UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Facultatea de Automatică și Calculatoare

Identificarea Sistemelor
Identificarea unei axe acționate cu motor BLDC

Prof. Coordonator
Prof. Dr. Ing.
Petru Dobra

Ionut-Adrian
Vinatoru 30146

Introducere 3
Capitolul 1..5
1.1 Vizualizarea datelor experimentale utilizand mediul Matlab...........................*5*
1.2 Identificarea si validarea modelului...*6*
Capitolul 2..9
2.1 Determinarea datelor pentru identificare si validare utilizand decimarea acestora...*9*
2.2 Identificare parametrica a modelului INTRARE-VITEZA *9*
2.3 Identificare parametrica a modelului VITEZA-POZITIE.............................*14*
2.4 Validarea Modelului *18*
2.5 Spatiul starilor *18*

Introducere
Identificarea unei axe acționate cu motor BLDC

Obtinerea datelor experimentale

Figura 1: CNC acționata cu motor BLDC

Sistemul mecanic de poziționare ¸si sistemul de acționare cu motor BLDC pentru o axa este prezentat în Figura 2.

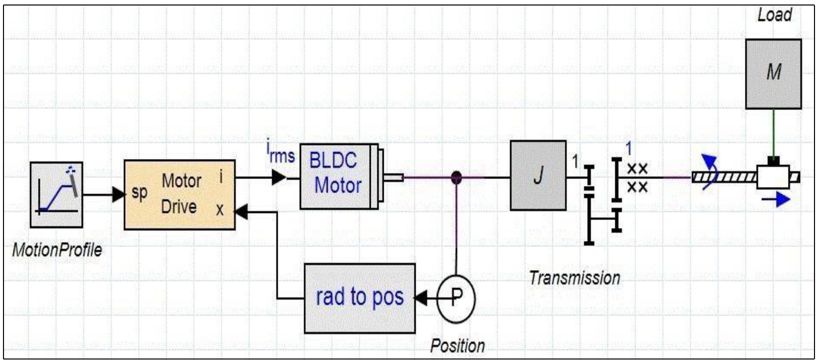

Figura 2: Modelul sistemului de acționare și poziționare al unei axe

Motorul este comandat cu ajutorul unui driver de putere de comandă în PWN. Viteza unghiulară și poziția se măsoară pe baza semnalelor provenite de la cei trei senzori Hall montați în statorul motorului. Folosind un sistem numeric de comanda se genereaza semnalele de comanda pentru motorul BLDC. Peste semnalul de comanda sinusoidal se suprapune un semnal de tip SPAB (Semnal pseudo-aleator binar).

Desfășurarea experimentului

1. Se alimentează ansamblul driver + motor BLDC cu Ua =24 V .
2. Se efectuează următorul experiment:

A1. Se generează un semnal de tip SPAB având caracteristicile corelat cu dinamica ansamblului „motor BLDC + axă”;

A2. Se vizualizează și se măsoară sincron intrarea și ieșirile, obținând datele experimentale: $[t_k,\ u_{k,}\ \omega_k,\ \theta_k]\ k\ = 1,2, \ldots$

Achiziția datelor de intrare-ieșire

Utilizând un sistem numeric de comandă se generează semnalele de comandă pentru motorul BLDC (SPAP + SP) ,și se achiziționează datele intrare-ieșire în vederea procesării ulterioare (comandă (factor de umplere), curent (i), viteza unghiulară (ω) ,si poziția unghiulara(θ)).

Procesarea datelor experimentale

Vizualizarea datelor experimentale utilizând mediul Matlab.
În funcție de datele experimentale obținute ($[t_k,\ u_{k,}\ \omega_k,\ \theta_k]\ k\ = 1,2, \ldots$)
se pot efectua următoarele operații: filtrare antidistorsiune de tip medie alunecătoare,eliminarea componentelor continue staționare sau cvasistaționare, scalarea intrărilor și ieșirilor.

Se vor determina funcțiile de transfer ale ansamblului „motor BLDC + axă” utilizând metodele de identificare parametrică (MCMMPR, MCMMPE, VI, MEP, etc.).

Validarea modelului

Validarea modelului determinat se face pe baza erorii de predicție reziduale și pe baza decorelării dintre observații și eroare de predicție.

De asemenea, se va compara răspunsul experimental cu răspunsul modelului la intrarea cu care a fost obținut răspunsul experimental. Să se calculeze eroarea medie pătratică normalizată:

$$\varepsilon_{MPN} = \frac{\|y - y^M\|}{\|y - \bar{y}\|}$$

Capitolul 1
Metoda neparametrica de identificare a sistemului

1.1 Vizualizarea datelor experimentale utilizand mediul Matlab

Importam fisierul ce contine datele pe care vom lucra. Urmeaza sa identificam si sa notam timpul, intrarea, viteza si pozitia. Calculam timpul de esantionare si afisam grafic separat datele de intrare suprapuse cu viteza, respectiv pozitia. Pentru a putea vizualiza mai bine datele am scalat intrarea, reprezentata de factorul de umplere, cu 100 (Figura 3).

```
load('vinatoru.mat');
t = double(vinatoru.X.Data)';
u = double(vinatoru.Y(3).Data)';
w = double(vinatoru.Y(2).Data)';
y = double(vinatoru.Y(1).Data)';

Te = t(5)-t(4);

figure,
hold on;
plot(t,[u*100, w ,y])

title('Comanda BLDC, viteza unghiulara si pozitia');
legend('Intrarea: u', 'Viteza unghiulara', 'Pozitia');
xlabel('t[s]');
ylabel('Amplitudinea (impulsuri)');
hold off;
```

