Proiect Modelare si Simulare
Pendulul Dublu

Dumitru Samuel Irinel - 331AB

December 31, 2023

Cuprins

Cerinta 1

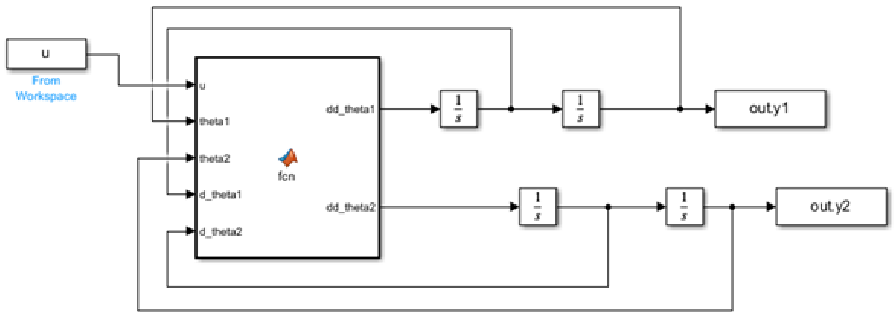

Figure 1: Matlab Function

```
function [dd_theta1,dd_theta2] = fcn(u, theta1,theta2,d_theta1,d_theta2)
zeta = 0.7;
g = 10;
m = 5;
l = 1;

A = - (d_theta2^2) * sin(theta1 - theta2)   - 2 *g*sin(theta1)/l - zeta*d_theta1 + u/(m*l*l)
B =  (d_theta1^2 ) * sin(theta1 - theta2) - g*sin(theta2)/l - zeta*d_theta2;

dd_theta1 = (B* cos( theta1 - theta2)  - A )/( - cos(2*(theta1 - theta2)));
dd_theta2 = (2*B -A* cos( theta1 - theta2)    )/( cos(2*(theta1 - theta2)) );
```

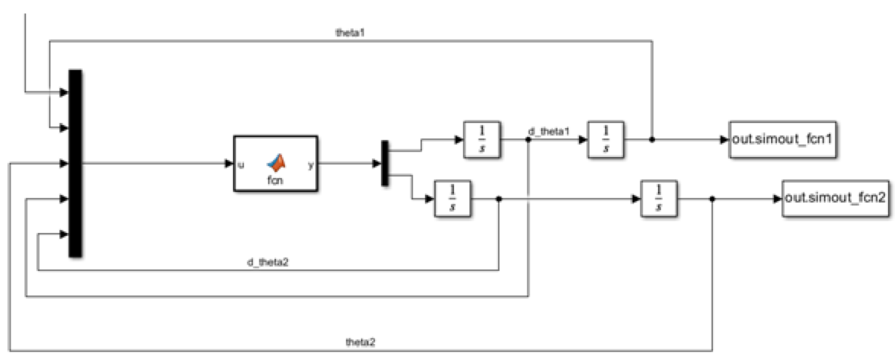

Figure 2: Bloc FCN SISO

```
function y = fcn(u)

zeta = 0.7;
g = 10;
m = 5;
l = 1;
ys = [0;0];
A = - (u(5)^2) * sin(u(2)- u(3))- 2 *g*sin(u(2))/l - zeta*u(4)+ u(1)/(m*l*l);
B =  (u(4)^2 ) * sin(u(2) - u(3)) - g*sin(u(3))/l - zeta*u(5);

ys(1) = (B* cos( u(2) - u(3))  - A )/( - cos(2*(u(2) - u(3))));
ys(2)= (2*B -A* cos( u(2) - u(3))    )/( cos(2*(u(2) - u(3))) );

y = ys;
```

Cerinta 2

Timpul de simulare l-am ales 45 secunde deoarece atunci ajunge modelul meu in regim permanent la intrare treapta*10

```
Tmax = 45;
t = linspace(0,Tmax,100);
u1 = 10.*double(t>=0);
usim = timeseries(u1,t);
```

