# Sistema de Transmisión AM con Codificación DTMF

Informe de Trabajo Integrador


**Procesamiento Digital de Señales**

**Ingeniería en Computación**

**Facultad de Ciencias Exactas y Tecnología**

**Universidad Nacional de Tucumán**

**Autores:**

Boeri, Benjamin
Campero, Leandro
Villafañe, Cristian

20 de agosto de 2023

**Resumen**

El procesamiento digital de señales se refiere a la manipulación, análisis y transformación de señales utilizando algoritmos y técnicas computacionales. En este proyecto, el procesamiento digital de señales se aplica para generar y decodificar los tonos DTMF, así como para simular la transmisión y detección de los dígitos enviados.

Este proyecto presenta la implementación de un sistema de Modulación en Amplitud (AM) y Codificación DTMF utilizando MATLAB/SIMULINK. Se busca transmitir dígitos numéricos codificados en DTMF a través de un enlace cableado simulado. El sistema involucra la generación de tonos DTMF, el diseño de filtros digitales pasa bandas para el decodificador, la configuración de la frecuencia de portadora RF y la modelización del canal de transmisión como un filtro analógico pasa banda.

Un filtro digital es un componente esencial en el procesamiento digital de señales que permite modificar las características de una señal. En este proyecto, se utilizan filtros digitales pasa bandas implementados mediante la técnica del filtro Butterworth. Estos filtros permiten seleccionar y aislar las frecuencias específicas asociadas a los tonos DTMF.

La implementación de los filtros digitales se logra mediante la transformación de los coeficientes del filtro en una representación numérica que se aplica a la señal de entrada. Esto puede lograrse utilizando algoritmos y técnicas de programación, así como también herramientas como MATLAB y SIMULINK.

El resultado de este proyecto demuestra la viabilidad y efectividad de la implementación del sistema propuesto. El procesamiento digital de señales y el uso de filtros digitales son fundamentales en diversas aplicaciones, incluyendo las comunicaciones y el procesamiento de señales de audio.

En resumen, este proyecto combina el procesamiento digital de señales, la modulación AM, el diseño de filtros digitales y la codificación DTMF para lograr la transmisión y detección de dígitos numéricos. La implementación exitosa de este sistema contribuye al avance y comprensión de las técnicas de procesamiento y transmisión de señales en el ámbito de las comunicaciones.

# Índice general

# Capítulo 1

# Introducción

## 1.1. Problema propuesto

La modulación en amplitud o Amplitud Modulada (AM), permite la transmisión de una señal de baja frecuencia superpuesta a una onda de alta frecuencia. Este sistema de modulación permite enviar mensajes en la forma de envolventes de la onda portadora, ya sea por un canal de aire o físico utilizando un enlace cableado.
El sistema de codificación Doble Tonos Múltiples Frecuencias (DTFM), utiliza una combinación de tonos de frecuencia audibles pera representar el conjunto de números del 0 al 9 disponible en el teclado telefónico, con lo cual es posible enviar una codificación numérica por la línea telefónica.
El modelo de trabajo está representado en la Figura 1.1, correspondientes al Modulador y Demodulador AM, el canal de cable telefónico, y las etapas de codificación y decodificación DTFM.


Figura 1.1: Diagrama de bloques general

## 1.2. Objetivo

El objetivo principal de este proyecto integrador es la implementación del sistema mostrado en la Figura 1.1, utilizando MATLAB, SIMULINK, o la combinación de ambos recursos de modelado computacional, para el envío de números (0-9) codificados en DTFM bajo modulación AM, y la detección del número enviado a la salida (uno número cada por vez).
A modo de referencia, el Cuadro 1.1, muestra la combinación de tonos audibles asociados al conjunto numérico, y en el enlace indicado se encuentra la información ampliada sobre la codificación DTFM.

| Frecuencia Baja | Frecuencia Alta | Digito | Frecuencia Final |
|---|---|---|---|
| 697 | 1209 | 1 | 1906 |
| 697 | 1336 | 2 | 2033 |
| 697 | 1477 | 3 | 2174 |
| 770 | 1209 | 4 | 1979 |
| 770 | 1336 | 5 | 2106 |
| 770 | 1477 | 6 | 2247 |
| 852 | 1209 | 7 | 2061 |
| 852 | 1336 | 8 | 2188 |
| 852 | 1477 | 9 | 2329 |
| 941 | 1336 | 0 | 2277 |

Cuadro 1.1: Combinación de tonos audibles (medido en [Hz])

## 1.3. Enunciado

a) A nivel simulación se deberán sintetizar los tonos asociados a cada digito numérico seleccionando una Frecuencia de Muestreo ($f_s$) apropiada (Teorema de Nyquist-Shannon).

b) El demodulador DTFM deberá ser implementado mediante filtros digitales pasa bandas, con un orden y respuestas apropiadas. El modo de indicar cuál fue el digito enviado queda a criterio del grupo de trabajo.

c) Para el modelo de trasmisión AM (enlace cableado) se deberán establecer y sintetizar la frecuencia de portadora $RF$ el índice de modulación apropiados (recordando que $f_s$ es única en todo el sistema).

d) El canal de transmisión se corresponde al de un filtro analógico (transformado a digital) pasa banda con un rango de 300 Hz a 3400 Hz, respuesta plana y orden apropiado. Se considera el rango útil asignado a la frecuencia telefónica, aunque el cable telefónico de cobre tipo AWG-24, por ejemplo, supera este ancho de banda a 1Mz en distancias inferiores a 200 Mts.

## 1.4. Lineamientos Generales

a) El grupo de trabajo deberá cumplir con las especificaciones del proyecto, utilizando criterios de diseños justificados para cada bloque del sistema.

b) Se deberán indicar el paso a paso para el diseño de los filtros digitales utilizados en las diferentes etapas.

c) El criterio de selección para el filtro analógico representativo del canal (Bessel, Butterworth, etc.), y el método de transformación analógico a discreto escogido, brindando una gráfica comparativa de la respuesta en frecuencia resultantes en ambos planos (Laplace y Z).

d) Se pide 3 aplicaciones posibles del sistema desarrollado en aplicaciones de tele comando (por ejemplo, aplicación de sistema de riego por comando telefónico de 3 zonas), y como se imprentaría en la práctica (no el desarrollo, solo la propuesta).

e) Problema de análisis: para el caso de que ocurran fallos en el canal de comunicación (por ejemplo, una atenuación en determinadas frecuencias), analizar la robustez del código detector para al menos 3 zonas atenuadas de frecuencias diferentes. Utilizar el código adjunto en Matlab para el diseño del canal con fallas. Justificar los resultados.

f) Escribir el informe, Incluir conclusiones, observaciones y sugerencias sobre los resultados obtenidos

# Capítulo 2

# Planteamiento

## 2.1. Sistema

Lo que se busca es simular un sistema DTFM cuya señal se transmite a través de un canal telefónico con modulación AM. Tal simulación debe comprender cada uno de los bloques intervinientes en el sistema, como se muestra en la Figura 2.1. Antes de diseñar y planificar la simulación hay que tener en cuenta las frecuencias intervinientes, particularmente hablando de la Frecuencia de Portadora ($f_p$) (que determina la frecuencia de la señal a ser modulada en la transmisión) y la $f_s$ (que determina la cantidad de muestras por segundo para la simulación).

Ya que esta simulación trata de la transmisión en AM a través de un canal telefónico, podemos tomar una $f_s$ utilizada universalmente en sistemas de audio, y esta equivale a 44 [kHz], y podemos ver que claramente cumple con el Teorema de Nyquist-Shannon ya que es mayor al doble de la señal de mayor frecuencia (1477 [Hz], componente alta de los digitos 3, 6 y 9). Para la $f_p$ tomamos 15 [kHz] ya que es 10 veces mayor a la señal antes mencionada y es menor a la mitad de $f_s$ (para poder seguir cumpliendo con el teorema).

