

## Práctica 1

### Introducción a Matlab

Matlab ([www.mathworks.com](http://www.mathworks.com)) es un programa interactivo con capacidad para cálculos numéricos y visualización de datos. Es ampliamente utilizado en el análisis y diseño de sistemas de control. Existen además una serie de “Toolboxes” los cuales extienden las funciones básicas de Matlab a varias áreas de aplicación. En este curso se estudiarán algunas de las funciones de las herramientas disponibles para control: *Control Systems Toolbox*. Matlab se encuentra disponible para una gran variedad de plataformas, incluyendo Unix y Linux, Macintosh y Windows.

Al iniciar Matlab, aparecerá una ventana de nombre *ventana de comandos*. Es aquí donde deberemos introducir los comandos que se describen en esta práctica, los cuales serán interpretados por Matlab. Finalmente, los resultados de las operaciones solicitadas serán desplegados de forma interactiva en la misma ventana.

Comencemos creando algo simple como un vector. Sus elementos se escriben entre corchetes, separados por comas o espacios, y su valor se asigna a una variable. Por ejemplo:

```
a = [1 2 3 4 5 6 9 8 7]
```

Matlab debe responder con

```
a =  
    1    2    3    4    5    6    9    8    7
```

## 1 Funciones, vectores y matrices

Supongamos que se desea crear un vector con los elementos del 0 al 20, espaciados uniformemente en incrementos de 2.

```
t = 0:2:20
```

```
t =  
    0    2    4    6    8   10   12   14   16   18   20
```

Este método es frecuentemente utilizado, por ejemplo, para crear un vector que represente a una variable independiente como el tiempo.

La manipulación de vectores es tan sencilla como su creación. Por ejemplo, supóngase que se desea sumar dos a cada uno de los elementos del vector a y asignar el resultado a b. Se procede de la siguiente manera:

```
b = a + 2
```

```
b =  
    3    4    5    6    7    8   11   10    9
```

Ahora, supóngase que se desea sumar (o restar) dos vectores. Si ambos son de la misma longitud, simplemente se pueden sumar como se muestra a continuación.

```
c = a + b
```

```
c =  
    4    6    8   10   12   14   20   18   16
```

Matlab también incluye un gran número de funciones estándar, tales como seno (`sin`), coseno (`cos`), logaritmo natural (`log`), exponencial (`exp`), raíz cuadrada (`sqrt`), etc, las cuales pueden operar sobre los elementos de un vector por medio de comandos simples. También incluye constantes predefinidas de uso frecuente, tales como  $\pi$ , y también  $i$  o  $j$  para la unidad imaginaria. Por ejemplo podemos escribir

```
sin(pi/4)
```

```
ans =
```

```
0.7071
```

Es posible determinar la forma de utilizar cualquier función escribiendo `help` [nombre de la función] en la línea de comandos de Matlab.

De igual forma, la graficación de funciones en Matlab es sencilla. Por ejemplo, supóngase que se desea graficar una onda senoidal como función del tiempo. Se comienza entonces, por crear un vector de tiempo (nótese el uso del signo punto y coma al final del comando, el cual indica a Matlab que no deseamos ver en pantalla el resultado de la operación)

```
t=0:0.25:7;  
y = sin(t);  
plot(t,y)
```

Esta gráfica contiene aproximadamente un periodo de la función seno. Adicionalmente, la instrucción `plot` incluye varias funciones que nos permiten etiquetar y manipular las gráficas. Se puede aprender más acerca de esto, escribiendo `help plot` en la línea de comandos.

La definición de matrices en Matlab es similar a la de vectores, con la excepción de que los renglones se separan por punto y coma (;). Por ejemplo

```
B = [1 2 3 4; 5 6 7 8; 9 10 11 12]
```

```
B =  
    1     2     3     4  
    5     6     7     8  
    9    10    11    12
```

Así mismo, la aritmética de matrices y vectores es muy sencilla. Por ejemplo, se puede definir un vector columna

```
x = transpose([1 2 3 4]);
```

La función `transpose` se usa para transponer una matriz o vector. Posteriormente es posible realizar la multiplicación con un simple comando como

```
B*x
```

Matlab también nos permite una enorme flexibilidad en las operaciones entre vectores y matrices. Por ejemplo, matemáticamente no es válido multiplicar un vector por otro de la misma dimensión, de forma que si escribimos

```
x*x
```

obtendremos un mensaje de error. Sin embargo, es posible realizar la multiplicación de un vector renglón por un vector columna de la misma dimensión, como por ejemplo

```
x*transpose(x)
transpose(x)*x
```

Incluso es posible indicar a Matlab que la operación se realice elemento a elemento, por lo que por ejemplo

```
a.*a
```

sí es posible (el punto antes de algún operador indica que se aplica por elementos y no como operador de vector o matriz).

## 2 Transformada inversa de Laplace

Además de la capacidad numérica, Matlab permite la manipulación de variables algebraicas, denominadas simbólicas. Para declarar un conjunto de variables como simbólicas, se utiliza la función `syms`. Por ejemplo,

```
syms s t
```

declara a las variables `s` y `t` como símbolos.

Seguidamente podemos utilizar la función `ilaplace`, la cual calcula la transformada inversa de Laplace de una función. Por ejemplo,

```
F=1/(s^2+1);
f=ilaplace(f)
```

Matlab responde con

```
f =
    sin(t)
```

Posteriormente es posible definir un rango para `t`

```
t=0:0.1:7;
```

y evaluar y graficar la función

```
plot(t,eval(f))
```

### 3 Cuestionario

1. Supóngase que A es una matriz de 3x3, b y c vectores columna de longitud 3, y d un vector columna de longitud 4. Determinar si las siguientes operaciones son posibles o no, y explicar brevemente la razón.

- (a)  $A*b$
- (b)  $\text{transpose}(b)*A$
- (c)  $A*A$
- (d)  $b*b$
- (e)  $A*d$
- (f)  $a+d$
- (g)  $a^3$
- (h)  $a.^3$
- (i)  $A^3$
- (j)  $A.^3$

2. Supóngase que se sabe que la función de transferencia de un sistema está dada por

$$F(s) = \frac{1}{s^2 + 5s + 4}$$

¿Cómo se graficaría su respuesta al impulso en Matlab?