

**Dan Ștefănoiu**

**Ion Matei**

**Petre Stoica**

**Aspecte practice în  
Modelarea și Identificarea Sistemelor**

## Cuprins

<b>Notatii și abrevieri .....</b>	<b>VII</b>
<b>Prefață .....</b>	<b>IX</b>
<b>Introducere .....</b>	<b>1</b>
<b>1. Caracterizări în timp și frecvență ale proceselor stocastice .....</b>	<b>11</b>
<b>1.1. Analize de proces prin metode ne-parametrice .....</b>	<b>11</b>
A. Analiza tranzitorie .....	11
B. Analiza în frecvență .....	11
C. Analiza bazată pe corelație .....	13
D. Analiza spectrală .....	14
<b>1.2. Aspecte practice în analiza proceselor stocastice .....</b>	<b>15</b>
A. Procese total neautocorelate – zgomotul alb .....	15
B. Zgomote colorate .....	17
<b>1.3. Exerciții .....</b>	<b>18</b>
<b>1.4. Probleme de simulare .....</b>	<b>20</b>
<b>2. Identificarea modelelor ne-parametrice .....</b>	<b>25</b>
2.1. Contextul general de lucru .....	25
2.2. Exerciții .....	26
2.3. Probleme de simulare .....	26
<b>3. Identificare parametrică prin Metoda Celor Mai Mici Pătrate .....</b>	<b>34</b>
3.1. Contextul general de lucru .....	34
3.2. Exerciții .....	34
3.3. Probleme de simulare .....	35
<b>4. Identificare parametrică prin Metoda Variabilelor Instrumentale .....</b>	<b>39</b>
4.1. Contextul general de lucru .....	39
A. Metoda Variabilelor Instrumentale .....	39
B. Criterii de alegere a structurii modelelor .....	40
C. Criterii de validare a modelelor .....	44
4.2. Exerciții .....	46
4.3. Probleme de simulare .....	48

<b>5. Identificare parametrică prin Metoda Minimizării Erorii de Predicție</b>	<b>58</b>
5.1. Contextul general de lucru	58
A. Metoda Celor Mai Mici Pătrate Extinsă	58
B. Metoda Minimizării Erorii de Predicție	60
5.2. Exerciții	62
5.3. Probleme de simulare	63
<b>6. Identificare recursivă</b>	<b>67</b>
6.1. Contextul general de lucru	67
A. Algoritmi recursivi de identificare	67
B. Rutine MATLAB pentru identificare recursivă	75
6.2. Exerciții	76
6.3. Probleme de simulare	77
<b>7. Aplicații de identificare recursivă</b>	<b>81</b>
7.1. Contextul general de lucru	81
A. Aproximarea modelelor complexe	81
B. Identificarea parametrilor fizici ai unui proces	87
7.2. Exerciții	95
7.3. Probleme de simulare	98
<b>8. Modelarea și predicția seriilor de timp</b>	<b>104</b>
8.1. Contextul general de lucru	104
A. Estimarea modelului polinomial al tendinței	105
B. Estimarea componentei sezoniere	107
C. Estimarea componentei nedeterminate (aleatoare)	113
D. Predicția seriei de timp	115
8.2. Exerciții	116
8.3. Probleme de simulare	117
<b>Anexe</b>	
A. Despre biblioteca de rutine MATLAB dedicate Identificării Sistemelor	129
B. Lista de verificare a mini-simulatoarelor și rutinelor de MIS	134
<b>Referințe bibliografice</b>	<b>137</b>

## Lista figurilor

1. <i>Reprezentarea sistemică a modelelor ARMAX.</i>	3
2. <i>Experimentul obținerii culorii albe din spectrul ROGVAIV.</i>	16
3. <i>Experimentul obținerii unei nuanțe de roz din spectrul ROGVAIV.</i>	16
4. <i>Două modele de procese stocastice echivalente.</i>	20
5. <i>Fereastra grafică tipică a rutinei ISLAB_1A.</i>	21
6. <i>Fereastra grafică tipică a rutinei ISLAB_1B.</i>	21
7. <i>Fereastra grafică tipică a rutinei NOISE.</i>	22
8. <i>Exemplu de analiză tranzitorie.</i>	27
9. <i>Exemplu de analiză pe bază de corelație.</i>	28
10. <i>Fereastra spectrală a lui Hamming.</i>	30
11. <i>Exemplu de analiză spectrală.</i>	31
12. <i>Exemplu de afișare a erorii de estimare cu MCMMP (răspuns în frecvență).</i>	36
13. <i>Exemplu de afișare a erorii de estimare cu MCMMP (dispersie zgomot).</i>	36
14. <i>Criterii de alegere a structurii modelelor.</i>	41
15. <i>Performanțele unui model estimat cu MCMMP.</i>	55
16. <i>Reprezentarea poli-zeroruri a unui model estimat cu MCMMP.</i>	55
17. <i>Dispersia estimată a zgomotului. Criteriul aplatizării și Testul F.</i>	56
18. <i>Potrivirea cu datele de identificare.</i>	56
19. <i>Potrivirea cu datele de validare.</i>	57
20. <i>Criteriul Akaike-Rissanen.</i>	57
21. <i>Performanțele unui model estimat cu MMEP.</i>	66
22. <i>Reprezentarea poli-zeroruri a unui model estimat cu MMEP.</i>	66
23. <i>Algoritmul recursiv de bază în IS.</i>	68
24. <i>Algoritmul recursiv cu fereastră dreptunghiulară.</i>	70
25. <i>Algoritmul recursiv cu fereastră exponențială.</i>	72

