UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Proiect la Identificarea Sistemelor

Identificarea unei axe actionate cu motor BLDC

Profesor coordonator:
Prof.univ.dr.ing. Petru Dobra
Student:
Voina Mihai Sebastian
Grupa 30131/2, an III

Cuprins

Capitolul 1:
Vizualizarea datelor

In spatiul de lucru Matlab incarcam fisierul primit cu datele experimentale ("voina.mat"), fisier in care se afla doua campuri X' si Y'.Vectorul X intervalul de timp pe care s a facut achizitia datelor, vectorul Y contine valorile intrarii(u-factorul de umplere al PWM), valorile vitezei unghiulare omega (ω-rad/s) si valorile pozitiei unghiulare theta(Ө-impulsuri).

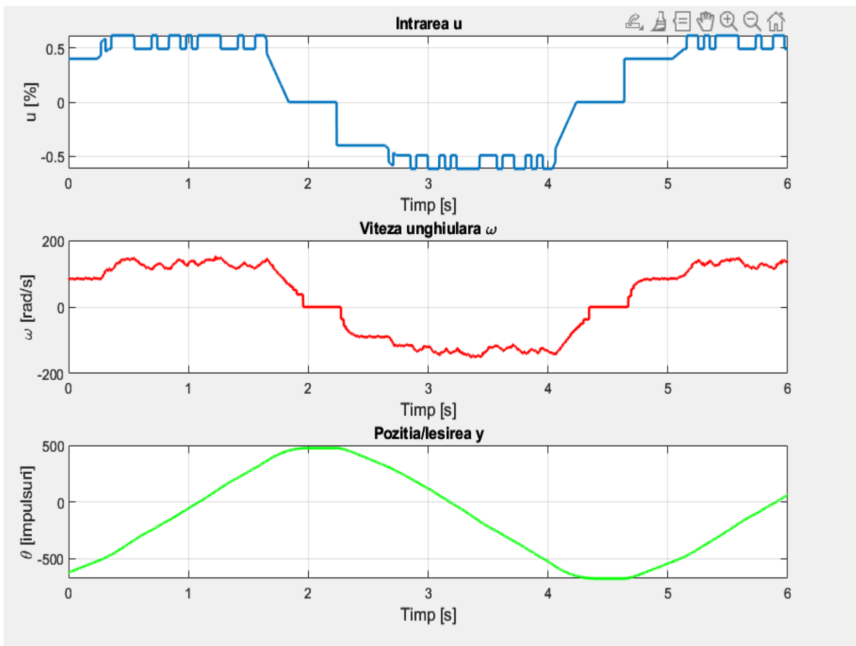


Fig 1: Date Experimentale luate individual

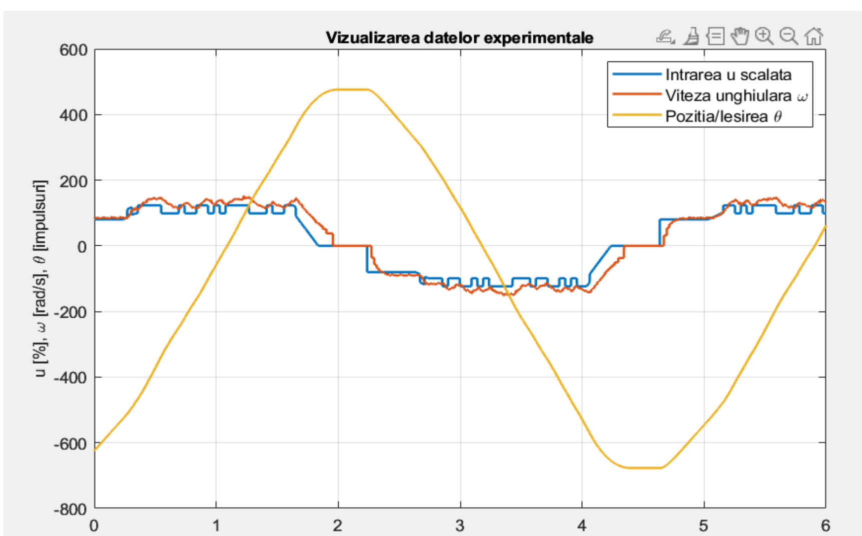


Fig2 Datele experimentale suprapuse cu u scalat

Capitol2:

Partea 1 :Identificarea sistemului prin metode neparametrice

1.1 Identificarea modelului intrare – viteza folosind metoda logaritmarilor succesive

Pentru identificarea modelului matematic al vitezei unghiulare vom cauta o secventa unde avem o intrare de tip treapta.Secventa identificata o avem in figura urmatoare:

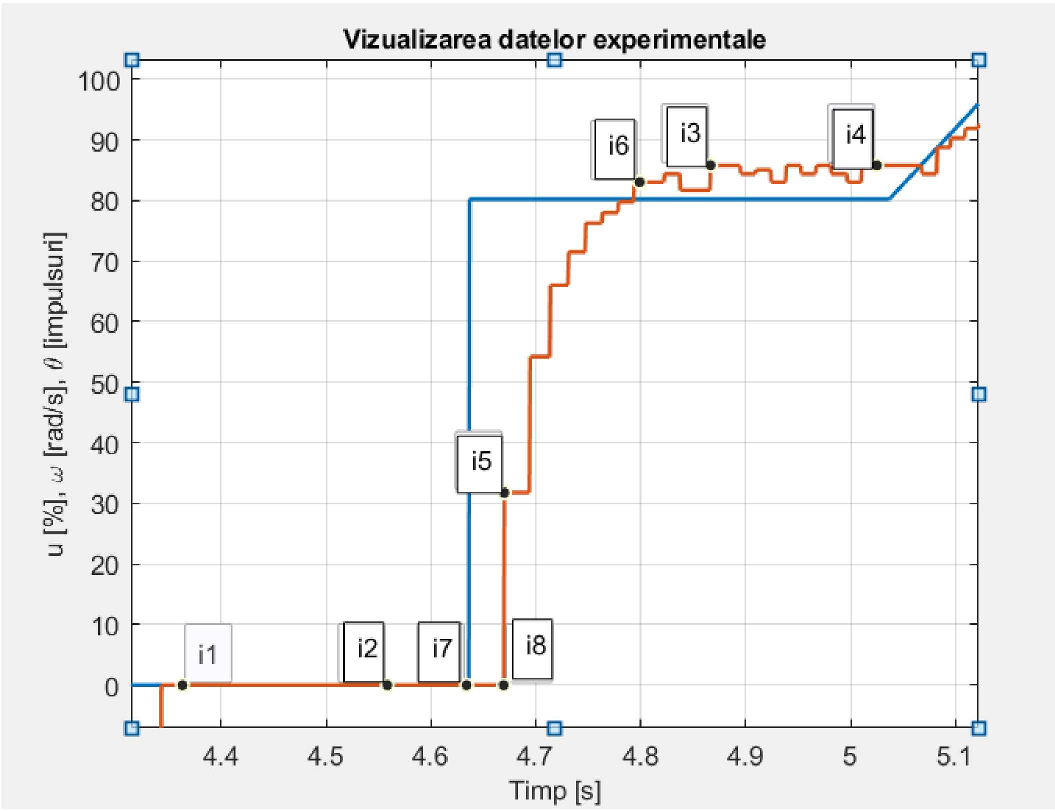


Fig 3 Treapta aleasa pentru identificarea modelului mathematic intrare-viteza

Modelul matematic pe care noi dorim sa il obtinem este sistem LTI(linear time invariant) de ordin I care are functia de transfer de forma: $\boldsymbol{H(s)} = \frac{\mathbf{K}}{\mathbf{Ts+1}}$, unde K este factorul de proportionalitate si T este constanta de timp.

