



**I
N
A
O
E**

**Códigos LDPC para confiabilidad y alta
eficiencia de transferencia de
información por canal corrupto por
ruido**

Ing. Loth Matheus Barba Maza

Tesis sometida como requisito parcial
para obtener el grado de

**Maestro en Ciencias con Especialidad
en Electrónica**

Supervisada por
Dra. Gordana Jovanovic Dolecek

Instituto Nacional de Astrofísica, Óptica y
Electrónica

Santa Maria de Tonanzintla, Puebla, México
Agosto de 2018



Dedicatoria

A mi mamá y mis hermanos

Resumen

EN este trabajo se presenta un modelo de comunicaciones digitales utilizando codificación de canal. Se utilizan Bancos de Filtros Sobremuestreados por su característica de agregar redundancia de manera controlada y Códigos de Verificación de Paridad de Baja Densidad (LDPC, *Low-Density Parity-Check*).

Se presenta primeramente una breve introducción sobre los sistemas de comunicación y la importancia de la codificación para la reducción de errores en la transmisión de información. Posteriormente se hace una descripción de los diferentes tipos de codificación. Después se hace la introducción de los Bancos de Filtros y sus propiedades y como diseñarlos.

También se muestra que se pueden usar los Bancos de Filtros para codificación y se presentan resultados de simulación que comprueban el desempeño superior del modelo propuesto en comparación con un método reportado en la literatura

Abstract

IN this thesis a novel digital communication model is presented. The model uses Oversampled Filter Banks (OFB) due its' characteristic of added redundance in a controlled way, and Low-Density Parity-Check codes (LDPC).

First, a brief drescription of a communication system is presented and the importance of coding to reduce the errors in the information transmission. Next a description of some different coding types is presented. After that, an introduction of what is an Oversampled Filter Banks (OFB), its properties and how to design it.

Also,it is explained how OFB can been used for channel coding and finally, the proposed model is presented with simulations that probe the performance of the proposed model, compared with others find in literature.

Contenido

Dedicatoria	II
Resumen	III
Abstract	IV
1. Introducción	1
1.1. Introducción a teoría de la información	1
1.2. Motivaciones	4
1.3. Objetivos	5
1.3.1. Objetivo general	5
1.3.2. Objetivos específicos	5
1.4. Organización de la tesis	5
2. Codificación	7
2.1. Códigos lineales	8
2.2. Códigos cíclicos	9
2.2.1. BCH Coding	10
2.3. Códigos convolucionales	11
2.3.1. Viterbi Algorithm [1]	14
2.4. Códigos de Comprobación de Paridad de Baja Densidad	15
3. Bancos de filtros	20
3.1. Bancos de Filtros Maximamente Decimados	20
3.2. Errores en Bancos de Filtros	21

3.2.1. Relaciones de entrada-salida para Bancos de filtros Maxima- mente Decimados	21
3.2.2. Reconstrucción perfecta	22
3.3. Representación polifásica	24
3.4. Bancos de Filtros Sobremuestreados	27
4. Diseño de Bancos de filtros sobremuestreados para codificación	28
4.1. Modulación por la Transformada Discreta del Coseno	28
4.2. Polinomios de comprobación de paridad	30
4.3. Creación del síndrome	31
4.4. Diseño del filtro prototipo	33
5. Bancos de Filtros para codificación LDPC	34
5.1. Simulación	36
5.2. Resultados	39
6. Conclusiones y trabajo a futuro	42
6.1. Trabajo a futuro	42
Bibliografía	43

Introducción

1.1. Introducción a teoría de la información

Los sistemas de comunicación digital modernos están en constante desarrollo y mejoramiento, debido a las necesidades del mundo actual, por lo cual cada vez se demanda un incremento mayor en la eficiencia y confiabilidad de la transmisión, lo cual también se traduce en una mayor complejidad del sistema, razón por la cual debe haber siempre un equilibrio entre las dos partes, para obtener un sistema confiable y sencillo de implementar.

En un principio, los esfuerzos se centraban únicamente en evitar el deterioro de la señal transmitida, teniendo una buena relación señal a ruido (SNR, *signal to noise ratio*), pero a partir del trabajo de Claude Shannon en 1948, llamado *A Mathematical Theory of Communication*, hubo un cambio, ya que demostró que se podían transmitir señales con una confiabilidad arbitrariamente alta en un canal corrupto por ruido, siempre y cuando se tuviera un adecuado sistema de codificación.

Para ello, Shannon tomó la fórmula de la entropía (1.1), que ya era usada en mecánica estadística, y la usó para medir la cantidad de información proporcionada por una fuente en un determinado sistema de comunicación, y auxiliado por este concepto definió el concepto de capacidad de canal C , como la cantidad máxima de información que se puede transmitir de manera confiable a través de un canal.

$$H(X) = - \sum_{i=1}^n p(x_i) \log p(x_i). \quad (1.1)$$

Con esto pudo demostrar que para cualquier canal era posible desarrollar un sistema de codificación que permitiera enviar información con una frecuencia de error tan pequeña como se deseara a través de él a una tasa C .

$$C = \max(H(x) - H_y(x)). \quad (1.2)$$

Este trabajo fue tan importante que sentó las bases de lo que ahora conocemos como teoría de la información, al introducir la definición de capacidad de canal y relacionar los conceptos de entropía e información. Y pese a que no mostró métodos prácticos para el diseño de los mencionados esquemas de codificación, fue el punto de partida para la creación de esquemas de codificación que aproximarán C para diversos canales.

Ya con esto en mente, en 1950, Richard Hamming desarrolló el primer código que era capaz de corregir errores, y en 1955, Elias propuso los códigos lineales, un tipo de códigos de corrección de error (ECC, Error Correcting code), capaz de aproximar C para diversos tipos de canales, lo cual ayudó para que tiempo después aparecieran más códigos, como los Turbo códigos, los cuales demostraron tener un desempeño muy bueno, cercano al límite de Shannon, pero con una complejidad modesta.

Las características a tomar en cuenta para decir que un código es bueno son

- Que sea fácil de codificar
- Que sea fácil de decodificar
- Que la distancia Hamming del código sea grande, la cual es el número mínimo de posiciones en que dos palabras de código (*codeword*) cualesquiera son diferentes.

1. Introducción

En 1963, Gallager propuso un tipo especial de códigos lineales en su tesis doctoral. Los Códigos de Verificación de Paridad de Baja Densidad (LDPC, *Low-Density Parity-Check*). Sin embargo estos trabajos fueron ignorados por más de 30 años, esto debido a la complejidad, lo cual hacía impráctica su implementación con la tecnología de la época.

Fue hasta 1995 cuando Mackay demostró el excelente desempeño de los códigos LDPC, mostró que bajo un esquema de decodificación iterativo del tipo *message-passing* y con longitudes de código suficientemente grandes, se aproximaban bastante al límite de Shannon.

Fue en ese momento que se empezó a hacer más investigación de la comunidad en el área de codificación, generó trabajos respecto a su construcción, técnicas de codificación, decodificación y análisis de desempeño.

Uno de los resultados más sobresalientes es el de un umbral de desempeño a solo 0.0045 dB del límite de Shannon. También se demostró que tenían superioridad respecto a los turbo códigos, los cuales eran los mejores códigos en términos de desempeño hasta entonces, sin mencionar que los códigos LDPC tienen la ventaja de requerir cálculos matemáticos de menor orden, además de que presentan la posibilidad de paralelizar completamente su algoritmo de decodificación.

Una palabra de código w está integrada por un bloque de bits de información y un bloque de bits de verificación. Las reglas de combinaciones lineales para los bits de verificación pueden ser representados de manera conveniente en una matriz de verificación de paridad (PCM, *Parity-Check Matrix*)

La PCM asociada a un código LDPC está compuesta principalmente por pocas entradas 1, y es precisamente esta característica la que le da el nombre de baja densidad, y aunque no exista un número que defina baja densidad, para matrices grandes y aplicaciones reales se toma un porcentaje de aproximadamente de 1.

La PCM representa un sistema lineal de ecuaciones homogéneas, las cuales son

conocidas como ecuaciones de verificación de paridad, el conjunto de soluciones a este sistema es el conjunto de palabras de código que forman el código, y esto nos da que la matriz de palabras multiplicado por la transpuesta de la PCM debe ser igual a 0.

Para un canal discreto sin memoria, es posible transmitir mensajes con una muy pequeña probabilidad de error si el radio de comunicación es igual o menor que la capacidad de canal C , esto es

$$r \leq C. \quad (1.3)$$

La razón de comunicación es definido como:

$$r = \frac{k}{n}, \quad (1.4)$$

donde k es la longitud del mensaje y n es la longitud de los bits transmitidos

1.2. Motivaciones

Se decidió investigar sobre códigos de comprobación de paridad de baja densidad debido a que han demostrado ser una herramienta muy eficaz para los sistemas digitales de comunicación de hoy en día, que están en constante expansión, lo que demanda una gran confiabilidad y eficiencia en la transmisión y almacenamiento de la información.

Su excelente desempeño ha hecho que sean considerados para diversas aplicaciones, como comunicaciones bajo el agua, en dispositivos de grabación magnética, discos duros y comunicaciones ópticas., incluso han sido adoptados por estándares como Ethernet a 10 Gigabits.

Esto los posiciona para sustituir a los turbo códigos en ciertas aplicaciones, pero para que eso suceda, es necesario desarrollar construcciones que aseguren un excelente desempeño frente al ruido Gaussiano y el ruido impulsivo, presentes en los sistemas de comunicaciones modernos, pero sin que esto signifique una elevada complejidad de implementación.

1. Introducción

1.3. Objetivos

Los objetivos de este trabajo se dividen en dos, el objetivo general y los objetivos específicos.

1.3.1. Objetivo general

Proponer un sistema de codificación LDPC usando Bancos de Filtros Sobremuestreados para reducir el Error Cuadrático Medio producido por Error de tipo Gaussiano y por ruido impulsivo.

1.3.2. Objetivos específicos

- Investigar los diferentes tipos de Códigos LDPC.
- Entender el proceso de codificación y decodificación de los códigos LDPC.
- Estudiar las construcciones de Bancos de filtros e identificar aquellas con un grado de complejidad de implementación menor.

1.4. Organización de la tesis

En el siguiente capítulo se presenta una introducción a la clasificación de los métodos de codificación más usados para alcanzar lo más posible la capacidad de canal establecida por Claude Shannon. Además se da la definición formal de los códigos LDPC.

Posteriormente en el Capítulo 3 se presentan los conceptos de los Bancos de Filtros. Dos tipos de Bancos de Filtros son introducidos, los Bancos de Filtros Máximamente decimados y los Bancos de Filtros sobremuestreados. Además es presentada la representación polifásica de un Banco de Filtros y el diseño de un Banco de Filtros modulado con reconstrucción perfecta (PR, *Perfect reconstruction*).

