Proiect
-Identificarea Sistemelor-

Profesor coordonator: **Prof.dr.ing Petru Dobra**

Student: Babota Sebastian-Egon

Grupa:30131

Cuprins

Capitolul 1

Identificarea neparametrica a sistemului exploatand fenomenul de rezonanta

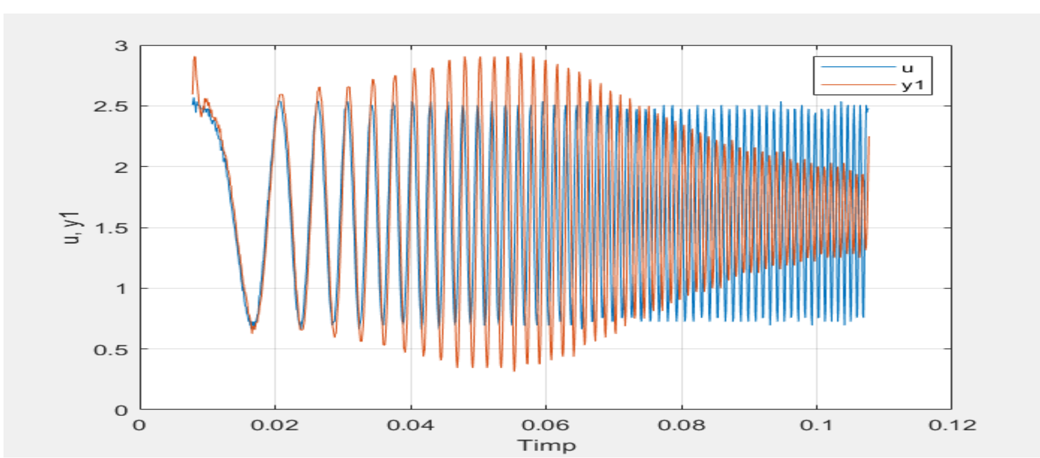
Pentru realizare identificarii parametrice am ales sa exploatam fenomenul de rezonata. Rezonanta este tendinta unui sistem de a oscila la anumite frecvente cu o amplitudine mai mare decat la altele. Frecventele de rezonanta se gasesc la amplitudinile maxime ale sistemului.

1.1 Vizualizarea datelor

Pentru vizualizarea datelor primite de la laborator am incarcat in MatLab

fisierul ce contine datale experimentale (scope86.csv). Aceste date ne vor fi de folos in identificarea neparametrica dar si parametrica a unui sistem de ordin 2. Fisierul contine patru coloane de de date : prima coloana reprezinta momentele de timp, a doua coloana reprezinta intrarea sistemului, iar ultimele doua coloane reprezinta cele doua iesiri.

Pentru primul set de date ne vom folosi de prima iesire a sistemului respectiv prima coloana, analiza se va face pe un sistem de ordin 2 cu poli complex-conjugati.



1.2 Parametrii calculati

Fiind un sistem de ordin 2 am folosit in urmatoarele etape functia de trasfer de forma :

$$G(s) = \frac{Kwn^2}{s^2+2\zeta wns+wn^2}$$

Aceasta functie de transfer necesita aflarea urmatorilor parametrii:

K – factor de proportionalitate

ζ- factorul de amortizare

wn – pulsatia naturala

Primul pas in indentificarea acestor paramterii este alegerea unor puncte de pe graficul din Fig1 din zona frecventelor de rezonanta.

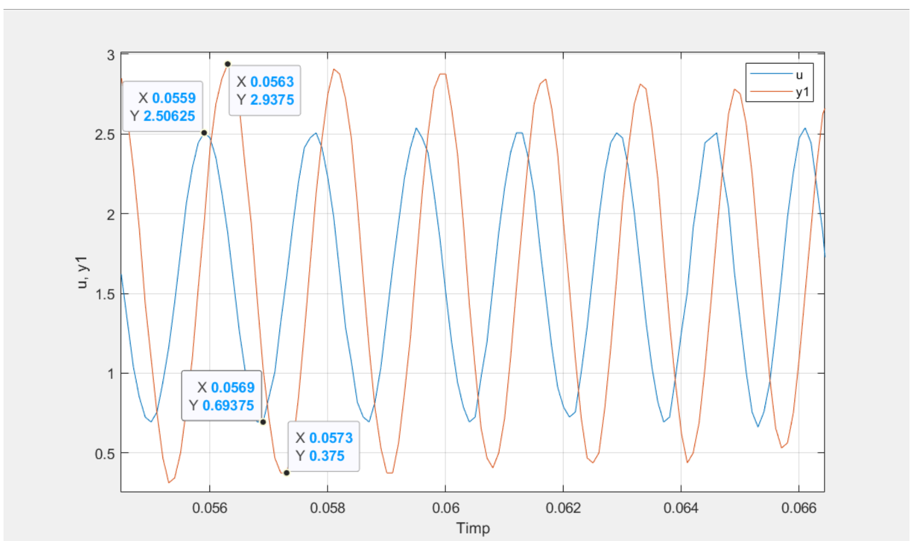


Figura 2. Pulsatiile la rezonanta ale sistemului si al intrarii

u_max = 482 ; u_min = 492; y_max = 486; y_min = 476;

Pentru a calcula <u>valoarea factorului de proportionalitate</u>(K) am folosit raportul dintre primul varf al semnalului de iesire(y) si primul varf al semnalului de intrare(u) :

$$K = \frac{y1(Ky_{max})}{u(Ku_{m}ax)} = \underline{1.014} \text{ ; (kymax = 130 ; kumax = 128)}$$

Calculul perioadei consta in scaderea a doua varfuri consecutive ale semnalului, t1 = 466 respectiv t2 = 486:

T_u = t(t2)-t(t1) = 0.0020 [s] ;

