Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Eléctrica

C3 Señales y Sistemas

Integrantes: Pía Contreras
Profesor: Claudio Pérez
Auxiliar: Juan Pablo Pérez
Ayudantes: Daniel Benalcazar
Daniel Montecino
Carlos Navarro
Jorge Zambrano
Fecha de realización: 12 de diciembre de 2019
Fecha de entrega: 12 de diciembre de 2019
Santiago, Chile

Índice de Contenidos

Índice de Figuras

1. P1

La respuesta en frecuencia para la ventana rectangular está dado por:

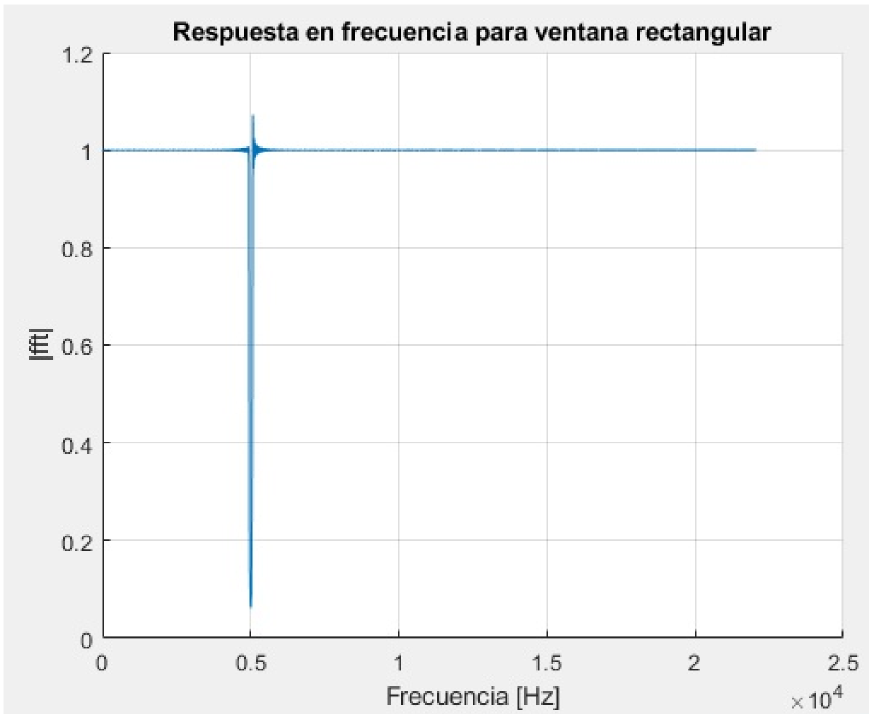


Figura 1: Respuesta en frecuencia Ventana Rectangular

La respuesta en frecuencia para la ventana de hamming está dada por:

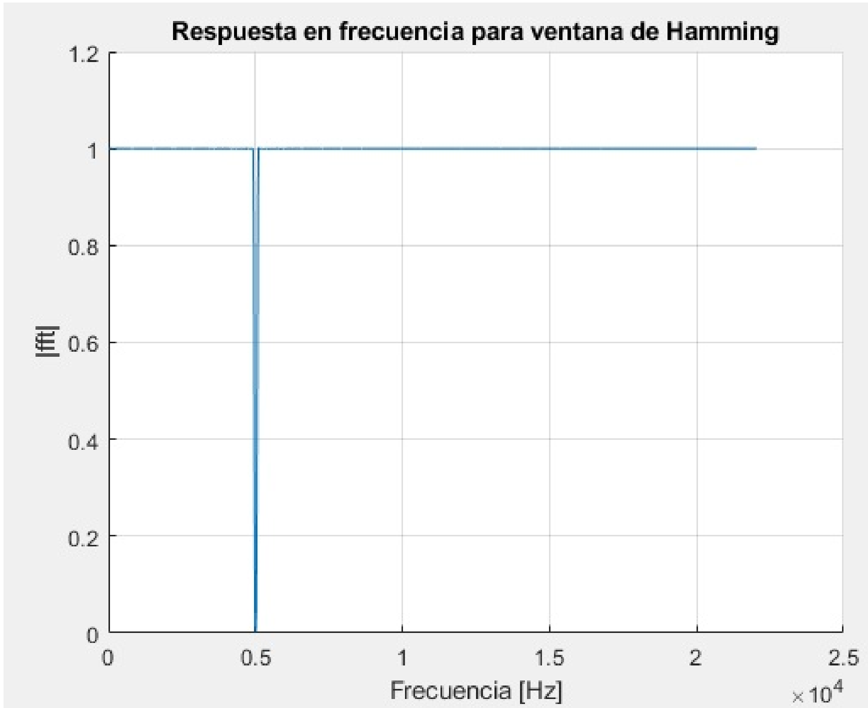


Figura 2: Respuesta en frecuencia Ventana de Hamming

El espectro de frecuencia de la señal de audio sin pasar por algun filtro está dada por:

