

Introducción a Matlab

José Luis Caroca Gaete

¹Facultad de Ciencias Físicas y Matemáticas

²Departamento de Ingeniería Civil

C142G- Dinámica de Estructuras

Agosto de 2009

Contenido

- 1 **¿Qué es Matlab?**
- 2 El Entorno Matlab
- 3 El lenguaje Matlab
- 4 Funciones
- 5 Gráficos
- 6 Problema propuesto

Introducción

- 1 MATLAB (*MATrix LABoratory*): Software matemático que permite realizar cálculos numéricos a través de vectores y matrices.
- 2 El ambiente MATLAB se potencia a través de su plataforma de simulación mutidominio, *Simulink*, y su conjunto de librerías especializadas o *Toolboxes*.
- 3 GNU OCTAVE es, al igual que Matlab, un lenguaje de alto nivel diseñado para realizar cálculos numéricos, pero de libre distribución (licencia pública general de GNU). Su sintáxis es similar a la de Matlab.

Introducción

- 1 Recientemente NETBEANS (plataforma para el desarrollo de aplicaciones de escritorio) puso a disposición el plugin OCTAVENB, el cual permite utilizar el entorno NETBEANS en el desarrollo de m-files y funciones propias de GNU OCTAVE.
- 2 Información en:
 - <http://www.gnu.org/software/octave/>
 - <https://octavenb.dev.java.net/>

Contenido

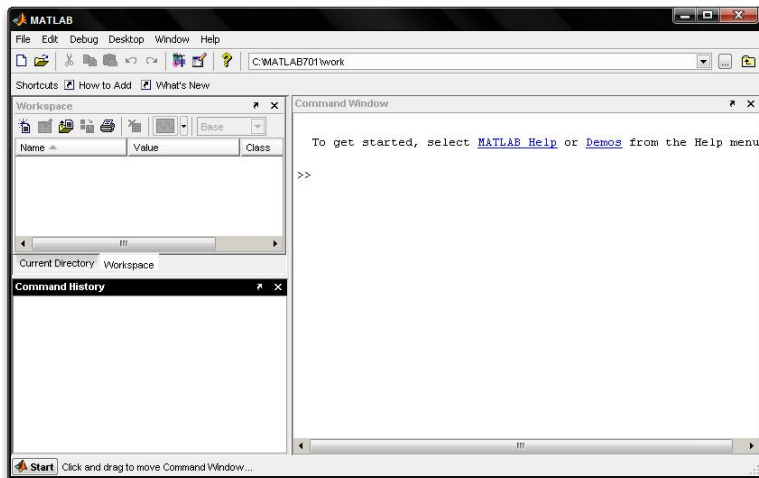
- 1 ¿Qué es Matlab?
- 2 El Entorno Matlab**
- 3 El lenguaje Matlab
- 4 Funciones
- 5 Gráficos
- 6 Problema propuesto

Una primera mirada a Matlab

El entorno MATLAB está organizado mediante ventanas:

- **Command Window:** Es la ventana de comandos para la ejecución de instrucciones.
- **Command History:** Ventana que contiene el registro de los comandos que han sido ingresados.
- **Workspace:** Explorador de las variables utilizadas en la sesión.

Entorno Matlab



Contenido

- 1 ¿Qué es Matlab?
- 2 El Entorno Matlab
- 3 El lenguaje Matlab**
- 4 Funciones
- 5 Gráficos
- 6 Problema propuesto

Operaciones básicas

- 1 Para ingresar comentarios anteponemos el símbolo % a lo que se quiere comentar.

- 2 Variables:

```
% Asignación del valor 7 a la variable x.  
x=7;
```

- 3 Estructuras:

```
% La siguiente estructura está compuesta  
% de los campos 'estación' y 'maximo'.  
registro.estacion='ILOCA EW';  
registro.maximo=max(abs(ag));
```

Operaciones básicas

- 1 Para ingresar comentarios anteponemos el símbolo % a lo que se quiere comentar.