26. Algoritmi recursivi de tip gradient. ....	73
27. Algoritmul recursiv cu filtrare Kalman. ....	74
28. Performanțele MCMMP-R. ....	79
29. Performanțele MVI-R. ....	79
30. Performanțele MMEP-R. ....	80
31. Performanțele MRLP-R. ....	80
32. Caracteristicile în frecvență ale filtrelor Butterworth de tip trece-jos. ....	83
33. Caracteristicile în frecvență ale filtrelor Butterworth de tip trece-bandă. ....	83
34. O estimare grosieră a spectrului procesului furnizor de date. ....	101
35. Caracteristicile filtrului Butterworth ales. ....	101
36. Performanțele modelului ARMAX pe toată lărgimea de bandă. ....	102
37. Performanțele modelului ARMAX pe lărgimea de bandă a filtrului. ....	102
38. Date de intrare-ieșire furnizate de un motor de curent continuu. ....	103
39. Ieșirea măsurată și cea simulată ale motorului de curent continuu. ....	103
40. Urmărirea parametrilor fizici ai motorului de curent continuu. ....	103
41. Determinarea perioadei optime cu Metoda Wittacker-Robinson. ....	109
42. Determinarea perioadei optime cu Metoda periodogramei Schuster. ....	112
43. Algoritmul Levinson-Durbin. ....	114
44. Rata lunară a numărului de șomeri din SUA. ....	121
45. Circulația monedei belgiene măsurată lunar, timp de 10 ani. ....	121
46. Media lunară a numărului de pete solare observate. ....	122
47. Distanța lunară parcursă la U.K. Airlines pe cursele interne. ....	122
48. Rata lunară a șomajului în Marea Britanie. ....	123
49. Rata lunară a șomajului în Franța. ....	123

---

50. Rata lunară a șomajului în Canada. ....	124
51. Veniturile lunare din impozitele pe telefoane în SUA. ....	124
52. Media lunară a timpului mediu de lucru săptămînal în SUA. ....	125
53. Numărul lunar al bolnavilor operați de amigdalită la Spitalul 23 August. ....	125
54. Intensitatea conștiinței colective pe Terra măsurată lunar. ....	126
55. Intensitatea radio cosmică măsurată la radio-telescopul din Indianapolis. ....	126
56. Cursul de schimb USD-LEI (eșantionare neuniformă). ....	127
57. Cursul de schimb EURO-LEI (eșantionare neuniformă). ....	127
58. Cursul de schimb USD-EURO (eșantionare neuniformă). ....	128
59. Fereastra grafică tipică a interfeței bibliotecii de IS. ....	131

## Lista tabelelor

<b>1. Intervale și nivele de încredere tipice pentru validarea modelelor.</b> .....	<b>45</b>
<b>2. Serii de timp disponibile pe Discul Compact.</b> .....	<b>120</b>

## Notatii matematice specifice

$\mathcal{F}$

Operatorul Fourier, definit prin:

$$X(e^{j\omega}) \stackrel{\text{def}}{=} \mathcal{F}(x)(e^{j\omega}) \stackrel{\text{def}}{=} \sum_{n \in \mathbb{Z}} x[n] e^{-j\omega n}, \quad \forall \omega \in \mathbb{R},$$

pentru orice secvență discretă de semnal, absolut sumabilă,  $x$ .

$\mathcal{F}^{-1}$

Operatorul Fourier invers, exprimat prin:

$$x[n] = \mathcal{F}^{-1}(X)[n] \stackrel{\text{def}}{=} \frac{1}{2\pi} \int_{-\pi}^{+\pi} X(e^{j\omega}) e^{+j\omega n} d\omega, \quad \forall n \in \mathbb{Z},$$

unde  $x$  este o secvență discretă de semnal, absolut sumabilă.

