matriz de control del código.
($\Longleftarrow$): Partimos de la hipótesis de que $Hc^t = 0 \in \mathbb{F}_q^r$, esto significa que

$$\begin{pmatrix} h(x) \\ xh(x) \\ \vdots \\ x^{r-1}h(x) \end{pmatrix} c(x) = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Con la primera fila obtenemos que $h(x)c(x) = 0$, lo que implica que $c \in \mathcal{C}$ por (I).

□

Ejemplo 4.18. *Tomando de nuevo el ejemplo de $\mathcal{H}_3$ (Continuación Ejemplo 4.16), teniendo en cuenta el polinomio de control $h(x) = x^4 + x^2 + x + 1$. La matriz de control será:*

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

Haciendo uso de la Proposición 3.23, podemos comprobar que las matrices G del Ejemplo 4.13 y H del Ejemplo 4.18 cumplen una condición necesaria:

Ejemplo 4.19. *Sean G y H las matrices generadora y de control de $\mathcal{H}_3$:*

$$G \cdot H^t = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}^t = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Capítulo 5

Códigos BCH

Este capítulo está basado en el Capítulo 3, 7 y 9 de la bibliografía [6]. Como se comentó en la introducción de este trabajo, los códigos BCH forman una familia especialmente relevante de códigos cíclicos por su capacidad para corregir múltiples errores. Construiremos estos códigos a partir de los Códigos de Hamming. A no ser que se especifique lo contrario, a partir de este punto se trabajará en el cuerpo $\mathbb{F}_{2^m}$ con $m \in \mathbb{N}$.

$$\mathbb{F}_{2^m} \cong \mathbb{F}_2[x] \Big/ (f(x))$$

donde $f(x)$ corresponde a un polinomio irreducible de grado m con coeficientes en $\mathbb{F}_2$, que divide a $\Phi_n(x)$. (Véase la Observación 2.59).

Observación 5.1. *Para poder trabajar con estos códigos es imprescindible aclarar la relación entre $\mathbb{F}_2^m$ y $\mathbb{F}_{2^m}$. Haciendo uso de la Proposición 2.27 y la Observación 2.49, tenemos que $\mathbb{F}_{2^m}$ es un $\mathbb{F}_2$-espacio vectorial de dimensión m. Por lo tanto, podemos establecer una biyección entre los elementos de $\mathbb{F}_2^m$ y $\mathbb{F}_{2^m}$. Siendo β un elemento primitivo de $\mathbb{F}_{2^m}$, se tiene una base del espacio vectorial $\{1, \beta, ..., \beta^{m-1}\}$, y cualquier elemento $a \in \mathbb{F}_{2^m}$ se puede expresar como $a = a_0 \cdot 1 + a_1 \cdot \beta + a_2 \cdot \beta^2 + ... + a_{m-1} \cdot \beta^{m-1}$ con $a_0, a_1, ..., a_{m-1} \in \mathbb{F}_2$. Entonces la biyección se expresa*

$$\begin{array}{ccc} \mathbb{F}_{2^m} & \longleftrightarrow & \mathbb{F}_2^m \\ a & \longleftrightarrow & \begin{pmatrix} a_{m-1} \\ a_{m-2} \\ \vdots \\ a_0 \end{pmatrix} \end{array} \tag{5.1}$$

Por esta relación, la suma en $\mathbb{F}_2^m$ respeta la operación suma en $\mathbb{F}_{2^m}$ y viceversa.

Podemos referirnos a los códigos de Hamming como códigos BCH correctores de un error, aunque no se suelen denotar de esta manera. Hemos introducido en la Sección 3.2.2 cómo obtener una matriz de control H que represente un código de Hamming. También es posible obtener la matriz de control del código a partir de un elemento primitivo β del cuerpo.

Proposición 5.2. *Sea el código de Hamming $\mathcal{H}_m \subseteq \mathbb{F}_2^n$ con parámetros $[n = 2^m - 1, k = n - m, d = 3]$, y sea $\beta \in \mathbb{F}_{2^m}$ un elemento primitivo, la matriz $H = (1, \beta, ..., \beta^{2^m-2})$ es una matriz de control del código donde cada potencia de β (elementos del cuerpo $\mathbb{F}_{2^m}$) está representada por una m-tupla de $\mathbb{F}_2^m$, mediante la biyección en la Ecuación 5.1.*

Ejemplo 5.3. *Volvemos al código $\mathcal{H}_3$ (Véase el Ejemplo 2.50 y el Ejemplo 3.36). Sea $\beta \in \mathbb{F}_{2^3}$ una raíz primitiva de $x^7 - 1$ y $g(x) = x^3 + x + 1$ su polinomio mínimo, se cumple $g(\beta) = \beta^3 + \beta + 1 = 0$, por lo tanto $\beta^3 = \beta + 1$. $H = (1, \beta, ..., \beta^6)$ es la matriz de control del código. Haciendo uso de la biyección de la Ecuación 5.1 entre $\mathbb{F}_{2^3}$ y $\mathbb{F}_2^3$ esta es la matriz:*

$$H = (1, \beta, ..., \beta^6) = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Observación 5.4. *Dada $c \in \mathcal{H}_m$ una palabra del código, se cumplen esta serie de implicaciones:*

$$c = (c_0, ..., c_{n-1}) \in \mathcal{H}_m \iff Hc^t = 0 \in \mathbb{F}_2^m \iff \begin{pmatrix} 1 & \beta & ... & \beta^{n-1} \end{pmatrix} \cdot \begin{pmatrix} c_0 & c_1 & ... & c_{n-1} \end{pmatrix}^t = 0$$
$$\iff \sum_{i=0}^{n-1} (\beta)^i c_i = 0 \iff c(\beta) = 0$$

con $c(x) = c_0 + c_1 x + ... + c_{n-1} x^{n-1}$.

Proposición 5.5. *Dado $c \in \mathcal{H}_m \subseteq \mathbb{F}_2^n$ con $n = 2^m - 1$, y H matriz de control del código $\mathcal{H}_m$ dada por las potencias de β elemento primitivo de $\mathbb{F}_{2^m}$ con $g(x) \in \mathbb{F}_2[x]$ polinomio mínimo se cumple*

- *$g(x) | c(x)$ $\forall c \in \mathcal{H}_m$.*
- *$g(x)$ es el polinomio generador del código $\mathcal{H}_m$.*

Demostración. Se ha visto en la Observación anterior (5.4) que $\beta \in \mathbb{F}_{2^m}$ es raíz de cualquier palabra $c \in \mathcal{H}_m$. Siendo $g(x)$ el polinomio mínimo de β, por la Proposición 2.56 $g(x)$ divide a c. Por otro lado, si $g(x)$ es un polinomio que divide a cualquier palabra del código, esto significa que cualquier palabra $c \in \mathcal{H}_m$ se puede obtener del producto de $g(x)$ por otro polinomio $f(x)$, lo cual corresponde con la definición de polinomio generador del código. □

5.1. Ejemplo binario: Construcción de código BCH corrector de 2 errores a partir de $\mathcal{H}_3$

Para comprender de forma progresiva la construcción y el funcionamiento de los códigos BCH, comenzaremos analizando un caso particular: un código BCH diseñado para corregir dos errores. Partiremos de un código de Hamming binario, que permite corregir un único error, y a partir de su estructura construiremos el nuevo código BCH, ampliando así su capacidad de corrección.

En un código de Hamming con longitud $n = 2^m - 1$, se requieren m operaciones con la matriz de control (de tamaño $m \times n$) para detectar y corregir un error. De forma intuitiva, supondremos que corregir dos errores podría implicar duplicar estas operaciones, es decir, utilizar $2m$ condiciones lineales independientes.

El objetivo es construir una nueva matriz de control H', a partir de la matriz H del código de Hamming, que permita detectar y corregir hasta dos errores. Esta nueva matriz será la base para definir el código BCH correspondiente.

En concreto, dado que ya se ha utilizado a lo largo del trabajo, vamos a desarrollar esta idea para el Código de Hamming $\mathcal{H}_3$. La idea de este ejemplo se ha obtenido del recurso [6]

de la Bibliografía, donde se construye la misma idea a partir de $\mathcal{H}_4$, un caso más complejo. Por el momento contamos con el código $\mathcal{H}_3$ y la matriz de control que lo representa H, de tamaño 3×7. Ahora, queremos construir la matriz H'. Para corregir dos errores, necesitamos que se cumplan el doble de condiciones independientes, por lo tanto buscamos una matriz H' de tamaño 6×7. Las 3 primeras filas de H' corresponderán a la matriz H, dado que queremos mantener esas condiciones. Las nuevas tres filas las obtendremos mediante una aplicación $f:\mathbb{F}_2^3\to\mathbb{F}_2^3$, añadimos bajo cada columna de H otra columna dada por la aplicación f. El objetivo principal es encontrar esa f que satisfaga todas las condiciones que hemos puesto, para posteriormente construir una matriz H' con esta apariencia:

$$H'=\begin{pmatrix}1 & \beta & \beta^2 & ... & \beta^6\\ f(1) & f(\beta) & f(\beta^2) & ... & f(\beta^6)\end{pmatrix}$$