En el Capítulo 4 se presenta una descripción de cómo los Bancos de Filtros Sobremuestreados son usados para codificación y cómo el síndrome puede ser usado para

corrección de errores, así como el diseño del filtro prototipo y la matriz polinomial de Comprobación de Paridad.

En el Capítulo 5 se desarrolla y presenta la propuesta de un sistema de codificación para corrección de errores usando Bancos de Filtros Sobremuestreados y codificación LDPC.

Finalmente el Capítulo 6 muestra las conclusiones derivadas de este trabajo

2

Codificación

En este capítulo se presentan técnicas de codificación de control de errores que proporcionan diferentes métodos de implementar el teorema de Shannon de codificación de canal. Cada técnica utiliza un codificador de canal en el transmisor y un decodificador en el receptor

La tarea con la que se topa un diseñador de un sistema digital de comunicaciones es la de proveer una forma efectiva de transmitir un mensaje de un punto a otro a una velocidad y nivel de calidad que sea aceptable por el usuario que la recibe. Una opción práctica y disponible para el desarrollador para tener una calidad en los datos transmitida aceptable es usar codificación de control de errores.

Las técnicas de codificación de control de errores mostradas a continuación incluyen las siguientes clases importantes de códigos:

- Códigos lineales

- Códigos cíclicos

- Códigos convolucionales

- Códigos de Verificación de Paridad de Baja Densidad

2.1. Códigos lineales

Un código es lineal si la suma de dos cualesquiera palabras de código genera una tercera palabra de código. Esta suma es realizada aritmeticamente en modulo-2, en un código lineal (n, k) , donde n es la longitud de la palabra codificada y k es la longitud del mensaje, $n - k$ bits son generados desde los bits del mensaje de acuerdo a las reglas de codificación que determine la estructura matemática. Estos $n - k$ bits son llamados bits de paridad [1].

El sistema de ecuaciones que define los bits de paridad son:

$$\mathbf{b} = \mathbf{mP}, \quad (2.1)$$

donde \mathbf{m} es el vector que contiene el mensaje, \mathbf{b} es el vector de paridad, y \mathbf{P} es la matriz de coeficientes:

$$\mathbf{P} = \begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,n-k-1} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,n-k-1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}, \quad (2.2)$$

Esta matriz es generada de tal manera que sus filas sean linealmente independientes, y con eso obtenemos la palabra de código \mathbf{c} , que la definimos como:

$$\mathbf{c} = [c_0, c_1, \dots, c_{n-1}] = [\mathbf{b} : \mathbf{m}]. \quad (2.3)$$

Y reemplazando (2.1) en (2.3) se puede notar que:

$$\mathbf{c} = \mathbf{m}[\mathbf{P} : \mathbf{I}_k], \quad (2.4)$$

donde \mathbf{I}_k es la matriz identidad y la matriz generadora \mathbf{G} es definida como:

$$\mathbf{G} = [\mathbf{P} : \mathbf{I}_k]. \quad (2.5)$$

2. Codificación

La palabra codificadora puede ser definida como:

$$\mathbf{c} = \mathbf{mG}. \quad (2.6)$$

La matriz generadora y la matriz de comprobación de paridad son básicas para la operación de un código lineal.

Durante el proceso de decodificación, definimos \mathbf{r} como el vector resultante después de mandar la palabra codificada \mathbf{c} a través del canal ruidoso como:

$$\mathbf{r} = \mathbf{c} + \mathbf{e}, \quad (2.7)$$

dónde \mathbf{e} es el vector de error.

El elemento de \mathbf{e} es 0 si el correspondiente elemento de \mathbf{r} es igual a \mathbf{c} , de otra manera es 1.

$$e = \begin{cases} 1, & \text{si un error ocurre en la posición } i \\ 0, & \text{otros casos} \end{cases} \quad (2.8)$$

2.2. Códigos cíclicos

Los códigos cíclicos son un subtipo de los códigos de bloques lineales. La mayoría de los bloques lineales descubiertos son códigos cíclicos o están relacionados con ellos.

Una ventaja de los bloques cíclicos en comparación con los demás es que la mayoría de los códigos cíclicos son fáciles de codificar. Además, los códigos cíclicos poseen una estructura matemática muy bien definida, lo que los hace que también se fácil de implementar unos esquemas de decodificación bastante eficientes para ellos. Un código se dice que es cíclico si posee las siguientes dos propiedades:

1. Linealidad: La suma de cualesquiera dos palabras de código es también una palabra de código, esta propiedad viene de que un código cíclico es un subtipo de un código de bloques lineales.
2. Propiedad cíclica: cualquier cambio cíclico de una palabra de código es también una palabra de código.

Si la lista ordenada $(c_0, c_1, \dots, c_{n-1})$ es un código de bloque lineal $(n-k)$, entonces el código es cíclico si la n -tupla:

$$\begin{aligned} &(c_{n-1}, c_0, \dots, c_{n-2}) \\ &(c_{n-2}, c_{n-1}, \dots, c_{n-3}) \\ &\vdots \\ &(c_1, c_2, \dots, c_{n-1}, c_0) \end{aligned}$$

es también una palabra de código. Estos coeficientes definen la palabra codificada polinomial como sigue:

$$c(X) = c_0 + c_1X + c_2X^2 \dots + c_{n-1}X^{n-1}, \quad (2.9)$$

dónde X representa un corrimiento en tiempo [1].

2.2.1. BCH Coding

El código Bose-Chaudhuri-Hocquenghem (BCH) es un tipo especial de código cíclico donde:

Longitud de bloque:	$n = 2^m - 1,$
Número de bits del mensaje:	$k \geq n - mt,$
Distancia mínima:	$d_{min} \geq 2t + 1,$

2. Codificación

donde m es un entero mayor o igual a 3.

Cada código BCH es un código de corrección de errores que puede detectar y corregir hasta t errores aleatorios por palabra codificada.

2.3. Códigos convolucionales

Un código convolucional genera los bits de redundancia aplicando la convolución en modulo-2. Su razón de codificación es igual a $1/n$, y puede ser descrita como una maquina de estados finitos que consiste en registros de corrimiento de M estados, n sumadores en modulo-2 y multiplexores que para la salida del sistema.

Un mensaje que contiene L bits, producirá a la salida un mensaje con una longitud de $n(L+M)$ bits. La razón de codificación es:

$$r = \frac{L}{n(L+M)}. \quad (2.10)$$

En la mayoría tenemos que $L \gg M$, por lo que podemos simplificar la ecuación a la siguiente:

$$r \simeq \frac{1}{n}. \quad (2.11)$$

En un codificador con M registros de corrimiento, la memoria del codificador es igual a los M bits del mensaje, y $K = M + 1$ corrimientos son necesarios para que un bit del mensaje de entrada influya en la señal e salida del codificador.

En la Figura 2.1 se puede apreciar el ejemplo de un codificador convolucional con $k = 2$ registros y $n = 2$ sumadores, y con (2.11) su razón de codificación es igual a $r = 1/2$.

Cada camino entre la entrada del codificador y su salida puede ser expresado por la respuesta al impulso, definida como la respuesta a un solo bit 1 a través del camino

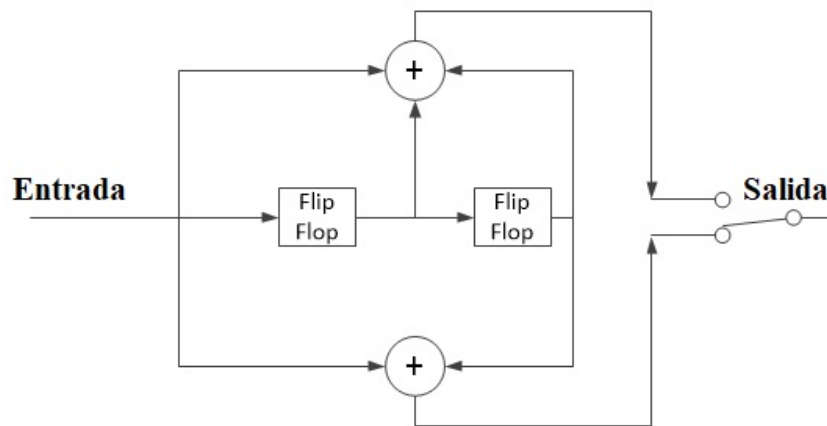


Figura 2.1: Codificador convolucional [1].

y un estado inicial cero. Además se puede caracterizar cada camino por un generador polinomial, definido como transformada de retardo unitario de la respuesta al impulso:

$$g^{(i)}(D) = g_0^{(i)} + g_1^{(i)}D + g_2^{(i)}D^2 + \dots + g_M^{(i)}D^M, \quad (2.12)$$

donde D es el retardo unitario variable, y $g_j^{(i)}$ son los coeficientes que describen la respuesta al impulso mediante el camino i .

El código convolucional de la Figura 2.1 es descrito por el siguiente conjunto de generadores polinomiales:

$$g^{(1)}(D) = 1 + D + D^2 \quad (2.13)$$

$$g^{(2)}(D) = 1 + D^2. \quad (2.14)$$

Por ejemplo, con el mensaje $\mathbf{m} = [10101]$, su representación polinomial es :

$$m(D) = 1 + D^2 + D^4. \quad (2.15)$$

La salida de la convolucion en el dominio del tiempo es un producto en el dominio D , entonces tenemos:

2. Codificación

$$c^{(1)} = g^{(1)}(D)m(D) \quad (2.16)$$

$$= (1 + D + D^2)(1 + D^2 + D^4) \quad (2.17)$$

$$= 1 + D^2 + D^4 + D + D^3 + D^5 + D^2 + D^4 + D^6 \quad (2.18)$$

$$= 1 + D + D^3 + D^5 + D^6, \quad (2.19)$$

$$c^{(2)} = g^{(2)}(D)m(D) \quad (2.20)$$

$$= (1 + D^2)(1 + D^2 + D^4) \quad (2.21)$$

$$= 1 + D^2 + D^4 + D^2 + D^4 + D^6 \quad (2.22)$$

$$= 1 + D^6 \quad (2.23)$$

la salida por el camino uno es [1101011] y mediante el camino 2 es [1000001]. Por ultimo, con los multiplexores se juntan las salidas de los dos caminos y se obtiene la siguiente palabra codificada:

$$\mathbf{c} = [11, 10, 00, 10, 00, 10, 11]. \quad (2.24)$$

Un enrejado o una maquina de estados es una manera gráfica de describir un código convolucional. La Figura 2.2 es un enrejado del código convolucional mostrado en la Figura 2.1.