Cerinta 3

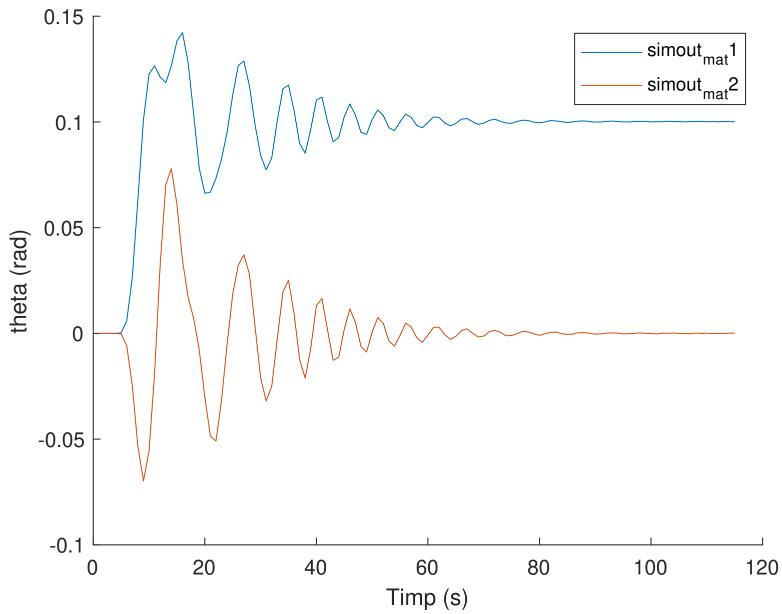

Figure 3: Sistemul MIMO

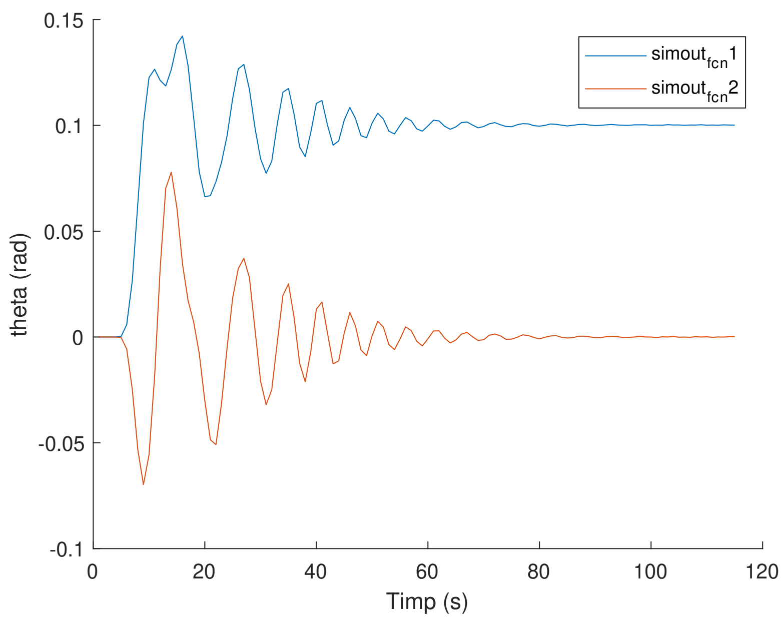

Figure 4: Sistemul SISO

Observam ca iesirile sunt identice chiar daca folosim blocul cu o intrare si o iesire in comparatie cu blocul ce accepta mai multe intrari si mai multe iesiri.

Cerinta 4

```
err1 = norm(simout_mat1 - squeeze(simout_fcn1),2 );
err2 = norm(simout_mat2 - squeeze(simout_fcn2),2 );
```

Asteptarile mele erau sa fie cel putin o diferenta infima intre iesirile respective dar vad conform err1 = 0 si err2 = 0 ca nu exista nicio diferenta intre cele doua implementari. Cred ca acest lucru se intampla deoarece multiplexorul si demultiplexorul sunt ideale, astfel nu exista nici-o intarziere in a selecta canalul dorit si a citi valoarea de pe acel fir.

Cerinta 5 & 6

```
ustar = [0.1 ;0.5;0.82; 1;1.5; 1.8; 2 ; 2.8; 3.1;3.6;4;4.5;10];% luam puncte random
p1 = polyfit(ustar,y1star,1); % polinomul ce trece cel mai aproape de punctele mele cele mai
p2 = polyfit(ustar,y2star,2);
alfa = 3;
beta = 7;
gamma = 11;
p1alfa = polyval(p1,alfa);
p2alfa = polyval(p2,alfa);
```

Valorile pentru scalari si a raspunsurilor lor sunt:

alfa = 3; p1alfa = 0.150621811592054 p2alfa = -9.362544382551802e-05
beta = 7; p1beta = 0.356268146304325 p2beta = 0.002418763033055
gamma = 11; p1gamma = 0.571731077316811 p3gamma = 0.015312791334766

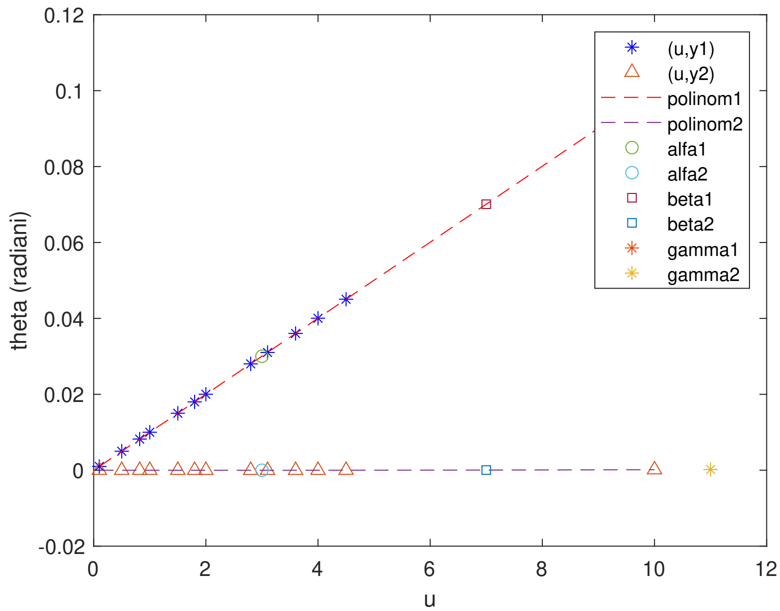

Figure 5: Caracteristica statica

Cerinta 7

Am construit un model separat

```
mdl_pin = 'penduldublu_pin';
```

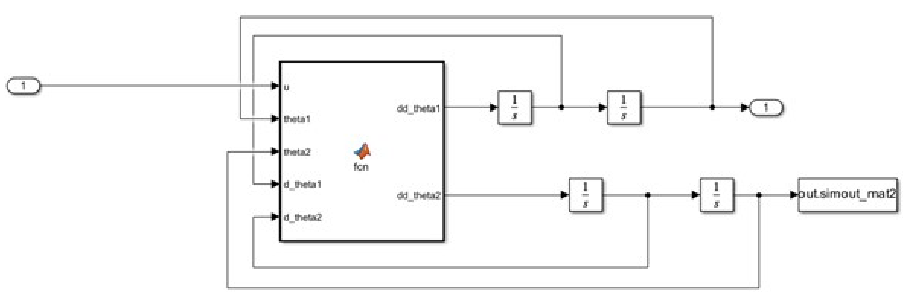

Figure 6: Schema model cu pini IN & OUT

Cerinta 8

```
u0 = 17;
iu = length(u0);

[xstar,ustar,y2star,~] = trim(mdl_pin,[],u0,[],[],iu,[]);

erru = abs(ustar-u0);
```

Cerinta 9

```
[A_lin,B_lin,C_lin,D_lin] = linmod(mdl_pin, xstar, ustar);
```

Cerinta 10

```
vp = eig(A_lin);% este stabil deoarece are 2 perechi de valori proprii complex conjugate in
stabil = -1;
for i = 1:size(vp)
    if (real(vp(i)) > 0)
        stabil = 0;
        disp("sistem instabil");
        break;
    else
        stabil = 1;
    end
end
if(stabil == 1)
    disp("Sistem STABIL");
end
```

```
Sistem STABIL
```

Cerinta 11

```
mdl_liniarizat = 'penduldublu_liniarizat';
r1 = 1.5.*double(t>=0);
usim = timeseries(r1,t);
set_param(mdl_liniarizat,'StopTime',num2str(Tmax));
```

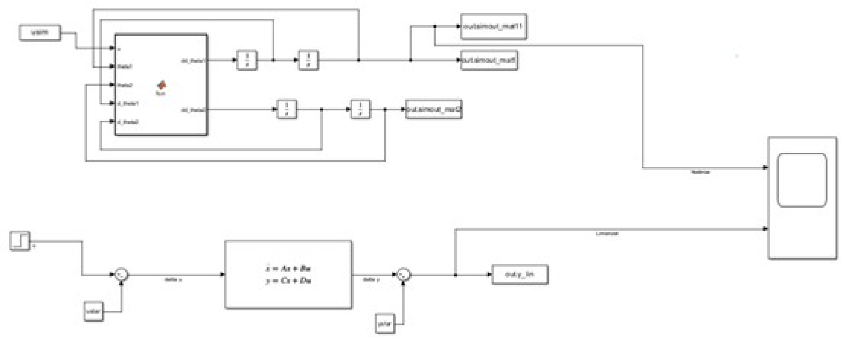

Figure 7: Schema pentru Liniarizare

Cerinta 12

```
y_nl = out1.simout_mat11; %116 linii
err5 = norm(y_nl.Data - y_lin.Data, 'inf')
```

```
err5 =

    0.0032
```

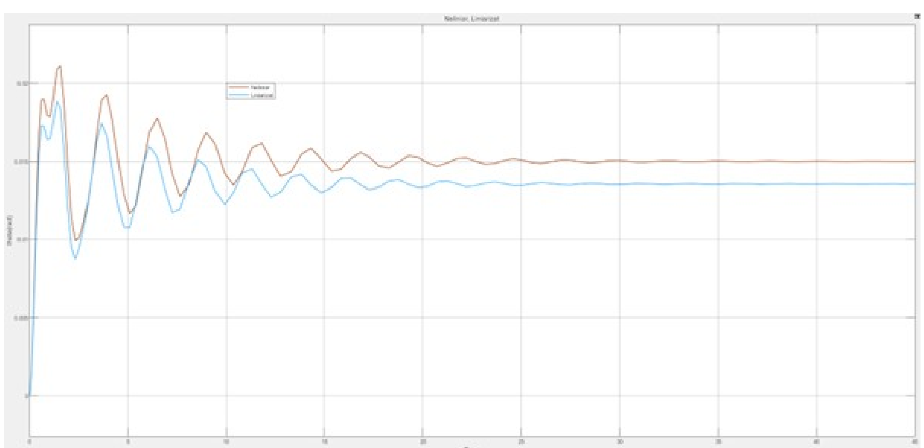

Figure 8: Grafic raspuns neliniar vs liniar

Cerinta 13

```
lin_model = ss(A_lin,B_lin,C_lin,D_lin);
```

```
Te = 0.07;
model_discretizat = c2d(lin_model,Te,'tustin');
isstable(model_discretizat)
```

cu tustin stabilitatea se pastreaza dar planul stabil se muta in discul unitate

Cerinta 14

Ecuatiile cu diferente sunt:
yk = 3.669302429617090*yk1 - 5.211069534922971*yk2 + ...
3.397234082780038*yk3 - 0.860313236420063*yk4 +...
0.000238988402665551*uk + 0.000024589213779549*uk1 - ...
0.000441092984659992*uk2 + 3.609e-15*uk3 + 0.000226693795777581*uk4;

Cerinta 15

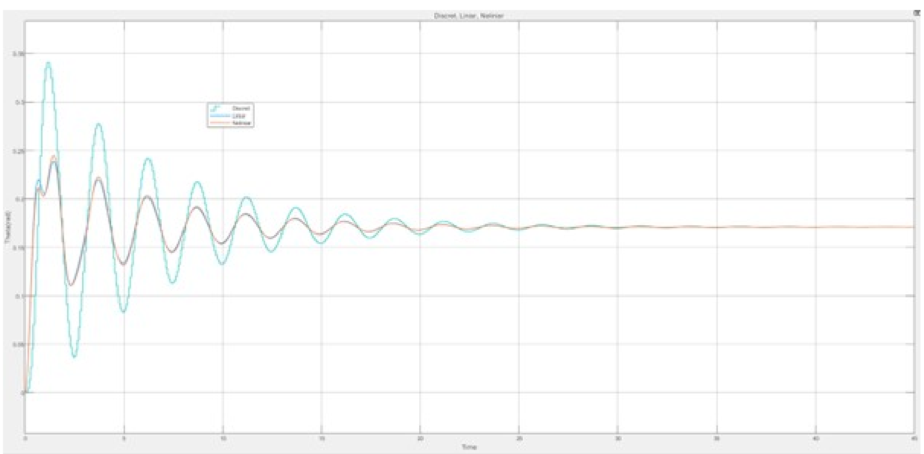

Figure 9: Discret vs neliniar vs liniar