

Teoría de Sistemas y Señales

Trabajo Práctico Nº 3 Análisis Frecuencial de Señales

Cátedra TeSyS

Mayo de 2014

Trabajo Práctico N° 3

Análisis Frecuencial de Señales

1. Objetivos

El objetivo de este Trabajo Práctico es estudiar el uso de la Transformada de Fourier en Tiempo Discreto (DTFT) para el análisis espectral de señales, a la vez que la implementación en **Matlab** de algoritmos de cómputo de la DTFT, y el estudio de la performance de diferentes tipos de ventanas empleadas por los algoritmos.

2. Introducción

En este Trabajo Práctico se estudiarán las propiedades fundamentales de la Transformada de Fourier en Tiempo Discreto en su uso para el análisis espectral de señales. Para algunas clases de señales (nominalmente aquellas que son de longitud finita, o aquellas que tienen una transformada Z del tipo racional) es posible implementar algoritmos que calculen exactamente la DTFT. Para estas señales se requerirá la programación de funciones **Matlab** para el cómputo de la DTFT. Esto se hace en la Subsección 3.1.1.. Para el caso de señales de longitud infinita, o que no tengan una Transformada Z racional, no es posible implementar algoritmos para el cómputo exacto de la DTFT. En estos casos se debe recurrir al uso de funciones *ventana* que trunquen la señal a un número finito de muestras, y permitan la implementación de algoritmos que aproximen la DTFT de la señal. El estudio de varios tipos de ventanas utilizadas en la práctica se realiza en la Subsección 3.1.2.. En la Subsección 3.1.3., se analizan las características de los algoritmos **Matlab** para el cálculo de la Transformada de Fourier en Tiempo Discreto (genéricamente denominados Transformada Discreta de Fourier (DFT)), en su versión conocida como Transformada Rápida de Fourier (FFT: Fast Fourier Transform). En esta misma Subsección se propone un ejemplo de aplicación para la detección de una señal senoidal inmersa en ruido. Finalmente, en la Subsección 3.2 se presenta el enunciado de los problemas que deberán ser resueltos y entregados en el informe.

3. Desarrollo del Trabajo Práctico

3.1. Problemas Propuestos de Simulación durante la Sesión de Laboratorio

3.1.1. Transformada de Fourier en Tiempo Discreto (DTFT)

El objetivo de esta primera subsección es estudiar las propiedades de la Transformada de Fourier en Tiempo Discreto (DTFT: Discrete-Time Fourier Transform), a la vez que desarrollar e implementar algoritmos en **Matlab** [4] para el cálculo exacto de la misma. Consideraremos dos casos diferentes: señales de longitud finita, para las cuales la DTFT puede calcularse exactamente a partir de la definición, y señales de longitud infinita pero del tipo exponencial, que poseen una Transformada Z racional, para las cuales puede calcularse una expresión analítica de la DTFT que permite también para este caso su cálculo exacto, que de lo contrario sería imposible [1].

Recordemos que para una señal en tiempo discreto $x(n)$, la DTFT viene dada por [2]:

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}, \quad -\pi \leq \omega \leq \pi \quad (1)$$

que resulta una función a valores complejos de la variable ω , que es periódica, con período fundamental 2π . En el contexto de **Matlab**, el cómputo de la DTFT basándose en la definición (1) presenta básicamente dos inconvenientes:

- i. No es posible usar la expresión (1) para el cálculo de la DTFT cuando la señal es de longitud infinita, ya que **Matlab** sólo puede manipular vectores de longitud finita. Hay sin embargo una excepción a esto, que es el caso en que pueda obtenerse una expresión analítica de la DTFT, que pueda usarse para el cómputo. Un ejemplo de este caso son las señales exponenciales discretas de la forma $a^n \mu(n)$, que tienen una DTFT racional en la variable $e^{j\omega}$.

- ii. La DTFT $X(\omega)$ es una función de la variable continua ω (que puede asumir infinitos valores en el intervalo $[-\pi, \pi)$). Para el cálculo con **Matlab**, esto representa un problema de *muestreo en frecuencia*. Lo mejor que se puede hacer es realizar el cálculo en un número finito de frecuencias, lo suficientemente grande como para obtener una aproximación *suave* de la DTFT.