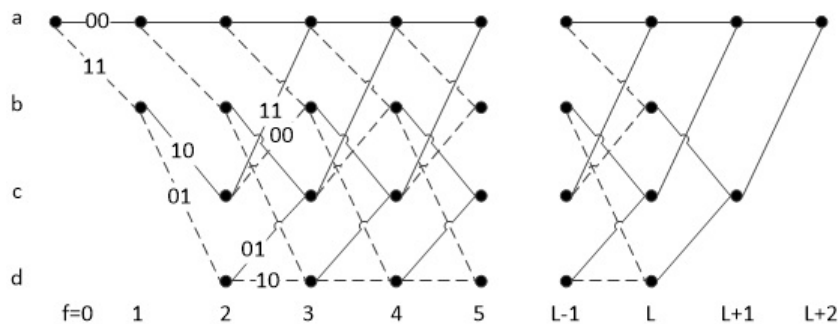


Figura 2.2: Enrejado [1].

Se pueden visualizar los estados y las salidas cuando la entrada del sistema es 0 o 1 con una línea sólida o punteada, respectivamente. El diagrama de estados en la Figura

2.3 contiene la misma información, pero de manera mas compacta.

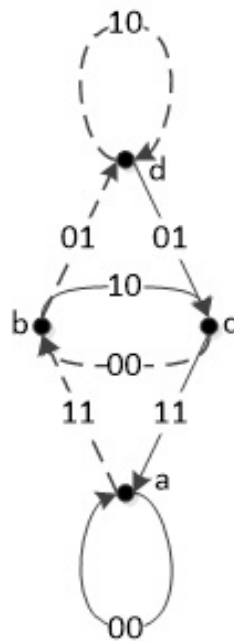


Figura 2.3: Diagrama de estados [1].

Para decodificar un mensaje hay que aplicar técnicas de máxima verosimilitud, que usan el hecho de que el mensaje correcto es el que tiene la menor distancia de Hamming entre el vector recibido \mathbf{r} y el vector transmitido \mathbf{c} . Para alcanzar esto se utiliza el algoritmo Viterbi.

2.3.1. Viterbi Algorithm [1]

1. Inicialización

- Etiqueta el estado mas a la izquierda del enrejado como 0.

2. Paso computacional $j + 1$

- Hacer $j = 0, 1, 2, \dots$, y considerar que en el paso anterior j dos cosas fueron realizadas:

2. Codificación

- Todos los caminos de supervivencia fueron identificados, de todos los caminos que terminan en un solo estado, solo los que tienen la menor distancia Hamming fueron guardados.
- Todos los caminos de supervivencia y sus distancias de Hamming fueron guardados.

3. Paso final

- Hasta que el algoritmo alcance el ultimo nodo, el camino con la menor distancia de Hamming es elegido y este es el mensaje decodificado.

2.4. Códigos de Comprobación de Paridad de Baja Densidad

Los códigos de comprobación de paridad de baja densidad son definidos por una matriz de comprobación de paridad (Parity Check Matrix (PCM)) \mathbf{A} la cual es de baja densidad, esto es que esta compuesta por su mayoría de 0 y un numero pequeño de 1.

Un código de comprobación de paridad de baja densidad (n, t_c, t_r) es una matriz de comprobación de paridad en la cual el valor n representa la longitud del bloque de la palabra codificada, t_c es el peso de Hamming (numero de 1) en cada columna, y t_r es el peso de Hamming en cada renglón. La razón de codificación esta dada por:

$$r = 1 - \frac{t_c}{t_r}. \quad (2.25)$$

Los renglones en la matriz \mathbf{A} deben ser linealmente independientes. Un ejemplo de una matriz de comprobación de paridad de baja densidad puede ser visto en (2.26), con $n = 10$, $t_c = 3$, $t_r = 5$ y una razón de codificación igual a $r = 0.40$.

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (2.26)$$

Los códigos LDPC pueden ser caracterizados completamente con un grafo de Tanner, el cual es un grafo bipartito, en el que sus nodos pueden ser divididos en dos grupos: los nodos de verificación (check-nodes) y los nodos de variables (variable-node). Las líneas que unen los nodos de verificación con los de variables se les conoce como bordes.

Entre la matriz PCM y el grafo de Tanner existe una correspondencia uno a uno, en la cual, las columnas de la matriz H corresponden a los nodos de variables y las filas de la matriz H corresponden a los nodos de verificación, y existirá un borde que conecte un nodo de verificación con uno de variable cuando haya una entrada no nula en la intersección de la fila y la columna correspondiente.

En la figura 2.4, en la intersección de c2 y v5 hay un elemento no nulo y en el grafo de Tanner hay un borde que conecta esos elementos, como se puede observar en la figura 2.5.

Un ciclo es una caminata cerrada en un grafo sin que se repitan nodos o bordes, la longitud de un borde está dada por el numero de bordes que participan en él, que para un grafo bipartito es un numero par, siendo la longitud mínima de cuatro, como se muestra en la figura 2.6.

La principal diferencia entre los códigos LDPC y los códigos de bloque clásico es en la forma en la que son decodificados, ya que los códigos de bloque clásicos se decodifican con algoritmos tipo ML (máximum likelihood), que son óptimos pero com-

2. Codificación

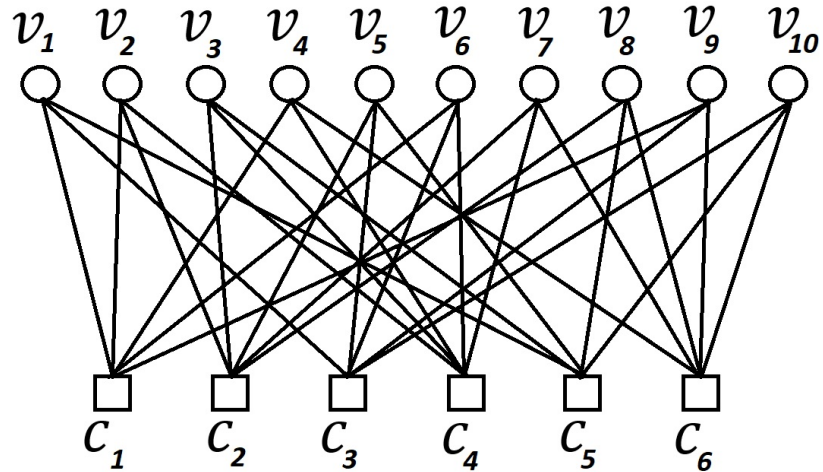


Figura 2.4: Grafo de Tanner matriz (2.1)

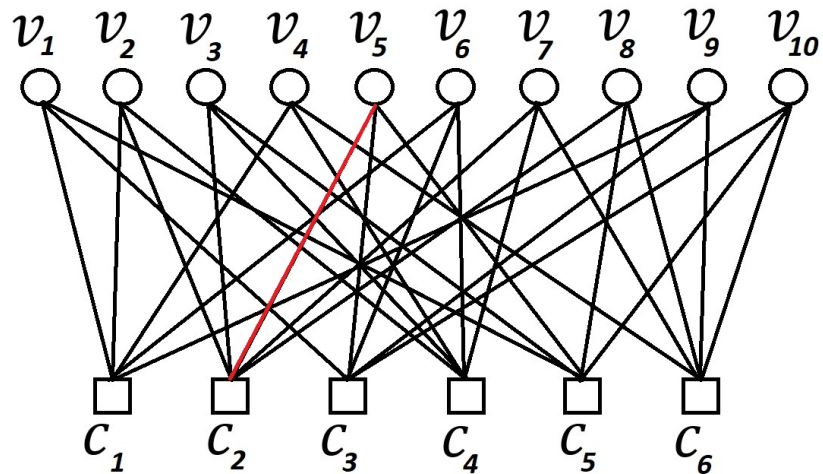


Figura 2.5: Borde entre elementos en Grafo de Tanner

plejos. Usualmente su complejidad crece exponencialmente con la longitud del código, mientras que los LDPC son decodificados usando algoritmos iterativos del tipo message passing, que son aproximaciones de los algoritmos ML, pero con una complejidad menor, ya que su complejidad suele ser lineal con la longitud del código.

De acuerdo con su construcción los códigos LDPC pueden ser divididos en dos grupos

1. Los construidos aleatoriamente: son generados por una guía computacional si-

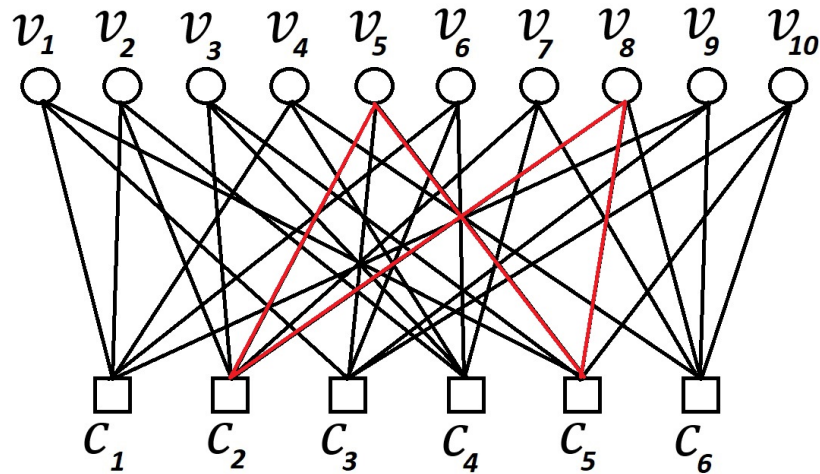


Figura 2.6: Ciclo en Grafo de Tanner

guiendo ciertas reglas de construcción o requerimientos estructurales en su grafo de Tanner.

2. Los códigos estructurales: construidos con base a métodos estructurales y/o combinatorios.

Tomando en cuenta los términos de desempeño, los códigos creados de manera aleatoria suelen ser mejor que los estructurales. Sin embargo, los códigos aleatorios pueden ser bastante complejos en su construcción, esto es debido a su falta de estructura.

Los códigos LDPC cuasi cíclicos (QC LDPC) son una clase de códigos estructurales que permite una codificación de baja complejidad ya que pueden ser codificados usando registros de corrimiento, lo que le da una complejidad lineal con el número de nodos de verificación en la PCM. En la literatura se han reportado códigos QCLDPC que tienen un desempeño muy parecido a los generados aleatoriamente por computadora. Los códigos QC LDPC pueden ser implementados eficientemente en circuitos integrados por su simetría cíclica, pues tiene un alambrado regular y una estructura modular.

La PCM de un código QC LDPC consiste en un arreglo de matrices circulantes y dispersas del mismo tamaño, al cual se le conoce como matriz generadora.