- 2 Variables:

```
% Asignación del valor 7 a la variable x.  
x=7;
```

- 3 Estructuras:

```
% La siguiente estructura está compuesta  
% de los campos 'estación' y 'maximo'.  
registro.estacion='ILOCA EW';  
registro.maximo=max(abs(ag));
```

Operaciones básicas

- 1 Para ingresar comentarios anteponemos el símbolo % a lo que se quiere comentar.

- 2 Variables:

```
% Asignación del valor 7 a la variable x.  
x=7;
```

- 3 Estructuras:

```
% La siguiente estructura está compuesta  
% de los campos 'estación' y 'maximo'.  
registro.estacion='ILOCA EW';  
registro.maximo=max(abs(ag));
```

Operaciones básicas

1 Vectores:

```
% Vector fila.  
vec=[1,2,3];  
vec=[1 2 3];  
% Vector columna.  
vec=[1; 2; 3];  
% Secuencias (<inicio>:<paso>:<fin>).  
vec=0:1:10;
```

2 Matrices

```
% Matriz de 3 filas y 2 columnas.  
M=[1, 2; 3, 4; 5, 6];
```

Indexación de elementos

- 1 $A(i, j)$: Obtiene el elemento de la fila i y columna j de la matriz A .
- 2 $A(i, :)$: Muestra la fila i -ésima de A .
- 3 $A(:, j)$: Muestra la columna j -ésima de A .
- 4 $A(Ind1, Ind2)$: Muestra una submatriz de A , compuesta por las filas y columnas indicadas por los elementos de los vectores $Ind1$ y $Ind2$.

Indexación de elementos

- 1 $A(i, j)$: Obtiene el elemento de la fila i y columna j de la matriz A .
- 2 $A(i:,)$: Muestra la fila i -ésima de A .
- 3 $A(:, j)$: Muestra la columna j -ésima de A .
- 4 $A(Ind1, Ind2)$: Muestra una submatriz de A , compuesta por las filas y columnas indicadas por los elementos de los vectores $Ind1$ y $Ind2$.

Indexación de elementos

- 1 $A(i, j)$: Obtiene el elemento de la fila i y columna j de la matriz A .
- 2 $A(i:,)$: Muestra la fila i -ésima de A .
- 3 $A(:, j)$: Muestra la columna j -ésima de A .
- 4 $A(Ind1, Ind2)$: Muestra una submatriz de A , compuesta por las filas y columnas indicadas por los elementos de los vectores $Ind1$ y $Ind2$.

Indexación de elementos

- 1 $A(i, j)$: Obtiene el elemento de la fila i y columna j de la matriz A .
- 2 $A(i:,)$: Muestra la fila i -ésima de A .
- 3 $A(:, j)$: Muestra la columna j -ésima de A .
- 4 $A(Ind1, Ind2)$: Muestra una submatriz de A , compuesta por las filas y columnas indicadas por los elementos de los vectores $Ind1$ y $Ind2$.

Matrices predefinidas

- 1 **A' : Matriz traspuesta de A .**
- 2 $\text{inv}(A)$: Matriz inversa de A .
- 3 $\text{eye}(n)$: Matriz identidad de dimensiones $n \times n$.
- 4 $\text{diag}(\text{Ind})$: Si Ind es un vector, $\text{diag}(\text{Ind})$ genera una matriz cuyos elementos de la diagonal coinciden con los elementos del vector. Si Ind es una matriz, $\text{diag}(\text{Ind})$ obtiene la diagonal de la matriz.
- 5 $\text{rand}(m, n)$: Matriz de componentes aleatorias de dimensiones $m \times n$.
- 6 $\text{zeros}(m, n)$: Matriz de ceros de dimensiones $m \times n$.
- 7 $\text{ones}(m, n)$: Matriz de unos de dimensiones $m \times n$.

