

Proiect IMS

Acest proiect are ca scop efectuarea mai multor operații pentru a analiza și modela un sistem. Proiectul începe cu inițializarea câtorva comenzi explicate în codul ce urmează

```
%% Incarcare a datelor de tipul timeseries  
clear; % sterge toate variabilele din workspace  
close all; % inchide toate ferestrele de tip plot  
clc; % sterge textul din linia de comanda a MATLAB
```

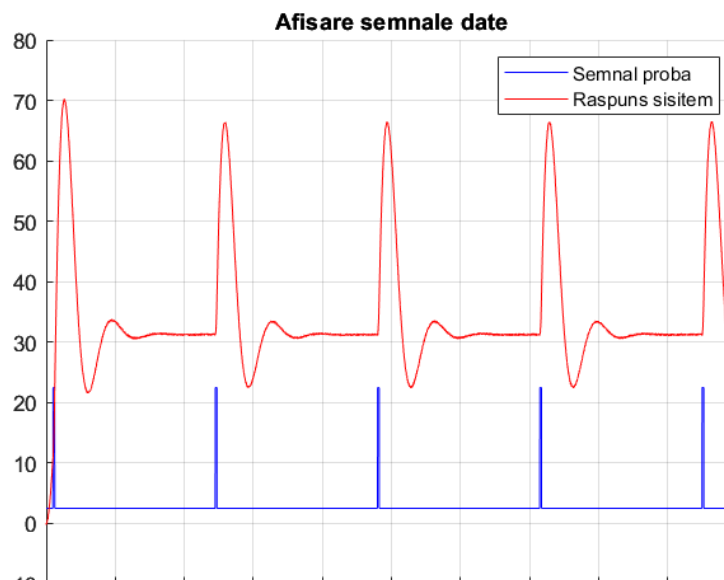
și încărcarea datelor de tipul „timeseries” din fișierul 'data24.mat' prin utilizarea funcției „load” care conține vectorul de timp (t), semnalul de probă (sp) și răspunsul sistemului (sr).

```
load('data24.mat');  
t=data24.Time'; % extragerea vectorului timp  
sp=data24.Data(:,1)'; % extragere semnal proba  
sr=data24.Data(:,2)'; % extragere raspuns sistem
```

Pentru a ne da seama ce fel de sistem este trebuie să afișăm semnalele de intrare și răspunsul sistemului. Pentru a realiza asta ne folosim de comenzile „figure” și „line” din librăria MATLAB.

```
%% Afișare semnale date - Mod 2  
h1=figure('color','w');  
Lsp=line(t,sp,'Color','b');  
Lsr=line(t,sr,'Color','r');  
legend('Semnal proba','Raspuns sisitem');  
title('Afișare semnale date');  
grid on;
```

În această secțiune, semnalele de intrare (sp) și răspunsul sistemului (sr) sunt afișate pe un grafic.