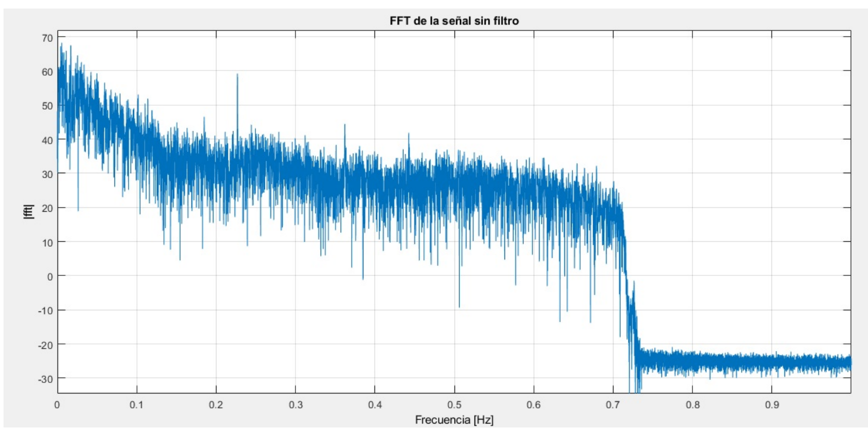


Figura 3: Espectro de frecuencia señal de audio sin pasar por filtro

El espectro de frecuencia de la señal de audio pasando por la ventana rectangular está dado por:

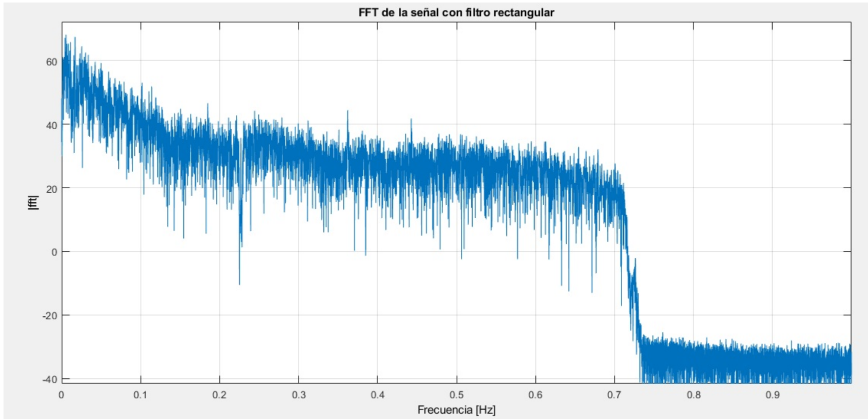


Figura 4: Espectro de frecuencia de señal de audio pasando por ventana rectangular

Finalmente, el espectro de frecuencia de la señal de audio pasando por la ventana de Hamming está dado por:

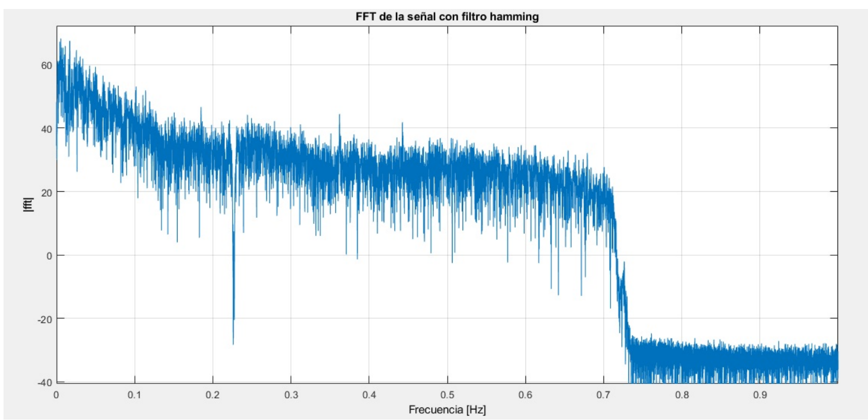


Figura 5: Espectro de frecuencia de señal de audio pasando por ventana de Hamming

2. P2

2.1. Parte a

La imagen obtenida por el filtrado es la siguiente:

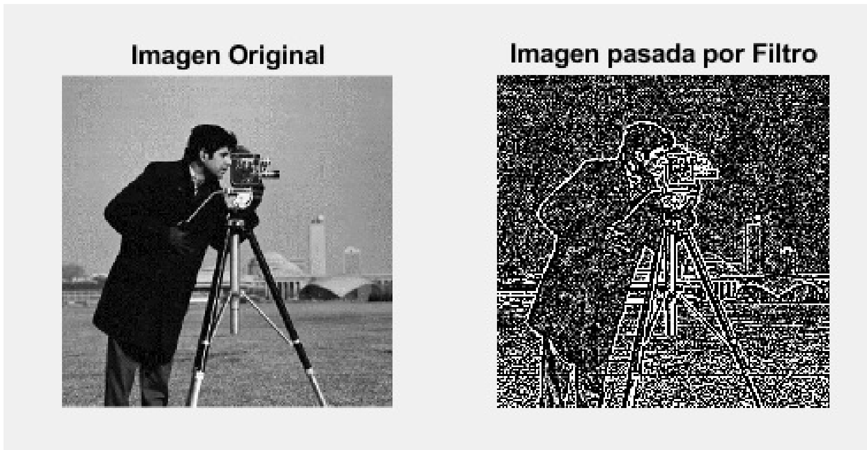


Figura 6: Imagen Original v/s Imagen pasada por filtro

2.2. Parte b

La imagen obtenida por el filtrado es la siguiente:

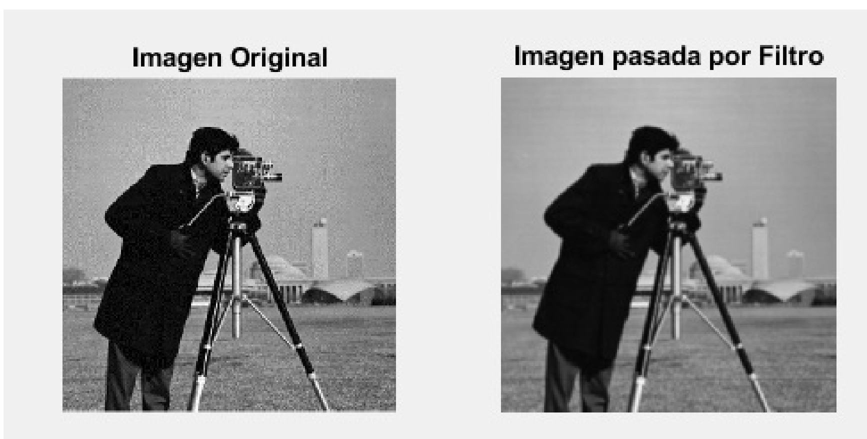


Figura 7: Imagen Original v/s Imagen pasada por filtro

2.3. Parte C

A grandes rasgos la convolución en 2D es el paso de una señal 2D, en este caso, una imagen, por un filtro para luego obtener una señal en 2D, en este caso, también se obtiene una imagen.

El filtro de la parte a) es un típico filtro de realce, es decir un filtro para alto. Suele tener valores positivos y negativos, y el valor positivo suele ser el píxel central y los valores negativos suelen ser los píxeles periféricos, como es posible ver en la siguiente figura:

0	0	-1	-1	-1	0	0
0	-1	-3	-3	-3	-1	0
-1	-3	0	7	0	-3	-1
-1	-3	7	24	7	-3	-1
-1	-3	0	7	0	-3	-1
0	-1	-3	-3	-3	-1	0
0	0	-1	-1	-1	0	0

Figura 8: Filtro A

Este filtro, es un filtro pasa alto estricto pues la suma de los dígitos del filtro es cero, eliminaré también la componente continua, la frecuencia 0.

El objetivo de este filtro es destacar los detalles finos e intensificar los detalles difuminados por error.

Por su parte, el filtro de la parte b) es un típico filtro de suavizado, es decir, filtro pasa bajo. Éste tipo de filtro, atenúa las altas frecuencias, dejando pasar las bajas, el efecto es una pérdida de nitidez y reducción del ruido, además de difuminar los bordes y otros detalles de realce.

El filtro está dado por:

1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1

Figura 9: Filtro B

A mayor tamaño del filtro, con píxeles iguales, aumento el efecto de suavizado. Por otro lado, si aumento el píxel central disminuye el efecto de suavizado.

P9 Pía Contreras Guerrero
19.840.187-0

a) Las alternativas que existen para corregir y detectar errores son 2, bit de paridad y distancia de Hamming.

Por su parte el bit de paridad permite detectar errores mas no corregirlos, a excepción de que la palabra sea de tamaño 1 bit, en específico en este caso, con 0010 1011 y 1100 0011, no es posible corregir errores con el bit de paridad, pero sí los identifica.

Así, por ejemplo:

8 bits de datos	byte con bit de paridad	
	par	impar
0010 10 11	001010110	001010111
11 0000 11	110000110	110000111

Por otra parte, la distancia de Hamming permite detectar y corregir errores. A mayor distancia, menos es la posibilidad de que un código válido se transforme en otro código válido por una serie de errores.

Así, por ejemplo, la distancia de Hamming entre 0010 10 11 y 11 0000 11 se ve como la diferencia que tiene entre ambos códigos, en los cuales se ve que poseen 4 de sus 8 bits diferentes.