2. Codificación

Una matriz circulante es una matriz cuadrada en la cual cada fila es un desplazamiento a la derecha de la fila superior, mientras que la primera fila es un desplazamiento de la última fila, cuando una matriz circulante tiene un único elemento no nulo por fila se dice que es una matriz de permutación circulante (CPM, Circulant Permutation Matrix), la cual se puede caracterizar completamente por la primera fila de la matriz, a la cual se le llama como generadora.

$$CPM = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (2.27)$$

3

Bancos de filtros

En este capítulo se presentan los fundamentos de dos tipos básicos de bancos de filtro, los Bancos de Filtros Maximamente Decimados y los Bancos de Filtros Sobremuestrados, estableciendo relación con los efectos generados a la salida como el aliasing. Además se describe la representación polifásica de los bancos de filtros

3.1. Bancos de Filtros Maximamente Decimados

Un banco de filtros está compuesto de dos partes; un banco de filtros de análisis y un banco de filtros de síntesis. El banco de filtros de análisis divide la señal de entrada $x(n)$ en diferentes señales de subbandas $v_k(n)$. El banco de filtros de síntesis combina las diferentes señales de subbandas $\hat{v}_k(n)$ en una señal de salida $\hat{x}(n)$. Ambos bancos de filtros están compuestos de una colección de N filtros digitales, con una entrada en común para el banco de filtros de análisis, y una salida en común para el banco de filtros de síntesis.

Las señales filtradas en el banco de análisis son submuestreadas por M , obteniendo $v_k(n)$, mientras que las señales $\hat{v}_k(n)$ que entran al banco de filtros de síntesis son sobremuestradas por M . Si el número de subbandas N es igual al factor de submuestreo M , el banco de filtros es llamado *Maximamente decimado*. En la Figura 3.1 se puede ver un ejemplo de un Banco de Filtros Maximamente decimado, en el cual tenemos $N = 4$ y $M = N = 4$

3. Bancos de filtros

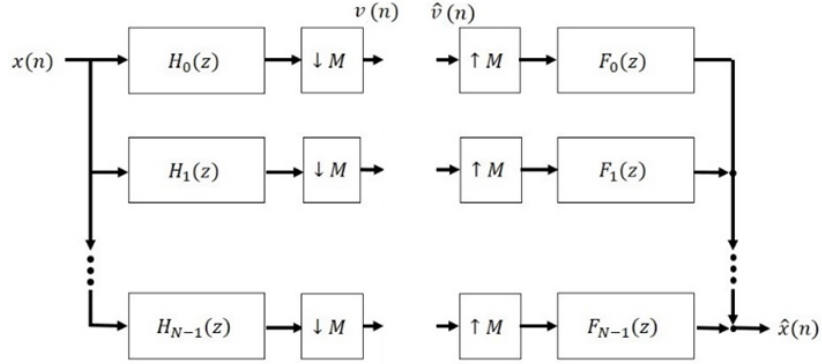


Figura 3.1: Banco de Filtros Maximamente Decimado

3.2. Errores en Bancos de Filtros

La señal reconstruida $\hat{x}(n)$ es, en el mejor de los casos, una aproximación de la señal de entrada original $x(n)$. Existen algunos errores que pueden ocurrir durante la división de la señal en subbandas hecha por el banco de filtros de análisis, siendo las siguientes:

- Relaciones de entrada-salida para Bancos de filtros Maximamente Decimados.
- Reconstrucción perfecta.

3.2.1. Relaciones de entrada-salida para Bancos de filtros Maximamente Decimados

La relacion de entrada-salida en el dominio z es expresado como en [2]:

$$Y(z) = T_0(z)X(z) + \sum_{l=1}^{M-1} T_l(z)X(ze^{-j2\pi l/M}), \quad (3.1)$$

donde:

$$T_0(z) = \frac{1}{M} \sum_{k=0}^{M-1} F_k(z)H_k(z), \quad (3.2)$$

y:

$$T_l(z) = \frac{1}{M} \sum_{k=0}^{M-1} F_k(z)H_k(ze^{-j2\pi l/M}) \quad \text{para} \quad l = 1, 2, \dots, M-1. \quad (3.3)$$

El efecto de aliasing esta relacionado con el termino $T_l(z)$ de la ecuación (3.1), y para poder obtener la señal de salida lo mas parecida a la señal de entrada, se tiene que procurar que:

$$T_l(z) = 0. \quad (3.4)$$

En este caso la señal de salida queda expresada en términos de la entrada como:

$$Y(z) = T_0(z)X(z). \quad (3.5)$$

El factor $T_0(z)$ es llamado la *función de transferencia de distorsión* y determina la distorsión causada a todo el sistema en amplitud y en fase.

Lo que se busca en esta función es que tenga una magnitud constante, para que la señal de salida sea una version escalada de la señal original, y que se tenga una fase lineal.

3.2.2. Reconstrucción perfecta

Si un banco de filtros no presenta ningún aliasing, distorsión en fase o amplitud, este entonces cumple con la propiedad de *reconstrucción perfecta*, esto es:

$$T_l(z) = 0. \quad (3.6)$$

Y que la función de transferencia de distorsión sea:

$$T_0(z) = cz^{-n_0}. \quad (3.7)$$

donde c es una constante diferente de cero y n_0 es un entero positivo.

3. Bancos de filtros

Teniendo estas consideraciones, se puede asegurar que la señal de salida es una aproximación muy cercana a la señal de entrada original, de modo que se puede expresar como:

$$\hat{X}(z) = cz^{-n_0}X(z), \quad c \neq 0. \quad (3.8)$$

Además, en el dominio del tiempo puede verse como:

$$\hat{x}(n) = cx(n - n_0), \quad c \neq 0. \quad (3.9)$$

Para cualquier señal $x(n)$. En otras palabras, se dice que $\hat{x}(n)$ es una versión escalada y atrasada de la señal de entrada $x(n)$, en la Figura 3.2 se puede observar un ejemplo de una señal que pasa por un banco de filtros con reconstrucción perfecta, siendo la señal de salida una versión retrasada y escalada de la versión original de la señal.

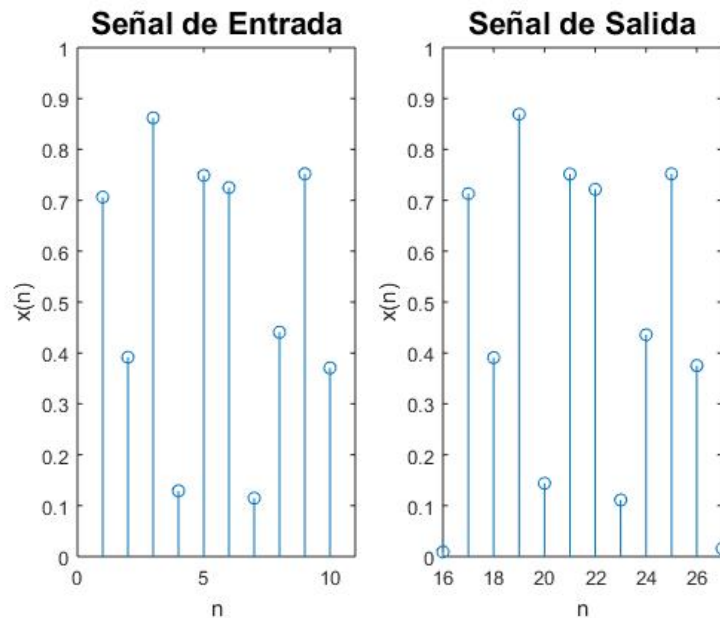


Figura 3.2: Señal de entrada y salida con Reconstrucción Perfecta

3.3. Representación polifásica

La función de transferencia $H_k(z)$ puede ser expresada como [3]:

$$\begin{bmatrix} H_0(z) \\ \vdots \\ H_{M-1}(z) \end{bmatrix} = \begin{bmatrix} E_{0,0}(z^M) & E_{0,1}(z^M) & \dots & E_{0,M-1}(z^M) \\ E_{1,0}(z^M) & E_{1,1}(z^M) & \dots & E_{1,M-1}(z^M) \\ \vdots & \vdots & \ddots & \vdots \\ E_{M-1,0}(z^M) & E_{M-1,1}(z^M) & \dots & E_{M-1,M-1}(z^M) \end{bmatrix} \begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-(M-1)} \end{bmatrix}, \quad (3.10)$$

o como:

$$h(z) = E(z^M)e(z), \quad (3.11)$$

donde:

Esta representación es mostrada en la Figura 3.3.

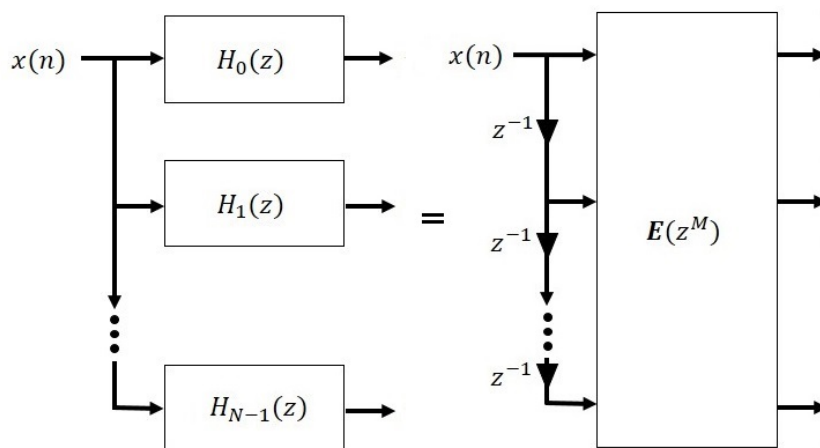


Figura 3.3: Representacion Polifasica tipo 1 de un banco de filtros de análisis [4].

Podemos expresar el banco de filtros de síntesis de igual manera.

Uniendo el banco de análisis y el banco de síntesis se obtiene el sistema mostrado en la Figura 3.4.

Como se puede apreciar en la Figura 3.4, en la parte del banco de analisis se van a computar datos que van a ser descartados por el submuestreador, y en el banco

3. Bancos de filtros

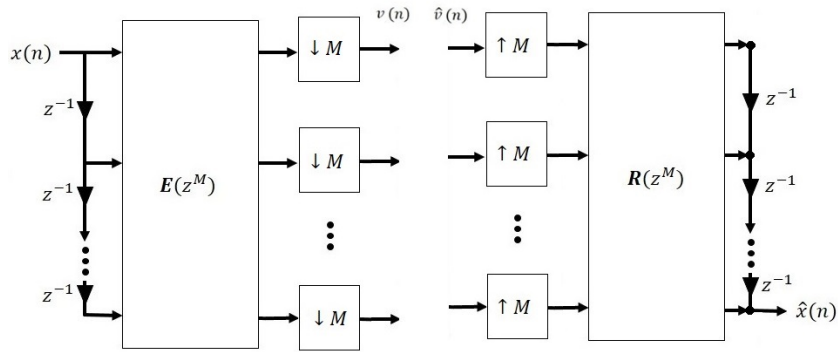


Figura 3.4: Representación Polifásica de un banco de filtros de M canales .

de filtros de análisis, el sobremuestreador agrega mas datos para computara la entrada de los filtros. Para hacer mas eficiente esta tarea se utilizan las identidades nobles [4] mostradas en la Figura 3.5(a) y en la Figura 3.5(b), para el banco de filtros de análisis y el banco de filtros de síntesis, respectivamente.

