



Proiect

Ingineria Sistemelor (IS)

Nume: Goga Alex-Andrei

Prof.: Mircea Susca

Materie: Ingineria Sistemelor

Cuprins

1.Identificarea sistemelor de ordin II utilizand metoda regresiei liniare.....	3
2.Identificarea sistemelor de ordin II utilizand semnale de tip Chirp.....	11
3.Calibrarea semnalelor pseudoaleatoare binare (SPAB)	19
4.Identificarea sistemelor utilizand metode parametrice	26

1.Identificarea sistemelor de ordin II utilizand metoda regresiei liniare

În acest proiect se realizează identificarea unui sistem dinamic de ordinul II cu poli reali, având un comportament aperiodic amortizat, pe baza datelor obținute prin simulare. Sistemul este modelat printr-o funcție de transfer de forma

$H(s) = \frac{K}{(T_1 s + 1)(T_2 s + 1)}$, unde parametrii K , T_1 și T_2 sunt determinați utilizând metodologia răspunsului la semnal de tip treaptă și metoda regresiei liniare.

Factorul de proporționalitate este calculat din valorile staționare ale semnalelor de intrare și ieșire, constanta de timp dominantă este estimată prin regresie liniară aplicată asupra semnalului de ieșire logaritmat, iar constanta de timp nedominantă este determinată folosind metoda punctului de inflexiune. Validarea modelului se realizează prin compararea răspunsului modelului identificat cu datele experimentale, utilizând indicatorii de performanță eroarea medie pătratică și eroarea medie pătratică normalizată, care confirmă o identificare corespunzătoare a sistemului.

Codul pentru acest proiect:

$$m = 4;$$

$$n = 14;$$

Parametrii m și n sunt valori individuale atribuite fiecarui student. Aceștia sunt utilizați pentru generarea parametrilor sistemului astfel încât fiecare experiment să fie unic.

$$\begin{aligned}
c1 &= (1000+n*300)/10000; \\
c2 &= (1.15+2*(m+n/10)/20); \\
a1 &= 2*c2*c1; \\
a2 &= c1; \\
b0 &= (1.2+m+n)/5.5;
\end{aligned}$$

Acești parametri definesc dinamica internă a procesului:

- **c1 și c2 sunt coeficienți de scalare calculați pe baza lui m și n;**
- **a1 și a2 sunt parametri derivați care descriu comportamentul dinamic;**
- **b0 influențează câștigul și limitele de saturatie ale ieșirii.**

Acești parametri sunt fixați de enunț și nu trebuie modificați.

$$rng(m+10*n)$$

$$\begin{aligned}
x0_slx &= [2*(m/2+rand(1)*m/5); \\
&\quad m*(n/20+rand(1)*n/100)];
\end{aligned}$$

rng(m+10*n) fixează generatorul de numere aleatoare pentru a obține rezultate reproductibile;

x0_slx reprezintă vectorul condițiilor inițiale ale stărilor sistemului hidraulic;

Valorile sunt generate aleator în jurul unor valori dependente de m și n.

$$Ts = 10*c2/c1/1e4*1.5;$$

$$Tfin = 30*c2/c1;$$

- **Ts este pasul fundamental de eşantionare al simulării;**
- **Tfin reprezintă timpul total de simulare, ales suficient de mare pentru atingerea regimului staţionar.**

$$gain = 10;$$

$$umin = 0;$$

$$umax = gain;$$

$$ymin = 0;$$

$$ymax = b0*gain/1.5;$$

Semnalul de intrare este limitat între umin și umax. Semnalul de ieșire este limitat între ymin și ymax, în funcție de parametrii sistemului. Aceste saturații modelează limitările fizice ale sistemului real.

$$whtn_pow_in = ...;$$

$$whtn_Ts_in = Ts^*3;$$

$$q_in = (umax-umin)/pow2(10);$$

$$whtn_pow_out = ...;$$

$$whtn_Ts_out = Ts^*5;$$

$$q_out = (ymax-ymin)/pow2(9);$$

Se adaugă zgomot alb atât la intrare, cât și la ieșire pentru a simula condiții reale. q_{in} și q_{out} modelează cuantizarea DAC și ADC. Rezoluția este de 10 biți la intrare și 9 biți la ieșire.

$$u0 = 0;$$

$$ust = 3;$$

$$t1 = 10*c2/c1;$$

Se aplică un semnal de tip treaptă;
 $u0$ este valoarea inițială;
 ust este amplitudinea treptei;
 $t1$ reprezintă momentul aplicării treptei.

$$u0 = \text{mean}(u(i1:i2));$$

$$ust = \text{mean}(u(i3:i4));$$

$$y0 = \text{mean}(y(i1:i2));$$

$$yst = \text{mean}(y(i3:i4));$$

$$K = (yst - y0) / (ust - u0);$$

Câștigul static este calculat folosind valorile medii ale semnalelor în regim staționar:

$$K = \frac{\Delta y}{\Delta u}$$

Această metodă reduce influența zgomotului de măsurare.

```
i5=6626;  
i6=8976;  
t_reg=t(i5:i6);  
y_reg=log(yst-y(i5:i6));  
figure;plot(t_reg,y_reg);
```

i5:i6 este intervalul ales pe porțiunea de “cădere exponentială” (după treaptă).

t_reg reține timpul în acel interval.

y_reg = log(yst - y(...)) → logaritmezi “distanța până la staționar”.