Por lo tanto, la distancia de Hamming es 4.

b)

i) En este caso, es más explícito verlo geométricamente con las esferas de hamming. Luego, como poseen 10 bits cada palabra válida, entonces, la mínima distancia de Hamming es ; $\frac{\# bit}{2} = 5 =$ distancia mínima de Hamming

Además, calculando la distancia de cada palabra con otra respectivamente

		dist Ham
0000000000	→ 0000011111	5
	→ 1111100000	5
	→ 1111111111	10
0000011111	→ 0000000000	5
	→ 1111100000	10
	→ 1111111111	5
1111100000	→ 0000000000	5
	→ 0000011111	10
	→ 1111111111	5
1111111111	→ 0000000000	10
	→ 1111100000	5
	→ 0000011111	5

∴ La distancia mínima de Hamming es 5.

b) ii) 0000000000

$$\left\{\begin{array}{l} 0000000001 \\ 0000000010 \\ 0000000100 \\ 0000001000 \\ 0000010000 \\ 0000100000 \\ 0001000000 \\ 0010000000 \\ 0100000000 \\ 1000000000 \end{array}\right.$$

Scanned by CamScanner

$$0\,0\,0\,0\,0\,1\,1\,1\,1\,1 \quad \left\{ \begin{array}{cccccccccc}
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1
\end{array} \right.$$

$$1\,1\,1\,1\,1\,0\,0\,0\,0\,0 \quad \left\{ \begin{array}{cccccccccc}
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0
\end{array} \right.$$

$$1\,1\,1\,1\,1\,1\,1\,1\,1\,1 \quad \left\{ \begin{array}{cccccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array} \right.$$

b) iii) Las alternativas para corregir y detectar errores son 2, bits de paridad y distancia de Hamming

En este caso, el bit de paridad para cada palabra válida sería lo siguiente:

10 bits de datos	byte con bit de paridad	
	par	impar
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 1
1 1 1 1 1 0 0 0 0 0	1 1 1 1 1 0 0 0 0 0 1	1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1	0 0 0 0 0 1 1 1 1 1 1	0 0 0 0 0 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 0	1 1 1 1 1 1 1 1 1 1 1

Cabe destacar que el bit de paridad sólo permite conocer los errores y no corregirlos, al menos en este caso.

Otra alternativa es la distancia de Hamming, que permite detectar y corregir errores, de acuerdo, a la diferencia en la comparación de 2 palabras

Así, por ejemplo el desarrollo de la parte b) i) muestra como obtener la distancia de hamming entre las palabras válidas del código.

P4)

Pía Contreras Guerrero
19.840.187-0

Dado el sistema y las transformadas de Fourier de $x(t)$ y $s(t)$, aplicaremos propiedades de la T. de Fourier dad como

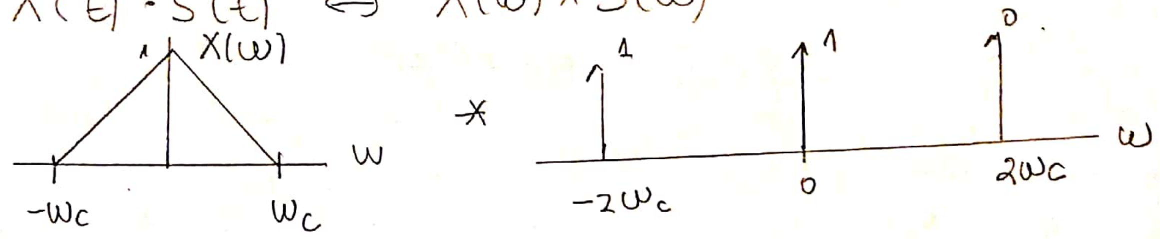
$$s_1(t) \cdot s_2(t) \longleftrightarrow S_1(\omega) * S_2(\omega)$$

luego de acuerdo al sistema dado

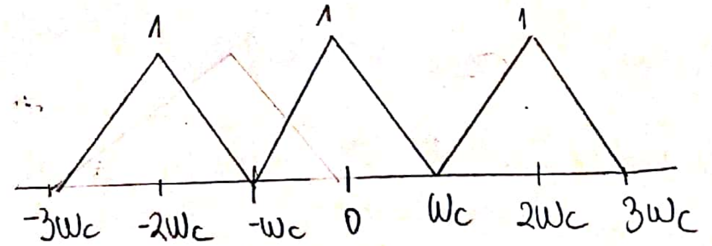
$$X(t) \cdot S(t) \cdot m(t) \longleftrightarrow X(\omega) * S(\omega) * M(\omega).$$

Como tenemos la transformada de Fourier de $s(t)$ y $x(t)$, $S(\omega)$ y $X(\omega)$ respectivamente, entonces hacemos convolución.

