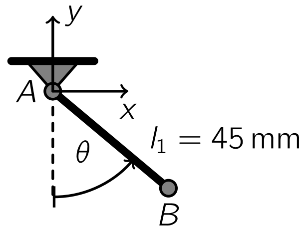
APLICACIONES COMPUTACIONALES [15063]
INGENIERÍA DE EJECUCIÓN MECÁNICA

Martes, 13 de mayo de 2014 (Tiempo: 120 min)

PRIMERA PRUEBA ESPECIAL PROGRAMADA

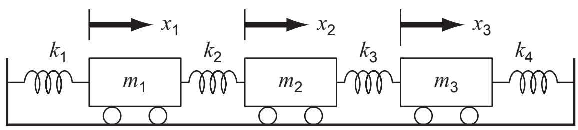
Problema 1 (2 pts): Según un ajuste de datos experimentales, el ángulo $\theta(t)$ del mecanismo mostrado en la figura puede expresarse mediante la siguiente función:

$$\theta(t) = \frac{\pi}{2} e^{-0,2\,t} \sin\left(3\,t + \frac{\pi}{2}\right)$$



Escriba un programa en Matlab que permita:
(a) Guardar en un archivo de datos la posición x e y del extremo B del mecanismo para los primeros 20 segundos, usando un paso de tiempo $\Delta t = 0{,}05$ (0,5 pts).
(b) Estimar la distancia recorrida por el extremo B (en arco de circunferencia) hasta que el mecanismo consigue una posición de reposo. Defina un criterio para establecer el reposo (1,5 pts).

Problema 2 (2 pts): Considere el sistema de tres masas ($m_1 = m_2 = m_3 = 2\,\mathrm{kg}$) y cuatro resortes ($k_1 = 10$, $k_2 = 40$, $k_3 = 30$, $k_4 = 20\,\mathrm{N/m}$) que aparece en la figura. Las ecuaciones de movimiento para cada masa están dadas por las siguientes ecuaciones diferenciales:

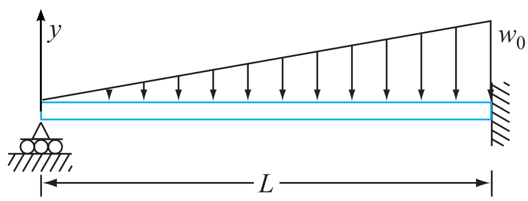


$$\begin{aligned}
(k_1 + k_2)x_1 - k_2x_2 &= -m_1\ddot{x}_1 \\
-k_2x_1 + (k_2 + k_3)x_2 - k_3x_3 &= -m_2\ddot{x}_2 \\
-k_3x_2 + (k_3 + k_4)x_3 &= -m_3\ddot{x}_3
\end{aligned}$$

Si para un instante de tiempo se conocen las aceleraciones del sistema $\ddot{x}_1 = 0{,}15\,\mathrm{m/s^2}$,$\ddot{x}_2 = -0{,}05\,\mathrm{m/s^2}$ y $\ddot{x}_3 = -0{,}45\,\mathrm{m/s^2}$. Se pide:
(a) Escribir el problema en forma matricial identificando la matriz de coeficientes, el vector del lado derecho y el vector de incógnitas (0,5 pts).
(b) Resolver el problema anterior utilizando el método de descomposición LU. Describa el procedimiento y escribas sus resultados paso a paso. Use Matlab para realizar las operaciones. (1,5 pts).

Problema 3 (2 pts): La figura muestra una viga uniforme sujeta a una carga distribuida que crece en forma lineal. La ecuación para la curva elástica resultante es la siguiente