Supongamos que queremos transmitir un mensaje. Hasta ahora se ha corregido un error. A través del síndrome se puede localizar la posición del error.
Siendo $y=(y_0,...,y_6)\in\mathbb{F}_2^7$ la palabra recibida, con un error e_k en la posición $k\in\{0,1,2,3,4,5,6\}$, su síndrome será $S(y)=s(e_k)=He_k^t=\begin{pmatrix}\beta^k\\ f(\beta^k)\end{pmatrix}$. Es decir, si se comete un error, el síndrome es $\begin{pmatrix}i\\ f(i)\end{pmatrix}$ para $i\in\mathbb{F}_{2^3}\setminus\{0\}$. Teniendo esto en cuenta, veamos que ocurre si se producen dos errores en el proceso (los que queremos que corrija el nuevo código). Supongamos que estos errores suceden en las posiciones $k,l\in\{0,1,2,3,4,5,6\}$. Siendo H'_k la columna en la posición k de la matriz H' y $S=\begin{pmatrix}z_1\\ z_2\end{pmatrix}$ el síndrome:

$$H'_k=\begin{pmatrix}\beta^k\\ f(\beta^k)\end{pmatrix},H'_l=\begin{pmatrix}\beta^l\\ f(\beta^l)\end{pmatrix}\Longrightarrow S(y)=S(e_k+e_l)=H'_k+H'_l=\begin{pmatrix}\beta^k+\beta^l\\ f(\beta^k)+f(\beta^l)\end{pmatrix}=\begin{pmatrix}z_1\\ z_2\end{pmatrix}$$

con $\beta^k,\beta^l,z_1,z_2\in\mathbb{F}_2^3$, y la suma dada por la biyección en la Ecuación 5.1. Más genéricamente, queremos buscar una f, tal que la solución del sistema

$$\begin{pmatrix}z_1\\ z_2\end{pmatrix}=\begin{pmatrix}i+j\\ f(i)+f(j)\end{pmatrix}$$

exista y sea única $i\neq j$, $i,j\in\mathbb{F}_{2^3}\setminus\{0\}$. Buscamos la función más adecuada. Sea $a\in\mathbb{F}_8$, y s una constante:

- $f(a)=sa\Longrightarrow\begin{cases}i+j=z_1\\ s(i+j)=z_2\end{cases}$
  Obtenemos un sistema redundante, por ser dos ecuaciones dependientes.

- $f(a)=a^2\Longrightarrow\begin{cases}i+j=z_1\\ i^2+j^2=z_2\end{cases}$
  Sistema redundante en $\mathbb{F}_2$, porque $i^2+j^2=i^2+2ij+j^2=(i+j)^2$ en $\mathbb{F}_2$.

Descartadas estas opciones, vamos a demostrar que $f(a)=a^3$ para $a\in\mathbb{F}_8$, es la f adecuada que resuelve el sistema de forma única. Antes, para poder operar en el cuerpo, describimos sus elementos en tres formas distintas, que nos serán útiles para diferentes operaciones.

3-tupla	Elementos en $\mathbb{F}_8$	Potencia de β elemento primitivo de $\mathbb{F}_8$
(000)	0	0
(001)	1	1
(010)	β	β
(100)	β^2	β^2
(011)	$\beta+1$	β^3
(110)	$\beta^2+\beta$	β^4
(111)	$\beta^2+\beta+1$	β^5
(101)	β^2+1	β^6

Tabla 5.1: Elementos del cuerpo $\mathbb{F}_{2^3}$

Veamos un lema previo

Lema 5.6. *Sean $i,j \in \mathbb{F}_{2^3} \setminus \{0\}$, $(i+j)^3 = i^3+j^3 \iff i=j$*

Demostración. Desarrollamos las operaciones:
$(i+j)^3 = i^3+i^2j+ij^2+j^3 = i^3+j^3 \iff i^2j+ij^2 = 0 \iff ij(i+j) = 0 \iff i=j$ □

Teorema 5.7. *Sea $y \in \mathbb{F}_2^7$ la palabra recibida, y sea $e \in \mathbb{F}_2^7$ su vector error tal que $S(y) = S(e) = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \in \mathbb{F}_2^6$ con $z_1, z_2 \in \mathbb{F}_2^3$ y $w(e) \leq 2$, se verifica que:*

- *$z_1 = z_2 = 0 \iff e = 0 \in \mathbb{F}_2^7$ (No se producen errores en la transmisión).*
- *$z_1 \neq 0$, $z_2 = z_1^3 \iff w(e) = 1$ y el error se ha cometido en la posición $k \in \{0,1,2,3,4,5,6\}$ tal que $z_1 = \beta^k$.*
- *$z_1 \neq 0$, $z_2 \neq z_1^3 \iff w(e) = 2$ y el error se ha cometido en las posiciones $k,l \in \{0,1,2,3,4,5,6\}$, con $k \neq l$. Siendo $i = \beta^k$ y $j = \beta^l$, i,j son las raíces simples de*

$$\sigma_z(x) = x^2 + z_1 x + (\frac{z_2}{z_1} + z_1^2) \tag{5.2}$$

Demostración. Veamos en primer lugar que los tres casos que pueden ocurrir son disjuntos. Los diferentes pesos en función del error obtenido son $w(e) = 0$, $w(e) = 1$, $w(e) = 2$. El síndrome será, respectivamente

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \in \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} i \\ i^3 \end{pmatrix}_{i \neq 0}, \begin{pmatrix} i+j \\ i^3+j^3 \end{pmatrix}_{i \neq j; i,j \neq 0} \right\}$$

Veamos que los síndromes no coinciden nunca para los diferentes pesos.

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} i \\ i^3 \end{pmatrix} \iff i = 0$$

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} i+j \\ i^3+j^3 \end{pmatrix} \iff i+j = 0 \iff i = j$$

$$\begin{pmatrix} r \\ r^3 \end{pmatrix} = \begin{pmatrix} i+j \\ i^3+j^3 \end{pmatrix} \iff \begin{cases} r = i+j \\ r^3 = i^3+j^3 \end{cases} \iff r^3 = (i+j)^3 = i^3+j^3 \iff i = j$$

Veamos ahora que dados $z_1 \neq 0$ y $z_2 \neq z_1^3$ el sistema

$$\begin{cases} i + j = z_1 \\ i^3 + j^3 = z_2 \end{cases} \tag{5.3}$$

tiene solución única entre los $i, j \in \mathbb{F}_8 \setminus \{0\}$ con $i \neq j$, y que dicha solución son, a su vez, raíces del polinomio σ_z de grado 2.

En $\mathbb{F}_8$ un polinomio de grado 2 de la forma $p(x) = a + bx + x^2 \in \mathbb{F}_2[x]$ tiene una raíz doble sí y solo sí $b = 0$, esto significa que $z_1 = 0$. Alguna raíz es no nula si $a \neq 0$, es decir si $\frac{z_2}{z_1} + z_1^2 \neq 0$. Esto ocurre por ser $z_2 \neq z_1^3$. Con esto se concluye que $z_1 \neq 0$, $z_2 \neq z_1^3$ son condiciones necesarias y suficientes para que $\sigma_z(x)$ tenga raíces simples no nulas.

Veamos que i, j son las raíces de $\sigma_z(x)$ sí y solo sí i, j son solución del sistema 5.3.
i, j raíces de $\sigma_z(x) \iff (x + i)(x + j) = x^2 + (i + j)x + ij = \sigma_z(x)$
Antes de continuar, algunas operaciones a tener en cuenta:

I. $(i + j)\left(i^2 + ij + j^2\right) = i^3 + i^2 j + ij^2 + i^2 j + ij^2 + j^3 = i^3 + j^3$

II. $(i + j)^2 = i^2 + 2ij + j^2 = i^2 + j^2$

Entonces teniendo en cuenta que $z_1 = i + j$, se desarrolla

$$i^3 + j^3 = z_2 \overset{(I)}{\Longrightarrow} (i + j)(i^2 + ij + j^2) = z_2 \overset{(II)}{\Longrightarrow} z_1(z_1^2 + ij) = z_2 \Longrightarrow ij = \frac{z_2}{z_1} + z_1^2$$

Con esto se tiene que las posiciones de los errores k, l están determinadas de manera única por los elementos $i, j \in \mathbb{F}_8$. Entonces, i, j son las raíces del polinomio $\sigma_z(x)$, que nos indicarán las posiciones de los errores. □

Observación 5.8. *Se supone que en la transmisión se han cometido a lo sumo* 2 *errores, entonces el único caso en el que no se puede decodificar será cuando* $z_1 = 0$ *y* $z_2 \neq 0$*, en cuyo caso necesariamente por el teorema anterior estamos en la situación* $w(e) \geq 3$*. Entonces, se han cometido como mínimo* 3 *errores.*

Para buscar soluciones probaremos con los elementos posibles. Sea $\beta \in \mathbb{F}_{2^3}$ un elemento primitivo y raíz primitiva n-ésima de $x^7 - 1$. Sea $H = \begin{pmatrix} 1 & \beta & \beta^2 & \beta^3 & \beta^4 & \beta^5 & \beta^6 \end{pmatrix}$ matriz de control del código,

$$H' = \begin{pmatrix} 1 & \beta & \beta^2 & \beta^3 & \beta^4 & \beta^5 & \beta^6 \\ 1 & \beta^3 & \beta^6 & \beta^9 & \beta^{12} & \beta^{15} & \beta^{18} \end{pmatrix} = \begin{pmatrix} 1 & \beta & \beta^2 & \beta^3 & \beta^4 & \beta^5 & \beta^6 \\ 1 & \beta^3 & \beta^6 & \beta^2 & \beta^5 & \beta & \beta^4 \end{pmatrix}$$

con $\beta^7 = 1$. Desarrollamos los elementos en 3-tuplas con la tabla 5.1.

$$H' = \left(\begin{array}{ccccccc} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ \hline 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{array}\right)$$

Duplicar el tamaño de la matriz de control no afecta a la relación de la matriz con el código. Es decir, el papel que cumple una matriz de control es comprobar si una palabra está en el código o no. Veámoslo en un resultado

Proposición 5.9. *Sea H una matriz de control del código $\mathcal{H}_3$ y H' la matriz definida por f*

$$\textit{Para cualquier } c \in \mathcal{C},\ Hc^t = 0 \in \mathbb{F}_2^3 \iff H'c^t = 0 \in \mathbb{F}_2^6$$

Demostración. Siendo H' una ampliación de H tan solo hay que demostrar que el producto de la mitad inferior de H' por c es un vector de ceros. Recordemos que las nuevas columnas de H' no son aleatorias y vienen dadas por $f(i) = i^3$. Si se cumplía $Hc^t = 0$ es porque $c_0 + c_1\beta + ... + c_6\beta^6 = 0 \in \mathbb{F}_{2^3}$, gracias a que β es un elemento primitivo del cuerpo y raíz de $g(x)$. Como $f(\beta) = \beta^3 = \alpha + 1 \in \mathbb{F}_{2^3}$ también es un elemento primitivo del cuerpo, por lo tanto raíz de $x^7 - 1$, y se cumple $c_0 + c_1\beta^3 + ... + c_6(\beta^3)^6 = 0 \in \mathbb{F}_{2^3}$ (el producto con la mitad inferior de H').

