

Molari Stefano 727197 12 12 2023

Inizio ore 12:37

```
#####
                                % esercizio1
#####
clear
clc
#####
% punto 1
#####
% a1 fattore di proporzionalità talento
% a2 fattore di proporzionalità forza mentale

% x1: indice livello tecnico fisico
% x2: indice forza mentale

% u1: carico di lavoro
% u2: ore di lavoro effettuato

% IL SISTEMA è IL SEGUENTE
%  $dx_1/dt = -0.1 \cdot x_1(t) + 1.1 \cdot u_1$ ;
%  $dx_2/dt = 0.5 \cdot x_1 + 1.5 \cdot u_2 - 0.3 \cdot x_2$ 
%  $y = a_1 \cdot x_1(t) + a_2 \cdot x_2(T)$ ;

% lo modellizzo in matlab
syms x1 x2 u1 u2
f=[1.1*u1-0.1*x1 ;
    0.5*x1+1.5*u2-0.3*x2]
%y= a1*x1+a2*x2;

#####
% punto 2
#####

% 2. Indicare, motivando la risposta, se `e preferibile utilizzare le ore di lavoro con
lo psicologo o i carichi di
% lavoro (o se `e indifferente) per controllare la probabilità di vittoria
dell'atleta.

% è un sistema lineare
% guardo se è raggiungibile
A=[-0.1 0;0.5 -0.3]
B=[1.1 0;0 1.5]

R=ctrb(A,B)
rR=rank(R)

% A =
%
%    -0.1000         0
%     0.5000    -0.3000
%
%
% B =
%
%     1.1000         0
%         0     1.5000
```

```

%
%
% R =
%
%      1.1000      0    -0.1100      0
%      0      1.5000    0.5500   -0.4500
%
%
%
% rR =
%
%      2

% valuto un ingresso per volta

B1=[1.1; 0]
B2=[0 ;1.5]

R1=ctrb(A,B1)
rR1=rank(R)
R2=ctrb(A,B2)
rR2=rank(R)

% B1 =
%
%      1.1000
%      0
%
%
% B2 =
%
%      0
%      1.5000
%
%
% R1 =
%
%      1.1000   -0.1100
%      0      0.5500
%
%
% rR1 =
%
%      2
%
%
% R2 =
%
%      0      0
%      1.5000  -0.4500
%
%
% rR2 =
%
%      2
% sono completamente raggiungibili, posso agire su tutti e de

% potrei decidere di agire su u2 visto che ha una costante di
% proporzionalità maggiore rispetto a quella davanti a u1
% in modo che a pari unità di tipo di lavoro (u1 o u2) messo nel sistema si
% ottiene un incremento maggiore

```

```
#####
%
%                               ESERCIZIO 2
#####

#####
% punto a
#####
% (a) Tracciare lo schema simulink del sistema evidenziando la
% trasformazione di uscita e la mappa di transizione dello stato.

syms x1 x2 u
f=[(-x2^2-x2)*x1 ; -x1-x2-x1*u+3*u]
y=5*x1;

#####
% punto b
#####
% (b) Simulare il sistema, su un orizzonte di simulazione di 50s,
% a partire dalla condizione iniziale x0=[3;-3] e per
% u=-0.1.
ueq=-0.1;
x0= [3;-3];
```

schema simulink:

in 1/S ho inserito x0 come condizione iniziale

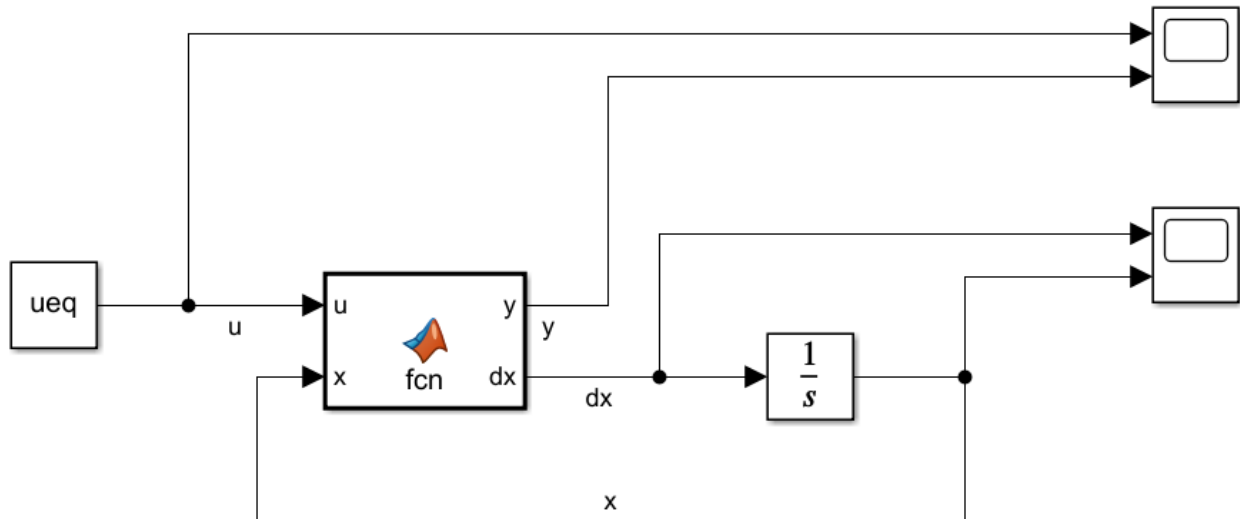


Grafico dell'Ingresso e dell'uscita

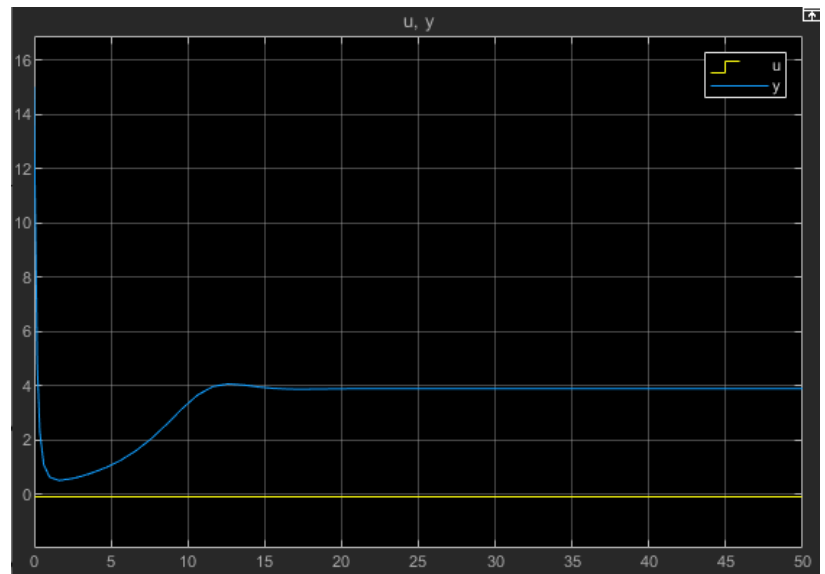
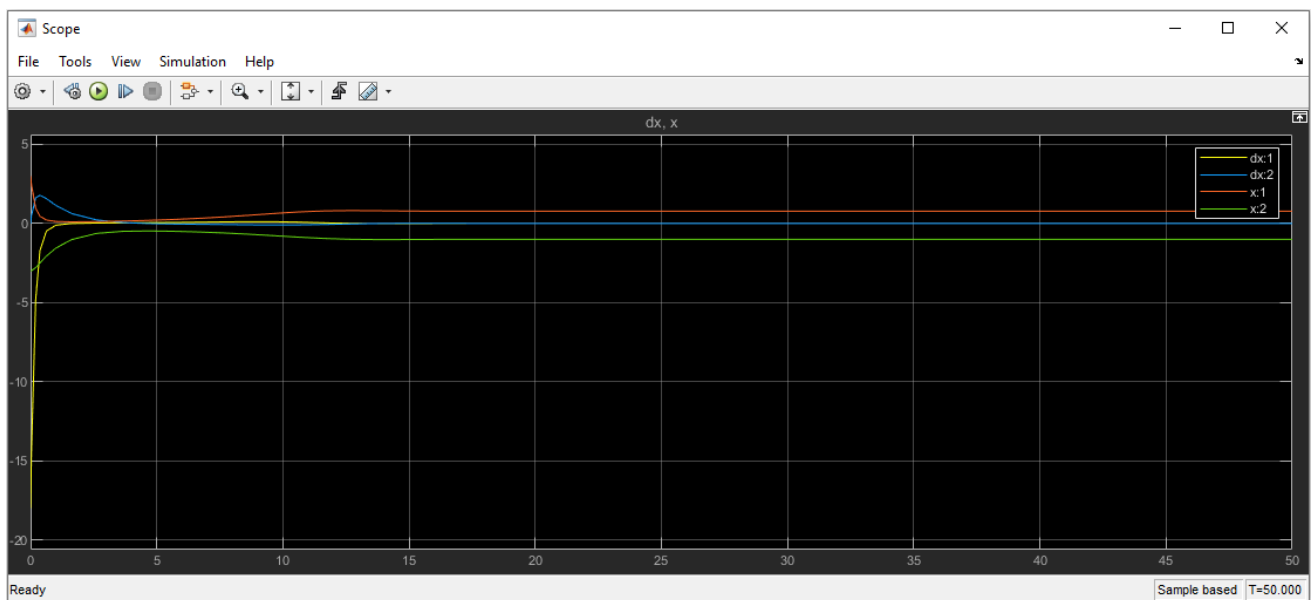
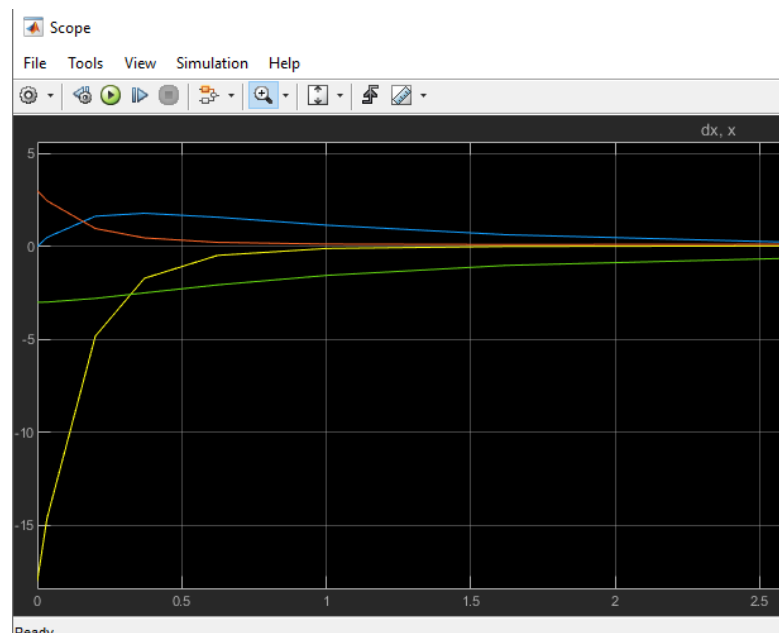


Grafico dello Stato e della derivata dello stato



Uno Zoom



```
#####
% punto c
#####
% b) Scrivere uno script MATLAB che permetta di:
% 1. Calcolare e studiare la stabilit  dei punti di equilibrio del sistema per u=-0.1.
% 2. Calcolare il controllore linearizzante per il sistema.
ueq=-0.1;
% per studiare la stabilit  devo calcolare i punti di equilibrio

xeq_s=solve(subs(f,u,ueq)==0)

% xeq_s =
%
% struct with fields:
%
% x1: [3x1 sym]
% x2: [3x1 sym]
% dall'output vedo che ha 3 possibili punti di equilibrio

xeq1=double([xeq_s.x1(1) xeq_s.x2(1)])
xeq2=double([xeq_s.x1(2) xeq_s.x2(2)])
xeq3=double([xeq_s.x1(3) xeq_s.x2(3)])

% xeq1 =
%
% -0.3333      0
%
%
% xeq2 =
%
%      0 -0.3000
%
%
% xeq3 =
%
% 0.7778 -1.0000

% calcolo la jacobiana
J_s=jacobian(f,[x1 x2])
% J_s =
%
% [- x2^2 - x2, -x1*(2*x2 + 1)]
% [      - u - 1,      -1]

% per studiare la stabilit  valuto la jacobiana in questi tre punti e
% studio gli autovalori

J1=double(subs(J_s,[x1 x2 u],[xeq1 ueq]));
J2=double(subs(J_s,[x1 x2 u],[xeq2 ueq]));
J3=double(subs(J_s,[x1 x2 u],[xeq3 ueq]));

aval1=eig(J1)
aval2=eig(J2)
aval3=eig(J3)
```

```

% aval1 =
%
%   -0.5000 + 0.2236i
%   -0.5000 - 0.2236i
%
%
%
% aval2 =
%
%   -1.0000
%    0.2100
%
%
%
% aval3 =
%
%   -0.5000 + 0.6708i
%   -0.5000 - 0.6708i

% ho un solo punto di equilibrio, xeq2, che è una sella instabile
% xeq1 e xeq2 non sono punti di equilibrio tato che hanno autovalori
% complessi coniugati

% per vedere se posso applicare la I/o linearization devo calcolare il
% grado relativo del sistema

% guardando l'uscita posso notare che essa non dipende esplicitamente da u,
% quindi ha gr>0 (gr:grado relativo)

% calcolo allora la derivata prima
% duplico le variabili per eseguire i calcoli
F=f;G=y;h=y;
dy=jacobian(h,[x1 x2])*F
% dy =
%
% -5*x1*(x2^2 + x2)
% guardando dy posso notare che essa non dipende esplicitamente da u,
% quindi ha gr>1

% calcolo allora la derivata seconda
ddy=jacobian(dy,[x1 x2])*F
% ddy =
%
% 5*x1*(2*x2 + 1)*(x1 - 3*u + x2 + u*x1) + x1*(5*x2^2 + 5*x2)*(x2^2 + x2)

% guardando ddy posso notare che essa dipende esplicitamente da u,
% quindi ha gr=2

% dato che il sistema è del secondo ordine, il grado relativo è uguale
% all'ordine del sistema quindi posso applicare la i/o linearization

% calcolo il controllo linearizzante
syms v
u_lin=solve(ddy==v,u)

% u_lin =
%
% -(5*x1*(2*x2 + 1)*(x1 + x2) - v + x1*(5*x2^2 + 5*x2)*(x2^2 + x2))/(5*x1*(2*x2 +
1)*(x1 - 3))

```

```

#####
% esercizio 3
#####
% punto a
#####
% (a) Scrivere lo script matlab che permetta di scegliere il modello migliore
% come compromesso prestazioni/complessità motivando la risposta.
% Per il modello selezionato riportare le prestazioni (correlazione e MAE)
% in validazione% sulla previsione a 1 mese e a 3 mesi.
% Utilizzare 180 dati del dataset per l'identificazione (i primi) e i
% restanti la validazione.

% per scegliere il modello migliore come compromesos
% prestazioni/complessità devo valutare l'FPE di tutti e trovare il modello
% con il minimo FPE

% carico i dati
load manila_river.mat

dati=iddata(portata,precip);
dati_id=dati(1:180);
dati_val=dati(181:end);

na_max=2; % coeff massimi parte auto regressiva
nb_max=2; % coeff massimi parte esogena
nk=0; % ritardo
orizzonte_previsione=1;

% calcolo i modelli
cont=0;
for iar=1:na_max
    for iex=0:nb_max% datoche è una previsione posso partire da 0
        cont=cont+1;
        na=iar;
        nb=iex;
        nk=0;
        modello=arx(dati_id,[na nb nk]);
        dati_out=predict(modello,dati_id,orizzonte_previsione);
        lista_modelli{cont}=modello;
        struttura(cont,:)= [na nb nk];

        d=na+nb;
        L=length(dati_id.y);% lunghezza serie dati
        e=dati_out.y-dati_id.y;
        FPE(cont)=e'*e/L*(1+d/L)/(1-d/L);
    end
end

[FPE_best,imodel]=min(FPE)
modello_migliore=lista_modelli(imodel)
%present(modello_migliore)

% ho avuto problemi con gli output
% modello_migliore =
%
% 1x1 cell array
%
% {1x1 idpoly}

```

Nome	Valore
cont	6
d	4
dati	250x1x1 iddata
dati_id	180x1x1 iddata
dati_out	180x1x0 iddata
dati_val	70x1x1 iddata
e	180x1 double
FPE	[24.4735,10.7060,10.1...
FPE_best	10.0473
iar	2
iex	2
imodel	6
L	180
lista_modelli	1x6 cell
modello	1x1 idpoly
modello_migliore	1x1 cell
na	2
na_max	2
nb	2
nb_max	2
nk	0
orizzonte_previsio...	1
portata	250x1 double
precip	250x1 double
struttura	6x3 double

```
#####
% punto b
#####
```

```
% Indicare se l'utilizzo della precipitazione nella previsione porta ad un
% miglioramento consistente delle prestazioni a 1 e 3 mesi.
```

```
% devo confrontare l'fpe a un orizzonte di previsione e a 3 oriuzzonti,
% il migliore è
% quello minimo aumentando iex
```