

Universidad de las Américas Puebla

Decanato de Ingeniería

Departamento de Computación, Electrónica y Mecatrónica

Laboratorio de Sistemas de Telecomunicaciones

Práctica 3

Análisis de señales en Matlab

Integrantes:

Carlos Andrés Reyes Evangelista

157068

Erick Siordia Nagaya

157504

The logo of the Universidad de las Américas Puebla (UDLAP) is displayed in a large, bold, orange font. The letters are thick and blocky, with a registered trademark symbol (®) at the end of the word "UDLAP". The logo is centered horizontally and occupies a significant portion of the lower half of the page.

Análisis de señales con Matlab

Tabla de contenido

- [Resumen](#)
- [Objetivo](#)
- [Análisis teórico](#)
 - [Marco teórico](#)
 - [Análisis matemático del experimento](#)
 - [Resultados experimentales](#)
 - [1. Generar una señal sinusoidal](#)
 - [2. Discretizar la señal](#)
 - [3. Sobreponer las gráficas](#)
 - [4. Generar el espectro de frecuencias](#)
 - [5. Agregar ruido a la señal analógica.](#)
 - [6. Obtener el espectro de la señal con ruido](#)
- [Conclusión](#)
- [Bibliografía](#)

Resumen

Matlab es una herramienta que facilita la manipulación y operación de matrices y funciones matemáticas. Durante esta práctica se utilizó dicha herramienta para simular y visualizar algunas señales analógicas en diferentes dominios, lo cual permitió analizar de una forma más profunda las propiedades de las ondas y señales.

Algunas de las actividades realizadas para lograr este fin fueron el mostrar gráficamente una señal, discretizarla, agregar ruido, y usar diferentes funciones para visualizar su espectro de frecuencias. Fue gracias a estas actividades que se pudo tener un primer acercamiento a Matlab, y las operaciones básicas que este ofrece para realizar modelos matemáticos y, en este caso, el estudio de las propiedades de las ondas analógicas sinusoidales o cosenoidales.

Objetivo

Familiarizarse con el uso de Matlab como auxiliar en el manejo de señales usadas en sistemas de Telecomunicación

Análisis teórico

Marco teórico

Matlab es una herramienta que proporciona un ambiente matemático de trabajo y un lenguaje de programación particular que permite y facilita realizar operaciones complejas con matrices -de ahí su nombre- y funciones matemáticas en general. Esta versatilidad permite a Matlab ser considerada una herramienta útil para una amplia cantidad de ámbitos: procesamiento de imagen, procesamiento de audio, técnicas de graficación, y, entre otras, análisis de señales analógicas. Es justamente esta última utilidad la que justifica la utilización de este software para alcanzar el objetivo perseguido.

Análisis matemático del experimento

Durante esta práctica se emularon digitalmente ciertas señales analógicas para su consecuente análisis mediante gráficas en diversos dominios. A fin de generar señales con las características deseadas, el primer conocimiento requerido fue la comprensión de la fórmula general de una señal sinusoidal común, definida como sigue:

$$s(t) = A \sin(2\pi f t \pm \theta) \text{ ó } s(t) = A \cos(\omega t \pm \theta)$$

Donde la señal $s(t)$ obtenida se encuentra en función de la amplitud (A) de la señal, su frecuencia (f), desplazamiento de fase (θ) y el tiempo (t) transcurrido. O bien es también expresada en función de ω , que representa la *frecuencia angular* y es medida en radianes en lugar de Hertz. Las señales a presentar en este reporte se encuentran en esta forma. Si la amplitud o la frecuencia no son especificadas, se asume una amplitud o frecuencia por defecto de 1, mientras que si el desplazamiento de fase no es indicado, se asume un desplazamiento por defecto de 0 grados.

Resultados experimentales

Durante la realización de esta práctica fueron solicitadas una serie de tareas que se enlistan y detallan a continuación:

1. Generar una señal sinusoidal analógica de 10Hz. La frecuencia de muestreo debe de ser 100 veces mayor que la frecuencia de la señal.