La implicación $\Longleftarrow$ es trivial. Si el producto de la matriz grande H' con c es 0, entonces lo es con las tres primeras filas de H' que corresponden a H. □

Veamos como funciona esta decodificación por síndrome con algunos ejemplos. Antes vamos a aclarar algunas operaciones. En el caso de esta matriz H', la columna en la posición $k \in \{0, ..., 6\}$ corresponde con $\begin{pmatrix} \beta^k \\ \beta^{3k} \end{pmatrix}$. Entonces probamos las potencias de β correspondientes que pueden satisfacer el sistema 5.3:

$$\left(x + \beta^k\right)\left(x + \beta^l\right) = x^2 + x\left(\beta^k + \beta^l\right) + \beta^k\beta^l \Longrightarrow \begin{cases} \beta^k + \beta^l = z_1 \\ \beta^{k+l} = \frac{z_2}{z_1} + z_1^2 \end{cases} \tag{5.4}$$

Ejemplo 5.10 (Decodificación)**.** *Se trabaja en el ejemplo y matrices de esta sección. Supongamos que hemos obtenido una palabra $y \in \mathbb{F}_2^7$:*

- *Obtenemos un síndrome con $z_1 = (100) = \beta^2$ y $z_2 = (101) = \beta^6$. Nos encontramos en el segundo caso, donde $\beta^k = \beta^2$ y $\beta^{3k} = \beta^6$. Por lo tanto solo ocurre un error y se encuentra en la posición $k = 2$ de la palabra y.*

- *Obtenemos un síndrome con $z_1 = (001) = 1$, $z_2 = (101) = \beta^6$. Puesto que estamos en el tercer caso $w(e) = 2$, siendo $\frac{z_2}{z_1} + z_1^2 = (001) = \beta^6 + 1 = (101) + (001) = (100) = \beta^2$, se resuelve la ecuación 5.2*

  $$x^2 + x + \beta^2 = 0$$

  Usando el sistema 5.4 se tiene, para $k, l \in \{0, ..., 6\}$:

  $$\begin{cases} \beta^k + \beta^l = 1 \\ \beta^{k+l} = \beta^2 \Longrightarrow k + l \equiv 2 \pmod 7 \end{cases} \Longrightarrow (k, l) \in \{(0, 2), (3, 6), (4, 5)\}$$

  Descartamos el caso donde $k = 0$ y $l = 2$ porque $\beta^0 + \beta^2 = 1 + \beta^2 \neq 1$ ya que $\beta \in \mathbb{F}_8 \setminus \{0\}$.
  Probamos la primera ecuación con $k = 3$, $l = 6$.

  $$\beta^3 + \beta^6 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \beta^4 \neq 1$$

No es una opción válida.
Probamos ahora $k = 4, l = 5$, que cumplen

$$\beta^4 + \beta^5 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 1$$

Cumple las condiciones del sistema. Por lo tanto estamos ante las dos soluciones de la ecuación.

$$x^2 + x + \beta^2 = \left(x + \beta^4\right)\left(x + \beta^5\right) = 0$$

La solución nos dice que se localizan los errores en las posiciones $k = 4$ y $l = 5$, es decir en las columnas $\begin{pmatrix} \beta^4 \\ \beta^5 \end{pmatrix}$, $\begin{pmatrix} \beta^5 \\ \beta \end{pmatrix}$. que son las dos únicas columnas que suman dicho síndrome.

- *Supongamos ahora que obtenemos como síndrome $z_1 = 0$ y $z_2 = (111) = \beta^5$. Estaríamos ante el caso de la Observación 5.8. No es posible dividir entre $z_1 = 0$, así que no podemos generar el sistema 5.4, por lo tanto no podemos decodificar el mensaje y suponemos que se producen 3 o más errores.*

Observando la construcción de H', nada depende de $m \in \mathbb{N}$, por lo tanto podemos generalizar de esta manera la matriz de control de un código BCH corrector de 2 errores de longitud $n = 2^m - 1$, en el cuerpo $\mathbb{F}_{2^m}$.

Proposición 5.11 (Capítulo 3, p.88 de [6])**.** *Siendo β un elemento primitivo del cuerpo $\mathbb{F}_{2^m}$, la matriz de control de un código BCH $n = 2^m - 1$, corrector de dos errores, se construye como:*

$$H = \begin{pmatrix} 1 & \beta & \beta^2 & \beta^3 & \dots & \beta^{2^m-2} \\ 1 & \beta^3 & \beta^6 & \beta^9 & \dots & \beta^{3(2^m-2)} \end{pmatrix}$$

donde sustituimos cada potencia de β con su correspondiente m-tupla de $\mathbb{F}_2^m$.

Veamos un breve ejemplo de como se construiría la matriz a partir del código de Hamming $\mathcal{H}_4$.

Ejemplo 5.12 (Construcción de código BCH corrector de 2 errores a partir de $\mathcal{H}_4$)**.** *Tomamos el código con $m = 4$, $n = 15$. Siguiendo la Proposición 5.11, la matriz de control del código BCH corrector de 2 errores creado a partir del código de Hamming $\mathcal{H}_4$ sería:*

$$H' = \begin{pmatrix} 1 & \beta & \beta^2 & \beta^3 & \beta^4 & \beta^5 & \beta^6 & \beta^7 & \beta^8 & \beta^9 & \beta^{10} & \beta^{11} & \beta^{12} & \beta^{13} & \beta^{14} \\ 1 & \beta^3 & \beta^6 & \beta^9 & \beta^{12} & 1 & \beta^3 & \beta^6 & \beta^9 & \beta^{12} & 1 & \beta^3 & \beta^6 & \beta^9 & \beta^{12} \end{pmatrix} =$$

$$\left(\begin{array}{cccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right)$$

Se puede observar que, a diferencia del ejemplo de $\mathcal{H}_3$, no todas las potencias del elemento primitivo $\beta \in \mathbb{F}_{16}$ aparecen en la segunda sección de filas de la matriz H. Esto ocurre porque no todas las potencias de β son elementos primitivos del cuerpo. Por ejemplo, el elemento β^3 tiene orden 5 porque $(\beta^3)^5 = 1$ (véase el Ejemplo 2.55). Se puede consultar este ejemplo desarrollado en el Capítulo 3 de [6].

5.2. Códigos BCH correctores de t errores con alfabeto $\mathbb{F}_p$

La matriz de control determina un código de manera única, entonces podemos hablar de un código $\mathcal{C}$ hablando de su matriz H. Continuando con la idea de la sección anterior, ampliaremos el tamaño de la matriz de control, en relación con los errores que queremos corregir. Vamos a definir formalmente los códigos BCH y demostrar algunas de sus propiedades. También se generalizan algunos resultados trabajados en el apartado anterior.

Notación. *Siendo $\beta \in \mathbb{F}_{p^m}$, $n = p^m - 1$ raíz primitiva n-ésima de la unidad, denotaremos por $M_b(x) \in \mathbb{F}_p[x]$ al polinomio mínimo de β^b,con $b \in \{1, 2, ..., n-1\}$.*

Definición 5.13. *Sea $\mathcal{C} \subseteq \mathbb{F}_p^n$ un código cíclico de longitud n, será un **código BCH** con distancia de diseño $\delta \in \mathbb{N}$, si para algún entero $b \geq 0$:*

$$g(x) = mcm\{M_b(x), M_{b+1}(x), ..., M_{b+\delta-2}(x)\}.$$

es su polinomio generador. Es decir, $g(x)$ es el polinomio mónico de menor grado en $\mathbb{F}_p[x]$, con ceros las siguientes potencias $\beta^b, \beta^{b+1}, ..., \beta^{b+\delta-2}$, todas ellas elementos de $\mathbb{F}_{p^m}$.

Proposición 5.14. *Para un b fijo, los códigos BCH están **anidados**, es decir, el código BCH con distancia de diseño δ_1 contiene al código BCH con distancia de diseño δ_2 si y solo si $\delta_1 \leq \delta_2$.*

Demostración. Llamaremos BCH_1 al código BCH con distancia de diseño δ_1. Llamaremos a su polinomio generador $g_1(x)$. Denotamos con BCH_2 al código con distancia de diseño δ_2. $g_2(x)$ será su polinomio generador. Por definición de polinomio generador, $c \in \text{BCH}_2$ si y solo sí $g_2|c$, de la misma forma en BCH_1.
Describimos los polinomios como $g_1(x) = mcm\{M_b(x), ..., M_{b+\delta_1-2}(x)\}$ y $g_2(x) = mcm\{M_b(x), ..., M_{b+\delta_2-2}(x)\}$. Tal y como están descritos, $\delta_1 \leq \delta_2 \iff g_1|g_2$. Demostramos ahora de forma equivalente que $\text{BCH}_2 \subseteq \text{BCH}_1 \iff g_1|g_2$.

- ($\Longleftarrow$): Sea $c \in \text{BCH}_2$ se tiene que $g_2|c$. Por hipótesis $g_1|g_2$, entonces $g_1|c$, por lo tanto $c \in \text{BCH}_1$. Se concluye que $\text{BCH}_2 \subseteq \text{BCH}_1$.
- ($\Longrightarrow$): Tomamos $c(x) := 1 \cdot g_2(x) \in \text{BCH}_2$. Como $\text{BCH}_2 \subseteq \text{BCH}_1$, $g_2 \in \text{BCH}_1$, por lo tanto $g_1|g_2$.