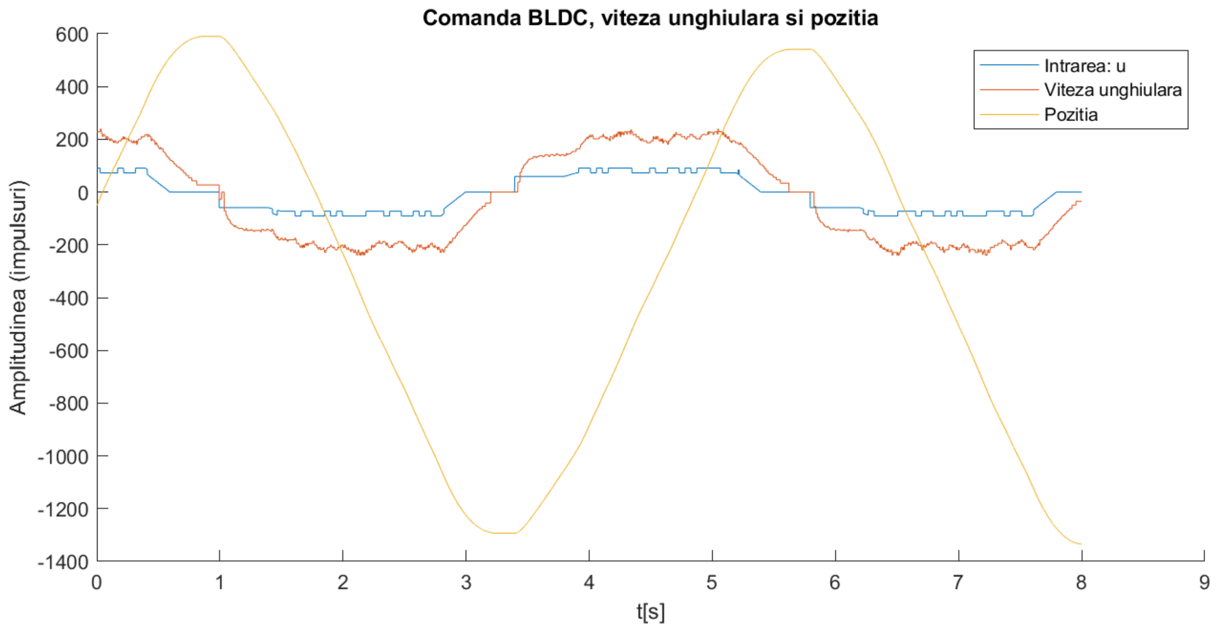

Figura 3

1.2 Identificarea si validarea modelului

Printre cele mai comune metode neparametrice de identificare se regasestesi identificarea unui sistem pe baza raspunsului la semnalul de tip treapta. Procesele fizice modelate printr-un sitem de ordinul I LTI cu timp mort au urmatoarea functie de transfer:

$$H(s) = \frac{K}{Ts+1}e^{-\tau_m s}$$

Parametrii care trebuie identificati sunt: factorul de proportionalitate K, constanta de timp T[sec] si timpul mort τm[sec].Factorul de proportionalitate este dat de raportul dintre iesire si intrare in regim stationar.

$$K = \frac{\overline{y_{st}} - \overline{y_0}}{\overline{u_{st}} - \overline{u_0}}$$

Pentru a identifica valoarea constantei de timp (T), se citeste variatia
de timp de la momentul declansarii treptei (neideale) pana la 63% din valoarea diferentei dintre cele doua paliere stationare ale iesirii.

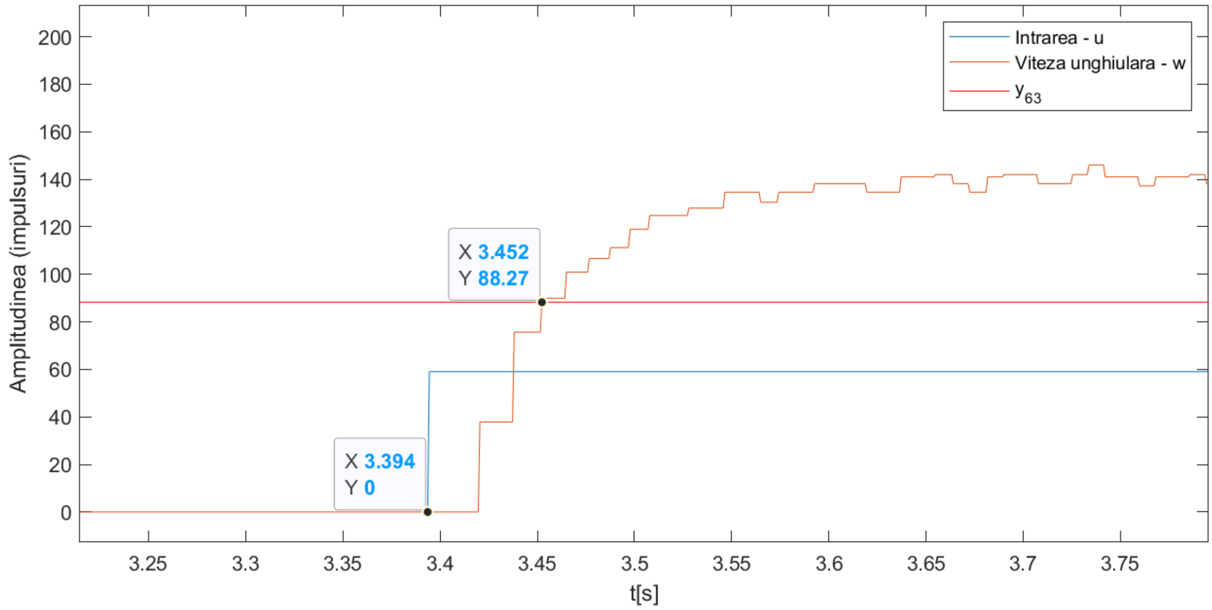

Figura 4: Alegerea indecsilor pentru determinarea constantei de timp

```
i1 = 4330;
i2 = 4415;
yst = mean(w(i1:i2));
ust = mean(u(i1:i2));


K = yst/ust % Calculul factorului de proportionalitate


Tm = t(4072)-t(4041) % Calculul timpului mort


y63 = 0.63*yst;
figure,
plot(t,u*100, t,w, t,y63*ones(1,length(t)),'r');
legend('Intrarea - u','Viteza unghiulara - w','y_6_3')


i5 = 4041;
i6 = 4111;
T = t(i6)-t(i5); % Calculul constantei de timp


Hc = tf(K,[T 1],'iodelay',Tm) %Functia de transfer rezultata
```

Functia de transfer rezultata:

$$H(s) = \frac{237.3}{0.0588s+1} e^{-0.026s}$$

Pentru simularea raspunsului la intrarea de tip treapta din conditii initiale nenule este necesar modelul de tip spatiul starilor. Pe baza formei canonice observabile dedusa din functia de transfer, se obtin matricele:

```
A = -1/T;
B = K/T;
C = 1;
D = 0;
```

Vom crea o noua intrare u_new la care vom adauga timpul mort, si vom simula.Validarea modelului se va face pe baza erorii medii patratice normalizate.