$$X(t) \cdot S(t) \leftrightarrow X(\omega) * S(\omega)$$



=

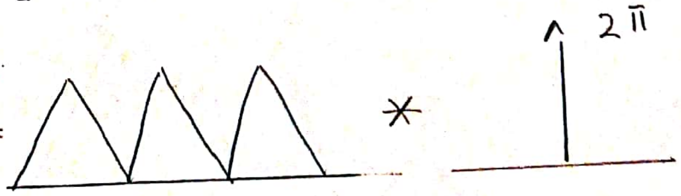


$= W(\omega)$

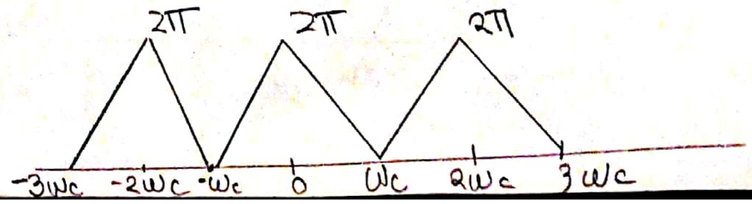
Analizando cada alternativa, es decir, convolucionando con $M(\omega)$

a) Como $m(t) = 1 \Rightarrow M(\omega) = 2\pi\,\delta(\omega)$, por propiedad de T de Fourier

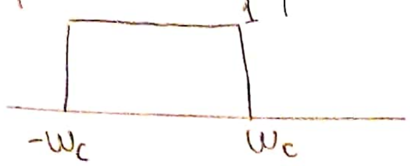
$\Rightarrow W(\omega) * M(\omega) =$



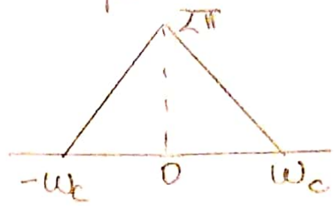
=



pasando por el filtro indicado por el sistema :

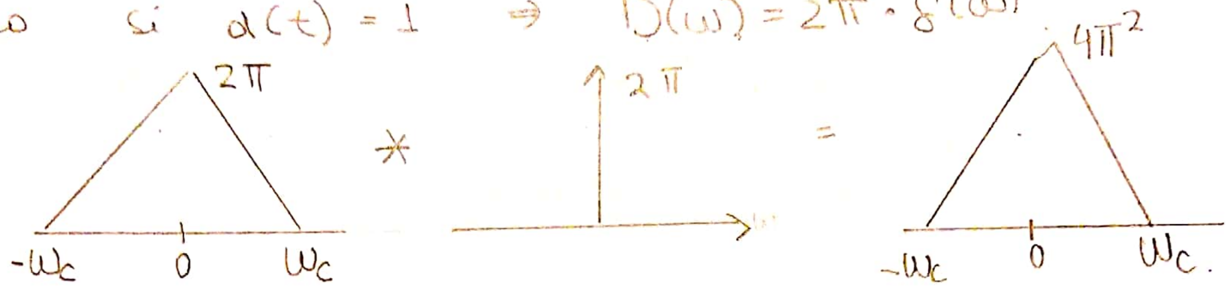


se obtiene

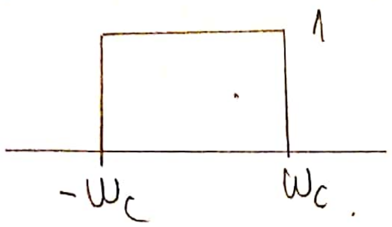


y luego si $d(t) = 1 \Rightarrow D(\omega) = 2\pi \cdot \delta(\omega)$

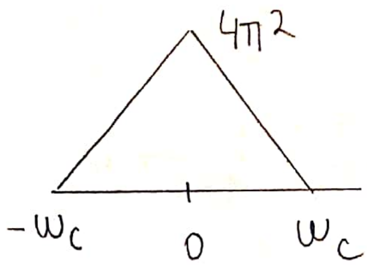
Luego



De acuerdo al sistema y pasando nuevamente por el filtro



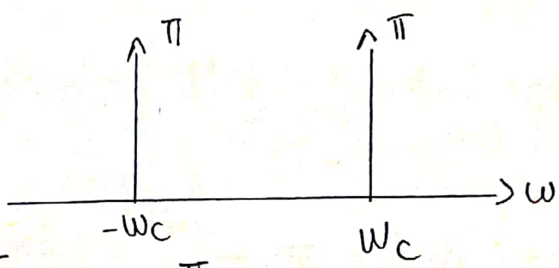
se obtiene, finalmente la salida :



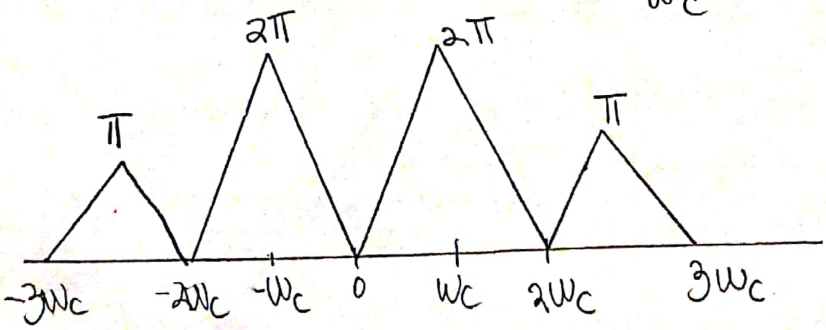
$= Y(\omega)$ NO NUCA

b) $m(t) = \cos(W_c t) \Rightarrow M(\omega) = \pi\,\delta(\omega - W_c) + \pi\,\delta(\omega + W_c)$