A continuación enumeramos el tratamiento de la señal en cada bloque del sistema:

1. Codificador DTMF: Se suman las señales sinusoidales correspondientes a las frecuencias que componen cada digito
2. Modulador AM: Se modula la señal portadora con la señal codificada
3. Cable Telefónico: La señal modulada pasa por un filtro pasa-banda para simular el canal de voz [300-3300][Hz]
4. Demodulador AM: Se bate la señal recibida con la misma portadora para obtener la moduladora
5. Decodificador DTMF: La señal pasa por un banco de filtros pasa-banda para determinar qué señales de la matriz fueron enviadas

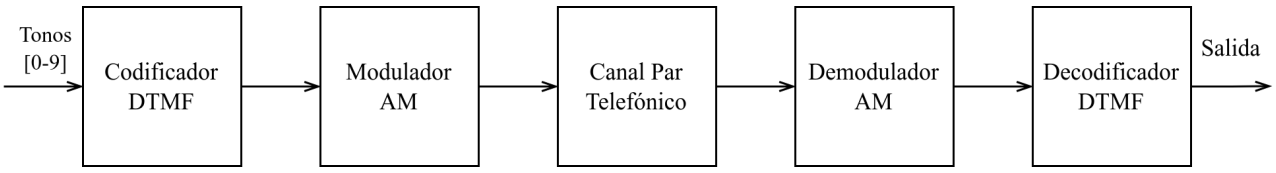


Figura 2.1: Diagrama de bloques a simular

## 2.2. Decodificación

Para decodificar la señal del tono se tiene que implementar un banco de filtros en base a la matriz de señales del sistema DTFM. Esto es, las filas se corresponden con las frecuencias bajas y las columnas con las frecuencias altas. La sumatoria de las señales se corresponde con la codificación del tono propuesto; esta señal llega a la matriz para devolver el tono correspondiente. El bloque decodificador se compone de filtros pasa-banda para cada frecuencia y el bloque detector como muestra la Figura 2.2. La razón de usar un filtro para cada frecuencia baja y alta, en lugar de un filtro por cada frecuencia resultante, se debe a que de esta forma usamos 7 filtros en lugar de 9 (uno por cada tono); además, estas frecuencias (altas y bajas) están más separadas en el espectro que las frecuencias resultantes, lo cual es provechoso a la hora de diseñar un filtro.

La matriz será un bloque lógico que devolverá el digito correspondiente en base a las señales que hayan logrado activarse luego pasar por el banco de filtros.

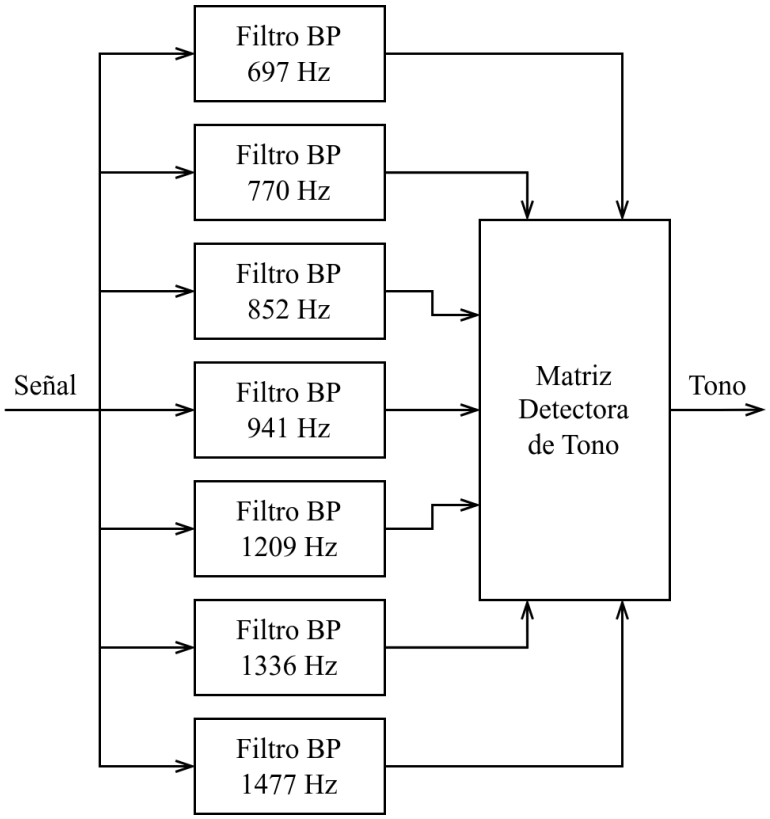


Figura 2.2: Composición del bloque decodificador

## 2.3. Simulación

La simulación de este sistema, que es el objetivo de este proyecto, se logrará a través del uso de distintas herramientas dentro del software MATLAB. Entre estas herramientas se encuentra simulink que sirve para realizar el estudio en el tiempo y frecuencia de las señales a través de cada bloque del sistema.

Cada señal de entrada al sistema será simulada a través de generadores de sinusoidales, mientras que los filtros serán bloques que actuen en base a polinomios obtenidos por librerías provistas por MATLAB. El resto de las herramientas serán provistas por simulink (sumadores, multiplicadores, analizadores de

espectro, etc.)

La razón de usar una herramienta para el diseño de los filtros es que estos se presumen de alto orden, lo cual es engorroso de calcular de manera analítica. Sin embargo, con el objetivo de comprender cómo es el diseño de filtros digitales, se explicará en el siguiente capítulo cómo es el proceso de diseño y análisis correspondiente.

# Capítulo 3

# Filtros Digitales

En este capítulo veremos como es el diseño de un filtro digital a partir de uno analógico, y para ello utilizaremos lo que se conoce como **Transformación Bilineal**. Esta transformación mapea el plano complejo del dominio de frecuencia analógico al dominio de frecuencia digital utilizando una relación no lineal.

Es importante tener en cuenta que la transformación bilineal introduce distorsiones en la respuesta en frecuencia del filtro analógico original al convertirlo en un filtro digital, como se muestra en la Figura 3.1. Estas distorsiones son inevitables debido a la naturaleza no lineal de la transformación. Sin embargo, en muchos casos, estas distorsiones son aceptables y se pueden compensar mediante técnicas de diseño adicionales. Una de estas técnicas es el **Pre-Warping**.

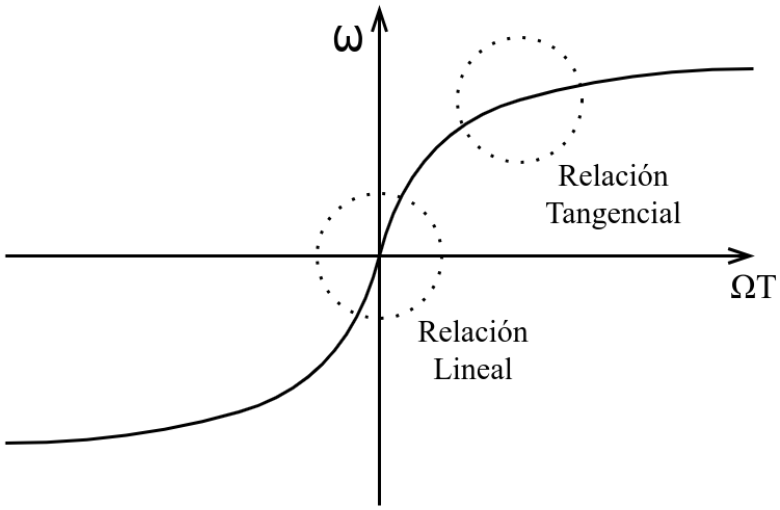


Figura 3.1: Distorsion de frecuencia introducida por la Transformación Bilineal

El pre-warping es un ajuste realizado a la Frecuencia de Corte ($f_c$) del filtro analógico antes de aplicar la transformación. Se hace con el fin de compensar la distorsión introducida por la transformación bilineal en las frecuencias más altas, esto es debido a la no linealidad de la función tangente utilizada en la transformación. El pre-warping ajusta la $f_c$ del filtro analógico antes de aplicar la transformación bilineal para compensar esta distorsión. Se utiliza una función que aproxima la inversa de la función tangente para ajustar la $f_c$ en el dominio analógico, como se muestra en la Ecuación 3.1 [RG95]. Para más información acerca de la Transformación Bilineal se pueden consultar en las siguientes bibliografías: [PM96], [OS89] y [OS99].

$$\Omega = \frac{2}{T} \tan \frac{\omega T}{2} = 2 f_s \tan \left( \pi \frac{f}{f_s} \right) \tag{3.1}$$

## 3.1. Diseño

Primero, se diseña el filtro analógico deseado utilizando un prototipo como el filtro Butterworth, Chebyshev o el Elíptico. En este caso usaremos el filtro Butterworth, que se caracteriza por tener una respuesta en frecuencia lo más plana y suave posible en la banda de paso, lo que significa que atenúa las frecuencias no deseadas de manera gradual y sin oscilaciones abruptas, como describre la Figura 3.2.