Matrices predefinidas

- 1 **A' : Matriz traspuesta de A .**
- 2 **$\text{inv}(A)$: Matriz inversa de A .**
- 3 $\text{eye}(n)$: Matriz identidad de dimensiones $n \times n$.
- 4 $\text{diag}(\text{Ind})$: Si Ind es un vector, $\text{diag}(\text{Ind})$ genera una matriz cuyos elementos de la diagonal coinciden con los elementos del vector. Si Ind es una matriz, $\text{diag}(\text{Ind})$ obtiene la diagonal de la matriz.
- 5 $\text{rand}(m, n)$: Matriz de componentes aleatorias de dimensiones $m \times n$.
- 6 $\text{zeros}(m, n)$: Matriz de ceros de dimensiones $m \times n$.
- 7 $\text{ones}(m, n)$: Matriz de unos de dimensiones $m \times n$.

Matrices predefinidas

- 1 **A' : Matriz traspuesta de A .**
- 2 **$\text{inv}(A)$: Matriz inversa de A .**
- 3 **$\text{eye}(n)$: Matriz identidad de dimensiones $n \times n$.**
- 4 $\text{diag}(\text{Ind})$: Si Ind es un vector, $\text{diag}(\text{Ind})$ genera una matriz cuyos elementos de la diagonal coinciden con los elementos del vector. Si Ind es una matriz, $\text{diag}(\text{Ind})$ obtiene la diagonal de la matriz.
- 5 $\text{rand}(m, n)$: Matriz de componentes aleatorias de dimensiones $m \times n$.
- 6 $\text{zeros}(m, n)$: Matriz de ceros de dimensiones $m \times n$.
- 7 $\text{ones}(m, n)$: Matriz de unos de dimensiones $m \times n$.

Matrices predefinidas

- 1 A' : Matriz traspuesta de A .
- 2 $\text{inv}(A)$: Matriz inversa de A .
- 3 $\text{eye}(n)$: Matriz identidad de dimensiones $n \times n$.
- 4 $\text{diag}(\text{Ind})$: Si Ind es un vector, $\text{diag}(\text{Ind})$ genera una matriz cuyos elementos de la diagonal coinciden con los elementos del vector. Si Ind es una matriz, $\text{diag}(\text{Ind})$ obtiene la diagonal de la matriz.
- 5 $\text{rand}(m, n)$: Matriz de componentes aleatorias de dimensiones $m \times n$.
- 6 $\text{zeros}(m, n)$: Matriz de ceros de dimensiones $m \times n$.
- 7 $\text{ones}(m, n)$: Matriz de unos de dimensiones $m \times n$.

Matrices predefinidas

- 1 A' : Matriz traspuesta de A .
- 2 $\text{inv}(A)$: Matriz inversa de A .
- 3 $\text{eye}(n)$: Matriz identidad de dimensiones $n \times n$.
- 4 $\text{diag}(\text{Ind})$: Si Ind es un vector, $\text{diag}(\text{Ind})$ genera una matriz cuyos elementos de la diagonal coinciden con los elementos del vector. Si Ind es una matriz, $\text{diag}(\text{Ind})$ obtiene la diagonal de la matriz.
- 5 $\text{rand}(m, n)$: Matriz de componentes aleatorias de dimensiones $m \times n$.
- 6 $\text{zeros}(m, n)$: Matriz de ceros de dimensiones $m \times n$.
- 7 $\text{ones}(m, n)$: Matriz de unos de dimensiones $m \times n$.

Matrices predefinidas

- 1 A' : Matriz traspuesta de A .
- 2 $\text{inv}(A)$: Matriz inversa de A .
- 3 $\text{eye}(n)$: Matriz identidad de dimensiones $n \times n$.
- 4 $\text{diag}(\text{Ind})$: Si Ind es un vector, $\text{diag}(\text{Ind})$ genera una matriz cuyos elementos de la diagonal coinciden con los elementos del vector. Si Ind es una matriz, $\text{diag}(\text{Ind})$ obtiene la diagonal de la matriz.
- 5 $\text{rand}(m, n)$: Matriz de componentes aleatorias de dimensiones $m \times n$.
- 6 $\text{zeros}(m, n)$: Matriz de ceros de dimensiones $m \times n$.
- 7 $\text{ones}(m, n)$: Matriz de unos de dimensiones $m \times n$.