Afișezi graficul ca să vezi dacă punctele sunt aproximativ pe o linie (condiție pentru regresie bună).

```
A_reg=[sum(t_reg.^2),sum(t_reg);  
sum(t_reg),length(t_reg)];  
b_reg=[sum(y_reg.*t_reg);sum(y_reg)];  
theta=inv(A_reg)*b_reg;
```

A_reg este matricea normalelor ($\mathbf{X}^T \mathbf{X}$) pentru regresie cu termen liniar + termen constant.

- **sum(t_reg.^2)** corespunde $\sum t^2$,
- **sum(t_reg)** corespunde $\sum t$,
- **length(t_reg)** corespunde numărului de eşantioane.

b_reg este $(X^T y)$:

- **sum(y_reg.*t_reg)** corespunde $\sum y t$,
- **sum(y_reg)** corespunde $\sum y$.

theta = inv(A_reg)*b_reg calculează coeficienții θ .

- **theta(1) = panta, theta(2) = intercept.**

```
i7=6667;
i8=7053;
Ti=t(i8)-t(i7);
T2vec=0.1:0.1:3.5;
Fun=Ti*T2vec.*log(T2vec)-T2vec*(Ti*T1*log(T1))+T1*Ti;
```

*figure
plot(T2vec, Fun);*

T2=0.553;

- **i7 este indicele aproximativ al momentului când începe treapta.**
- **i8 este indicele aproximativ al punctului de inflexiune observat pe graficul ieșirii.**
- **$T_i = t(i8) - t(i7)$ este timpul până la inflexiune măsurat de la aplicarea treptei.**

T2vec este un vector de valori candidate pentru T_2 (scanare brută).

Fun construiește funcția din ecuația transcendentă, evaluată pentru fiecare T2vec.

plot(T2vec, Fun) îți arată unde funcția trece prin zero → acolo e soluția.

T2=0.553; alegi manual valoarea unde graficul indică rădăcina.

$$H = tf(K, [T1 * T2 \quad T1 + T2 \quad 1]);$$

$$ysim = lsim(H, u, t);$$

Ieșirea simuată este comparată cu ieșirea reală pentru a evalua calitatea identificării.

$$sys = ss(A, B, C, D);$$

$$ysim2 = lsim(sys, u, t, [y(1), 10]);$$

Modelul este validat și în reprezentare stare-spațiu, folosind condiții initiale nenule.

$$J = 1/sqrt(length(t)) * norm(y - ysim2);$$

$$eMPN = norm(y - ysim2) / norm(y - mean(y)) * 100;$$

J reprezintă o măsură RMS a erorii.eMPN exprimă eroarea relativă procentuală. Valorile obținute confirmă calitatea modelului identificat.

- Eroarea identificata este de **5.8153%** ceea ce confirma o identificare corecta a modelului.

