# LABORATORIO #6 TRANSFORMADA DE FOURIER DISCRETA

Profesor: M.C. Fernando Hermosillo Reynoso

# **Nombre del Alumno:**

**ID:**

Instrucciones; *Lee cuidadosamente y contesta lo que se te pregunta*.

1. **OBJETIVOS**

Al finalizar este laboratorio el alumno:

1. Será capaz de comprenderá la importancia de la transformada de Fourier Discreta (DFT).
2. Aterrizara los conceptos vistos durante el análisis del dominio de la frecuencia.
3. Así mismo será capaz de aplicar la DFT para calcular la respuesta de un sistema LTI.
4. **MATERIAL**

* Microcontrolador ESP32
* Tarjeta de audio I2S MAX98357/PCM5102
* Bocina de 4/8 Ω o en su defecto auriculares con conexión Jack 3.5mm

1. **MARCO TEÓRICO**
2. **Filtros Digitales**

Como se ha hecho hincapié a lo largo del curso, las señales sinusoidales son una parte importante en el análisis de señales, dado que una amplia gama de señales del mundo real puede ser expresadas como una combinación lineal de señales senoidales (véase la Ecuación 1).

La representación de una señal discreta en el dominio del tiempo provee información importante que permite describirla, como bien pudiera ser el periodo, en caso de ser una señal periódica, su amplitud y duración. Sin embargo, en el mundo real, las señales discretizadas comúnmente no tienen una representación determinística, es decir una expresión matemática que permita obtener esta serie de parámetros con facilidad, o simplemente no nos provee la suficiente información que describe a la señal. Una solución a dicho problema, es el aplicar una transformación matemática a dicha señal, que nos permitan expresar las señales discretas por medio de expresiones más sencillas de analizar, como lo son las señales sinusoidales. Este modelo de descomposición en señales sinusoidales se conoce como la Transformada de Fourier, siendo el objeto de análisis de hoy.

La Transformada de Fourier Discreta (DFT) es una técnica matemática que se utiliza para analizar señales digitales en el dominio de la frecuencia. Es una herramienta importante en el procesamiento de señales digitales, la comunicación digital y la teoría de la información.

Para una señal discreta de longitud N (muestras), la DFT se define como una secuencia de valores discretos complejos , que permite descomponer a la señal como una serie de señales senoidales ponderadas en amplitud y cuya frecuencia se encuentra distribuida uniformemente desde to , a intervalos de . Es decir, se puede considerar a la DFT como una versión discreta de la Transformada de Fourier, y es expresada de manera forma como a continuación:

Así mismo, a fin de recuperar la señal en el dominio del tiempo, existe una expresión que permite sintetizar (formar) la señal original, conocida como la Transformada Discreta Inversa de Fourier (IDFT):

En donde , y comúnmente se le conoce como el espectro de frecuencia de la señal .

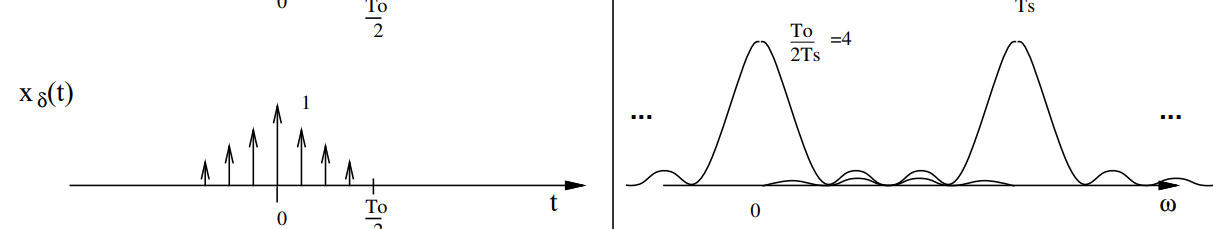
El cálculo de la DFT mediante la expresión anterior puede ser computacional-mente costoso, ya que requiere multiplicaciones. Sin embargo, existe un algoritmo llamado Transformada de Fourier Rápida (FFT) que reduce el tiempo de cálculo de la DFT a multiplicaciones, e implementa la DFT de manera recursiva.