$\mathcal{X}$

Operatorul Z (Transformata Z), definit(ă) prin:

$$X(z) \stackrel{\text{def}}{=} \mathcal{X}(x)(z) \stackrel{\text{def}}{=} \sum_{n \in \mathbb{Z}} x[n] z^{-n}, \quad \forall z \in \mathcal{C}_x,$$

pentru orice secvență discretă de semnal  $x$ . Suma din definiție este convergentă într-o coroană circulară centrată în originea planului complex,  $\mathcal{C}_x$ , care depinde de semnalul  $x$ .

$[x]_{\text{dB}}$

valoarea în decibeli a numărului  $x > 0$ , definită prin:

$$[x]_{\text{dB}} \stackrel{\text{def}}{=} 10 \lg x \quad (\text{în Teoria Sistemelor});$$

$$[x]_{\text{dB}} \stackrel{\text{def}}{=} 20 \lg x \quad (\text{în Prelucrarea Semnalelor}).$$

Deoarece  $2^{10} = 1024 \cong 10^3$ , în practică se consideră că  $[2]_{\text{dB}} = 3 \text{ dB}$  (în Teoria Sistemelor) sau  $[2]_{\text{dB}} = 6 \text{ dB}$  (în Prelucrarea Semnalelor).

$\lfloor x \rfloor$

partea întreagă a numărului real  $x \in \mathbb{R}$ , adică întregul cel mai mare inferior lui  $x$ .



## Abrevieri

[Ref]	Se citează referința cu eticheta [Ref] din lista bibliografică.
AR	<i>Auto-Regressive model</i> (model auto-regresiv)
ARMA	<i>Auto-Regressive Moving Average model</i> (model auto-regresiv, de medie alunecătoare)
ARMAX	<i>Auto-Regressive Moving Average with eXogenous control model</i> (model auto-regresiv, de medie alunecătoare, cu control extern)
ARX	<i>Auto-Regressive with eXogenous control model</i> (model auto-regresiv cu control extern)
cmmmc	cel mai mic multiplu comun
BJ	model Box-Jenkins
dB	decibel, decibeli
FIR	<i>Finite Impulse Response</i> (sistem cu răspuns finit la impuls)
FPE	<i>Final Prediction Error</i> (eroare finală de predicție)
GUI	<i>Graphical User Interface</i> (interfață grafică convivială cu utilizatorul)
IA	Inteligență Artificială
IIR	<i>Infinite Impulse Response</i> (sistem cu răspuns infinit la impuls)
IS	Identificarea Sistemelor
LTI	<i>Linear Time Invariant Systems</i> (sisteme liniare invariante la deplasări temporale)
MA	<i>Moving Average model</i> (model de medie alunecătoare)
MCMMP	Metoda Celor Mai Mici Pătrate
MCMMP-R	Metoda Celor Mai Mici Pătrate în variantă recursivă
MCMMP-E	Metoda Celor Mai Mici Pătrate Extinsă
MGN	Metoda Gauss-Newton
MIMO	<i>Multiple Input multiple Output</i> model (model cu intrări și ieșiri multiple)
MIS	Modelarea și Identificarea Sistemelor
MMEP	Metoda Minimizării Erorii de Predicție
MMEP-R	Metoda Minimizării Erorii de Predicție în variantă recursivă
MRPL-R	Metoda de Regresie Pseudo-Liniară în variantă recursivă
MVI	Metoda Variabilelor Instrumentale
MVI-R	Metoda Variabilelor Instrumentale în variantă recursivă
OE	<i>Output Error model</i> (model de tip “eroare de ieșire”)
PE	Programare Evoluționistă/Evolutivă
PS	Prelucrarea Semnalelor
SISO	<i>Single Input Single Output</i> model (model cu o intrare și o ieșire)
SNR	<i>Signal-to-Noise Ratio</i> (raportul semnal-zgomot)
SPA	Semnal Pseudo-Aleator
SPAB	Semnal Pseudo-Aleator Binar
TF	Transformata Fourier
TFD	Transformata Fourier Discretă
TLC	Teorema Limită Centrală
TS	Teoria Sistemelor
TZ	Transformata Z

# Aspecte practice în Modelarea și Identificarea Sistemelor

**Dan Ștefănoiu, Ion Matei, Petre Stoica\***

Universitatea “Politehnica” din București  
Facultatea de Automatică și Calculatoare  
Grupul de Identificare a Sistemelor și Prelucrare de Semnal  
Splaiul Independenței nr. 313, Sector 6  
77206 – București, ROMÂNIA  
Tel. (+ 40 21) 402 9318; Fax. (+ 40 21) 411 9163