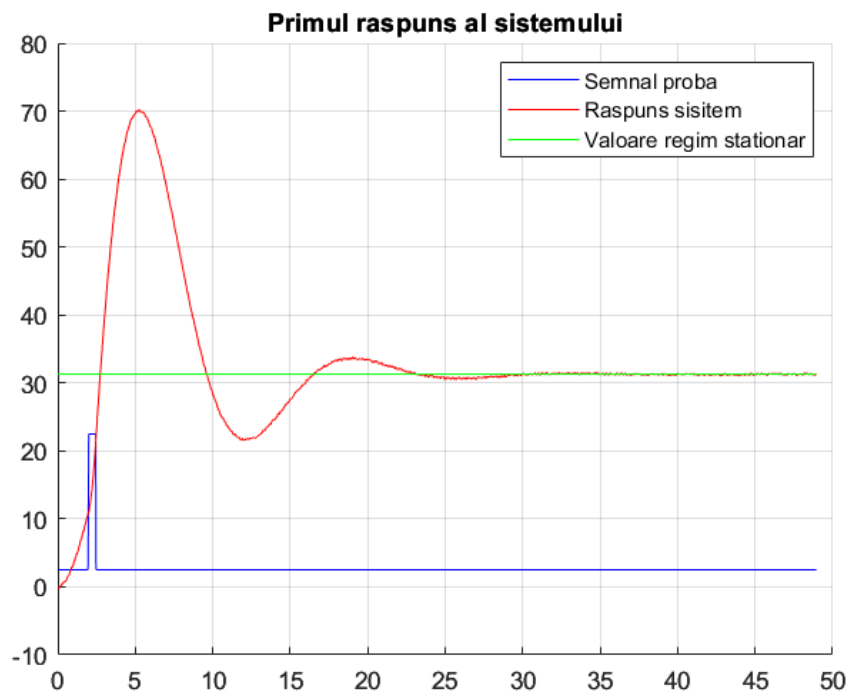


Din grafic ne dam seama ca avem un sistem de ordin II oscilant cu răspuns la impuls. Mai departe vom selecta și afișa primul răspuns al sistemului. Pentru aceasta se folosesc variabilele „t1”, „ysp” și „ysr” pentru a stoca și afișa aceste date. Din graficul anterior observam ca următorul impuls se petrece la t=49, deci in consecință vom lua timpul pana la acesta.

```
%% Selectare si afisare primul raspuns al sistemului  
idx=(find(t<=48.95));%timpul in care apare primul raspuns al sistemului  
t1=t(idx);  
ysp=sp(idx);  
ysr=sr(idx);  
% afisare semnal primul raspuns  
h2=figure('color','w');  
Lysp=line(t1,ysp,'color','b');  
Lysr=line(t1,ysr,'color','r');  
grid on;
```

Se creează o nouă figură grafică (h2) și se afișează semnalul de probă în albastru și răspunsul sistemului în roșu. In același grafic se va prezenta si valoarea de regim staționar notata „ValRst” prin intermediul unei linii verzi orizontale corespunzătoare valorii regimului staționar. Pentru calculul acestuia se găsește indicele timpului la care începe regimul staționar (30 in acest caz) si se calculează valoarea medie a răspunsului sistemului în regimul staționar utilizând funcția „mean” și se stochează în variabila „ValRst”.

```
%% calcul valoare regim stationar  
idx1=find(t1>=30);  
ValRst=mean(ysr(idx1));  
Lrst=line(t1,ones(size(t1))*ValRst,'color','g');  
legend('Semnal proba','Raspuns sisitem','Valoare regim stationar');  
title('Primul raspuns al sistemului');
```



În continuare se determină valoarea maximă (MaxOsc) și minimă (MinOsc) ale răspunsului sistemului în primul interval în care acesta oscilează. Pentru aceasta se folosesc funcțiile MATLAB „max” și „min”. Funcția „intersect” e folosită aici pentru a se selecta perioada de timp în care sistemul oscilează.

```
%% aflare maxim si minim
MaxOsc=max(ysr);
idx2=intersect(find(t1>=2.75),find(t1<=48.95));
MinOsc=min(ysr(idx2));
```

Calculul ariei A

Mai departe se calculează ariile primelor două oscilații din primul răspuns al sistemului în intervalele de timp corespunzătoare părților pozitive și negative a răspunsului.

```
%% calcul Arie A+
idxA1=intersect(find(t1>=2.75),find(t1<=9.65));

A1=0;
for i=2:length(idxA1)
    med=(( ysr(idxA1(i)) + ysr( idxA1(i-1)) )/2)-ValRst;
    A1=A1+med*(t1(idxA1(i))-t1(idxA1(i-1)));
end

%% calcul Arie A-
idxA2=intersect(find(t1>=9.65),find(t1<=16.55));
A2=0;
for i=2:length(idxA2)
    med=(( ysr(idxA2(i-1)) + ysr(idxA2(i)) )/2)-ValRst;
    A2=A2+med*(t1(idxA2(i-1))-t1(idxA2(i)));
end
```

Amplificarea

Prin analizarea graficului sau al fișierului aflăm prima valoare a semnalului de impuls. Prin împărțirea acesteia cu valoarea regimului staționar aflăm amplificarea.

```
%% Amplificarea
k=ValRst/sp(1:1);
```

$$K=y_{ss}/u_{ss}$$

Raportul dintre arii (M)

Putem calcula raportul dintre ariile primelor doua oscilații cu ajutorul ariilor calculate anterior.

```
%% M(raportul dintre arii)  
M=A2/A1;
```

$$M = \frac{A_-}{A_+}$$

Factorul de amortizare (E)