Pulsatia de rezonanta se calculeaza cu ajutorul raportului :

$$wr = \frac{2\pi}{T} = 3.14e+03 \text{ [Hz] ;}$$

Modulul la rezonanta sau amplificarea l-am calculat prin raportul dintre amplitudinea semnalului de iesire(y1) si amplitudinea semanlului de intrare(u) la rezonata :

$$Mr = \frac{(y1(y_{m}ax)-y1(y_{m}in))}{(u(u_{m}ax)-u(u_{m}in))} = 1.4283.$$

Factorul de amortizare il calculam in raport cu amplificarea avand in vedere ca lucrez cu valori experimentale ale amplificarii la rezonanta:

$$\zeta = \sqrt{\frac{M_r-\sqrt{M_r^2-1}}{2M_r}} = 0.378 \text{ ;}$$

Pulsatia naturala o voi calcula cu ajutor factorului de amortizare si al pulsatiei de rezonanta :

$$\omega_n = \frac{\omega_r}{\sqrt{1-2\zeta^2}} = 3.7179e+03[Hz].$$

1.3 Verificarea datelor

Primul pas in verificarea si validarea datelor calculate mai sus il reprezinta construirea functie de transfer a sistemului. Daca ne folosim de formula pe care am definit-o ca fiind functia de transfer de ordin 2 : G(s) = $\frac{Kwn^2}{s^2+2\zeta wns+wn^2}$, si inlocuim cu datele obtinute la subcapitolul anterior obtinem :

num = [$K\omega_n^2$] = 1.4016e+07.

den = [1,2ζwn, wn^2] = [1, 2.81e+03, 1.382e+07];

$$\text{Sys(s)} = \text{tf(num, den)} = \frac{1.402 \cdot 10^7}{s^2 + 2812 \cdot s + 1.382 \cdot 10^7}$$

In urma plotarii semanlului initial dat y1 si suprapunerea acestuia peste semnalul recreat din parametri calculati se observa ca pornirile celor doua semnale nu coincid.

//Cod Matlab//

y1c = lsim(sys,u,t);

plot(t,[y1,y1c])

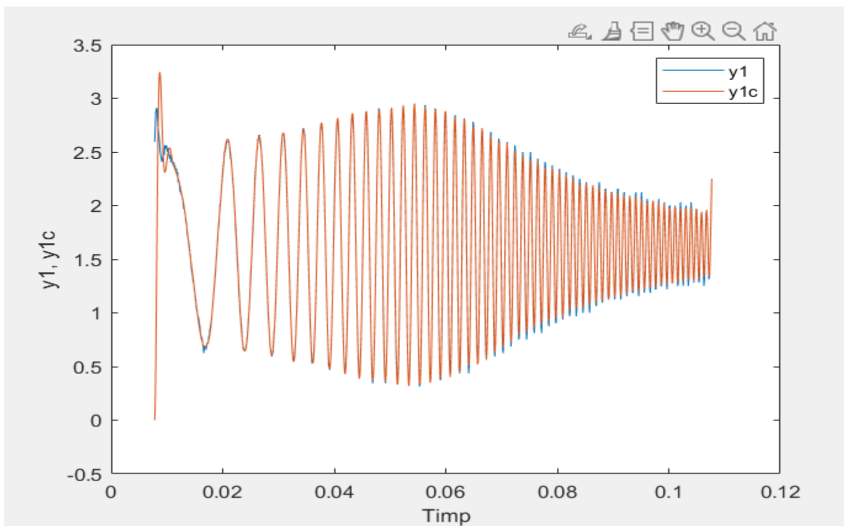


Figura 3. Compararea raspunsului calculat cu iesirea

Pentru a rezolva aceasta problema trecem tot sistemul calculat in spatiul starilor astfel incat sistemul nostru sa nu mai porneasca din conditii initiale nule. Realizarea spatiului starilor consta in creearea matricilor de stare.

$$A = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ K\omega_n^2 \end{bmatrix}$$

$$C = [1 \quad 0] \quad D = [0]$$

sys2 = ss(A,B,C,D); -construim sistemul cu matricile calculate

$A = 10^7 \cdot [0 , 0.0000; -1.3823 , -0.0003]$ $\qquad$ $C = [1\ 0]$

$B = 10^7 \cdot [0; 1.4016]$ $\qquad$ $D = [0]$

Dupa realizarea si calcularea matricilor si al sistemului rezultat din acestea o sa repetam plotarea si observam diferentele aparute.

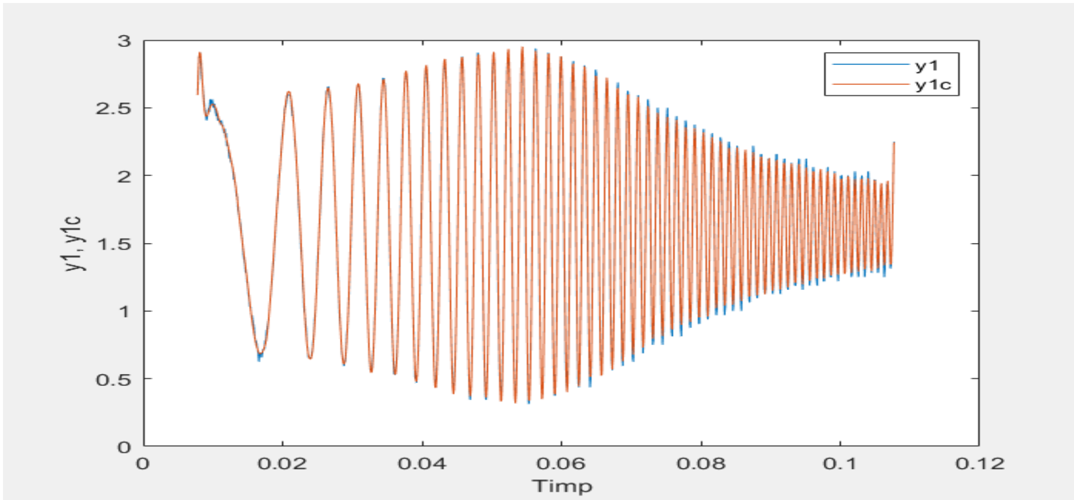


Figura 4. Compararea dupa impunerea conditilor

//Cod Matlab//

y1c = lsim(sys2,u,t,[y1(1),2000]);

plot(t,[y1,y1c])

Cea de a patra componenta a functiei *lsim* (y1(1),2000) reprezinta punctul de unde dorim sa pornesca sistemul nostru, punct care coincide cu punctul de pleacare al primei iesiri, respectiv panta aproximata de pornire.