El filtro de Butterworth más típico es el filtro pasa bajo de primer orden, el cual puede ser modificado a un filtro pasa banda a traves transformaciones que veremos más adelante, o añadir en serie otros formando un filtro pasa banda o elimina banda y filtros de mayores órdenes. La función transferencia para un filtro prototipo Butterworth pasa-bajos que corta en 1rad/s se describe en la Ecuación 3.2. Si deseamos diseñar un filtro pasa-altos, o pasa-banda o de rechaza-banda, basta con partir del filtro paso bajo prototipo y realizar una transformación en frecuencia.

$$H(s) = \frac{1}{s+1} \tag{3.2}$$

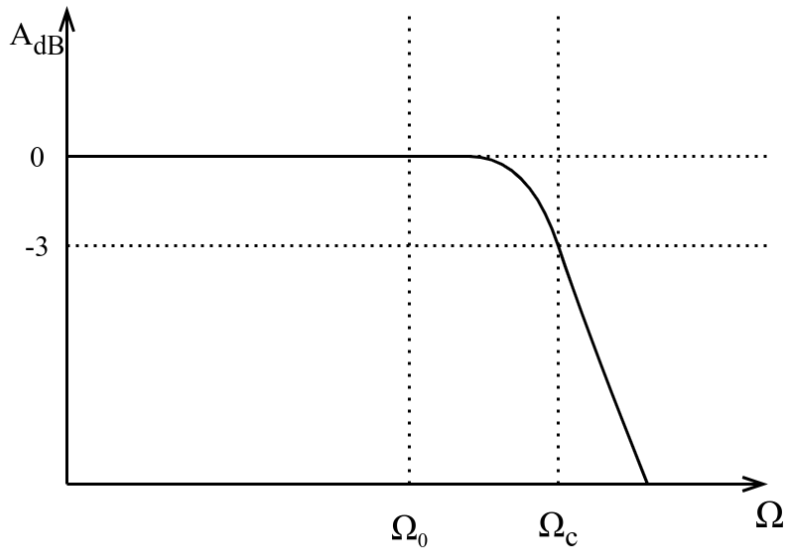


Figura 3.2: Respuesta en frecuencia de un filtro pasa-bajos

Según [RG95] existen dos enfoques distintos para el diseño de filtros digitales de tipo pasa-banda, pasa-alto y rechaza-banda. Estos enfoques se resumen en la Figura 3.3. Estos dos enfoques difieren en que la técnica 1 transforma un filtro de paso bajo analógico normalizado en otro filtro analógico que se puede digitalizar para obtener el filtro digital deseado, mientras que la técnica 2 digitaliza el filtro de paso bajo normalizado inmediatamente y luego aplica una transformación de banda de frecuencia digital para obtener el filtro digital deseado. Dado que ya hemos discutido cómo diseñar el filtro de paso bajo normalizado y cómo digitalizarlo, en esta sección consideraremos las transformaciones de banda de

frecuencia apropiadas para los casos continuo y discreto. Para nuestro diseño vamos a considerar la técnica 1, que consiste en pasar el filtro prototipo pasa-bajos a un pasa-banda para luego digitalizarlo utilizando la transformación bilineal.

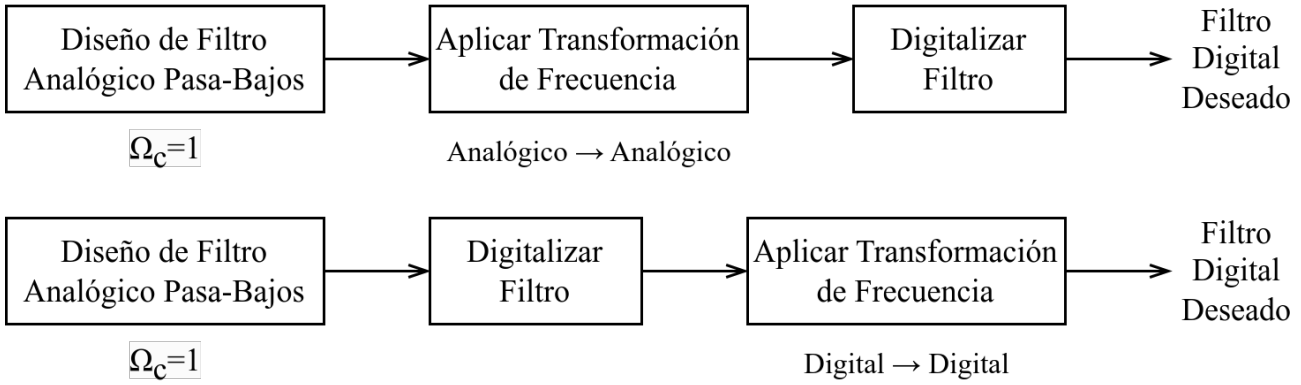


Figura 3.3: Enfoques para el diseño de un filtro digital

Existe una gran variedad de técnicas para transformar un filtro pasa-bajas de corte 1 rad/s en otro filtro pasa-bajas (con diferente $f_c$) o en un filtro pasa-banda, pasa-altas o pasa-banda. A continuación se presenta un conjunto de transformaciones especialmente sencillas.

$$s \to \frac{s}{\Omega_h} \qquad \text{pasa-bajos} \to \text{pasa-bajos} \tag{3.3}$$

$$s \to \frac{\Omega_l}{s} \qquad \text{pasa-bajos} \to \text{pasa-altos} \tag{3.4}$$

$$s \to \frac{s^2 + \Omega_l \Omega_h}{s(\Omega_h - \Omega_l)} \qquad \text{pasa-bajos} \to \text{pasa-banda} \tag{3.5}$$

$$s \to \frac{s(\Omega_h - \Omega_l)}{s^2 + \Omega_l \Omega_h} \qquad \text{pasa-bajos} \to \text{rechaza-banda} \tag{3.6}$$

$\Omega_l$ : $f_c$ inferior
$\Omega_h$ : $f_c$ superior

Cómo en este diseño vamos a utilizar la transformación de un pasa-bajos a un pasa-banda utilizaremos la Ecuación 3.5. Para utilizar el método, necesitamos establecer las frecuencias intervinientes: $f_c$ superior e inferior, que las calculamos a partir de la elección de una frecuencia central. Para ello tomamos la frecuencia baja de los digitos 1, 2 y 3 (697 [Hz], Cuadro 1.1) que resulta ser la frecuencia más baja del espectro de este dominio. Las frecuencias de corte serán 60 [Hz] por encima y por debajo de la frecuencia central. Recordemos que para utilizar la transformación bilineal necesitamos compensar la distorsion de frecuencia, y para ello debemos realizar el pre-warping. Entonces las frecuencias angulares de corte analógicas superior e inferior estan dadas por la Ecuación 3.7 y la Ecuación 3.8 respectivamente, donde $f_s$ es igual a 44 [kHz].

$$\Omega_h = 2 f_s \tan\left(\pi \frac{f_c + 60}{f_s}\right) = 4761,01 \left[\frac{\text{rad}}{\text{s}}\right] \tag{3.7}$$

$$\Omega_l = 2f_s \tan\left(\pi \frac{f_c - 60}{f_s}\right) = 4005,15 \left[\frac{\text{rad}}{\text{s}}\right] \tag{3.8}$$

El siguiente paso es encontrar función transferencia realizando la transformación de frecuencia antes mencionada. En la Ecuación 3.11 podemos ver variables nuevas: $BW$ (*Band-Width* o Ancho de Banda) y $\Omega_0$ que es la media geométrica del ancho de banda ($\sqrt{\Omega_h \Omega_l}$). Hacemos este reemplazo por motivos de simplicidad y para tener una mejor comprensión de la función transferencia.

$$H_{LP}(s) = H_{BP}\left(\frac{s^2 + \Omega_l\Omega_h}{s(\Omega_h - \Omega_l)}\right) \tag{3.9}$$

$$H(s) = \frac{1}{\frac{s^2+\Omega_l\Omega_h}{s(\Omega_h-\Omega_l)} + 1} \tag{3.10}$$

$$H(s) = \frac{(\Omega_h - \Omega_l)s}{s^2 + (\Omega_h - \Omega_l)s + \Omega_h\Omega_l} = \frac{BWs}{s^2 + BWs + \Omega_0^2} \tag{3.11}$$

$$H(s) = \frac{755,85s}{s^2 + 755,85s + 19,07 \times 10^6} \tag{3.12}$$

$$H(s) = \frac{s}{0,00132s^2 + s + 25220} \tag{3.13}$$

Para demostrar que este filtro diseñado es en realidad un pasa-banda que corta en las frecuencias angulares especificadas, utilizamos MATLAB para graficar la respuesta en frecuencia del mismo (Código 3.1). En la Figura 3.4 podemos ver que la respuesta en frecuencia es la esperada, en los puntos marcados aproximadamente en y(0,707) las frecuencias se aproximan a $\Omega_l$ y $\Omega_h$. Por lo tanto queda demostrado que el filtro analógico diseñado es realmente un pasa-banda.

```
a = [0 1 0];
b = [1.32e-3 1 25.22e3];
[h, w] = freqs(a, b, 5000);
phase = angle(h);
phasedeg = phase * 180 / pi;
mag = abs(h);

subplot(2, 1, 1)
loglog(w, mag)
grid on
xlabel('Frecuencia (rad/s)')
ylabel('Magnitud')
xlim([3500, 5500])
ylim([0.2, 1.2])

subplot(2, 1, 2)
semilogx(w, phasedeg)
grid on
xlabel('Frecuencia (rad/s)')
ylabel('Fase (grados)')
xlim([3500, 5500])
ylim('auto')
```