E-mails: [danny@router.indinf.pub.ro](mailto:danny@router.indinf.pub.ro),  
[ion\\_matei\\_ro@yahoo.com](mailto:ion_matei_ro@yahoo.com),  
[ps@SysCon.uu.se](mailto:ps@SysCon.uu.se)

## Prefață

*Cartea de față prezintă o serie de aspecte practice uzuale destinate în special completării cursurilor introductive de Modelarea Sistemelor, Identificare (Experimentală) a Sistemelor, Prelucrare (Numerică) a Semnalelor și/sau de Comandă (Numerică) a Sistemelor, de la facultățile tehnice de profil electric. Fiind cursuri tipice de matematică aplicată în Automatică și/sau Electronică, ele beneficiază de o argumentație riguroasă, dar cu multe aspecte teoretice a căror înțelegere poate fi ușurată printr-o colecție de exemple sau exerciții de gândire și probleme de simulare pe un mijloc automat de calcul. Printre altele, cartea se dorește a fi o alternativă a îndrumărilor de laborator [StD9603] (destinat modelării și predicției seriilor de timp), [StD9605] (o culegere de probleme rezolvate din domeniul Identificării Sistemelor) și [StD9602b] (destinat implementării algoritmilor de tip FFT – Fast Fourier Transform – din cadrul Prelucrării Numerice a Semnalelor).*

---

\* Universitatea din Uppsala, Departamentul de Sisteme și Control Automat, P.O. Box 27, 75103 – Uppsala, SUECIA.

*Cartea debutează cu o succintă privire de ansamblu asupra domeniilor Modelării și Identificării Sistemelor. Interacțiunea cu domeniul Prelucrării Semnalelor este de asemenea amintită. Capitolele sunt apoi descrise astfel încât cititorul să fie mai întâi familiarizat cu un suport teoretic minimal necesar înțelegerii aplicațiilor abordate. O serie de exerciții pregătitoare graduale au rolul de a obișnui cititorul cu terminologia și notațiile specifice contextului de lucru în care se desfășoară aplicațiile. Exercițiile de gândire sau problemele de simulare pe calculator sunt formulate în finalul fiecărui capitol. O colecție de programe scrise în MATLAB® (versiunea 6.\*) sunt înregistrate pe Discul Compact atașat cărții. Unele dintre aceste programe au fost concepute la Universitatea din Uppsala (Departamentul de Control Automat) și Institutul Tehnologic Lund (ambele din Suedia), fiind disponibile gratuit și pe internet la adresa <http://www.syscon.uu.se/Courses/>. Ele au fost comentate în limba engleză și ușor modificate, pentru mai ușoara lor înțelegere de către cititori. O succintă explicație în limba română este de asemenea furnizată. Alte programe (mai multe) au fost în întregime proiectate de către grupul de cercetare de Identificare a Sistemelor și Prelucrare de Semnal din cadrul Facultății de Automatică și Calculatoare a Universității "Politehnica" din București. Cu toate acestea, cititorii sunt invitați să conceapă și propriile lor programe în cursul abordării aplicațiilor descrise.*

*Sperăm ca prin această carte să venim în întâmpinarea tuturor celor care doresc să studieze Modelarea și Identificarea Sistemelor dintr-o perspectivă practică.*

**Autorii.**  
**București, Mai 2004**

---

® MATLAB și SIMULINK sunt mărci înregistrate ale firmei MathWorks Inc. din SUA.  
(<http://www.mathworks.com/>)

## Introducere

Identificarea Sistemelor (IS) este o disciplină al cărei obiect de studiu îl constituie modelarea proceselor/sistemelor dinamice folosind date experimentale achiziționate în cursul exploatării acestora. Modelele matematice cu care se operează în cadrul IS sunt în principal bazate pe conceptele de *ecuație diferențială* (pentru sistemele cu evoluție în timp continuu) și *ecuație cu diferențe* (pentru sistemele cu evoluție în timp discret). Cu toate acestea, modele ce apelează la alte concepte sunt de asemenea utilizate, dar mai mult în scopul unor descrieri calitative ale comportamentului procesului ce trebuie identificat.