1.2Calcularea factorului de proportionalitate K

Factorul de proportionalitate este dat de raportul dintre viteza și intrare in regim stationar. Datorita zgomotului de masura este cenesara alegerea cu atentie a valorilor initiale si statioare:

$$\boldsymbol{K} = \frac{\mathbf{wst - w0}}{\mathbf{ust - u0}}$$

Unde w stationar este valoarea medie statioara pe portiunea delimitata de indecsii i3 si i4 si w0 este valoarea medie initiala pe portiunea delimitata de indexi i1 si i2. Analog pentru u stationar mediu si u0 mediu.

```
i1 = 6109;
i2 = 6412;
i3 = 6762;
i4 = 6983;
% portiunile de valori medii de pe grafic
w0 = mean(w(i1:i2));
wst = mean(w(i3:i4));
u0 = mean(u(i1:i2));
ust = mean(u(i3:i4));
K = (wst-w0)/(ust-u0) % K =211.5206
```

1.3Calcularea constantei de timp T,timpului mort si determinarea functiei de transfer

Pentru determinarea constatnei de timp T vom folosi metoda logaritmarilor succesive.
Plecam de la relatia: $w(t) = w0 + (wst - w0) * (1 - e^{-\frac{t}{T}})$
Relatia respectiva se va logaritma de unde rezulta relatia: $\ln(|w(tk) - yst|) = c - \frac{1}{T} * tk$

Unde tk sunt esantioane de timp alese de la momentul declansarii treptei pana la intrarea in regim statioar cu ajutorul parapentrilor i5 si i6 din figura 2.1.

```
% timpu de urcare

i5 = 6486;
i6 = 6670;
tk = t(i5:i6)';   Tm = t(i8) - t(i7); % Tm = 0.0331s = 33.1 ms
```

Se considera parametrii α si β ai regresiei liniare unde α este coeficientul lui x si β este termenul liber al ecuatiei dreptei w = α *x + β. Pentru determinarea acestor parametrii se va rezovla sistemul de ecuatii:

$$\binom{\alpha}{\beta} = \begin{pmatrix} \sum_{k=1}^{N} tk^2 & \sum_{k=1}^{N} tk \\ \sum_{k=1}^{N} tk & N \end{pmatrix} \begin{pmatrix} \sum_{k=1}^{N} xk * tk \\ \sum_{k=1}^{N} xk \end{pmatrix}$$

Unde xk este dreapta pentru care se face regresia (logaritm(w stationar mediu – w intre indecsii i5 si i6)).

```
xk = log(wst-w(i5:i6));  % dreapta pentru care se face regresia

Areg =[sum(tk.^2) sum(tk);
      sum(tk)   length(tk)];

breg = [sum(tk.*xk) ; sum(xk)];

sol = Areg\breg; % inv(areg)*breg

T = -1/sol(1) % T = 0.0411s = 41.1 ms
```

Observam ca T = 1/α, α determinat mai sus rezulta T = 41.1ms

Pentru a determina timpul mort luam indecsii intre momentul declansarii treptei pana la momentul in care viteza se modifica adica i7, i8 din fig 2.1

In urma identificarii parametrilor T,K,Tm modelul matematic de tip functie de transfer este

$$\mathbf{H = exp(-0.0331 \cdot s) \cdot \frac{211.5}{0.04107\, s + 1}}$$

```
H = tf(K, [T 1], 'iodelay', Tm)
```

1.4 Validarea modelului si calculul erorii medii patratice normalizata

Pentru simularea modelului matematic obtinut folosim modelul spatiul starilor(pentru a evita problema conditilor initiale nule introduse de functia de transfer) si calculam cu ajutorul vitezei simulate eroarea medie minima patratica(12.41%)

```
% etapa de verificare ( trecerea in spatiul starilor )
N = round(Tm/(t(2)-t(1))); % cate esantioane de timp are tm
uN = [u(1)*ones(N,1); u(1:end-N)];
A = -1/T;
B = K/T;
C = 1;
D = 0;

figure
plot(t,200*u);
wsim2 = lsim(A,B,C,D,uN,t,w(1));
hold on
plot(t,[w,wsim2]), grid;
legend('Intrarea u scalata', 'Viteza unghiulara \omega', 'Viteza unghiulara simulata \omegasim');
xlabel('Timp [s]');
ylabel('u [%], \omega [rad/s], \omegasim [rad/s]');

eMPN = norm(w-wsim2)/norm(w-mean(w)); % eraore medie partatica normalizata  = 12.41%
```

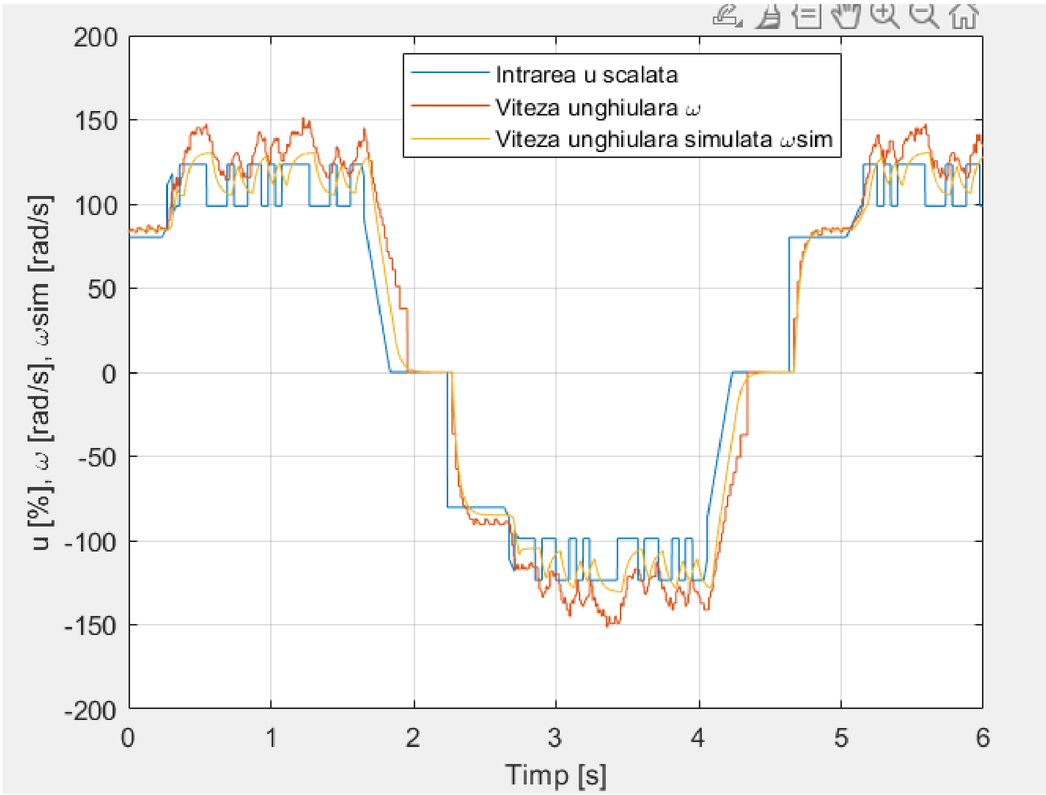


Fig4 Viteza unghiulara simulata

1.5 Testul de albire a erorii de predictie

Se considera secventa erorilor de predictive $\varepsilon[n]$ unde n ia valori intre 1,N

Unde N reprezinta lungimea valorilor pentru viteza.

N = i4-i1(parametrii din fig 1.1);

e = w(i1:i4)-wsim2(i1:i4); *% erorile dintre viteza masurata si cea simulata*

si se calculeaza autocorelatia acestei secvente din:

$$R[i] = \frac{1}{N}\sum_{k=1}^{N} \varepsilon[k]\varepsilon[k-i].$$

*R(i) = sum([e(1:N-i)]'*e((i+1):N))/N;*

Calculam apoi intr o bluca for autocorelatiile normalizate astfel:

$RN[i] = \frac{R[i]}{R[0]}$ *% RN(i) = R(i)/R0;*

unde R0 reprezinta eroare medie patratica calculate ca si raport dintre suma patratelor erorii dintre viteza masurata si cea simulata prezentata mai sus si lungimea valorilor pentru viteza

% R0 = sum(e.^2)/N;

In bucla se verifica numarul de erori care nu se afla in banda de incredere data de relatia $\frac{2.17}{\sqrt{N}}$ si observam ca acest numar este N1 = 116.De unde rezulta ca numarul de erori care se afla in banda de incredere este egal cu N-N1 = 758 in cazul meu

if abs(RN(i)) <= (2.17/sqrt(N)) *% 2.17/sqrt(N) = 0.0743*

Se verifica manual acest lucru de unde se observa ca intr-adevar doar dupa 116 erori calculate, eroarea de predictie formeaza o secventa de tip zgomot alb :

```
R1 = sum(e(1:N-1).*e(2:N))/N;
Rn1 = R1/R0; % Rn1 = 0.9254
R2 = sum(e(1:N-2).*e(3:N))/N;
Rn2 = R2/R0; % Rn2 = 0.8543
R116 = sum(e(1:N-115).*e(116:N))/N;
Rn116 = R116/R0; % Rn116 = 0.0775
R117 = sum(e(1:N-116).*e(117:N))/N;
Rn117 = R117/R0; % Rn117 = 0.0732 < 2.17/sqrt(N) = 0.0743
```

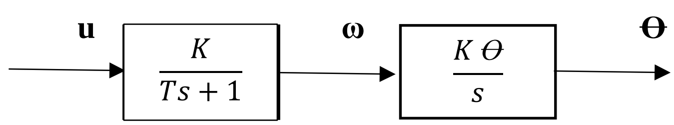
In concluzie modelul indentificat la treapta nu este validat prin autocorelatie din cauza rezultatelor care nu se incadreaza in banda de incredere, dar acest lucru este de asteptat din cauza faptului ca nu avem date consistente fiind un semnal sarac cu conditii initiale nule deci primele erori vor depinde unele de altele.