Ca o validare a identificarii realizate putem calcula *eroarea medie patratica normalizata* intre sistemul dat si cel calculat.

$$e_{MPN} = \frac{||y_1 - y_{1c}||}{||y_1 - \text{mean}(y_1)||} \cdot 100$$

$$e_{MPN} = 4.4608$$

//Cod MatLab//

*eMPN = norm(y1-y1c)/norm(y1-mean(y1))*100 ;*

Capitolul 2

Raspunsul in frecventa

In aceasta parte a proiectului vom incerca realizarea raspunsului in frecventa al sistemului, prin aliniamentul diagramei Bode cu niste puncte calculate cu ajutorul paramteriilor necesari luati din trei zone de frecvente, zona frecventelor joase, zona frecventelor ridicate, si la frecventele de rezonanta.

2.1 Colectarea si calcularea datelor

Colectarea datelor pentru aceasta parte de proiect a constat in extragerea maximelor si a minimeleor semnalului de intrare si a celui de iesire din cel putin doua armonici consecutive luate din trei parti diferite ale intregii perioade de oscilatie a sistemului, adica armonici luate la frecvente joase, la frecvente ridicate si la frecventele de rezonanta.

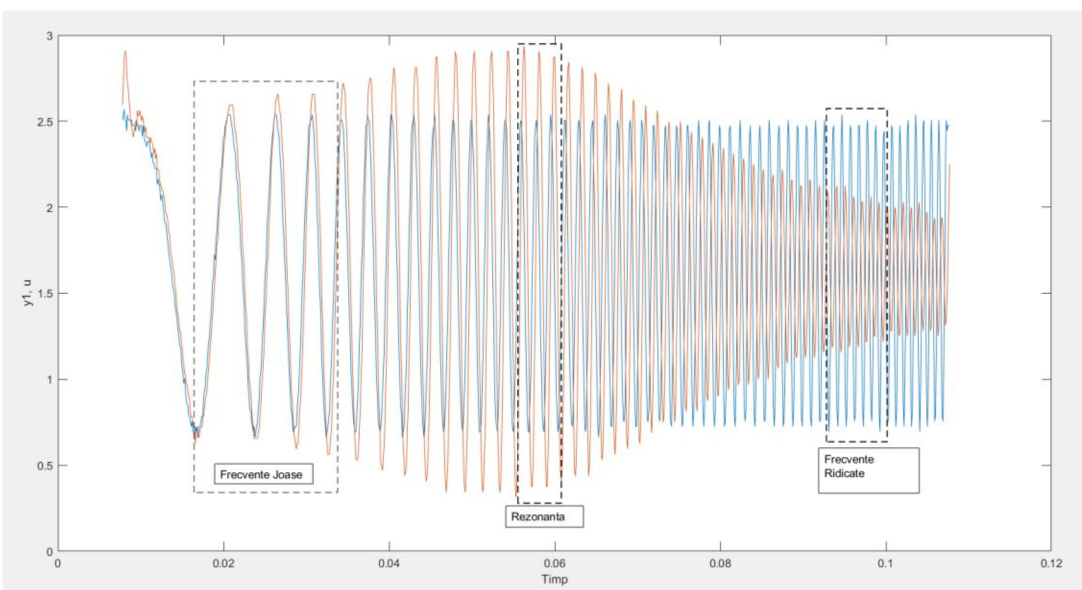


Figura 5. Alegerea zonelor de colectare a datelor

Pentru luarea punctelor procedam la fel ca la experimentul anterior luand minimele si maximele celor doua semnale.

% zona frecv joase

umax1 = 130; y1max1 = 132; umin1 = 160; y1min1 = 162; umax2 = 185; y1max2 = 188;

%zona frecv inalte

umax1_2 = 969; y1max1_2 = 973; umin1_2 = 973; y1min1_2 = 977; umax2_2 = 978;

y1max2_2 = 982;

% frecv medii

umedmax1 = 747; y1medmax1 = 751; umedmin1 = 753; y1medmin1 = 757; umedmax2 = 759;

y1medmax2 = 763;

Dupa colectarea tuturor indicilor necesari putem incepe calculu Pulsatilor(w), Modulul de rezonanta(Mr), si Faza(phi).

<u>Pulsatia</u> : $\omega \quad = \frac{2\pi}{T_u}$

<u>Modul de rezonanta</u> : Mr = $\frac{\left(y1(y_m ax)-y1(y_m in)\right)}{\left(u(u_m ax)-u(u_m in)\right)}$

<u>Faza</u>: phi = ω(t(y_max)-t(u_max))

//Cod Matlab//

ph1 = rad2deg(w1(t(y1max1)-t(umax1)))*

Urmatorul pas in calcularea datelor obtinute pana acum ar fi transformarea valorii modulelor de rezonanta in decibeli, si punerea in vectori a valorilor aflate la pasul anterior.

Pentru transformarea in deciblei vom folosi formula $dB_{M_{r1}} = 20 \cdot \log_{10}(M_{r1})$.

Vectorii de valori sunt :

Pulsatiile = [1.047 1.256 6.283 5.236 5.236 5.236 6.283 4.488 3.1416]$* 10^3$[Hz];

Mr_dB =[0.5793 0.8550 -8.6731 -2.4988 -2.5516 -5.4213 -7.0050 -1.3389 3.0963];

Phi = [-12.0000 -21.6000 -144.0000 -120.0000 -120.0000 -72.0000]

Pentru a finaliza experimentul vom reecrea diagarma Bode a sistmului si vom pune pe aceasta punctele calculate la pasii anteriori totoadata urmarind daca punctele respecta traiectoria diagramei de faza dar si a diagramei de castig.