Domeniul IS a fost conturat în special odată cu publicațiile lui K.J. Åström și P. Eykhoff din anii '70-'80 [AsEy71], [EyP74], [EyP81]. În paralel, pot fi menționate contribuții importante la dezvoltarea domeniului și conturarea unor direcții de cercetare prin publicațiile lui R.L. Kashyap și A.R. Rao [KaRa76], R.K. Mehra și D.G. Lainiois [MeLa76], G.C. Goodwin și R.L. Payne [GoPa77] sau T. Söderström [SoT84]. Aplicațiile tehnicilor de identificare și estimare parametrică (care presupun și modelare matematică) nu au întârziat să apară. Ele sunt descrise într-o serie de simpozioane IFAC dedicate IS și tehnicilor de estimare parametrică, cum ar fi cele de la: Praga (1967, 1970), Haga (1973), Tbilisi (1976), Darmstadt (1979), Washington DC (1982). Numeroase lucrări de sinteză și priviri de ansamblu au fost publicate în special în revistele *Automatica* editate de comitetul IFAC [IFAC80], [IFAC82]. Dar una dintre cele mai complete caracterizări ale domeniului a fost publicată în [SoSt89] – probabil cea mai citată referință din ultimul deceniu. Au urmat [LjGl94] și [LjL99] – două referințe orientate către algoritmi de identificare.

În România, perioada cea mai prolifică în materie de publicații din domeniul IS (anii '70-'80) nu a rămas fără ecou. Astfel, se poate spune că școala românească de Identificări a fost inițiată în special prin lucrările lui C. Penescu, M. Tertîșco și P. Stoica [PITC71], [TeSt80], [TeSt85], [TSP87]. O viziune extrem de practică legată de IS (în contextul controlului automat al sistemelor) a fost publicată de către I.D. Landau în [LaID93] (în limba franceză), carte care a fost tradusă și în limba română [LaID97].

Indiscutabil, acest scurt istoric nu poate cuprinde panopia vastă a contribuțiilor care au condus la diversificarea și îmbogățirea domeniului IS. Astăzi, IS își continuă dezvoltarea în special prin deschiderea față de aplicațiile necesitând abordări interdisciplinare. Astfel, algoritmi rapizi și tehnici neconvenționale de identificare au început să apară încă de la începutul anilor '90, prin interacțiunea cu alte domenii de cercetare, în special cu Prelucrarea Semnalelor (PS), Inteligența Artificială (IA) și Programarea Evoluționistă (PE).

Importanța studierii domeniului IS rezidă în însuși conceptul de *modelare matematică*. Numeroase aplicații de Automatică și/sau de Știința Calculatoarelor apelează la modele matematice. În multe cazuri, procesele studiate sunt atât de complexe încât nu este posibilă o caracterizare a lor prin descrierea fenomenelor fizice de la baza comportamentului lor, adică folosind principiile și legile fizicii exprimate prin ecuații de bilanț. De multe ori, chiar ecuațiile obținute în acest fel conțin un număr de parametri necunoscuți. În asemenea situații, utilizatorul este obligat de împrejurări să apeleze la modele și tehnici de identificare.

Cadrul de lucru specific din IS este structurat în jurul a 3 concepte fundamentale: *modelul matematic*, *semnalul de stimul* și *metoda de identificare*.

## Aspecte practice în Modelarea și Identificarea Sistemelor

În termeni generali, identificarea unui proces/sistem dinamic necesită parcurgerea următoarelor etape:

- stimularea procesului cu un anumit semnal (dacă este posibil);
- achiziționarea pe un orizont finit de timp a datelor de intrare-ieșire astfel obținute și prelucrarea primară a lor (atenuare grosieră de zgomot);
- alegerea unui model matematic adecvat (care să concorde cu datele achiziționate), dintr-o clasă specifică de modele;
- determinarea modelului selectat folosind o metodă de identificare corespunzătoare;
- validarea modelului matematic obținut prin intermediul unei metode de validare.

Succesul unui *experiment de identificare* constând în etapele de mai sus depinde în mare măsură de maniera în care au fost precizate cele 3 concepte fundamentale amintite.

**Modelele matematice** pot fi ne-parametrice sau parametrice. Modelele ne-parametrice sunt utilizate în special pentru a obține descrieri apriorice, mai mult de ordin calitativ, ale procesului ce trebuie identificat. În acest caz, datele achiziționate sunt privite ca date statistice referitoare la evoluția procesului. Metode statistice relativ simple (în general bazate pe tehnica (auto-)corelației) sunt aplicate datelor pentru a obține modele atât în domeniul timpului cât și al frecvenței. Aceste modele sunt descrise prin reprezentări grafice sau tabele, dar fără a apela la conceptul de parametru. Ele folosesc la analizarea proceselor din diferite perspective. În principiu, 4 tipuri de analize pot fi efectuate: analiza în frecvență, analiza regimului tranzitoriu, analiza de auto-corelație și analiza spectrală. Primele 2 capitole au ca obiectiv principal ilustrarea modului în care modelele ne-parametrice pot caracteriza evoluția unui proces și pot fi identificate.