2Identificarea neparametrica de la viteza la pozitie

2.1 Calculul factorului de integrare Ki si determinarea functiei de transfer

Deoarece pozitia este integrala vitezei, regimul stationar al vitezei se translateaza ca si o dreapta oblica.

Schema bloc a sistemului:



avem noive de analiza blocului 2:

$$\Theta(t) = \int_{t0}^{t1} ki * w(t)dt =$$

$$= \int_{t0}^{t1} ki * wst\, dt = ki * wst * t|_{t0}^{t1} + c = \ ki * wst(t1 - t0) + c => \Theta(t) - \ \Theta(t0) =$$

$$= \ ki * wst * \Delta$$

$$\boxed{=> \ \boldsymbol{ki} = \frac{\boldsymbol{\Theta(t) - \ \Theta(t0)}}{\boldsymbol{\Delta(t) * wst}}}$$

Folosim formulele de mai sus pentru calcularea constatei de integrare Ki, astfel obtinem

```
% Constanta de integrare:
Ki = (y(i4)-y(i3))/wst/(t(i4)-t(i3))
%Ki = 4.9625
```

Avand factorul de integrare aflat mai sus putem obtine functia de transfer a integratorului folosit pentru obtinerea pozitiei din vietza determinata anterior.Functia de transfer este de forma

$$\boldsymbol{Hyw} = \frac{\mathbf{Ki}}{\mathbf{s}}$$

```
Hyw = tf(Ki, [1, 0])
%          4.9625
% Hyw =     -----
%             s
```

2.2Validarea modelului obtinut si calcularea erorii medii patratice normalizate

Pentru a simula modelul matematic obtinut mai sus folosim trecerea in spatiul starilor Si calculam eroare medie patratica normalizata

```
% Spatiul starilor:
A = 0;
B = Ki;
C = 1;
D = 0;

ysim = lsim(A, B, C, D, w, t, y(1));
figure
plot(t, [y, ysim],"LineWidth",2), grid;
eMPNy= norm(y-ysim)/norm(y-mean(y)); % Eroarea = 4.11 %
```

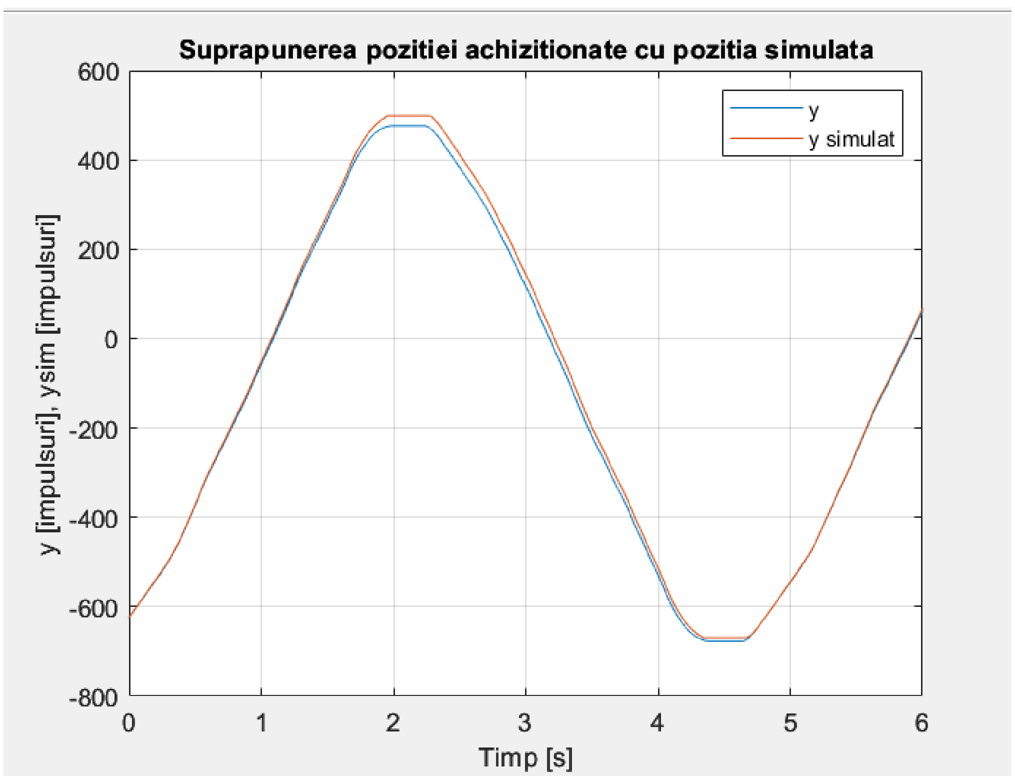


Fig.5 Suprapunerea pozitiei achizitionate cu pozitia simulate

Capitol 3

Partea 2 :Identificarea sistemului prin metode parametrice

2.1 Identificarea parametrica la viteza

2.1.1 Vizualizarea si alegerea datelor de identificare respective validare

Datele se vor alege de pe SPAB(semnal periodic aleatoriu binar) deoarece este semnal bogat in armonice(viteza maxima).