$$\epsilon_{MPN} = \frac{\|Y - Y_M\|}{\|Y - \overline{Y}\|}$$

```
N = round(Tm/(t(12)-t(11)));
u_new = [u(1)*ones(N,1); u(1:length(u)-N)];

sys = ss(A, B, C, D);
ysim = lsim(sys, u_new, t, w(1));
plot(t, [ysim u*100 w])
legend('ysim','Intrarea: u','Viteza unghiulara: w')

empn = norm(w-ysim)/norm(w-mean(w))
```

ϵ_{MPN}= 0.1314(13.14%)

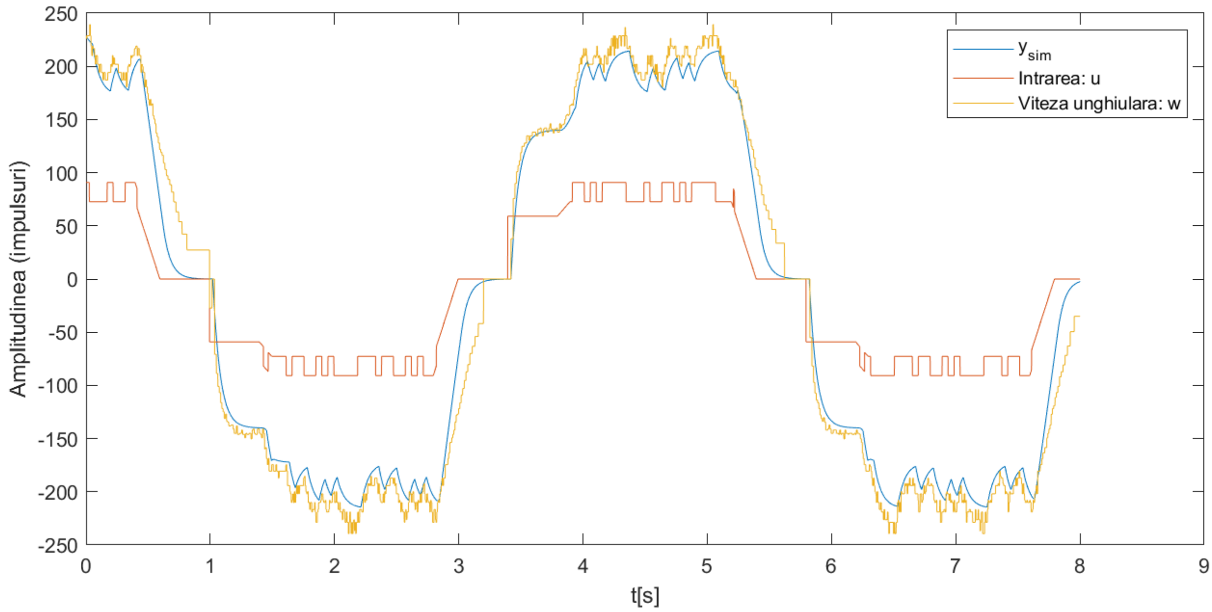

Figura 5: Modelul simulat suprapus pe iesirea sistemului

Capitolul 2
Metode parametrice de identificare a sistemului

2.1 Determinarea datelor pentru identificare si validare utilizand decimarea acestora

Perioada cu care se realizează achiziția datelor poate fi prea mică în raport cu capacitatea elementelor de măsură de a furniza noi măsurători. Astfel, sistemul va introduce aceleași măsurători de mai multe ori și poate să apară fenomenul de supraeșantionare.

Pentru a elimina acest lucru, este de dorit o decimarea datelor astfel încât valoarea maximă să se regăsească o singură dată. De aceea se determină numărul de tacți pentru care semnalul de ieșire are valoare maximă N și apoi se consideră toate semnalele cu care vom lucra doar din N în N măsurători.

```
i1 = 1670;
i2 = 3374;
i3 = 4685;
i4 = 6100;
Te = t(2)-t(1); % Te=0.00084


N = 7;
Te = Te*N; % Te=0.0059
t_dc = t(1:N:end);
u_dc = u(1:N:end);
w_dc = w(1:N:end);
y_dc = y(1:N:end);

i1 = floor(i1/N);
i2 = floor(i2/N);
i3 = floor(i3/N);
i4 = floor(i4/N);
```

2.2 Identificare parametrica a modelului INTRARE-VITEZA

A) Metoda celor mai mici patrate extinsa (ARMAX)

Identificarea consta in estimarea coeficientilor polinoamelor A, B si C. Parametrii de structura ai sistemului sunt: nA = degA, nB = degB, nC = deg C, respectiv nd = numarul tactilor intarziere. Modelul discret de tip *proces+perturbatie* corespunzator metodei ARMAX este:

$$A(z^{-1})Y(z) = z^{-d}B(z^{-1})U(z) + C(z^{-1})\,E(z)$$

unde:

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-nA}$$

$$B(z^{-1}) = b_1 z^{-1} + \cdots + b_n z^{-nB}$$

$$C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2} + \cdots + c_n z^{-nC}$$

Pentru identificarea modelului vom folosi:

Model = armax(data_id,[n_A,n_B, n_C ,n_d])

n_A – ordinul polinomului A
n_B – ordinul polinomului B+1
n_C – ordinul polinomului C
n_d – timpul mort (număr tacți întârziere)

```
%Crearea obiectelor de tip iddata ce urmeaza sa fie trimise ca parametrii in
%cadrul functiilor de identificare si validare a modelului intrare-viteza
data_id = iddata(w_id,u_id,Te);
data_vd = iddata(w_vd,u_vd,Te);
```

```
m_armax = armax(data_id,[1 1 1 1]);
```

Validarea modelului prin autocorelatie

```
resid(m_armax,data_vd,5) %validarea se va realiza pe portiunea de date
%specifiva acesteia (data_vd)
```