Modelele parametrice cele mai utilizate în aplicații fac parte din clasa ARMAX (*Auto-Regressive Moving Average with exogenous control*). Reamintim că ecuația generală a clasei ARMAX[na,nb,nc] (o ecuație cu diferențe) este următoarea:

$$\underbrace{A(q^{-1})y[n]}_{AR} = \underbrace{B(q^{-1})u[n]}_X + \underbrace{C(q^{-1})e[n]}_{MA}, \quad \forall n \in \mathbb{N}, \quad (1)$$

unde, prin convenție, parantezele drepte indică timpul discret sau normalizat (pentru timp continuu, se utilizează parantezele rotunde). Tot în ecuația (1), au fost utilizate următoarele notații (consacrate):

- $u$  este semnalul de intrare sau de stimul.
- $y$  este semnalul de ieșire sau răspunsul sistemului.
- $e$  este semnalul stocastic ideal numit *zgomot alb*. Din punct de vedere statistic, zgomotul alb este prototipul semnalelor total neautocorelate, adică:

$$E\{e[n]e[m]\} = \lambda^2 \delta_0[n-m], \quad \forall n, m \in \mathbb{Z}, \quad (2)$$

unde  $E$  reprezintă operatorul de mediere statistică,  $\delta_0$  este impulsul unitar centrat în origine (*simbolul lui Kronecker*), iar  $\lambda^2$  este varianța zgomotului, necunoscută.

- $q^{-1}$  este operatorul de întârziere cu un pas (de eşantionare), definit prin:  
 $(q^{-1}f)[n] = f[n-1], \forall n \in \mathbb{Z}$  pentru orice şir de date  $f$  (scalar sau vectorial).
- $A, B, C$  sunt polinoame de grade finite:

$$\begin{cases} A(q^{-1}) = 1 + a_1 q^{-1} + \dots + a_{na} q^{-na} \\ B(q^{-1}) = b_1 q^{-1} + \dots + b_{nb} q^{-nb} \\ C(q^{-1}) = 1 + c_1 q^{-1} + \dots + c_{nc} q^{-nc} \end{cases}, \quad (3)$$

unde atît coeficienţii  $\{a_i\}_{i \in \overline{1, na}}, \{b_i\}_{i \in \overline{1, nb}}, \{c_i\}_{i \in \overline{1, nc}}$  (adică *parametrii* modelului), cît şi gradele lor  $na, nb, nc$  (adică *indicii structurali* ai modelului) sunt necunoscuţi şi trebuie determinaţi.

Modelul general al clasei ARMAX arată de fapt că semnalul de ieşire se obţine ca rezultat al superpoziţiei dintre un semnal *util* obţinut prin filtrarea semnalului de intrare şi un semnal *parazit* obţinut prin filtrarea zgomotului alb, aşa cum este ilustrat în **Figura 1**. Particularitatea principală a clasei de modele o constituie faptul că ambele filtre (notate prin  $H$  şi  $G$  în figură) au aceiaşi poli (daţi de rădăcinile polinomului  $A$ ). Cu alte cuvinte, filtrele sunt simultan stabile sau instabile. În acest context de lucru al IS, se urmăreşte determinarea modelelor stabile (zerourile lui  $A$  trebuie să fie amplasate în interiorul discului unitar din planul complex).

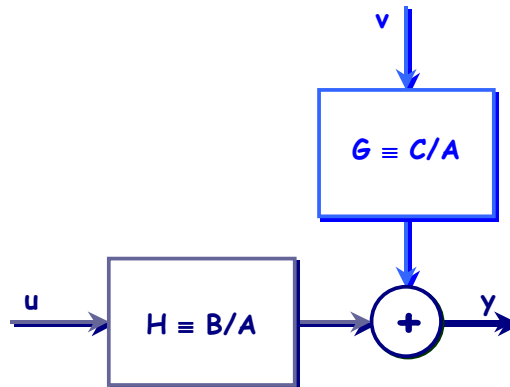


Figura 1. *Reprezentarea sistemică a modelelor ARMAX.*

Cazurile particulare cele mai utilizate în aplicaţii sunt modelele: ARX[na,nb] (pentru  $C \equiv 1$ ), AR[na] (pentru  $B \equiv 0$  şi  $C \equiv 1$ ), MA[nc] (pentru  $A \equiv 1$  şi  $B \equiv 0$ ) şi ARMA[na,nc] (pentru  $B \equiv 0$ ). Primul model este tipic aplicaţiilor de control numeric optimal, în timp ce ultimele 3 sunt utilizate în special pentru modelarea şi predicţia *seriilor de timp* (mai precis, a componentei lor stocastice). O *serie de timp* (sau un *proces stocastic în timp discret*) este văzută ca o realizare a unui proces stimulat de zgomotul alb.