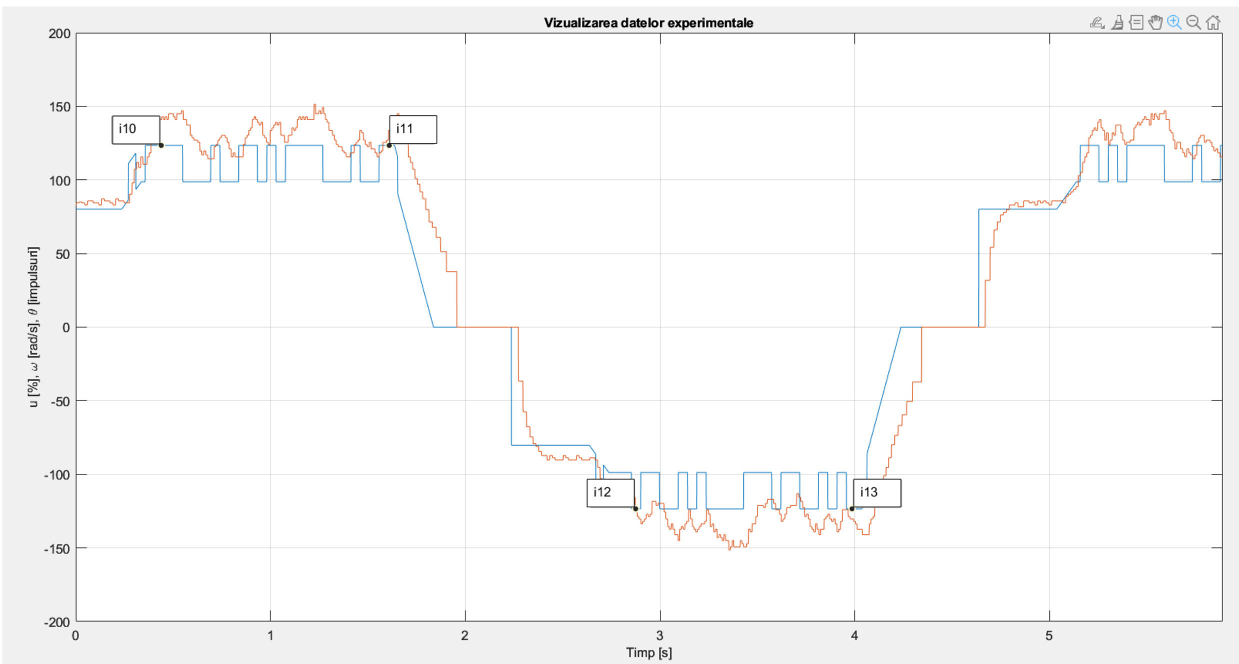


Fig.6 Alegerea datelor de identificare si validare

Datele alese sunt intre indecsii i10 si i11 pentru identificare si intre i12 si 13 pentru validare. Aceste date sunt create cu ajutorul functiei iddata avand ca parametrii viteza, intrarea si perioada de esantionare Te unde Te reprezinta diefenta dintre 2 esantioane de timp.

```
%identificare      %validare
i10 = 583;       i12 = 4022;
i11 = 2236;       i13 = 5508;

Te = t(2)-t(1)  % Te = 0.72 ms = 720 µs
data_id_w = iddata(w(i10:i11), u(i10:i11), Te);
data_vd_w = iddata(w(i12:i13), u(i12:i13), Te);
```

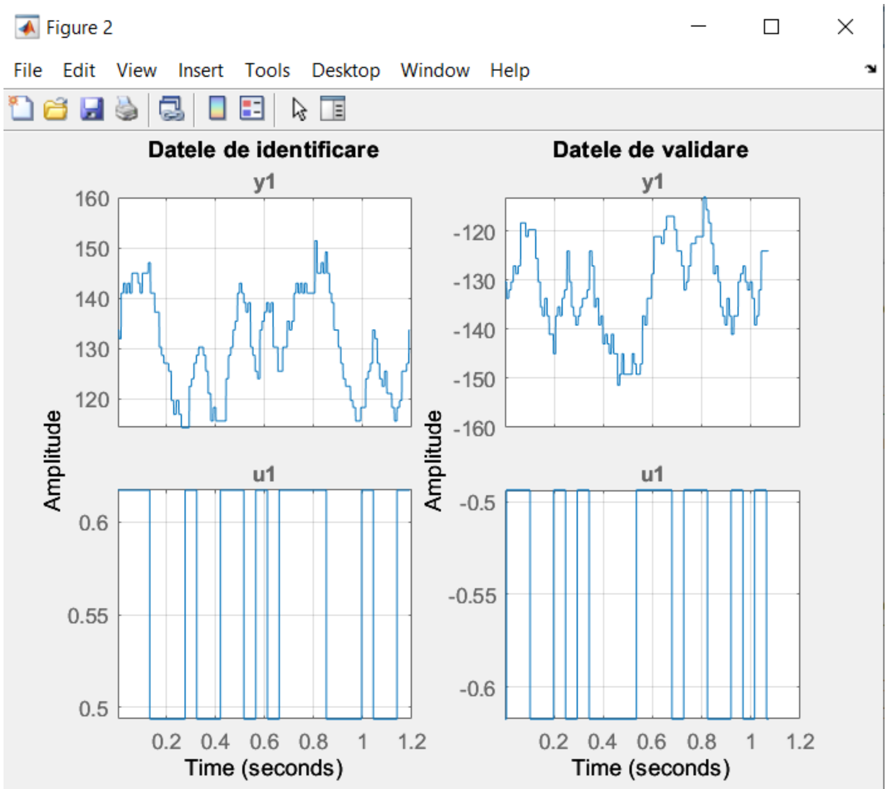
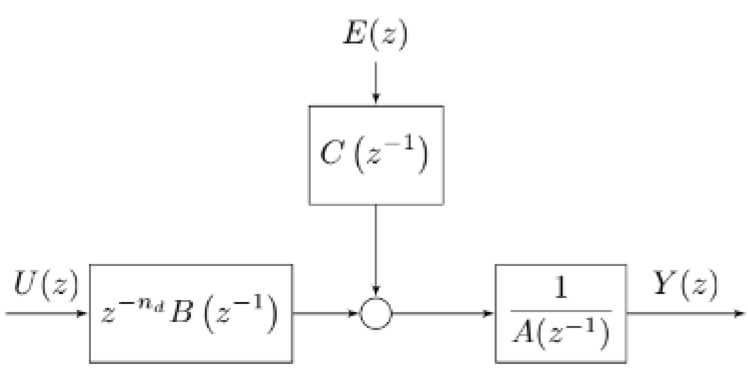


Fig.7 Vizualizarea datelor intrare – viteza

2.1.2 Validarea sistemului prin autocorelatie folosing metoda ARMAX(metoda celor mai mici patrate extinsa)

Dupa testarea in mediul matlab am ales sa folosesc metoda ARMAX fara decimare obtinand cu aceasta cele mai bune rezultate.ARMAX primeste la intrare un obiect de tip iddata si parametrii de structura [nA,nB,nC,nD] si returneaza un obiect de tip idploy care contine modelul matematic al sistemului.

Schema bloc:



Pentru optimizare se creste parametrul de sctructura nD (numarul tactilor de intarziere)pan ace obtinem cel mai bun fit.

```
mw_armax = armax(data_id_w, [1,1,1,1]);
mw_armax_9 = armax(data_id_w, [1,1,1,9]) % cel mai bun fit
```

Modelul matematic in discret obtinut din armax este :

A(z)y(t) = B(z)u(t) + C(z)e(t)

A(z) = 1 - 0.985 z^-1 ➔ $Hduw = \frac{3.548\ z\hat{}-9}{1 - 0.985\ z\hat{}-1} = \frac{3.548\ z\hat{}-1}{1 - 0.985\ z\hat{}-1} * z^{-8}$

B(z) = 3.548 z^-9

C(z) = 1 - 0.002421 z^-1

Cu functia resid se verifica validarea statistica prin autocorelatie iar cu din functia compare obtinem gradul de surprapunere al modelului obtinut cu datele de validare.

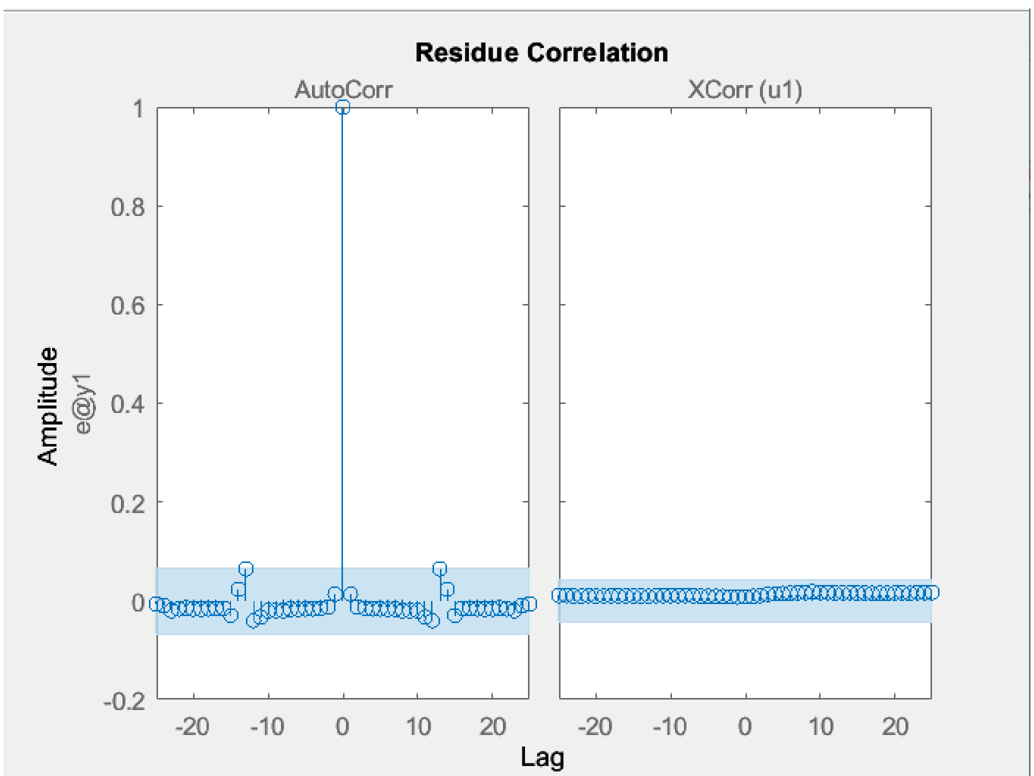


Fig.8 Validarea statistica prin autocorelatie

Din grafic se observa ca este trecut testul de autocorelatie, in banda de incredere aflandu-se cei nA+nB termeni care ne intereseaza.