Matrices predefinidas

- 1 A' : Matriz traspuesta de A .
- 2 $\text{inv}(A)$: Matriz inversa de A .
- 3 $\text{eye}(n)$: Matriz identidad de dimensiones $n \times n$.
- 4 $\text{diag}(\text{Ind})$: Si Ind es un vector, $\text{diag}(\text{Ind})$ genera una matriz cuyos elementos de la diagonal coinciden con los elementos del vector. Si Ind es una matriz, $\text{diag}(\text{Ind})$ obtiene la diagonal de la matriz.
- 5 $\text{rand}(m, n)$: Matriz de componentes aleatorias de dimensiones $m \times n$.
- 6 $\text{zeros}(m, n)$: Matriz de ceros de dimensiones $m \times n$.
- 7 $\text{ones}(m, n)$: Matriz de unos de dimensiones $m \times n$.

Operaciones matriciales

- 1 $A \pm B$: Suma/resta entre A y B .
- 2 $A * B$: Multiplicación matricial de A y B .
- 3 $A .* B$: Multiplicación elemento a elemento de A y B .
- 4 $A ./ B$: División entre los elementos de A y B .
- 5 $\text{cross}(v1, v2)$: Producto cruz entre los vectores $v1$ y $v2$.
- 6 $\text{dot}(v1, v2)$: Producto punto entre los vectores $v1$ y $v2$.

Operaciones matriciales

- 1 $A \pm B$: Suma/resta entre A y B .
- 2 $A * B$: Multiplicación matricial de A y B .
- 3 $A .* B$: Multiplicación elemento a elemento de A y B .
- 4 $A ./ B$: División entre los elementos de A y B .
- 5 $\text{cross}(v1, v2)$: Producto cruz entre los vectores $v1$ y $v2$.
- 6 $\text{dot}(v1, v2)$: Producto punto entre los vectores $v1$ y $v2$.

Operaciones matriciales

- 1 $A \pm B$: Suma/resta entre A y B .
- 2 $A * B$: Multiplicación matricial de A y B .
- 3 $A . * B$: Multiplicación elemento a elemento de A y B .
- 4 $A ./ B$: División entre los elementos de A y B .
- 5 $\text{cross}(v1, v2)$: Producto cruz entre los vectores $v1$ y $v2$.
- 6 $\text{dot}(v1, v2)$: Producto punto entre los vectores $v1$ y $v2$.

Operaciones matriciales

- 1 $A \pm B$: Suma/resta entre A y B .
- 2 $A * B$: Multiplicación matricial de A y B .
- 3 $A . * B$: Multiplicación elemento a elemento de A y B .
- 4 $A ./ B$: División entre los elementos de A y B .
- 5 `cross(v1, v2)`: Producto cruz entre los vectores $v1$ y $v2$.
- 6 `dot(v1, v2)`: Producto punto entre los vectores $v1$ y $v2$.

Operaciones matriciales

- 1 $A \pm B$: Suma/resta entre A y B .
- 2 $A * B$: Multiplicación matricial de A y B .
- 3 $A .* B$: Multiplicación elemento a elemento de A y B .
- 4 $A ./ B$: División entre los elementos de A y B .
- 5 `cross(v1, v2)`: Producto cruz entre los vectores $v1$ y $v2$.
- 6 `dot(v1, v2)`: Producto punto entre los vectores $v1$ y $v2$.

Operaciones matriciales

- 1 $A \pm B$: Suma/resta entre A y B .
- 2 $A * B$: Multiplicación matricial de A y B .
- 3 $A . * B$: Multiplicación elemento a elemento de A y B .
- 4 $A ./ B$: División entre los elementos de A y B .
- 5 `cross(v1, v2)`: Producto cruz entre los vectores $v1$ y $v2$.
- 6 `dot(v1, v2)`: Producto punto entre los vectores $v1$ y $v2$.