Código fuente 3.1: Graficar respuesta en frecuencia de Filtro Analógico

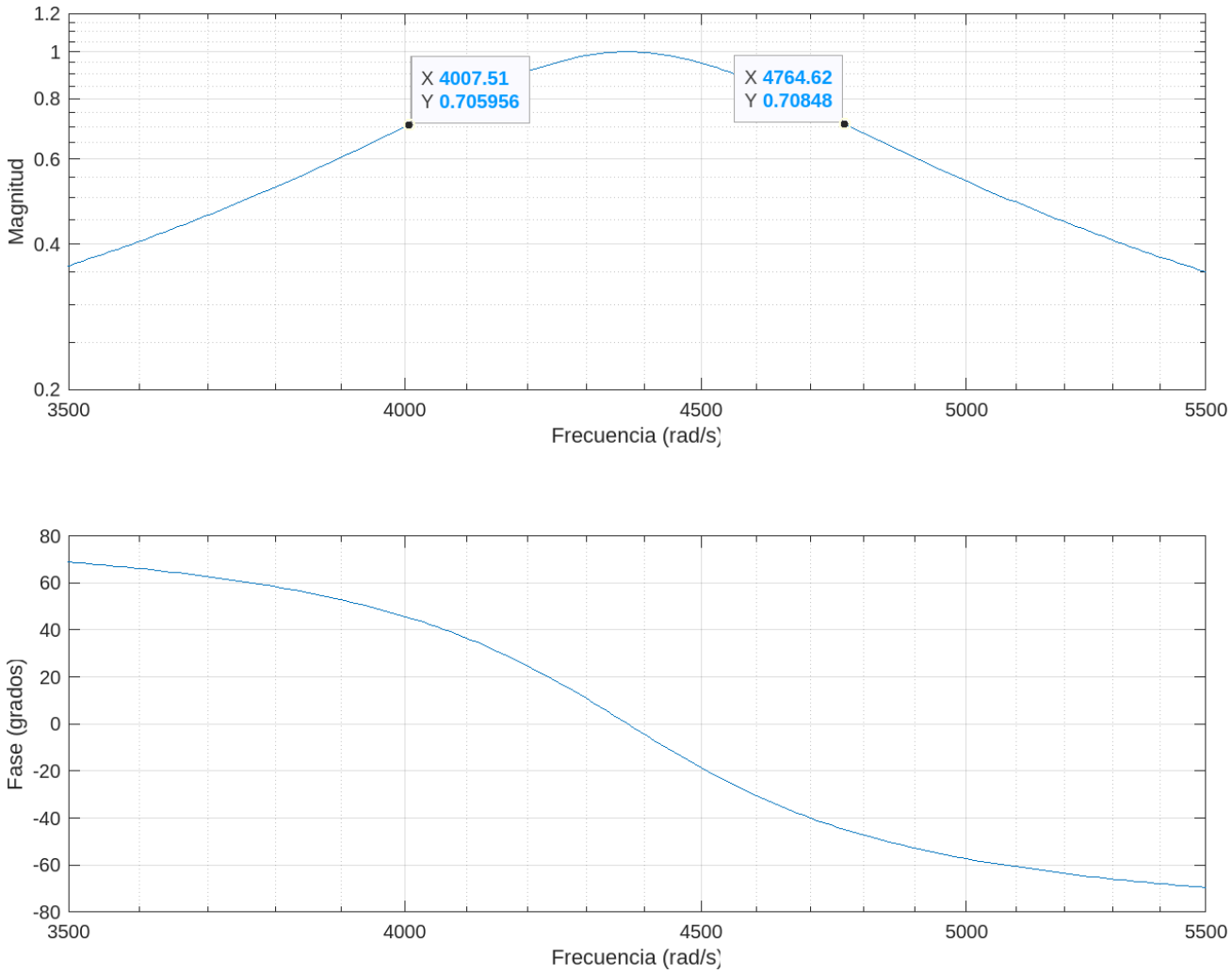


Figura 3.4: Respuesta en frecuencia del filtro analógico

## 3.2. Conversión a digital

Ahora que tenemos nuestro filtro analógico, el siguiente paso es transformarlo a digital a tarvés de la Transformación Bilineal. Para realizarla consideramos la correspondencia entre el plano s y el plano z, dada por la Ecuación 3.14. Luego particularizamos la función transferencia de nuestro filtro pasa-banda analógico para la correspondencia antes mencionada, trabajamos algebráicamente (ver Apéndice A.1) y encontramos H(z), que está dado por la Ecuación 3.17.

$$s = \frac{2}{T}\frac{(1 - \mathrm{z}^{-1})}{(1 + \mathrm{z}^{-1})} \tag{3.14}$$

$$H(\mathrm{z}) = H(s)\big|_{s=\frac{2}{T}\frac{(1-\mathrm{z}^{-1})}{(1+\mathrm{z}^{-1})}} \tag{3.15}$$

$$H(\mathrm{z}) = \frac{BW\left(\frac{2}{T}\frac{(1-\mathrm{z}^{-1})}{(1+\mathrm{z}^{-1})}\right)}{\left(\frac{2}{T}\frac{(1-\mathrm{z}^{-1})}{(1+\mathrm{z}^{-1})}\right)^2 + BW\left(\frac{2}{T}\frac{(1-\mathrm{z}^{-1})}{(1+\mathrm{z}^{-1})}\right) + \Omega_0^2} \tag{3.16}$$

$$H(\mathrm{z}) = \frac{8,49 \times 10^{-3} - 8,49 \times 10^{-3}\mathrm{z}^{-2}}{1 - 1,97\mathrm{z}^{-1} + 0,98\mathrm{z}^{-2}} \tag{3.17}$$

Para demostrar que el filtro digital resultante funciona correctamente, utilizamos MATLAB para graficar la respuesta en frecuencia del mismo (ver Código 3.2). En la Figura 3.5 podemos ver que la respuesta en frecuencia es la esperada. En los puntos marcados aproximadamente en y(0,707) las frecuencias se aproximan a $f_l$ y $f_h$, que eran 60 [Hz] por debajo y por encima respectivamente de la frecuencia central 697 [Hz].

```
Fs = 44000;
Ts = 1 / Fs;
num = [8.4953e-3 0 -8.4954e-3];
den = [1 -1.9732 0.9830];

[h, w] = freqz(num, den, 20000);
f = w / (2 * pi);
plot(f * (Fs / 2) / max(f), abs(h))
title('Filtro Butterworth')
xlabel('Hertz')
ylabel('Magnitud')
xlim([0, 1200])
ylim('auto')
```