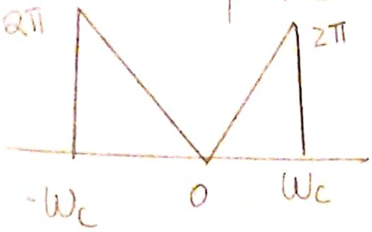
$d(t) = \cos(W_c t) \Rightarrow D(\omega) = \pi\,\delta(\omega - W_c) + \pi\,\delta(\omega + W_c)$



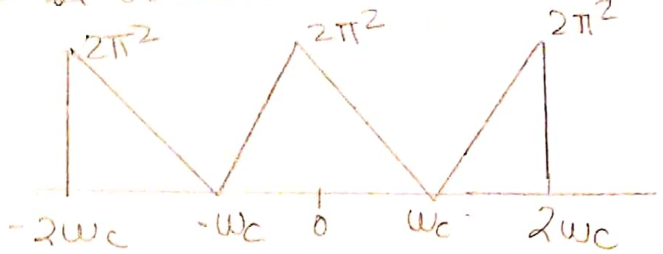
$W(\omega) * M(\omega) =$



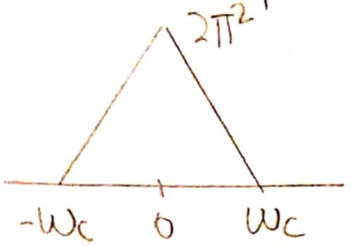
pasando por el primer filtro se obtiene:



$* \; D(\omega) =$



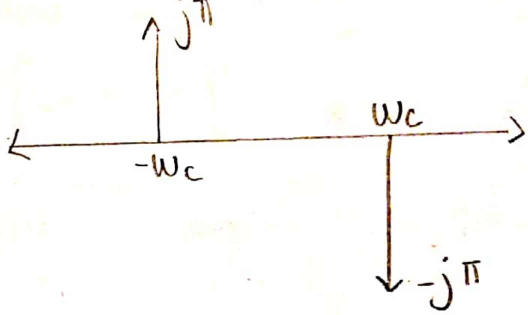
pasando por filtro nuevamente se obtiene



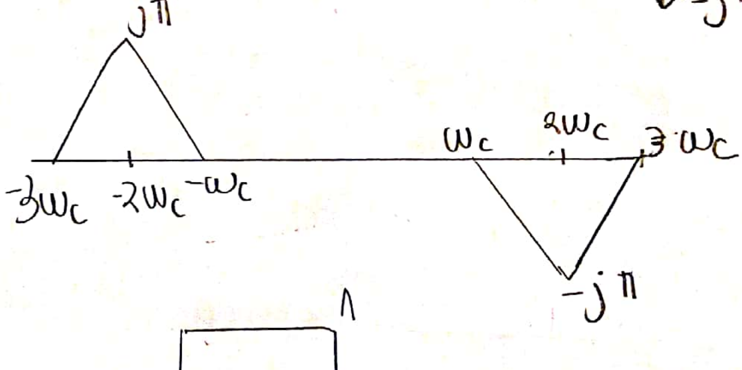
$= Y(\omega)$ NO NULA

c) $m(t) = \operatorname{sen}(W_c t) \rightarrow M(\omega) = j\pi\,\delta(\omega + W_c) - j\pi\,\delta(\omega - W_c)$

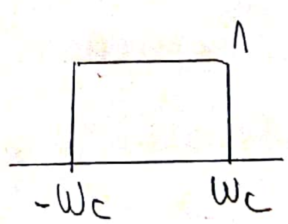
$d(t) = \operatorname{sen}(W_c t) \rightarrow D(\omega) = j\pi\,\delta(\omega + W_c) - j\pi\,\delta(\omega - W_c)$



$W(\omega) * M(\omega) =$



pasando por el filtro



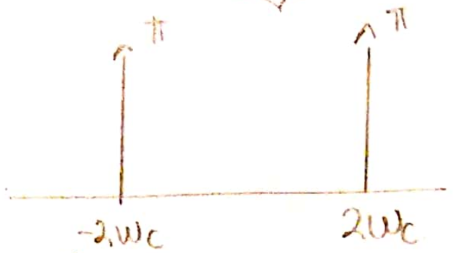
se obtiene señal nula $\therefore$ hacia adelante será señal nula