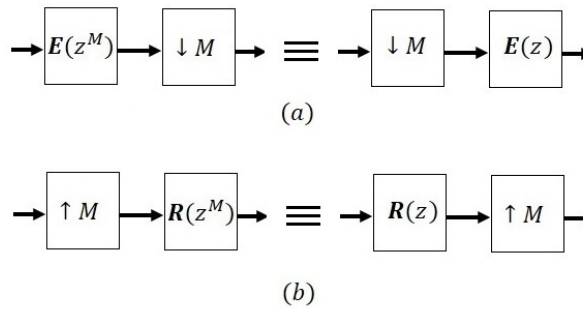


Figura 3.5: (a)Identidad noble submuestreador (b)Identidad noble sobremuestreador.

Aplicando las identidades nobles al sistema de la Figura 3.4 obtenemos el la siguiente forma simplificada del sistema, mostrada en la Figura 3.6.

Para satisfacer la propiedad de reconstrucción perfecta, podemos definir a $R(z)$, como se muestra a continuacion [3]:

$$R(z) = cz^{-K} \tilde{E}(z). \quad (3.12)$$

que es llamada la propiedad *paraunitaria* [4].

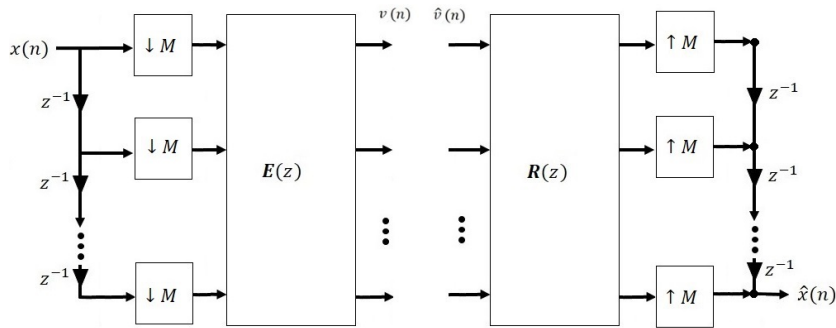


Figura 3.6: Representación Polifásica de un banco de filtros de M canales usando identidades nobles [4].

La propiedad paraunitaria implica que:

$$\tilde{E}(z)E(z) = dI, \quad \text{para todo } z \quad d > 0. \quad (3.13)$$

En otras palabras, la matriz polifásica $P(z)$ en la Figura 3.7, debe ser $I(z)$ para tener reconstrucción perfecta.

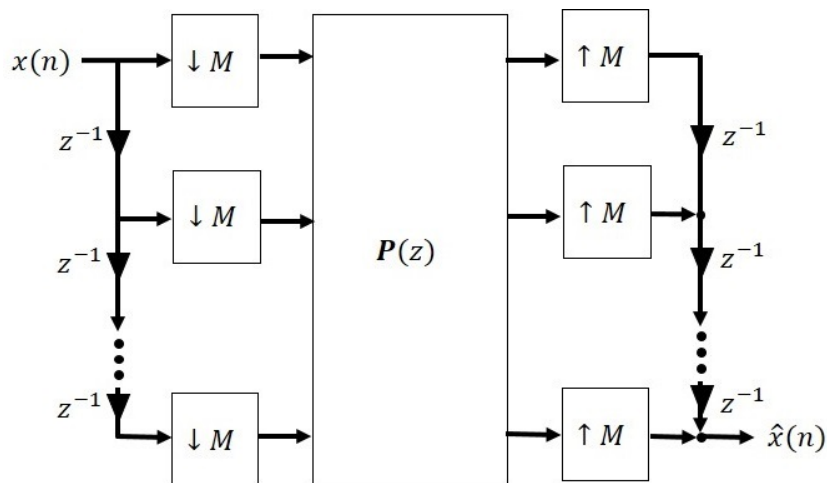


Figura 3.7: Representación Polifásica simplificada de un banco de filtros de M canales [3].

3.4. Bancos de Filtros Sobremuestreados

Un Banco de Filtros, en el cual el número de subbandas N es mayor que el factor de submuestreo M , es decir, que $N > M$, las señales de subbanda $v_k(n)$ son redundantes, quiere decir que se tiene un mayor número de muestras por minuto que la señal original $x(n)$. Este tipo de filtros es llamado Banco de Filtros Sobremuestreados (Oversampled Filter Banks (OFB)), como se muestra en la Figura 3.8, y de los cuales se hablará más en profundidad en el siguiente capítulo.

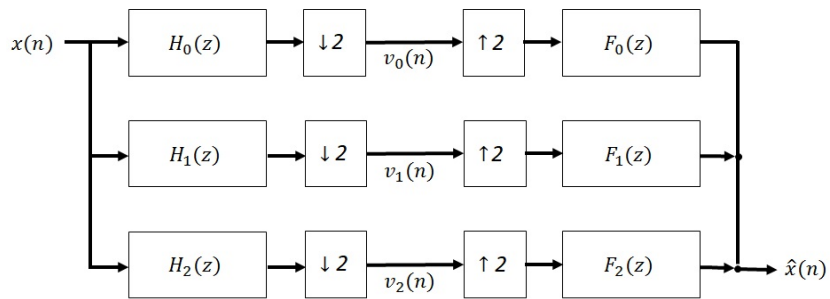


Figura 3.8: Banco de Filtros Sobremuestreado para $M = 3$ y $N = 2$.

Diseño de Bancos de filtros sobremuestreados para codificación

En este capítulo se presenta el diseño de un Banco de Filtros sobremuestreado modulado por la transformada del coseno, así como su utilización en codificación.

4.1. Modulación por la Transformada Discreta del Coseno

Los bancos de filtros modulados por la transformada del coseno es una sub-clase especial de los bancos de filtros generales de M – canales como el mostrado en la Figura 3.1. En diferentes áreas de procesamiento digital de señales, como modulación multicanal [5], estos bancos de filtros han sido usados extensivamente.

En los bancos de filtros modulados por coseno, el banco de filtros de análisis y de síntesis son creados a partir del mismo filtro prototipo pasa-bajas $p(n)$. y aplicando la Transformada Discreta del Coseno (Discrete Cosine Transform, DCT) se obtienen las respuestas al impulso de los filtros de banco de analisis como $h_k(n)$, de longitud L [6],[7]

$$h_{2k}(n) = 2p(n) \cos \left[\frac{\pi}{N} (k + 0.5) \left(n - \frac{k}{2} \right) + (-1)^k \frac{\pi}{4} \right], \quad (4.1)$$

4. Diseño de Bancos de filtros sobremuestreados para codificación

donde $n = 0, \dots, L - 1$ y $k = 0, \dots, N - 1$

Las $N \times M$ matrices polifasicas de analisis y sintesis, $\mathbf{H}_p(z)$ and $\mathbf{F}_p(z)$, respectivamente son descritas como:

$$\mathbf{H}_p(z) = \mathbf{W}_N^H \mathbf{E}(z) \quad (4.2)$$

$$\mathbf{F}_p(z) = \tilde{\mathbf{H}}_p(z) \quad (4.3)$$

$$= \tilde{\mathbf{E}}(z) \mathbf{W}_N, \quad (4.4)$$

donde \mathbf{W}_N es la matriz de coeficientes DCT de tamaño $N \times N$.

La restricciones de reconstrucción perfecta para bancos de filtros paraunitarios sobremuestreados pueden ser escritas como:

$$\mathbf{F}_p(z) \mathbf{H}_p(z) = \tilde{\mathbf{H}}_p(z) \mathbf{H}_p(z) = \tilde{\mathbf{E}}(z) \mathbf{E}(z) = \mathbf{I}_M, \quad (4.5)$$

y $\mathbf{E}(z)$ puede ser dividido en b conjuntos independientes de matrices de reconstrucción perfecta:

$$\begin{aligned} \tilde{\mathbf{E}}_l(z) \mathbf{E}_l(z) &= \mathbf{I}_{M/b} \quad \text{con} \quad [\mathbf{E}_l(z)]_{k,j} = [\mathbf{E}(z)]_{l+kb, l+jb}, \quad (4.6) \\ l &= 0, \dots, b-1, \quad k = 0, \dots, J-1, \\ j &= 0, \dots, M/b-1. \end{aligned}$$

Dado que en $\mathbf{E}_l(z)$ contiene la componentes polifasicas sobremuestreadas retrasadas J veces del filtro prototipo con el mismo retraso en cada diagonal, este puede ser expresado como[8]:

$$\mathbf{E}_l(z) = \Lambda_1(z) \mathbf{E}_{l\downarrow}(z^J) \Lambda_2(z). \quad (4.7)$$

Con las matrices paraunitarias diagonales apropiadas $\Lambda_1(z)$ y $\Lambda_2(z)$ conteniendo monomios en z . La version submuestreadas J veces de las restricciones de reconstrucción perfecta puede ser ahora reescrita como:

$$\tilde{\mathbf{E}}_{l\downarrow}(z) \mathbf{E}_{l\downarrow}(z) = \mathbf{I}_{M/b}. \quad (4.8)$$

4. Diseño de Bancos de filtros sobremuestreados para codificación

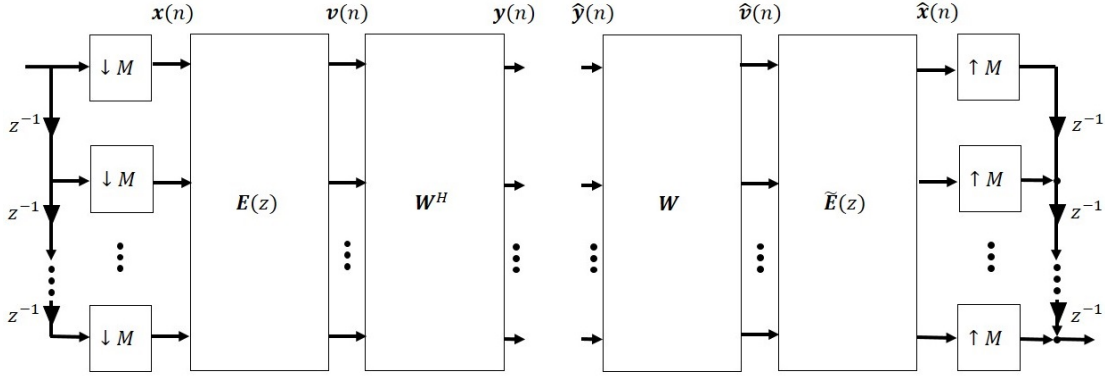


Figura 4.1: Realización polifásica de un banco de filtros sobremuestreado.