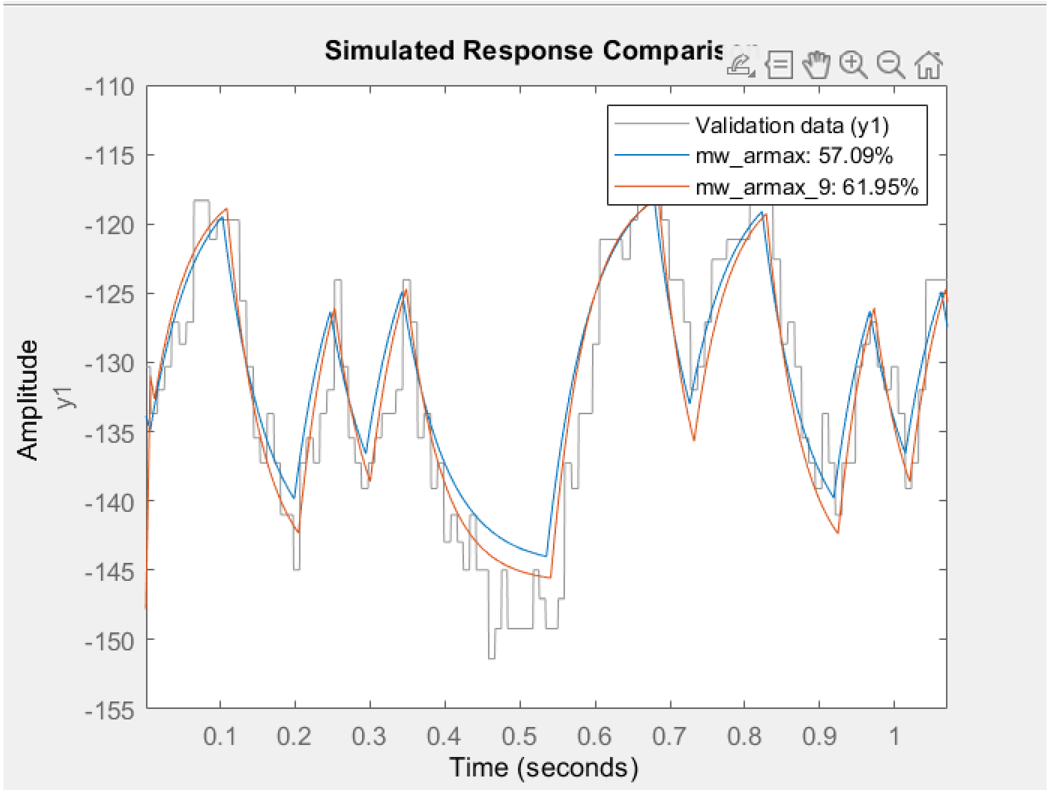


Fig.9 Gradul de suprapunere

Din figura 2.4 se observa ca am obtinut un fit de 61.95% adica o eroare de 100-61.95=38.05%

Mai departe trecem modelul obtinut prin ARMAX in continuu folosind functia matlab d2c si metoda 'zero order hold' si extragem funtia de transfer:

```
% Trecerea modelului ARMAX in continuu:
Mwc = d2c(mw_armax_9, 'zoh');
% Extragerea functiei de transfer din modelul ARMAX:
Hwc = tf(Mwc)
```

Astfel obtinem o functie de transfer Hwc = $e^{-0.00567*s}\frac{4965}{s+20.99}$ pe care o trecem in spatiul starilor pentru a putea obtine conditii initiale nule pentru simularea vitezei unghiulare si calcularea erorii medie patratica normalizata.

```
sys_w = ss(Hwc);
figure
plot(t,200*u);
wsim2 = lsim(sys_w.A, sys_w.B, sys_w.C, sys_w.D, u, t, w(1)/sys_w.C);
hold on
plot(t,[w,wsim2]), grid;
legend('Intrarea u scalata', 'Viteza unghiulara \omega', 'Viteza unghiulara simulata
\omegasim');
xlabel('Timp [s]');
```

ylabel('u [%], \omega [rad/s], \omegasim [rad/s]');

eMPNw= norm(w-wsim2)/norm(w-mean(w)) % Eroarea = 10.60 %

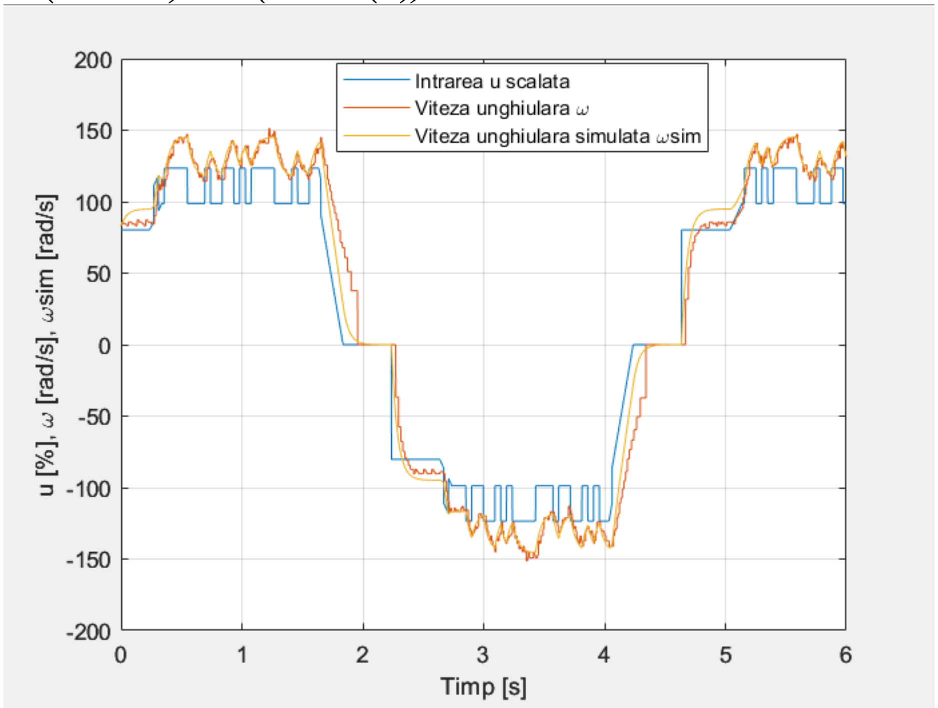
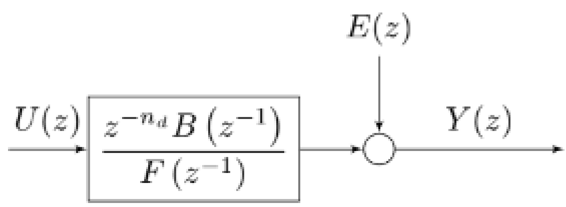


Fig 10 Viteza unghiulara simulate

2.1.3 Validarea sistemului prin intercorelatie folosind metoda OE(metoda erorii de iesire)

Dupa testarea in mediul matlab am ales sa folosesc metoda OE fara decimare obtinand cu aceasta cele mai bune rezultate.OE primeste la intrare un obiect de tip iddata si parametrii de structura [nB,nF,nD] si returneaza un obiect de tip idploy care contine modelul matematic al sistemului.Aceasta metoda introduce o nuoua structura pe baza modelului general descries la inceput, presupunerea de baza fiind ca toata perturbatia se gaseste nemodelata la iesire.