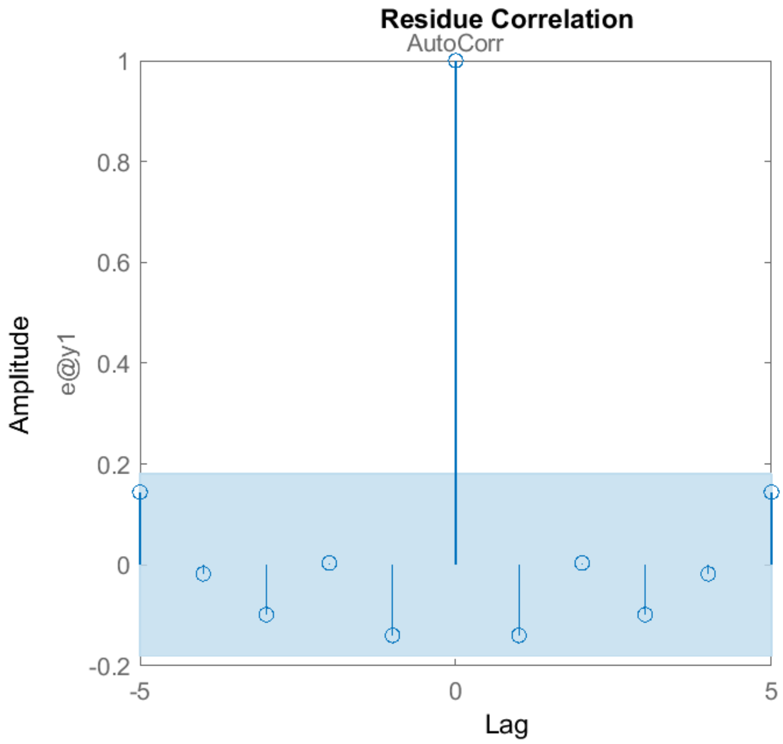

Figura 6

- Se poate observa din resid faptul că testul de validare prin autocorelație este trecut, deoarece banda de încredere nu este depășită.

```
compare(m_armax,data_vd)
```

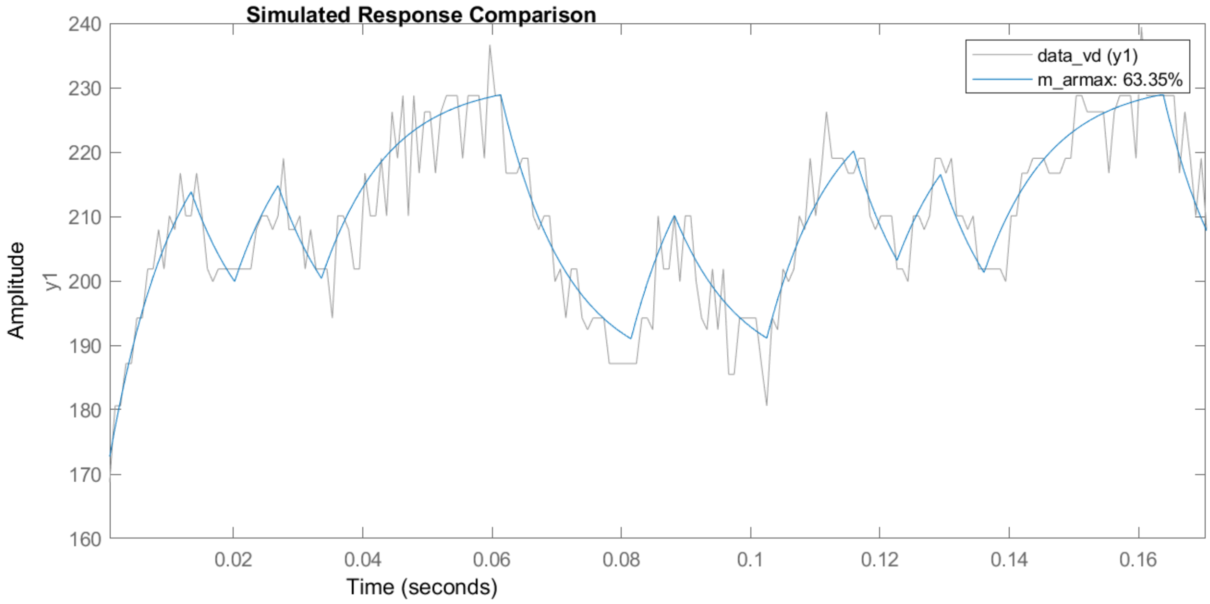

Figura 7

- Din compare: FIT=63.35% => Empn=1-63.35% = 36.65%

- Funcția de transfer Z se află direct din model (numărătorul : m_armax.B, numitorul : m_armax.A)

```
Hd_armax=tf(m_armax.B,m_armax.A,Te,'variable','z^-1')
```

$$Hd\_armax = \frac{19.91z^{-1}}{1-0.9217z^{-1}}$$

- Funcția de transfer încontinuu (cu funcția discrete to continous):

```
Hc_armax=zpk(minreal(d2c(Hd_armax,'zoh')))
```

$$Hc\_armax = \frac{2.468*10\text{^}4}{s+97.1}$$

B) Metoda erorii de iesire (OE)

Ideea de bază a acestei metode se bazează pe observația că, în absența perturbațiilor,ieșirea predictată prin predictorul CMMPR, $\hat{y}= (t + 1)$ tinde către y (t + 1).

În aceste condiții, se poate înlocui $y(t)$ în ecuația predictorului cu $\hat{y}(t)$ (predicție a posteriori).Interesul acestei modificări apare clar în prezența perturbațiilor.

Modelul de tip 'proces + perturbație':

$$\boldsymbol{y}(\boldsymbol{z}) = \frac{\boldsymbol{z}^{-\boldsymbol{n}_{\boldsymbol{d}}}\boldsymbol{B}\cdot(\boldsymbol{z}^{-1})}{\boldsymbol{F}(\boldsymbol{z}^{-1})} \cdot \boldsymbol{U}(\boldsymbol{z}) + \boldsymbol{E}(\boldsymbol{z})$$

unde,

$$\mathbf{F}\,(\boldsymbol{z}^{-1}) = \mathbf{1} + f_1\boldsymbol{z}^{-1} + f_2\boldsymbol{z}^{-2} + \cdots + f_{n_F}\boldsymbol{z}^{-\boldsymbol{n}_F}$$

$$\boldsymbol{B}(\boldsymbol{z}^{-1}) = \boldsymbol{b}_1\boldsymbol{z}^{-1} + \cdots + b_{n_B}\boldsymbol{z}^{-\boldsymbol{n}\boldsymbol{B}}$$

Pentru identificarea modelului vom folosi :

Model = oe(data_id,[n_F,n_B,n_d])

n_F – ordinul polinomului F

n_B – ordinul polinomului B+1

n_d – timpul mort (număr tacți întârziere)

```
m_oe = oe(data_id,[1 1 1]);
```

Validarea modelului prin intercorelatie

```
resid(m_oe,data_vd) %validarea se va realiza pe portiunea de date specifiva
                     acesteia (data_vd)
```