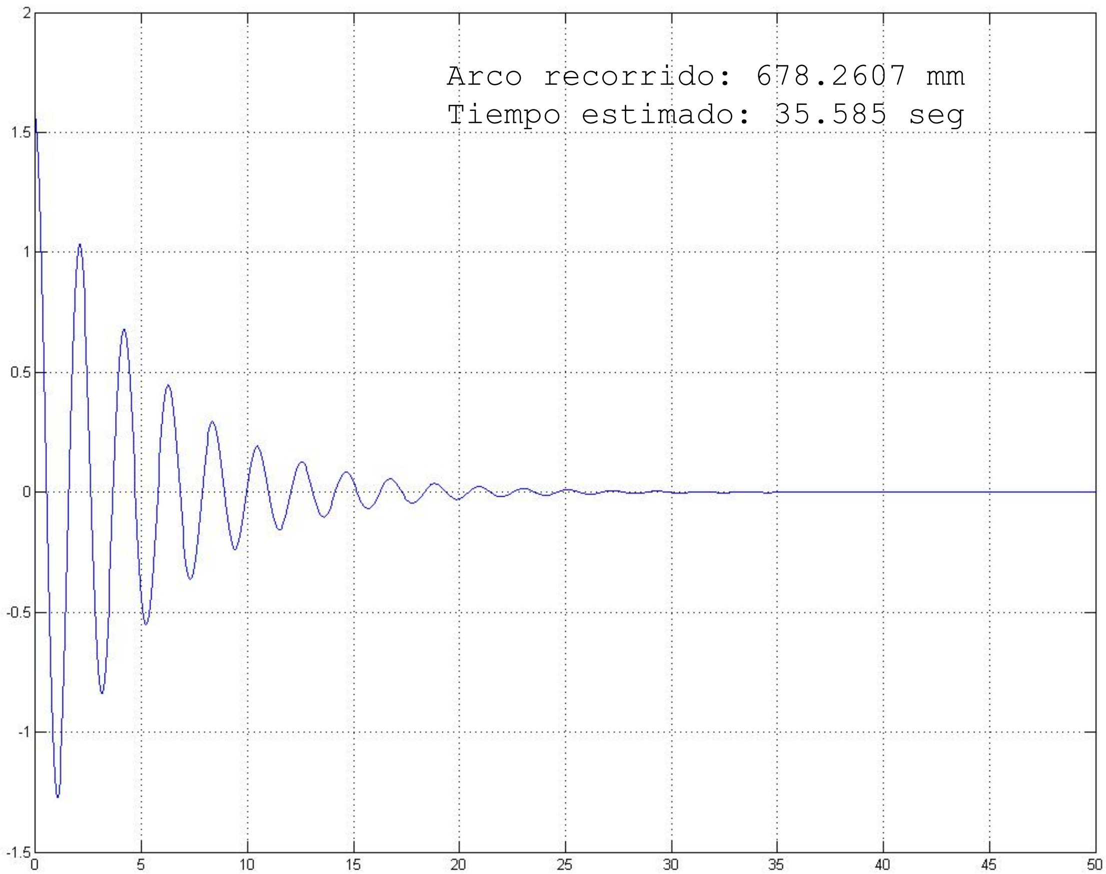
$$y(x) = \frac{w_0}{120EIL}(-x^5 + 2L^2x^3 - L^4x)$$



Para $L = 600\,\mathrm{cm}$, $E = 50000\ \mathrm{kN/cm^2}$, $I = 30000\ \mathrm{cm^4}$ y $w_0 = 2{,}5\,\mathrm{kN/cm}$, se pide utilizar el método de la bisección o de Newton-Raphson para determinar el punto de la viga donde se produce la máxima deflexión.
(a) Escriba en Matlab el algoritmo de solución, comentando las funciones y/o comandos utilizados (1 pts).
(b) Defina y explique el criterio de error utilizado (0,3 pts).
(c) Muestre las últimas 3 iteraciones de su algoritmo (0,7 pts).

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%!
% PEP1 / 2014 / S1 / 15063
% Problema 1
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%!
clc, clear all, close all

% Funcion theta(t) on-line
theta =@(t) pi/2 * exp(-0.2*t).*sin(3.*t+pi/2);

% Intervalo de tiempo y posiciones x(t), y(t)
t = 0:0.05:20;
x = +45*sin(theta(t)); % vector de coordenadas x_i
y = -45*cos(theta(t)); % vector de coordenadas y_i

% Guarda en un archivo de datos los vectores [t], [x] e [y]
% Se usa una matriz auxiliar [aux] para almacenar {t}, {x} e {y}
% en columna

aux = [t' x' y'];
save('position.dat','aux','-ascii')

% Dibujamos {theta} en funcion del tiemp {t}
t = 0:0.05:50;
plot(t,theta(t))
grid on