4.2. Polinomios de comprobación de paridad

Como se vio en el capítulo anterior, un Banco de Filtros Sobremuestreado tiene la característica de que tiene una mayor cantidad de muestras por minuto que la señal original en cada subbanda. Esto es llamado redundancia.

Esta redundancia puede ser usada en el proceso de codificación de la señal de entrada, teniendo en cuenta $M < N$, donde N es el número de subbandas en el Banco de filtros y M es el factor de submuestreo, esto nos da como resultado una razón de codificación:

$$\frac{M}{N} < 1, \quad (4.9)$$

con la cual se asegura la robustez de la codificación frente al ruido, evitando así errores en la información.

Para lograr esto, es necesario la creación del síndrome, que es la matriz que nos va a ayudar en el control de errores.

Se necesita matriz de polinomios de comprobación de paridad $\mathbf{C}^{PC}(z)$ en el receptor para calcular las señales del síndrome $N - M$ para cada subbanda. Teniendo que $\mathbf{C}^{PC}(z)$ satisface la siguiente ecuación

$$\mathbf{C}^{PC}(z)\mathbf{H}_p(z) = \mathbf{0}_{(N-M) \times M}, \quad (4.10)$$

4. Diseño de Bancos de filtros sobremuestreados para codificación

en la cual se puede ver que:

$$\hat{y}(n) = y(n), \quad (4.11)$$

el síndrome es igual a cero.

Teniendo eso, factorizamos la matriz polinomial de comprobación de paridad como:

$$\mathbf{C}^{PC}(z) = z^{-v} \tilde{\mathbf{C}}(z) \mathbf{W}^H, \quad (4.12)$$

donde $\tilde{\mathbf{C}}(z)$ es la matriz polinomial de $(N - M) \times N$. Y tomando (4.2) y (4.12) podemos reescribir (4.10) como:

$$\tilde{\mathbf{C}}(z) \mathbf{E}(z) = \mathbf{0}_{(N-M) \times M}. \quad (4.13)$$

y podemos expresar la anterior ecuación en matrices polinomiales de comprobación de paridad de tamaño $(N - M)/b \times J$:

$$\begin{aligned} \tilde{\mathbf{C}}_l(z) \mathbf{E}_l(z) &= \mathbf{0}_{(N-M) \times M/b}, \quad (4.14) \\ \text{con } [\tilde{\mathbf{C}}_l(z)]_{i,j} &= [\tilde{\mathbf{C}}(z)]_{l+ib, l+jb}, \\ 0 \leq l < b, \quad 0 \leq i < (N-M)/b, \quad 0 \leq j < J. \end{aligned}$$

Una version submuestreada es definida expresando $\tilde{\mathbf{C}}_l(z)$ como un producto de la matriz paraunitaria diagonal $\tilde{\Lambda}_3(z)$, la matriz $\tilde{\mathbf{C}}_{l\downarrow}(z^J)$ formada de polinomios en z^J y $\tilde{\Lambda}_1(z)$ que es la matriz paraconjugada de la matriz $\Lambda_1(z)$ en (4.7):

$$\tilde{\mathbf{C}}_l(z) = \tilde{\Lambda}_3(z) \tilde{\mathbf{C}}_{l\downarrow}(z^J) \tilde{\Lambda}_1(z), \quad l = 0, \dots, b-1. \quad (4.15)$$

y con (4.7), (4.15) to (4.14), la version submuestreda de la ecuación (4.14) queda como:

$$\tilde{\mathbf{C}}_{l\downarrow}(z) \mathbf{E}_{l\downarrow}(z) = \mathbf{0}_{(N-M)/b \times M/b}. \quad (4.16)$$

4.3. Creación del síndrome

De (4.13) se analiza que la matriz polifásica $\mathbf{E}(z)$, cuando es multiplicada por la matriz de comprobación de paridad $\tilde{\mathbf{C}}(z)$ se forma un espacio nulo o matriz cero.

4. Diseño de Bancos de filtros sobremuestreados para codificación

Entonces, cuando las señales de subbanda $\hat{V}(n)$ alimentan la entrada de la matriz de paridad como se muestra en la Figura 4.2, las señales de salida resultante, que son llamadas como síndrome están dadas por:

$$\mathbf{S}(z) = \tilde{\mathbf{C}}(z)\hat{\mathbf{V}}(z). \quad (4.17)$$

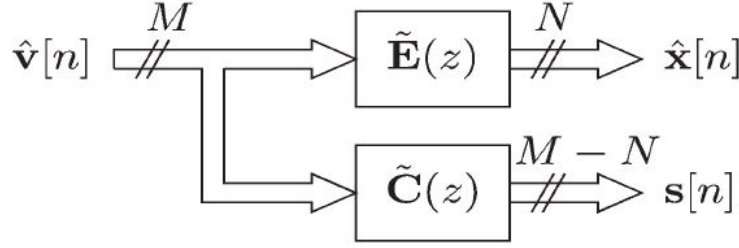


Figura 4.2: Generación del síndrome.

Las señales de subbanda $\hat{\mathbf{Y}}(z)$ en el banco de síntesis pueden ser escritas como:

$$\hat{\mathbf{Y}}(z) = \mathbf{Y}(z) + \mathbf{Q}(z), \quad (4.18)$$

donde $\mathbf{Q}(z)$ representa la transformada z del ruido agregado a las señales de subbanda $\mathbf{Y}(z)$. A partir de la Figura 4.1 podemos escribir las señales de subbanda $\hat{V}(n)$ como:

$$\hat{\mathbf{V}}(z) = \mathbf{W}\hat{\mathbf{Y}}(z) \quad (4.19)$$

$$= \mathbf{W}[\mathbf{Y}(z) + \mathbf{Q}(z)] \quad (4.20)$$

$$= \mathbf{W}\mathbf{W}^H\mathbf{E}(z)\mathbf{X}(z) + \mathbf{W}\mathbf{Q}(z) \quad (4.21)$$

$$= \mathbf{E}(z)\mathbf{X}(z) + \mathbf{W}\mathbf{Q}(z). \quad (4.22)$$

Ahora sustituimos en la ecuación (4.17) obtenemos:

$$\mathbf{S}(z) = \tilde{\mathbf{C}}(z)[\mathbf{E}(z)\mathbf{X}(z) + \mathbf{W}\mathbf{Q}(z)], \quad (4.23)$$

$$= \tilde{\mathbf{C}}(z)\mathbf{E}(z)\mathbf{X}(z) + \tilde{\mathbf{C}}\mathbf{W}\mathbf{Q}(z), \quad (4.24)$$

donde el primer término de la ecuación se hace cero por (4.13), quedando:

$$\mathbf{S}(z) = \tilde{\mathbf{C}}\mathbf{W}\mathbf{Q}(z). \quad (4.25)$$

4. Diseño de Bancos de filtros sobremuestreados para codificación

Si $\hat{y}[n] = y[n]$, lo que pasa en ausencia de ruido en las señales de subbanda procesadas, entonces el las señales del síndrome serán igual a cero. Sin embargo, si hay ruido en las señales de subbanda, el síndrome sera diferente de cero y esta información puede ser usada para verificar la presencia de errores en el vector de información recibido.

4.4. Diseño del filtro prototipo

Las matrices polinomiales paraunitarias de comprobación de paridad pueden ser calculadas dentro del diseño del filtro prototipo. La función de costo aplicada minimiza la energía de la banda de rechazo. Además, podemos factorizar $[\mathbf{E}_{l\downarrow}(z)\mathbf{C}_{l\downarrow}(z)]$ en bloques de construcción paraunitarios de grado 1 del tipo Householder [3].

$$\begin{aligned} [\mathbf{E}_{l\downarrow}(z)\mathbf{C}_{l\downarrow}(z)] &= \prod_{i=1}^{L_e-1} \mathbf{V}_{l,i}(z) \prod_{j=1}^{J-1} \mathbf{U}_{l,j}, & (4.26) \\ \mathbf{V}_{l,i}(z) &= \mathbf{I}_J - \mathbf{v}_{l,i}\mathbf{v}_{l,i}^T + z^{-1}\mathbf{v}_{l,i}\mathbf{v}_{l,i}^T, \\ \mathbf{U}_{l,j} &= \mathbf{I}_J - 2\mathbf{u}_{l,j}\mathbf{v}_{l,j}^T, \end{aligned}$$

donde $L_e = \lceil L/K \rceil$ y $J = MN/g^2$. Los parámetros $\mathbf{v}_{l,i}$ y $\mathbf{u}_{l,j}$ son optimizados para que el banco de filtros paraunitario tenga una inserción de aliasing mínima.

5

Bancos de Filtros para codificación LDPC

En este capítulo se presenta un modelo de comunicaciones para codificación de control de errores usando Bancos de Filtros sobremuestreados modulados por Coseno y códigos de comprobación de paridad de baja densidad. El modelo propuesto es comparado con enfoques similares encontrados en la literatura, usando simulaciones en Matlab.

Como se mostró en el capítulo anterior, los Bancos de Filtros sobremuestreados son una herramienta útil para codificación, dado sus características intrínsecas. Asimismo, las técnicas de codificación LDPC han mostrado un desempeño excelente, al ser los únicos que se aproximan al límite de Shannon, además de requerir cálculos matemáticos de menor orden, comparados con los Turbo códigos, que son la segunda mejor opción en cuanto a rendimiento, además, aunque su decodificación suele ser muy compleja, presenta la posibilidad de paralelizar completamente su algoritmo de decodificación.

El modelo propuesto en [3] es presentado en la figura 5.1 con el propósito de reducir la probabilidad de error para un canal con ruido Blanco Gaussiano Aditivo (Additive White Gaussian Noise AWGN) y error impulsivo.

El transmisor (Figura 5.1(a)) recibe la señal de entrada y la divide en varias subbandas con el banco de filtros de análisis. El siguiente paso es cuantizar la información

5. Bancos de Filtros para codificación LDPC

obtenida en cada sub-banda y definir una cadena bits con esos datos. La salida de la cadena de bits puede tener diferentes longitudes de acuerdo a los diferentes valores de asignación óptima de bits asignados a cada sub-banda. La cadena de caracteres es codificada con la matriz LDPC generadora definida por la matriz de comprobación de paridad. Después de esto, cada 0 binario es codificado como un 1, y cada 1 es codificado como -1 . Entonces la señal es enviada al canal corrupto por ruido.