A fin de conocer las frecuencias que están asociadas al k-eximo coeficiente , se especifica un parámetro en la DFT conocido como la resolución espectral , que básicamente se calcula como:

Esto es, para k=0, la frecuencia asociada es 0, para k = 1, la frecuencia es , etc.

Finalmente, si quiere calcular los valores de amplitud asociados a cada coeficiente , básta con calcular la magnitud del k-esimo coeficiente:

En resumen, la DFT permite obtener un espectro discreto de frecuencia, o básicamente de la DTFT , véase Figura 1, de donde se sabe que este espectro es periódico en con un periodo de , con la diferencia de que es periódico, pero de periodo N. Nótese además que es complejo.





Finalmente, una de las aplicaciones mas importantes de la DFT es el cálculo eficiente (rápido) de la convolución para sistemas. Recordando que la convolución en frecuencia es prácticamente una multiplicación:

Entonces el cálculo de la convolución

Operación que cuando es relativamente grande, es calculada con menos cantidad de operaciones que la convolución tradicional.

1. **DESARROLLO**

En la plataforma Moodle, descargue la plantilla de Arduino (.ino) asociada al laboratorio #6.

1. Diseñe dos funciones en C llamadas dft e idft que permitan calcular la Transformada Discreta de Fourier y su inversa para una señal de longitud N, mediante la definición matemática, cuyas funciones prototipo son las siguientes:

void dft(float \*x, uint32\_t N, complex\_t \*Xk);

void idft(complex\_t \*Xk, uint32\_t N, float \*x);

* 1. A fin de calcular la exponencial compleja, puede emplear diversos algoritmos, el más básico consiste en descomponer la exponencial compleja en sus coordenadas cartesianas
  2. Ejemplo: si desea multiplicar
  3. La estructura , se tiene que definir como una estructura

typedef struct {

float re;

float im;

} complex\_t;

* 1. Entonces la operación del ejemplo del inciso b), se realizaría como a continuación:

complex\_t z;

z.re = 2.3\*cos(M\_PI/3);

z.im = 2.3\*sin(M\_PI/3);

1. Conecte su teléfono a la tarjeta ESP32 por medio de Bluetooth, donde en su teléfono deberá de estar transmitiendo audio hacia la ESP32, en este caso señales senoidales (<https://onlinetonegenerator.com/>).
2. En su código del laboratorio #6, deberá de aplicar la DFT a la señal muestreada asociada al canal derecho o izquierdo únicamente. Recuerde que cuando se transmite una señal de audio, comúnmente se transmiten dos canales, el izquierdo y el derecho, por lo que debe de guardar primeramente la información del canal a analizar previamente en un arreglo antes de aplicar la función dft. Como sugerencia, no tome todo el bloque de muestras asociadas a cada medición, por el contrario, use las primeras 128 muestras del canal a analizar.
3. Identifique el valor máximo de la magnitud de sin contemplar la frecuencia cero, a este valor lo vamos a llamar la magnitud de la frecuencia dominante. Imprima en consola serial la frecuencia dominante.

Cálculo de la convolución por medio de la FFT:

1. En MATLAB u Octave, diseñe un filtro pasa-bajas con el método de ventaneo, para ello:
   1. Defina su frecuencia de corte discreta normalizada como a continuación:

Donde

* 1. Ejecute el comando *h = fir1(M,fc);*, donde M es el orden del filtro (número de coeficientes) y fc es la frecuencia de corte normalizada.
  2. Esta función permite calcular la respuesta al impulso truncada de un filtro pasa bajas ideal. Para mayor información visite la página:

<https://la.mathworks.com/help/signal/ref/fir1.html>

* 1. La frecuencia de corte deberá de ser única por cada equipo, y deberá ser un número entre 0 y para este experimento.
  2. El orden del filtro M deberá ser potencia de 2, por ejemplo 2, 4, 8, 16, 32, …, con la restricción que sea >= 32.
  3. Use de nuevo la plantilla de Arduino del laboratorio #6, a fin de calcular la salida del filtro por medio de la operación de convolución
  4. Como nota, a fin de poder calcular la salida, debe de utilizar bloques de M muestras de la señal de audio muestreada, esto es, si la cantidad de muestras que se adquieren en la señal de audio es de 1024, y M = 32, debera procesar en bloques de 32 muestras la convolución por la DFT. A continuación se muestra un pseudocódigo

N=1024;

M=32;

P=N/M; // Número de subloques

n = 0;

complex\_t Xk[M];

complex\_t Hk[M];

complex\_t Yk[M];

float y[N];

dft(h,M,Hk); // Calcular H(k)

for p = 0; p < P; p++

// Almacenar las muestras en un bloque de M muestras

float xblock[M];

for m = 0; m < M; m++

xblock[m] = x[n++];

end

dft(xblock,M,Xk); // DFT del bloque de M muestras

// Multiplicar X(k) por H(k)

for k = 0; k < M; k++

Yk[k] = Xk[k] \* H[k];

end

// Obtener la respuesta en el dominio del tiempo

idft(Yk,M,(y+n-M));

end

* 1. Notese que “y” es un puntero, y de tamaño N, sin embargo, la IDFT debe de ser calculada para solo M muestras, M < N. En la línea final, (y + n - M) permite tomar el punto y[n-M], a fin de sobrescribir solamente en las direcciones que correspondan con el bloque a procesar, por ejemplo cuando se procesa el primer bloque, se debe de procesar de la muestra x(0) hasta la muestra x(M-1), y la salida por igual debe estar en ese rango (y(0) a y(M-1)). Por otra parte para el siguiente bloque, se debe de procesar de x(M) hasta x(2M-1), y la salida por igual debe estar en ese rango(y(M) a y(2M-1)). Por lo que ese desplazamiento de n – M, realiza justamente esa corrección en el índice de inicio que se desea procesar.
  2. Añada la respuesta en frecuencia del filtro diseñado (ver notas del laboratorio #5).

1. **CUESTIONARIO**

Responda a las siguientes preguntas:

1. Se sabe que la respuesta en frecuencia de un sistema esta dada por , si se tienen los coeficientes del sistema y que forman la ecuación en diferencias del sistema (ver ecuación siguiente), ¿Sería posible aproximar la respuesta en frecuencia ? ¿Cómo se podría aproximar?
2. Sea la respuesta en frecuencia de un filtro promediador, Grafique la respuesta en frecuencia en Matlab usando el comando:
   1. Recuerde definir primero a como un vector.
   2. En este caso, h representa los coeficientes de “” y “1” representa los coeficientes de “”, mientras que “100” representa el parámetro N en la DFT.
3. En MATLAB, para calcular la DFT se utiliza el comando Xk=fft(x,N); donde N es la cantidad de muestras que se van a emplear para discretizar a .
   1. Calcule la DFT de del ejercicio 2, utilizando 100 muestras: B=fft(h,100);
   2. Calcule la DFT de “1”, utilizando 100 muestras: A=fft(1,100);
   3. Divida el resultado del inciso a), sobre el resultado obtenido en el inciso b), y grafique su magnitud utilizando el comando plot(w, abs(B./A).
   4. El vector w debe ser w=0: (2\*pi)/100 : 2\*pi; y es básicamente el vector de frecuencia discreta que va de 0 a 2 pi, muestreado uniformemente con 100 muestras.
   5. ¿Tiene alguna relación con el resultado obtenido por el comando freqz?
   6. Grafique además las magnitudes de A y B obtenidos en los incisos a) y b) respectivamente.
   7. Repita este experimento utilizando “1000” muestras en lugar de “100”.
   8. ¿Existe diferencias entre los espectros obtenidos para 100 y para 1000? ¿Qué cambios se produjeron?