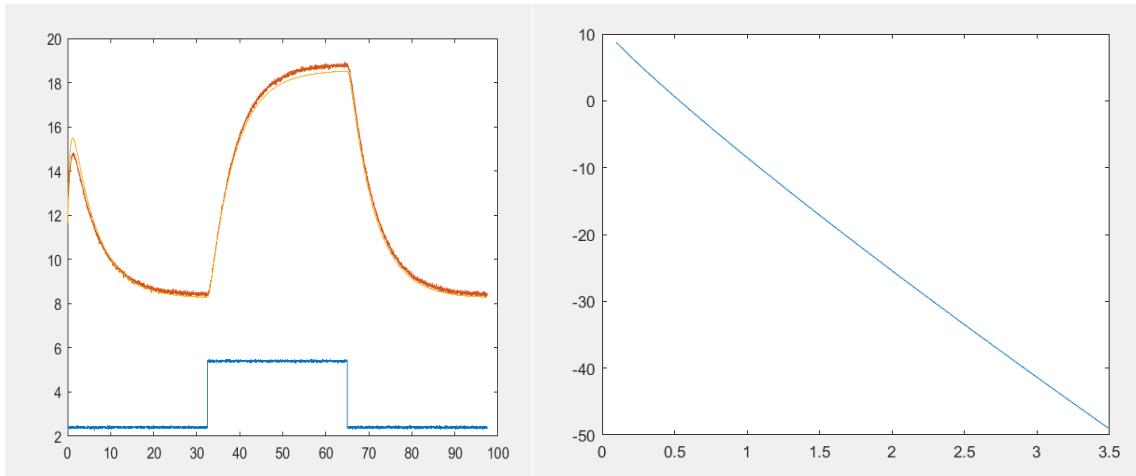


Fig.1.Metoda regresiei liniare

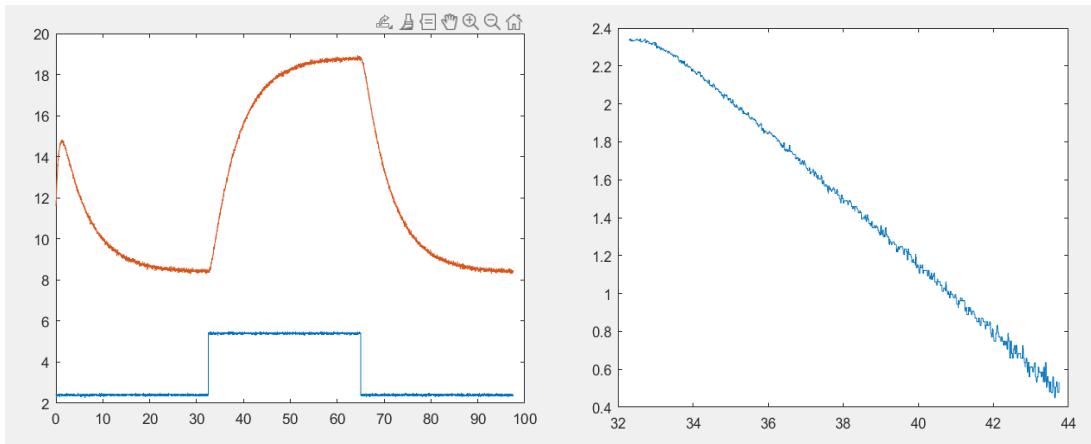


Fig.2.Semnalul de start

2.Identificarea sistemelor de ordin II utilizând semnale de tip Chirp

În cadrul acestui proiect se realizează identificarea unui sistem dinamic de ordinul II cu poli reali, utilizând un semnal de intrare de tip Chirp și metoda estimării răspunsului în frecvență. Procesul analizat este cel hidraulic cu două rezervoare inseriate, modelat prin funcția de transfer

$$H(s) = \frac{K}{(T_1 s + 1)(T_2 s + 1)}, \text{ unde } K, T_1 \text{ și } T_2 \text{ sunt parametrii necunoscuți.}$$

Semnalul Chirp este generat în jurul punctului de funcționare specificat, cu o variație controlată a frecvenței, astfel încât să permită excitarea dinamicii sistemului pe o bandă largă de frecvențe.

Identificarea parametrilor se bazează pe analiza relației de fază și amplitudine dintre semnalele de intrare și ieșire, în special în zona în care faza atinge valoarea de $-\pi/2$. Factorul de proporționalitate este determinat din raportul amplitudinilor semnalelor sau din valorile staționare, iar pe baza modulului și fazei funcției de transfer se deduc constantele de timp ale sistemului. Modelul obținut este validat prin compararea răspunsului simulat cu cel al sistemului real, folosind indicatorii de performanță eroarea medie pătratică și eroarea medie pătratică normalizată, care confirmă o bună concordanță între model și datele experimentale.

Codul pentru acest proiect:

$$c1 = (1000+n*300)/10000;$$

$$c2 = (1.15+2*(m+n/10)/20);$$

$$a1 = 2*c2*c1;$$

$$a2 = c1;$$

$$b0 = (1.2 + m + n) / 5.5;$$

c1 și c2 sunt coeficienți calculați din m, n (4,14).

a1, a2 sunt parametri derivați (Simulink)

b0 este un factor care influențează nivelul ieșirii (și implicit saturarea maximă de ieșire).

$$rng(m+10*n)$$

$$x0_slx = [2 * (m/2 + rand(1) * m/5); m * (n/20 + rand(1) * n/100)];$$

rng(m+10*n) fixează “seed”-ul generatorului aleator → simularea e reproductibilă pentru același m,n.

rand(1) generează un număr aleator uniform în [0,1].

x0_slx este un vector 2×1 cu stări inițiale pentru modelul Simulink

$$Ts = 10 * c2 / c1 / 1e4 * 1.5;$$

$$Tfin = 30 * c2 / c1 * 10;$$

Ts este pasul fundamental de eşantionare (sampling time).

Tfin este timpul final al simulării.

Observație: aici este *10 față de varianta anterioară

```

gain = 10;
umin = 0; umax = gain;
ymin = 0; ymax = b0*gain/1.5;

```

gain – scalare internă.

umin, umax limitează intrarea (model fizic: actuator limitat).

ymin, ymax limitează ieșirea (model fizic: senzor/rezervor limitat).

```
whtn_pow_in = 1e-6*5*((m-1)*8+n/2)/5)/2*6/8; % input white noise power and
sampling time
```

```
whtn_Ts_in = Ts*3;
```

```
whtn_seed_in = 23341+m+2*n;
```

```
q_in = (umax-umin)/pow2(10); % input quantizer (DAC)
```

```
whtn_pow_out = 1e-5*5*((m-1)*25+n/2)/5)*6/80*(0.5+0.3*(m-2)); % output
white noise power and sampling time
```

```
whtn_Ts_out = Ts*5;
```

```
whtn_seed_out = 23342-m-2*n;
```

```
q_out = (ymax-ymin)/pow2(9); % output quantizer (ADC)
```

whtn_pow_in = puterea zgomotului alb la intrare.

whtn_Ts_in = pasul cu care se actualizează zgomotul de intrare.

whtn_seed_in = seed pentru zgomotul de intrare.

q_in = quantizare DAC cu 10 biți (2^{10} nivele).

whtn_pow_out = puterea zgomotului la ieșire (zgomot de măsurare).

whtn_Ts_out = actualizare zgomot ieșire mai rară.

whtn_seed_out = seed ieșire.

q_out = cuantizare ADC pe 9 biți (2^9 nivele).

$T1=4;$

$wf=1/T1;$

$fmin=wf/2/pi/10;$

$fmax=wf/2/pi*10;$

$Ain=1.5;$

T1=4; – aici îl folosești ca o valoare inițială/estimare grosieră pentru o constantă de timp.

wf=1/T1; – definești o frecvență caracteristică în rad/s (aprox. inversul constantei de timp).

fmin=wf/2/pi/10; – transformi din rad/s în Hz împărțind la 2π , apoi împărți la 10 → frecvența minimă (de scanare sau setare a sinusului).

fmax=wf/2/pi*10; – frecvența maximă, de 10 ori mai mare (interval logaritmic aproximativ).

