**Juan Camilo Guzmán - 2175559**

**Juan Manuel Becerra - 2175775**

**Jose Alejandro López - 2175425**

**Juan Felipe Peña - 2175507**

**LABORATORIO #2**

**Primer punto:**

1. **i.** Para este punto primero realizamos la serie trigonométrica de fourier. Después de hallarla, la complementamos con la parte compleja Analizamos la señal a encontrar y era:

b[k]=0 Ya que es PAR

a[0]=0

También vemos que tiene simetría de media onda por ende los a[k] pares son 0.

k=1=, 2+2=

k=2=, 0+0=0

k=3=, -2-2=

Compleja:

Hallamos C[k] a partir de a[k] hallado

para k=1 c[k]=

para k=2 c[k]=0

para k=3 c[k]=

**ii.** Posteriormente metimos la respuesta compleja en Octave con un respectivo ciclo for con valores de k para poder graficar la señal cuadrada correspondiente (se puede evidenciar en el código).

**iii.** Para este punto ajustamos los parámetros correspondientes al documento para poder hallar x(n) y así poder reproducir un sonido usando **sound(real(x),fs).** La onda tiene un sonido brusco y medio largo, simulando el sonido de llamada de un teléfono.

**iv.** Para este punto usamos los mismo incisos del punto anterior pero cambiamos la frecuencia por 420 Hz. Ahora el sonido se agudiza un poco.

**v.** De una forma subjetiva, buscamos un rango con los valores de k para lograr que el sonido previamente reproducido se escuchara “bien” o mejor.

Para este caso usamos el ciclo for k=1:2:1, obteniendo un sonido limpio, entendiendo que sería la onda fundamental sin modificación suena mejor, pero solo es la base de nuestra onda deseada.

1. En este punto realizamos los pasos del apartado **a,** empleando una nueva integral, la cual fue:

a[0]=2

b[k]=0 Ya que es PAR

a[k] pares =0 ya que posee simetría de media onda

Compleja:

**Segundo punto:**

1. Tomamos la escala A mayor para formar las notas de una parte de la

siguiente canción infantil:

notas=[ 3 2 1 2 3 3 3 2 2 2 3 5 5 3 2 1 2 3 3 3 2 2 3 2 1]

Donde los números dentro del vector (notas) son los índices de las

frecuencias pertenecientes a la escala A mayor. Realizamos un código

el cual usa el sintetizador de la parte 1 para generar tonos

de 0.5 segundos para cada nota. Pero ahora reproducimos la canción usando

**sound(real(y), Fs).**

1. Modificamos el código creando un vector de duración para cada nota:

duraciones=[0.5 0.5 0.5 0.5 0.5 0.5 1 0.5 0.5 1 0.5 0.5 1 0.5 0.5 0.5

0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 2]

Y asociamos la duración a cada tono.

**Tercer punto:**

1. **Pitch:** Cualidad que determina si un sonido es agudo o grave, la cual dependerá de la frecuencia en que se repita la onda sonora. A mayor repetición de la onda, más agudo será el sonido y a menor repetición, más grave será el mismo. Por esta razón, las notas musicales están determinadas por un número de frecuencia.

**Utterance:** Es la mínima unidad del habla que comienza y termina con una pausa clara, esta determina el contexto de la oración.

1. 2 segundos si es un tiempo adecuado para poder realizar las pruebas correspondientes con las palabras: derecha, izquierda y saltar.
2. Primero definimos nuestra ventana en este caso teniendo en cuenta la atenuación de la banda de rechazo 50 dB como criterio, tomando en cuenta esto usando la ventana tipo **hamming,** primero definimos nuestro orden en 40:

Para M=40

Tendríamos una banda de transición de:

**Largo**= 40+1=41

**Asi tendriamos nuestra ventana**:

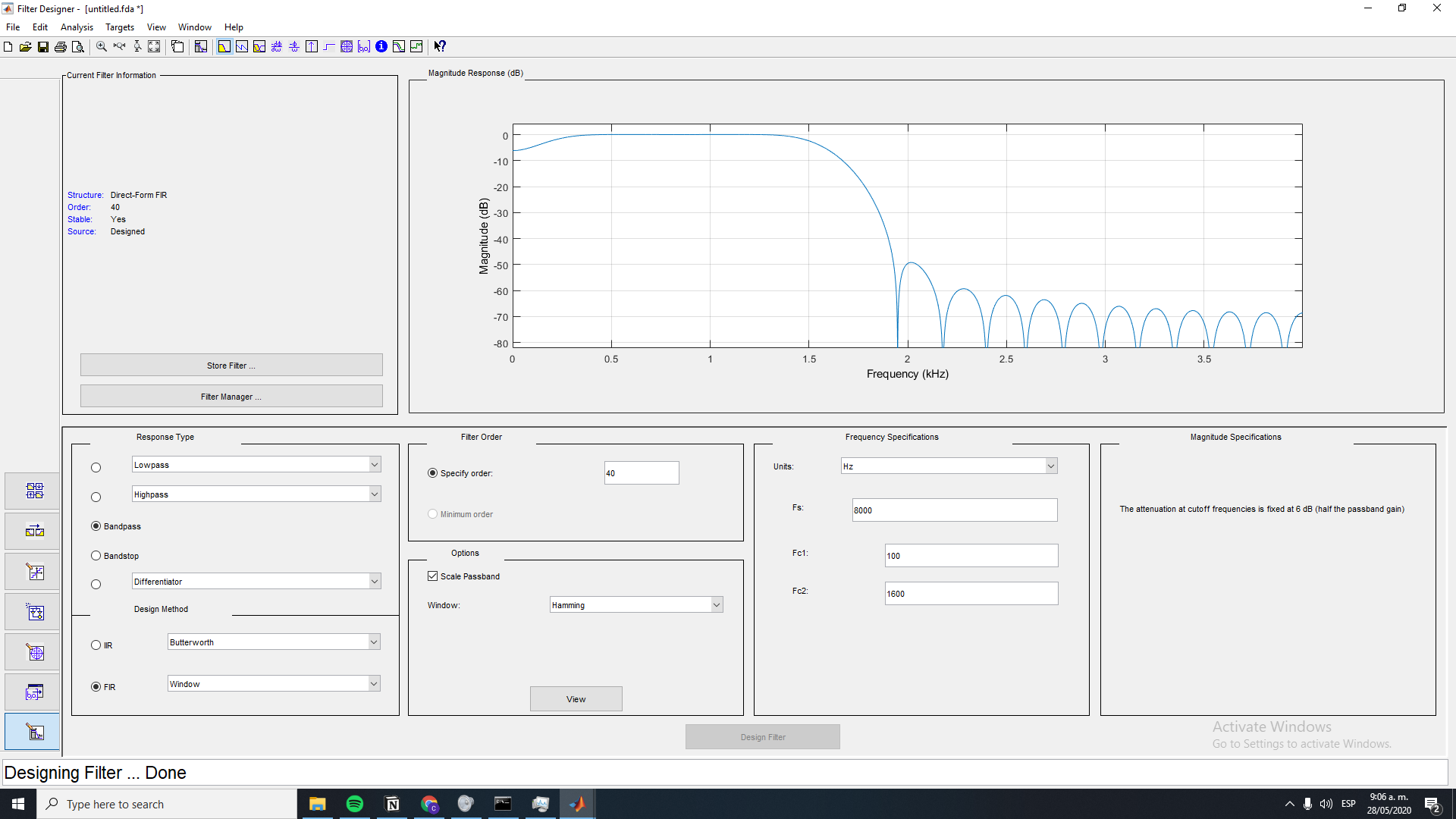
**Frecuencia de corte**:

**Pasa baja**:

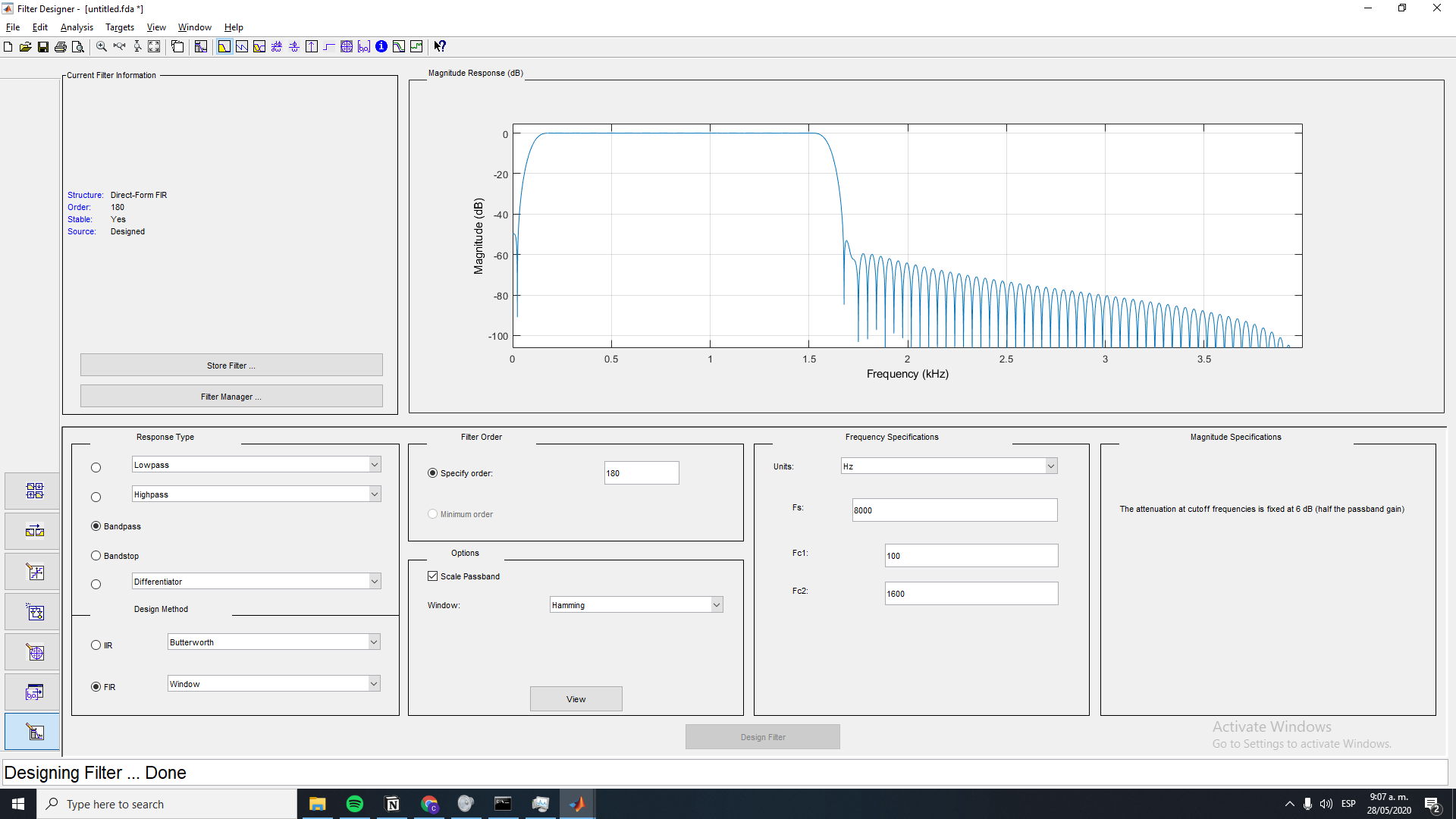
**PasaBanda F1=100 Hz-1600 Hz**

**Pasa banda:**

* Usando el comando sound para reproducir la señal filtrada y notamos que usando el pasa bajas elimina ruido procedente de las frecuencias altas y limpia bastante la señal pero al filtrarlo con el pasa bandas, notamos que un ruido grave de las frecuencias bajas también se elimina, pero notamos que en la primera frecuencia de corte, la banda de transición es muy grande y no elimina las frecuencias deseadas en su totalidad, por se tomó la decisión de subir el orden a 180 para disminuir la banda de transición y tener la atenuación deseada.

Para M=40

Para M=180



M=180

**Largo**= 180+1=181

**Pasa baja**:

**Pasa banda:**

1. Para este punto realizamos una normalización para lograr que nuestro punto anterior quedará más sensitivo y detallado a la hora de la pronunciación y el acento. Aparte de lo anterior, graficamos cada señal logrando que ambas señales quedarán en un rango entre -1 y 1.
2. Convertimos nuestra de señal de sonido a componentes de magnitud usando la fft. Posteriormente elevamos al cuadrado la muestra obtenida para poder trabajarla mejor.
3. Realizamos 10 audios por cada palabra (“DERECHA”, “IZQUIERDA” y “SALTAR”) así creamos nuestra base de datos de fft^2 de cada palabra. Posteriormente promediamos la fft^2 de cada palabra.
4. Después grabamos nuestros audios prueba, buscamos las diferencias entre fft^2 de la prueba y de cada palabra en la base de datos, utilizando la distancia euclidiana se halló la diferencia en frecuencia y decidimos que palabra es tomando como criterio la de menor diferencia.

**Código:**

En la carpeta Punto 3 el código filters lo usamos para probar los filtros, decidir cual usar y además guardar los 10 valores de cada palabra.

En la carpeta se encuentra los 10 datos de cada palabra. denotados por la palabra un número.

El código prom lo usamos para generar el promedio entre los datos de cada palabra.

Genera (Palabra+F.mat).

El código VoiceFinal toma los datos ya promediados, graba y reconoce la voz.