

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

19 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

Índice general	1
1. Introducción	3
1.1. Problema propuesto	3
1.2. Objetivo	3
1.3. Enunciado	4
1.4. Lineamientos Generales	4
2. Planteamiento	6
2.1. Sistema	6
2.2. Decodificación	7
2.3. Simulación	7
3. Filtros Digitales	9
3.1. Diseño	10
3.2. Conversión a digital	13
3.3. Conclusiones	15
4. Desarrollo	17
4.1. Análisis	17
4.2. Diseño	19
4.3. Especificaciones	21
4.4. Prototipo	21
5. Conclusiones	22
5.1. Resultados	22
5.2. Aplicaciones	22
5.3. Problemas en la práctica	22
Bibliografía	23
Siglas	24

Índice de figuras	25
Índice de cuadros	26
Índice de bloques de código fuente	27
A. Cálculos algebraicos	28
A.1. Diseño de filtro digital $H(z)$	28

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 envoltorios 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 para 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.

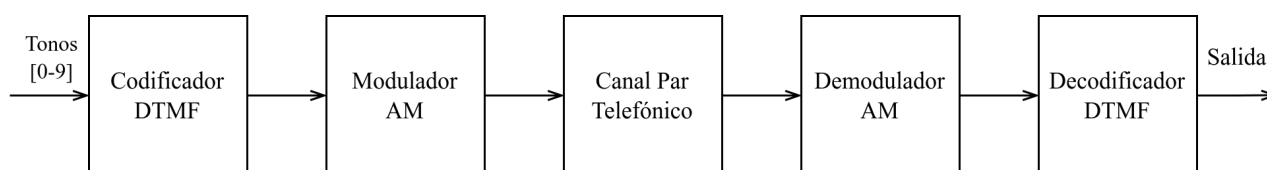


Figura 1.1: Diagrama de bloques general

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 nivel simulación se deberán sintetizar los tonos asociados a cada dígito numérico seleccionando una Frecuencia de Muestreo (f_s) apropiada (Teorema de Nyquist-Shannon).
- El demodulador DTFM deberá ser implementado mediante filtros digitales pasa bandas, con un orden y respuestas apropiadas. El modo de indicar cuál fue el dígito enviado queda a criterio del grupo de trabajo.
- Para el modelo de transmisión 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).
- 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

- El grupo de trabajo deberá cumplir con las especificaciones del proyecto, utilizando criterios de diseños justificados para cada bloque del sistema.
- Se deberán indicar el paso a paso para el diseño de los filtros digitales utilizados en las diferentes etapas.
- 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 imprimiría 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 dígitos 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 dígito
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

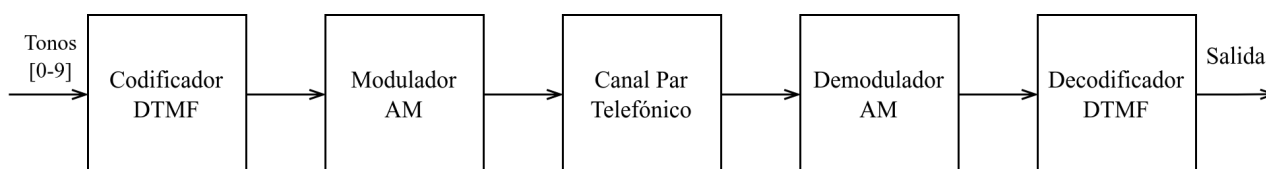


Figura 2.1: Diagrama de bloques a simular

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 dígito correspondiente en base a las señales que hayan logrado activarse luego pasar por el banco de filtros.