$\Rightarrow Y(\omega) =$ ——— NULA

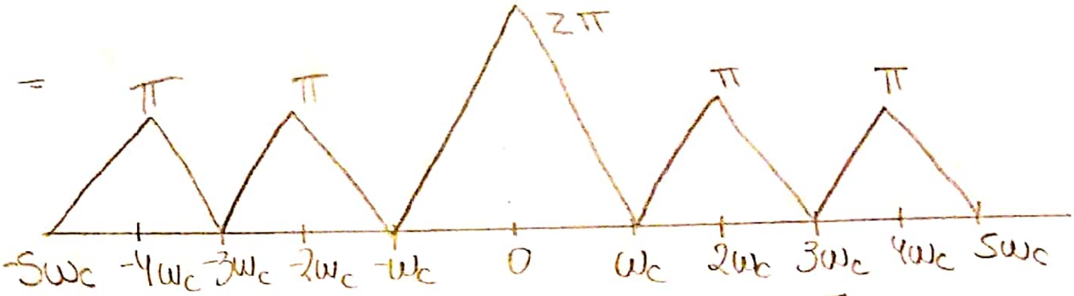
Scanned by CamScanner

d) $m(t) = \cos(2w_c t) \Rightarrow M(w) = \pi\delta(w - 2w_c) + \pi\delta(w + 2w_c)$

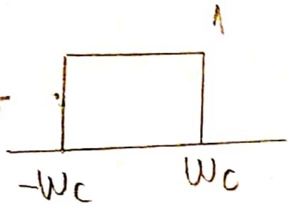
$d(t) = \cos(2w_c t) \Rightarrow D(w) = \pi\delta(w - 2w_c) + \pi\delta(w + 2w_c)$



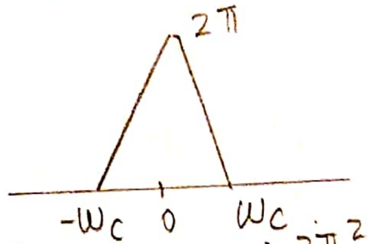
$W(w) * M(w) =$

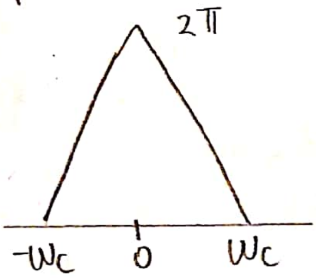


pasando por el filtro

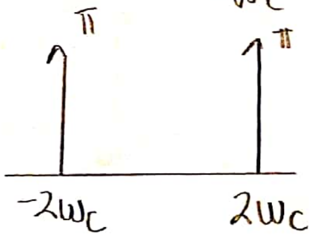


$\Rightarrow$

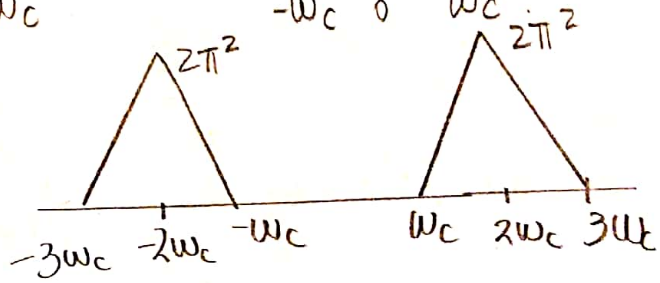




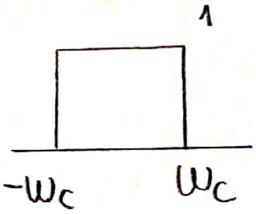
$*$



$=$



pasando por filtro

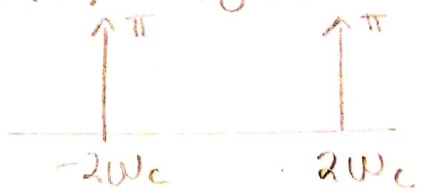


$\Rightarrow$ ________

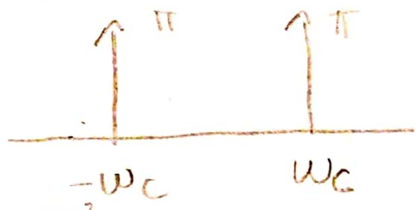
SEÑAL NULA

$\Rightarrow Y(w) =$ ________ NULA

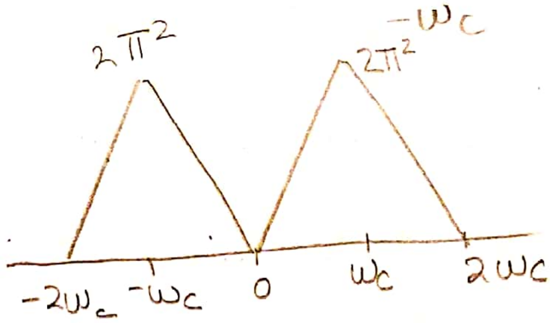
e) $m(t) = \cos(2\omega_c t) \Rightarrow M(\omega) = \pi\delta(\omega - 2\omega_c) + \pi\delta(\omega + 2\omega_c)$



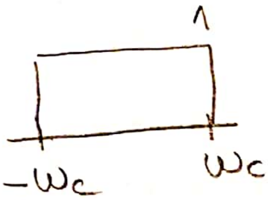
$d(t) = \cos(\omega_c t) \Rightarrow D(\omega) = \pi\delta(\omega - \omega_c) + \pi\delta(\omega + \omega_c)$