□

Existen casos especiales de códigos BCH.

Definición 5.15 (Tipos de códigos BCH)**.** *Destacan algunos tipos de códigos BCH:*

- *Si $b = 1$, el código se llamará **Código BCH en sentido estricto**.*
- *Si $p \neq 2$, $n = p - 1$, estos son llamados **Códigos de Reed-Solomon**.*

Entre los códigos BCH destacan los **Reed-Solomon** porque son especialmente eficaces cuando los errores afectan a varios elementos consecutivos dentro de una palabra (errores en ráfaga), lo que los hace muy adecuados para canales en los que los errores tienden a concentrarse.

A diferencia de otros códigos BCH que operan sobre alfabetos binarios, los Reed-Solomon trabajan sobre cuerpos finitos más grandes (extensiones de cuerpos de Galois), lo que les permite tratar bloques de información como elementos del cuerpo.

Esta característica les proporciona una gran capacidad de corrección, especialmente útil cuando los errores no están aislados, sino agrupados, como ocurre en muchas aplicaciones prácticas de transmisión y almacenamiento de datos.

Teorema 5.16. *Sea $\mathcal{C} \subseteq \mathbb{F}_p^n$ un código BCH con distancia de diseño δ y con polinomio generador $g(x)$. Bajo las condiciones anteriores para $\beta \in \mathbb{F}_{p^m}$, H es matriz de control del código*

$$H = \begin{pmatrix} 1 & \beta^b & \dots & \beta^{(n-1)b} \\ \vdots & & & \vdots \\ 1 & \beta^{b+\delta-2} & \dots & \beta^{(n-1)(b+\delta-2)} \end{pmatrix}$$

formada por $\delta - 1$ filas de m-tuplas. Por lo tanto, de tamaño $m(\delta-1) \times n$.

Demostración. H es matriz del código si y solo sí para todo $c \in \mathcal{C}$ cumple $Hc^t = 0$. Veámoslo. Se tiene que las potencias de β son raíces de $g(x)$, entonces son raíces de $c(x)$ para todo $c \in \mathcal{C}$. Entonces para cada potencia de β tenemos:

$$\sum_{i=0}^{n-1} c_i \left(\beta^b\right)^i = 0 \iff c_0 + c_1\beta^b + \cdots + c_{n-1}\beta^{b(n-1)} = 0$$

$$\sum_{i=0}^{n-1} c_i \left(\beta^{b+1}\right)^i = 0 \iff c_0 + c_1\beta^{b+1} + \cdots c_{n-1}\beta^{(b+1)(n-1)} = 0$$

$$\vdots$$

$$\sum_{i=0}^{n-1} c_i \left(\beta^{b+\delta-2}\right)^i = 0 \iff c_0 + c_1\beta^{b+\delta-2} + \cdots + \ c_{n-1}\beta^{(b+\delta-2)(n-1)} = 0$$

Si escribimos en forma matricial, sea $c = (c_0, c_1, \dots, c_{n-1}) \in \mathcal{C}$:

$$\begin{pmatrix} 1 & \beta^b & \beta^{2b} & \dots & \beta^{b(n-1)} \\ 1 & \beta^{b+1} & \beta^{2(b+1)} & \dots & \beta^{(n-1)(b+1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \beta^{b+\delta-2} & \beta^{2(b+\delta-2)} & \dots & \beta^{(n-1)(b+\delta-2))} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \iff Hc^t = 0 \in \mathbb{F}_p^{m(\delta-1)}$$

Veamos ahora que, si $Hc^t = 0$, entonces $c \in \mathcal{C}$. Siendo H la matriz del enunciado y $c \in \mathbb{F}_p^n$, si $Hc^t = 0$, entonces se cumple $c(\beta^{b+i}) = 0$ con $i \in \{0, 1, ..., \delta-2\}$. Esto implica que los polinomios mínimos de cada potencia de β, $M_{b+i}(x)$ para $i \in \{0, 1, ..., \delta-2\}$ dividen a $c(x)$. Siendo $g(x) = mcm\{M_b(x), M_{b+1}(x), ..., M_{b+\delta-2}(x)\}$ el polinomio generador de $\mathcal{C}$, $g(x)$ divide a $c(x)$, por lo tanto $c \in \mathcal{C}$. □

Teorema 5.17. *Sea $\mathcal{C} \subseteq \mathbb{F}_p^n$ un código BCH con distancia de diseño δ, la dimensión del código será como mínimo $n - m(\delta - 1)$*

Demostración. Por la Definición 3.21, tenemos que el rango de la matriz H es $n-k$. Entonces si el rango de H es menor o igual que las filas de la matriz, $n-k \leq m(\delta-1)$. Por lo tanto $k \geq n - m(\delta-1)$. □

Teorema 5.18 (Teorema 6.6.2 de [9])**.** *La distancia mínima de un código BCH con distancia de diseño δ es como mínimo δ.*

Demostración. Sea la matriz de control H descrita en el Teorema 5.16, teniendo en cuenta que cada entrada β^i es una $m-tupla$ sobre $\mathbb{F}_p^m \cong \mathbb{F}_{p^m}$ su tamaño es $m(\delta-1)\times n$. Tengamos en cuenta que $c \in BCH \iff Hc^t = 0$.
No todas las $m(\delta-1)$ filas de H tienen por qué ser independientes. Consideramos $\delta-1$ columnas de H, las encabezadas por las potencias $\beta^{i_1 b}, \beta^{i_2 b}, ..., \beta^{i_{\delta-1} b}$. Calculamos el determinante de Vandermonde de esta submatriz de H cuadrada (Determinante de Vandermonde p.116 de [6]):

$$\det \begin{pmatrix} \left(\beta^b\right)^{i_1} & ... & \left(\beta^b\right)^{i_{\delta-1}} \\ \left(\beta^{b+1}\right)^{i_1} & ... & \left(\beta^{b+1}\right)^{i_{\delta-1}} \\ \vdots & & \vdots \\ \left(\beta^{b+\delta-2}\right)^{i_1} & ... & \left(\beta^{b+\delta-2}\right)^{i_{\delta-1}} \end{pmatrix} = \beta^{(i_1+...+i_{\delta-1})b} \cdot \det \begin{pmatrix} 1 & ... & 1 \\ \beta^{i_1} & ... & \beta^{i_{\delta-1}} \\ \vdots & & \vdots \\ \beta^{(\delta-2)i_1} & ... & \beta^{(\delta-2)i_{\delta-1}} \end{pmatrix} =$$

$$= \beta^{(i_1+i_2+...+i_{\delta-1})b} \prod_{1\leq k<l\leq \delta-1} \left(\beta^{i_k} - \beta^{i_l}\right) \neq 0 \Longrightarrow \det(\hat{H}) \neq 0$$

puesto que $\beta^{i_k} \neq \beta^{i_l}$ mientras $k \neq l$ y β sea una raíz primitiva n-ésima de la unidad. Por lo tanto cualesquiera $\delta-1$ columnas de H son linealmente independientes. Teniendo en cuenta la Proposición 3.27, $d(BCH) \geq \delta$.

□

Proposición 5.19. *Para un código BCH corrector de t errores, la distancia de diseño será $2t+1$, de manera que $t \leq \left\lfloor \frac{d-1}{2} \right\rfloor$ y el código corrije t errores (Teorema 3.15).*

5.2.1. Particularidades de los códigos BCH con $p = 2$ y $b = 1$

Proposición 5.20 (Propiedad cuando $p = 2$)**.** *Sea $\mathcal{C}$ un código BCH **binario**, y $M_b(x)$ polinomio mínimo de β^b, se cumple que*

$$M_{2b}(x) = M_b(x) \in \mathbb{F}_2[x]$$

Demostración. Sea $f(x) \in \mathbb{F}_2[x]$, $\gamma \in \mathbb{F}_2^m$, se cumple $f(\gamma^2) = f(\gamma)^2$. En particular, para β y su polinomio mínimo, β^b y β^{2b} tienen el mismo polinomio mínimo. □

La proposición anterior se cumple para cualquier b, pero será realmente útil aplicarla cuando $b = 1$, ya que nos permite obviar parte de los polinomios, como se aprecia en la siguiente observación

Observación 5.21. *La Proposición 5.20 sugiere que bien sea la distancia de diseño $\delta = 2t+1$ o bien $\delta = 2t$, el polinomio generador $g(x)$ será el mismo.*

- *Si $\delta = 2t+1$, entonces $g(x) = mcm\{M_1(x), M_2(x), ..., M_{2t-1}(x), M_{2t}(x)\} = mcm\{M_1(x), M_3(x), ..., M_{2t-1}(x)\}$*

- *Si* $\delta = 2t$ *entonces* $g(x) = mcm\{M_1(x), M_2(x), ..., M_{2t-1}(x)\} = mcm\{M_1(x), M_3(x), ..., M_{2t-1}(x)\}$

Observación 5.22. *Sea* $\mathcal{C}$ *un código BCH binario, teniendo en cuenta la Proposición 5.20, y habiendo definido su matriz de control en el Teorema 5.16, podemos tomar la matriz siguiente como matriz de control del código* $\mathcal{C}$*. Esta resulta de quitar a la matriz del Teorema 5.16 las filas alternas de m-tuplas.*

$$H = \begin{pmatrix} 1 & \beta & \beta^2 & ... & \beta^{n-1} \\ 1 & \beta^3 & \beta^6 & ... & \beta^{3(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \beta^{\delta-2} & \beta^{2(\delta-2)} & ... & \beta^{(n-1)(\delta-2)} \end{pmatrix}_{\frac{m(\delta-1)}{2} \times n}$$

El tamaño de esta matriz son, por como se ha construido, la mitad de filas que la matriz H del Teorema 5.16, por lo tanto $\frac{m(\delta-1)}{2}$ *filas. Mantiene las n columnas.*