Código fuente 3.2: Graficar respuesta en frecuencia de Filtro Digital

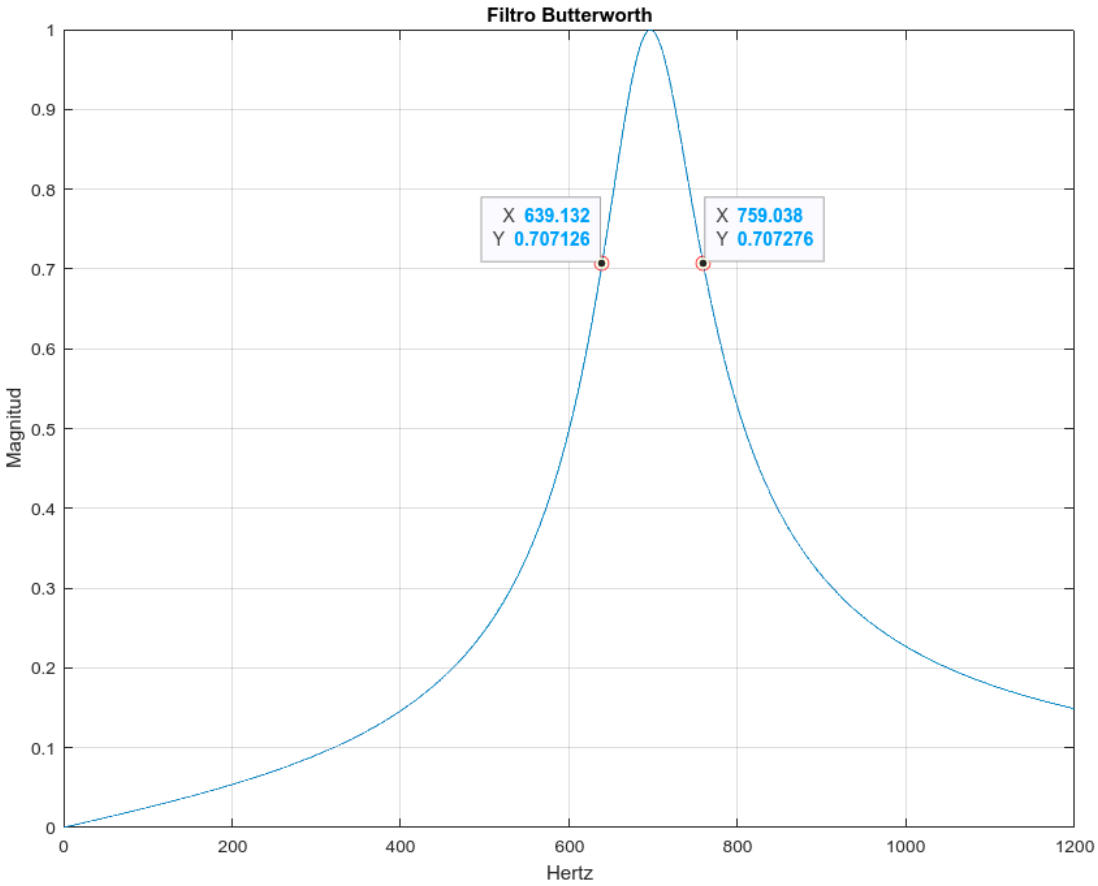


Figura 3.5: Respuesta en frecuencia del filtro digital

## 3.3. Conclusiones

En las secciones anteriores demostramos cómo es el procedimiento para el diseño de un filtro digital pasa-banda a partir de transfomaciones aplicadas a un filtro analógico pasa-bajos. Analizamos las respectivas respuestas en frecuencia de los filtros analógicos y digitales obtenidos, y podemos observar que cumplen con el requisito de filtrar aquellas frecuencias que se encuentren fuera de la banda especificada [637 [Hz] ; 757 [Hz]]. Sin embargo este proceso fue solamente para un filtro, una banda de frecuencia para una frecuencia central en particular que es la primera del espectro de la matriz DTFM. La siguiente frecuencia en el espectro es 770 [Hz], y si observamos la Figura 3.5 podemos ver que no está muy lejos de la $f_c$ superior del filtro diseñado. Esto va a devenir en un solapamiento de señales, lo cual puede introducir falsos disparos a la hora de implementar la lógica para detectar el tono. Queda claro que se necesita un filtro pasa-banda pero de mayor orden y una banda más reducida. El objetivo es definir un Frecuencia de Atenuación o Paso ($f_a$) superior e inferior, y en las que el filtro atenúe en 10 [dB], y estas frecuencias tienen que ser antes de solapar con la frecuencia siguiente o anterior del espectro de la matriz. En la Figura 3.6 podemos ver cómo sería un filtro ideal pasa-banda que cumple con esos requisitos.

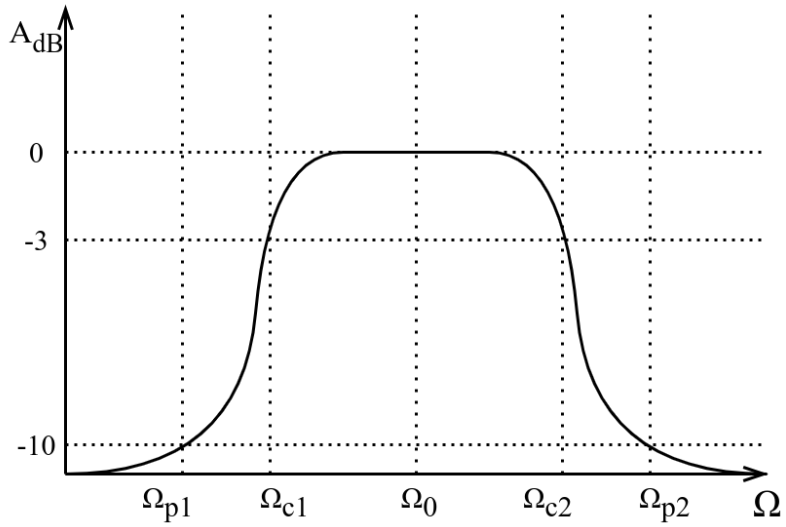


Figura 3.6: Respuesta en frecuencia de un filtro pasa-banda con frecuencias de parada

Podemos calcular el orden del filtro pasa-bajo necesario para cumplir con esas características, luego sabremos que el filtro pasa-banda resultante será el doble de ese orden. Pero antes debemos definir las frecuencias críticas involucradas. A diferencia del filtro anterior, vamos a definir $f_c$ 40 [Hz] por encima de la frecuencia central 697 [Hz], y luego $f_a$ será 20 [Hz] por encima de $f_c$. Pero estas frecuencias son digitales, debemos calcularlas con la distorsión de la Transformación Bilineal (Pre-Warping) para resolver el orden. En las Ecuaciones 3.18 y 3.19 podemos ver las frecuencias analógicas resultantes. Luego, establecemos que la atenuación al final de la banda de transición y comienzo de la banda de rechazo (en $f_a$) es de 10 [dB]. Con estos datos podemos obtener el orden mínimo para que se requiere para el filtro pasa-bajos analógico. De la Ecuación 3.20 vemos que el valor es 41, que es mucho mayor que el orden 1 que utilizamos en el diseño a partir del filtro prototipo.

$$\Omega_c = \frac{2}{T}\tan\frac{\omega_c T}{2} = 2f_s \tan\left(\pi\frac{f_c}{f_s}\right) = 4634,99\left[\frac{\text{rad}}{\text{s}}\right] \tag{3.18}$$

$$\Omega_a = \frac{2}{T}\tan\frac{\omega_a T}{2} = 2f_s \tan\left(\pi\frac{f_a}{f_s}\right) = 4761,01\left[\frac{\text{rad}}{\text{s}}\right] \tag{3.19}$$

$$n = \frac{\log\left(10^{\frac{A_{dB}}{10}}-1\right)}{2\log\left(\frac{\Omega_a}{\Omega_c}\right)} = 40,95 \approx 41 \tag{3.20}$$

Con esto demostramos y concluímos que para obtener un filtro más preciso en la simulación, necesitamos que sea de mayor orden. Dado que el análisis es engorroso y debemos repetirlo por cada filtro del banco de filtros a diseñar, vamos a utilizar librerías provistas por MATLAB para obtener el filtro digital con el orden necesario para garantizar (o aproximarse lo mayor posible) los resultados esperados a partir de las especificaciones realizadas.

# Capítulo 4

# Desarrollo

Recordemos de la Figura 2.1 que nuestro sistema está compuesto por 5 bloques. En este capítulo vamos descomponer y analizar cada bloque para su posterior implementación en la simulación.

## 4.1. Análisis

### Codificador y Modulador

Estos bloques son en realidad bastante simples. Consta de la sumatoria de dos señales sinusoidales. Para la simulación serían dos generadores establecidos en la combinación de frecuencias que corresponden a un determinado digito. Para crear la codificación, basta con ingresar ambas señales a un sumador. Esto luego debe sumarse a una señal constante de 1, con el fin de poder hacer el producto de la señal resultante con la señal portadora. El resultado de esto es la señal ya modulada, como se muestra en la Figura 4.1.

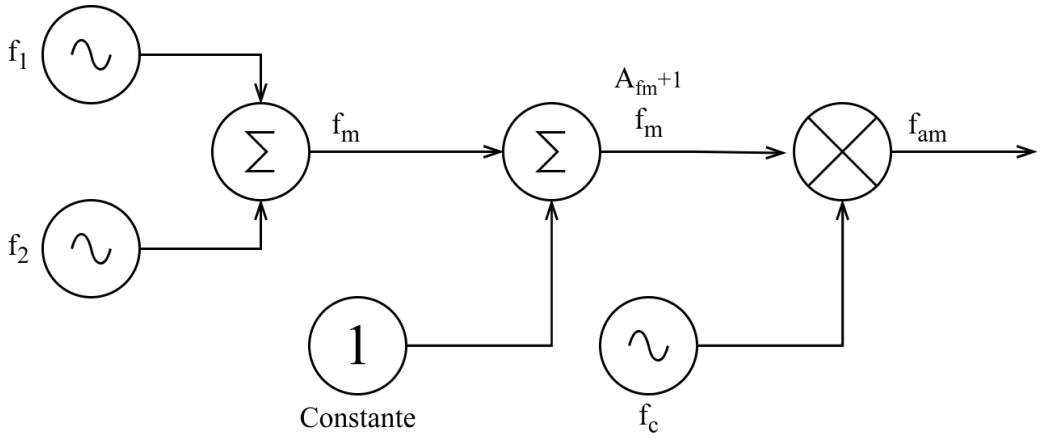


Figura 4.1: Codificador y Modulador

### Transmisión

La transmisión será simulada a través de un filtro pasa-banda, estableciendo las frecuencias de corte de tal forma que el ancho de banda sea el espectro audible por el oído humano, como muestra la Figura 4.2.