//Cod MatLab//

w = logspace(2,5,1000);

[M,Ph] = bode(sys2,w)

M = squeeze(M);

Ph = squeeze(Ph);

*subplot(211),semilogx(w,20*log10(M),frecv,magnitudini_dB,'x');*

subplot(212),semilogx(w,Ph,frecv,ph,'x');

<u>Explicatia codului</u>

w = logspace(2,5,1000); - Creeaza un vector logaritmic de frecvente cuprinse intre $10^2\ si\ 10^5$, cu 1000 de puncte

[M,Ph] = bode(sys2,w) – Functia bode calculeaza magnitudinea si faza pentru sistemul dat la frecventele specificate de w.

M = squeeze(M);.

Ph = squeeze(Ph); - Funcția squeeze transformă datele din matricee tridimensionale în vectori, pentru procesare ușoară

*subplot(211),semilogx(w,20*log10(M),frecv,magnitudini_dB,'x');*

-Primul grafic (subplot(211)) reprezinta magnitudinea (*20*log10(M))* pe o scara logaritmic a frecventelor w

-Frecvențele din vectorul frecv și amplitudinile din magnitudini_dB sunt reprezentate cu simboluri “x” pentru comparație.

subplot(212),semilogx(w,Ph,frecv,ph,'x');

-Se reprezinta faza sistemului pe o scara logaritmica a frecventei *w*

-Se plasazea simboluri ‘x’ la pozitiile din (frecv, ph) pentru a indica punctele de referinta

2.2 Validarea datelor

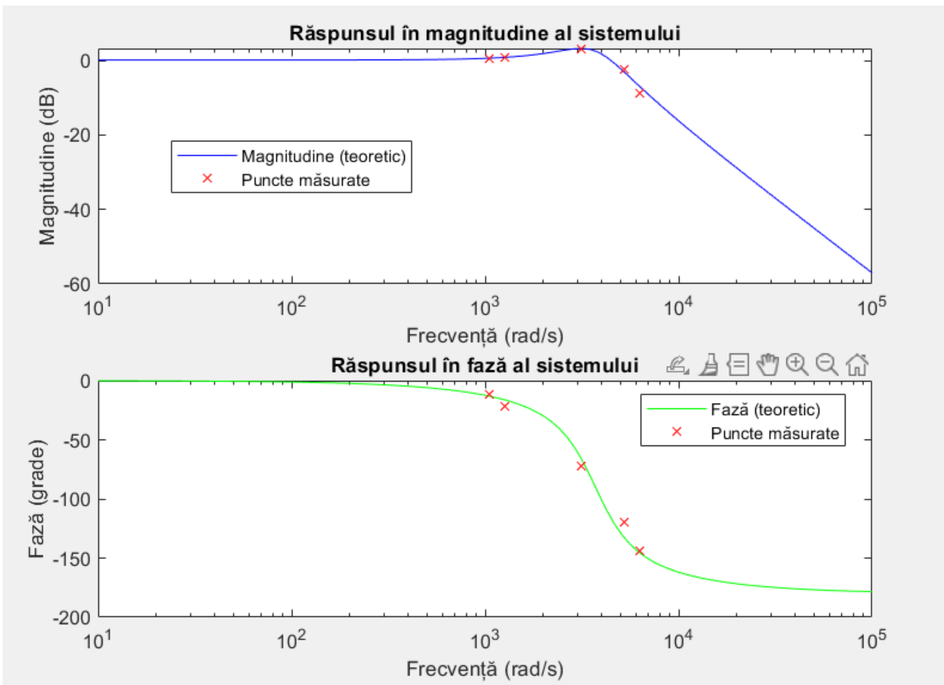


Figura 6. Diagrama Bode a sistemului si punctele calculate

Validarea datelor se poate deduce din reprezentarea grafica a punctelor calculate, si din aliniamentul acestora cu cele doua diagrame.

Capitolul 3

Metode parametrice de identificare

Metodele parametrice de identificare a sistemelor reprezinta tehnici utilizate pentru a determina parametrii unui model matematic ce descrie comportamentul unui sistem fizic pe baza datelor experimentale, cum ar fi intrarile si iesirile sistemului. Aceste metode presupun alegerea unei structuri de model adecvate (precum ARX, ARMAX, OE etc.) si estimarea parametrilor astfel încat modelul să reproduca cat mai precis comportamentul sistemului real.

3.1 Metode parametrice

În această secțiune a proiectului, voi valida cele două semnale ale sistemelor analizate, y1și y2, utilizând metode de autocorelație și intercorelație. Validarea se va realiza pe baza datelor obținute din aplicarea diferitelor modele (ARX, ARMAX, OE, N4SID).

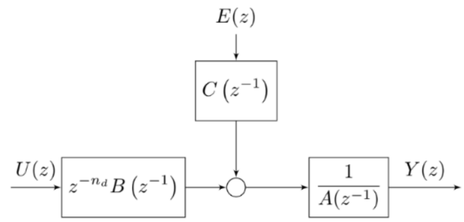
Mai întâi, ne vom concentra pe primul sistem (y1), pe care îl vom valida prin autocorelație folosind două modele distincte, și prin intercorelație, de asemenea, utilizând două modele. Cele patru modele utilizate sunt:

- **ARX** – bazat pe metoda celor mai mici pătrate recursive,
- **ARMAX** – utilizând metoda celor mai mici pătrate extinsă,
- **OE** – metoda erorii de ieșire,
- **N4SID** – metoda de identificare în spațiul stărilor.

Rezultatelor acestor metode le au fost aplicate inca o metoda parametrica pentru minimizarea erorilor dintre valorile observate si cele prezise de model, acesta metoda se numeste PEM.