Ejemplo 5.23. *Consideramos* $n = 2^3 - 1$ *con* $m = 3$ *y* $p = 2$*. Tomamos* $b = 1$*. Trabajamos en el cuerpo* $\mathbb{F}_8$ *y* β *primitivo con* $\beta^7 = 1$*. Consideramos un código* $\mathcal{C}$ *con* $\delta = 3$*. Su* $g(x)$ *tiene por raíces* β*,* β^2*. Como* $m = 3$*, el polinomio mínimo de* β *tiene grado* 3*. Se tiene* $x^7 + 1 = (x+1)(x^3 + x^2 + 1)(x^3 + x + 1) \in \mathbb{F}_2[x]$ *y* β *puede ser raíz de* $(x^3 + x^2 + 1)$ *o de* $(x^3 + x + 1)$*. Suponemos que* β *es raíz del primero, entonces ese polinomio será el generador de* $\mathcal{C}$ *el código BCH buscado.*
Si tomamos ahora $\delta = 4$*, para hallar el código BCH* $\hat{\mathcal{C}}$*, con* β *raíz y su polinomio mínimo* $M_1(x) = (x^3 + x^2 + 1)$*, el polinomio generador del código será* $g(x) = \text{mcm}\{M_1(x), M_3(x)\}$*, siendo* $M_3(x) = x^3 + x + 1$ *el polinomio mínimo de* β^3*. Entonces el polinomio generador de* $\hat{\mathcal{C}}$ *es* $(x^3 + x^2 + 1)(x^3 + x + 1)$*.*

Observando que ahora H tiene a lo sumo rango mt, obtenemos el siguiente resultado.

Corolario 5.24. *Para cualquier entero* $m \geq 3$ *y* $t < 2^m - 1$*, existe un código BCH binario corrector de t errores con los siguientes parámetros*

- $n = 2^m - 1$
- $n - k \leq mt$ *siendo* $n - k$ *la codimensión del código.*
- $d \geq 2t + 1$ *siendo d la distancia mínima del código y* $\delta = 2t + 1$ *la distancia de diseño.*

que determinan un código BCH corrector de t errores.

Ejemplo 5.25 (Códigos BCH primitivos en $\mathbb{F}_2^7$)**.** *Vamos a organizar en una tabla todos los códigos BCH que tenemos con los parámetros del Ejemplo 5.23.*
Sea $m = 3$ *y* $t < 4$*, y* β *elemento primitivo de* $\mathbb{F}_8$*. Vamos a conocer todos los códigos BCH bajo estas características que corrijan diferentes t errores.*

t	$g(x)$	$n-k$	mt	d	δ
1	$x^3 + x + 1$	3	3	3	3
1	$x^3 + x^2 + 1$	3	3	3	3
2	$(x^3 + x + 1)(x^3 + x^2 + 1)$	6	6	7	5
3	$(x^3 + x + 1)(x^3 + x^2 + 1)$	6	9	7	7

Tabla 5.2: Códigos BCH binarios primitivos con $n = 7$

Veamos para la cantidad de errores que corrigen la obtención de los datos.

- *Si $t = 1$: código $\mathcal{H}_3$ ya visto.*
- *Si $t = 2$: Visto en el Ejemplo 5.23. El polinomio generador $g(x)$ es el polinomio mínimo en $\mathbb{F}_2[x]$ con raíces $\beta, \beta^2, \beta^3, \beta^4$. Las raíces de $x^3 + x + 1$ son β^2, β^4, β, por otro lado $x^3 + x^2 + 1$ tiene como raíces $\beta^3, \beta^5, \beta^6$.*
- *Si $t = 3$: En este caso $g(x)$ es el polinomio mínimo en $\mathbb{F}_2[x]$ con raíces β^i con $i = 1, ..., 6$. De nuevo tenemos $g(x) = (x^3 + x^2 + 1)(x^3 + x + 1)$.*

Téngase en cuenta que este ejemplo se encuentra desarrollado a modo ilustrativo, pero que los últimos dos casos, donde la distancia $d = 7 = n$, el único código lineal que cumple estas características es el código trivial formado por las palabras (0000000) *y* (1111111).

Conocimos en el capítulo introductorio de códigos la distancia mínima d de un código, y ahora en BCH la distancia de diseño δ, que establece cuántos errores se espera que el código corrija. Son conceptos que parecen relacionarse pero que no representan lo mismo. En la última proposición en concreto se dice $d \geq \delta$. Veamos algunos resultados relacionados.

Lema 5.26 (Cota de Hamming. Teorema 6, p.19 de [6])**.** *Un código binario corrector de t errores y longitud n, que contiene M palabras debe satisfacer*

$$M\left(1 + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{t}\right) \leq 2^n$$

Teorema 5.27 (Farr)**.** *Sea $\mathcal{C}$ un código BCH binario con $n = 2^m - 1$ y $\delta = 2t + 1$. Si se cumple*

$$\sum_{i=0}^{t+1} \binom{2^m - 1}{i} > 2^{mt}$$

entonces $d(\mathcal{C}) = 2t + 1 = \delta$

Demostración. El sumatorio

$$\sum_{i=0}^{t+1} \binom{2^m - 1}{i}$$

representa el número total de vectores binarios de longitud $n = 2^m - 1$ con un peso menor que $t+1$. Se supone por reducción al absurdo que $d(\mathcal{C}) > \delta$, tomamos entonces $d(\mathcal{C}) = 2t+3$. Se puede demostrar que d es impar (Corolario 17 del Capitulo 8 de [6]). Véase la Observación 5.21 para observar que dos distancias de diseño consecutivas generan el mismo código, por lo tanto se tiene que si $d \geq 2t$, entonces $d \geq 2t + 1$.
Se cumple que $n - k \leq mt$ (Proposición 5.24). Sabiendo que hay 2^k palabras en el código (Definición 3.17) y aplicando el Lema 5.26:

$$2^k \left(\sum_{i=0}^{t+1} \binom{2^m - 1}{i}\right) \leq 2^n \Longrightarrow \sum_{i=0}^{t+1} \binom{2^m - 1}{i} \leq 2^{n-k} \leq 2^{mt}$$

Donde llegamos a una contradicción con la hipótesis del enunciado, donde tenemos que el sumatorio es mayor que 2^{mt}. Por lo tanto $d(\mathcal{C}) = \delta$. □

Teorema 5.28 (Peterson)**.** *Sea $\mathcal{C}$ un código BCH binario de longitud n y distancia de diseño δ, si δ divide a n entonces $d(\mathcal{C}) = \delta$.*

Demostración. Tomamos a $n = s\delta$ como un múltiplo de δ. Sea β una raíz primitiva n-ésima de la unidad se cumple que $\beta^i \neq 1$ para $i < \delta$. Sea

$$(x^n - 1) = (x^s - 1)(1 + x^s + x^{2s} + ... + x^{(\delta-1)s})$$

Las potencias $\beta, \beta^2, ..., \beta^{\delta-1}$ no son raíces de $x^s - 1$, por lo tanto son raíces de $(1 + x^s + ... + x^{(\delta-1)s})$. Esto significa que $1 + x^s + ... + x^{(\delta-1)s} \in \mathcal{C}$ y tiene peso δ. Por lo tanto, teniendo en cuenta que $d(\mathcal{C}) \geq \delta$ y que existe una palabra en el código con peso δ, entonces $d(\mathcal{C}) = \delta$. □

5.3. Construcción y decodificación de un código BCH binario con $b = 1$

Ahora que se cuenta con una idea global de los códigos BCH, veamos la construcción de un código BCH y un método de decodificación para estos códigos. Vamos a esquematizar la construcción de un código BCH contenido en $\mathbb{F}_2^n$ y distancia de diseño δ, para un $b \in \mathbb{N}$.

Algoritmo (Construcción de un código BCH binario)**.** *Sea $n = 2^m - 1$, $b \in \mathbb{N}$ y $\delta = 2t + 1$ con $t \geq 1$.*

I. *Factorizamos $x^n - 1$ en $\mathbb{F}_2[x]$ en polinomios irreducibles.*

II. *Escogemos un factor de $\Phi_n(x)$ que será el polinomio mínimo de β, siendo $\beta \in \mathbb{F}_{2^m}$ una raíz primitiva n-ésima de la unidad.*

III. *Calculamos los polinomios mínimos $M_{b+i}(x)$ sobre $\mathbb{F}_2[x]$ de β^{b+i}, con $i = 1, 2, ..., \delta - 2$.*

IV. *Calculamos el polinomio generador del código como*

$$g(x) = mcm\{M_b(x), M_{b+1}(x), ..., M_{b+\delta-2}(x)\}$$

Supongamos que se ha producido ya la codificación y transmisión de un mensaje. Se toma $b = 1$ a partir de aquí.