Ain=1.5; – amplitudinea semnalului de intrare (sinus). Trebuie aleasă astfel încât să nu satureze semnalele.

$$yst = (9.18 + 7.68)/2;$$

$$ust = 2.37;$$

$$K = 3.499;$$

yst – calculezi o valoare medie a ieșirii pe un interval.

ust – valoarea medie a intrării (offset / nivel de lucru).

K – câștigul static

$$wI = pi / (391.28 - 388.39);$$

$$\Delta T_1 = 290.291 - 288.877;$$

$$\phi_1 = \text{rad2deg}(-wI * \Delta T_1);$$

391.28-388.39 reprezintă o diferență de timp dintre două puncte caracteristice consecutive (de regulă două maxime consecutive sau două treceri prin zero). ΔT_1 e întârzierea temporală dintre intrare și ieșire (diferență între două momente “corespondente” – de ex. vârf intrare vs vârf ieșire). $\phi_1 = \text{rad2deg}(-wI * \Delta T_1)$ calculează fază: $\phi = -\omega \Delta T$ iar rad2deg convertește din radiani în grade.

$$Ay = (18.6326 - 16.4527)/2;$$

$$Au = (3.9707 - 1.85547)/2;$$

$$M = Ay/Au;$$

Ay e amplitudinea iesirii si se calculeaza $y_{max} - y_{min}/2$.

Au e amplitudinea intrării si se calculeaza $u_{max} - u_{min}/2$.

M = Ay/Au este modulul răspunsului în frecvență la acea frecvență.

$$Im=-M;$$

$$wn=wI;$$

$$zeta=-K/2/Im;$$

Im=-M setează “partea imaginară”

wn = w1 iezi frecvența măsurată ca fiind frecvența naturală

zeta = -K/2/Im calculează amortizarea pe baza unei relații asumate între K, partea imaginară și ζ .

$$H=tf(K*wn^2,[1,2*zeta*wn,wn^2]);$$

$$zpk(H)$$

Construirea functiei de transfer de ordin 2 iar zpk afiseaza sistemul in forma zero-uri poli castig.

$$T1=1/0.1485;$$

$$T2=1/6.9;$$

Definești două constante de timp T1 si T2 ca inverse ale unor rate(dupa poli 1/p).

$$\begin{aligned}
A &= [0, 1; \\
&\quad -1/T1/T2, -(1/T1+1/T2)]; \\
B &= [0; K/T1/T2]; \\
C &= [1 \ 0]; \\
D &= 0;
\end{aligned}$$

Construiești o realizare în spațiul stărilor.

$$ysim2 = lsim(sys, u, t, [y(1), 25]);$$

Isim simulează ieșirea modelului sys la intrarea u(t) pe timpul t.Ultimul argument [y(1),25] este vectorul condițiilor inițiale ale stărilor: prima stare pornește din y(1) ca să pornească ieșirea simulatului de la valoarea inițială reală,a doua stare e setată la 25(inceputul semnalului).

$$J = 1/sqrt(length(t)) * norm(y - ysim2);$$

$$eMPN = norm(y - ysim2) / norm(y - mean(y)) * 100;$$

norm(y-ysim2) calculează norma euclidiană a erorii .J este echivalent cu o măsură RMS (eroare medie pătratică sub radical).

eMPN este eroarea normalizată procentual de 17.3278%.