Validarea prin autocorelatie (N4SID,ARMAX)

1. ARMAX



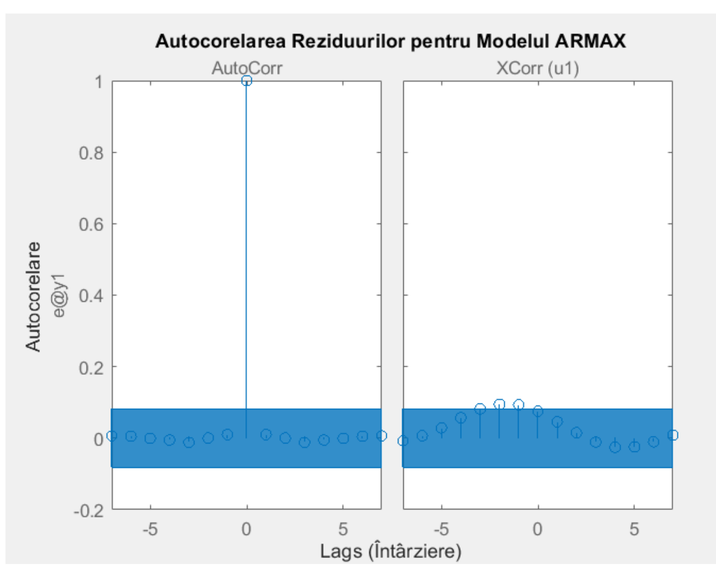
-Structura metodei ARMAX-

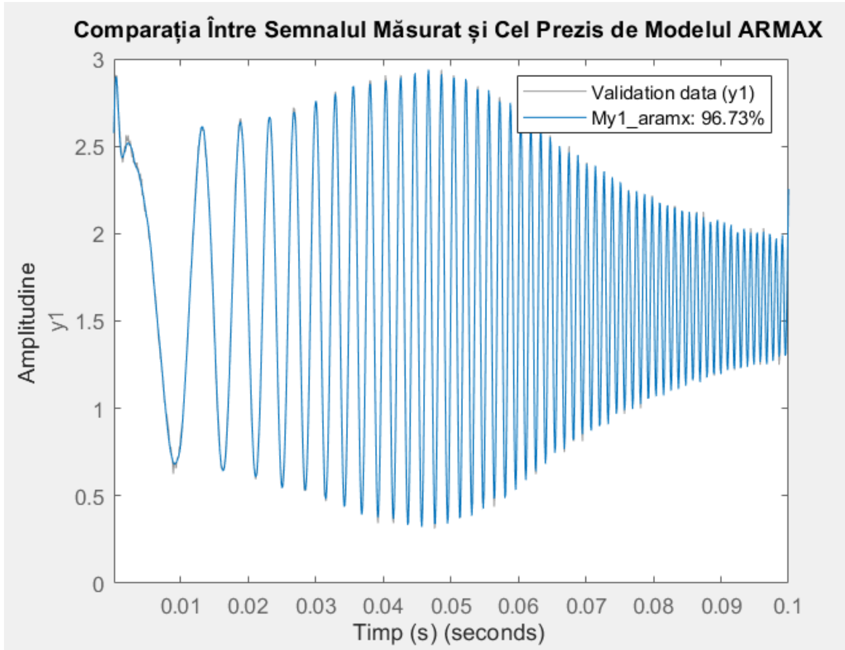
$$A(z^{-1})Y(z) = z^{-nd} \cdot B(z^{-1})U(z) + C(z^{-1})E(z)$$

Identificarea consta in estimarea coeficientilor polinoamelor A,B si C nA, nB si nC respectiv si nD numarul tactilor de intarziere.

//Cod MatLab//

```
dt = t(2) - t(1)

dy1 = iddata(y1,u,dt)

My1_aramx = armax(dy1,[2 1 2 1])

resid(dy1,My1_aramx,'corr',7)

compare(dy1, My1_aramx);
```





dt – perioada de esantionare, dy1 – datele/obiectul creat pentru iesirea y1 intrarea u si perioada de esantionare dt.

resid – primeste ca parametrii obiectul,modelul realizat cu pem, si nr de pct pe care vrem sa facem validarea

-calculeaza reziduurile (adică diferentele dintre iesirile masurate si cele prezise de model) și ploteaza aceste reziduuri pentru a vizualiza performanta modelului.

compare – afiseaza un grafic de comparatie intre semnalul de iesire cel masurat si cel estimat de modeul dat ca parametru in apelul functiei, si iti ofera precizia estimarii facute.

Model rezultat

$$A(z) = 1 - 1.624z^{(-1)} + 0.7462z^{(-2)}$$

$$B(z) = 0.1237z^{(-1)}$$

$$C(z) = 1 - 1.39z^{(-1)} + 0.5392z^{(-2)}$$

2. N4SID

Metoda **N4SID** folosește o tehnică bazată pe subspațiu pentru a obtine o estimare optima a parametrilor modelului. Acesta este un algoritm de identificare a sistemului care functioneaza bine, chiar si în cazul în care datele sunt zgomotoase.

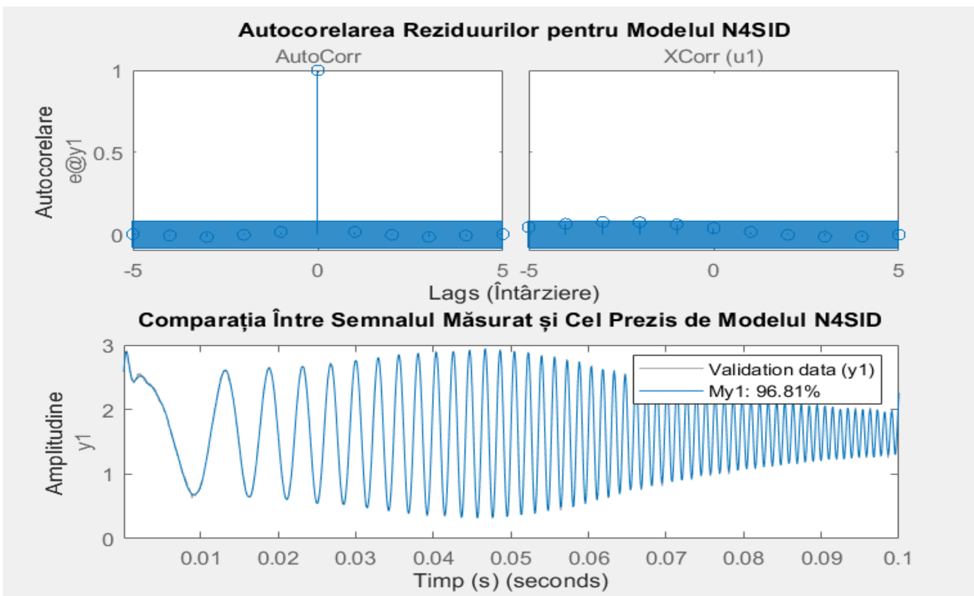
//Cod MatLab//

My1_ss = n4sid(dy1,1:7)

