



Univerzitet u Sarajevu  
Elektrotehnički fakultet u Sarajevu  
Odsjek za računarstvo i informatiku



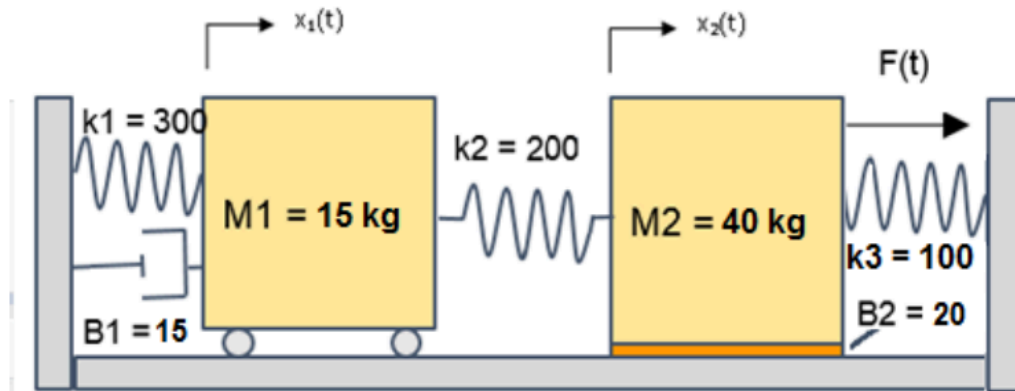
# Zadaca 1

RMS

Ime i prezime: Amer Mujalo  
Index: 19101  
Datum: 22.12.2024g

Neka postoji sistem koji se sastoji od tijela M1 i M2, opruga krutosti  $k_1$ ,  $k_2$  i  $k_3$  i prigušnice sa koeficijentom B1 pri čemu se trenje kotrljanja između tijela M1 i podloge zanemaruje, dok se pretpostavlja da je trenje između tijela M2 i podloge viskozno sa koeficijentom B2.

a) Sila  $F(t) = 0$ , početan položaj M1 je 5 nadesno, prikazati kretanje tijela M1 i M2 na grafikonu.



#### Zadaca1a.m

```
% Glavni program
% Parametri sistema
global M1 M2 k1 k2 k3 B1 B2 Ts;
M1 = 15; M2 = 40; % mase [kg]
k1 = 300; k2 = 200; k3 = 100; % krutosti opruga [N/m]
B1 = 15; B2 = 20; % koeficijenti prigušenja [Ns/m]
Ts = 0.01; % Vrijeme semplinga [s]

% Pocetni uslovi
x1_0 = 5; v1_0 = 0; % Pocetni položaj i brzina M1
x2_0 = 0; v2_0 = 0; % Pocetni položaj i brzina M2
y0 = [x1_0; v1_0; x2_0; v2_0]; % Pocetni uslovi

% Vremenski opseg simulacije
tspan = 0:Ts:20; % Od 0 do 20 sekundi sa korakom Ts

% Rješavanje diferencijalnih jednačina koristeći vlastitu simulaciju
[t, y] = SimulateSystem(@OdeFun, tspan, y0);

% Ekstrakcija rezultata
x1 = y(:, 1);
x2 = y(:, 3);

% Crtanje rezultata
figure;
plot(t, x1, 'r', 'LineWidth', 1.5); hold on;
plot(t, x2, 'b', 'LineWidth', 1.5);
xlabel('Vrijeme [s]');
```

```
ylabel('Položaj [m]');  
legend('x_1(t)', 'x_2(t)', 'Location', 'best');  
title('Kretanje tijela M1 i M2');  
grid on;
```

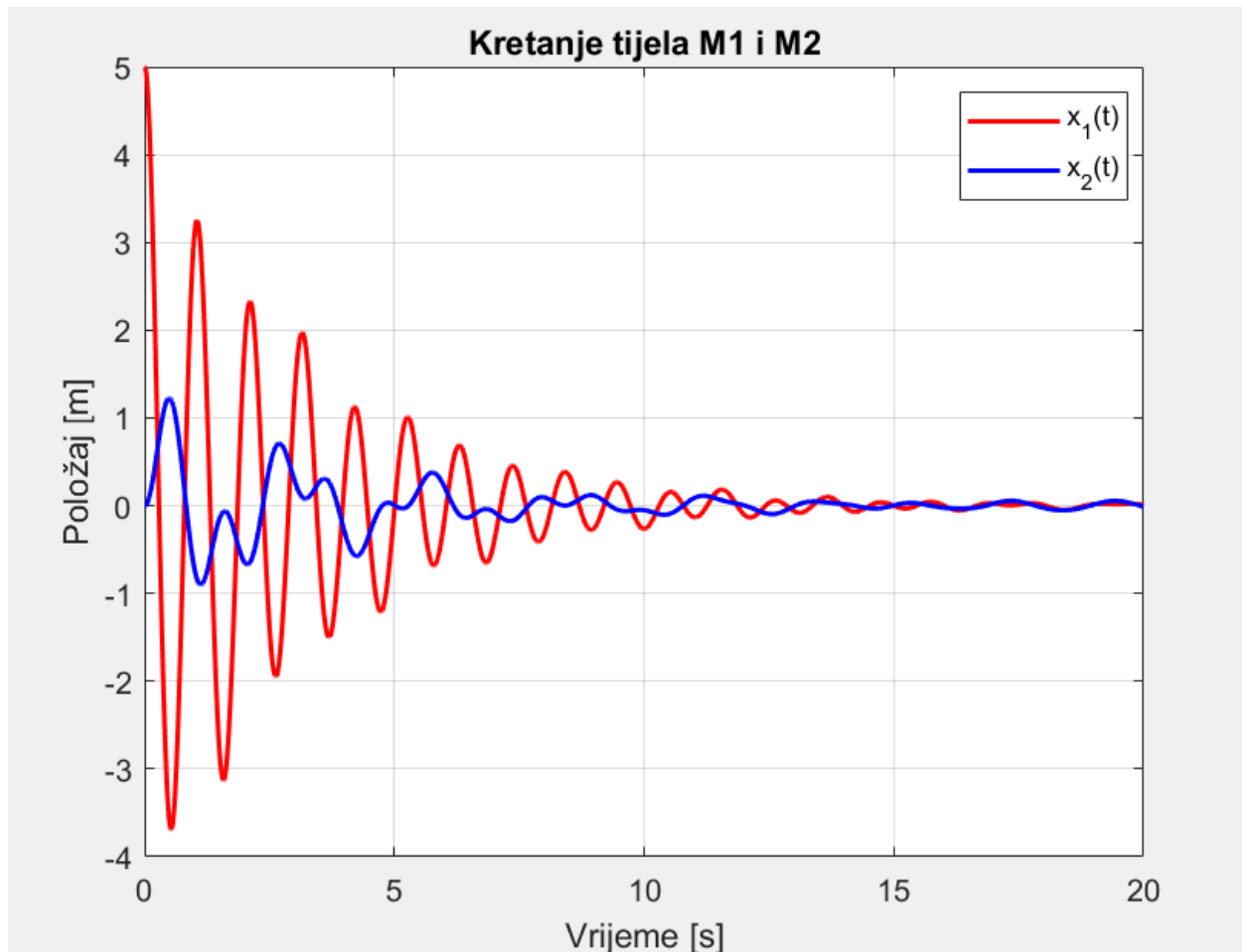
### *SimulateSystem.m*

```
function [t, y] = SimulateSystem(odefun, tspan, y0)  
    % Numericka simulacija sistema koristeći Eulerovu metodu  
    global Ts;  
    N = length(tspan);           % Broj vremenskih koraka  
    y = zeros(N, length(y0));    % Memorija za rezultate  
    t = tspan';                  % Vremenski niz (kolona)  
  
    % Pocetni uslovi  
    y(1, :) = y0';  
  
    % Iterativno rješavanje diferencijalnih jednačina  
    for i = 1:N-1  
        dydt = odefun(t(i), y(i, :)); % Izracun derivacije  
        y(i+1, :) = y(i, :) + Ts * dydt; % Euler korak  
    end  
end
```

### *OdeFun.m*

```
function dydt = OdeFun(t, y)  
    global M1 M2 k1 k2 k3 B1 B2;  
  
    % Stanja sistema  
    x1 = y(1); v1 = y(2);  
    x2 = y(3); v2 = y(4);  
  
    % Sila F(t)  
    F = 0;  
  
    % Diferencijalne jednacine  
    dx1dt = v1;  
    dv1dt = (-k1*x1 - B1*v1 + k2*(x2 - x1)) / M1;  
    dx2dt = v2;  
    dv2dt = (-k2*(x2 - x1) - k3*x2 - B2*v2 + F) / M2;  
  
    % Povratna vrijednost  
    dydt = [dx1dt; dv1dt; dx2dt; dv2dt];  
end
```

>> Zadaca1a



b) Sila  $F(t)$  je stotinu puta veća od signala sa grafikona, a početni uslovi su nulti. Na jednom grafikonu prikazati brzinu i ubrzanje tijela M1, a na drugom brzinu u funkciji vremena tijela M2 i tačku kada se dostiže maksimalna apsolutna brzina tijela M2.

*ForceF.m*

```
function F = ForceF(t)
```

```
% F = ForceF(t)
```

```
F = zeros(size(t)); % Initialize the output F to have the same size as t
```

```

for i = 1:length(t)
    time = t(i);

    time_mod = mod(time, 4);

    if time_mod >= 0 && time_mod < 1
        F(i) = time_mod;
    elseif time_mod >= 1 && time_mod < 2
        F(i) = -time_mod + 2;
    elseif time_mod == 2
        F(i) = 2;
    elseif time_mod > 2 && time_mod <= 3
        F(i) = time_mod;
    elseif time_mod >= 3 && time_mod < 4
        F(i) = -time_mod+6;
    elseif time_mod == 4
        F(i) = 0;
    end

end
end

```

### *OdeFun.m*

```

function dydt = OdeFun(t, y)
    global M1 M2 k1 k2 k3 B1 B2;

    % Stanja sistema
    x1 = y(1); v1 = y(2);
    x2 = y(3); v2 = y(4);

    % Sila F(t)
    F = 100*ForceF(t); %Jedina izmjena je ovdje

    % Diferencijalne jednacine
    dx1dt = v1;
    dv1dt = (-k1*x1 - B1*v1 + k2*(x2 - x1)) / M1;
    dx2dt = v2;
    dv2dt = (-k2*(x2 - x1) - k3*x2 - B2*v2 + F) / M2;

    % Povratna vrijednost
    dydt = [dx1dt; dv1dt; dx2dt; dv2dt];
end

```

### *Zadaca1b.m*

```
% Glavni skript
clc; clear; close all;

% Globalni parametri sistema
global k1 k2 k3 B1 B2 M1 M2 Ts;
% Parametri sistema
k1 = 300; k2 = 200; k3 = 100;
B1 = 15; B2 = 20;
M1 = 15; M2 = 40;
Ts = 0.01; % Korak vremenske simulacije

% Vremenski period simulacije
T = 20; % sekunde
tspan = 0:Ts:T;

% Početni uslovi [x1, v1, x2, v2]
y0 = [0 0 0 0];

% Simulacija sistema
[t, y] = SimulateSystem(@OdeFun, tspan, y0);

% Računanje brzina i ubrzanja
v1 = y(:, 2); % Brzina M1
a1 = diff(v1) / Ts; % Ubrzanje M1
v2 = y(:, 4); % Brzina M2
a1 = [a1; a1(end)]; % Popunjavanje poslednjeg elementa ubrzanja

% Pronalazak maksimalne brzine M2
[max_v2, idx_max_v2] = max(abs(v2));
time_max_v2 = t(idx_max_v2);

% Prikaz rezultata
figure;
subplot(2, 1, 1);
plot(t, v1, 'b', 'LineWidth', 1.5); hold on;
plot(t, a1, 'r', 'LineWidth', 1.5);
legend('Brzina M1', 'Ubrzanje M1');
xlabel('Vrijeme [s]');
ylabel('Brzina / Ubrzanje');
grid on;

subplot(2, 1, 2);
plot(t, v2, 'k', 'LineWidth', 1.5); hold on;
plot(time_max_v2, -max_v2, 'ro', 'MarkerSize', 8, 'LineWidth', 2);
legend('Brzina M2', 'Maksimalna brzina M2');
xlabel('Vrijeme [s]');
ylabel('Brzina');
grid on;
```

>> Zadaca1b