Schema bloc:



Pentru optimizare se creste parametrul de sctructura nD (numarul tactilor de intarziere)pan ace obtinem cel mai bun fit.

```
%
mw_oe = oe(data_id_w, [1, 1, 1]);
mw_oe10 = oe(data_id_w, [1, 1, 10]);
```

Cu functia resid se verifica validarea statistica prin intercorelatie iar cu din functia compare obtinem gradul de surprapunere al modelului obtinut cu datele de validare.

```
figure
resid(data_vd_w, mw_oe);
figure
compare(data_vd_w, mw_oe,mw_oe10);
```

Modelul matematic scos de OE

B(z) = 3.109 z^-10

F(z) = 1 - 0.9867 z^-1

Functia de transfer in distret scoata din modelul OE

$$\text{Hduw}=\frac{\mathbf{3.109\ z^{\wedge}-10}}{\mathbf{1-0.9867\ z^{\wedge}-1}}=\frac{\mathbf{3.109\ z^{\wedge}-1}}{\mathbf{1-0.9867\ z^{\wedge}-1}}\ *\ z^{-9}$$

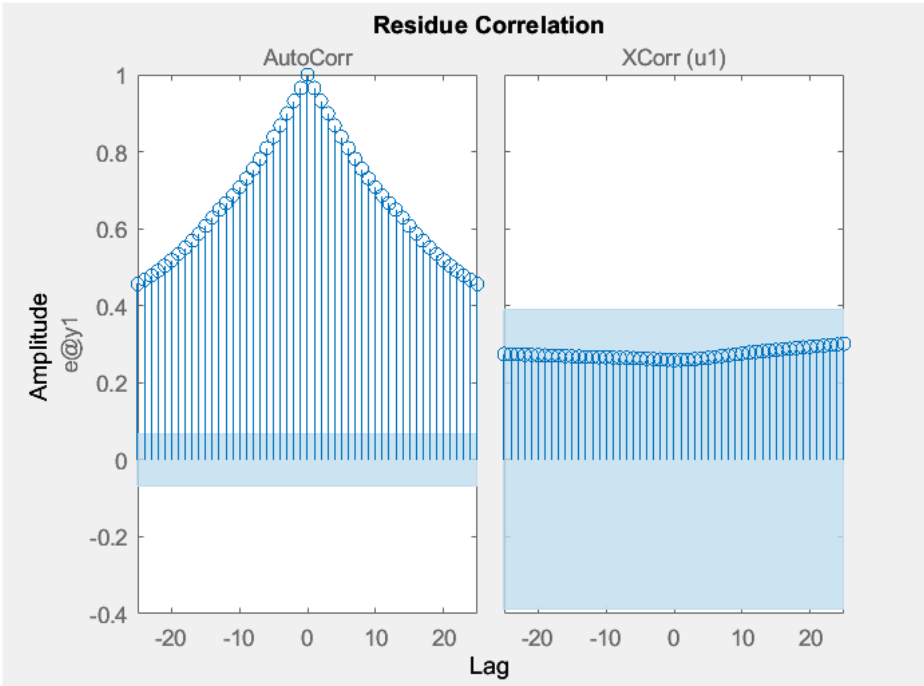


Fig.11 Validarea statistica prin intercorelatie

Din grafic se observa ca este trecut testul de intercorelatie, in banda de incredere aflandu-se cei nB+nF termeni care ne intereseaza.

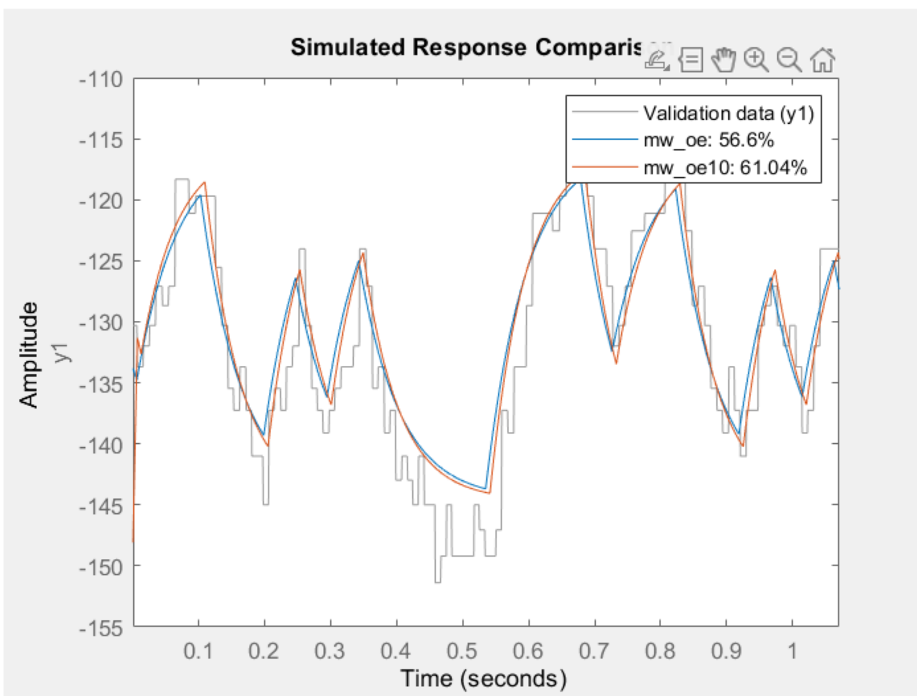


Fig.12 Gradul de suprapunere

Din figura 2.4 se observa ca am obtinut un fit de 61.04% adica o eroare de 100-61.04=38.96%

Mai departe trecem modelul obtinut prin OE in continuu folosind functia matlab d2c si metoda ‘zero order hold’ si extragem funtia de transfer:

$$\textbf{Hwc\_oe} = e^{-0.00684*s}\frac{4347}{s+18.55}$$

pe care o trecem in spatiul starilor pentru a putea obtine conditii initiale nule pentru simularea vitezei unghiulare si calcularea erorii medie patratica normalizata.

Obtinem o eroare eMPN = 9.55%

2.1 Identificarea parametrica la viteza

2.2.1 Vizualizarea si alegerea datelor de identificare respective validare

La fel ca la identificarea la viteza datele alese sunt intre indecsii i10 si i11 pentru identificare si intre i12 si 13 pentru validare. Aceste date sunt create cu ajutorul functiei **iddata** avand ca parametrii viteza, pozitia si perioada de esantionare Te unde Te reprezinta diefenta dintre 2 esantioane de timp.Diferenta in acest caz este constituita de decimarea datelor. Vom decima datele cu un numar de 11 esantioane deoarece valoarea maxima a vitezei adica 151 se repeta de minim 11 ori

```
w1 = w(i10:11:i11);

y1 = y(i10:11:i11);

w1_v = w(i12:11:i13);
y1_v = y(i12:11:i13);
% Date identificare si validare pentru viteza cu decimare
dw1_id = iddata(w1, u1, 11*Te);
dw1_vd = iddata(w1_v, u1_v, 11*Te);

% Date identificare si validare pentru pozitie cu decimare
dy1_id = iddata(y1, w1, 11*Te);
dy1_vd = iddata(y1_v, w1_v, 11*Te);
```

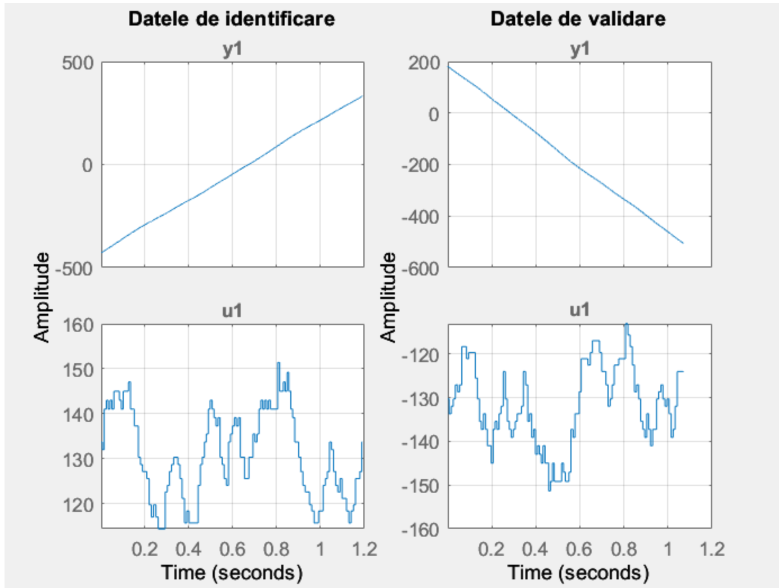


Fig.13 vizualizarea datelor viteza-pozitie

2.2.2 Validarea sistemului prin intercorelatie folosing metoda ARMAX(metoda celor mai mici patrate extinsa)

Metoda se aplica la fel ca la identificarea la viteza difera doar parametrul de structura nD care este 0 in acest caz deoarece nu avem timp mort.