Loops y condicionales a través de un ejemplo

Generación del vector:

```
v=[1, 0, 9, 0, 25, 0, 49, 0, 81, 0]
```

- Operador FOR:

```
v=zeros(1,10);  
for i=1:10  
    if rem(i,2)~=0 % i es impar  
        v(i)=i^2;  
    end  
end
```

Loops y condicionales a través de un ejemplo

Generación del vector:

```
v=[1, 0, 9, 0, 25, 0, 49, 0, 81, 0]
```

- Operador WHILE:

```
v=zeros(1,10);  
i=1; % contador  
while (i<=10)  
    if rem(i,2)~=0 % i es impar  
        v(i)=i^2;  
    end  
    i=i+1;  
end
```

Contenido

- 1 ¿Qué es Matlab?
- 2 El Entorno Matlab
- 3 El lenguaje Matlab
- 4 Funciones**
- 5 Gráficos
- 6 Problema propuesto

Declaración de Funciones

- 1 En Matlab, el nombre de la función y el nombre del archivo `.m` en el que se almacena esta función deben coincidir.
- 2 Todas las funciones que utilicen deben incluir (por medio comentarios):
 - Descripción u objetivo de la función.
 - Detalle de los parámetros de entrada.
 - Detalle de los parámetros de salida.
 - Su nombre y fecha.
- 3 Los comentarios definidos en el paso anterior constituyen el `help` de sus funciones.

Ejemplo

```
[pares, impares]=espar(lista)
% [pares, impares]=espar(lista)
% Función que separa los elementos pares e impares de un vector de
% entrada arbitrario.
% Parámetros de entrada:
% - lista: Vector de números reales.
% Parámetros de salida:
% - pares : Elementos par de lista.
% - impares: Elementos impar de lista.
%
% José Luis Caroca Gaete
% 02 de Agosto de 2009

n=length(lista);
pares=[];
impares=[];
for i=1:n
    if (lista(i)-floor(lista(i)))==0 % Sólo números enteros
        if rem(lista(i),2)==0 % Par
            pares=[pares, lista(i)];
        else % Impar
            impares=[impares, lista(i)];
        end
    end
end
end
```

Contenido

- 1 ¿Qué es Matlab?
- 2 El Entorno Matlab
- 3 El lenguaje Matlab
- 4 Funciones
- 5 Gráficos**
- 6 Problema propuesto

Comandos básicos

- 1 **figure:** Crea una nueva ventana gráfica.
- 2 `plot(X,Y,'opciones')`: Comando que realiza el gráfico Y vs X.
- 3 `subplot(m,n,i)`: Permite realizar $m \times n$ gráficos en una misma ventana (i es el gráfico actual).
- 4 `xlabel('eje')`: Asigna el rótulo 'eje' al eje X.
- 5 `ylabel('eje')`: Asigna el rótulo 'eje' al eje Y.
- 6 `title('titulo')`: Define el título del gráfico.
- 7 `legend('n1','n2',...)`: Rótulos de las curvas presentes en el gráfico actual.
- 8 `grid on/off`: Hace visible/invisible la grilla del gráfico.

Comandos básicos

- 1 **figure:** Crea una nueva ventana gráfica.
- 2 **plot(X,Y,'opciones'):** Comando que realiza el gráfico Y vs X.
- 3 **subplot(m,n,i):** Permite realizar $m \times n$ gráficos en una misma ventana (*i* es el gráfico actual).
- 4 **xlabel('eje'):** Asigna el rótulo 'eje' al eje X.
- 5 **ylabel('eje'):** Asigna el rótulo 'eje' al eje Y.
- 6 **title('titulo'):** Define el título del gráfico.
- 7 **legend('n1','n2',...):** Rótulos de las curvas presentes en el gráfico actual.
- 8 **grid on/off:** Hace visible/invisible la grilla del gráfico.

Comandos básicos

- 1 **figure**: Crea una nueva ventana gráfica.
- 2 **plot**(X,Y,'opciones'): Comando que realiza el gráfico Y vs X.
- 3 **subplot**(m,n,i): Permite realizar $m \times n$ gráficos en una misma ventana (i es el gráfico actual).
- 4 **xlabel**('eje'): Asigna el rótulo 'eje' al eje X.
- 5 **ylabel**('eje'): Asigna el rótulo 'eje' al eje Y.
- 6 **title**('titulo'): Define el titulo del gráfico.
- 7 **legend**('n1','n2',...): Rótulos de las curvas presentes en el gráfico actual.
- 8 **grid on/off**: Hace visible/invisible la grilla del gráfico.