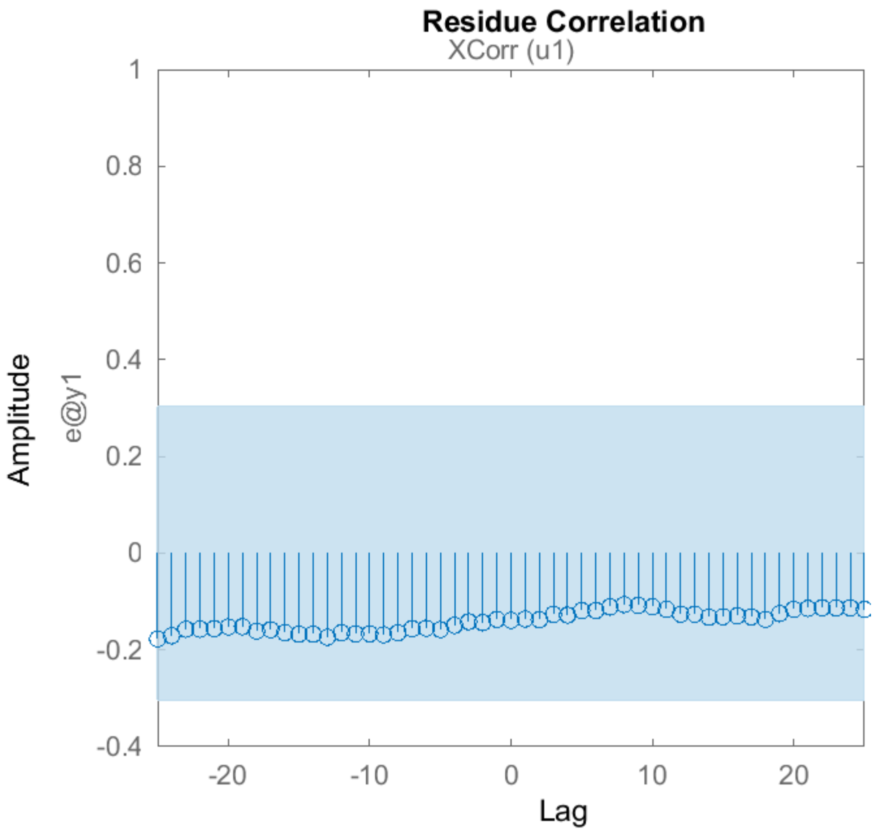

Figura 8

- Se poate observa din resid faptul că testul de validare prin intercorelație este trecut, deoarece banda de încredere nu este depășită.

```
compare(data_vd,m_oe)
```

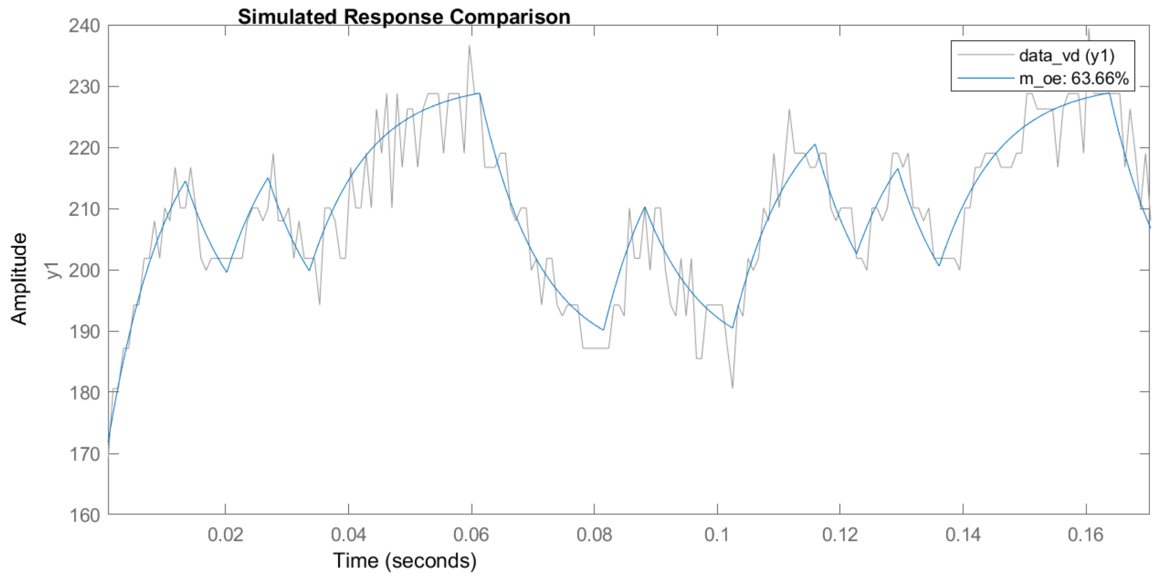

Figura 9

- Din compare: FIT=63.66% => Empn=1-63.66% = 36.34%

- Funcția de transfer Z se află direct din model (numărătorul : m_oe.B, numitorul : m_oe.F)

```
Hd_oe=tf(m_oe.B,m_oe.F,dt,'variable','z^-1')
```

$$Hd\_oe = \frac{20.97z^{-1}}{1-0.9174z^{-1}}$$

- Funcția de transfer încontinuu (cu funcția discrete to continous):

```
Hc_oe=zpk(minreal(d2c(Hd_oe,'zoh')))
```

$$\text{Hc\_oe} = \frac{2.605*10\text{^}4}{s+102.6}$$

2.3 Identificare parametrica a modelului VITEZA-POZITIE

A) Metoda celor mai mici patrate extinsa (ARMAX)

Pentru identificarea modelului vom folosi:

Model = armax(data_id,[n_A,n_B, n_C ,n_d])

n_A – ordinul polinomului A
n_B – ordinul polinomului B+1
n_C – ordinul polinomului C
n_d – timpul mort (număr tacți întârziere)

```
%Crearea obiectelor de tip iddata ce urmeaza sa fie trimise ca parametrii in
%cadrul functiilor de identificare si validare a modelului viteza-pozitie
data_id = iddata(y_id,w_id,Te);
data_vd = iddata(y_vd,w_vd,Te);
```

```
m_armax = armax(data_id,[1 1 1 0]);
```