My1 = pem(dy2,My1_ss)

resid(dy1,My1,'corr',7)

compare(dy1,My1)



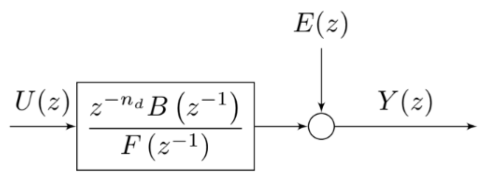
<u>Model rezultat</u>

A = [0.7314 -0.2229 0.4170 0.8964] C = [14.3894 0.2892]

B = [0.0096 ; -0.0341] D = [0]

Validarea prin intercorelatie (OE)

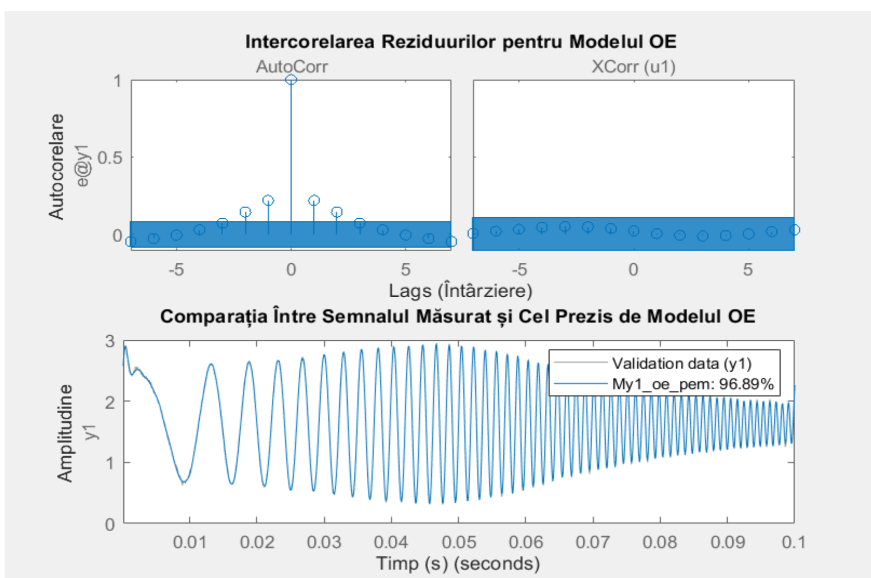
1. OE



-Structura OE –

Identificarea consta in estimarea coeficientilor polinoamelor B si F. Parametrii de structura ai sistemului sunt: nB = deg B, nF = deg F, respectiv nD numarul tactilor de intarziere. In matlab aceasta metoda se gaseste implementata in functia *oe* iar parametrii alesi pentru acest sistem sunt [2 2 0].

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

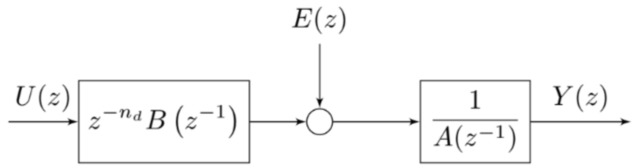


Model rezulata OE

$$\text{B(z)} = 0.006097 + 0.1165z^{(-1)}$$

$$\text{F(z)} = 1 - 1.628z^{(-1)} + 0.7492z^{(-2)}$$

2. IV4

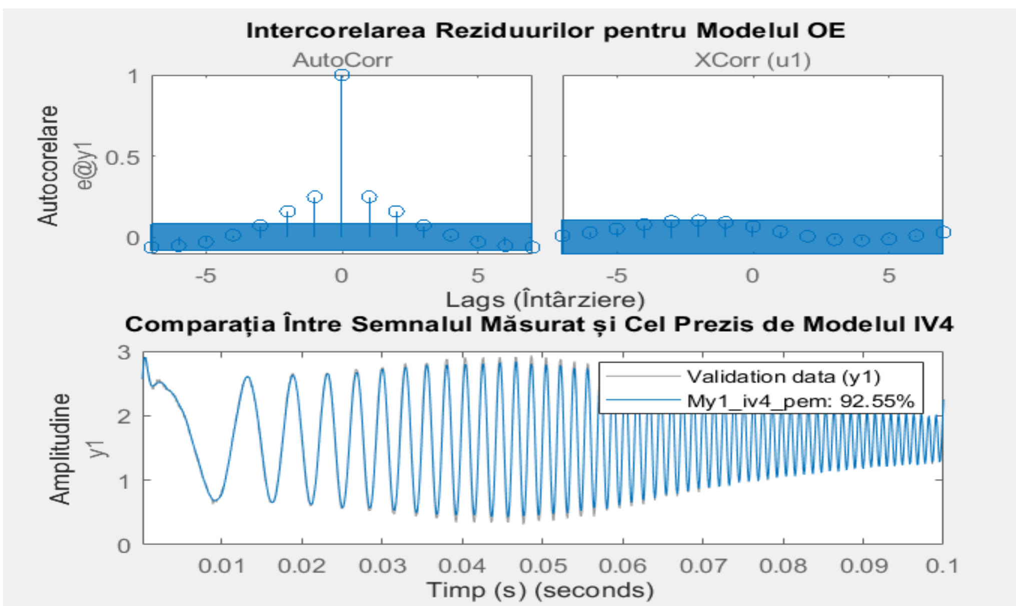


Structura IV4

$$A(z^{-1})Y(z) = z^{-nd} \cdot B(z^{-1})U(z) + E(z)$$

Metoda variabilelor instrumentale se bazează pe aceeași structură ca și metoda ARX, însă diferența constă în utilizarea unor valori noi pentru parametrii vectorului, care sunt selectați astfel încât să îmbunătățească estimarea modelului.