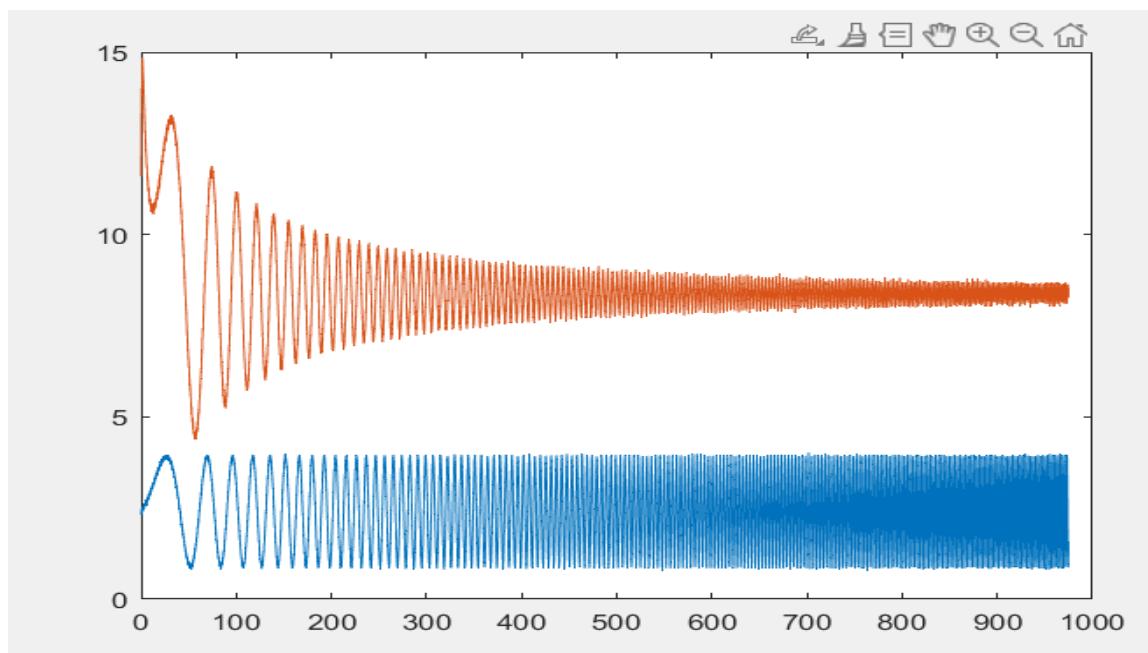


Fig. 3 Semnalul dat

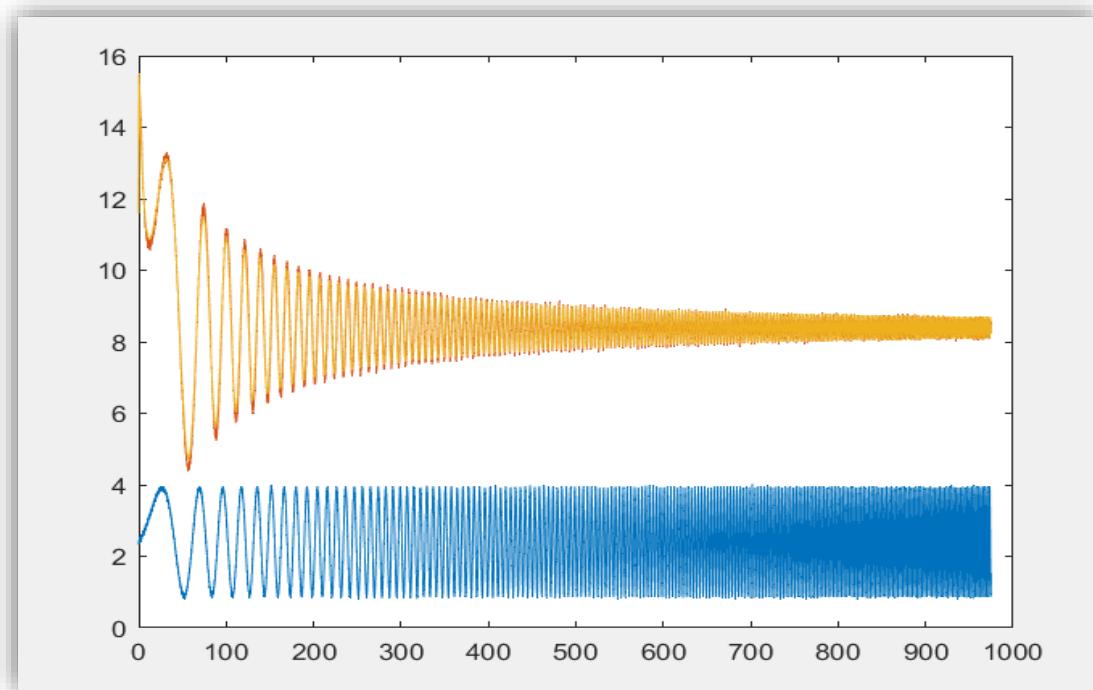


Fig.4 Identificarea sistemelor de ordin 2 utilizand semnalele de tip Chirp

3.Calibrarea semnalelor pseudoaleatoare binare (SPAB)

În cadrul acestui proiect se realizează proiectarea și calibrarea unui semnal de intrare de tip pseudoaleator binar (SPAB) pentru identificarea unui sistem dinamic neliniar, utilizând metode parametrice. Sunt analizate procese precum convertorul DC-DC ridicător de tensiune sau pendulul simplu acționat cu motor electric, ambele caracterizate prin neliniarități statice sau dinamice. Scopul este obținerea unui model aproximativ liniar al sistemului în vecinătatea unor puncte de funcționare bine definite.

Semnalul de intrare este compus dintr-un semnal trapezoidal, care stabilește două puncte de funcționare apropriate (pentru identificare și validare), peste care se suprapune un semnal SPAB de amplitudine redusă. Rolul semnalului SPAB este de a excita sistemul pe o bandă largă de frecvențe, fără a părăsi regiunea de funcționare aproape liniară. Parametrii semnalului sunt aleși conform teoriei prezentate în curs, iar semnalul generat este utilizat într-un singur experiment Simulink, asigurând obținerea unor date adecvate pentru identificare și validare ulterioară.

Cod proiect:

$$Ts = 500e-6; \text{ % fundamental step size}$$

$$Ts = 500e-6 \text{ secunde} = 0.0005 \text{ s.}$$

Este pasul fundamental (sampling time) folosit în simulare și în generarea semnalului.

$$u_star = 1.2 + n * 0.075;$$

$$delta = 0.125;$$

$$\text{delta_spab} = 0.075;$$

u_star este punctul de lucru (offset-ul intrării).

delta este o amplitudine “generală” a variației intrării.

delta_spab este amplitudinea specifică pentru treptele/nivelurile SPAB (de obicei mai mică decât delta ca să rămână în limite și să fie “sigură”).

$$umin = -5; umax = 5; \text{ \% input saturation}$$

$$ymin = -100; ymax = 100; \text{ \% output saturation}$$

Intrarea actuatorului este limitată la [-5,5]

Ieșirea senzorului este limitată la [-100,100]

Aceste limite simulează constrângeri reale.

$$g = 9.81;$$

g accelerația gravitațională (m/s²), folosită în ecuațiile pendulului.

$$M = 0.8-n/48;$$

$$l = 1.2-m/24;$$

$$b = 0.3+m/24;$$

M masa echivalentă care scade cu n.

I lungimea pendulului, scade cu m.

b coeficient de frecare/amortizare, crește cu m.

Aceste relații sunt “presetate” în proiect ca să personalizeze dinamica.

$$c1 = 180/\pi;$$

$$c2 = 4+n/2;$$

c1 = 180/pi este factor de conversie radian → grade.

c2 = 4 + n/2 este un factor suplimentar (de scalare/offset/calibrare) definit de proiect pentru lanțul de măsurare.