Comandos básicos

- 1 **figure:** Crea una nueva ventana gráfica.
- 2 **plot(X,Y,'opciones'):** Comando que realiza el gráfico Y vs X.
- 3 **subplot(m,n,i):** Permite realizar $m \times n$ gráficos en una misma ventana (*i* es el gráfico actual).
- 4 **xlabel('eje'):** Asigna el rótulo 'eje' al eje X.
- 5 **ylabel('eje'):** Asigna el rótulo 'eje' al eje Y.
- 6 **title('titulo'):** Define el título del gráfico.
- 7 **legend('n1','n2',...):** Rótulos de las curvas presentes en el gráfico actual.
- 8 **grid on/off:** Hace visible/invisible la grilla del gráfico.

Comandos básicos

- 1 `figure`: Crea una nueva ventana gráfica.
- 2 `plot(X,Y,'opciones')`: Comando que realiza el gráfico Y vs X.
- 3 `subplot(m,n,i)`: Permite realizar $m \times n$ gráficos en una misma ventana (i es el gráfico actual).
- 4 `xlabel('eje')`: Asigna el rótulo 'eje' al eje X.
- 5 `ylabel('eje')`: Asigna el rótulo 'eje' al eje Y.
- 6 `title('titulo')`: Define el título del gráfico.
- 7 `legend('n1','n2',...)`: Rótulos de las curvas presentes en el gráfico actual.
- 8 `grid on/off`: Hace visible/invisible la grilla del gráfico.

Comandos básicos

- 1 `figure`: Crea una nueva ventana gráfica.
- 2 `plot(X,Y,'opciones')`: Comando que realiza el gráfico Y vs X.
- 3 `subplot(m,n,i)`: Permite realizar $m \times n$ gráficos en una misma ventana (i es el gráfico actual).
- 4 `xlabel('eje')`: Asigna el rótulo 'eje' al eje X.
- 5 `ylabel('eje')`: Asigna el rótulo 'eje' al eje Y.
- 6 `title('titulo')`: Define el título del gráfico.
- 7 `legend('n1','n2',...)`: Rótulos de las curvas presentes en el gráfico actual.
- 8 `grid on/off`: Hace visible/invisible la grilla del gráfico.

Comandos básicos

- 1 `figure`: Crea una nueva ventana gráfica.
- 2 `plot(X,Y,'opciones')`: Comando que realiza el gráfico Y vs X.
- 3 `subplot(m,n,i)`: Permite realizar $m \times n$ gráficos en una misma ventana (i es el gráfico actual).
- 4 `xlabel('eje')`: Asigna el rótulo 'eje' al eje X.
- 5 `ylabel('eje')`: Asigna el rótulo 'eje' al eje Y.
- 6 `title('titulo')`: Define el título del gráfico.
- 7 `legend('n1','n2',...)`: Rótulos de las curvas presentes en el gráfico actual.
- 8 `grid on/off`: Hace visible/invisible la grilla del gráfico.

Comandos básicos

- 1 `figure`: Crea una nueva ventana gráfica.
- 2 `plot(X,Y,'opciones')`: Comando que realiza el gráfico Y vs X.
- 3 `subplot(m,n,i)`: Permite realizar $m \times n$ gráficos en una misma ventana (i es el gráfico actual).
- 4 `xlabel('eje')`: Asigna el rótulo 'eje' al eje X.
- 5 `ylabel('eje')`: Asigna el rótulo 'eje' al eje Y.
- 6 `title('titulo')`: Define el título del gráfico.
- 7 `legend('n1','n2',...)`: Rótulos de las curvas presentes en el gráfico actual.
- 8 `grid on/off`: Hace visible/invisible la grilla del gráfico.