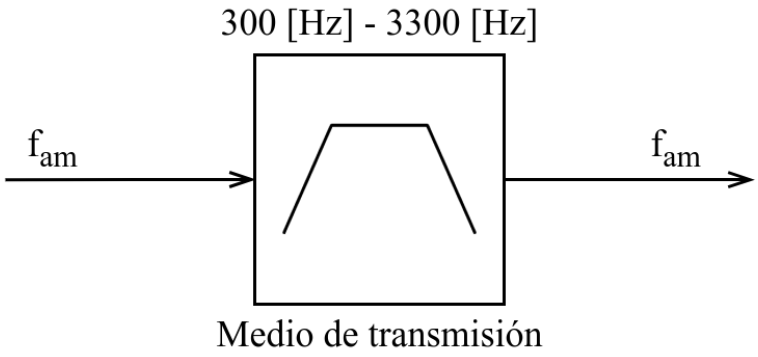


Figura 4.2: Transmisión

**Demodulador**

Para demodular la señal, basta con realizar nuevamente el producto con la señal portadora. Como resultado obtenemos la frecuencia moduladora (que es la señal codificada, suma de las frecuencias que componen a un digito específico) como muestra la Figura 4.3. Luego debemos aplicar un filtro pasa bajos para limpiar la señal de ruidos que puedan haberse introducido, entre esos, algunos vestigios de la señal portadora.

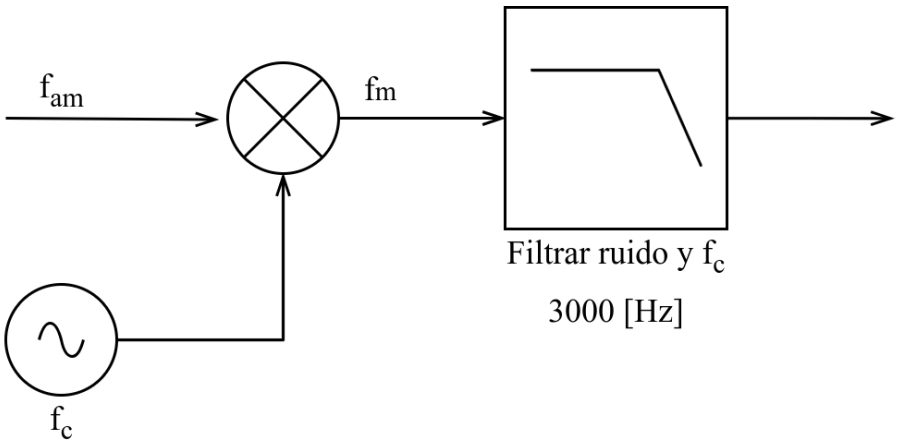


Figura 4.3: Transmisión

**Decodificador**

El bloque decodificador es el más complejo de todos, ya que este tiene la lógica para detectar las señales que componen la señal codificada. Como ya vimos en la Figura 2.2, necesitamos 7 filtros pasa-banda para aislar cada una de las frecuencias de la matriz DTFM, luego viene la matriz decodificadora. En la Figura 4.4 vemos una simplificación de cómo estaría compuesta esta lógica de decodificación. Cada salida de control es una compuerta AND que se activara cuando sus dos entradas se encuentren activas (o en ”1”lógico). Entonces cada compuerta representa la combinación de tonos para detectar cuál fué el digito enviado.

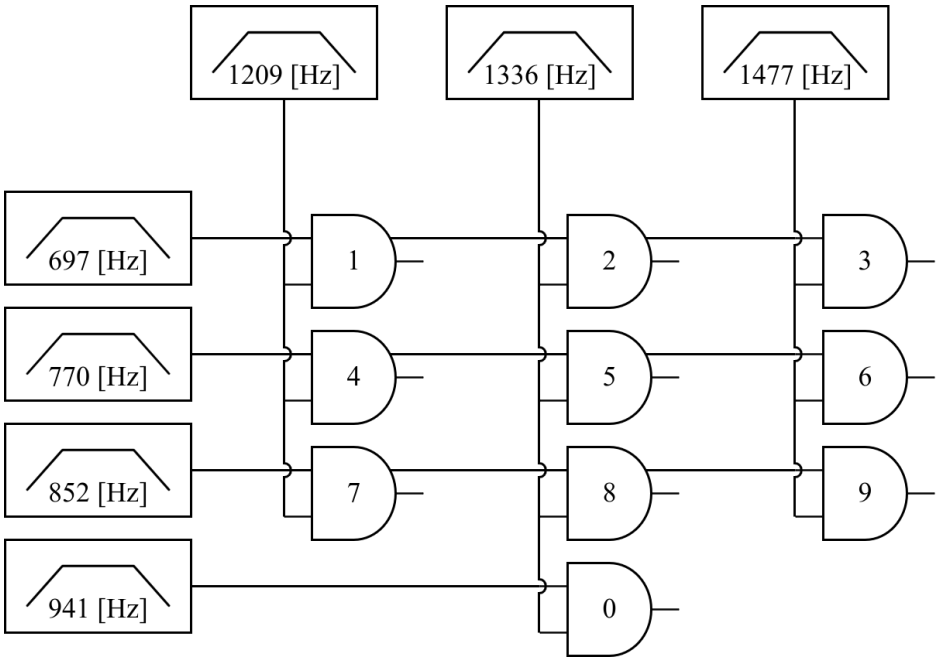


Figura 4.4: Decodificador

## 4.2. Diseño

Como mencionamos en la sección anterior, para poder implementar los filtros digitales del banco de filtros (bloque decodificador) necesitamos librerías de MATLAB para generarlos, ya que estos filtros serán de alto orden lo que es engorroso para el cálculo analítico. Para hacer uso de estas librerías se crearon 2 *scripts* con el objetivo de automatizar la generación de los filtros en base a parámetros de entrada. En el Código 4.1 podemos ver que se encarga de calcular las frecuecias de corte (inferior y superior) para cada frecuencia central provista, bajo una determinada frecuencia de muestreo y orden específico, esto para el banco de filtros del decodificador. Además de eso, también podemos ver que genera las gráficas para analizar la respuesta en frecuencia de cada filtro, como muestra la Figura 4.5. Luego en el Código 4.2 tenemos el algorítmo principal, el cual establece las especificaciones generales del sistema, llama a la función para crear el banco de filtros y además crea el resto de los filtros involucrados como el que representa el canal de transmición y el filtro para eleminar la portadora del espectro de trabajo (en la fase de demodulación).

Se puede notar en el Código 4.2 que establecemos el orden de los filtros en 6 (para pasa-bajos y/o pasa-altos; se interpreta 12 para pasa-banda). Este valor arbitrario, contrario a los cálculos analíticos del capítulo anterior, es empírico; durante varias pruebas de simulación, los filtros pasa-banda de alto orden (20 aproximadamente) demostraban comportamientos inesperados y no concluyentes a la hora de filtrar señales específicas. Por eso, luego de varias pruebas encontramos que el orden 6 era suficiente para realizar la simulación con resultados favorables y realistas.

```
function filtros = banco_decodificador(muestreo, frecuencias, orden)
    % Generador de coeficientes de polinomio de filtros BP Butterworth

    % Crear una figura para la gráfica de respuesta en frecuencia
    figure;

    % Crear variables para almacenar las respuestas en frecuencia
    % y las frecuencias
    all_responses = [];
    all_frequencies = [];

    for idx = 1:length(frecuencias)
        freq = frecuencias(idx);

        % Establecer frecuencias de corte
        fc_low = freq - 40;
        fc_high = freq + 40;

        % Establecer coeficientes para el filtro pasa-banda actual
        [a, b] = butter(orden / 2, [fc_low fc_high] / (muestreo / 2));

        % Generar nombre de variable correspondiente
        % a cada arreglo de coeficientes
        temp_num = strcat('f_', num2str(freq), '_num');
        temp_den = strcat('f_', num2str(freq), '_den');

        % Guardar coeficientes
        filtros.(temp_num) = a;
        filtros.(temp_den) = b;

        % Calcular la respuesta en frecuencia
        [h, w] = freqz(a, b, 10000, muestreo);

        % Agregar los datos a las variables de almacenamiento
        all_responses = [all_responses, abs(h)];
        all_frequencies = [all_frequencies, w];

    end

    % Graficar todas las respuestas en frecuencia en una misma gráfica
    plot(all_frequencies, all_responses);
    title('Respuestas en Frecuencia de Filtros Butterworth Pasa-Banda');
    xlabel('Frecuencia (Hz)');
    ylabel('Magnitud');
    grid on;
    legend(cellstr(num2str(frecuencias', 'Filtro en %d Hz')));

    % Ajustar el rango del eje x
    xlim([500, 1700]);
end
```