In MatLab aceasta metoda este implementata in functia *iv4* si accepta un vector cu 3 parametrii de structura nA nB nd [2 2 1]. Eroarea obtinuta este *7.45%.*



<u>Model rezulata IV4</u>

$$\text{A(z)} = 1 - 1.586z^{(-1)} + 0.7128z^{(-2)}$$

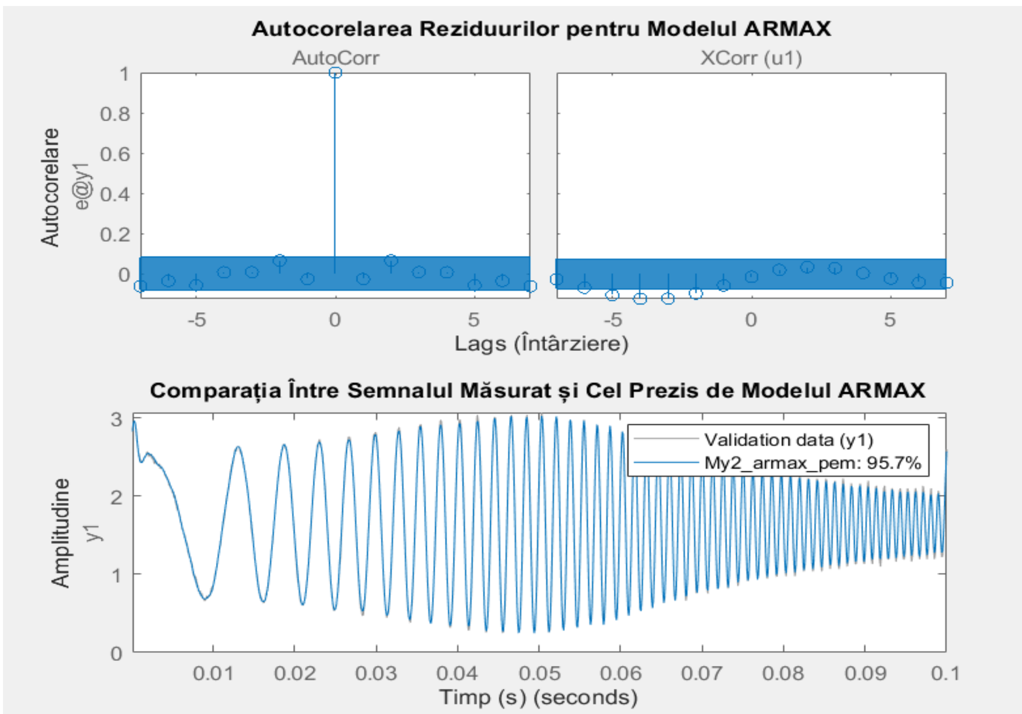
$$\text{B(z)} = -0.01517 + 0.1437z^{(-1)}$$

Sistemul 2 cu zero

Validarea prin autocorelatie (ARMAX)

1. ARMAX

Validarea acestui model se face prin autocorelatie iar eroarea obtinuta este aporximativ *4.3%*. Parametrii nA, nB, nC, nd pentru acest sistem sunt [2 2 2 0].



Model rezultat

$$A(z) = 1 - 1.626z^{(-1)} + 0.7559z^{(-2)}$$

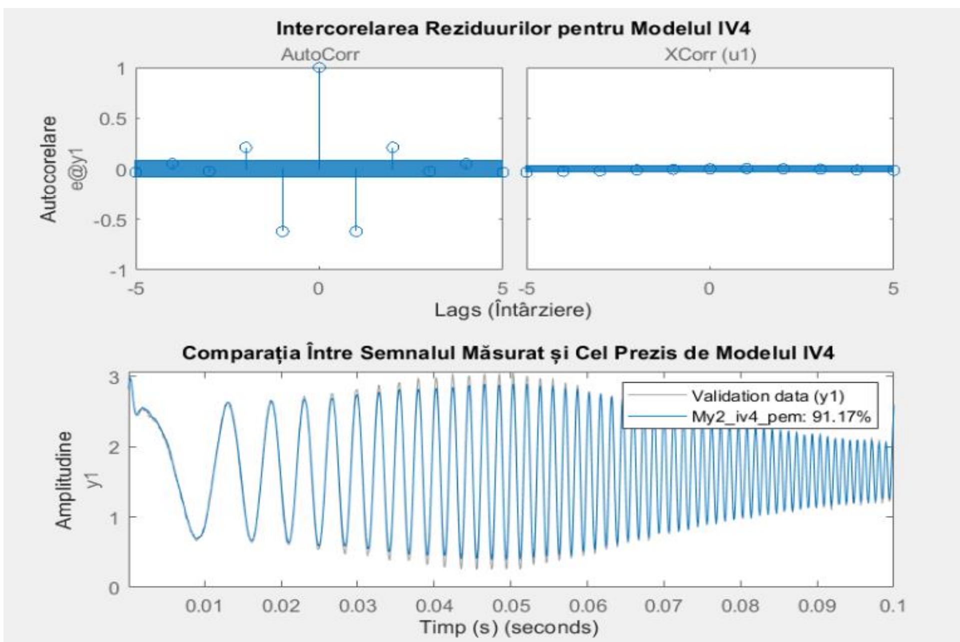
$$B(z) = 0.1352 - 0.003346z^{(-1)}$$

$$C(z) = 1 - 1.163z^{(-1)} + 0.307z^{(-2)}$$

Validarea prin intercorelatie (IV4)

1. IV4

Parametrii nA, nB, nd alesi pentru validarea acestui sistem cu un zero sunt [2 2 0] , din graficul al doilea reiese ca avem o eroare de aproximativ *8.2%,* iar din primul grafic putem observa ca testul de intercorelatie este trecut.