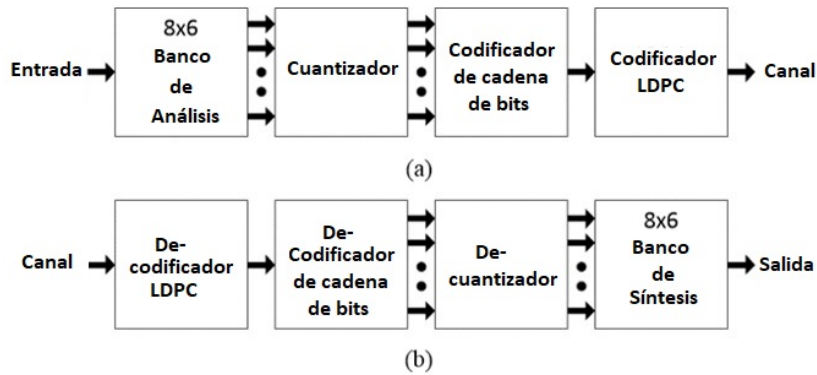


Figura 5.1: Modelo propuesto en [3] a) Transmisor. b) Receptor

En el lado del receptor (Figura 5.1(b)) se realiza el proceso inverso. La señal corrompida por el ruido es corregida con el decodificador LDPC. Después, la cadena de bits es decodificada y los datos son enviados a su correspondiente sub-banda. Una vez obtenida la información en cada sub-banda, el proceso de decuantización es aplicado. Finalmente la reconstrucción del mensaje es realizado con el banco de filtros de síntesis.

El cuantizador y decuantizador son diseñados de acuerdo con un proceso de asignación óptima de bits de acuerdo con [4]:

$$b_k = b + \frac{1}{2} \log_2 \left[\frac{\sigma_{xk}^2}{\left(\prod_{i=0}^{N-1} \sigma_{xi}^2 \right)^{1/N}} \right], \quad (5.1)$$

donde b_k es la asignación de bits en la subbanda k , b es la asignación de bits promedio y σ_{xk}^2 es la varianza de la señal en la subbanda k

5.1. Simulación

La simulación fue realizada en MatLab, siguiendo con los siguientes pasos

Paso 1: Diseño del filtro prototipo $p(n)$

El filtro prototipo es diseñado utilizando (4.26), donde $L_e = \lceil L/K \rceil$ y $J = MN/g^2$. Los parámetros $\mathbf{v}_{l,i}$ y $\mathbf{u}_{l,j}$ son optimizados para que el banco de filtros para-unitario tenga una inserción de aliasing mínima debido al proceso de submuestreo. La energía de la banda de rechazo es computada y minimizada por un procedimiento de optimización escalonado.

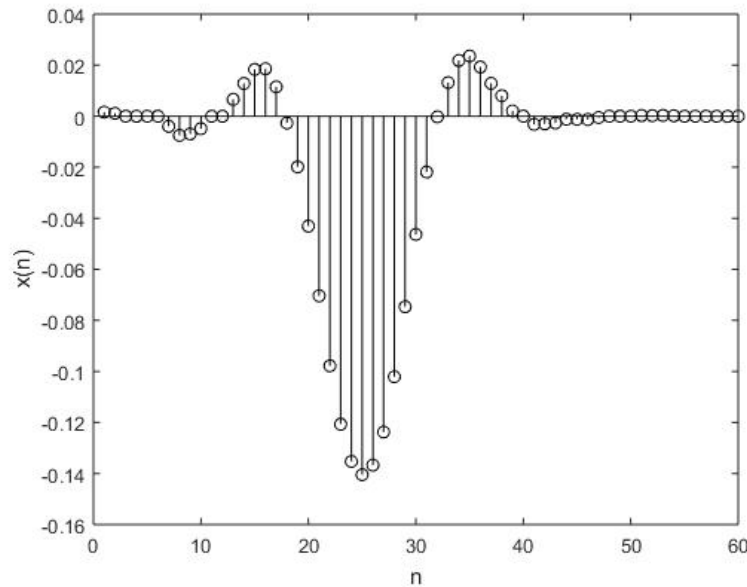


Figura 5.2: Respuesta al impulso del filtro prototipo $p(n)$

En esta simulación se generó un banco de filtros sobremuestreado con $N = 8$ sub-bandas, un factor de decimación de $M = 6$. El filtro prototipo tiene una longitud de $L = 60$. El máximo común divisor es $g = 2$, el mínimo común múltiplo es $K = 24$. La respuesta al impulso y la respuesta en magnitud del filtro prototipo son mostradas en la Figura 5.2 y Figura 5.3, respectivamente.

Paso 2: Modulación

5. Bancos de Filtros para codificación LDPC

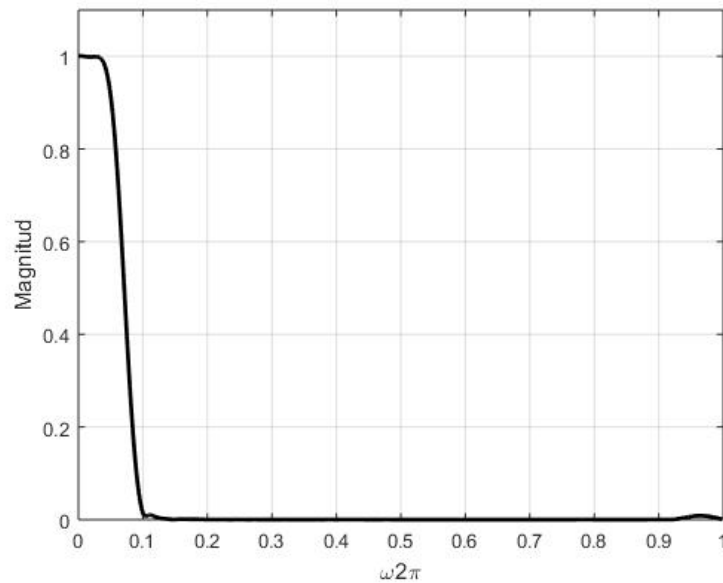


Figura 5.3: Respuesta en magnitud del filtro prototipo $p(n)$

El filtro prototipo $p(n)$ es modulado usando (??) y (4.1), para modulación DCT. La Figura 5.4 nos muestra las respuestas en frecuencia de los filtros diseñados por la transformada discreta del coseno

Paso 3: Codificación

Cada subbanda es codificada usando una matriz de comprobación de paridad $(273, 3, 10)$ tomada de [9], la cual tiene una longitud de la palabra codificada de $n = 273$ y una razón de codificación de $r = 0.7$.

Paso 4: Canal

El canal es modelado como Ruido Blanco Gaussiano Aditivo (Additive White Gaussian Noise (AWGN)) con media cero y $\sigma_w^2 = 0.001$ en todas las subbandas, en la Figura 5.5 se puede ver un ejemplo de ruido AWGN, y ruido impulsivo con Probabilidad de Ocurrencia de Impulso (POI) variando de 10^{-3} hasta 10^{-1} . En la Figura 5.6 se muestra un ejemplo de ruido impulsivo con probabilidad de 10^{-3} .

Paso 5: Señal de salida y computación del Error Cuadrático Medio

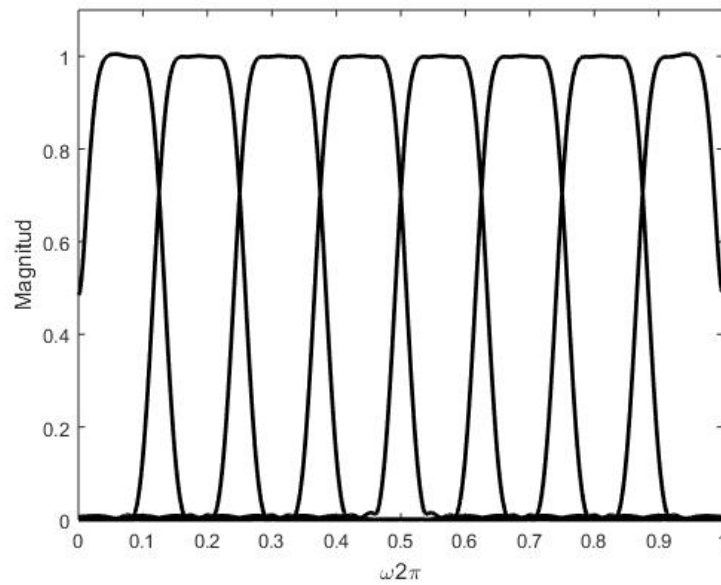


Figura 5.4: Banco de filtros modulado por la transformada discreta del coseno

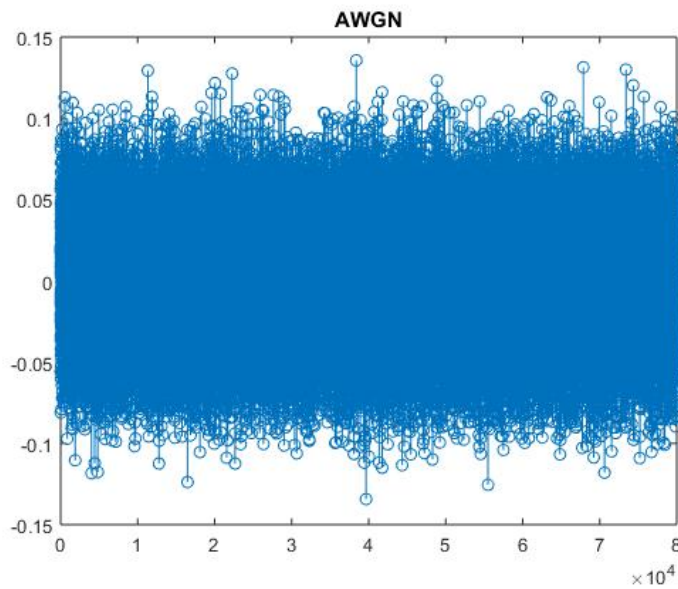


Figura 5.5: Ejemplo de ruido AWGN

La señal de salida es generada a la salida del banco de filtros de síntesis, y a partir de esa información es que se hace el cálculo del Error Cuadrático Medio (Mean Squared Error (MSE)), usando 5.2 para verificar la calidad de la señal

5. Bancos de Filtros para codificación LDPC

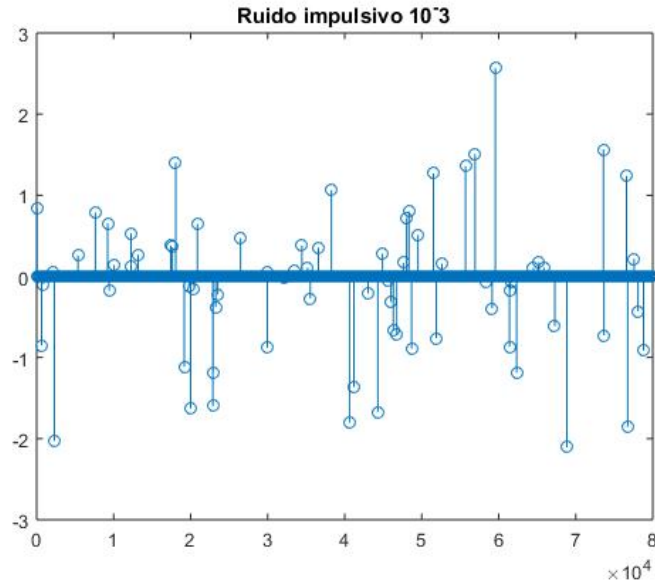


Figura 5.6: Ejemplo de Ruido Impulsivo con probabilidad 10^{-3}

reconstruida después de todo el procesamiento realizado [10].

$$MSE = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2, \quad (5.2)$$

5.2. Resultados

Los resultados son mostrados en la Figura 5.7 para modulación DCT, en la cual se pueden apreciar los valores obtenidos de MSE para diferentes valores de cuantización ($b = 4$, $b = 6$, $b = 7$).

