Progetto Controlli Automatici

Autori: Raffaele Giacomo Giovanni Di Maio & Emilio Meroni

Esperimenti Equilibrio

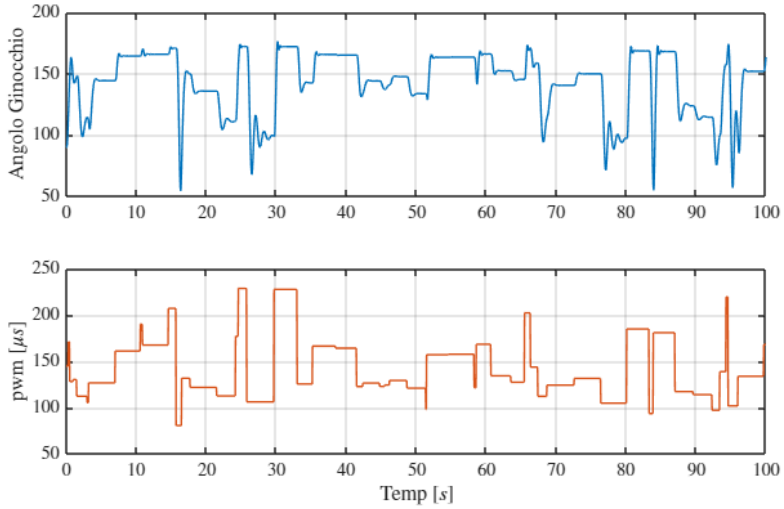
```
clc
clear
close all

addpath(genpath("..\"));

set(groot, 'DefaultTextInterpreter', 'latex');
set(groot, 'DefaultAxesTickLabelInterpreter', 'latex');
set(groot, 'DefaultLegendInterpreter', 'latex');

load("modello\data.mat");
warning("off" , "all");
load_system("modello\model.mdl");

N = length(time);
Ts = time(2) - time(1);

colori = lines(2);
```

Simulazione con i segnali dati:

```
sim("model");
```

```
i = 1
```

```
figure;
subplot(2 , 1 ,1);
plot(time , y , Color=colori(1 , :));
ylabel("Angolo Ginocchio")
grid on;
subplot(2 , 1 , 2);
plot(time , u , Color=colori(2 , :));
ylabel("pwm $[\mu s]$");
xlabel("Temp $[s]$");
grid on;
```

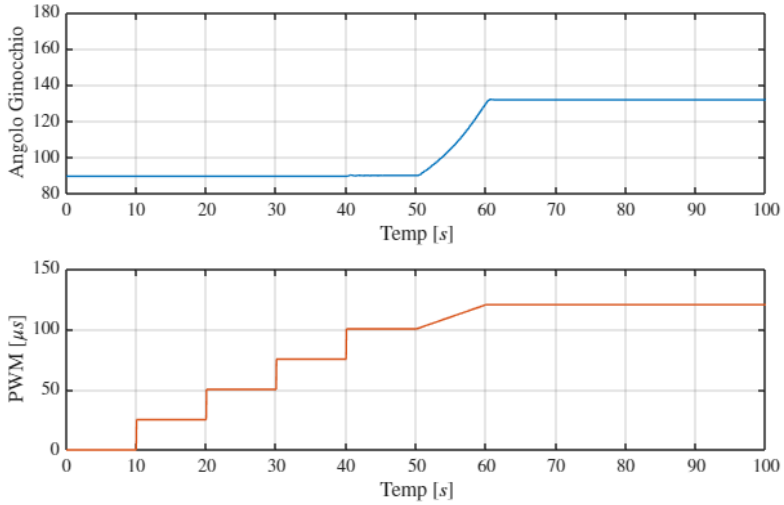


Test di diverse intensità

```
pw = [zeros(201,1); 25 * ones(200 , 1) ; 50 * ones(200 , 1);
75* ones(200 , 1);100 * ones(200 , 1); linspace(100 , 120 , 200)';
120*ones(800 , 1)];
sim("model");
```

```
i = 1
```

```
figure
subplot(2 , 1 , 1);
plot(time , y , DisplayName="Angolo Ginocchio" , Color=colori(1 , :));
ylim([80 180]);
ylabel("Angolo Ginocchio");
xlabel("Temp $[s]$");
grid on

subplot(2 , 1 , 2);
plot(time , u , DisplayName="PWM" , Color=colori(2 , :));
ylim([0 150]);
ylabel("PWM $[\mu s]$");
xlabel("Temp $[s]$");
grid on
```



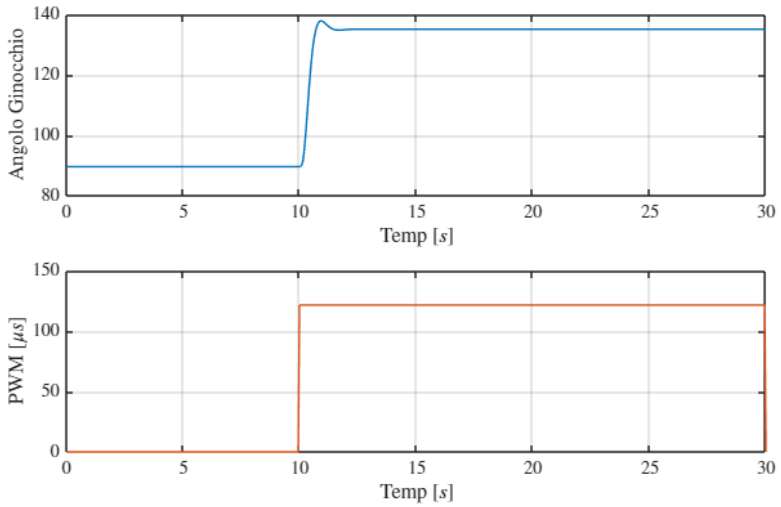
Si sceglie un punto di equilibrio a $135°$

Risposta allo scalino:

```
pw_eq = 121.35;

pw = [zeros(10/Ts , 1) ;            % 10 secondi a 0
        pw_eq * ones(20/Ts , 1);    % 20 secondi a 121.35
        zeros(70/Ts +1 , 1)];
sim("model");
```

```
i = 1
```

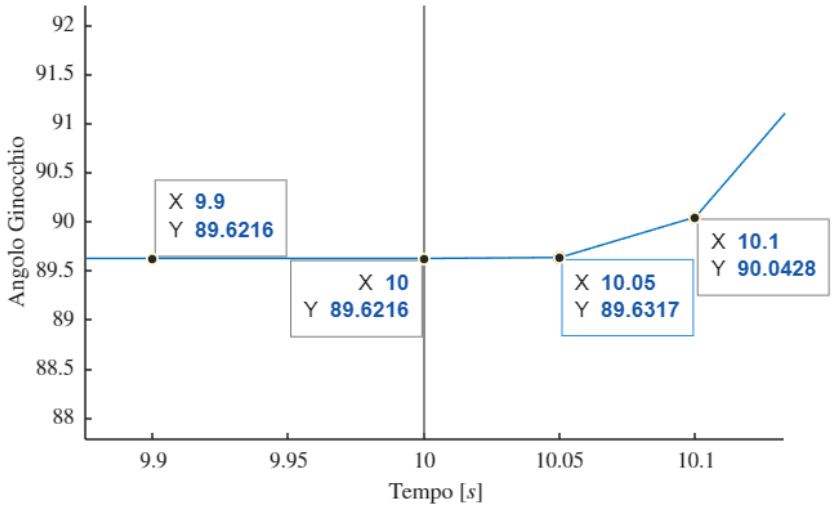
```
figure
subplot(2 , 1 , 1);
plot(time , y , DisplayName="Angolo Ginocchio" , Color=colori(1 , :));
ylabel("Angolo Ginocchio");
xlabel("Temp $[s]$");
xlim([0 30])
grid on

subplot(2 , 1 , 2);
plot(time , u , DisplayName="PWM" , Color=colori(2 , :));
ylabel("PWM $[\mu s]$");
xlabel("Temp $[s]$");
xlim([0 30])
grid on
```



Osservazione del ritardo puro d'ingresso

Si trovano circa $5/10\mu s$ di ritardo che corrispondono a $1/2$passi di simulazione:

```
figure;
indice_inizio = floor(9.95/Ts);
indice_fine = floor(10.5/Ts);
intervallo = indice_inizio:indice_fine;
y_zoom = y(intervallo);
passo_1 = 202;
passo_2 = 203;
time_zoom = time(intervallo);
hold on
plot(time_zoom , y_zoom);
plot(time(passo_1) , y(passo_1) , Marker="x" , LineWidth=3);
plot(time(passo_2) , y(passo_2) , Marker="x" , LineWidth=3);
xline(10)
ylabel("Angolo Ginocchio");
xlabel("Tempo $[s]$")
hold off;

xlim([9.875 10.133])
ylim([87.78 92.21])

hDataTip = findobj(gca,"DataIndex",1);
set(hDataTip,"Location","northwest");
ax2 = gca;
chart = ax2.Children(2);
datatip(chart,10.1,89.95,"Location","southeast");
chart = ax2.Children(4);
datatip(chart,10,89.62,"Location","southwest");
datatip(chart,9.9,89.62,"Location","northeast");
chart = ax2.Children(3);
datatip(chart,10.05,89.63,"Location","southeast");
```



Identificazione

Si vuole controllare l'angolo nell'introno $\pm 10°$ rispetto all'equilibrio ($125°$, $145°$) si è scelto di creare 3 segnali con intensità in un range di circa $115°$ e $155°$:

1. Randomico a blocchi costanti di $5s$ con valori compresi tra $113.5\mu s$ e $135\mu s$;
2. PRBS con ampiezza $113.5\mu s$ e $135\mu s$;
3. Valori randomici tra $113.5\mu s$ e $135\mu s$ a blocchi con tempi variabili tra $0.5s$ e $5s$.

```
pw_max = 135;
pw_min = 113.5;

% i primi tre secondi tolti poiché il segnale non è stazionario in quel
% periodo
primi_secondi = 3/0.05;
time_ridotto = time(primi_secondi:end);
```

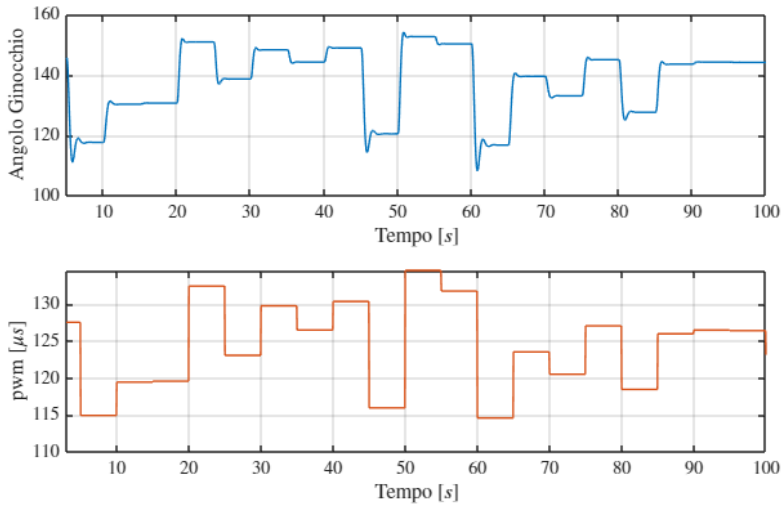
Randomico a blocchi costanti di $5s$ con valori compresi tra $113.5\mu s$ e $135\mu s$

```
load("pw_1.mat");

pw = pw_1;
sim("model");
```

```
i = 1
```

```
y_1 = y(primi_secondi:end);
y_1_media = mean(y_1);
y_1_no_media = y_1 - y_1_media;
pw_1 = pw_1(primi_secondi:end);
pw_1_media = mean(pw_1);
pw_1_no_media = pw_1 - pw_1_media;
figure;
subplot(2 , 1 , 1);
plot(time_ridotto , y_1, Color = colori(1 ,:));
```

```
ylabel("Angolo Ginocchio");
xlabel("Tempo $[s]$")
xlim([5 100])
grid on;
subplot(2 , 1 , 2);
plot(time_ridotto , pw_1, Color = colori(2 ,:));
ylabel("pwm $[\mu s]$")
xlabel("Tempo $[s]$")
xlim([3 100])
grid on
```

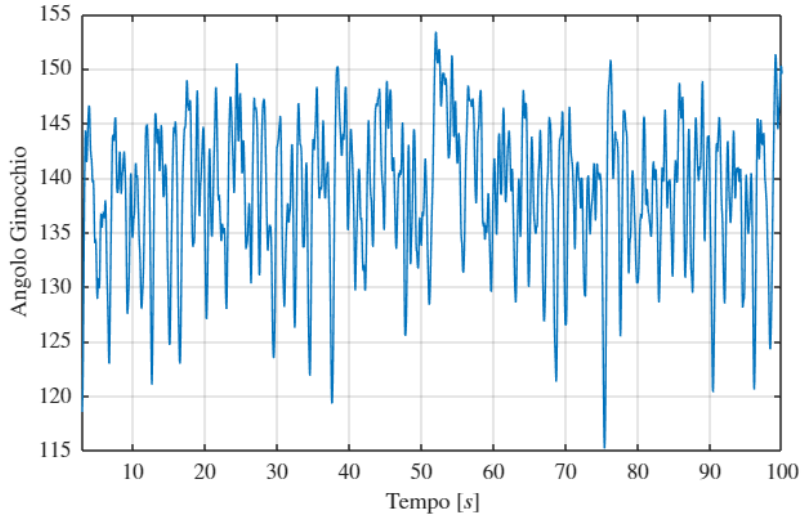


PRBS con ampiezza $113.5\mu s$ e $135\mu s$;

```
range_PRBS = [0 1];
pw_2 = idinput(N , 'prbs' , range_PRBS , [pw_min , pw_max]);
pw = pw_2;

sim("model");
```

```
i = 1
```

```
y_2 = y(primi_secondi:end);
y_2_media = mean(y_2);
y_2_no_media = y_2 - y_2_media;
pw_2 = u(primi_secondi:end);
pw_2_media = mean(pw_2);
pw_2_no_media = pw_2 - pw_2_media;
figure;
subplot(1 , 1 , 1);
plot(time_ridotto , y_2, Color = colori(1 ,:));
ylabel("Angolo Ginocchio");
xlabel("Tempo $[s]$")
xlim([3 100])
grid on;
```

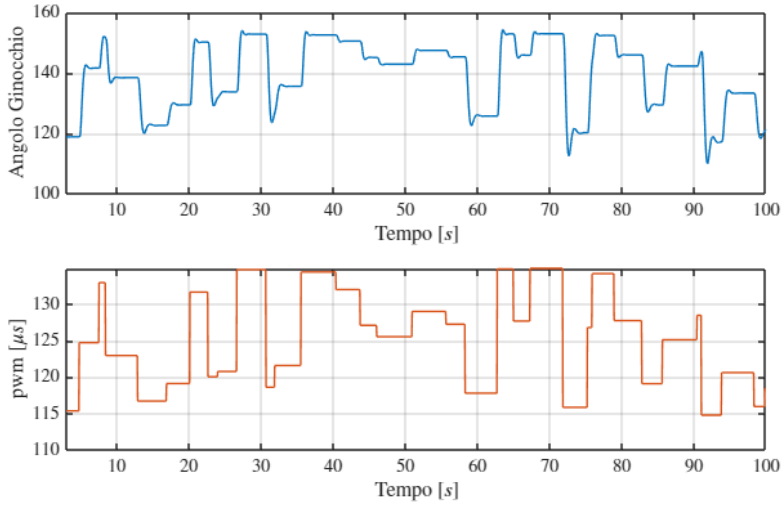


Valori randomici tra $113.5\mu s$ e $135\mu s$ a blocchi con tempi variabili tra $0.5s$ e $5s$

```
load("pw_3.mat");
pw = pw_3;
sim("model");
```

```
i = 1
```

```
y_3 = y(primi_secondi:end);
y_3_media = mean(y_3);
y_3_no_media = y_3 - y_3_media;
pw_3 = u(primi_secondi:end);
pw_3_media = mean(pw_3);
pw_3_no_media = pw_3 - pw_3_media;
figure;
subplot(2 , 1 , 1);
plot(time_ridotto , y_3 , Color=colori(1 , :));
ylabel("Angolo Ginocchio");
xlabel("Tempo $[s]$")
xlim([3 100])
grid on;
subplot(2 , 1 , 2);
plot(time_ridotto , pw_3 , Color=colori(2 , :));
ylabel("pwm $[\mu s]$")
xlabel("Tempo $[s]$")
xlim([3 100])
grid on
```



Identificazione del Modello

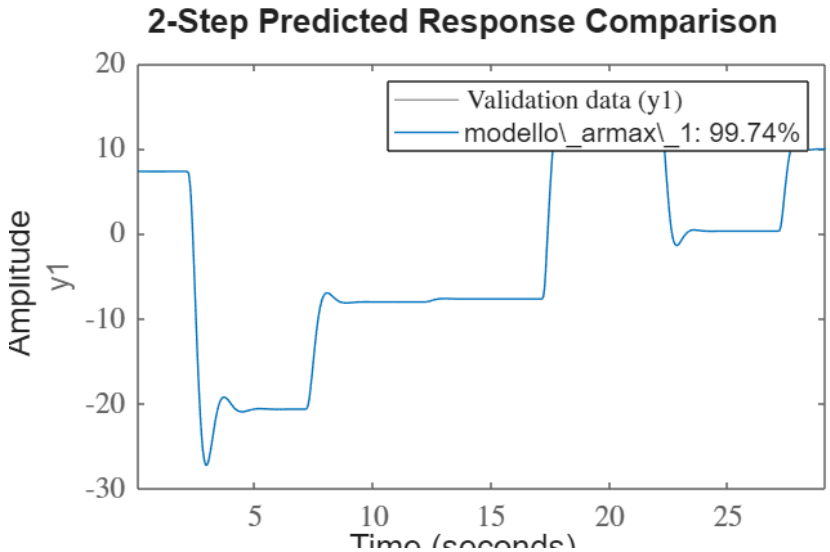
Si utilizza il segnale randomico a blocchi costanti di $5s$

```
% Divisione del data set 70% e 30%
N = length(y_1);

n_70 = floor(N*0.7);
y_modello = y_1_no_media(1:n_70);
u_modello = pw_1_no_media(1:n_70);

data_ident = iddata(y_modello , u_modello , Ts);

% Validazione con il 30% del primo set
n_30 = N - n_70;

y_val = y_1_no_media(1:n_30);
u_val = pw_1_no_media(1:n_30);

data_val = iddata(y_val , u_val , Ts);

%definizione ordini AR MA X e K
n_ar = 4;
n_x = 3;
n_ma = 1;
k = 2;

%Sima del modello
modello_armax_1 = armax(data_ident , [n_ar n_x n_ma k])
```

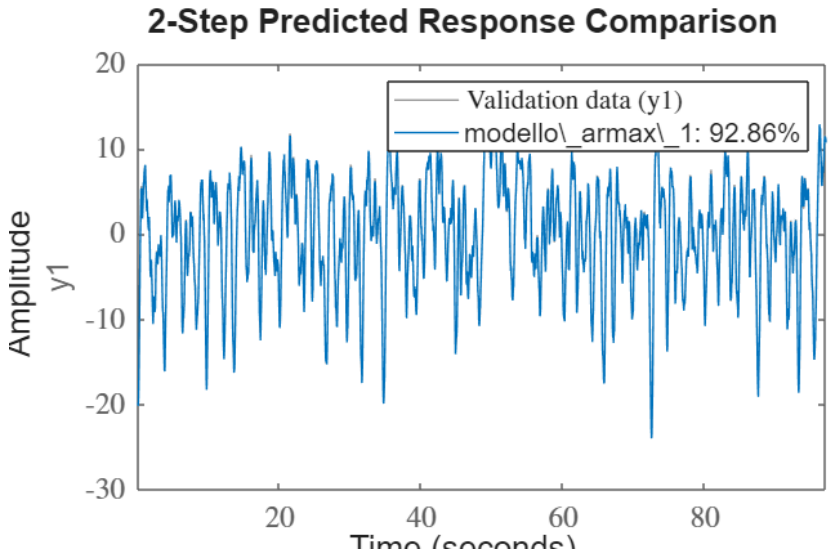
```
modello_armax_1 =
Discrete-time ARMAX model: A(z)y(t) = B(z)u(t) + C(z)e(t)
  A(z) = 1 - 3.31 z^-1 + 4.159 z^-2 - 2.359 z^-3 + 0.5113 z^-4

  B(z) = 0.01985 z^-2 + 0.009946 z^-3 - 0.02851 z^-4

  C(z) = 1 + 0.8559 z^-1
```

```
Sample time: 0.05 seconds

Parameterization:
   Polynomial orders:   na=4   nb=3   nc=1   nk=2
   Number of free coefficients: 8
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using ARMAX on time domain data "data_ident".
Fit to estimation data: 99.94% (prediction focus)
FPE: 5.431e-05, MSE: 5.336e-05

Model Properties
```

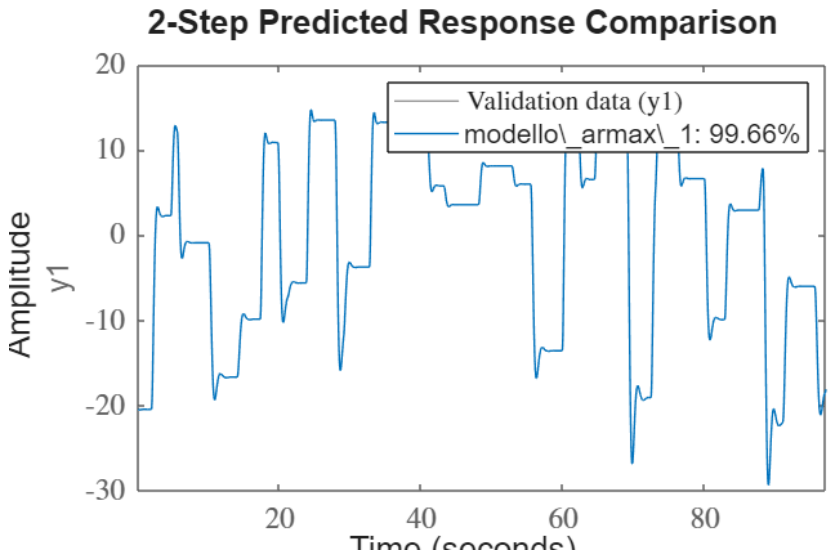
```
figure
compare(data_val, modello_armax_1 , k);
```




```
% Validazione con gli altri set
y_val = y_2_no_media;
u_val = pw_2_no_media;

data_val = iddata(y_val , u_val , Ts);
figure

compare(data_val, modello_armax_1 , k);
```




```
y_val = y_3_no_media;
u_val = pw_3_no_media;

data_val = iddata(y_val , u_val , Ts);

compare(data_val, modello_armax_1 , k);
```



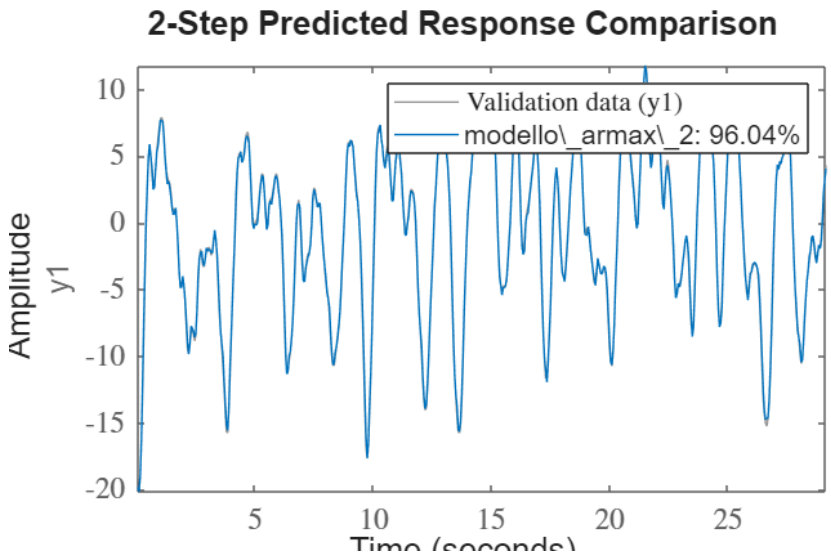
Si utilizza il segnale PRBS

```
% 70% dei dati
y_modello = y_2(1:n_70);
u_modello = pw_2(1:n_70);

data_ident = iddata(y_modello , u_modello , Ts);

% Validazione con il 30% del secondo set
n_30 = N - n_70;

y_val = y_2_no_media(1:n_30);
u_val = pw_2_no_media(1:n_30);
```

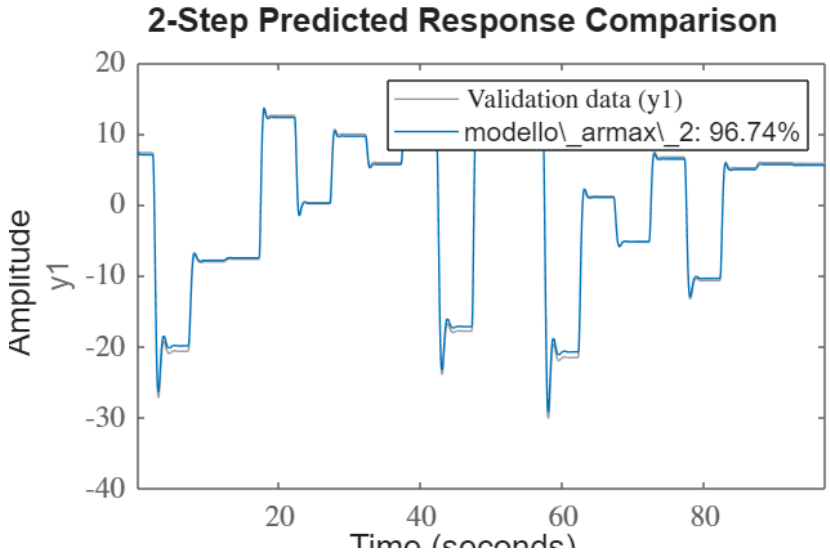
```
data_val = iddata(y_val , u_val , Ts);

%definizione ordini AR MA X e K
% n_ar = 5;
% n_x = 3;
% n_ma = 1;
% k = 2;

%Stima del modello
modello_armax_2 = armax(data_ident , [n_ar n_x n_ma k])
```

```
modello_armax_2 =
Discrete-time ARMAX model: A(z)y(t) = B(z)u(t) + C(z)e(t)
  A(z) = 1 - 2.699 z^-1 + 2.877 z^-2 - 1.487 z^-3 + 0.3371 z^-4

  B(z) = 0.01928 z^-2 + 0.02255 z^-3 - 0.01093 z^-4

  C(z) = 1 + 0.5572 z^-1

Sample time: 0.05 seconds

Parameterization:
   Polynomial orders:   na=4   nb=3   nc=1   nk=2
   Number of free coefficients: 8
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using ARMAX on time domain data "data_ident".
Fit to estimation data: 98.86% (prediction focus)
FPE: 0.005429, MSE: 0.005334

Model Properties
```

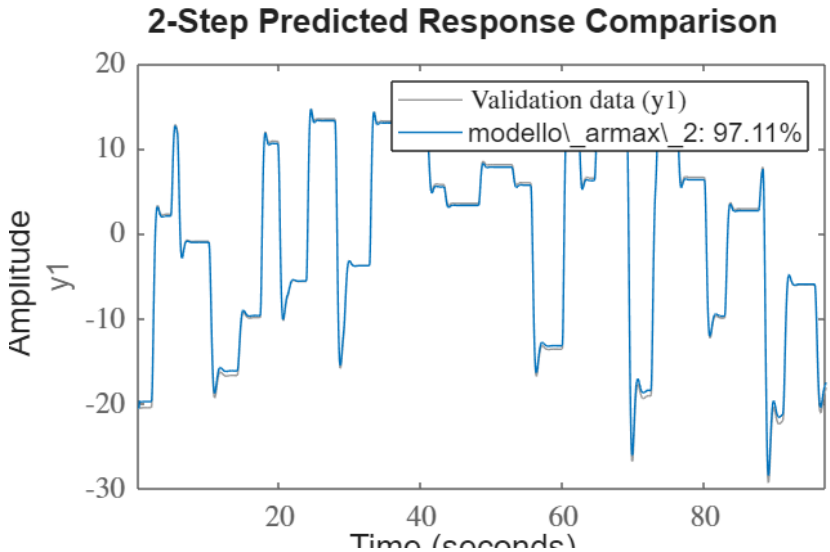
```
figure
compare(data_val, modello_armax_2 , k);
```



```
% Validazione con gli altri set
y_val = y_1_no_media;
u_val = pw_1_no_media;

data_val = iddata(y_val , u_val , Ts);
figure
compare(data_val, modello_armax_2 , k)
```




```
% Validazione con il terzo set
y_val = y_3_no_media;
u_val = pw_3_no_media;

data_val = iddata(y_val , u_val , Ts);
figure
compare(data_val, modello_armax_2 , k)
```



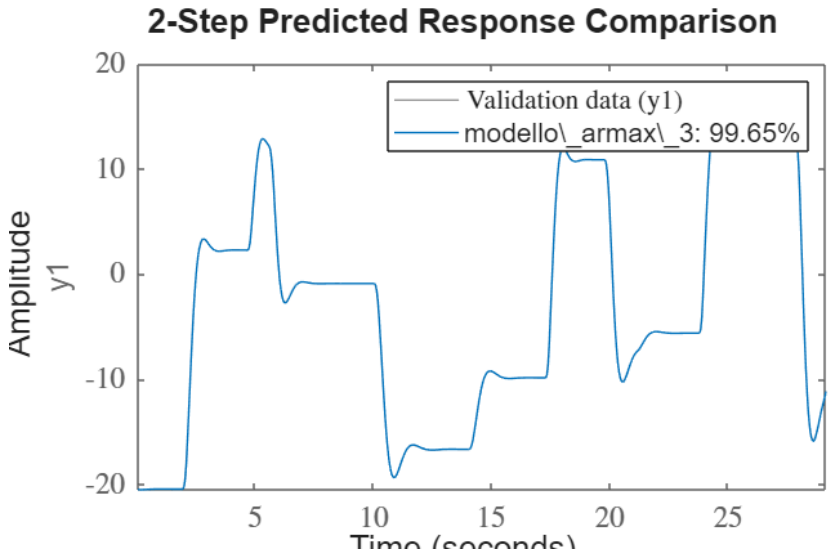
Si utilizza il segnale randomico a blocchi variabili

```
% 70% dei dati
n_70 = floor(N*0.7);
```

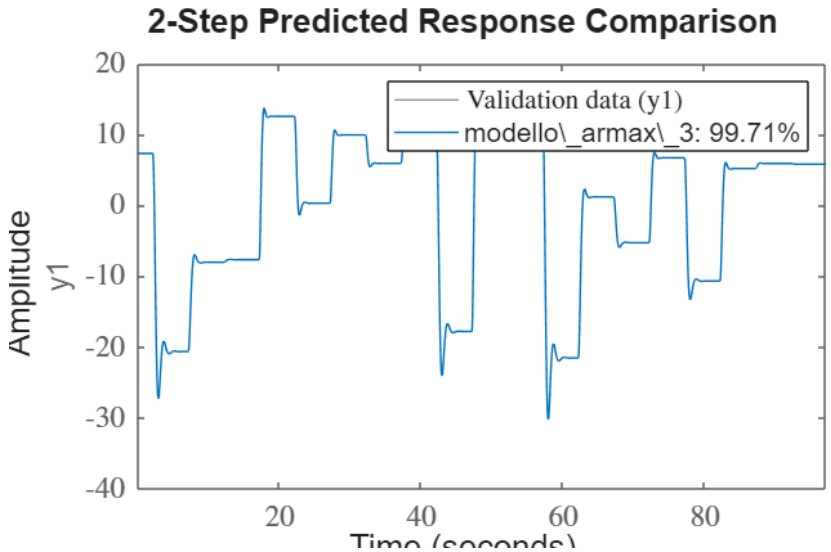
```
y_modello = y_3_no_media(1:n_70);
u_modello = pw_3_no_media(1:n_70);


data_ident = iddata(y_modello , u_modello , Ts);

% Validazione con il 30%
n_30 = N - n_70;

y_val = y_3_no_media(1:n_30);
u_val = pw_3_no_media(1:n_30);

data_val = iddata(y_val , u_val , Ts);

%definizione ordini AR MA X e K
% n_ar = 5;
% n_ma = 1;
% n_x = 3;
% k = 2;

%Sima del modello
modello_armax_3 = armax(data_ident , [n_ar n_x n_ma k])
```

```
modello_armax_3 =
Discrete-time ARMAX model: A(z)y(t) = B(z)u(t) + C(z)e(t)
  A(z) = 1 - 3.248 z^-1 + 4.015 z^-2 - 2.253 z^-3 + 0.4866 z^-4

  B(z) = 0.01999 z^-2 + 0.01016 z^-3 - 0.02898 z^-4

  C(z) = 1 + 0.7998 z^-1

Sample time: 0.05 seconds

Parameterization:
   Polynomial orders:   na=4   nb=3   nc=1   nk=2
   Number of free coefficients: 8
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using ARMAX on time domain data "data_ident".
Fit to estimation data: 99.92% (prediction focus)
FPE: 6.339e-05, MSE: 6.228e-05

Model Properties
```

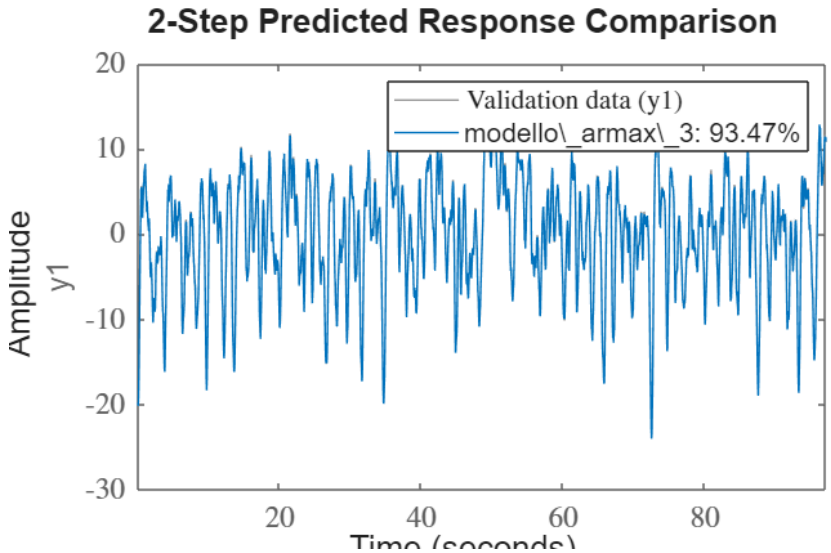
```
figure
compare(data_val, modello_armax_3 , k)
```




```
% Validazione con il primo set
y_val = y_1_no_media;
u_val = pw_1_no_media;

data_val = iddata(y_val , u_val , Ts);
figure
compare(data_val, modello_armax_3, k)
```




```
% Validazione con il secondo set
y_val = y_2_no_media;
u_val = pw_2_no_media;

data_val = iddata(y_val , u_val , Ts);
figure
compare(data_val, modello_armax_3 , k)
```



Di seguito si riportano i determinanti di $R_{uu}^{\{i\}}$ con i il grado di persistenza eccitazione per i tre diversi segnali:

1.Randomico a blocchi costanti:

```
gamma_uu = autocorr(pw_1_no_media, "NumLags", n_x);

for i = 1 : n_x+1
    R_z = toeplitz(gamma_uu(1:i));
    det_R_pw_1(i) = det(R_z);
end
det_R_pw_1
```

```
det_R_pw_1 = 1×4
    1.0000    0.0244    0.0006    0.0000
```

2. PRBS

```
gamma_uu = autocorr(pw_2_no_media, "NumLags", n_x);

for i = 1 : n_x+1
    R_z = toeplitz(gamma_uu(1:i));
    det_R_pw_2(i) = det(R_z);
end
display(det_R_pw_2);
```

```
det_R_pw_2 = 1×4
    1.0000    0.9998    0.9997    0.9995
```

3. Randomico a blocchi variabili:

```
gamma_uu = autocorr(pw_3_no_media, "NumLags", n_x);

for i = 1 : n_x+1
```

```
    R_z = toeplitz(gamma_uu(1:i));
    det_R_pw_3(i) = det(R_z);
end
display(det_R_pw_3);
```

```
det_R_pw_3 = 1×4
    1.0000    0.0345    0.0012    0.0000
```

Controllo a Minima Varianza

Verifica Preliminare sulle radici di $B(z)$

```
modello = modello_armax_2;
modello
```

```
modello =
Discrete-time ARMAX model: A(z)y(t) = B(z)u(t) + C(z)e(t)
  A(z) = 1 - 2.699 z^-1 + 2.877 z^-2 - 1.487 z^-3 + 0.3371 z^-4

  B(z) = 0.01928 z^-2 + 0.02255 z^-3 - 0.01093 z^-4

  C(z) = 1 + 0.5572 z^-1

Sample time: 0.05 seconds

Parameterization:
   Polynomial orders:   na=4   nb=3   nc=1   nk=2
   Number of free coefficients: 8
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using ARMAX on time domain data "data_ident".
Fit to estimation data: 98.86% (prediction focus)
FPE: 0.005429, MSE: 0.005334

Model Properties
```

```
disp("Valore assoluto delle radici di B(z)")
```

```
Valore assoluto delle radici di B(z)
```

```
abs(roots(modello.B))
```

```
ans = 2×1
    1.5382
    0.3686
```

```
disp("Valore assoluto delle radici di A(z)")
```

```
Valore assoluto delle radici di A(z)
```

```
abs(roots(modello.A))
```

```
ans = 4×1
    0.9141
    0.9141
    0.6351
```

```
0.6351
```

Lunga Divisione

$$\frac{C(z)}{A(z)} = E(z) + \frac{\tilde{R}(z)}{A(z)} z^{-k}$$

```
C_z = NumeratoreTF(modello.C , Ts)
```

```
C_z =

  1 + 0.5572 z^-1

Sample time: 0.05 seconds
Discrete-time transfer function.
Model Properties
```

```
A_z = NumeratoreTF(modello.A , Ts)
```

```
A_z =

  1 - 2.699 z^-1 + 2.877 z^-2 - 1.487 z^-3 + 0.3371 z^-4

Sample time: 0.05 seconds
Discrete-time transfer function.
Model Properties
```

```
B_z = NumeratoreTF(modello.B , Ts)
```

```
B_z =

  0.01928 + 0.02255 z^-1 - 0.01093 z^-2

Sample time: 0.05 seconds
Discrete-time transfer function.
Model Properties
```

Eseguiamo la lunga divisione

```
[E_array , R_array] = LungaDivisione(cell2mat(C_z.Numerator), cell2mat(A_z.Numerator) , k);
E_z = NumeratoreTF(E_array , Ts)
```

```
E_z =

  1 + 3.257 z^-1

Sample time: 0.05 seconds
Discrete-time transfer function.
Model Properties
```

```
R_z = NumeratoreTF(R_array ,Ts)
```

```
R_z =

  5.914 - 7.882 z^-1 + 4.506 z^-2 - 1.098 z^-3

Sample time: 0.05 seconds
Discrete-time transfer function.
```

```
Model Properties
```

```
z2 = RitardoPuro(2 , Ts);
z1 = RitardoPuro(1 , Ts);

b=100;
```

$$G(z) = E(z)B(z) + \beta(1 - z^{-1})C(z)$$

```
G_z = E_z * B_z + b*(1 - z1) * C_z
```

```
G_z =

  100 - 44.2 z^-1 - 55.66 z^-2 - 0.03559 z^-3

Sample time: 0.05 seconds
Discrete-time transfer function.
Model Properties
```

```
H_z = C_z
```

```
H_z =

  1 + 0.5572 z^-1

Sample time: 0.05 seconds
Discrete-time transfer function.
Model Properties
```

```
L_z = (1/G_z) * (B_z/A_z)*z2 * R_z
```

```
L_z =

         0.114 z^-2 - 0.0186 z^-3 - 0.1555 z^-4 + 0.1666 z^-5 - 0.07399 z^-6 + 0.012 z^-7
  ------------------------------------------------------------------------------------------
  100 - 314.2 z^-1 + 351.4 z^-2 - 125.7 z^-3 - 60.59 z^-4 + 67.76 z^-5 - 18.71 z^-6 - 0.012 z^-7

Sample time: 0.05 seconds
Discrete-time transfer function.
Model Properties
```

```
phi_z = cell2mat(L_z.Numerator) + cell2mat(L_z.Denominator);
```

Radici di $\varphi(z) = N_L(z) + D_L(z)$

```
abs(roots(phi_z))'
```

```
ans = 1×7
    0.5572    0.9887    0.9195    0.9195    0.6350    0.6350    0.0000
```

```
Fs_z = ((1/G_z) * (B_z/A_z)*z2)/(1+L_z)
```

```
Fs_z =
```

```
                                        1.928 z^-2 - 3.801 z^-3 - 1.403 z^-4 + 8.934 z^-5 - 7.842 z^-6 +
  1.314 z^-7 + 1.829 z^-8 - 1.163 z^-9 + 0.2042 z^-10 + 0.0001311 z^-11
    ----------------------------------------------------------------------------------------------------------
  ----------------------------------------------------------------------------------------------------------
    1e04 - 6.285e04 z^-1 + 1.69e05 z^-2 - 2.46e05 z^-3 + 1.904e05 z^-4 - 3.667e04 z^-5 - 7.323e04 z^-6 +
  7.472e04 z^-7 - 2.655e04 z^-8 - 3523 z^-9 + 6879 z^-10 - 2543 z^-11 + 350.6 z^-12 + 0.2253 z^-13 + 2.081e-20
  z^-14

  Sample time: 0.05 seconds
  Discrete-time transfer function.
  Model Properties
```

Guadagno della $F(z)$

```
dcgain(Fs_z)
```

```
ans = 0.6422
```

Simulazione

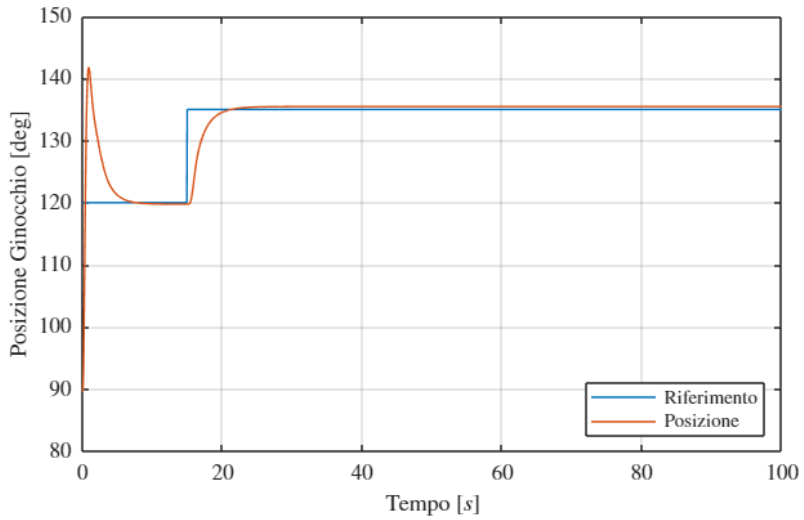
Risposta allo scalino / Sinusoide

```
load_system("controllo_simulink")
%Segnale di ingresso

    valore_medio = 135;%[deg]
    ampiezza_sinusoide = 0;
tempo_simulazione =100 ;%[secondi]
        delay = 15; %[secondi]
            T =10; % Periodo di oscillazione [s]

n_tempo_simulazione = floor(tempo_simulazione/Ts);
n_delay = floor(delay/Ts);
y_0 = [125*ones(1 , n_delay) , valore_medio * ones(1 , n_tempo_simulazione-n_delay)];
time = 0:Ts:tempo_simulazione-Ts;

w = 2*pi/T;
y_0 = [120*ones(1 , n_delay) , ampiezza_sinusoide * sin(w * time(1:n_tempo_simulazione-
n_delay)) + valore_medio];

out = sim("controllo_simulink.slx");
```

```
i = 1
```

```
figure
plot(time , y_0 , displayName = "Riferimento");
hold on
plot(time , out.y1(1:n_tempo_simulazione) , displayName = "Posizione");
xlim([0 , tempo_simulazione]);
legend(location= "southeast");
ylabel("Posizione Ginocchio [deg]" , interpreter = "latex");
xlabel("Tempo $[s]$" , interpreter = "latex");
grid on;
```

```
hold off
```

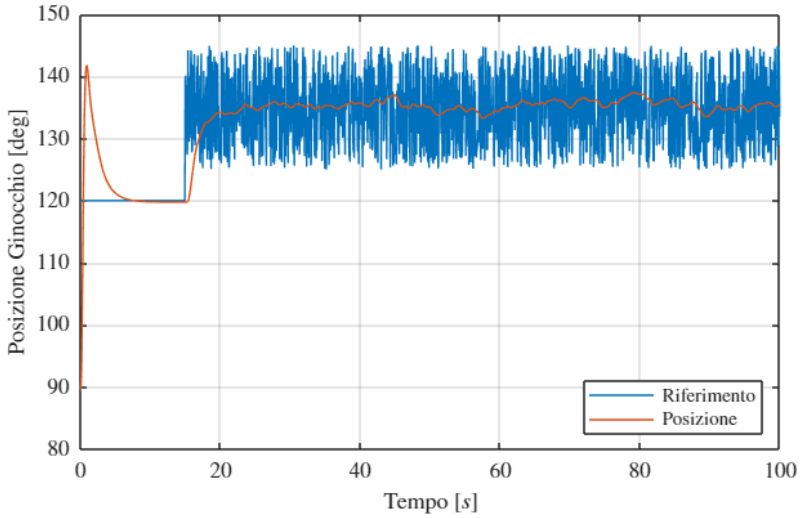


Rumore Bianco

```
val=[125 145] ;
min_val = val(1);
max_val = val(2);
y_0 = [120*ones(1 , n_delay) , min_val + (max_val - min_val) * rand(1 ,
length(1:n_tempo_simulazione-n_delay))];

out = sim("controllo_simulink.slx");
```

```
i = 1
```

```
figure
plot(time , y_0 , displayName = "Riferimento");
hold on
plot(time , out.y1(1:n_tempo_simulazione) , displayName = "Posizione");
xlim([0 , tempo_simulazione]);
legend(location= "southeast");
ylabel("Posizione Ginocchio [deg]" , interpreter = "latex");
xlabel("Tempo $[s]$" , interpreter = "latex");
grid on;
hold off
```


Posizione Ginocchio [deg]
150
140
130
120
110
100
90
80
0
20
40
60
80
100
Tempo [s]
Riferimento
Posizione