% Animacion del movimiento del pendulo
% for i=1:length(x)
%     xB = x(i);
%     yB = y(i);
%
%     plot([0 xB],[0 yB],'-o')
%     xlim([-45 45])
%     ylim([-45 45])
%     pause(0.01)
% end

tol = 1e-6;
dis = 1;
k = 1;
t = 0;
dt = 0.005;
arco = 0;

while dis > tol

    theta_1 = theta(t);
    theta_2 = theta(t+dt) ;
```

```
    delta = 45*abs(theta_2 - theta_1);
    arco = arco + delta;

    t = t + dt;
    k = 1 + k;

    dis = delta;
end
disp(['Arco recorrido: ' num2str(arco) ' mm'])
disp(['Tiempo estimado: ' num2str(t) ' seg'])
```



```
clc, clear all, close all

m1 = 2;
m2 = 2;
m3 = 2;

k1 = 10;
k2 = 40;
k3 = 30;
k4 = 20;

a1 = +0.3/m1;
a2 = -0.1/m2;
a3 = -0.9/m3;

A = [k1+k2 -k2 0;
     -k2   k2+k3 -k3
     0     -k3    k3+k4]

b = [-m1*a1; -m2*a2; -m3*a3]

n = length(b);
U = A;
L = tril(ones(n));

% Obtencion de L y U mediante eliminacion de Gauss
for k = 1:n-1
    for i = k+1:n
        f = U(i,k) / U(k,k);
        L(i,k) = f;
        U(i,:) = U(i,:) - f * U(k,:);
    end
    U
end
L

% La idea es resolver en dos etapas el sistema descompuesto [L][U]{x}={b}
% 1) Resolver [L]{y}={b}, mediante sustitucion
%    hacia adelante
%    Donde {y} es un vector incognita intermedio.
%
% 2) Resolver [U]{x}={y}, mediante sustitucion
%    hacia atras
%    Donde {x} es el vector incognita buscado.

c = b;
% 1) Sustitucion hacia adelante
%    para resolver [L]{y}={b}
%
y(1) = c(1) / L(1,1);
for i = 2:n
```

```
    sum = c(i);
    for j = 1:i-1
        sum = sum - L(i,j) * y(j);
    end
    y(i) = sum / L(i,i);
end
y

% 2) Sustitucion hacia atras
%    para resolver [U]{x}={y}
%
x(n) = y(n) / U(n,n);
for i = n-1:-1:1
    sum = y(i);
    for j = i+1:n
        sum = sum - U(i,j) * x(j);
    end
    x(i) = sum / U(i,i);
end
x
%
A =
    50   -40     0
   -40    70   -30
     0   -30    50


b =
   -0.3000
    0.1000
    0.9000


U =
    50   -40     0
     0    38   -30
     0   -30    50


U =
   50.0000  -40.0000         0
         0   38.0000  -30.0000
         0         0   26.3158


L =
    1.0000         0         0
   -0.8000    1.0000         0
         0   -0.7895    1.0000


y =
   -0.3000   -0.1400    0.7895


x =
    0.0100    0.0200    0.0300
```

```
clc, clear all, close all

w0 = 2.5;
L = 600;
E = 50E3;
I = 30E3;

y =@(x) w0/(120*E*I*L)*(-x.^5 +2*L^2*x.^3 -L^4*x);
g =@(x) w0/(120*E*I*L)*(-5*x.^4 +6*L^2*x.^2 -L^4);
f =@(x) w0/(120*E*I*L)*(-20*x.^4 +12*L^2*x^2);
%ezplot(g,[0,600])

% Funcion de Matlab para encontrar el cero de la ecuacion
%x_max = fzero(g,L/2)
%y_max = y(x_max)

err = 1;
tol = 1e-7;
x_max(1) = 1/2*L;
%x_max = 4/5*L;

i = 1;
while err > tol

    x_max(i+1) = x_max(i) - g(x_max(i))/f(x_max(i));

    err = g(x_max)/L*100;
    i = i + 1;
end
y_max = y(x_max);

% Resultados ultimas tres iteraciones
x_max(end-2:end)
y_max(end-2:end)

x = 0:L;
plot(x,y(x),[x_max],[y_max],'o')
grid on
```

```
ans=
  268.3619   268.3617   268.3616
ans =
   -0.5152    -0.5152    -0.5152
```