În acest context, problema principală a IS este determinarea parametrilor, a indicilor structurali şi a varianţei zgomotului alb, folosind date achiziţionate pe un orizont finit de măsură:  $\{y[n]\}_{n \in \overline{1, N}}$  şi, dacă este posibil,  $\{u[n]\}_{n \in \overline{1, N}}$ .

## Aspecte practice în Modelarea și Identificarea Sistemelor

În vederea rezolvării acestei probleme, ecuația (1) a modelelor ARMAX este exprimată în mod echivalent în *forma de regresie liniară*:

$$y[n] = \varphi^T[n]\theta + e[n], \quad \forall n \in \mathbb{N}, \quad (4)$$

unde  $\theta \in \mathbb{R}^{n\theta}$  este vectorul parametrilor necunoscuți, iar  $\varphi \in \mathbb{R}^{n\theta}$  este vectorul regresorilor (format din date măsurate și, eventual, estimate). Prin convenție, vectorii sunt de tip coloană, ca și în cazul altor discipline (Teoria Sistemelor (TS) sau PS).

Dimensiunea și configurația celor 2 vectori din ecuația (4) depind de modelul selectat. În general, înălțimea vectorilor este  $n\theta = na + nb + nc$ , iar configurația include 3 componente (cîte una pentru fiecare polinom):

$$\begin{bmatrix} \varphi^T[n] \stackrel{def}{=} [-y[n-1] - y[n-2] \cdots - y[n-na] \mid u[n-1] u[n-2] \cdots u[n-nb] \mid \cdots \\ \cdots e[n-1] e[n-2] \cdots e[n-nc]] \\ \theta^T \stackrel{def}{=} [a_1 \ a_2 \cdots a_{na} \mid b_1 \ b_2 \cdots b_{nb} \mid c_1 \ c_2 \cdots c_{nc}] \end{bmatrix} \quad \forall n \in \mathbb{N} \quad (5)$$

În mod evident, deoarece zgomotul  $e$  nu poate fi măsurat separat, ultima componentă din  $\varphi$  poate fi cel mult estimată printr-un procedeu recursiv, care afectează, în general, precizia modelului.

Două dintre modelele de interes (ARX și AR), conduc totuși la eliminarea componentei datorate zgomotului alb în ecuațiile (5), care devin:

$$\text{ARX: } \begin{bmatrix} \varphi^T[n] \stackrel{def}{=} [-y[n-1] - y[n-2] \cdots - y[n-na] \mid u[n-1] u[n-2] \cdots u[n-nb]] \\ \theta^T \stackrel{def}{=} [a_1 \ a_2 \cdots a_{na} \mid b_1 \ b_2 \cdots b_{nb}] \end{bmatrix} \quad \forall n \in \mathbb{N}^* \quad (6)$$

$$\text{AR: } \begin{bmatrix} \varphi^T[n] \stackrel{def}{=} [-y[n-1] - y[n-2] \cdots - y[n-na]] \\ \theta^T \stackrel{def}{=} [a_1 \ a_2 \cdots a_{na}] \end{bmatrix} \quad \forall n \in \mathbb{N}^* \quad (7)$$

În definițiile (6) și (7), se remarcă exprimarea vectorilor regresorilor folosind numai date măsurate (care totuși sunt corupte și de zgomotul alb filtrat).

Comparativ cu modelele matematice obținute prin scrierea ecuațiilor de bilanț rezultate din exprimarea legilor fizicii, modelele de identificare prezintă următoarele caracteristici:

- au o generalitate și validitate limitată la anumite clase de procese, semnale de stimul și chiar numai la anumite puncte de funcționare ale aceluiași proces;
- au o interpretare fizică dificil de dat, deoarece, în majoritatea cazurilor, parametrii nu au semnificații fizice clare; parametrii sunt mai degrabă utilizați ca instrumente menite să ușureze descrierea funcționării procesului;
- determinarea lor este adesea realizabilă prin metode algoritmice, ceea ce le conferă eficiență și simplitate.

Alegerea **semnalelor de stimul** se bazează pe un principiu general: dacă procesul este integrat într-un complex sistemic mai larg – adică funcționează în buclă închisă –, atunci semnalul de stimul este cel utilizat în cursul exploatării; dacă procesul poate funcționa și în buclă deschisă, atunci un model matematic mai precis se obține prin stimularea acestuia cu un semnal *persistent*. Conceptul de *persistență* este crucial în IS. Prin definiție, un semnal  $u$  este persistent de ordin  $M \geq 1$  dacă matricea de auto-covarianță de ordin  $M$ , notată cu  $R_M(u)$ , este strict pozitiv definită (adică inversabilă):

$$R_M(u) \stackrel{\text{def}}{=} \begin{bmatrix} r_u[0] & r_u[1] & r_u[2] & \cdots & r_u[M-1] \\ r_u[1] & r_u[0] & r_u[1] & \cdots & r_u[M-2] \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ r_u[M-2] & \cdots & r_u[1] & r_u[0] & r_u[1] \\ r_u[M-1] & \cdots & r_u[2] & r_u[1] & r_u[0] \end{bmatrix} > 0. \quad (8)$$