Generalmente se toman frecuencias equiespaciadas en el intervalo $[-\pi, \pi)$, con lo que la expresión (1) resulta:

$$X(\omega_k) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega_k n} = \sum_{n=-\infty}^{\infty} x(n)e^{-j(2\pi k/N)n}, \quad k = 0, \dots, N-1 \quad (2)$$

Sin embargo, esta última expresión todavía no es computable, salvo para el caso en que la señal sea de longitud finita. En este caso, asumiendo que la señal sea de longitud L , la expresión (2) resulta

$$X(\omega_k) = \sum_{n=0}^{L-1} x(n)e^{-j\omega_k n} = \sum_{n=0}^{L-1} x(n)e^{-j(2\pi k/N)n}, \quad k = 0, \dots, N-1 \quad (3)$$

Es fácil ver que la expresión (3) corresponde a la definición de la Transformada Discreta de Fourier (DFT) con N puntos, que para el caso $N = L$ puede ser calculada eficientemente usando alguno de los algoritmos de la Transformada Rápida de Fourier (FFT: Fast Fourier Transform).

A continuación se proponen algunos problemas que requieren la programación de funciones **Matlab**, para el cálculo de la DTFT.

Problema 1:

- a. Escriba una función **Matlab** que compute exactamente la DTFT de una señal $x(n)$ de longitud finita L .
 - Los argumentos de entrada de la función deben ser:
 - i. x : vector de longitud L conteniendo las muestras de la señal $x(n)$.
 - ii. N : número de frecuencias equiespaciadas en el intervalo $[-\pi, \pi)$ donde la DTFT es evaluada.
 - Los argumentos de salida de la función deben ser:
 - i. X : vector (complejo) conteniendo los valores de la DTFT $X(\omega)$.
 - ii. ω : vector conteniendo las frecuencias (en rad/seg) donde la DTFT es evaluada.
 - La función deberá así mismo graficar los espectros de amplitud y fase de la señal, y el espectro de densidad de energía.
 - La función debe incluir un "help" que pueda invocarse con el comando

>> help function_name

- b. Considere la señal

$$x(n) = \begin{cases} (0,9)^n & 0 \leq n \leq 100 \\ 0 & \text{c.o.c.} \end{cases}$$

y use la función desarrollada en el apartado a. para calcular la DTFT.

- c. **Trabajo previo:** Verifique el resultado calculando una expresión analítica de la DTFT y comparando la gráfica correspondiente con la obtenida con la función **Matlab**.

□

Problema 2: En este problema consideraremos señales de longitud infinita, pero que poseen una DTFT del tipo racional en la variable $e^{j\omega}$. Esto es equivalente a decir que la señal tenga una Transformada Z del tipo racional en la variable z (o z^{-1}). En efecto, recordando la definición de la Transformada Z de una señal

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}, \quad (4)$$

puede verse que si la Región de Convergencia (RDC) de la Transformada Z incluye a la circunferencia de radio unitario, entonces la DTFT $X(\omega)$ de la señal puede calcularse evaluando la transformada Z sobre la circunferencia unitaria (es decir para $z = e^{j\omega}$).

- a. Escriba una función **Matlab** que compute exactamente la DTFT de una señal de longitud infinita, cuya transformada Z es una función racional de la variable z .
 - Los argumentos de entrada de la función deben ser
 - i. num: vector conteniendo los coeficientes del polinomio numerador de la Transformada Z de la señal, en orden decreciente de potencias de z .
 - ii. den: vector conteniendo los coeficientes del polinomio denominador de la Transformada Z de la señal, en orden decreciente de potencias de z .
 - iii. N: número de frecuencias equiespaciadas en el intervalo $[-\pi, \pi)$, donde la Transformada de Fourier es evaluada.
 - Los argumentos de salida deben ser
 - i. X: vector conteniendo los valores de la DTFT.
 - ii. w: vector conteniendo las frecuencias donde la DTFT es evaluada.
 - La función deberá así mismo graficar los espectros de amplitud y fase de la señal, y el espectro de densidad de energía, en función de la frecuencia ω .
 - La función debe incluir un "help" que pueda invocarse con el comando

>> help function_name

NB: Para la implementación de esta función puede hacer uso de la función de **Matlab** `polyval(u1,u2)` que permite evaluar el polinomio, cuyos coeficientes en orden decreciente de potencias se ingresan como primer argumento `u1`, en el segundo argumento `u2`.

- b. Considere la señal

$$x(n) = \left(\frac{1}{2}\right)^{n-2} \mu(n-2) + \left(\frac{7}{8}\right)^n \mu(n),$$

y use la función desarrollada en el apartado a. para calcular la DTFT.

- c. **Trabajo Previo:** Verifique el resultado calculando una expresión analítica de la DTFT
- d. Compare la gráfica correspondiente con la obtenida con la función **Matlab**.

□

3.1.2. Cómputo de la DTFT usando ventanas

En esta Subsección estudiaremos el uso de ventanas para el cómputo de la DTFT de señales de longitud infinita, que no necesariamente tienen una transformada Z racional. Como se vió en la Subsección anterior, no es posible usar la expresión (1) para el cómputo de la DTFT de una señal de longitud infinita, excepto para el caso en que la señal tenga una transformada Z racional. La alternativa es entonces truncar la señal a un número finito de muestras usando lo que se denomina una *función de ventana*, y luego computar la DTFT de la señal truncada que aproximará la DTFT de la señal original, bajo ciertas condiciones. Numerosos tipos diferentes de ventanas han sido propuestos en la literatura para su uso en

análisis espectral de señales (ver [1],[2] para más detalles). El tipo más común es la denominada *ventana rectangular* que sólo trunca la señal a un número finito de muestras, sin afectar el valor de las muestras. En el dominio temporal, la ventana rectangular de longitud L viene dada por:

$$w(n) = \begin{cases} 1 & 1 \leq n \leq L \\ 0 & \text{c.o.c.} \end{cases} \quad (5)$$

Si $x(n)$ es la señal de longitud infinita, la señal truncada $\hat{x}(n)$ se puede obtener como

$$\hat{x}(n) = x(n)w(n) . \quad (6)$$

La DTFT, $X(\omega)$, de la señal original está entonces relacionada con la DTFT, $\hat{X}(\omega)$, de la correspondiente señal truncada, a través de la convolución (periódica) en el dominio frecuencial

$$\hat{X}(\omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\lambda)W(\omega - \lambda)d\lambda . \quad (7)$$

Puede verse entonces que al aproximar el espectro $X(\omega)$, de la señal original, por el espectro, $\hat{X}(\omega)$, de la correspondiente señal truncada, se comete un error que dependerá de la forma de la función de ventana $w(n)$ (o de su espectro $W(\omega)$) a través de la relación (7).

Se proponen a continuación algunos problemas donde se estudiarán las características espectrales de algunas de las funciones de ventana disponibles en **Matlab** [3] (nominalmente las funciones `hamming`, `hanning` y `boxcar`, que implementan las ventanas de Hamming, Hann, y rectangular respectivamente), y se analizará el error introducido en la aproximación de $X(\omega)$ por $\hat{X}(\omega)$.

Problema 3:

Las funciones **Matlab** `hamming`, `hanning` y `boxcar`, permiten implementar las funciones de ventana de Hamming, Hann, y rectangular respectivamente. Estas funciones están definidas en el dominio temporal como [3]:

- Ventana Hamming (`hamming`)

$$w(n+1) = \begin{cases} 0.54 - 0.46 \cos\left(\frac{2\pi n}{L-1}\right) & 0 \leq n \leq L-1 \\ 0 & \text{c.o.c.} \end{cases}$$

- Ventana de Hann (`hanning`)

$$w(n) = \begin{cases} \frac{1}{2} \left(1 - \cos\left(\frac{2\pi n}{L+1}\right) \right) & 1 \leq n \leq L \\ 0 & \text{c.o.c.} \end{cases}$$

- Ventana Rectangular (`boxcar`)

$$w(n) = \begin{cases} 1 & 1 \leq n \leq L \\ 0 & \text{c.o.c.} \end{cases}$$

- Trabajo Previo:** Calcule la DTFT de la ventana rectangular en forma teórica y determine los cruces por ceros del espectro de amplitud.
- Utilice la función **Matlab** implementada en el Problema 1 para computar el espectro de amplitud de la ventana rectangular. En una misma gráfica muestre superpuestos el espectro para tres longitudes L distintas, $L = 20, 31, 81$. Use una escala lineal para el espectro de amplitud.

- c. Mida la amplitud y el ancho del lóbulo principal de la DTFT. Analice los puntos de cruces por cero y la amplitud de los lóbulos laterales. Además, verifique la ubicación de los puntos de cruces por cero a partir del cálculo analítico de la DTFT de un pulso rectangular.

NOTA: Recuerde que puede usar el comando `ginput` para extraer valores de una gráfica en **Matlab**.

- d. Considere la siguiente señal de longitud infinita

$$x(n) = \cos\left(\frac{\pi}{2}n\right)$$

Utilice una ventana rectangular de longitud $L=45$ para computar una aproximación del espectro de la señal y graficarlo.

- e. Sea la señal

$$x(n) = \cos(0.5\pi.n) + \cos(0.51\pi.n)$$

Use la ventana rectangular para computar y graficar una aproximación del espectro de la señal $x(n)$. Determine la relación entre la resolución en frecuencia y la longitud L de la ventana. A partir de este resultado, calcule la longitud L mínima de modo que se distingan las componentes armónicas de la señal $x(n)$.

□

3.1.3. Transformada Discreta de Fourier (DFT)

En esta sección se estudiará el uso de la Transformada Discreta de Fourier en el análisis espectral de señales. Particularmente se considerará la implementación de la DFT en un algoritmo del tipo FFT (Fast Fourier Transform) disponible en **Matlab** (la función `fft`). Como un problema de aplicación se empleará la DFT para determinar la amplitud y frecuencia de una señal senoidal inmersa en ruido.

Recordemos que para una señal en tiempo discreto $x(n)$, de longitud finita L , la Transformada de Fourier en Tiempo Discreto viene dada por:

$$X(\omega) = \sum_{n=0}^{L-1} x(n)e^{-j\omega n}, \quad 0 \leq \omega \leq 2\pi$$

Si se muestrea $X(\omega)$ en frecuencias equiespaciadas $\omega_k = \frac{2\pi k}{N}$, $k = 0, 1, \dots, N-1$, y donde $N \geq L$,

las muestras resultan:

$$X(k) = X\left(\frac{2\pi k}{N}\right) = \sum_{n=0}^{L-1} x(n)e^{-j2\pi kn/N}$$

o equivalentemente

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1$$

ya que $x(n) = 0$ para $L \leq n \leq N-1$. La última ecuación constituye la definición de la denominada **Transformada Discreta de Fourier con N puntos** (N -point DFT) [2]. Existen diversas formas eficientes de implementar computacionalmente la DFT. Una de las más difundidas es la denominada Transformada Rápida de Fourier (FFT: Fast Fourier Transform), que permite computar la DFT con un tiempo de ejecución reducido. En **Matlab**, la FFT está implementada en la función `fft`. Esta función emplea un algoritmo radix-2 si la longitud de la secuencia es una potencia entera de 2 (ver [2], Sección 9.3.3 en Cap 9, para mayores detalles), y un algoritmo más lento si no lo es.

Dada una señal almacenada en el vector \mathbf{x} , el siguiente comando

```
>> y = fft(x);
```

computará la DFT con $N=\text{length}(x)$ puntos de la secuencia x . El número de puntos de la DFT puede especificarse independientemente de la longitud de la secuencia mediante un argumento de entrada adicional de la función. Por ejemplo, los comandos para computar la FFT con $N=100$ puntos serían:

```
>> N = 100;
>> y = fft(x,N);
```

Si N es menor que la longitud de x , entonces el vector x es completado con ceros hasta la longitud N . En caso de ser mayor, la secuencia es truncada a una longitud igual a N .

Nota: La función `fft` no genera el vector de frecuencias, que debe ser generado independientemente. La función devuelve los valores de la DFT que corresponderían a frecuencias entre 0 y 2π .

El retorno al dominio temporal puede realizarse mediante el cómputo de la IDFT, utilizando para ello la función **Matlab** `ifft`, de la siguiente manera:

```
>> x = real(ifft(y));
```

donde el vector y corresponde a la DFT de la señal x , y x es la señal temporal original. Cabe aclarar que se toma la parte real de la señal obtenida mediante la función `ifft`, dado que la señal resultante puede presentar valores complejos con parte imaginaria muy pequeña debido a errores numéricos.

También puede especificarse el número N de puntos considerados en el cálculo de la IDFT, el cual debe coincidir con el valor de N usado previamente en la función `fft`. Si se omite el valor N , la función `ifft` utilizará por defecto $N=\text{length}(y)$.

Nota: La función `ifft` de Matlab asume que el espectro se encuentra definido entre 0 y 2π .

Los siguientes ejemplos ilustran el uso de las funciones `fft` e `ifft`.

Ejemplo 1: Supongamos que se desea calcular la DFT de la señal en tiempo discreto $x(n)$ que se obtiene al muestrear la señal:

$$x(t) = 4 + 3\sin(2\pi 5t)$$

con una frecuencia $F_s = 100$ Hz. La siguiente sucesión de comandos **Matlab** permite generar la señal $x(n)$ y calcular y graficar la amplitud y fase de la DFT de $x(n)$.

```
t=0:0.01:1;
x=4+3*sin(5*pi*2*t);
y=fft(x,10000);
magnitud=abs(y);
fase=unwrap(angle(y));
frec=[0:2*pi/10000:2*pi];
frec=frec(:,1:10000);
subplot(211)
plot(frec,magnitud);xlabel('frecuencia w [rad/seg]');
ylabel('Magnitud');
title('Espectro de x(n)');axis([0 2*pi 0 200]);grid;
subplot(212)
plot(frec,fase*180/pi);xlabel('frecuencia w [rad/seg]');
ylabel('Fase [grados]');
axis([0 2*pi -4e4 2e4]);grid;
```

La salida gráfica correspondiente está representada en la Figura 3.1.1.

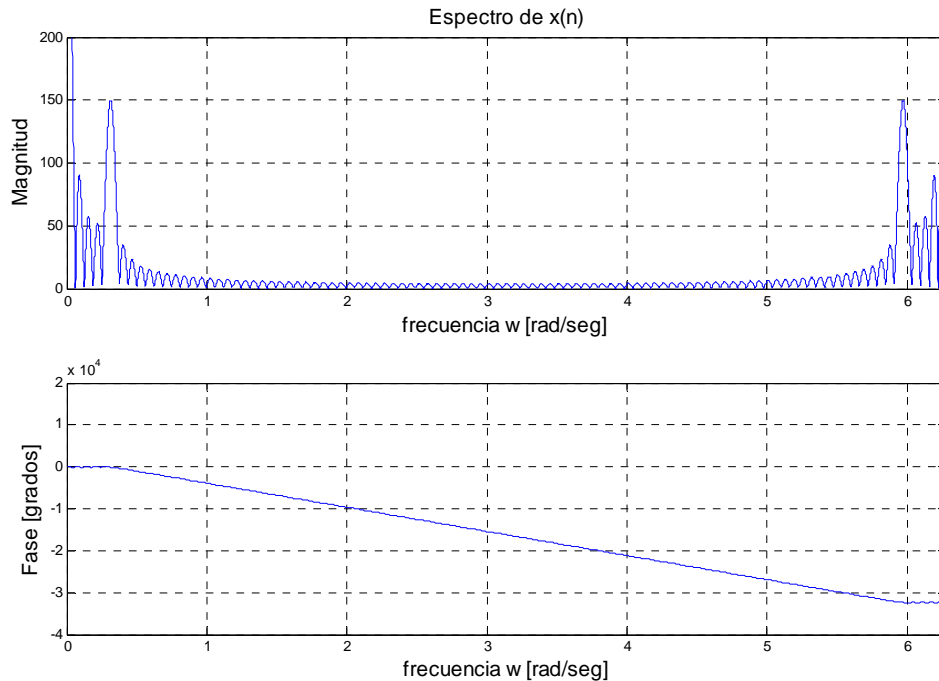


Figura 3.1.1: Salida gráfica correspondiente al Ejemplo 1.

Ejemplo 2: La siguiente secuencia de comandos permite obtener nuevamente la señal temporal a partir del cálculo de la IDFT, visualizando en una misma gráfica la señal original y el resultado obtenido a partir de la IDFT:

```
x_new=real(iff(y,10000));
figure, plot(t,x,'r-o',t,x_new(1:length(x)),'b-x');
title('Señal original vs. resultado de la IDFT');
xlabel('tiempo [segundos]');
ylabel('Amplitud');
legend('Señal original','Resultado de la IDFT');
grid;
```

La salida gráfica correspondiente se visualiza en la Figura 3.1.2.

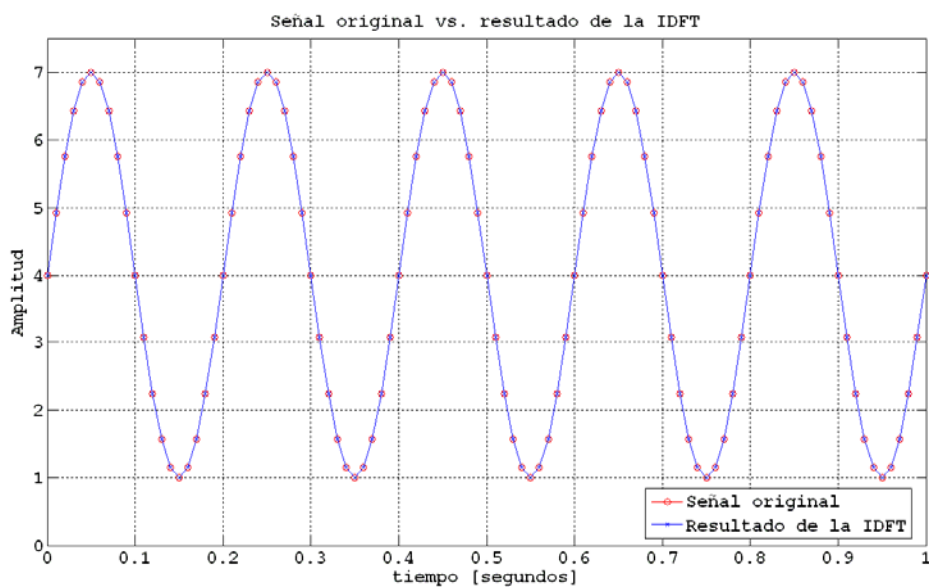


Figura 3.1.2: Salida gráfica correspondiente al Ejemplo 2.

Problema 4:

El archivo de datos **tp3_1.mat** contiene un vector x que consiste de las muestras de una señal senoidal inmersa en ruido, muestreada con una frecuencia de $F_s = 1$ KHz. Se asume que el ruido es blanco Gaussiano.

- a. Use la función `fft` de **Matlab** para computar y graficar la DTFT de la señal x , usando una ventana rectangular de longitud apropiada. Grafique la amplitud de la DTFT en por unidad, y la fase en grados, con el eje de abscisas calibrado en frecuencias continuas.
- b. Basándose en las gráficas obtenidas determine la amplitud y frecuencia de la señal senoidal. Justifique teóricamente los cálculos realizados.
- c. Repita los apartados **a.** y **b.**, pero usando ahora una ventana de Hann. Compare los resultados con los obtenidos en **a.** y **b.**
- d. Repita los apartados **a.** y **b.** pero usando una longitud de ventana igual a la mitad de la usada en **a.** y **b.** Compare los resultados obtenidos.

□

4. Referencias

- [1] Burrus, C. & McClellan, J. & Oppenheim, A. & Parks, T. & Schafer, R. & Schuessler, H. (1994). *Computer-based Exercises for Signal Processing using Matlab*, Prentice Hall, Englewood Cliffs, New Jersey.
- [2] Proakis, J. & Manolakis, D. (1992). *Digital Signal Processing: Principles, Algorithms and Applications*. 2nd Edition, Macmillan Publishing Company, New York.
- [3] The Math Works, Inc. (1996). *Signal Processing Toolbox - User's Guide, V.4*. The Math Works, Inc., 24 Prime Park Way, Natick, MA 01760-1500.
- [4] The MathWorks, Inc. (1997). *Matlab, The Language of Technical Computing – Getting Started with Matlab, Version 5.*, 24 Prime Park Way, Natick, MA 01760-1500.