Factorul de amortizare se calculează implementând următoarea formula in MATLAB:

$$\xi = \frac{\ln \frac{1}{M}}{\sqrt{\pi^2 + \ln^2 M}}$$

```
%% Factorul de amortizare E  
n1=log(1/M);  
n2=sqrt(pi^2+log(M)^2);  
E=n1/n2;
```

Pulsația naturala (W)

Pentru calculul pulsației naturale vom avea nevoie de timpii când primele 3 oscilații ating maximul. Timpul T_0 este perioada de oscilație si poate fi calculat prin 2 metode:

$$T_0 = t_3 - t_1 \text{ sau } T_0 = 2(t_2 - t_1).$$

t_1 -timpul când prima oscilație atinge valoarea maxima a acesteia

t_2 - timpul când a II-a oscilație atinge valoarea maxima a acesteia

t_3 - timpul când a III-a oscilație atinge valoarea maxima a acesteia

In continuare se va implementa formula pulsației in MATLAB:

$$\omega_n = \frac{2\pi}{T_0 \sqrt{1 - \xi^2}}$$

Sau

$$\omega_n = \frac{2}{T_0} \sqrt{\pi^2 + \ln^2 M}$$

```
%% pulsatie naturala
b1=find(sr==MaxOsc);
T1=t(b1); %timp t1
b2=find(sr==MinOsc);
T2=t(b2); %timp t2
T0=2*(T2-T1); %perioada de oscilatie
W=2*pi/(T0*sqrt(1-E^2)); %pulsatie naturala
```

Funcția de transfer

Următoarele linii construiesc polinomul numărător și polinomul numitor ale funcției de transfer. Polinomul numărător este reprezentat de „num”, iar polinomul numitor este reprezentat de „den”. Aceste polinoame sunt definite utilizând valorile calculate ale amplificării (k), pulsației naturale (W) și factorului de amortizare (E).

```
%% functie transfer
num=(k*W^2);
den=[1 2*E*W W^2];
sys=tf(num,den); %construire functie transfer
```

Formula funcției de transfer:

$$H(s) = \frac{K \omega_n^2}{s^2 + 2\xi \omega_n s + \omega_n^2}$$

Răspuns la impuls

Această secțiune a codului este responsabilă de generarea și calculul răspunsului la impuls al sistemului.

```
%% raspuns la impuls
t_imp=linspace(t1(1),t1(end),numel(t1)); %face un vector pentru a genera un interval de timp
u_imp=ones(1,length(t_imp))*2.5; %face un vector cu val de 2.5 de lungimea t1
u_imp(42:50) = 22.5;
%y_imp=impulse(sys, t_imp)*(k);
y_imp=lsim(sys, u_imp, t_imp); %raspuns la impuls
```

Mai departe vom afișa răspunsul la impuls(răspunsul matematic al modelului) si cel al sistemului intr-un singur grafic, putând fi astfel comparate direct vizual.

```
%% afisare raspuns la impuls  
%hold ON;  
h3=figure('color','w');  
L_ysr_t1=line(t1,ysr,'Color','black');  
L_u_imp=line(t_imp,u_imp,'Color','blue');  
L_y_imp=line(t_imp,y_imp,'Color','green');  
grid ON;  
legend('Raspuns sistem','Impuls','Raspuns la impuls');  
title('Raspuns la impuls');
```



Eroare pătratică

Această secțiune a codului calculează eroarea pătratică între răspunsul la impuls simulat și răspunsul sistemului măsurat.

```
%% calcul eroare patratice  
errPatratice = 0;  
for i=1:length(t_imp)  
    errPatratice = errPatratice + (y_imp(i,1) - ysr(1,i))^2;  
end
```

Observații:

-nu s-a putut observa timpul mort, deci in consecința l-am considerat ca fiind zero.

```
%% timp mort  
t_mort=0;
```

Mitu Eugen-Renato
Grupa 6301-B
Specializare: EPAE

Mitu Eugen-Renato
Grupa 6301-B
Specializare: EPAE

Mitu Eugen-Renato
Grupa 6301-B
Specializare: EPAE