Validarea modelului prin autocorelatie

```
resid(m_armax,data_vd)
```

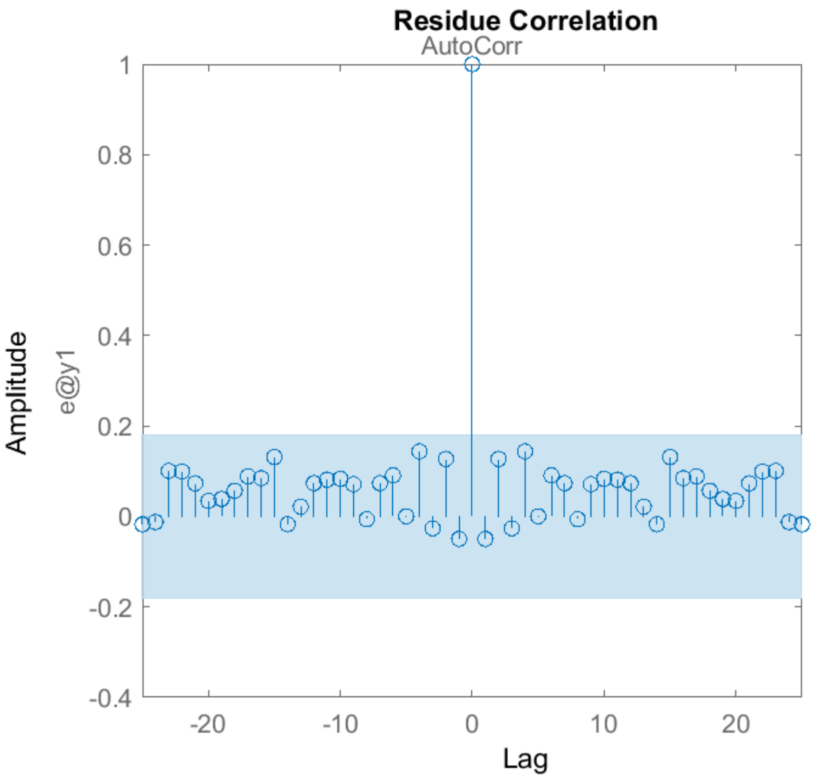

Figura 10

- Se poate observa din resid faptul că testul de validare prin autocorelație este trecut, deoarece banda de încredere nu este depășită.

```
compare(m_armax,data_vd)
```

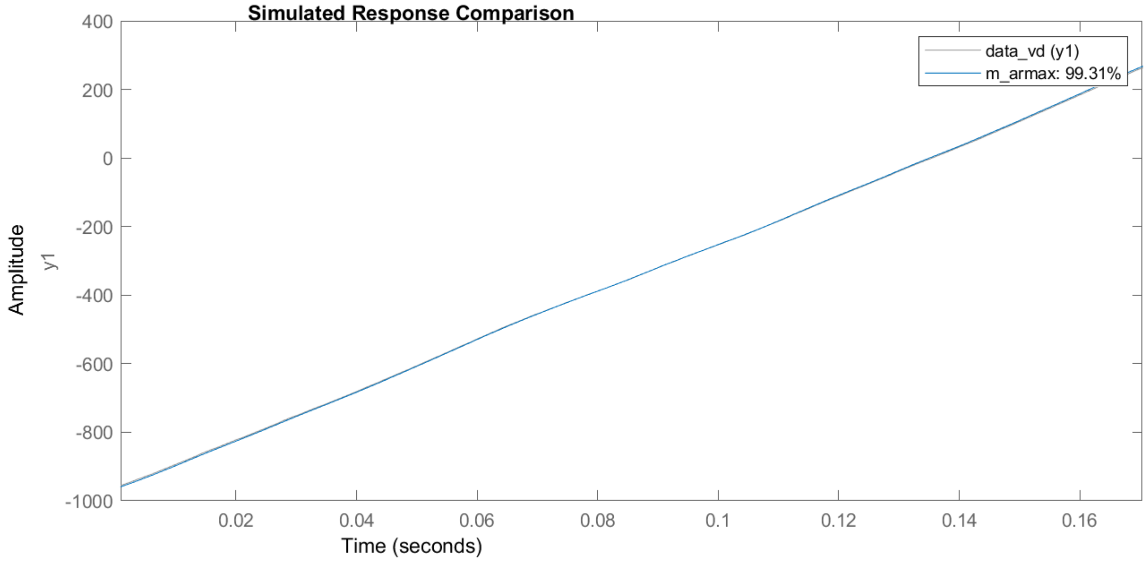

Figura 11

- Din compare: FIT=99.31% => Empn=1-99.31% = 0.69%

- Funcția de transfer Z se află direct din model (numărătorul : m_armax.B, numitorul : m_armax.A)

```
Hd_armax=tf(m_armax.B,m_armax.A,Te,'variable','z^-1')
```

$$Hd\_armax = \frac{0.02893}{1-z^{-1}}$$

Deoarece nu avem timp mort (de la viteză la poziție e un proces fizic, fără intervenție de la calculator) nu putem calcula funcția de transfer în continuu prin comanda d2c. Astfel se va folosi metoda invarianței răspunsului la impuls.

$$Hc\_armax = \frac{0.02893}{Te}\frac{1}{s}$$

Te=0.0059

$$Hc\_armax = \frac{4.9201}{s} = \frac{1}{0.2032s}$$

B) <u>Metoda erorii de iesire (OE)</u>

Modelul de tip 'proces + perturbație':

$$y(z) = \frac{z^{-n_d}B\cdot(z^{-1})}{F(z^{-1})}\cdot U(z) + E(z)$$

unde,

$$F\,(z^{-1}) = 1 + f_1 z^{-1} + f_2 z^{-2} + \cdots + f_{n_F} z^{-n_F}$$

$$B(z^{-1}) = b_1 z^{-1} + \cdots + b_{n_B} z^{-nB}$$

Pentru identificarea modelului vom folosi :