% input white noise power and sampling time

$$whtn_pow_in = 1e-10*(Ts*1e4)/2;$$

$$whtn_Ts_in = Ts^2;$$

$$whtn_seed_in = 23341+m+2*n;$$

$$q_in = (umax-umin)/pow2(13); \text{ \% input quantizer (DAC)}$$

% output white noise power and sampling time

$$whtn_pow_out = 1e-3*Ts;$$

$$whtn_Ts_out = Ts^2;$$

$$whtn_seed_out = 23342-m-2*n;$$

$$q_out = (ymax-ymin)/pow2(13); \text{ \% output quantizer (ADC)}$$

whtn_pow_in definește puterea zgomotului alb adăugat pe intrare.

- expresia include Ts ca să scaleze zgomotul cu rata de eșantionare.

whtn_Ts_in = $Ts^2 \rightarrow$ zgomotul intrării se actualizează la fiecare 2 eșantioane.

whtn_seed_in seed pentru zgomotul intrării.

q_in este cuantizarea DAC pe 13 biți:

- $\text{pow2}(13)=8192$ nivele,
- pasul de cuantizare este $(umax - umin)/8192$.

whtn_pow_out puterea zgomotului alb pe măsurare (ieșire).

whtn_Ts_out actualizare la fiecare 2 eșantioane.

whtn_seed_out seed ieșire.

q_out cuantizare ADC pe 13 biți în intervalul $[ymin, ymax]$.

meas_rep = round(7+n/2); % data acquisition hardware sampling limitation

meas_rep reprezintă faptul că hardware-ul nu poate salva la fiecare eșantion, ci “reține”/loghează la fiecare **meas_rep** eșantioane (sub-eșantionare).

round(7+n/2) îți dă o valoare între ~7 și mai mare, în funcție de n.

$tI=20;$

$tu=0.9;$

$tpo=1.7;$

t1=20 – momentul (sau timpul) după care începe efectiv secvența

tu=0.9 – timp urcare

tpo=1.7 – o perioadă/constantă caracteristică folosită la dimensionarea pașilor SPAB.

%calibrare SPAB

$N=4;$

$p=round(tpo/N/Ts);$

$DeltaT=(2^N-1)*p*Ts*3;$

N este numarul de biti.

p=round(tpo/N/Ts);

- p este numărul de eșantioane (samples) menținut pentru fiecare “chip”/nivel.
- tpo/N îți dă o durată țintă per nivel (în secunde),
- împărțirea la Ts o transformă în eșantioane,
- round face p întreg

DeltaT=(2^N-1)*p*Ts*3;

- $(2^N - 1)$ este lungimea unei secvențe MLS/PRBS pentru ordinul N.
- p^*T_s este durata unui nivel.
- înmulțirea cu 3 înseamnă că rulezi 3 perioade ale secvenței (sau o întinzi de 3 ori).

```
[input_LUT_dSPACE,Tfin] =
generate_input_signal(Ts,t1,DeltaT,N,p,u_star,delta,delta_spab);
```

Apelăzi o funcție auxiliară generate_input_signal(...) care:

Construiește secvența SPAB în timp discret.

Aplică offset-ul u_star.

Scalează amplitudinea cu delta și/sau delta_spab.