Algoritmo (Decodificación de un código BCH binario)**.** *Sea $\mathcal{C}$ un código BCH binario corrector de t errores con parámetros $n = 2^m - 1, k, d$ y distancia de diseño $\delta = 2t + 1$, se recibe la palabra $y \in \mathbb{F}_2^n$. Ordenamos la decodificación de la palabra y en tres pasos:*

I. *Cálculo del síndrome de y.*

- *Si $S(y) = 0 \in \mathbb{F}_2^{m(\delta-1)}$ entonces $y \in \mathcal{C}$. Hemos terminado.*
- *Si $S(y) = S(e) \neq 0$ con e el vector error, entonces $y \notin \mathcal{C}$. Pasamos al paso II.*

II. *Búsqueda de $\sigma_z(x)$ polinomio localizador de errores.*

III. *Búsqueda de raíces de $\sigma_z(x)$*

Se menciona el polinomio localizador de errores $\sigma_z(x)$ que ya ha sido utilizado por primera vez en la Sección 5.1. Se define formalmente en la Sección 5.3.2. Antes de comenzar a describir el proceso, un poco de notación

Notación. *Sea $y = c + e \in \mathbb{F}_2^n$, con $c \in \mathcal{C}$, $S(y) = S(e)$, se tiene $y(\beta) = e(\beta)$. Si $w = w(e)$ tenemos los elementos $e_{i_1}, e_{i_2}, ..., e_{i_w}$ distintos de 0. Las coordenadas i_j corresponden a las coordenadas de e erróneas.*

5.3.1. Paso I: Cálculo del Síndrome

Sea H la matriz de control de un código BCH $\mathcal{C} \subseteq \mathbb{F}_2^n$, con $b = 1$, $y \in \mathbb{F}_2^n$ la palabra recibida y $S(y) = S(e)$ el síndrome de y:

$$S(y) = Hy^t = \begin{pmatrix} 1 & \beta & \beta^2 & ... & \beta^{n-1} \\ 1 & \beta^2 & \beta^4 & ... & \beta^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \beta^{\delta-1} & \beta^{2(\delta-1)} & ... & \beta^{(n-1)(\delta-1)} \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} y(\beta) \\ y(\beta^2) \\ \vdots \\ y(\beta^{\delta-1}) \end{pmatrix} = \begin{pmatrix} e(\beta) \\ e(\beta^2) \\ \vdots \\ e(\beta^{\delta-1}) \end{pmatrix}$$

Siendo $w = w(e)$ el peso de la palabra del error cometido en y, se puede observar que

$$e(x) := \sum_{i=0}^{n-1} e_i x^i = \sum_{j=1}^{w} e_{i_j} x^{i_j}$$

Además con un código binario, como son los casos en los que estamos trabajando, $e(x) = \sum_{j=1}^{w} x^{i_j}$, por ello:

$$e(\beta^i) = \sum_{j=1}^{w} \beta^{i_j} \tag{5.5}$$

Si el código es binario no es necesario tomar la matriz H anterior completa, se puede tomar la matriz de control vista en la Observación 5.22.

También podemos calcular cada $e(\beta^i)$ dividiendo $e(x)$ entre el polinomio mínimo $M_i(x)$ de β^i, es decir

$$e(x) = q(x) \cdot M_i(x) + r(x) \text{ con } deg(r(x)) < deg(M_i(x))$$

Entonces $e(\beta^i) = r(\beta^i)$. De esta manera calculamos fácilmente $e(\beta^i)$ con $i \in \{1, 2, ..., \delta - 1\}$. Además en un código binario $e(\beta^{2i}) = e(\beta^i)^2$.

5.3.2. Búsqueda de $\sigma_z(x)$ polinomio localizador de error.

Definición 5.29 (Polinomio localizador de error)**.** *Sea $e \in \mathbb{F}_2^n$, con $\hat{y} = (y_{i_1}, y_{i_2}, ..., y_{i_w})$ sus elementos no nulos, sea $\beta \in \mathbb{F}_{2^m}$ una raíz primitiva n-ésima de la unidad, definimos el polinomio localizador de errores de e como*

$$\sigma_z(x) = \prod_{j=1}^{w} (1 - \beta^{i_j} x) = \sum_{j=0}^{w} \sigma_j x^j \in \mathbb{F}_{2^m}[x] \tag{5.6}$$

siendo los errores i_j tal que $\sigma_z(\beta^{-i_j}) = 0$.

Notación. *De esta manera, desarrollando la expresión del polinomio 5.6 tenemos que los*

coeficientes del polinomio $\sigma_z(x) = \sigma_0 + \sigma_1 x + \sigma_2 x^2 + ... + \sigma_w x^w$ *son*

$$\begin{aligned}
\sigma_0 &= 1 \\
\sigma_1 &= -\left(\beta^{i_1} + \cdots + \beta^{i_w}\right) \\
\sigma_2 &= \beta^{i_1}\beta^{i_2} + \beta^{i_1}\beta^{i_3} + \cdots + \beta^{i_{w-1}}\beta^{i_w} \\
&\vdots \\
\sigma_w &= (-1)^w \beta_1^i \ldots \beta_w^i
\end{aligned} \tag{5.7}$$

Una vez se ha calculado del síndrome de e, teniendo en cuenta las identidades obtenidas en las Fórmulas 5.7, la Expresión 5.5 y las identidades de Newton se obtienen las siguientes ecuaciones que relacionan el polinomio $\sigma_z(x)$ y el síndrome de e (obtenido en el capítulo 8 de la bibliografía [6]):

$$\begin{aligned}
e(\beta) + \sigma_1 &= 0 \\
e(\beta^2) + \sigma_1 e(\beta) + 2\sigma_2 &= 0 \\
e(\beta^3) + e(\beta^2)\sigma_1 + e(\beta)\sigma_2 + 3\sigma_3 &= 0 \\
&\vdots \\
e(\beta^w) + \sigma_1 e(\beta^{w-1}) + \cdots + \sigma_{w-1} e(\beta) + w\sigma_w &= 0 \\
\text{con } i > w, \quad e(\beta^i) + \sigma_1 e(\beta^{i-1}) + \cdots + \sigma_w e(\beta^{i-w}) &= 0
\end{aligned} \tag{5.8}$$

Vista esta relación, podemos encontrar el polinomio $\sigma_z(x)$ de distintas formas. Describimos el Modelo de Peterson, que trata de resolver el sistema anterior. Es el método más intuitivo, sin embargo para un t grande no es recomendable.

Modelo de Peterson (para un código BCH binario)

En 1960, W. Wesley Peterson presentó un método para decodificar códigos BCH. Su contribución fue fundamental porque propuso una manera de determinar el polinomio localizador de errores, que como hemos visto, es clave para identificar las posiciones donde han ocurrido errores. Aunque el Método de Peterson fue un gran paso adelante, posteriormente fue refinado y generalizado por Daniel Gorenstein y Neal Zierler en 1961, quienes extendieron la aplicabilidad del método y formalizaron muchas de las bases. Como hemos centrado el capítulo en Códigos binarios, veremos la versión más simple del Método. Tengamos en cuenta la relación que hemos descrito entre los $e(\beta)$ y los coeficientes de σ, por las Ecuaciones 5.8. En códigos binarios trabajamos en un cuerpo con característica 2, y se cumple $e(\beta^{2i}) = e(\beta^i)^2$, por lo tanto las ecuaciones sufren alguna modificación.

Ejemplo 5.30. *Veamos algún caso de modificaciones en las Ecuaciones 5.8 a modo ilustrativo.*

- $e(\beta^2)+\sigma_1 e(\beta)+2\sigma_2 = 0 \Longrightarrow e(\beta)^2 - e(\beta)e(\beta) = 0 \Longrightarrow 0 = 0$. *Esta ecuación desaparece. Para obtener el coeficiente* σ_2 *y* σ_3 *tendremos las dos siguientes ecuaciones.*
- $e(\beta^3) + e(\beta^2)\sigma_1 + e(\beta)\sigma_2 + 3\sigma_3 = 0 \Longrightarrow e(\beta^3) + e(\beta)^2 e(\beta) + e(\beta)\sigma_2 + \sigma_3 = 0$
- $e(\beta^4)+\sigma_1 e(\beta^3)+\sigma_2 e(\beta^2)+\sigma_3 e(\beta)+4\sigma_4 = 0 \Longrightarrow e(\beta)^4+\sigma_1 e(\beta^3)+\sigma_2 e(\beta)^2+\sigma_3 e(\beta) = 0$

De las dos últimas ecuaciones se tiene como incógnitas σ_2 *y* σ_3*, por lo tanto se pueden obtener de forma sencilla.*

Si extendemos esto, vamos a obtener los coeficientes de $\sigma_z(x)$ mediante el sistema:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ e\left(\beta^2\right) & e\left(\beta\right) & 1 & 0 & 0 & \cdots & 0 \\ e\left(\beta^4\right) & e\left(\beta^3\right) & e\left(\beta^2\right) & e\left(\beta\right) & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ e\left(\beta^{2w-2}\right) & e\left(\beta^{2w-3}\right) & e\left(\beta^{2w-4}\right) & e\left(\beta^{2w-5}\right) & e\left(\beta^{2w-6}\right) & \cdots & e\left(\beta^{w-1}\right) \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \vdots \\ \sigma_w \end{pmatrix} = \begin{pmatrix} e\left(\beta\right) \\ e\left(\beta^3\right) \\ e\left(\beta^5\right) \\ \vdots \\ e\left(\beta^{2w-1}\right) \end{pmatrix}$$

Para conocer la compatibilidad del sistema y resolverlo, se necesita conocer el valor de w. En el siguiente teorema suponemos que $v \leq t$ y tantearemos hasta encontrar el peso w.

Teorema 5.31 (Teorema de Peterson. Capitulo 9, p.274 de [6])**.** *Sea la matriz de tamaño $v \times v$, con $t = \frac{\delta-1}{2}$ y $v \leq t$,*

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ e\left(\beta^2\right) & e\left(\beta\right) & 1 & 0 & 0 & \cdots & 0 \\ e\left(\beta^4\right) & e\left(\beta^3\right) & e\left(\beta^2\right) & e\left(\beta\right) & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ e\left(\beta^{2v-2}\right) & e\left(\beta^{2v-3}\right) & e\left(\beta^{2v-4}\right) & e\left(\beta^{2v-5}\right) & e\left(\beta^{2v-6}\right) & \cdots & e\left(\beta^{v-1}\right) \end{pmatrix}_{v\times v}$$

La matriz es no singular si $w = v$ o $w = v - 1$ y singular si $w < v - 1$.

Observación 5.32. *Se recuerda al lector que una matriz es singular si es cuadrada con determinante nulo. Una matriz no singular o regular es una matriz con inversa, es decir con determinante distinto de* 0*.*

Haciendo uso de este teorema desarrollamos un algoritmo iterativo para obtener los coeficientes del polinomio localizador de errores, $\sigma_z(x)$, de un código BCH binario con distancia de diseño $\delta = 2t + 1$.