m_armax_poz = armax(dy1_id, [1, 1, 1, 0])

figure
resid(dy1_vd, m_armax_poz);
figure
compare(dy1_vd,m_armax_poz); % Pozitia validata conform autocorelatiei cu ARMAX

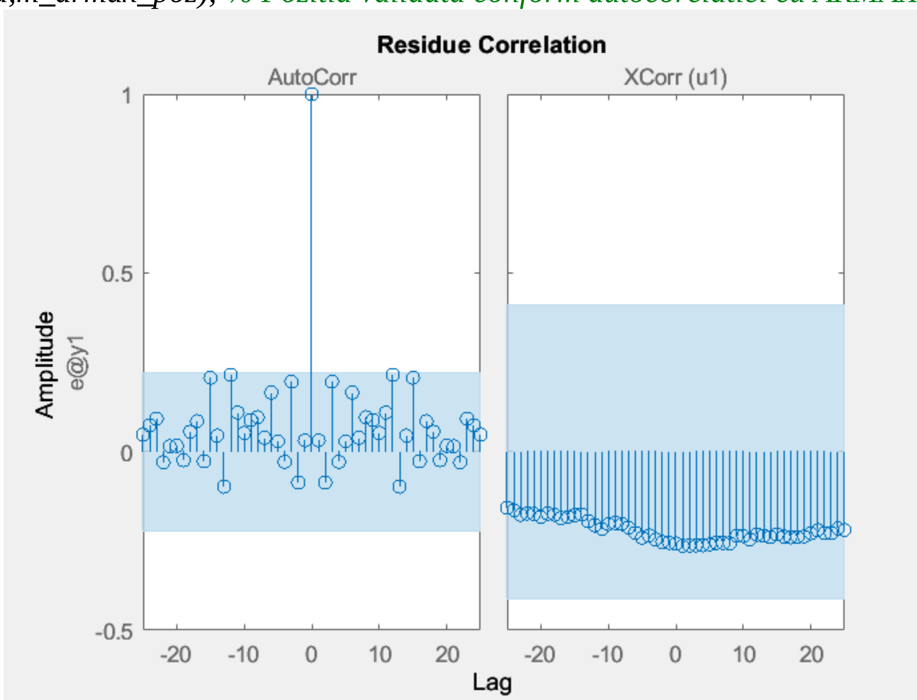


Fig.14 Validarea statistica prin autocorelatie

Din grafic se observa ca este trecut testul de autocorelatie, in banda de incredere aflandu-se cei nA+nB termeni care ne intereseaza.

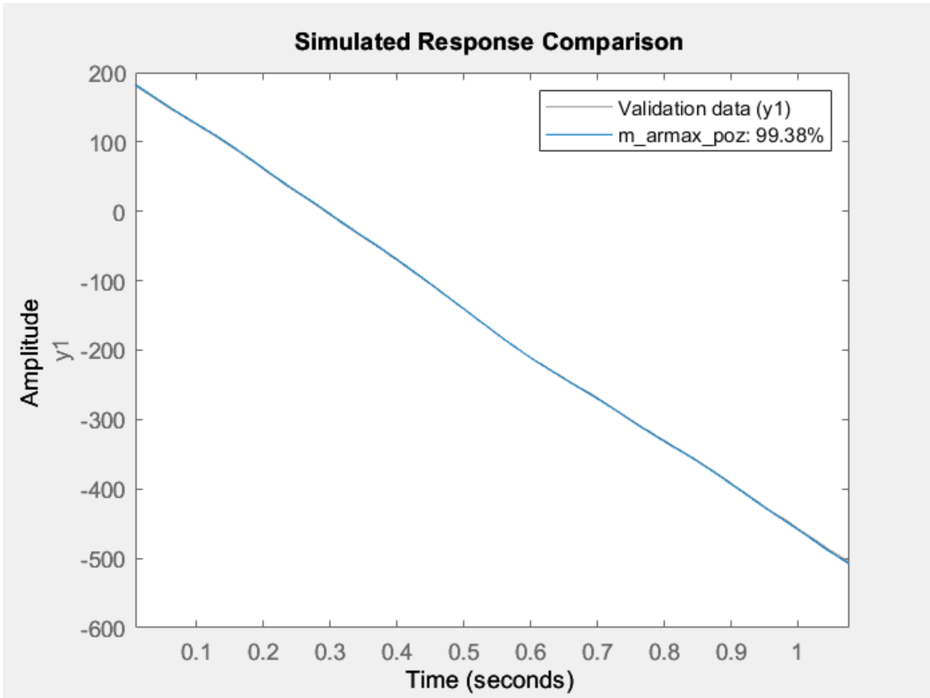


Fig.15 Gradul de suprapunere

Din graficse observa ca am obtinut un fit de 99.38% adica o eroare de 100-99.38=0.62%

Scoatem functia de transfer in discret astfel :

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

% Hyw =
%
%  0.03879
%   --------
%   1 - z^-1
```

Apoi calculam Ki necesar pentru a afla functia de transfer in continuu astfel:

$\boldsymbol{Ki = \frac{b1}{Te\_p}}$ unde b1 este polinomul B scosdin modelul matematic creeat de ARMAX

Iar Te_p este perioada de esantionare dupa decimare.Este necesara impartirea lui b1 la Te deoarece spatiul Matlab nu are impuls de arie unitara.Direct se poate obtine doar cu functia d2c care nu merge aplicata in cazul acesta

```
b1 = m_armax_poz.B;
Te_p = 0.00792; % Perioada de esnationare dupa decimare
Ki_n = b1/Te_p; % 4.907

Hyw_c_2 = tf(Ki_n, [1, 0])
% Hyw_c_2 =
%
%   4.898
%   -----
%     s
```

Trecem in spatiul starilor pentru a putea obtine conditii initiale nule pentru simularea pozitiei si calcularea erorii medie patratica normalizata.

```
sys_y = ss(Hyw_c_2);
ysim2 = lsim(sys_y.A, sys_y.B, sys_y.C, sys_y.D, w, t, y(1)/sys_y.C);

figure
plot(t,[y, ysim2]), grid;
title('Pozitia si Pozitia simulata');
legend('y', 'y simulat')
xlabel('Timp [s]');
ylabel('y [impulsuri], ysim2 [impulsuri]')

eMPN= norm(y-ysim2)/norm(y-mean(y)) % 2.49%
```

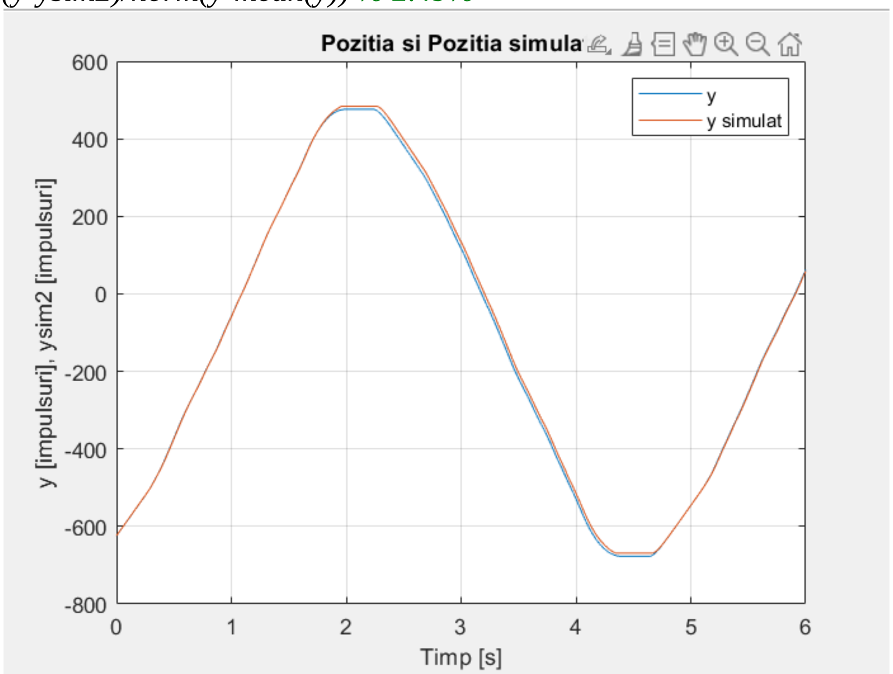


Fig.16 pozitia simulata

2.2.3 Validarea sistemului prin intercorelatie folosind metoda OE(metoda erorii de iesire)

Metoda se aplica la fel ca la identificarea la viteza difera doar parametrul de structura nD care este 0 in acest caz deoarece nu avem timp mort.