Model = oe(data_id,[n_F,n_B,n_d])

n_F – ordinul polinomului F

n_B – ordinul polinomului B+1

n_d – timpul mort (număr tacți întârziere)

```
m_oe = oe(data_id,[1 1 0]);
```

Validarea modelului prin intercorelatie

```
resid(data_vd,m_oe,10)
```

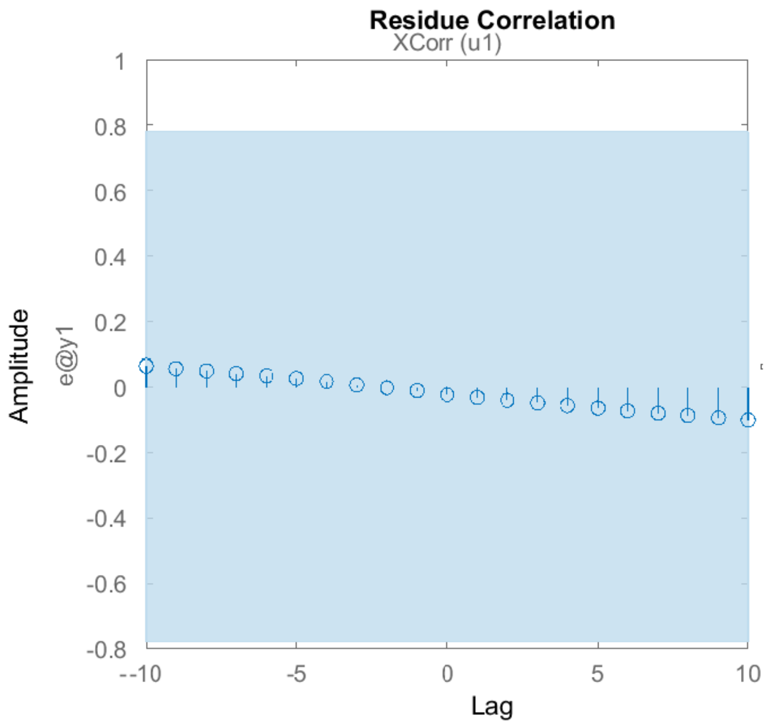

Figura 12

- Se poate observa din resid faptul că testul de validare prin intercorelație este trecut, deoarece banda de încredere nu este depășită.

```
compare(data_vd,m_oe)
```

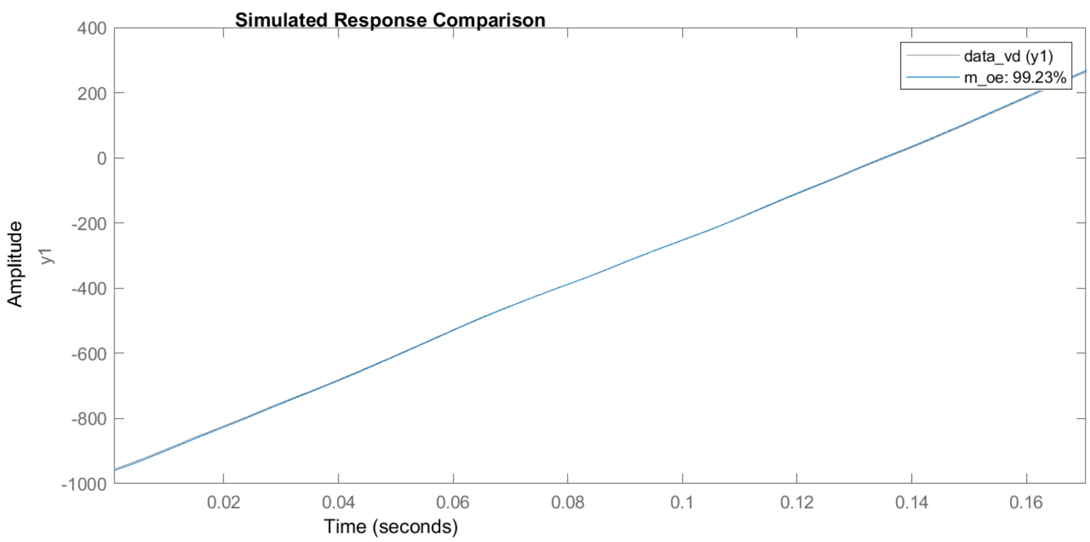

Figura 13

- Din compare: FIT=99.23% => Empn=1-99.23% = 0.77%

- Funcția de transfer Z se află direct din model (numărătorul : Mw_oe.B, numitorul : Mw_oe.F)

$$Hd\_oe = \frac{0.02894}{1-z^{-1}}$$

$$=> \ Hc\_oe = \frac{4.9218}{s} = \frac{1}{0.2032s}$$

2.4 Validarea Modelului

a) Validarea modelului prin autocorelație(testul de albire a erorii de predicție):
- Pentru viteză:
  FIT= 63.35% => Empn=1-63.35%=36.65%
- Pentru poziție:
  FIT= 99.31% => Empn=1-99.31%=0.69%

b) Validarea modelului prin intercorelație(testul de decorelare)
- Pentru viteză:
  FIT= 63.66% => Empn=1-63.66%=36.34%
- Pentru poziție:
  FIT= 99.23% => Empn=1-99.23%=0.77%

2.5 Spatiul starilor

- **Autocorelație:**

Funcția de transfer dintre intrarea u și viteza ω:

$$Hc\_armax\_v = \frac{k_m}{T_m \cdot s+1} = \frac{2.468*10^4}{s+97.1} = \frac{254.2177}{0.0721s+1}$$

Funcția de transfer dintre viteza ω și poziția y:

$$Hc\_armax\_p = \frac{1}{T_i \cdot s} = \frac{\mathbf{1}}{\mathbf{0.2032s}}$$

$$\Rightarrow \quad H1 = \frac{\mathbf{254.2177}}{\mathbf{0.0721s+1}} \frac{\mathbf{1}}{\mathbf{0.2032s}}$$

Tm=0.0721;
Km=254.2177;
Ti= 0.2032

Codul in Matlab:

```
Tm=0.0721;
Km=254.2177;
Ti= 0.2032
%FCO
A = [-1/Tm 0 ; 1/Ti 0];
B = [Km/Tm; 0];
C = [1 0; 0 1];
D = [0 ; 0];
sys = ss( A , B , C , D );
ysim=lsim(sys,u,t,[w(1) y(1)]);
figure,
plot(t,[ysim w y]);
grid on;
title("Identificarea sistemului prin autocorelatie cu spatiul starilor");
xlabel('t[s]');
ylabel('viteza unghiulara si pozitia ungiulara');
legend('viteza identficata','pozitia identificata','viteza','pozitia');

E=norm(w-ysim(:,1))/norm(w-mean(w))*100
E=norm(y-ysim(:,2))/norm(y-mean(y))*100
```