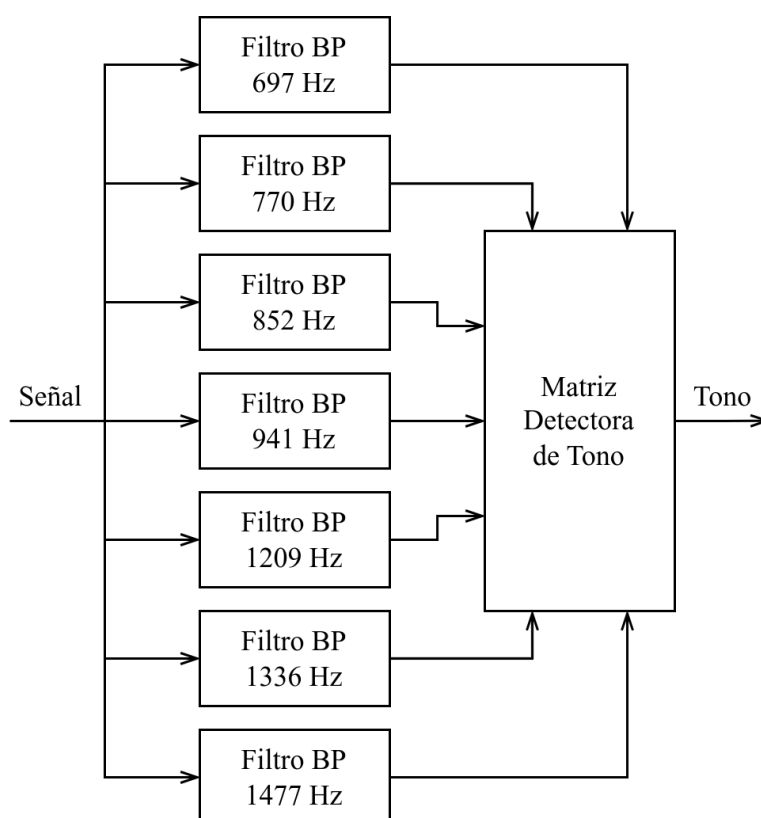


Figura 2.2: Composición del bloque decodificador

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 actúen 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**.

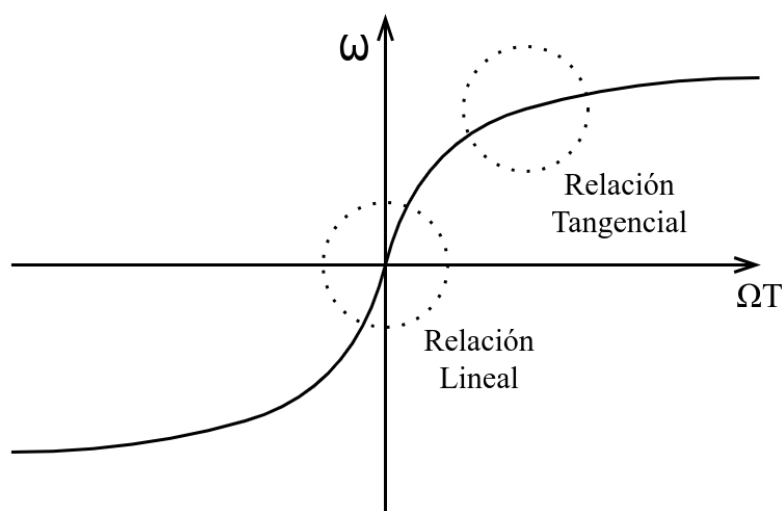


Figura 3.1: Distorsión de frecuencia introducida por la Transformación Bilineal

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].

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

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 describe 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 través 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.

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

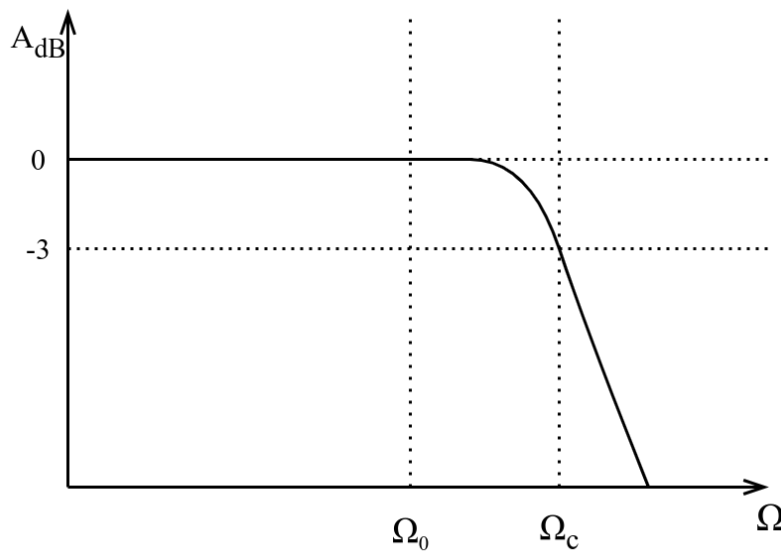


Figura 3.2: Respuesta en frecuencia de un filtro pasa-bajos

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.

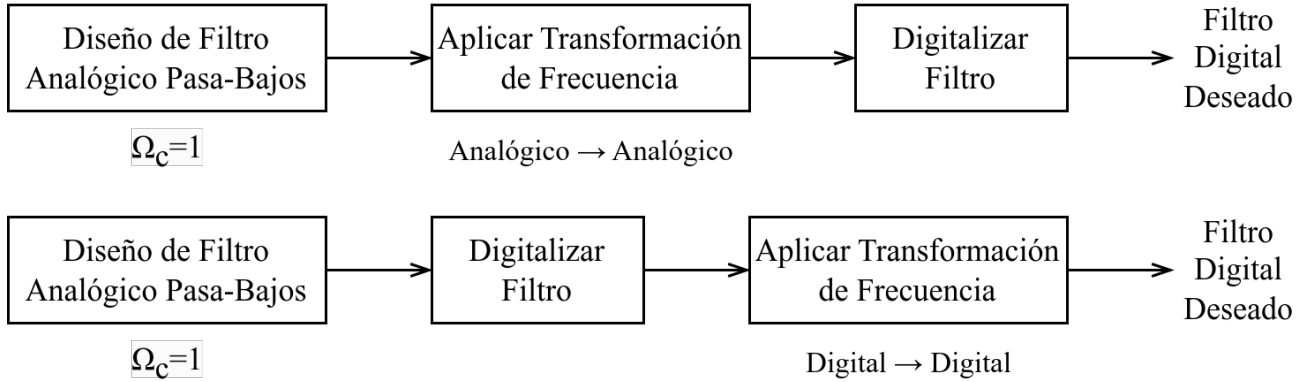


Figura 3.3: Enfoques para el diseño de un filtro digital

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.

$$s \rightarrow \frac{s}{\Omega_h} \quad \text{pasa-bajos} \rightarrow \text{pasa-bajos} \quad (3.3)$$

$$s \rightarrow \frac{\Omega_l}{s} \quad \text{pasa-bajos} \rightarrow \text{pasa-altos} \quad (3.4)$$

$$s \rightarrow \frac{s^2 + \Omega_l \Omega_h}{s(\Omega_h - \Omega_l)} \quad \text{pasa-bajos} \rightarrow \text{pasa-banda} \quad (3.5)$$

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

$$\Omega_l : f_c \text{ inferior}$$

$$\Omega_h : f_c \text{ superior}$$

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 dígitos 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 distorsión de frecuencia, y para ello debemos realizar el pre-warping. Entonces las frecuencias angulares de corte analógicas superior e inferior están dadas por la Ecuación 3.7 y la Ecuación 3.8 respectivamente, donde f_s es igual a 44 [kHz].

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

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

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 Ω_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.

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

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

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

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

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

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 Ω_l y Ω_h . Por lo tanto queda demostrado que el filtro analógico diseñado es realmente un pasa-banda.

```

1  a = [0 1 0];
2  b = [1.32e-3 1 25.22e3];
3  [h, w] = freqs(a, b, 5000);
4  phase = angle(h);
5  phasedeg = phase * 180 / pi;
6  mag = abs(h);
7
8  subplot(2, 1, 1)
9  loglog(w, mag)
10 grid on
11 xlabel('Frecuencia (rad/s)')
12 ylabel('Magnitud')
13 xlim([3500, 5500])
14 ylim([0.2, 1.2])
15
16 subplot(2, 1, 2)
17 semilogx(w, phasedeg)
18 grid on
19 xlabel('Frecuencia (rad/s)')
20 ylabel('Fase (grados)')
21 xlim([3500, 5500])
22 ylim('auto')