Código fuente 4.1: Banco Decodificador

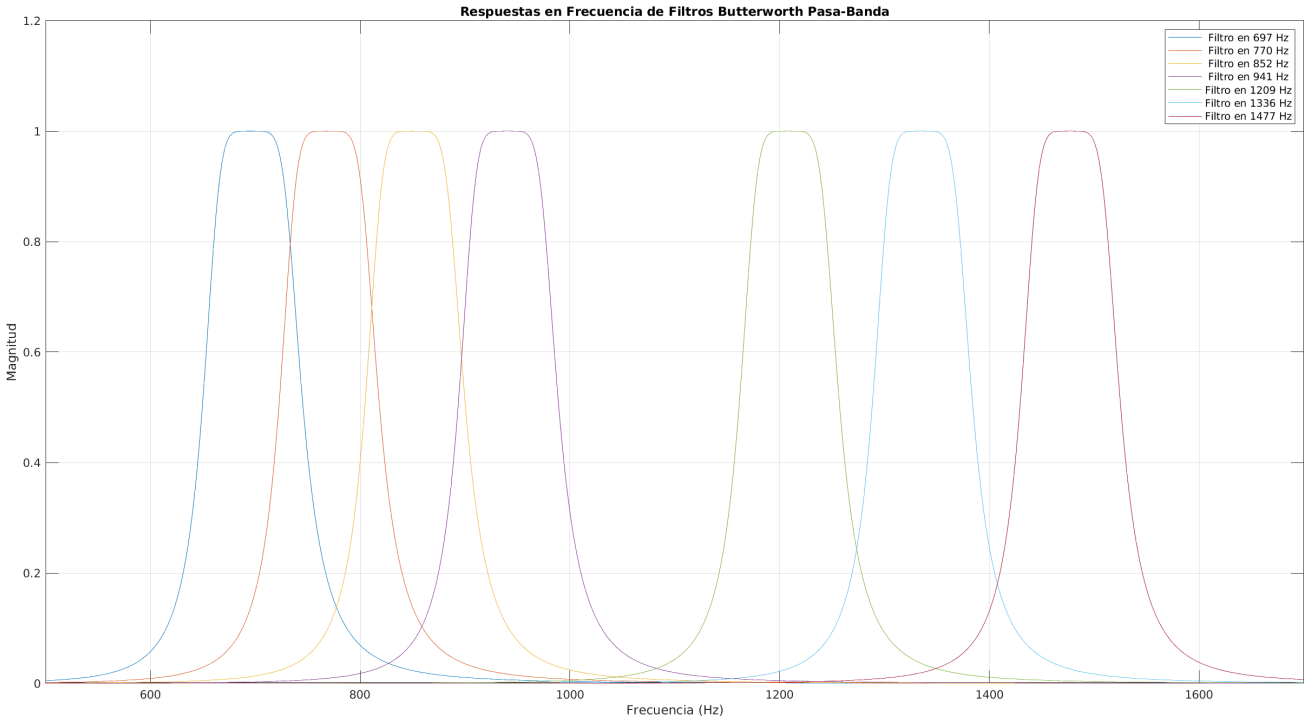


Figura 4.5: Respuesta en frecuencia del Banco de Filtros

```
clc
close all
clear

% Frecuencia de muestreo
Fs = 44000;

% Arreglo con frecuencias centrales a filtrar
frecuencias_centrales = [697, 770, 852, 941, 1209, 1336, 1477];

% Orden de cada filtro pasa-banda del banco de filtros
orden = 6;

% Coeficientes del filtro pasa-bajos para filtrar la frecuencia portadora
[f_portadora_num, f_portadora_den] = butter(orden, 1500 / (Fs / 2));

% Coeficientes del filtro pasa-banda para emular el canal
[f_canal_num, f_canal_den] = butter(2, [300, 3500] / (Fs / 2));

% Estructura de datos para generar y guardar
% los coeficientes del banco de filtros
filtros = banco_decodificador(Fs, frecuencias_centrales, orden)
```

Código fuente 4.2: Algorítmo principal

## 4.3. Prototipo

En base a los resultados obtenidos de los *scripts* diseñamos el sistema completo, en los que cada bloque se alimenta de los datos resultantes. En la Figura 4.6 podemos ver los bloques intervinientes en la primera parte del sistema, esta incluye la selección de los tonos (inferior y superior) para luego sumarlos y lograr la codificación DTFM del número 5 en este caso. Luego pasamos a los bloques intervinientes en la modulación AM de las señales, sumando antes una constante 1 para poder realizar el producto con la señal portadora. Una vez obtenida la señal modulada en AM, esta se transmite por el canal de modulación, que según las especificaciones tiene el ancho de banda del espectro audible por el oído humano. Luego llega a la etapa de demodulación, en la que la señal se vuelve a batir (producto) con la portadora, y el resultado es una señal que tiene una componente en la frecuencia de la portadora y debe ser filtrada, por ello utilizamos un filtro pasa-bajos con frecuencia de corte 3[kHz]. Pasado este filtro, la señal obtenida es casi identica a la sumatoria de los dos tonos.

La segunda parte del sistema comprende la decodificación DTFM a través de un banco de filtros y una matriz decodificadora, como se muestra en la Figura 4.7. Primero debemos aislar las señales por tonos diferenciados, esto lo hace el banco de filtros digitales. Cada uno de estos es un filtro pasa-banda con frecuencia central en uno de los 7 tonos, de esta forma logramos aislar cada señal. Dado que estas son sinusoidales, necesitamos calcular el valor efectivo de las mismas para poder operar lógicamente ellas, entonces se coloca un bloque que realiza el cálculo. Luego, el valor obtenido es un número decimal, del cual nos interesa la parte entera, ya que con este dato vamos a validar la amplitud con la que la señal sale del filtro. La razón de hacer este procedimiento es que para poder comparar lógicamente las señales presente para determinar qué tono fue codificado y enviado, lo cual se explica a continuación.