Asumimos que ocurren w errores con $w \leq t$.

- Si partimos de que ocurren $w = t$ o $w = t - 1$ errores, existe solución del sistema, se resuelve y se obtiene el polinomio $\sigma_z(x)$.
- Si ocurren menos de $t - 1$ errores, las ecuaciones no tienen solución. En este caso asumimos que ocurren $w = t - 2$ errores y repetimos el proceso hasta encontrar una solución.

Dada la cantidad de operaciones que se deben realizar, no se recomienda este método para t grande. Si t supera los 4 errores, se utiliza el Algoritmo de Berlekamp-Massey. (p.246, 275 de [6]).

5.3.3. Búsqueda de raíces de $\sigma_z(x)$

Una vez tenemos el polinomio localizador de errores, debemos encontrar sus raíces para conocer las posiciones de los errores. Si observamos la definición de $\sigma_z(x)$ (Definición 5.29), las raíces del polinomio son los inversos de β^{i_j}, es decir $\frac{1}{\beta^{i_j}}$. Si el grado del polinomio es 1 o 2 es sencillo encontrar sus raíces. Para un polinomio de mayor grado recurrimos al algoritmo **Búsqueda de Chien**. Es un método sencillo que consiste en probar cada potencia de β.

Teorema 5.33 (Búsqueda de Chien)**.** *Sea β un elemento primitivo de $\mathbb{F}_{2^m}$ y raíz del polinomio generador del código BCH binario $\mathcal{C}$, sea $\sigma_z(x)$ el polinomio localizador de errores de la palabra $y \in \mathbb{F}_2^n$ recibida:*

$$\textit{Hay un error en la posición } k \textit{ si y solo sí } \sigma_z(\beta^{-k}) = 0$$

Una vez encontramos las raíces de $\sigma_z(x)$, conocemos las posiciones de error en la palabra y recibida y por lo tanto podemos corregir el mensaje.

5.3.4. Decodificación de otros códigos BCH

Dado que esta sección y el trabajo en general ha estado centrado en Códigos binarios no daremos demasiado protagonismo a otros códigos de mayor dificultad pero comentaremos brevemente los pasos a seguir en otros casos.

En la **decodificación de Códigos BCH no binarios** el cálculo del síndrome ya está descrito en la Sección 5.3.1. En este caso simplemente se usa la matriz H completa, sin eliminar las filas alternas. Para la búsqueda del polinomio $\sigma_z(x)$ las ecuaciones obtenidas de las identidades de Newton (Ecuaciones 5.8) ya no son útiles, en su lugar se utiliza la recurrencia

$$e(\beta^{j+w}) + \sigma_1 e(\beta^{j+w-1}) + ... + \sigma_w e(\beta^j) = 0 \text{ para } j = 1, 2, ..., w$$

obtenida en la página 244 de la bibliografía [6]. Tras obtener $\sigma_z(x)$ se busca un nuevo polinomio para los casos no binarios, **el polinomio evaluador de error**. En códigos binarios, el alfabeto tiene dos elementos: 0, 1. Una vez se conoce la posición del error, para corregir la palabra simplemente se cambia un elemento por el otro. Cuando el alfabeto tiene más elementos, se debe implementar un paso más en la decodificación. Este nuevo polinomio nos da las herramientas para conocer el elemento correcto. En concreto, obtenemos el vector de error e. Por lo tanto si la palabra recibida es $y = c + e$, es sencillo encontrar la palabra correcta. El **Algoritmo de Berlekamp-Massey** es otro método útil en estos casos, ya que calcula el polinomio localizador de error y el polinomio evaluador de error simultáneamente.

Capítulo 6

Aplicaciones

En este capítulo se presentan dos aplicaciones prácticas de los códigos BCH en diferentes ámbitos. La primera, tratada con mayor detalle, es su uso en el estándar DVB-T2, un sistema de transmisión de televisión digital adoptado en países como España. Los códigos BCH permiten corregir errores en la señal, mejorando su calidad incluso en condiciones adversas. La segunda aplicación se sitúa en el ámbito médico, donde estos códigos aseguran la integridad de los datos en sistemas de telemedicina. Su capacidad de corrección resulta clave para transmitir información sensible con precisión.

Ambas aplicaciones ilustran la importancia de los códigos BCH en la mejora de la fiabilidad en sistemas de comunicación complejos.

6.1. Presencia de Códigos BCH en el estándar DVB-T2

El 14 de febrero de 2024 se produjo en España un "apagón" de la TDT en el que cesaron las emisiones en definición estándar (SDTV), consolidando la alta definición (HD) como el estándar mínimo para las emisiones en televisión. Además, se prevé que en 2025 se realice una nueva reordenación de la TDT en España, que incluirá la adopción del estándar DVB-T2. Esto llevará a que los usuarios que aún no hayan cambiado los televisores más antiguos, deban hacerlo este año.

Antes de conocer el papel de los códigos BCH en este asunto, conozcamos algunas definiciones mínimas en el campo de la telecomunicación para poder movernos con mayor soltura en este tema (obtenido en [10] y [8]).

- Estándar de TV: conjunto de especificaciones técnicas que definen cómo se deben transmitir, recibir y decodificar las señales de televisión, para que dispositivos de diferentes fabricantes sean compatibles. El estándar DVB-T es un sistema de transmisión de televisión digital desarrollado para reemplazar las transmisiones analógicas.

- Capa física de un estándar: es el medio de comunicación físico o las tecnologías para transmitir datos a través de ese medio. Por ejemplo: cables de fibra óptica, cableado de cobre, aire, etc.

- DVB-T2 (Digital Video Broadcasting - Second Generation Terrestrial): es la extensión (nueva versión) del estándar de televisión DVB-T ideado para la transmisión de difusión en TDT. En concreto, DVB-T es la tecnología que hace funcionar el TDT.

DVB-T2 cuenta con un 30 % más de capacidad para transmitir en alta definición que DVB-T, lo que se traduce en un mayor flujo de datos.

Una vez se han presentado los datos básicos de telecomunicación que debemos conocer, vamos a descubrir donde se encuentran situados los códigos BCH.
En la comunicación existe un transmisor (o emisor) y un receptor. El siguiente esquema representa el proceso de transmisión en la capa física del estándar DVB-T2. En el receptor se recorre el proceso de manera inversa, deshaciendo los cambios, sustituyendo las codificaciones por decodificaciones. Como se aprecia en el esquema, los códigos BCH aparecen en la

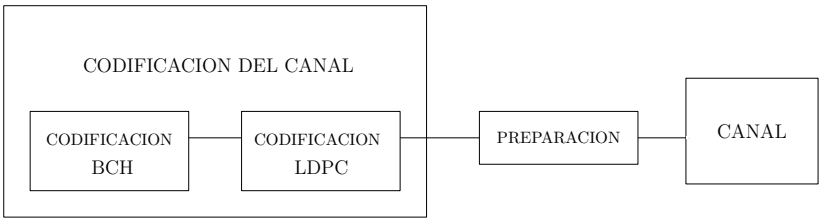

Figura 6.1: Esquema simple de la capa física de DVB-T2

codificación de canal, junto a otros códigos llamados códigos LDPC (Low Density Parity Check). Son códigos lineales que pueden ser descritos con una matriz de control H hueca (matriz con la mayoría de sus elementos 0). Este tipo de codificación combinando varios códigos en el canal se conoce como **codificación en cascada**. El objetivo principal de esta codificación en cascada es minimizar la tasa de error de bit (BER) en la recepción del mensaje, incluso cuando existe mucho ruido o interferencias. El proceso simplificado sería el siguiente. Inicialmente se codifican los bits de información con un código BCH. Después la palabra codificada, se codifica con un código LDPC. Los parámetros n y k representan la longitud y dimensión del código respectivamente. En cada parámetro viene especificado como subíndice a qué código representan. El siguiente esquema ilustra los cambios de fuente entre las codificaciones.

$$\mathbb{F}_2^{k_{BCH}} \xrightarrow{\text{BCH}} \mathbb{F}_2^{n_{BCH}} = \mathbb{F}_2^{k_{LDPC}} \xrightarrow{\text{LDPC}} \mathbb{F}_2^{n_{LDPC}}$$

El vector de información codificada tiene el siguiente aspecto, con las longitudes indicadas.

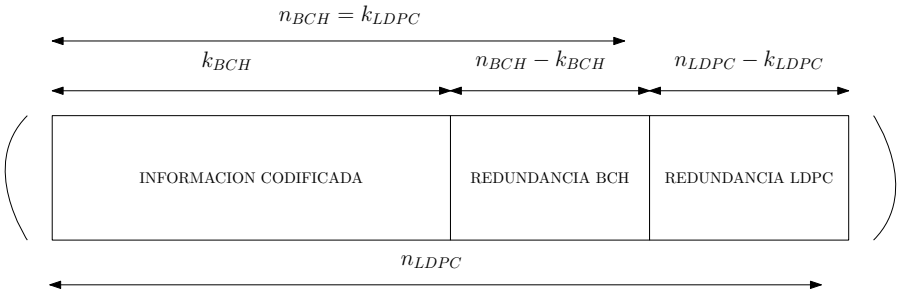

Figura 6.2: Aspecto de la información codificada

Se transmite la información por el canal. Cuando se recibe la información se realiza el proceso inverso. Se decodifica con LDPC, y el resultado se decodifica con BCH.
En concreto, se realiza en este orden por lo siguiente. Los códigos LDPC son muy eficaces en la corrección de la mayoría de errores. Sin embargo, aparece un problema que denominamos

suelo de error, aquí aparece la importancia de BCH. Aunque exista buena señal en la transmisión, existen algunos errores residuales que forman ese suelo de error que LDPC no es capaz de corregir. Son de un orden muy bajo (10^{-7}), pero en transmisiones, por ejemplo, de televisión pueden causar interrupciones o pixelaciones. Aquí BCH actúa como limpiador final de esos errores, evitando que la curva de la tasa BER que relaciona la señal y el ruido, se detenga en el suelo de error y corrija los últimos errores.