Elementul generic al matricii Toeplitz simetrice  $R_M(u)$ , definite în (8), este dat de funcția de auto-covarianță  $r_u$ , la rîndul ei definită prin:  $r_u[k] \stackrel{\text{def}}{=} E\{u[n]u[n-k]\}$ ,  $\forall k \in \mathbb{Z}$ . Datorită ipotezei ergodice, funcția de auto-covarianță poate fi aproximată folosind valorile semnalului măsurate pe un orizont finit de timp,  $\{u[n]\}_{n \in \overline{1, N}}$ . Mai precis:

$$r_u[k] \cong \frac{1}{N-k} \sum_{n=k+1}^N u[n]u[n-k], \quad \forall k \in \overline{1, \lfloor N/4 \rfloor}. \quad (9)$$

Semnificația conceptului de persistență poate fi explicată atât în domeniul timpului, cît și în cel al frecvenței.

În domeniul timpului, cu un semnal persistent de ordin  $M$  se pot determina primele  $M$  valori ale funcției pondere pentru unui sistem dinamic liniar, ca model asociat procesului de identificat. Dacă  $h$  este funcția pondere în cauză și  $\theta \in \mathbb{R}^M$  este vectorul format din primele  $M$  valori ale lui  $h$ , atunci  $\theta$  se obține rezolvînd *ecuația Wiener-Hopf*:

$$R_M(u)\theta = r_M(y, u) \Leftrightarrow \theta = R_M^{-1}(u)r_M(y, u). \quad (10)$$

În ecuațiile (10),  $r_M(y, u)$  este vectorul primelor  $M$  valori ale corelației încrucișate dintre ieșirea și intrarea procesului. Corelația încrucișată,  $r_{y,u}$ , se definește ca și auto-

corelația, adică:  $r_{y,u}[k] \stackrel{\text{def}}{=} E\{y[n]u[n-k]\}$ ,  $\forall k \in \mathbb{Z}$ . Datorită aceleiași ipoteze ergodice, o relație aproximativă similară cu (9) poate fi utilizată și în evaluarea corelației încrucișate ( $u[n]$  trebuie înlocuit cu  $y[n]$ ). Cu cît semnalul de intrare este mai persistent, cu atît modelul sistemului liniar asociat procesului este mai precis, deoarece cu atît mai multe valori ale funcției pondere pot fi estimate.



În domeniul frecvenței, definiția echivalentă a persistenței este următoarea: un semnal  $u$  este persistent de ordin  $M \geq 1$  dacă și numai dacă densitatea sa spectrală de putere, notată tradițional prin  $\phi_u$ , posedă cel puțin  $M$  linii spectrale nenule. Reamintim că *densitatea spectrală de putere* a lui  $u$  se obține aplicând Transformata Fourier (TF) asupra funcției de auto-covarianță  $r_u$ :

$$\phi_u(\omega) = \mathcal{F}(r_u)(e^{j\omega}) \stackrel{\text{def}}{=} \sum_{n \in \mathbb{Z}} r_u[k] \exp(-j\omega k), \quad \forall \omega \in \mathbb{R}. \quad (11)$$

Funcția de auto-covarianță se poate recupera din densitatea spectrală folosind inversabilitatea TF și  $2\pi$ -periodicitatea sa:

$$r_u[k] = \mathcal{F}^{-1}(\phi_u)[k] = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \phi_u(\omega) \exp(+j\omega k) d\omega, \quad \forall k \in \mathbb{Z}. \quad (12)$$

În definiția (11) și formula duală (de inversiune) (12),  $j$  este unitatea imaginară (complexă), iar  $\omega$  se numește pulsație (armonică) normalizată. O linie spectrală de armonică  $\omega$  are amplitudinea  $\phi_u(\omega)$ . Se poate demonstra că  $\phi_u(\omega) \geq 0$ , ceea ce justifică termenul de *densitate spectrală de putere*. Așadar, un semnal  $u$  este persistent de ordin  $M \geq 1$  dacă și numai dacă există cel puțin  $M$  pulsații  $\omega_1, \dots, \omega_{M-1}$  astfel încît  $\phi_u(\omega_i) > 0, \forall i \in \overline{1, M}$ . În acest fel, procesul de identificat este stimulat pe cel puțin  $M$  armonice, pe care este forțat să le amplifice sau să le atenueze, în funcție de comportamentul său intrinsec. