# 

# *Proiect la Identificarea Sistemelor*

##### Identificarea unei axe actionate cu un motor BLDC

Coordonator: Student:

Prof.univ.dr.ing Petru Dobra Tamas Bogdan

gr 30144

Ianuarie 2020

# Cuprins

#### 1.Identificarea unei axe actionate cu motor BLDC

##### Obtinerea datelor experimentale..............................................3

* + 1. *Introducere....................................................................*3
  1. Achizitia datelor de intrare-iesire.............................................4
     1. *Desfasurarea experimentului........................................*4
     2. *Preluarea datelor experimentale...................................*5
  2. Procesarea datelor experimentale.............................................6
     1. *Validarea modelului......................................................*6
     2. *Identificarea si validarea pentru viteza unghiulara......*7

*Metoda* ***ARX****..................................................................8*

*Metoda* ***IV4****...................................................................9*

* + 1. *Identificarea si validarea pentru pozitia unghiulara...10*

*Metoda* ***ARX****.................................................................11*

*Metoda* ***IV4****...................................................................12*

* + 1. *Simulare viteza unghiulara...........................................14*
    2. *Simulare pozitie unghiulara..........................................15*
    3. *Validarea modelului......................................................15*

1.4*.* Cod Matlab*...............................................................................17*

**Identificarea unei axe actionate cu un motor BLDC**

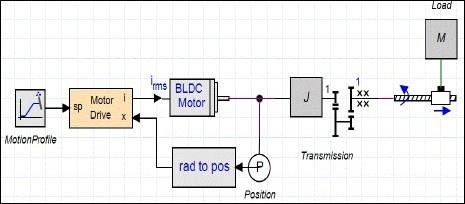
* 1. Obtinerea datelor experimentale
     1. Introducere

În *Figura 1.1* este prezentat un CNC actionat cu motore BLDC.



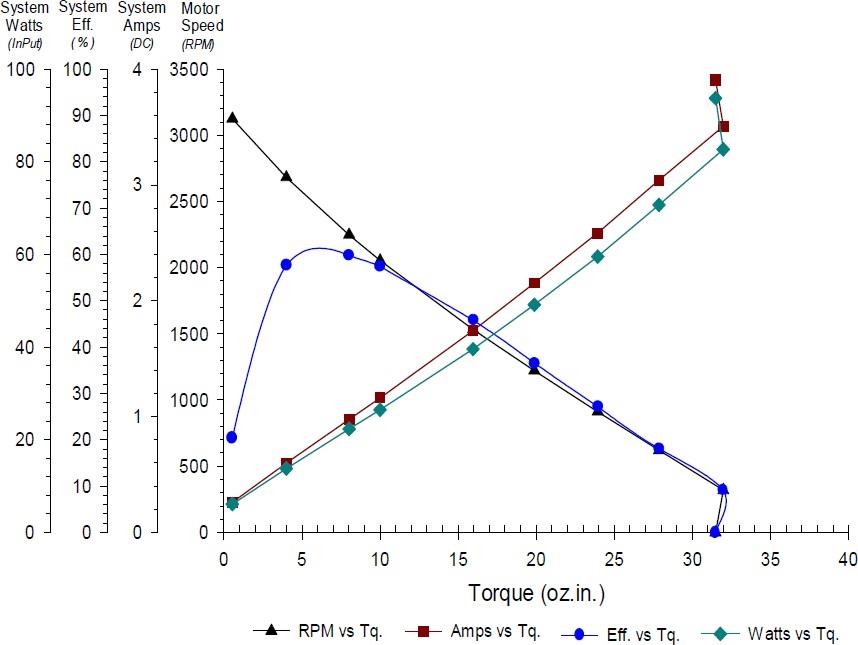
*Figura 1.1*: CNC actionata cu motor BLDC

Sistemul mecanic de pozitionare si sistemul de actionare cu motor BLDC pentru o axa este prezentat in *Figura 1.2:*



*Figura 1.2*: Modelul de actionare si pozitionare al unei axe

Motorul este comandat cu ajutorul unui driver de putere comandat in PWM. Viteza unghiulara si pozitia se masoara pe baza semnalelor provenite de la cei trei senzori Hall montati in statorul motorului. Rotorul motorului BLDC are cinci de poli magnetici, iar caracteriscticile electro-mecanice ale motorului sunt prezentate în *Figura 1.3*:



*Figura 1.3*: Caracteriscticile electro-mecanice ale motorului BLDC

Aparatura utilizata: sursa de alimentare, multimetru, driver de putere, osciloscop, sistem numeric comanda si achizitie a datelor.

* 1. Achizitia datelor de intrare-iesire

Utilizând un sistem numeric de comanda se genereaza semnalele de comanda pentru motorul BLDC (SPAP+SP)¸ si se achizitioneaza datele intrare-iesire în vederea procesarii ulterioare (comanda (factor de umplere), curent(i), viteza unghiulara(ω) si pozitia unghiulara(θ).

* + 1. Desfasurarea experimentului

1. Se alimenteaza ansamblul driver + motor BLDC cu = 24 V .
2. Se efectueaza urmatorul experiment:
   1. Se genereaza un semnal de tip SPAB având caracteristicile corelate cu dinamica ansablului „motor BLDC + axa“;
   2. Se vizualizeaza ¸si se masoara sincron intrarea, si iesirile, obtinând datele experimentale: [tk,u k, ɷk, Ɵk] k=1, 2, ....
      1. Preluarea datelor experimentale

Vom importa datele din fisierul “Tamas.mat” si vom atribui fiecare coloana unei variabile sugestive:

t=double(date.X.Data') - timp

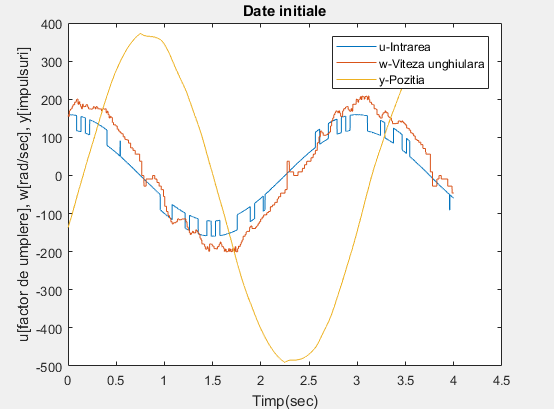
u=double(date.Y(1,3).Data') - intrare

w=double(date.Y(1,2).Data') – viteza unghiulara

y=double(date.Y(1,1).Data') – pozitie unghiulara

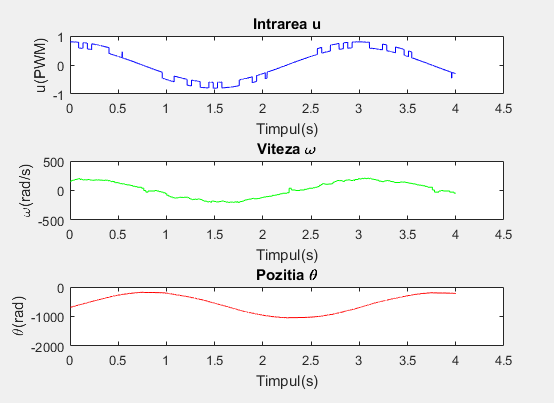
Dupa preluare datelor, acestea se vor ilustra prin comanda plot, iar rezultatul se poate observa in

*Figura 1.4*:



*Figura 1.4*: Date initiale

Comportamentele individuale ale intrarii, vitezei si pozitiei se pot observa in Figura 1.5:

****

*Figura 1.5*: Date initiale preluate individual

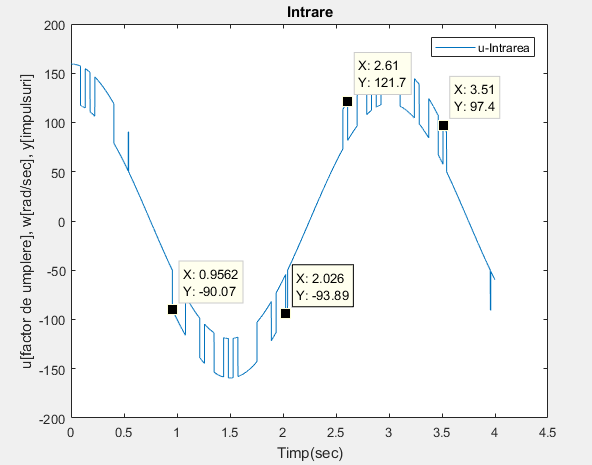
* 1. Procesarea datelor experimentale

Vizualizarea datelor experimentale utilizând mediul Matlab.

Se vor determina functiile de transfer ale ansablului „motor BLDC + axa“ utilizând metodele de identificare parametrica (MCMMPR, MCMMPE, VI, MEP, etc.).

* + 1. Vizualizarea datelor experimentale

Datele de identificare se vor prelua de pe primul sens al semnalului de intrare, iar cele de validare de pe cel de-al doilea dupa cum se poate observa si in *Figura 1.6:*

**

*Figura 1.6:* Preluare indici date identificare si validare

Indicii preluati vor avea valoriile:

-identificare:

**i1=2126;**

**i2=4503;**

**iden=[i1:i2]';**

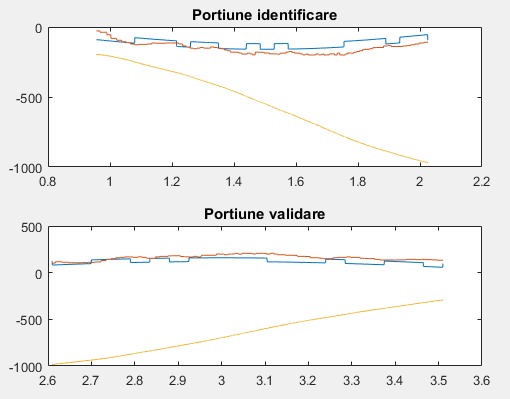
-validare:

**v1=5800;**

**v2=7801;**

**valid=[v1:v2]';**

Se vor ilustra portiunea semnalului de identificare (dintre i1 si i2) si portiunea semnalului de validare (dintre v1 si v2) pentru a observa validitatea alegerii indicilor ( *Figura 1.7* ):



*Figura 1.7:* Portiune identificare + portiune validare

Vom calcula perioada de esantionare prin formula Te=t(2)-t(1), obtinand valoarea

4.500000e-04 (0.45 milisecunde).

* + 1. Identificarea si validarea pentru viteza unghiulara

Folosind indicii preluati din Figura 1.6 vom declara datele de identificare si validare dupa cum

urmeaza:

d\_iden\_w=iddata(w(iden), u(iden), Te);

d\_valid\_w=iddata(w(valid), u(valid), Te);

Pentru identificarea si validare vitezei unghiulare voi folosi metodele **ARX** si **IV4**.

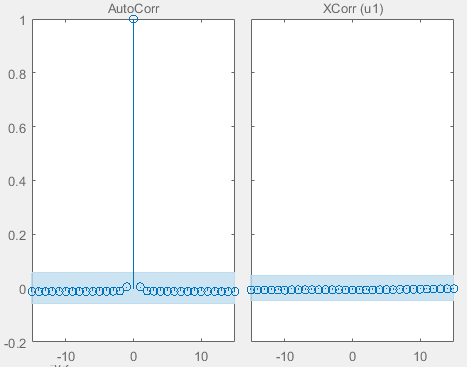
Metoda **ARX** (Metoda celor mai mici patrate recursiva)

**𝐴(𝑞−1)\*𝑦(𝑡)=𝑞-d𝐵(𝑞−1)𝑢(𝑡)+𝑒(𝑡), unde e(t) - eroarea**

Forma model : arx(data, [ nA nB nk ]) ; nA - ordinul polinomului A, nB - ordinul polinomului B, nk - timpul mort.

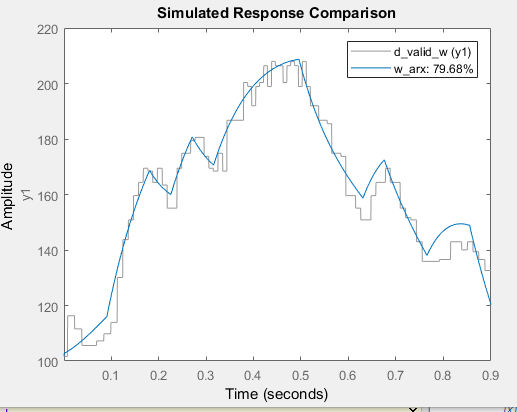
Etape proces:

* Creare model viteza unghiulara : w\_arx=arx(d\_iden\_w,[2 2 1]);
* Validare model (cu metoda resid din matlab) : resid(w\_arx,d\_valid\_w,15);



*Figura 1.8*: Autocorelatie si intercorelatie w, metoda ARX

* Comparare iesire masurata cu iesire date de validare: compare(d\_valid\_w, w\_arx);



*Figura 1.9*: Comparare iesire ARX cu iesire masurata.

* Calcul functii de transfer aferente:
* Functie de transfer in discret: Hzw\_arx=tf(w\_arx.B,w\_arx.A,Te,'variable','z^-1');

1.553 z^-1

Hzw\_arx = ------------------------------

1 - 0.9791 z^-1 - 0.01517 z^-2

* Functie de transfer in continuu:

**Hsw\_arx=d2c(Hzw\_arx,'zoh');**

3623 s^2 + 6.648e07 s + 4.606e11

Hsw\_arx = ----------------------------------------

s^3 + 1.86e04 s^2 + 1.354e08 s + 1.71e09

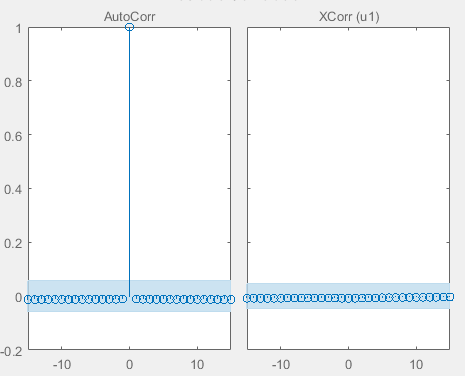
Metoda **IV4** (Metoda variabilelor instrumentale)

**𝐴(𝑞−1)\*𝑦(𝑡)=𝑞-d\*B(𝑞−1)\*𝑢(𝑡)+w’(𝑡), unde w(t)=A(q-1)\*w(t) – perturbatie oarecare**

Forma model : iv4(data, [ nA nB nk ]) ; nA - ordinul polinomului A, nB - ordinul polinomului B, nk - timpul mort.

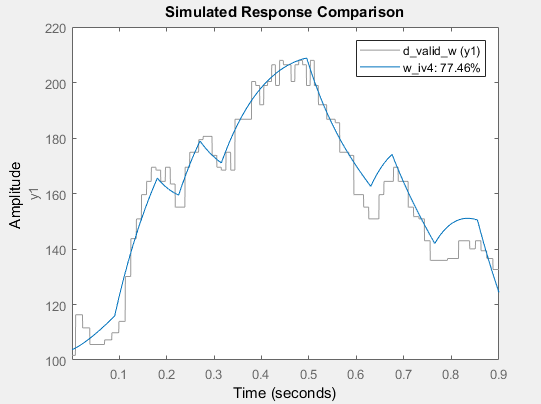
Etape proces:

* Creare model viteza unghiulara : w\_iv4=iv4(d\_iden\_w,[1 1 0]);
* Validare model (cu metoda resid din matlab) : resid(w\_iv4,d\_valid\_w,15);



*Figura 1.10*: Autocorelatie si intercorelatie w, metoda IV4

* Comparare iesire masurata cu iesire date de validare: compare(d\_valid\_w, w\_arx);



*Figura 1.11*: Comparare iesire IV4 cu iesire masurata.

* Calcul functii de transfer aferente:
* Functie de transfer in discret: Hzw\_iv4=tf(w\_iv4.B,w\_iv4.A,Te,'variable','z^-1');

1.342

Hzw\_iv4 = **---------------**

1 - 0.9951 z^-1

* Functie de transfer in continuu:

**Hsw\_iv4=d2c(Hzw\_iv4,'zoh');**

1.342 s + 2990

Hsw\_iv4 = ---------------

s + 11.01

* + 1. Identificarea si validarea pentru pozitia unghiulara

Folosind indicii preluati din Figura 1.6 vom declara datele de identificare si validare dupa cum

urmeaza:

d\_iden\_y=iddata(y(iden),w(iden),Te);

d\_valid\_y=iddata(y(valid),w(valid),Te);

Pentru identificarea si validarea pozitiei unghiulare voi folosi metodele **ARX** si **IV4**.

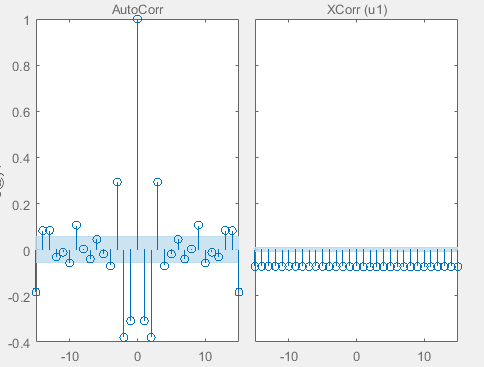
Metoda **ARX** (Metoda celor mai mici patrate recursiva)

**𝐴(𝑞−1)\*𝑦(𝑡)=𝑞-d𝐵(𝑞−1)𝑢(𝑡)+𝑒(𝑡), unde e(t) - eroarea**

Forma model : arx(data, [ nA nB nk ]) ; nA - ordinul polinomului A, nB - ordinul polinomului B, nk - timpul mort.

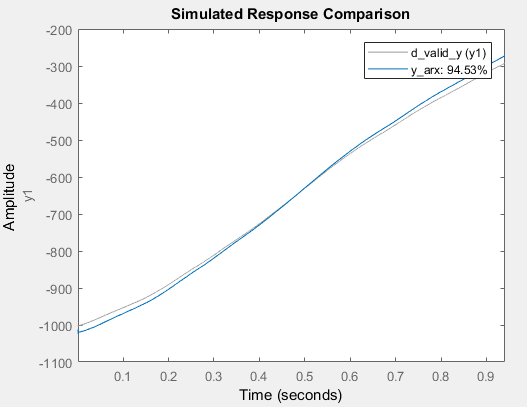
Etape proces:

* Creare model pozitie unghiulara : y\_arx=arx(d\_iden\_y,[2 2 1]);
* Validare model (cu metoda resid din matlab) : resid(y\_arx,d\_valid\_y,15);



*Figura 1.12*: Autocorelatie si intercorelatie y, metoda ARX

* Comparare iesire masurata cu iesire date de validare: compare(d\_valid\_y, y\_arx);



*Figura 1.13*: Comparare iesire ARX cu iesire masurata.

* Calcul functii de transfer aferente:
* Functie de transfer in discret: Hzy\_arx=tf(y\_arx.B,y\_arx.A,Te,'variable','z^-1');

0.001288 z^-1 + 0.002078 z^-2

Hzy\_arx = -----------------------------

1 - 0.4769 z^-1 - 0.5231 z^-2

* Functie de transfer in continuu:

**Hsy\_arx=d2c(Hzy\_arx,'zoh');**

4.04 s^2 - 1.663e04 s + 2.496e08

Hsy\_arx = --------------------------------------

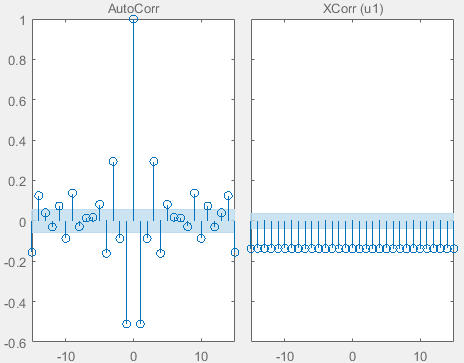
s^3 + 2880 s^2 + 5.081e07 s + 1.843e06

Metoda **IV4** (Metoda variabilelor instrumentale)

**𝐴(𝑞−1)\*𝑦(𝑡)=𝑞-d\*B(𝑞−1)\*𝑢(𝑡)+w’(𝑡), unde w(t)=A(q-1)\*w(t) – perturbatie oarecare**

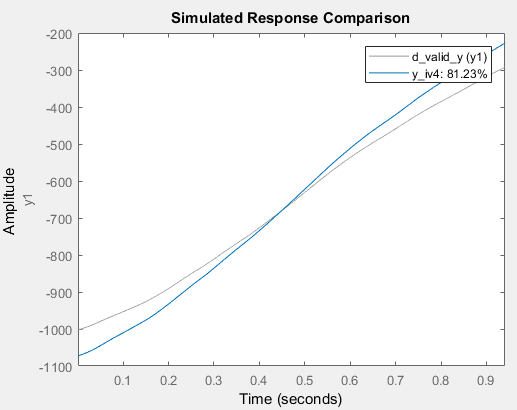
Forma model : iv4(data, [ nA nB nk ]) ; nA - ordinul polinomului A, nB - ordinul polinomului B, nk - timpul mort.

Etape proces:

* Creare model pozitie unghiulara : y\_iv4=iv4(d\_iden\_y,[1 1 0]);
* Validare model (cu metoda resid din matlab) : resid(y\_iv4,d\_valid\_y,15);
* 

*Figura 1.14*: Autocorelatie si intercorelatie y, metoda IV4

* Comparare iesire masurata cu iesire date de validare: compare(d\_valid\_y, y\_arx);



*Figura 1.15:* Comparare iesire IV4 cu iesire masurata.

* Calcul functii de transfer aferente:
* Functie de transfer in discret: Hzy\_iv4=tf(y\_iv4.B,y\_iv4.A,Te,'variable','z^-1');

0.002361 z^-1

Hzy\_iv4 = ----------------

1 - z^-1

* Functie de transfer in continuu:

**Hsy\_iv4=d2c(Hzy\_iv4,'zoh');**

5.246

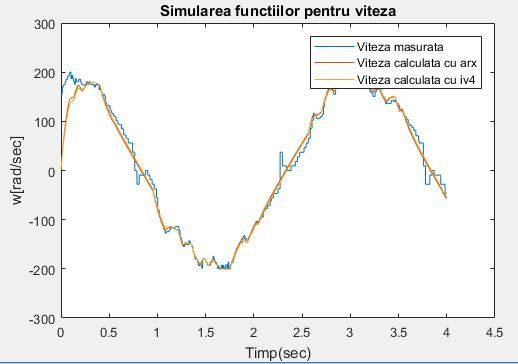
Hsy\_iv4 = ------------

s + 0.1146

* + 1. Simularea viteza unghiulara

Functiile de transfer pentru viteza masurata si caculata cu metodele arx si iv4 sunt afisate in graficul din

*Figura 1.16*:

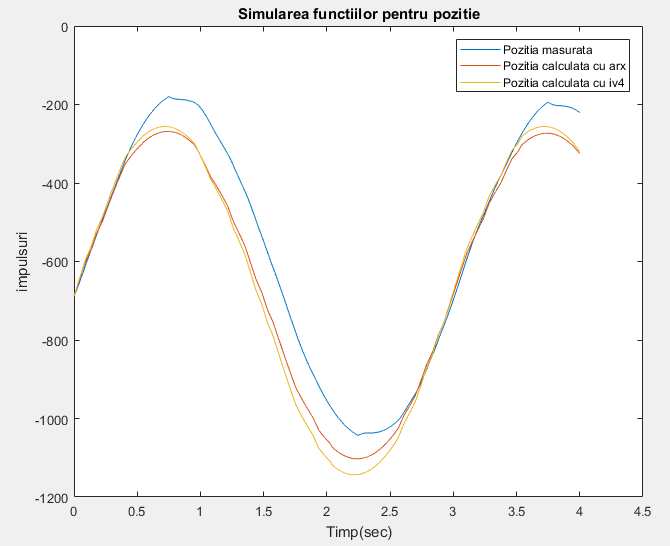


*Figura 1.16:* Simularea vitezelor calculate.

* + 1. Simularea pozitie unghiulara

Functiile de transfer pentru pozitia masurata si caculata cu metodele arx si iv4 sunt afisate in graficul din

*Figura 1.17*:



*Figura 1.17:* Simularea pozitiilor calculate.

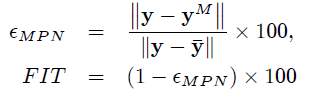
Cod: yc\_arx=lsim(Hsy\_arx,u\*250,t);

yc\_iv4=lsim(Hsy\_iv4,u\*250,t);

* + 1. Validarea modelului

Validarea modelului determinat se face pe baza erorii de predictie reziduale¸ si pe baza decorelarii dintre observatii¸ si eroarea de predictie.

De asemenea se va compara raspunsul experimental cu raspunsul modelului la intrarea cu care a fost obtinut raspunsul experimental. Se calculeaza eroarea medie patratica normalizata (εMPN) ¸si eroarea de urmarire (FIT):



unde y este vectorul masuratorilor, yM raspunsul modelului ¸si este valoarea medie a vectorului masuratorilor a imparti graficul in doua pentru a elimina datele.

Vom calcula folosind intrarea, viteza unghiulara si pozitia unghiulara eroarea medie patratica apeland la urmatoarele functii din matlab: resid, compare, iddata si n4sid.

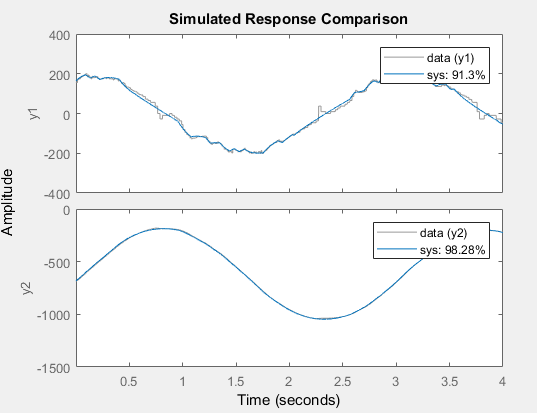
main\_data = iddata([w, y], u, Te);

sys = n4sid(main\_data);

resid(sys, data); figure

compare(data, sys);

Rezultatul il putem observa in *Figura 1.18:*



*Eroare de urmarire viteza unghiulara* : **91.3%**

*Eroare medie patratica viteza* unghiulara : 100 - 91.3 **= 8.7%**

*Eroare de urmarire pozitie unghiulara* : **98.28%**

*Eroare medie patratica pozitie unghiulara* : 100 - 98.28 **= 1.72%**

**1.4. Cod Matlab**

date=dx116;

t=double(date.X.Data')

u=double(date.Y(1,3).Data')

w=double(date.Y(1,2).Data')

y=double(date.Y(1,1).Data')

%%%%% Vizualizare date initiale

plot(t,[u\*200 w y]);title('Intrare')

xlabel('Timp(sec)');ylabel('u[factor de umplere], w[rad/sec], y[impulsuri]');

legend('u-Intrarea','w-Viteza unghiulara','y-Pozitia'); figure

% intrarea-u

subplot(311); plot(t,u,'b');

title('Intrarea u'); ylabel('u(PWM)'); xlabel('Timpul(s)');

% viteza unghiulara-w

subplot(312); plot(t,w,'g')

title('Viteza \omega'); ylabel('\omega(rad/s)'); xlabel('Timpul(s)');

% pozitia-y

subplot(313); plot(t,y,'r')

title('Pozitia \theta'); ylabel('\theta(rad)'); xlabel('Timpul(s)');

figure

%%%%% Identificare

i1=2126;

i2=4503;

iden=[i1:i2]';

subplot(211); plot(t(iden),[u(iden)\*200 w(iden) y(iden)]);

title('Portiune identificare');

%%%%% validare

v1=5711;

v2=7801;

valid=[v1:v2]';

subplot(212); plot(t(valid),[u(valid)\*200 w(valid) y(valid)]);

title('Portiune validare'); figure

%%%%% perioada esantionare

Te=t(2)-t(1);

%%%%% Date identificare si validare viteza unghiulara

%identificare viteza unghiulara

d\_iden\_w=iddata(w(iden),u(iden),Te);

%validare viteza unghiulara

d\_valid\_w=iddata(w(valid),u(valid),Te);

%%%%% Identificare si validare viteza unghiulara prin metoda ARX

% model viteza

w\_arx=arx(d\_iden\_w,[2 2 1]);

% autocorelatie si intercorelatie

resid(w\_arx,d\_valid\_w,15); figure

% comparare date de validare

compare(d\_valid\_w,w\_arx); figure

% functie de transfer in z (discret)

Hzw\_arx=tf(w\_arx.B,w\_arx.A,Te,'variable','z^-1');

% functie de transfer in s (continuu)

Hsw\_arx=d2c(Hzw\_arx,'zoh');

%%%%% Identificare si validare viteza unghiulara prin metoda IV4

% model viteza

w\_iv4=iv4(d\_iden\_w,[1 1 0]);

% autocorelatie si intercorelatie

resid(w\_iv4,d\_valid\_w,15); figure

% comparatie date de validare

compare(d\_valid\_w,w\_iv4); figure

% functie de transfer in z (discret)

Hzw\_iv4=tf(w\_iv4.B,w\_iv4.A,Te,'variable','z^-1');

% functie de transfer in s (continuu)

Hsw\_iv4=d2c(Hzw\_iv4,'zoh');

%%%%% Date identificare si validare pozitie unghiulara

% identificare pozitie unghiulara

d\_iden\_y=iddata(y(iden),w(iden),Te);

% validare pozitie unghiulara

d\_valid\_y=iddata(y(valid),w(valid),Te);

%%%%% Identificare si validare pozitia unghiulara prin metoda ARX

% model viteza

y\_arx=arx(d\_iden\_y,[2 2 1]);

% autocorelatie si intercorelatie

resid(y\_arx,d\_valid\_y,15); figure

% comparatie date validare

compare(d\_valid\_y,y\_arx); figure

%functie de transfer in z (discret)

Hzy\_arx=tf(y\_arx.B,y\_arx.A,Te,'variable','z^-1');

% functie de transfer in s (continuu)

Hsy\_arx=d2c(Hzy\_arx,'zoh');

%%%%% Identificare si validare pozitia unghiulara prin metoda IV4

% model viteza

y\_iv4=iv4(d\_iden\_y,[1 1 1]);

% autocorelatie si intercorelatie

resid(y\_iv4,d\_valid\_y,15); figure

% comparatie date validare

compare(d\_valid\_y,y\_iv4); figure

% functie de transfer in z (discret)

Hzy\_iv4=tf(y\_iv4.B,y\_iv4.A,Te,'variable','z^-1');

% functie de transfer in s (continuu)

Hsy\_iv4=d2c(Hzy\_iv4,'zoh');

%%%%% Simulare viteza unghiulara

wc\_arx=lsim(Hsw\_arx,u,t);

wc\_iv4=lsim(Hsw\_iv4,u,t);

plot(t,w,t,wc\_arx,t,wc\_iv4);shg;

title('Simularea functiilor pentru viteza'); xlabel('Timp(sec)')

ylabel('w[rad/sec]')

legend('Viteza masurata','Viteza calculata cu arx','Viteza calculata cu iv4');

figure

%%%%% Simulare pozitie unghiulara

yc\_arx=lsim(Hsy\_arx,u\*250,t);

yc\_iv4=lsim(Hsy\_iv4,u\*250,t);

plot(t,y,t,yc\_arx+y(1),t,yc\_iv4+y(1));shg;

title('Simularea functiilor pentru pozitie'); xlabel('Timp(sec)')

ylabel('impulsuri')

legend('Pozitia masurata','Pozitia calculata cu arx','Pozitia calculata cu iv4');

%%%%% Calcularea vitezei si pozitiei cu SS

main\_data = iddata([w, y], u, Te);

sys = n4sid(main\_data);

resid(sys, data); figure

compare(data, sys);