Los resultados son comparados en términos de MSE vs POI, con el método propuesto en [10] y [3]. En las Figuras 5.8 y 5.9 se muestra la comparación de resultados con el método propuesto con cuantización $b = 4$ y $b = 6$, respectivamente, con los obtenidos en 5.8 y 5.9, en el cual se puede apreciar una reducción marcada del MSE compara con [10], en el cual el MSE siempre esta arriba de 10^{-2} para ambos casos. Además se puede ver que el peor caso obtenido en el modelo propuesto es mejor que el mejor de los casos obtenidos en [10]. Además, en 5.10 se puede ver que los resultados obtenidos son mejores al utilizar la transformada discreta de Fourier, sin embargo, la implementación de la transformada de Fourier es más compleja que la implementa-

5. Bancos de Filtros para codificación LDPC

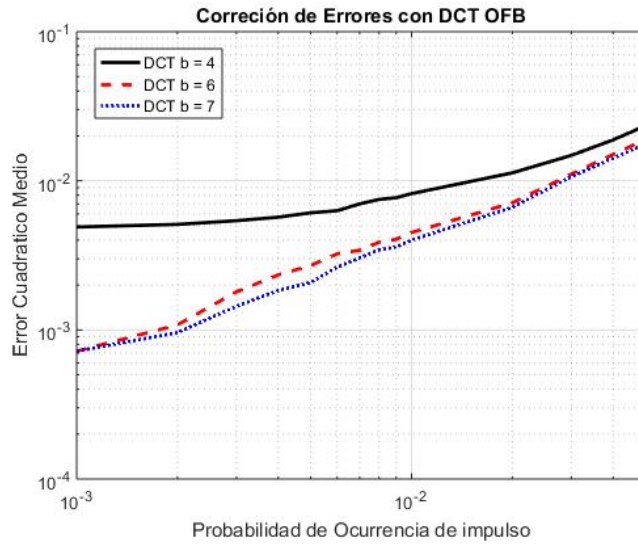


Figura 5.7: Resultados de simulación del modelo con DCT

ción que la transformada discreta del coseno, que solo utiliza números reales para los coeficientes y valores obtenidos.

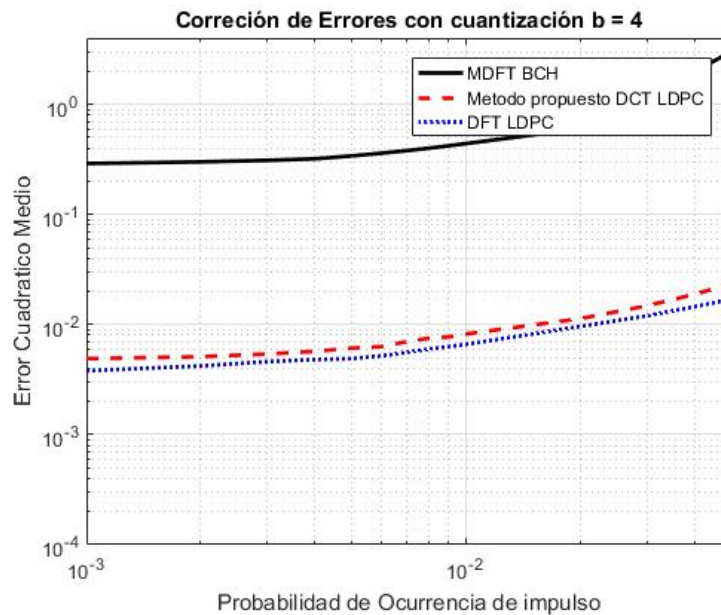


Figura 5.8: Cuantización $b = 4$

5. Bancos de Filtros para codificación LDPC

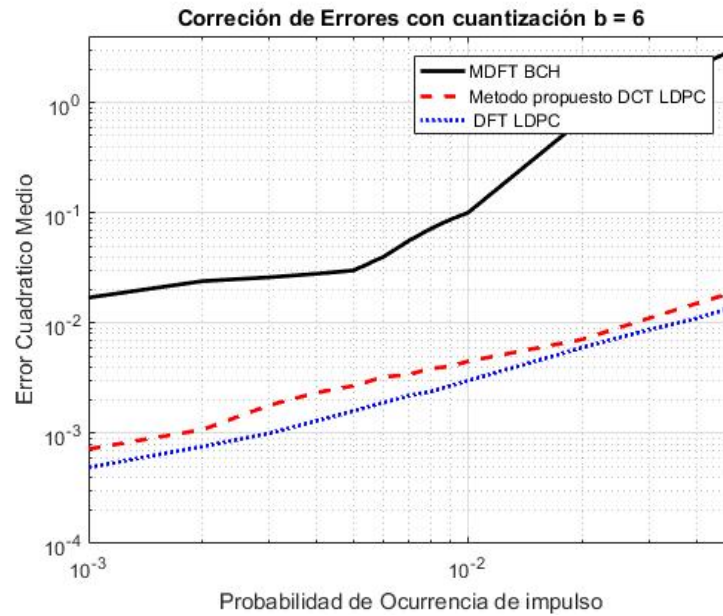


Figura 5.9: Cuantización $b = 6$

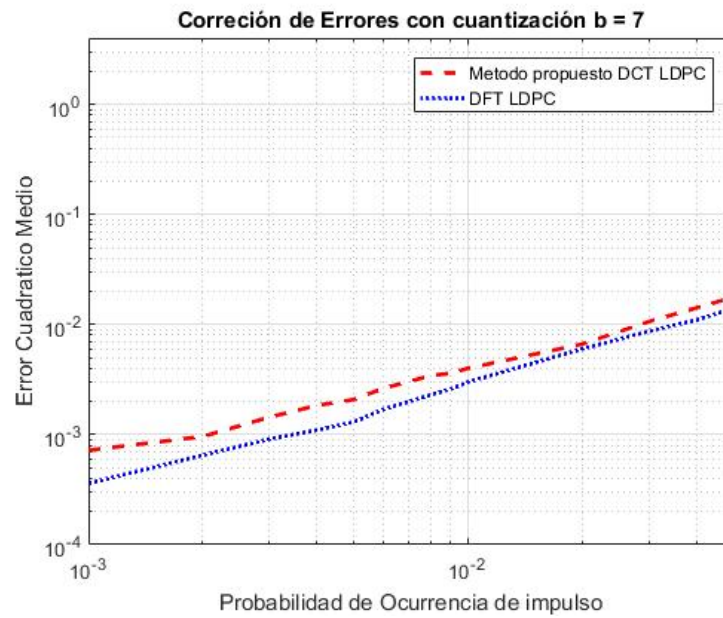


Figura 5.10: Cuantización $b = 7$

Conclusiones y trabajo a futuro

En este trabajo, incentivados por la importancia de los códigos LDPC y las características propias de los bancos de filtros sobremuestreados, se propuso el uso de Bancos de Filtros modulados por Coseno para codificación LDPC con el propósito de disminuir el Error Cuadrático Medio, en presencia de ruido Gaussiano Aditivo Blanco y ruido impulsivo y la utilización de un banco de filtros de valores reales para la disminución de la complejidad de implementación.

Los resultados de las simulaciones son presentados con las gráficas de probabilidad de error contra la probabilidad de ocurrencia de ruido impulsivo. En las gráficas se puede ver los beneficios de la técnica de codificación propuesta en términos de la probabilidad de error.

Las comparaciones del método de codificación propuesto con el método [10] en muestra que el método de codificación propuesto presenta mejores resultados para cualquier caso.

6.1. Trabajo a futuro

- Uno de los problemas observados en el proceso de simulación fue que el tiempo necesario para obtener los resultados de cada parte de la curva de desempeño es muy prolongado. Esta limitante es la motivación de implementar el proceso de simulación en un FPGA, esto con la idea de mejorar el tiempo de simulación.
- Una vez teniendo implementado el proceso, para mejorar aun mas el tiempo, se planean investigar esquemas de cuantización mas efectivos.

Bibliografía

- [1] S. Haykin, *Communication Systems*. John Wiley & Sons, Inc., 4 ed., 2001.
- [2] G. Jovanovic-Dolecek, *Multirate Systems: Design and Applications*. Idea Group Publishing, 2002.
- [3] J. de Jesús Carmona Suárez, “Error control coding and filter banks,” Master’s thesis, INAOE, 2017.
- [4] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ; Prentice-Hall, 1993.
- [5] M. E. Domínguez-Jiménez, G. Sansigre, P. Amo-López, and F. Cruz-Roldán, “Dct type-iii for multicarrier modulation,” in *2011 19th European Signal Processing Conference*, pp. 1593–1597, Aug 2011.
- [6] J. Ogale and A. Jain, “Design of an m-channel cosine modulated filter bank by new cosh window based fir filters,” 2012.
- [7] J. Kliewer and A. Mertins, “Oversampled cosine-modulated filter banks with arbitrary system delay,” *IEEE Transactions on Signal Processing*, vol. 46, pp. 941–955, Apr 1998.
- [8] T. Karp, M. Kieffer, and P. Duhamel, “Parity-check matrix calculation for para-unitary oversampled DFT filter banks,” *IEEE Transactions on Signal Processing*, vol. 56, October 2008.
- [9] D. J. C. MacKay, “Encyclopedia of sparse graph codes,” (revisada en Junio, 2018).
- [10] A. S. Lakshmi, “Oversampled filter banks with built-in error correction capabilities,” Master’s thesis, Texas Tech University, 2012.