

Análisis de la variable compleja y Análisis de Fourier

PRÁCTICA 1: Operaciones con señales

Helena García Escudero

8 de abril, 2019

0.1. Introduction

Vamos a comenzar explicando las nociones esenciales que permitan realizar las operaciones básicas que permitan, posteriormente, analizar señales.

0.2. Práctica 1a- Operaciones básicas

Esta primera parte tiene como objetivo aprender a escribir y a manipular señales en MATLAB. Empezaremos definiendo una función, que llamaremos *sinusoide.m*, la función recibirá como parámetros de entrada los extremos del intervalo $[a,b]$, en el que estará definida, la amplitud A , el pulso w y la fase inicial ϕ :

El código que contiene dicho script es el siguiente :

Se define una función que crea senos:

```
function [t,x]=sinusoide(a,b,A,w,phi);
t=[a:.05:b];
x=A*sin(t*w+phi);
stem(t,x) (pinta los valores de la función en los puntos considerados)
```

pause (detiene momentáneamente la ejecución hasta que se pulse cualquier tecla)

figure (permite pintar una nueva figura sin borrar la anterior)

plot(t,x) (pinta los valores de la función en todo el rango interpolando con senos)

pause

A continuación dibujamos en una ventana doble ambas representaciones

figure

```
subplot(2,1,1)
stem(t,x)
title('stem(x)')
subplot(2,1,2)
plot(t,x)
title('plot(x)')
```

```
pause
```

```
close all (Para borrar todas las figuras que se hayan creado)
```

Ejemplo de utilización: `sinusoide(0,10,4,7,0);`

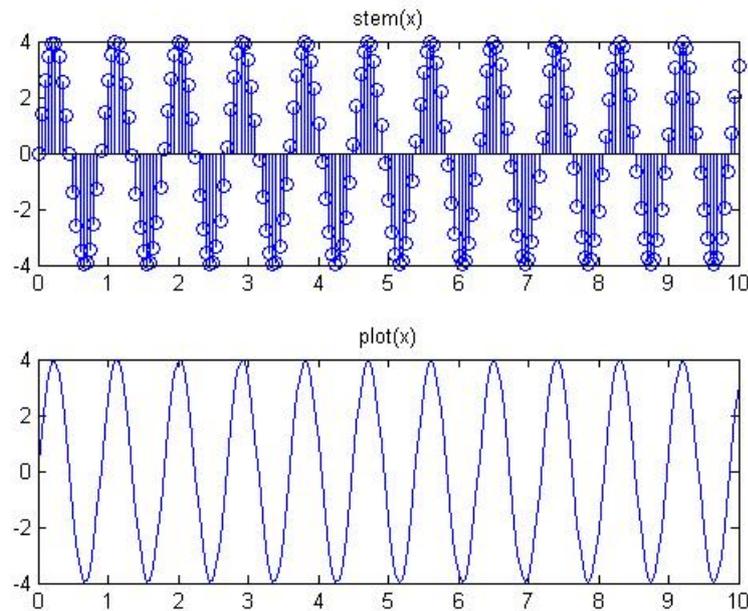


Figura 1: Sinusoide

0.2.1. Operaciones con señales

A continuación, considerando una señal sinusoidal, la escalaremos, desplazaremos e invertiremos. También fijaremos nuestra atención en el orden de ejecución de dichas operaciones, no se obtiene el mismo resultado si se escala y luego se desplaza, a si se desplaza primero y se escala después.

- Escalado

Para escalar una señal, ¿cómo deberíamos modificar el código para obtener una señal de este tipo?

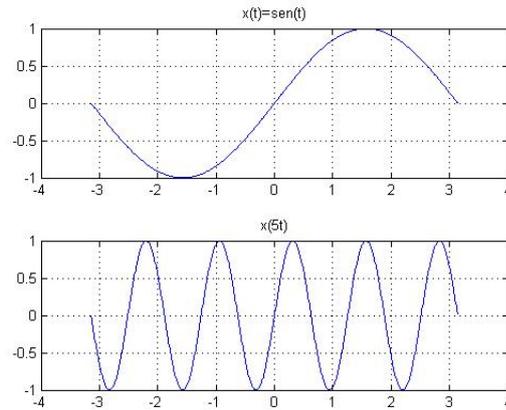


Figura 2: Escalado

- Desplazamiento o retardo

Para retrasar una señal, ¿cómo deberíamos modificar el código para obtener una señal de este tipo?

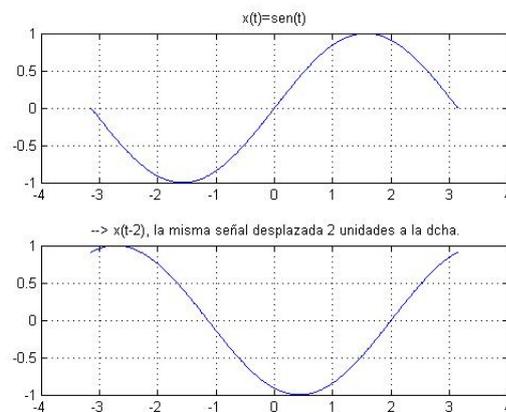


Figura 3: Desplazamiento

- Inversión

Para invertir una señal, ¿cómo deberíamos modificar el código para obtener una señal de este tipo?