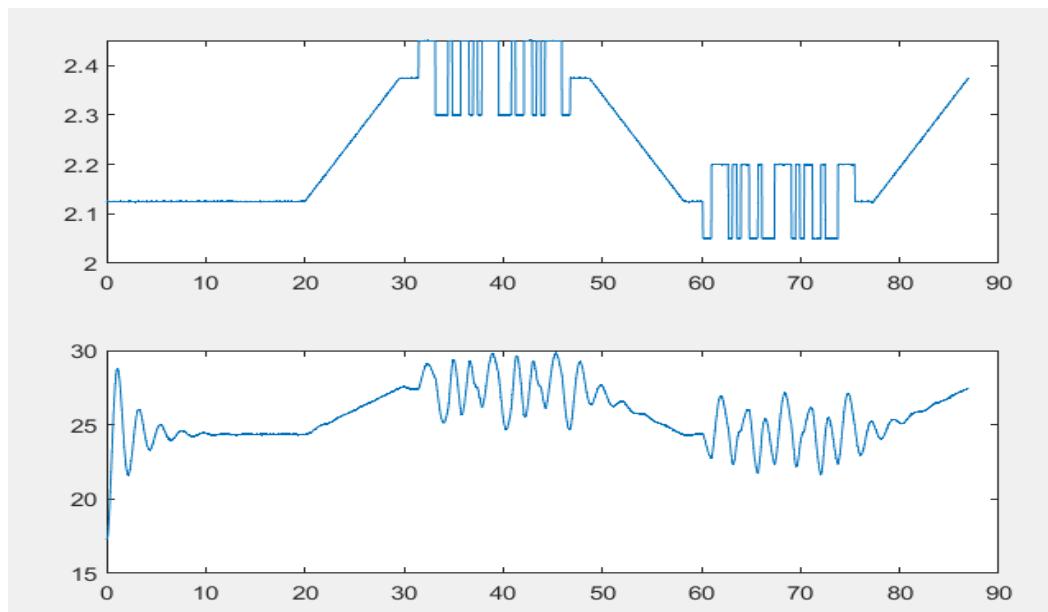


Fig 5. Calibrare SPAB

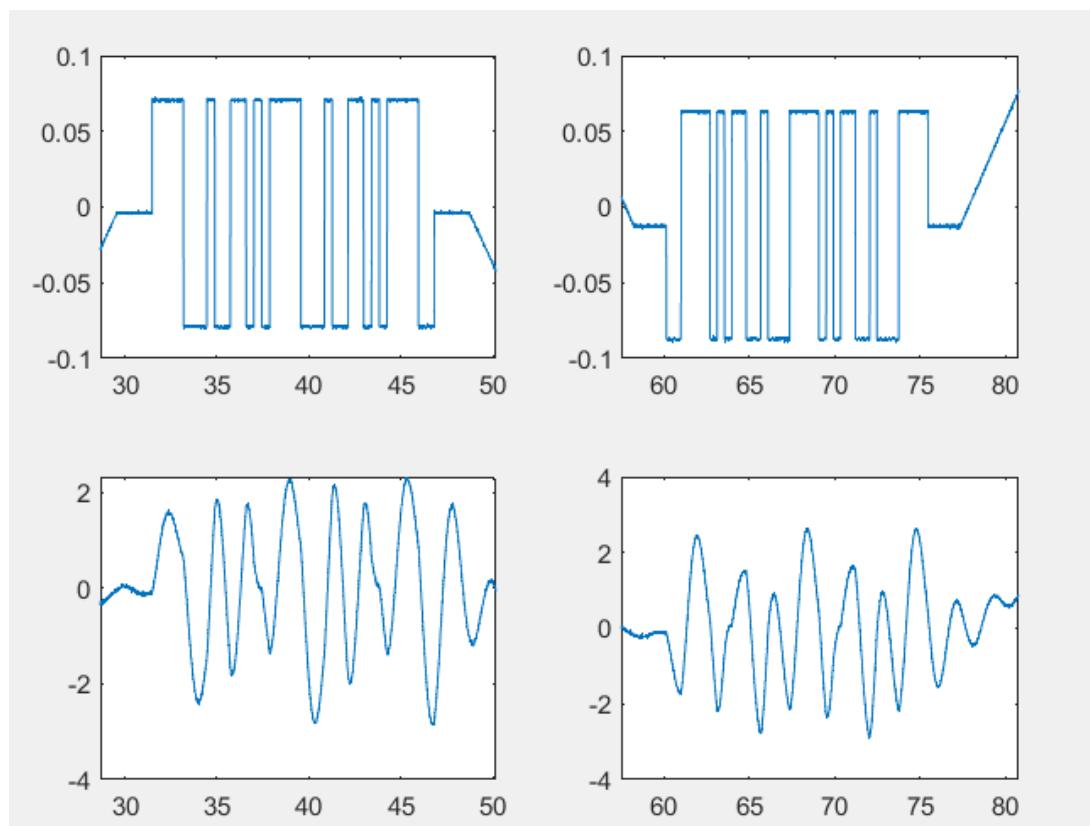


Fig.6 Calibrare SPAB

4.Identificarea sistemelor utilizand metode parametrice

În cadrul acestei etape, s-a realizat achiziția și pre-procesarea datelor experimentale pentru un pendul mecanic, utilizând un semnal de intrare de tip trapezoidal suprapus peste un semnal pseudoaleator binar (SPAB). Pentru a asigura o identificare corectă, datele brute au fost segmentate în două subseturi distincte: unul pentru identificare (între indecșii i1 și i2) și unul pentru validare (între i3 și i4), ambele fiind stocate sub formă de obiecte de tip iddata. O măsură critică de rafinare aplicată a fost eliminarea componentelor continue prin scăderea mediei (mean) și aplicarea unei decimări pentru a combate suprașantionarea și a adapta datele la dinamica sistemului, conform recomandărilor metodologice.

Folosind toolbox-ul de identificare din MATLAB, au fost estimate mai multe structuri de modele pentru a îndeplini criteriile de performanță solicitate. S-au construit modele polinomiale de tip **ARX** și **ARMAX** (validate prin testul de autocorelație), modele de tip **IV4** și **OE** (validate prin intercorelație), precum și modele în spațiul stărilor utilizând algoritmii **N4SID** și **SSEST**. Validarea fiecărui model a fost realizată prin analiza reziduurilor cu funcția resid și prin evaluarea gradului de suprapunere (FIT) folosind funcția compare, urmărindu-se obținerea unei precizii cât mai apropiate de 100% pentru a confirma viabilitatea modelului matematic obținut.

Erorile obținute în urma modelelor polinomiale:

Eroare ARMAX – 90.96%

Eroare OE – 93.86%

Eroare N4SID – 93.04%

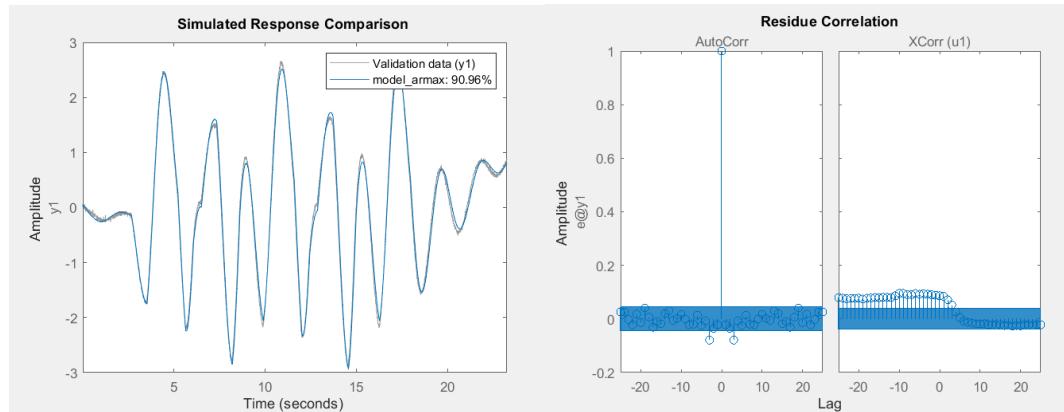


Fig.7 ARMAX autocorelatie

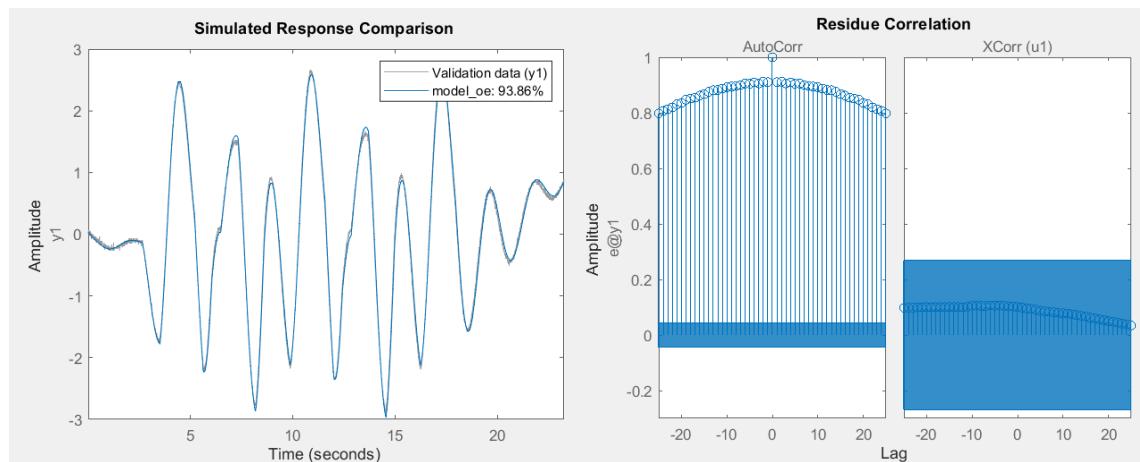


Fig.8 OE intercorelatie

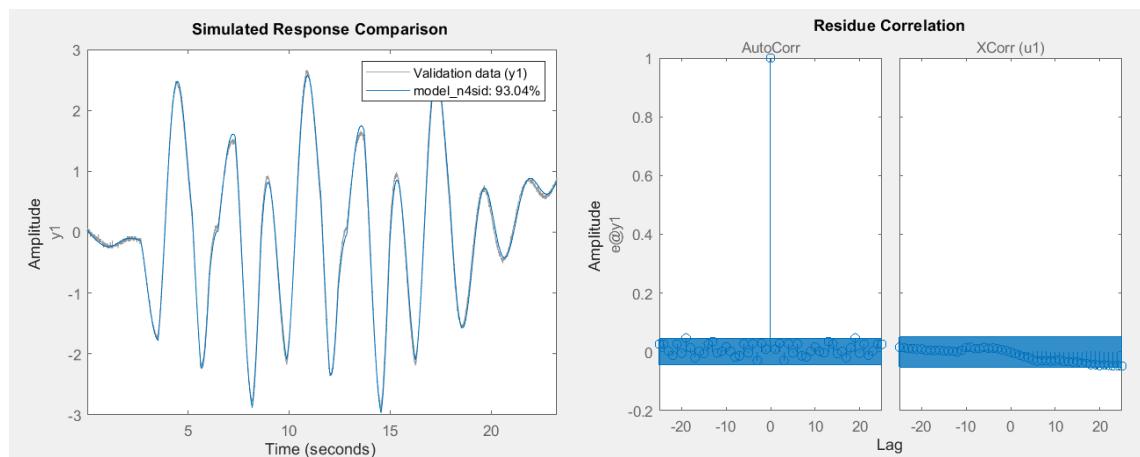


Fig.7 N4SID

Cod proiect:

%% armax

*model_armax=armax(dat_id,[2 2 3 4]) %luam valori la nc de la 1 pana la ...
figure,resid(model_armax,dat_vd)
figure,compare(model_armax,dat_vd)*

Ecuatia este: $A(q-1)^*y(t)=B(q-1)^*u(t-nk)+C(q-1)^*e(t)$.

na,nb,nc ordinul polinoamelor.

nk – intarizarea.

Modelul ARMAX permite modelarea zgomotului colorat.

%% oe

*model_oe=oe(dat_id,[2 2 1])
figure,resid(model_oe,dat_vd)
figure,compare(model_oe,dat_vd)*

Ecuatia este: $y(t)=B(q-1)/F(q-1)^*u(t-nk)+e(t)$

B/F este modelul determinist,

e(t) este eroare aditivă

Modelul OE separa dinamica de zgomot,eroarea fiind doar la iesire.

%% n4sid

model_n4sid=n4sid(dat_id,1:15)

figure,resid(model_n4sid,dat_vd)

figure,compare(model_n4sid,dat_vd)

Se identifica direct prin spatial starilor:

$$x_{k+1} = Ax_k + Bu_k, y_k = Cx_k + Du_k + v_k$$

Este o metodă robustă pentru sisteme MIMO/SISO și pentru ordine mai mari.