```

Código fuente 3.1: Graficar respuesta en frecuencia de Filtro Analógico

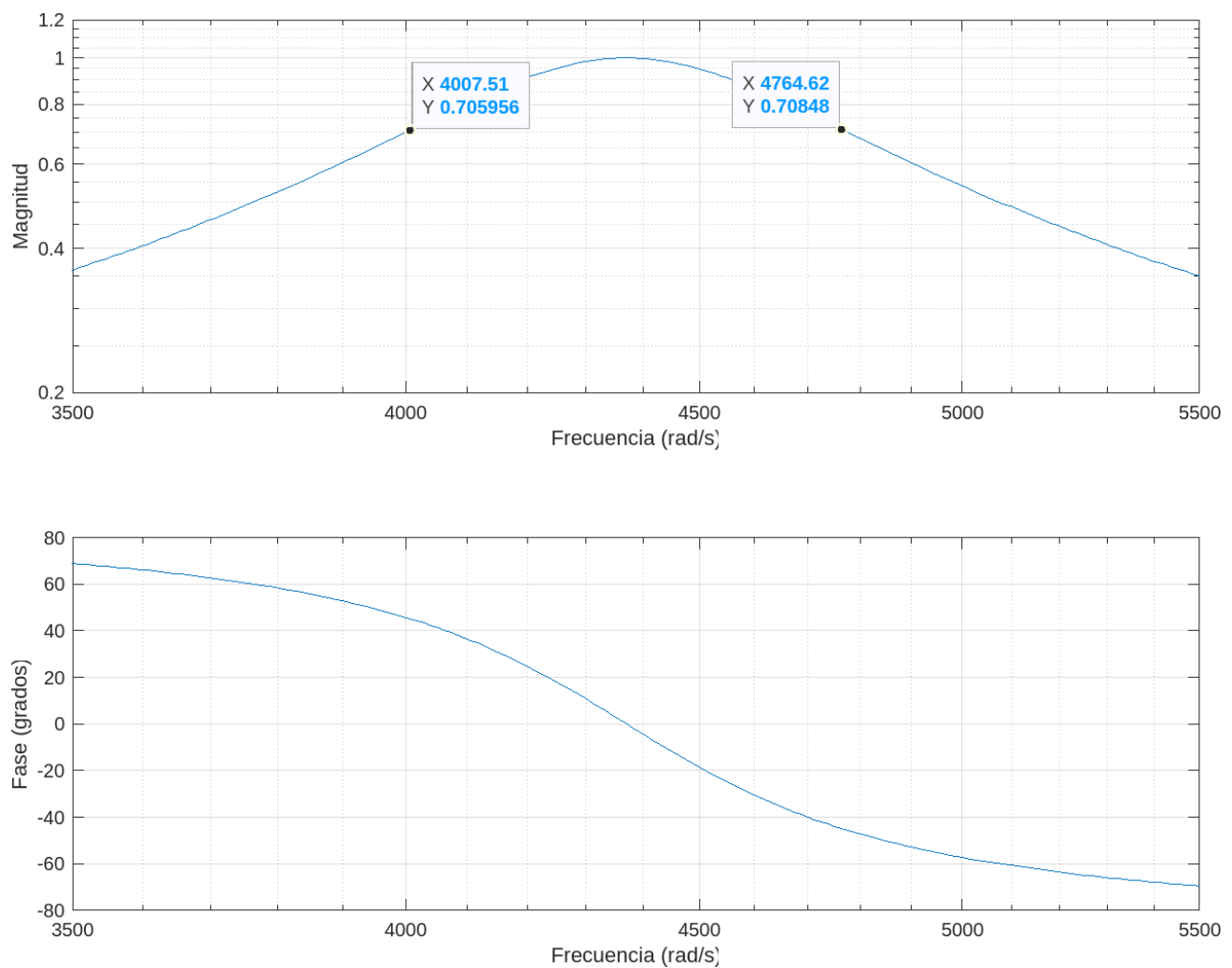


Figura 3.4: Respuesta en frecuencia del filtro analógico

3.2. Conversión a digital

Ahora que tenemos nuestro filtro analógico, el siguiente paso es transformarlo a digital a través 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 algebraicamente (ver Apéndice A.1) y encontramos $H(z)$, que está dado por la Ecuación 3.17.

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

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

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

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

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].

```

1 Fs = 44000;
2 Ts = 1 / Fs;
3 num = [8.4953e-3 0 -8.4954e-3];
4 den = [1 -1.9732 0.9830];
5
6 [h, w] = freqz(num, den, 20000);
7 f = w / (2 * pi);
8 plot(f * (Fs / 2) / max(f), abs(h))
9 title('Filtro Butterworth')
10 xlabel('Hertz')
11 ylabel('Magnitud')
12 xlim([0, 1200])
13 ylim('auto')

```

Código fuente 3.2: Graficar respuesta en frecuencia de Filtro Digital

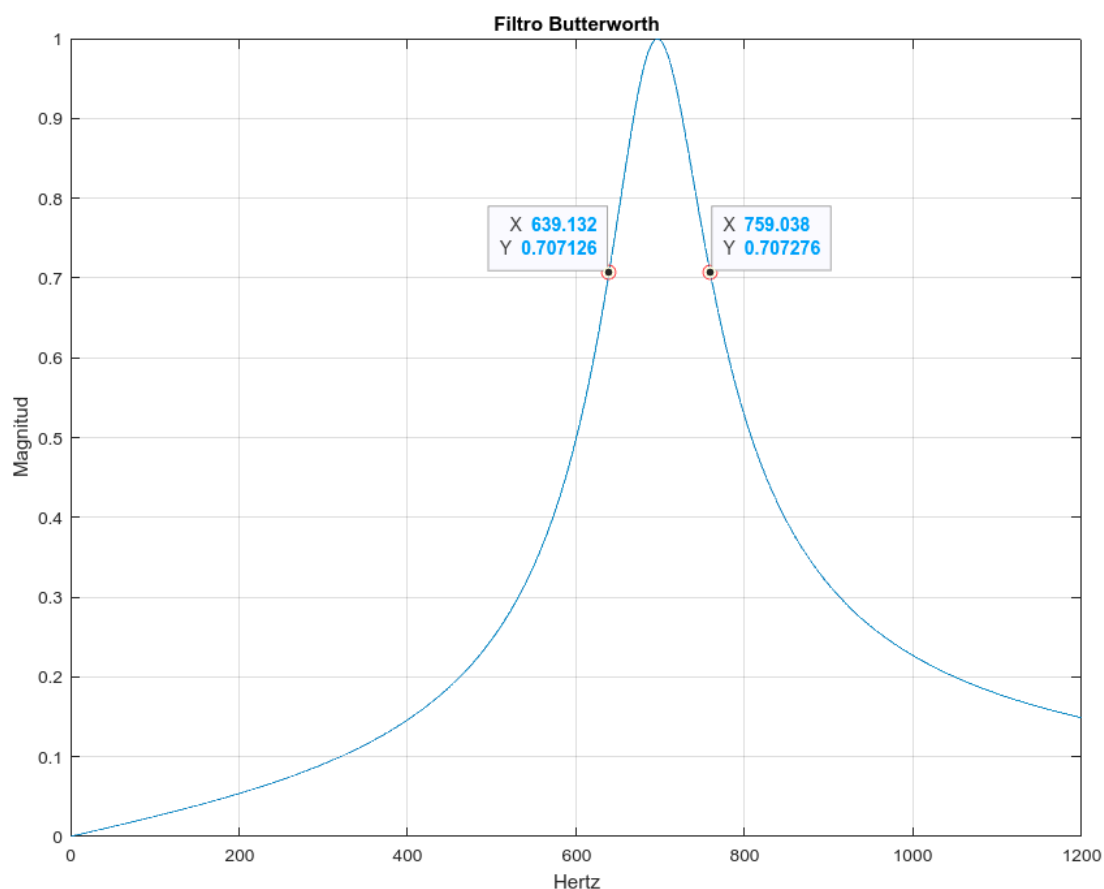


Figura 3.5: Respuesta en frecuencia del filtro digital

3.3. Conclusiones

En las secciones anteriores demostramos cómo es el procedimiento para el diseño de un filtro digital pasa-banda a partir de transformaciones 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 una 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.

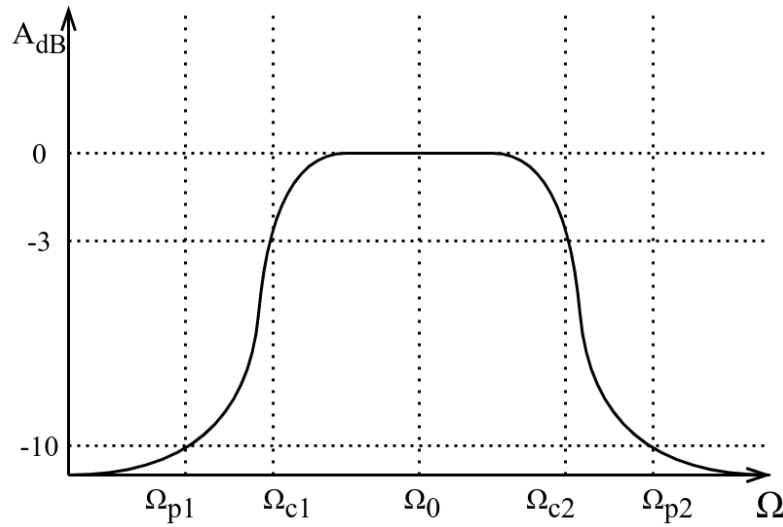


Figura 3.6: Respuesta en frecuencia de un filtro pasa-banda con frecuencias de parada

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.

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

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

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

Con esto demostramos y concluimos 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 dígito. 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.

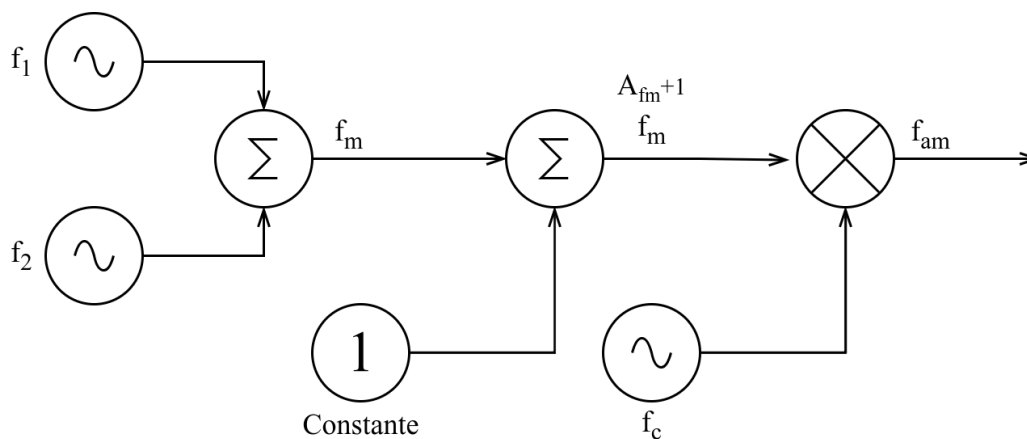


Figura 4.1: Codificador y Modulador

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.

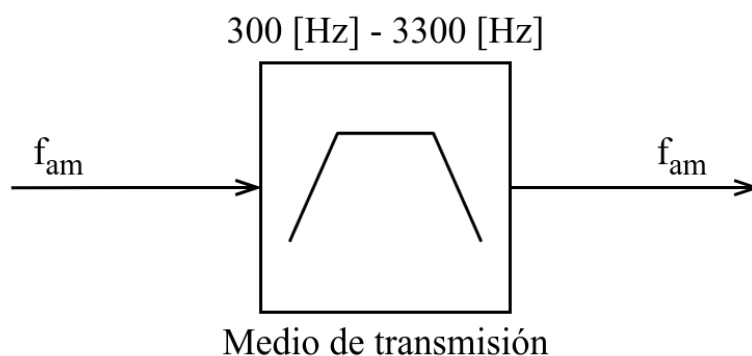


Figura 4.2: Transmisión

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 dígito 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.

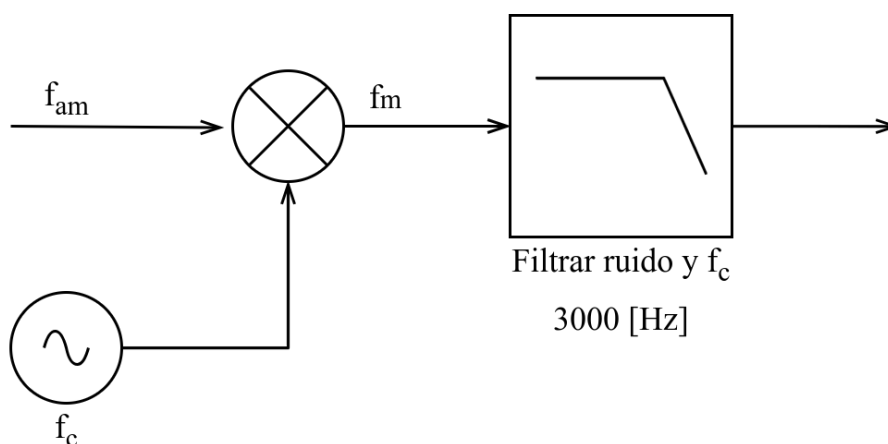


Figura 4.3: Transmisión

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 fue el dígito enviado.

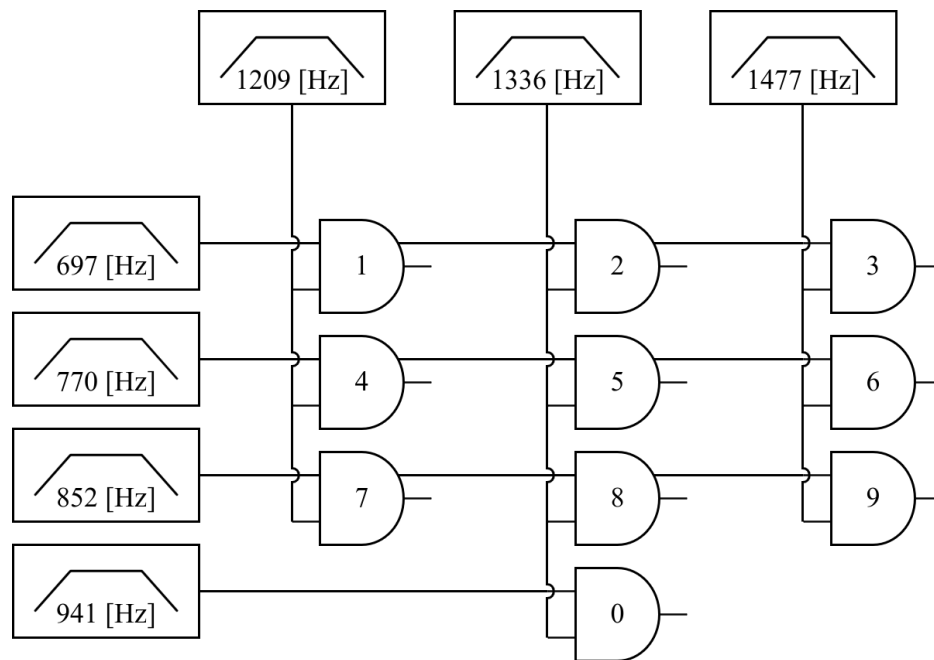


Figura 4.4: Decodificador

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 frecuencias 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. Luego en el Código 4.2 tenemos el algoritmo 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 transmisión y el filtro para eliminar 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.

```

1  function filtros = banco_decodificador(muestreo, frecuencias, orden)
2      % Generador de coeficientes de polinomio de filtros BP Butterworth
3
4      for freq = frecuencias
5          % Establecer frecuencias de corte
6          fc_low = freq - 40;
7          fc_high = freq + 40;
8
9          % Establecer coeficientes para filtro pasa-banda actual
10         [a, b] = butter(orden / 2, [fc_low fc_high] / (muestreo / 2));
11
12         % Generar nombre de variable correspondiente
13         % a cada arreglo de coeficientes
14         temp_num = strcat('f_', num2str(freq), '_num');
15         temp_den = strcat('f_', num2str(freq), '_den');
16
17         % Guardar coeficientes
18         filtros.(temp_num) = a;
19         filtros.(temp_den) = b;
20     end
21
22 end

```

Código fuente 4.1: Banco Decodificador

```

1  clc
2  close all
3  clear
4
5  % Frecuencia de muestreo
6  Fs = 44000;
7
8  % Arreglo con frecuencias centrales a filtrar
9  frecuencias_centrales = [697, 770, 852, 941, 1209, 1336, 1477];
10
11 % Orden de cada filtro pasa-banda del banco de filtros
12 orden = 6;
13
14 % Coeficientes del filtro pasa-bajos para filtrar la frecuencia portadora
15 [f_portadora_num, f_portadora_den] = butter(orden, 1500 / (Fs / 2));
16
17 % Coeficientes del filtro pasa-banda para emular el canal
18 [f_canal_num, f_canal_den] = butter(2, [300, 3500] / (Fs / 2));
19
20 % Estructura de datos para generar y guardar
21 % los coeficientes del banco de filtros
22 filtros = banco_decodificador(Fs, frecuencias_centrales, orden)

```

Código fuente 4.2: Algoritmo principal

En base a los resultados obtenidos de los *scripts* diseñamos el sistema completo, en los que cada bloque

se alimenta de los datos resultantes. (Continuar que no queda nada)

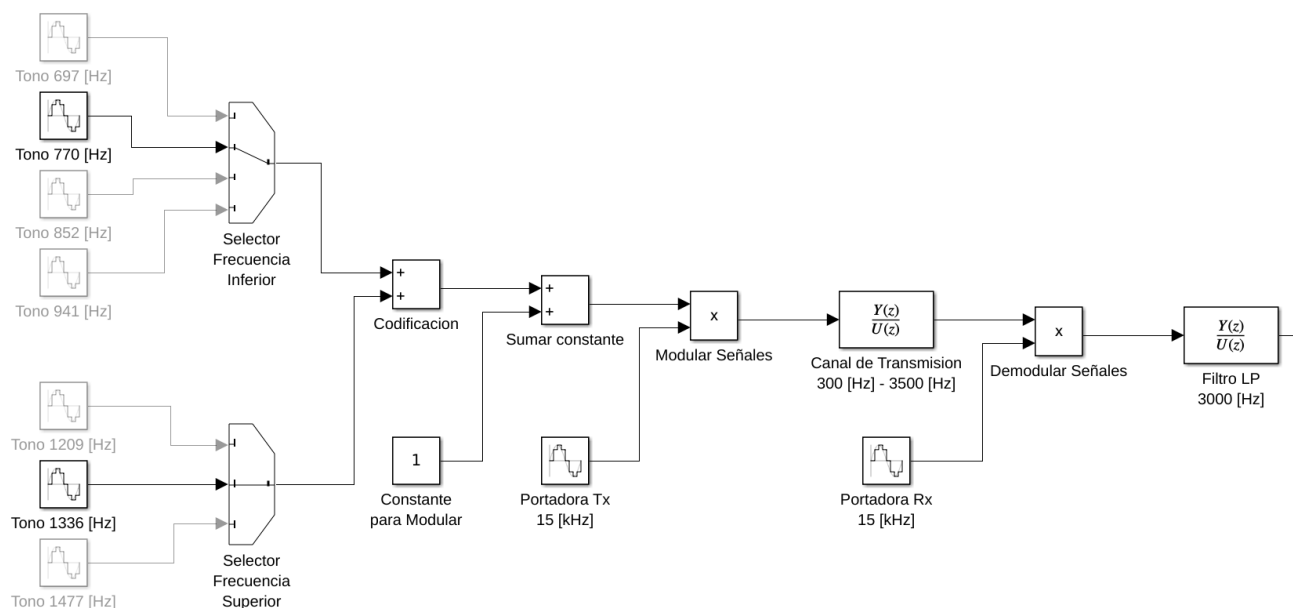


Figura 4.5: Codificación, Modulación, Transmisión y Demodulación

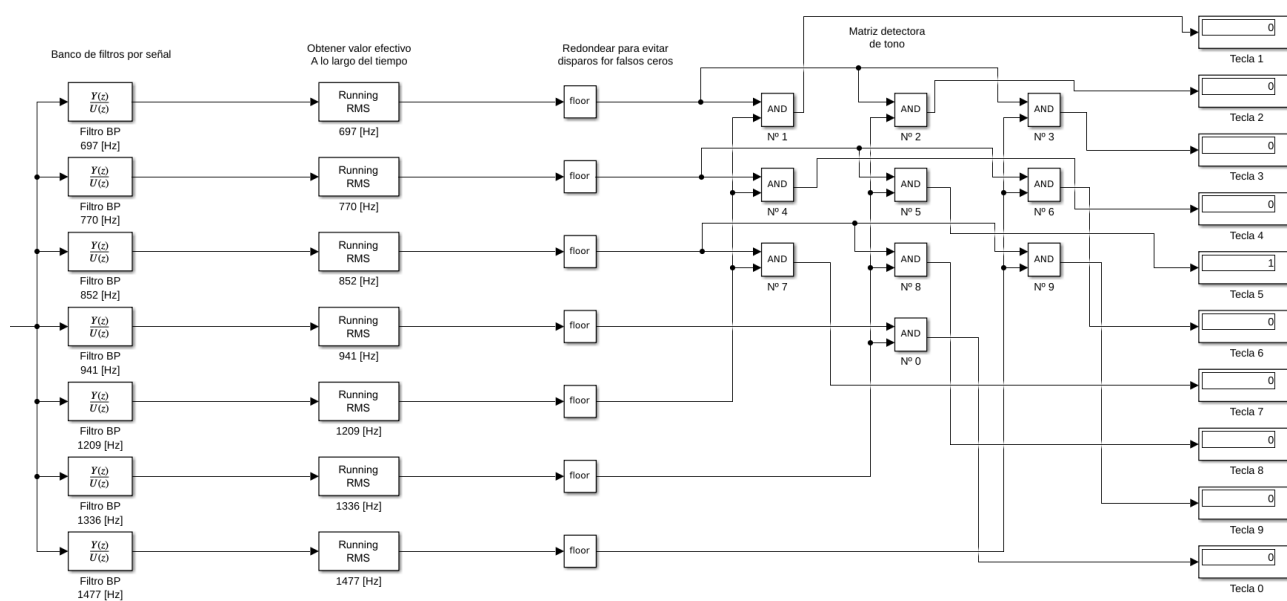


Figura 4.6: Banco de filtros y Decodificador

4.3. Especificaciones

4.4. Prototipo

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. Schaffer. *Discrete-Time Signal Processing*. Prentice Hall, Upper Saddle River, NJ, 1989.
- [OS99] Alan V. Oppenheim and Ronald W. Schaffer. *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

1.1. Diagrama de bloques general	3
2.1. Diagrama de bloques a simular	6
2.2. Composición del bloque decodificador	7
3.1. Distorsión de frecuencia introducida por la Transformación Bilineal	9
3.2. Respuesta en frecuencia de un filtro pasa-bajos	10
3.3. Enfoques para el diseño de un filtro digital	11
3.4. Respuesta en frecuencia del filtro analógico	13
3.5. Respuesta en frecuencia del filtro digital	15
3.6. Respuesta en frecuencia de un filtro pasa-banda con frecuencias de parada	16
4.1. Codificador y Modulador	17
4.2. Transmisión	18
4.3. Transmisión	18
4.4. Decodificador	19
4.5. Codificación, Modulación, Transmisión y Demodulación	21
4.6. Banco de filtros y Decodificador	21

Índice de cuadros

1.1. Combinación de tonos audibles (medido en [Hz])	4
---	---

Índice de bloques de código fuente

3.1. Graficar respuesta en frecuencia de Filtro Analógico	12
3.2. Graficar respuesta en frecuencia de Filtro Digital	14
4.1. Banco Decodificador	20
4.2. Algoritmo principal	20

Apéndice A

Cálculos algebraicos

A.1. Diseño de filtro digital $H(z)$

$$\begin{aligned}
 H(z) &= \frac{BW \left(\frac{2}{T} \frac{(1-z^{-1})}{(1+z^{-1})} \right)}{\left(\frac{2}{T} \frac{(1-z^{-1})}{(1+z^{-1})} \right)^2 + BW \left(\frac{2}{T} \frac{(1-z^{-1})}{(1+z^{-1})} \right) + \Omega_0^2} \\
 H(z) &= \frac{BW \left(\frac{2}{T} \frac{(1-z^{-1})}{(1+z^{-1})} \right)}{\frac{4}{T^2} \frac{(1-z^{-1})^2}{(1+z^{-1})^2} + BW \left(\frac{2}{T} \frac{(1-z^{-1})}{(1+z^{-1})} \right) + \Omega_0^2} \frac{(1+z^{-1})^2}{(1+z^{-1})^2} \\
 H(z) &= \frac{BW \frac{2}{T} (1-z^{-1})(1+z^{-1})}{\frac{4}{T^2} (1-z^{-1})^2 + BW \frac{2}{T} (1-z^{-1})(1+z^{-1}) + \Omega_0^2 (1+z^{-1})^2} \\
 H(z) &= \frac{2f_s BW (1-z^{-1})(1+z^{-1})}{4f_s^2 (1-z^{-1})^2 + 2f_s BW (1-z^{-1})(1+z^{-1}) + \Omega_0^2 (1+z^{-1})^2} \\
 H(z) &= \frac{2f_s BW (1-z^{-2})}{4f_s^2 (1-2z^{-1}+z^{-2}) + 2f_s BW (1-z^{-2}) + \Omega_0^2 (1+2z^{-1}+z^{-2})} \\
 H(z) &= \frac{2f_s BW - 2f_s BW z^{-2}}{(4f_s^2 + 2f_s BW + \Omega_0^2) + (-8f_s^2 + 2\Omega_0^2) z^{-1} + (4f_s^2 - 2f_s BW + \Omega_0^2) z^{-2}} \quad (A.1)
 \end{aligned}$$

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

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

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

$$H(z) = \frac{8,4953 \times 10^{-3} - 8,4953 \times 10^{-3} z^{-2}}{1 - 1,9732 z^{-1} + 0,9830 z^{-2}}$$