Ejemplo - Respuesta de un sistema amortiguado de 1 GDL

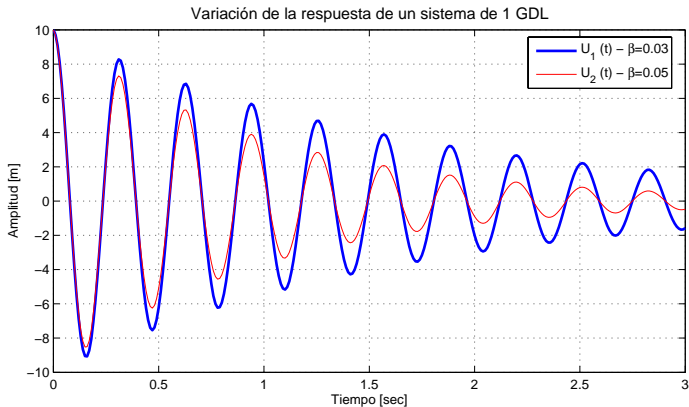
```
%% Generación de gráficos
% Análisis de la variación de la respuesta de un sistema de 1 GDL con
% el amortiguamiento del sistema.
clear all, close all, clc

%% Utilización del comando 'plot'
t=0:0.01:3;
beta=[0.03 0.05];
u1=10*exp(-20*beta(1)*t).*cos(20*t);
u2=10*10*exp(-20*beta(2)*t).*cos(20*t);

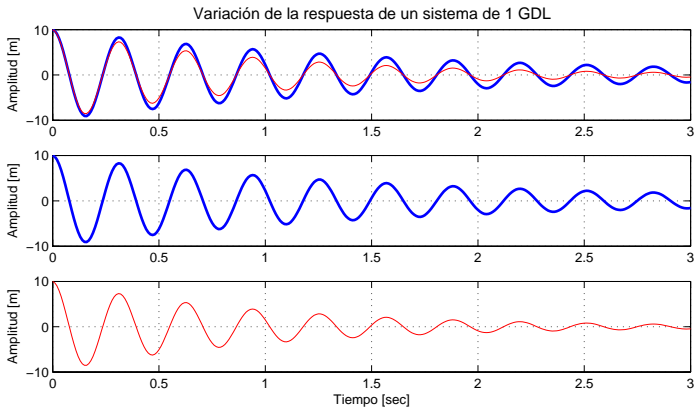
figure
plot(t,u1,'b',t,u2,'r');    grid on;
title('Variación de la respuesta de un sistema de 1 GDL','fontsize',12);
legend('U_1 (t) - \beta=0.03','U_2 (t) - \beta=0.05');
xlabel('Tiempo [sec]');
ylabel('Amplitud [m]');

%% Utilización del comando 'subplot'
figure('Name','Sistema de 1 GDL');
subplot(3,1,1)
plot(t,u1,'b',t,u2,'r');    grid on;
title('Variación de la respuesta de un sistema de 1 GDL','fontsize',12);
ylabel('Amplitud [m]');
subplot(3,1,2)
plot(t,u1,'b');    grid on;
ylabel('Amplitud [m]');
subplot(3,1,3)
plot(t,u2,'r');    grid on;
ylabel('Amplitud [m]');
xlabel('Tiempo [sec]');
```

Ejemplo - PLOT



Ejemplo - SUBPLOT



Contenido

- 1 ¿Qué es Matlab?
- 2 El Entorno Matlab
- 3 El lenguaje Matlab
- 4 Funciones
- 5 Gráficos
- 6 Problema propuesto**

Enunciado

Los archivos `registro1.txt` y `registro2.txt` contienen dos columnas de datos, la primera asociada al tiempo y la segunda al desplazamiento de un sistema amortiguado de 1 GDL (vibraciones libres). Su tarea es determinar el período, T_d , de ambas señales. Para ello,

- Defina una función que permita determinar las raíces de una señal arbitraria (parámetros: t , u).
- Dentro de un m-file, grafique para cada registro, la señal entrada y sus raíces. El título del gráfico debe indicar el período de la respuesta.

Formato del resultado