```
My_oe = oe(dy1_id, [1, 1, 0])

figure
resid(dy1_vd, My_oe);
figure
compare(dy1_vd, My_oe); % Intercorelatia validata pentru pozitie cu OE
```

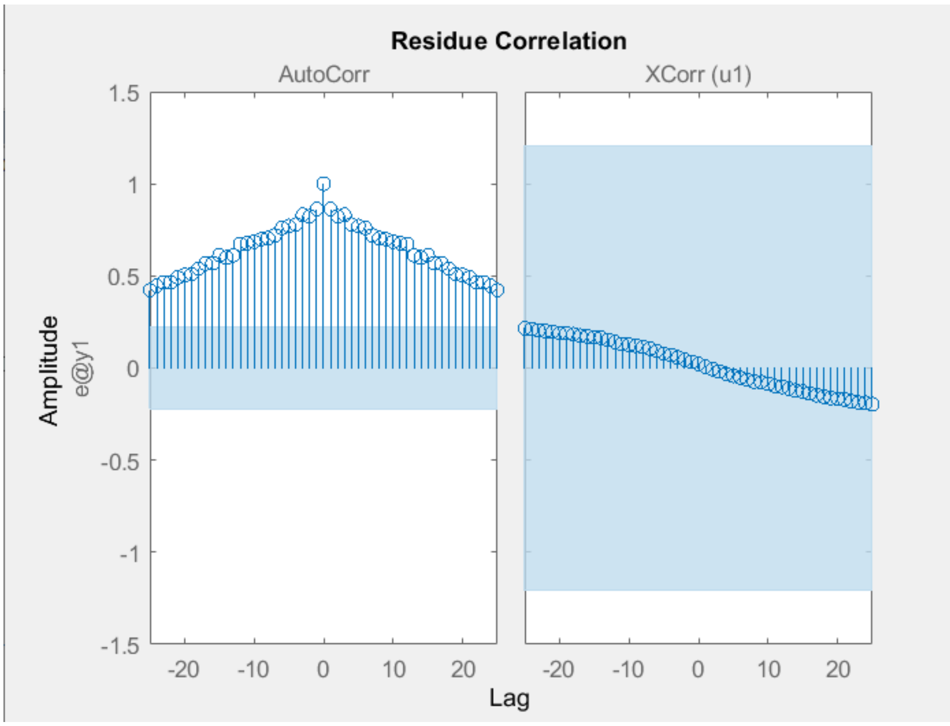


Fig. 17 validarea statistica prin intercorelatie

Din grafic se observa ca este trecut testul de intercorelatie, in banda de incredere aflandu-se cei nB+nF termeni care ne intereseaza.

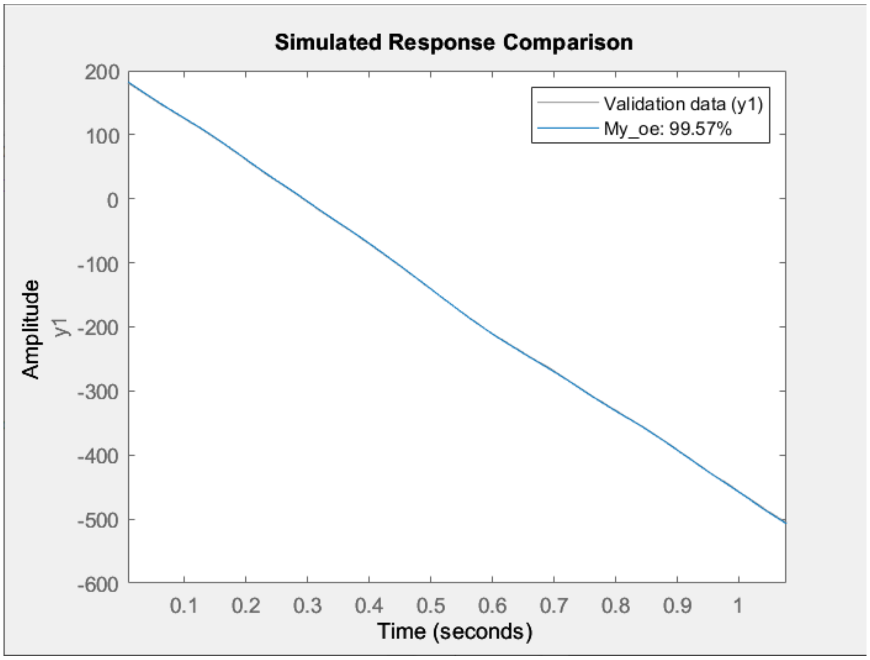


Fig.17 gradul de suprapunere

Din graficse observa ca am obtinut un fit de 99.57% adica o eroare de 100-99.57=0.43%

Scoatem functia de transfer in discret astfel :

```
Hyw_oe = tf(My_oe.B, My_oe.F, Te, 'variable', 'z^-1')

% Hyw_oe =
%
%   0.03877
%   --------
%   1 - z^-1
```

Apoi calculam Ki necesar pentru a afla functia de transfer in continuu astfel:

$\boldsymbol{Ki} = \frac{\boldsymbol{b1}}{\boldsymbol{Te\_p}}$ unde b1 este polinomul B scos din modelul matematic creeat de OE

Iar Te_p este perioada de esantionare dupa decimare.Este necesara impartirea lui b1 la Te deoarece spatiul Matlab nu are impuls de arie unitara.Direct se poate obtine doar cu functia d2c care nu merge aplicata in cazul acesta

```
b1_oe = My_oe.B;
Te_OE = 0.00792; % Perioada de esnationare dupa decimare
% Constanta de integrare:
Ki_oe = b1_oe/Te_OE; % 4.9067
Hyw_oe_c = tf(Ki_oe, [1, 0])
% Hyw_c_2 =
%   4.895
%   -----
%     s
```

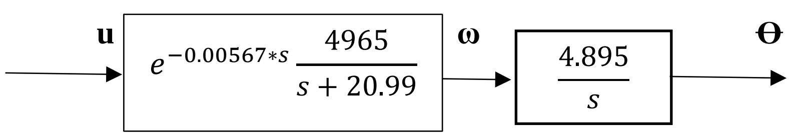
Trecem in spatiul starilor pentru a putea obtine conditii initiale nule pentru simularea pozitiei si calcularea erorii medie patratica normalizata.

```
sys_y = ss(Hyw_c_2);
ysim2 = lsim(sys_y.A, sys_y.B, sys_y.C, sys_y.D, w, t, y(1)/sys_y.C);

sys_y2 = ss(Hyw_oe_c);
ysim3 = lsim(sys_y2.A, sys_y2.B, sys_y2.C, sys_y2.D, w, t, y(1)/sys_y2.C);

figure
plot(t,[y, ysim3]), grid;
title('Pozitia si Pozitia simulata');
legend('y', 'y simulat')
xlabel('Timp [s]');
ylabel('y [impulsuri], ysim2 [impulsuri]')

eMPN= norm(y-ysim3)/norm(y-mean(y)) % 2.43%
```

Fig.18 Pozitia simulate

Capitol 4 :Concluzii

In urma indetificarii modelelor matematice care formeaza sistemul de control al axei motorului BLDC, putem observa ca modelul matematic obtinut pentru indentificarea sistemului de tip intrare-viteza alegem metoda ARMAX deoarece are cea mai buna eroare 10.60% si cea mai buna suprapunere fata de celelate metode, pentru sistemul de tip viteza-pozitie alegem metoda OE, avand deasemenea cele mai bune estimari ale acestui model.
Avand cele 2 sisteme obtinute, punandu-le in serie, putem observa dinamica intregului sistemului si analizarea acestuia in cel mai precis mod, avand functia de transfer: (…).



$$\textbf{Hfinal} = \exp(-0.00567*s) * \frac{2.43e04}{s^2 + 20.99\, s}$$