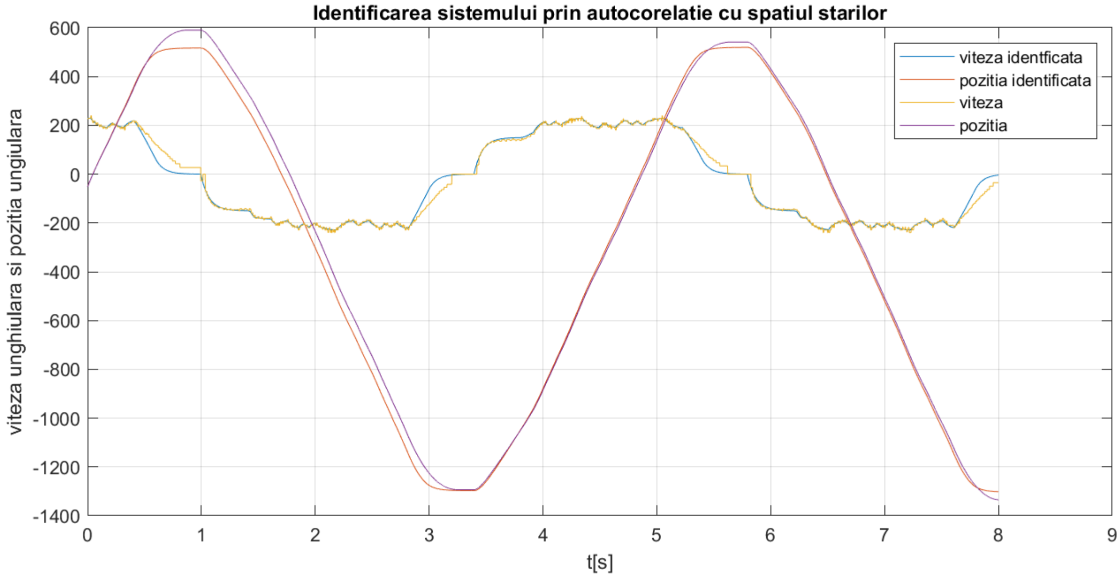

Figura 14

In urma simularii modelului obtinut prin spatiul starilor pe intreg setul de date am obtinut urmatoarele erori:

$$\epsilon_{MPN\_w} = 11.86\%$$
$$\epsilon_{MPN\_p} = 6.04\%$$

- **Intercorelație:**
  Funcția de transfer dintre intrarea u și viteza ω:

$$\mathbf{Hc\_oe\_v} = \frac{\mathbf{k_m}}{\mathbf{T_m \cdot s+1}} = \frac{\mathbf{2.605*10^\wedge 4}}{\mathbf{s+102.6}} = \frac{\mathbf{253.8986}}{\mathbf{0.0097s+1}}$$

  Funcția de transfer dintre viteza ω și poziția y:

$$\boldsymbol{Hc\_oe\_p} = \frac{1}{T_i \cdot s} = \frac{\mathbf{1}}{\mathbf{0.2032s}}$$

⇨ $$\boldsymbol{H2} = \frac{\mathbf{253.8986}}{\mathbf{0.0097s+1}} \frac{\mathbf{1}}{\mathbf{0.2032s}}$$

unde,

$\boldsymbol{k_m}$ **= 253.8986**

$\mathbf{T_m}$ **= 0.0097**

$\mathbf{T_i}$ **= 0.2032**

Codul in Matlab:

```
Tm=0.0097;
Km=253.8986;
Ti= 0.2032;
%FCO
A = [-1/Tm 0 ; 1/Ti 0];
B = [Km/Tm; 0];
C = [1 0; 0 1];
D = [0 ; 0];
sys = ss( A , B , C , D );
ysim=lsim(sys,u,t,[w(1) y(1)]);
figure,
plot(t,[ysim w y]);
grid on;
title("Identificarea sistemului prin intercorelatie");
xlabel('t[s]');
ylabel('viteza unghiulara si pozitia ungiulara');
legend('viteza identficata','pozitia identificata','viteza','pozitia');

E=norm(w-ysim(:,1))/norm(w-mean(w))*100
E=norm(y-ysim(:,2))/norm(y-mean(y))*100
```

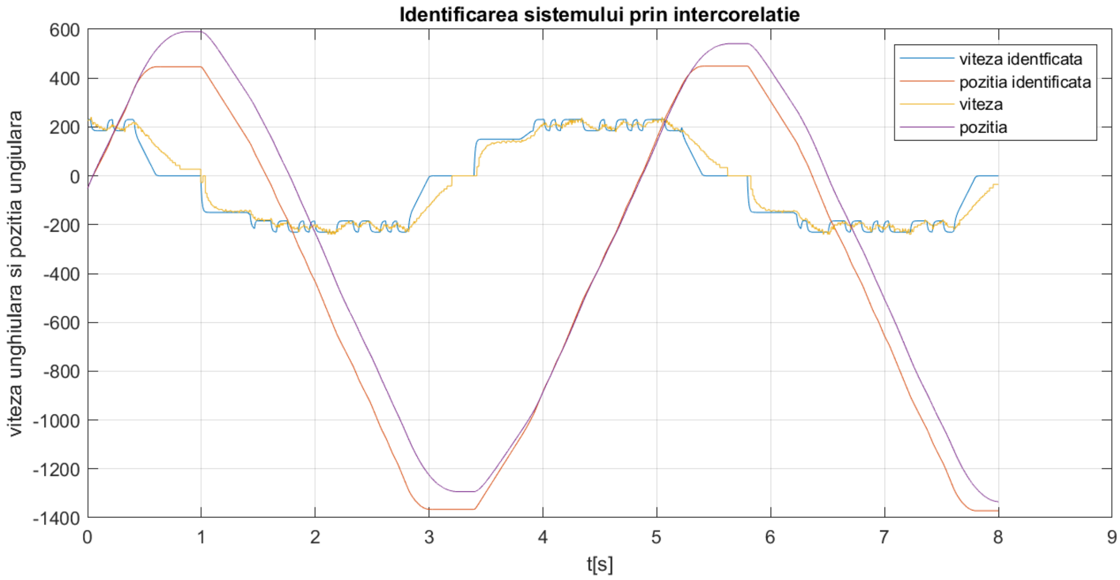

Figura 15

In urma simularii modelului obtinut prin spatiul starilor pe intreg setul de date am obtinut urmatoarele erori:

$\epsilon_{MPN\_w} = 23.58\%$

$\epsilon_{MPN\_p} = 19.88\%$