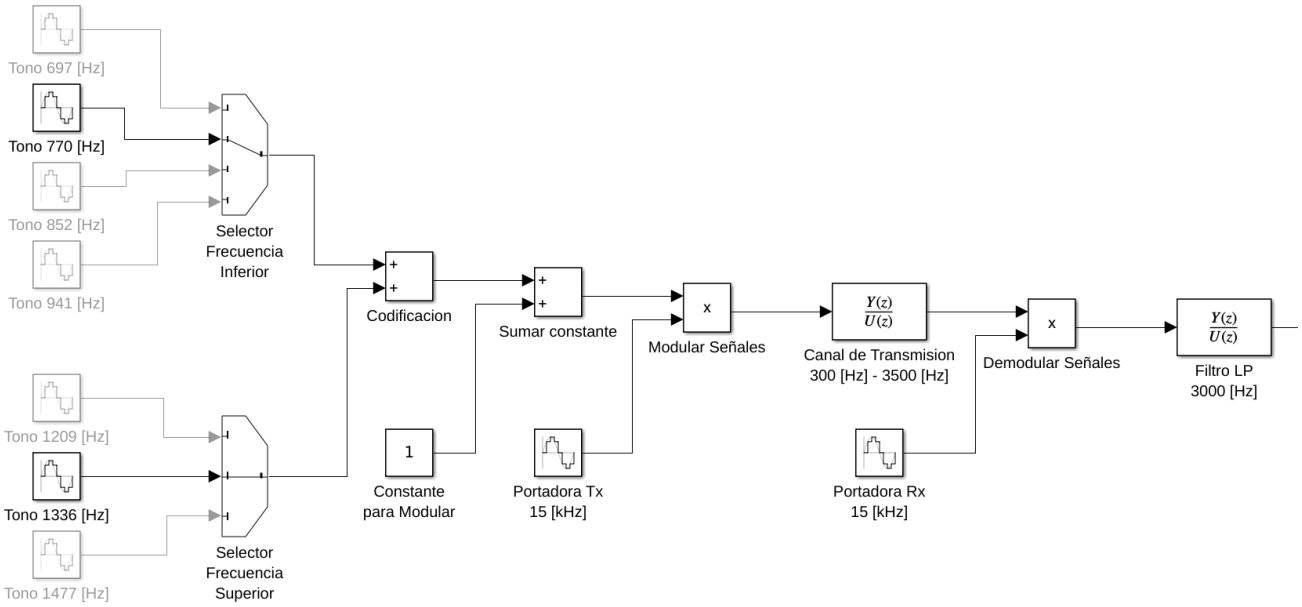


Figura 4.6: Codificación, Modulación, Transmisión y Demodulación

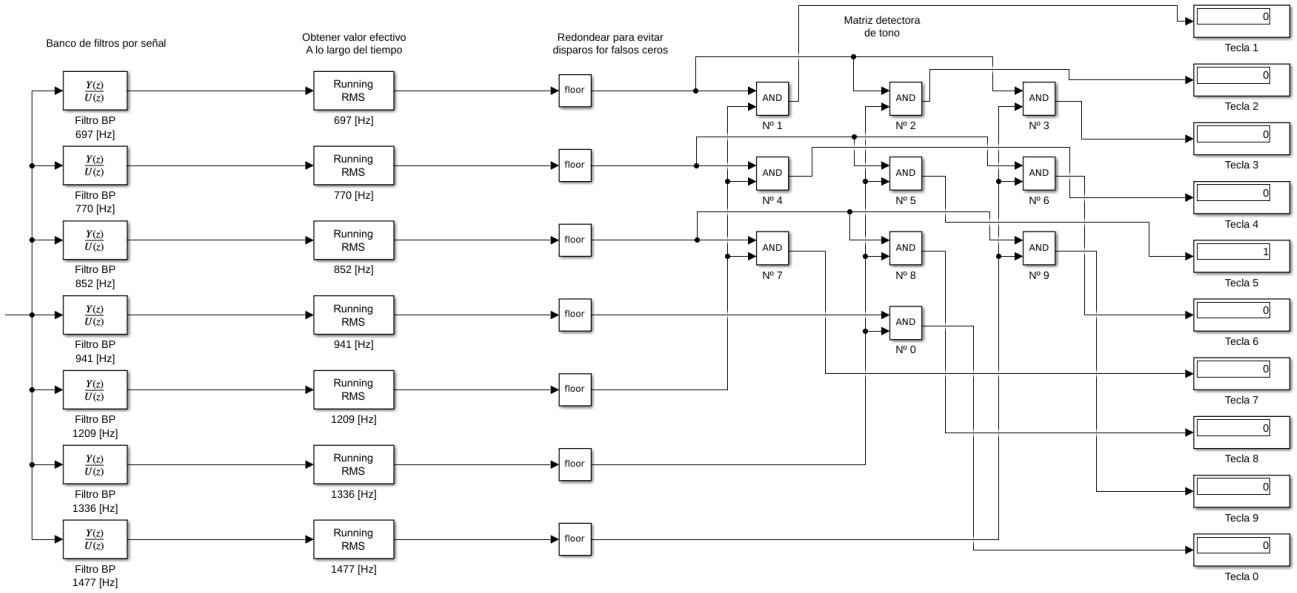


Figura 4.7: Banco de filtros y Decodificador

La matriz decodificadora se compone de bloques de operación lógica AND. La salida de este bloque será 1 si y solo si ambas entradas son diferentes a 0[1]. Como los filtros anteriores son de orden relativamente bajo, es posible que al tratar las señales dejen pasar vestigios de la otra señal en la codificación pero con mucha menor amplitud. El valor efectivo de este resultado puede ser muy bajo, infinitesimal, del orden de $10^{-2}$, pero no 0 y por consiguiente el operador lógico lo tomará como válido y puede disparar falsos valores. Por esa razón debemos tomar la parte entera del valor eficaz, para garantizar que solo se debe tomar como válida a las señales que realmente son de la misma frecuencia que la frecuencia central de cada filtro.

## 4.4. Simulación

Para llevar a cabo la simulación vamos a considerar una serie de escenarios posibles, en los que se diferencia mayormente la confiabilidad en el canal de transmisión. Es decir, confiaremos en los bloques de control ya que son implementados a través de sistemas computacionales, pero el medio de transmisión es analógico y su eficacia depende de muchos factores físicos. Este se puede ver alterado de diversas maneras haciendo que parte de la información se pierda o corrompa. Para ello vamos a probar los siguientes escenarios:

1. El canal de transmisión es del tipo AWG-24 de menos de 200 [m] de largo, cuyo ancho de banda es de 1 [MHz].
2. El canal de transimsión es extenso y tiene el ancho de banda del espectro audíble por los humanos.
3. El mismo canal anterior presenta fallas atenuando en diferentes frecuencias dentro del espectro de tonos DTFM

[1] En el álgebra booleana aplicada en sistemas de control, todo valor igual a 0 se toma como ”Falso”, mientras que cualquier valor distinto de 0, por más infinitesimal que sea, es ”Verdadero”

### 4.4.1. Ancho de banda Extendido - 1 [MHz]

### 4.4.2. Ancho de banda Reducido - 3 [kHz]

### 4.4.3. Canal con Fallas

# Capítulo 5

# Conclusiones

## 5.1. Resultados

Aquí explicas tu metodología de investigación.

## 5.2. Aplicaciones

## 5.3. Problemas en la práctica

# Bibliografía

[OS89] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, Upper Saddle River, NJ, 1989.

[OS99] Alan V. Oppenheim and Ronald W. Schafer. *Digital Signal Processing: Principles, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, NJ, 1999.

[PM96] John G. Proakis and Dimitris G. Manolakis. *Digital Signal Processing*. Prentice Hall, Upper Saddle River, NJ, 1996.

[RG95] Lawrence R. Rabiner and Bernard Gold. *Theory and Application of Digital Signal Processing*. Prentice Hall, Upper Saddle River, NJ, 1st edition, 1995.

# Siglas

$f_a$ Frecuencia de Atenuación o Paso. 9, 10

$f_c$ Frecuencia de Corte. 9, 10

$f_p$ Frecuencia de Portadora. 6

$f_s$ Frecuencia de Muestreo. 4, 6, 9, 10

**AB** Ancho de Banda. 8

**AM** Amplitud Modulada. 3, 6, 7

**DTFM** Doble Tonos Múltiples Frecuencias. 3, 4, 6

# Índice de figuras

# Índice de cuadros

# Índice de bloques de código fuente

# Apéndice A

# Cálculos algebráicos

## A.1. Diseño de filtro digital H(z)

$$H(\mathrm{z}) = \frac{BW\left(\frac{2}{T}\frac{(1-\mathrm{z}^{-1})}{(1+\mathrm{z}^{-1})}\right)}{\left(\frac{2}{T}\frac{(1-\mathrm{z}^{-1})}{(1+\mathrm{z}^{-1})}\right)^2 + BW\left(\frac{2}{T}\frac{(1-\mathrm{z}^{-1})}{(1+\mathrm{z}^{-1})}\right) + \Omega_0^2}$$

$$H(\mathrm{z}) = \frac{BW\left(\frac{2}{T}\frac{(1-\mathrm{z}^{-1})}{(1+\mathrm{z}^{-1})}\right)}{\frac{4}{T^2}\frac{(1-\mathrm{z}^{-1})^2}{(1+\mathrm{z}^{-1})^2} + BW\left(\frac{2}{T}\frac{(1-\mathrm{z}^{-1})}{(1+\mathrm{z}^{-1})}\right) + \Omega_0^2}\,\frac{(1+\mathrm{z}^{-1})^2}{(1+\mathrm{z}^{-1})^2}$$

$$H(\mathrm{z}) = \frac{BW\frac{2}{T}(1-\mathrm{z}^{-1})(1+\mathrm{z}^{-1})}{\frac{4}{T^2}(1-\mathrm{z}^{-1})^2 + BW\frac{2}{T}(1-\mathrm{z}^{-1})(1+\mathrm{z}^{-1}) + \Omega_0^2(1+\mathrm{z}^{-1})^2}$$

$$H(\mathrm{z}) = \frac{2f_s\,BW(1-\mathrm{z}^{-1})(1+\mathrm{z}^{-1})}{4f_s^2(1-\mathrm{z}^{-1})^2 + 2f_s\,BW(1-\mathrm{z}^{-1})(1+\mathrm{z}^{-1}) + \Omega_0^2(1+\mathrm{z}^{-1})^2}$$

$$H(\mathrm{z}) = \frac{2f_s\,BW(1-\mathrm{z}^{-2})}{4f_s^2(1-2\mathrm{z}^{-1}+\mathrm{z}^{-2}) + 2f_s\,BW(1-\mathrm{z}^{-2}) + \Omega_0^2(1+2\mathrm{z}^{-1}+\mathrm{z}^{-2})}$$

$$H(\mathrm{z}) = \frac{2f_s\,BW - 2f_s\,BW\mathrm{z}^{-2}}{(4f_s^2 + 2f_s\,BW + \Omega_0^2) + (-8f_s^2 + 2\Omega_0^2)\,\mathrm{z}^{-1} + (4f_s^2 - 2f_s\,BW + \Omega_0^2)\,\mathrm{z}^{-2}} \tag{A.1}$$

Donde

$$\alpha = 4f_s^2 + 2f_s\,BW + \Omega_0^2$$
$$\beta = 2\Omega_0^2 - 8f_s^2$$
$$\gamma = 4f_s^2 - 2f_s\,BW + \Omega_0^2$$
$$\delta = 2f_s\,BW$$

Luego

$$H(\mathrm{z}) = \frac{\delta - \delta\mathrm{z}^{-2}}{\alpha + \beta\mathrm{z}^{-1} + \gamma\mathrm{z}^{-2}}$$

$$H(\mathrm{z}) = \frac{\frac{\delta}{\alpha} - \frac{\delta}{\alpha}\mathrm{z}^{-2}}{1 + \frac{\beta}{\alpha}\mathrm{z}^{-1} + \frac{\gamma}{\alpha}\mathrm{z}^{-2}}$$

$$H(\mathrm{z}) = \frac{8,4953\times10^{-3} - 8,4953\times10^{-3}\mathrm{z}^{-2}}{1 - 1,9732\mathrm{z}^{-1} + 0,9830\mathrm{z}^{-2}}$$