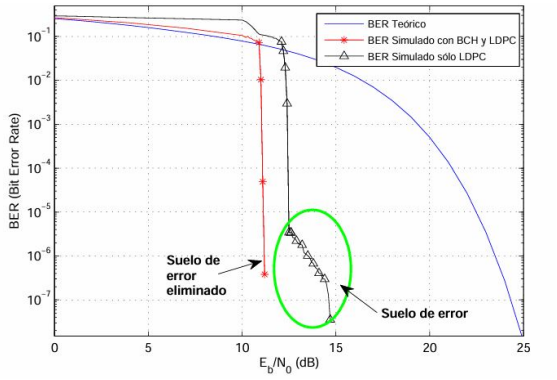

Figura 6.3: Relación entre tasa BER, transmisión con y sin código BCH. Obtenido en [5]

En el estándar DVB-T2 se utilizan códigos BCH binarios de longitud $n = 2^{16} - 1 = 65535$ sobre $\mathbb{F}_{16}$. Sin embargo, se emplea una técnica de acortamiento (shortening) para reducir la longitud de las palabras código, adaptándolas al tamaño requerido por el estándar. En particular, se reduce la longitud a 64800. Existen 6 posibles combinaciones de parámetros de la codificación para trabajar en el estándar, en función del ratio de LDPC ($\frac{k_{LDPC}}{n_{LDPC}}$) y el número de errores que corrige BCH.

Ratio LDPC	K_{bch} (información)	K_{ldpc} N_{bch}	**corrección errores BCH**	$N_{bch} - K_{bch}$	**bloque codificado LDPC** N_{ldpc}
1/2	32208	32400	12	192	64800
3/5	38688	38880	12	192	64800
2/3	43040	43200	10	160	64800
3/4	48408	48600	12	192	64800
4/5	51648	51840	12	192	64800
5/6	53840	54000	10	160	64800

Figura 6.4: Posibles codificaciones en DVB-T2. Obtenido en [5]

La elección del tipo de codificación influirá en la calidad de la transmisión.

Codificación BCH

El valor de m utilizado es 16, y por lo tanto $n = 2^{16} - 1$. Sin embargo, el valor n utilizado para la codificación BCH varía entre $n_{BCH} = 32400$ y $n_{BCH} = 5400$.

Los códigos BCH, como hemos visto a lo largo del trabajo, quedan determinados por su polinomio generador. Este $g(x)$ viene especificado en el estándar. Siendo t el número de errores que puede corregir el código BCH, en este proceso $10 \leq t \leq 12$.

$M_1(x)$	$1 + x^2 + x^5 + x^{16}$
$M_2(x)$	$1 + x + x^4 + x^5 + x^6 + x^8 + x^{16}$
$M_3(x)$	$1 + x^2 + x^3 + x^4 + x^7 + x^8 + x^{10} + x^{11} + x^{16}$
$M_4(x)$	$1 + x^3 + x^4 + x^7 + x^{11} + x^{12} + x^{16}$
$M_5(x)$	$1 + x^2 + x^5 + x^6 + x^{10} + x^{12} + x^{16}$
$M_6(x)$	$1 + x + x^5 + x^7 + x^9 + x^{12} + x^{16}$
$M_7(x)$	$1 + x^2 + x^6 + x^8 + x^{10} + x^{13} + x^{15} + x^{16}$
$M_8(x)$	$1 + x^3 + x^5 + x^7 + x^8 + x^{11} + x^{13} + x^{16}$
$M_9(x)$	$1 + x^4 + x^6 + x^7 + x^9 + x^{11} + x^{14} + x^{16}$
$M_{10}(x)$	$1 + x^3 + x^5 + x^8 + x^{12} + x^{13} + x^{14} + x^{16}$
$M_{11}(x)$	$1 + x^2 + x^5 + x^6 + x^9 + x^{11} + x^{12} + x^{16}$
$M_{12}(x)$	$1 + x^2 + x^5 + x^6 + x^9 + x^{11} + x^{12} + x^{16}$

Tabla 6.1: Polinomios mínimos para BCH

Si aplicamos el Corolario 5.24, bajo estas características, sabemos que existe un código que corrige t errores, con el grado del polinomio generador menor o igual a mt. Se calcula el polinomio generador, siendo $\delta = 2t + 1$.

$$g(x) = mcm\{M_1(x), M_2(x), ..., M_{\delta-1}(x)\}$$

Dado el polinomio generador, ya podemos codificar la información. Esta será la entrada de LDPC.

Decodificación

La decodificación se hace mediante el cálculo del síndrome de la palabra recibida. Si la palabra recibida no pertenece al código utilizaremos el Algoritmo de Berlekamp. Como máximo se corrigen 12 errores, una cifra insignificante en una palabra de tal longitud. En realidad BCH corregirá los errores que LDPC no ha conseguido corregir, los que se encuentran en el suelo de error.

6.2. Presencia de Códigos BCH en sistemas de telemedicina

Como se expone en el artículo [12], los códigos BCH desempeñan un papel fundamental en la transmisión de datos médicos a través de medios de comunicación ruidosos. En este ámbito, garantizar la integridad y fiabilidad de la información transmitida es esencial. Los datos transmitidos incluyen imágenes diagnósticas, historiales clínicos y otros tipos de información médica crítica. Errores en la transmisión de estos materiales pueden desencadenar en malos diagnósticos lo que puede acarrear graves consecuencias. Veamos donde aparecen aquí los códigos BCH.

Este artículo se centra en electrocardiogramas. Vamos a ver como se trabaja y almacena esta información. Una señal electrocardiográfica se capta y se traduce en conjuntos de cuarenta muestras de 11 bits, almacenando 500 muestras por segundo. Un ciclo de señal electrocargiográfico dura 0.96 s y produce 12 vectores de información. Pueden ser dos tipos de vectores: de alta energía (pueden corresponder con una actividad intensa, picos o ruido) y de baja energía (corresponden a una señal débil o a regiones de reposo). Los vectores de alta

energía necesitan mayor precisión en la toma de datos y su almacenamiento. En cambio, los de baja energía tienen menos bits de información y se pueden comprimir con menor detalle.

A pesar de que los canales tienen sistemas de protección contra errores en la transmisión, en los extremos (emisor y receptor) existe una probabilidad de error residual, que en ocasiones es tolerable y en otras debe tratarse. Este artículo [12] propone incrementar la seguridad con un mecanismo adicional de corrección de errores y otro de detección, y además aportar información sobre la validez de la información recibida.

Se utiliza un código BCH concatenado con un CRC (Código de Redundancia Cíclica) para detectar errores residuales en los grupos de 12 vectores. Este esquema de codificación ofrece una probabilidad de 0,999985 de evitar errores residuales. En la etapa interior donde se encuentra la codificación BCH, se utiliza un código BCH corrector de $t = 2$ errores con parámetros $n = 31, k = 21$. Se utiliza además una técnica de intercalado (o *interleaving*). Esta técnica consiste en reorganizar los bits antes de transmitirlos para dispersar errores en ráfaga. Se agrupan los datos en una matriz de tamaño 13×31, donde hay 13 (12 vectores de BCH + 1 vector de CRC) vectores verticales con una longitud de 31 elementos (longitud de BCH). Por ejemplo, si hay una ráfaga de errores de 62 bits consecutivos, el *interleaving* los reparte en los 13 vectores. En una situación ideal, ningún vector recibe más de 2 bits erróneos. Como BCH corrige hasta 2 errores, se garantiza la corrección completa de la ráfaga.

Capítulo 7

Bibliografía

[1] J. M. Aguado. *Introducción a las Teorías de la Información y la Comunicación.* Universidad de Murcia, 2004. Disponible en `https://www.um.es/tic/Txtguia/Introduccion%20a%20las%20Teorias%20de%20la%20Informa%20(20)/TIC%20texto%20guia%20completo.pdf`.

[2] Luis Atala. Ejemplo de código qr, 2008. Disponible en: `https://es.wikipedia.org/wiki/Archivo:C%C3%B3digo_QR_Ej%C3%A9mplo_de_Estructura.svg` [Accedido el 10 de septiembre de 2024].

[3] R. W. Hamming. *Error detecting and error correcting codes.* 1950. Bell System Technical Journal, Volumen 29, Páginas 147-160. 1950.

[4] T. W. Hungerford. *Algebra*, chapter 3,5. Springer, 2003.

[5] Omar Ahmad San José. *Análisis y simulación de la capa física del estándar DVB-T2*, Julio 2009. Proyecto final de carrera, Tutor: Ana García Armada.

[6] F. J. MacWilliams. *The Theory of Error-Correcting Codes.* North-Holland Mathematical Library, 2006.

[7] J. M. V. Hernández Morales. *Programa sin distancias: Codificación y transmisión de la información.* uned., 2011. `https://canal.uned.es/video/5a6f7b2eb1111ff1168b4b98` [Accedido el 2 de junio de 2024].

[8] Iván Ramírez. Qué es dvb-t2 o tdt2 y qué ventajas tiene sobre la tdt normal. `https://www.xataka.com/basics/que-dvb-t2-tdt2-que-ventajas-tiene-tdt-normal` [Accedido el 28 de enero de 2025].

[9] J. H. van Lint. *Introduction to Coding Theory.* Springer, 1999.

[10] Wikipedia. Dvb-t2, 2025. `https://es.wikipedia.org/wiki/DVB-T2` [Accedido el 17 de abril de 2025].

[11] Wikipedia contributors. Teoría de códigos — wikipedia, la enciclopedia libre, 2024. `https://www.wikiwand.com/es/Teor%C3%ADa_de_c%C3%B3digos` [Accedido el 12 de junio de 2025].

[12] L. E. Aparicio Pico y P. J. Arco Ríos. *Código concatenado para la capa de aplicación para uso en telemedicina.* 25(1), 2004.