Para realizar esta labor se requirió declarar cada uno de los componentes de la señal analógica para después ser utilizados en una ecuación como la descrita anteriormente para obtener una simulación de una señal analógica con dichas características. El código utilizado en Matlab es declarado a continuación y de inmediato se muestra la gráfica resultante en la *Figura 1* que refleja justamente una señal sinusoidal pura.

```
t = linspace(0, 1, 10000);    % Time
a = 1;                        % Amplitude
f = 10;                        % Frequency (Hertz)
w = 2 * pi * f;               % Angular frequency (radians)
s = a * sin(w * t);           % Sine wave
plot(t, s);                   % Plot sine signal
```

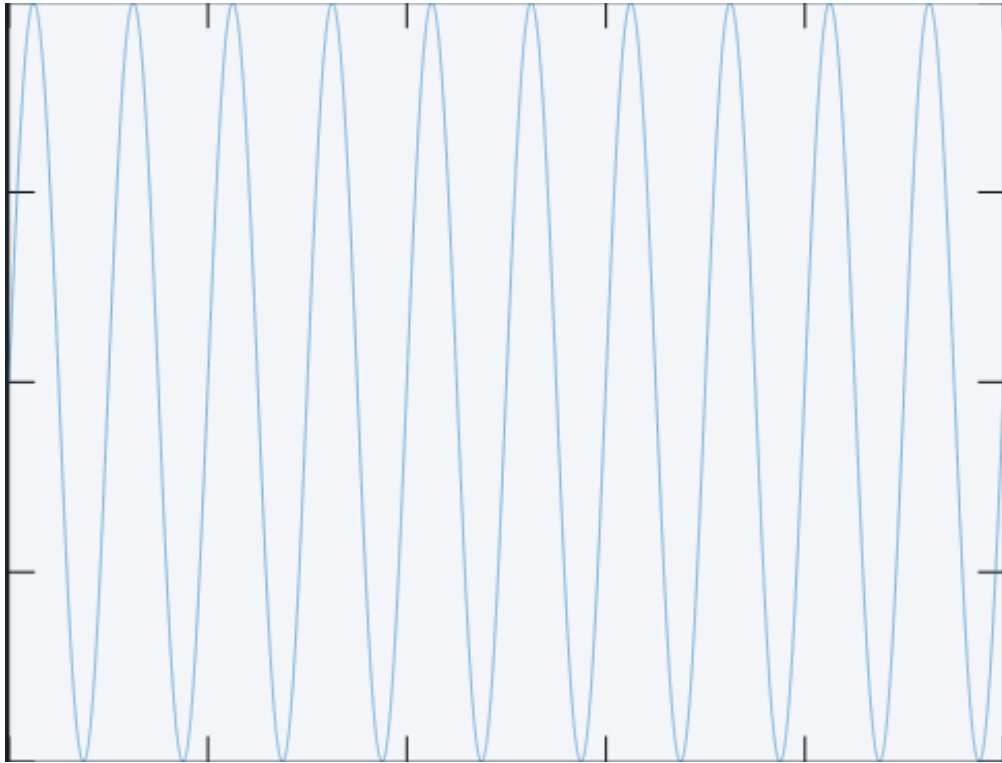


Figura 1. Señal sinusoidal de 10 Hertz

2. Generar la versión discreta de la misma señal

Para generar la versión discreta de esta señal se hizo uso de la función `stem(x, y)` que grafica una secuencia de datos discretizada, donde `x` y `y` deben de ser matrices o vectores del mismo tamaño o puede `x` ser una matriz y `y` un vector, pero el número de filas en la matriz debe ser de la misma longitud del vector.

Fueron utilizados los mismos componentes de la función anterior para realizar esta gráfica que es desplegada en la *Figura 2*. La forma refleja el mismo comportamiento, pero de forma discreta.

```
stem(t, s);
```

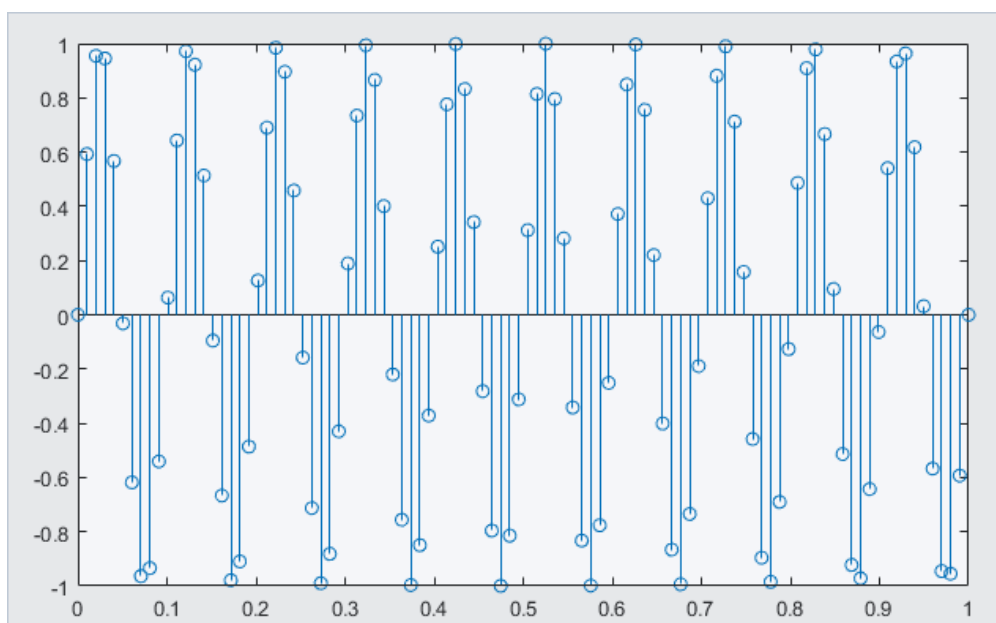


Figura 2. Versión discretizada de una señal sinusoidal de 10 Hertz

3. En una sola figura grafique ambas señales.

Esta faena requirió mantener las dos gráficas sobrepuestas en un solo lugar, con este objeto se utilizó la instrucción `hold on` que mantiene la gráfica dibujada anteriormente y dibuja encima de ella la gráfica consecuente. Finalmente, el código finalizó de la siguiente manera con la modificación de las últimas líneas entre las que `hold on`; figura. La gráfica en la *Figura 3* resultante expresa simplemente las dos anteriores sobrepuestas una sobre la otra.

```
t = linspace(0, 1, 10000); % Time
a = 1; % Amplitude
f = 10; % Frequency (Hertz)
w = 2 * pi * f; % Angular frequency (radians)
s = a * sin(w * t); % Sine wave
plot(t, s); % Plot sine signal
hold on;
stem(t, s);
```

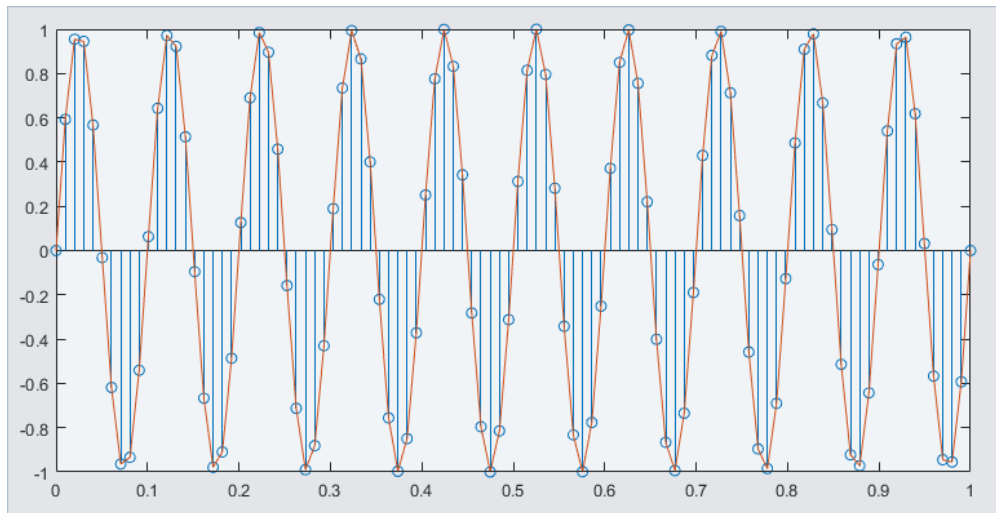


Figura 3. Gráficas sobrepuestas de las dos versiones anteriores

4. Genere el espectro de frecuencias de la señal analógica obtenida.

Para visualizar el espectro de frecuencias de esta señal fueron utilizadas las siguientes dos funciones:

- `fft(x)`: recibe un vector y devuelve el mismo vector con la transformada de Fourier aplicada
- `abs(x)`: recibe un vector y regresa el valor absoluto de cada elemento del vector

```
t = linspace(0, 1, 10000); % Time
a = 1; % Amplitude
f = 10; % Frequency (Hertz)
w = 2 * pi * f; % Angular frequency (radians)
s = a * sin(w * t); % Sine wave
plot(t, s); % Plot sine signal

esp = abs(fft(s)); % Espectrum of original signal
plot(esp)
```

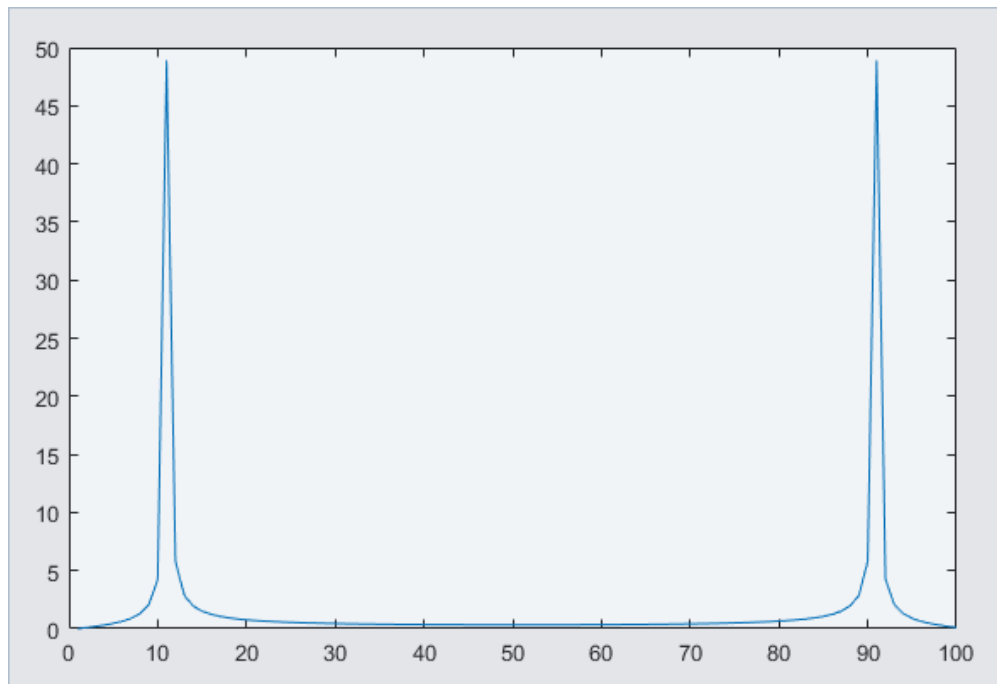


Figura 4. Espectro de frecuencias de la señal analógica

En la *Figura 4* se muestra el espectro de frecuencias obtenido de la señal. Como se puede notar, hay 2 picos en diferentes frecuencias, los cuales representan la frecuencia de los diferentes armónicos que la señal necesita para ser formada.

5. Agregue ruido a la señal analógica.

Con el fin de agregar ruido a una señal se generó una nueva señal analógica con valores aleatorios dentro de un rango arbitrario que serán posteriormente agregados a la señal original. Este ruido fue generado de la siguiente manera:

```
n = 0.5 * rand(length(t)); % Noise
```

Esta instrucción genera una matriz cuadrada de las mismas dimensiones que las muestras de tiempo previamente declaradas en la tarea 1 lleno de valores aleatorios que representarán su *amplitud* en cada intervalo de tiempo.

6. Obtenga nuevamente el espectro de la señal + ruido, comente los resultados.

Para lograr este cometido fue utilizada la señal generada en el punto número 1 de este reporte y el ruido creado en el punto anterior. Estas dos señales fueron sumadas para obtener una señal con cierto ruido generado. Posteriormente, para Esto fue conseguido con el siguiente código:

```
t = linspace(0, 1, 10000); % Time
a = 1; % Amplitude
f = 10; % Frequency (Hertz)
w = 2 * pi * f; % Angular frequency (radians)
s = a * sin(w * t); % Sine wave
```

```
n = 0.5 * rand(length(t)); % Noise
ns = s + n; % Noisy signal

esp = abs(fft(ns)); % Spectrum of noisy signal
plot(esp)
```

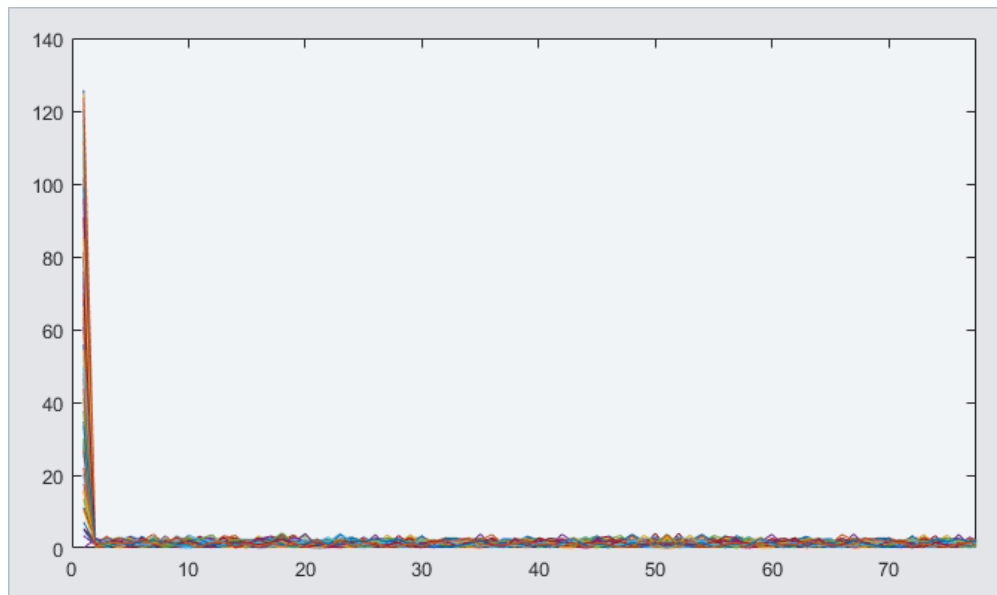


Figura 5. Espectro de la señal con ruido

Como se puede observar en la *Figura 5*, el espectro de frecuencias de la señal ahora muestra pequeños picos a lo largo de todo el espectro de frecuencias. Estos picos son generados debido a la suma que se realizó de la señal original con el ruido generado, por lo que la señal ahora presenta ruido o interferencia en diferentes frecuencias aleatorias.

Conclusión

La transformación inmediata de una ecuación a su equivalente visual en dos dimensiones permitió que las asociaciones neuronales que han sido desarrolladas paulatinamente en la teoría de estos cursos fuese afianzada con más solidez.

La elaboración de la presente práctica permitió tener un mejor entendimiento de las propiedades de las ondas, ya que permite realizar modelos matemáticos manipulables que ofrecen facilidades para el análisis de la estructura o comportamiento de la onda modelada.

La práctica también sirvió como introducción al ambiente matemático *Matlab* en el área de sistemas de telecomunicación, ya que con el uso conjunto de las funciones como `plot()`, `stem()` y `fft()`, y la aplicación de las propiedades matemáticas de las ondas, se pudo observar que puede ser una herramienta sólida para el análisis de ondas analógicas o cualquier modelo matemático.

Bibliografía

- Attaway, S. (2012). *Matlab : A Practical Introduction to Programming and Problem Solving (Vol. 2nd ed)*. Amsterdam: Butterworth-Heinemann.