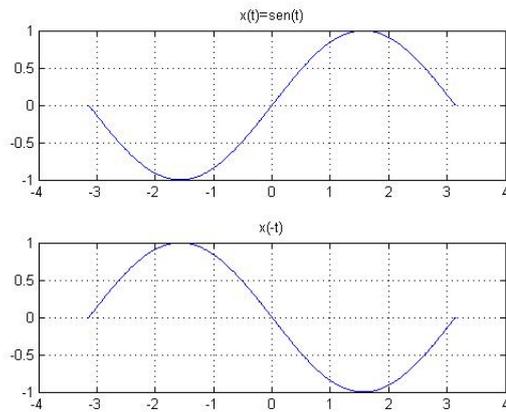


Figura 4: Inversion

Una vez manejadas estas tres operaciones básicas, podemos combinarlas para manipular las señales de la forma que deseemos. Pero es necesario señalar la importancia que tiene el orden en el que realizamos las operaciones, no es lo mismo aplicar primero un retardo a la señal para, posteriormente escalarla, que ejecutar las operaciones de forma inversa, escalado previo al retardo de la señal. Observemos gráficamente este proceso:

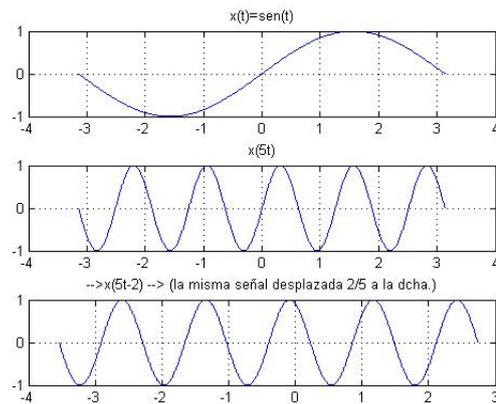


Figura 5: Escalado y retardo

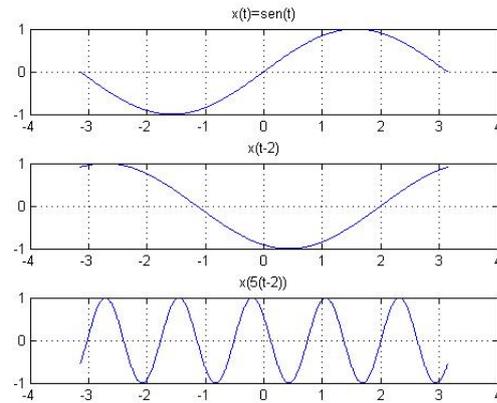


Figura 6: Señal desplazada y posteriormente escalado

Convolución

La operación más importante en la teoría de señales es, probablemente, el producto de convolución. Este producto se calcula combinando las anteriores operaciones explicadas. La fórmula que define el producto de convolución de dos señales discretas, $x[n]$ e $y[n]$ es:

$$z[n] = \sum_{i=-\infty}^{\infty} x[k]y[n - k]$$

El proceso llevado a cabo para calcular $z[n]$ es: en primer lugar invertir la señal y , y desplazarla n unidades a la derecha y multiplicarla escalarmente con la x .

En las siguientes imágenes se puede estudiar el paso a paso de como se va desarrollando el proceso de un modo gráfico:

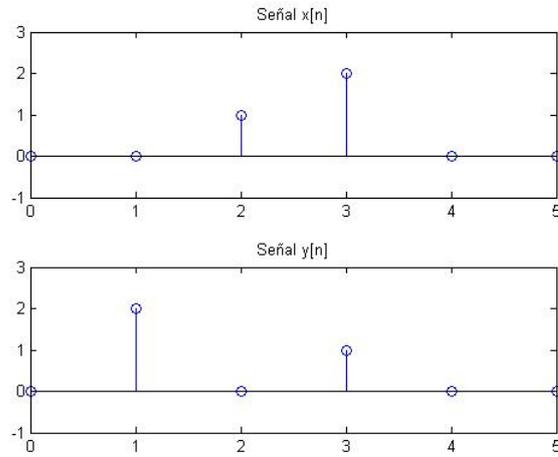


Figura 7: Señales iniciales

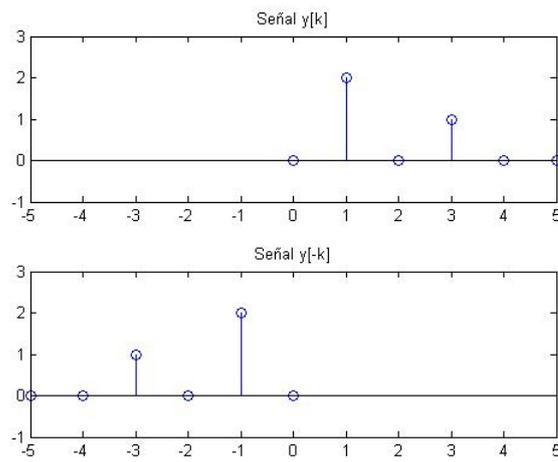


Figura 8: Inversion señal y

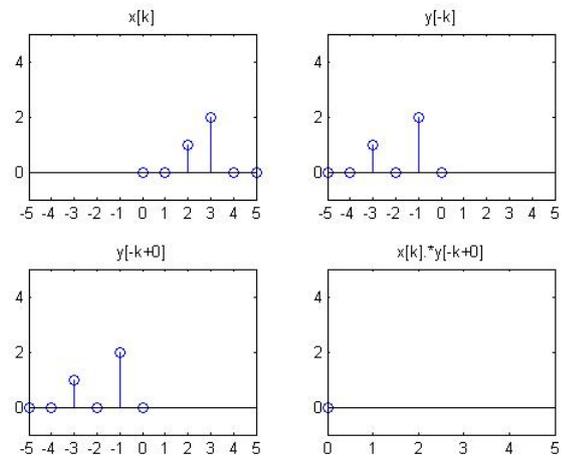


Figura 9: Inicio sumatorio

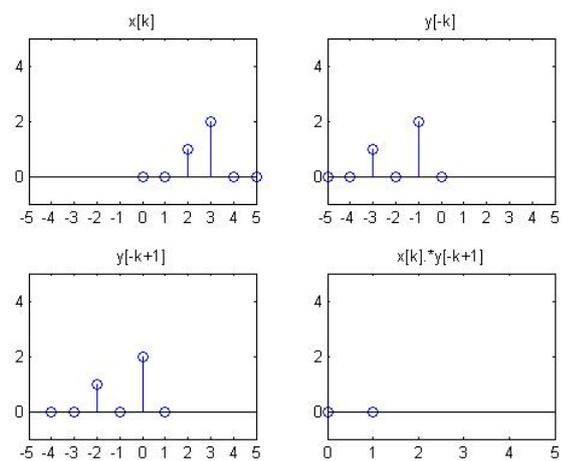


Figura 10: Segundo paso sumatorio

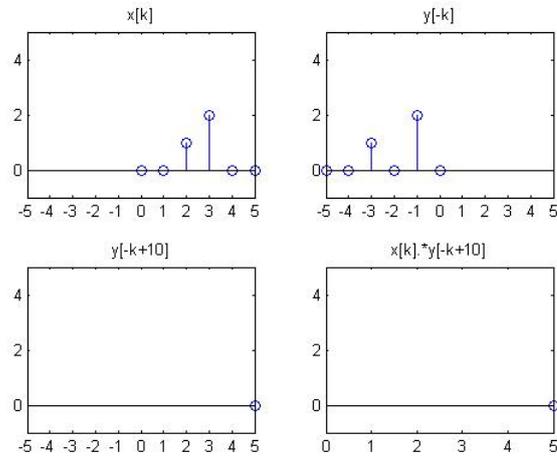


Figura 11: Decimo paso sumatorio

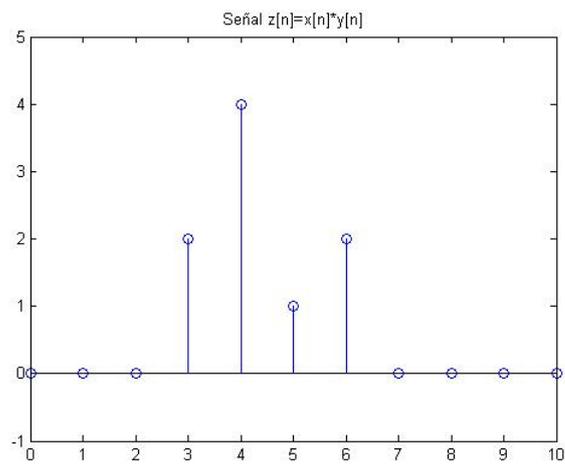


Figura 12: Final sumatorio

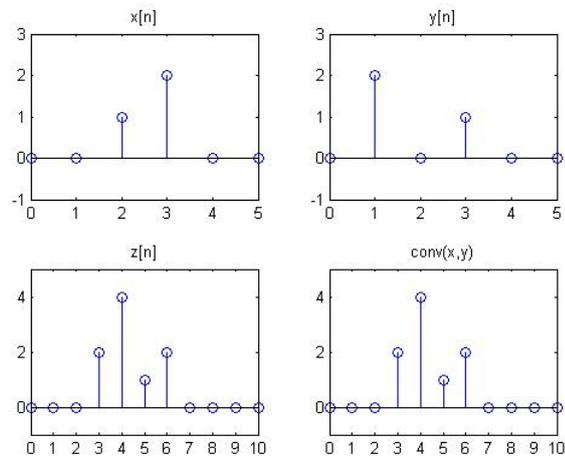


Figura 13: Resultado Convolución

El código ha sido creado manualmente y paso a paso en Matlab, pero este programa tiene un comando propio, llamado *conv*, que hace convoluciones de señales de forma automática.

Bibliografía

DISCRETE-TIME SIGNAL PROCESSING (2ND ED.) *ALAN V. OPPENHEIM ,RONALD W. SCHAFER*