Model rezulata IV4

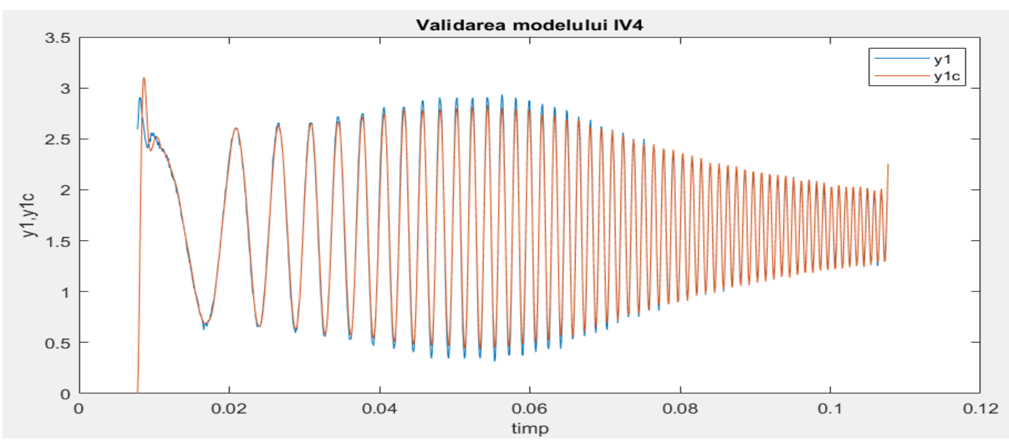
$A(z) = 1 - 1.57z^{(-1)} + 0.7137z^{(-2)}$ $B(z) = 0.1126 + 0.03459z^{(-1)}$

3.2 Validarea modelelor

In aceasta parte a proiectului o sa incercam sa validam o parte din modele obtinute la punctul anterior, algoritmul de validare este indentic pentru toate modele asa ca voi incerca sa validez pentru primul sistem un model. Urmarim sa obtinem o eroare cat mai mica de preferat cea obtinuta in urma funtiei compare pentru toate moedele in parte, eroare o sa o calculam cu ajutorul erorii medii patratice normalizate(EMPN).

Validarea modelului *IV4* pentru primul sistem

Primul pas in '*validarea'* modelului obtinut in urma apelarii metodei IV4 este scoaterea modelelului in spatiu satrilor. Acest lucru l-am facut cu ajutorul functiei MatLab *tf2ss* care a primit ca parametrii cele doua polinoame ale modelului, *B(z)* respectiv *A(z).* Dupa extragerea matricilor sistemului punem sa simulam raspunsul sistemului cu functia *dlsim,* insa daca pentru aceasta simulare calculam si eroarea observam ca valoarea acestea este mult prea mare.



empn = 23.0988

Aceasta problema provine de la faptul ca nu am impus conditiile initiale semanlului calculat. Functia *tf2ss* ne-a returnat spatiul starilor in forma caninca de control, insa o sa dorim sa il transpunem in forma canonica de observare, pentru a ne putea folosi de propietatea FCO-ului de relatie directa intre iesire si intrare $x1(k) = y(k)$.

Ecuatiile din care scoatem conditiile initiale sunt:

$$x(k+1) = Ax(k) + Bu(k)$$

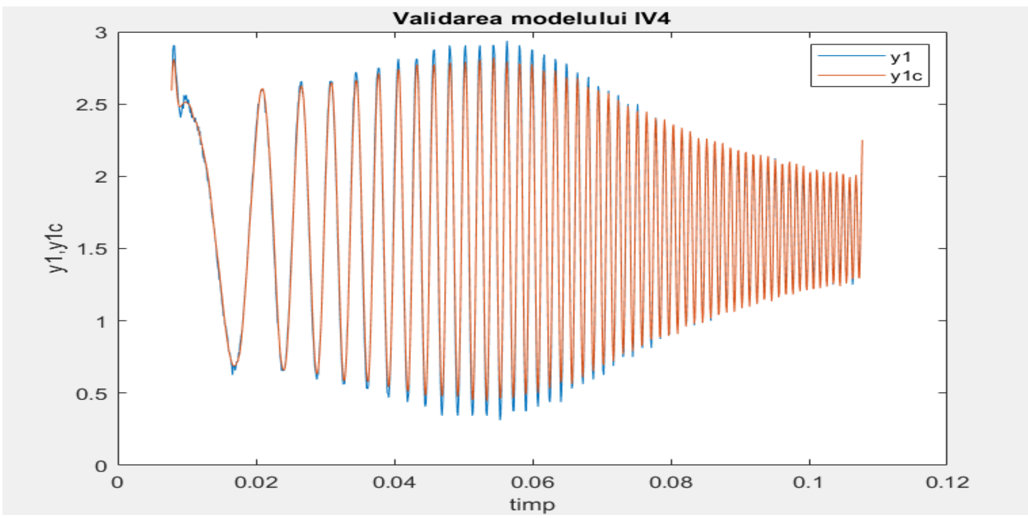
$$x1(k+1) = a11x1(k) + a12x2(k) + b1u(k)$$

$a11, a12$ – elemente ale matricii A in forma de observare.

$$y[k] = Cx[k] + Du[k]$$

$$x1(k) = y(k)$$

Din cea dea doua ecuatie urmarim sa il calculam pe $x2(k)$. Dupa ce am stabilit conditiile putem simula din nou si observam ca sistemul calculat de aceasta data urmareste corect si sistemul dat, iar eroarea este aproximativ identica cu cea data prin *compare.*



empn = 8.0484

//Cod MatLab//

```
[A_ss B_ss C_ss D_ss] = tf2ss(My1_iv4_pem.B,My1_iv4_pem.A);
y1c_iv4 = dlsim(A_ss',C_ss',B_ss',D_ss,u,[y1(1),y1(2)-A_ss(1,1)*y1(1) - C_ss(1,1)*u(1)]);
```