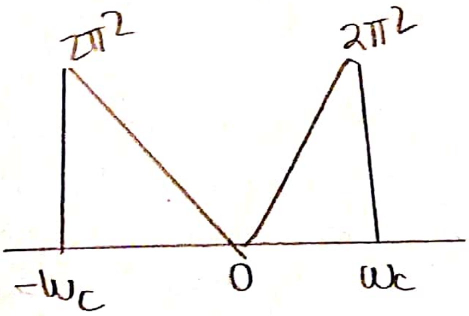
$W(\omega) * M(\omega) =$



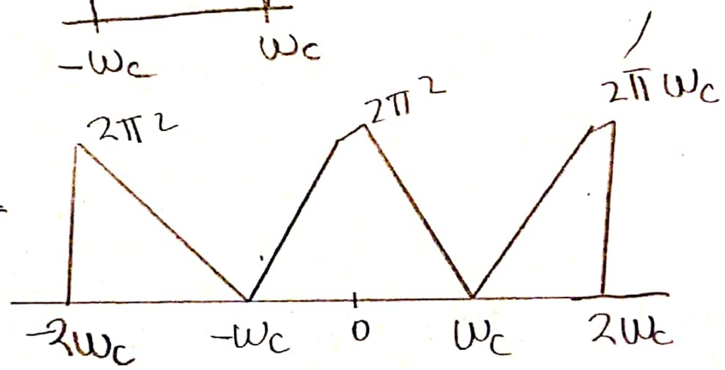
pasando por el filtro



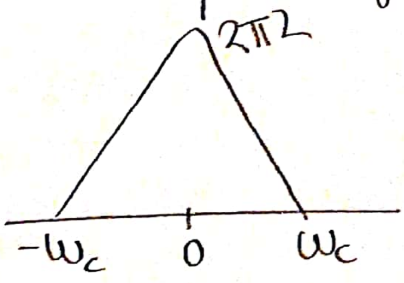
$\Rightarrow$



$* D(\omega) =$



pasando por el filtro $\Rightarrow$



$= Y(\omega)$ NO NULA.

$\therefore$ a) NO NULA

b) NO NULA

c) NULA

d) NULA

e) NO NULA

b) ¿ frecuencia oscilador local ?

Datos : · portadora ubicada en 550 KHz = f_c

· frecuencia intermedia del receptor 455 KHz = f_{FI}

La formula de frecuencia del oscilador local :

$$f_{OL} = f_c \pm f_{FI}$$

$$\Rightarrow \quad f_{OL} = 550\ KHz \pm 455\ KHz$$

(1) $f_{OL} = 550\ KHz + 455\ KHz = 1005\ KHz$

(2) $f_{OL} = 550\ KHz - 455\ Kz = 95\ KHz$

No existe solución única

c) $\phi_{FM}(t) = 10 \cos(2 \cdot 10^7 \cdot \pi \cdot t + 20 \cdot \cos 1000\ \pi t)$

Además, se tiene la sgte fórmula :

$\phi_{FM}(t) = A \cos(w_c t + \beta \operatorname{sen} w_m t)$, $\beta = \frac{\Delta w}{w_m}$

en la cual identificaremos términos.

Por fórmulla de Carson, se obtiene el ancho de banda

como $W \approx 2(\Delta W + W_m) = 2 W_m (1 + \beta)$

Identificando términos se tiene :

$A = 10$

$W_c = 2 \cdot 10^7 \cdot \pi$

$\beta = 20$

$W_m = 1000\pi$

$$W \approx 2 w_m (1 + \beta)$$

$$\Rightarrow W \approx 2 \cdot 1000 \cdot \pi (1 + 20)$$

$$\boxed{W \approx 42000\pi}$$

P4) a) La foto es de 600×400 píxeles, es decir, en total tendría 240000 píxeles

Para una imagen blanco y negro cada pixel tendrá 1 bits ⇒ multiplicando para obtener la cantidad de bits

$240000 \cdot 1 = 240.000$ bits

El formato NTSC indica una velocidad de 30 bits por segundo

$\Rightarrow 240.000 \cdot 30 = 7.200.000$ [bits por segundo]

Aplicando en teorema de Shannon-Hartley

$C = B \cdot \log_2 \left(1 + \frac{S}{N}\right)$ tal que

$C = 7.200.000$ [bits por segundo]

B = banda ancho en Hz

$\frac{S}{N} = \frac{\text{señal}}{\text{ruido}}$

Suponiendo $\frac{S}{N} \approx n^2 - 1 \approx n^2$ y como las fotos en blanco y negro son $n = 1$ $\Rightarrow \frac{S}{N} = 1$

$$\Rightarrow 7\,200.000 = B \cdot \underbrace{\log_2(2)}_{1}$$

$$\Rightarrow B = 7.200.000 = 7{,}2 \text{ MHz}$$

banda de ancho mínimo aproximado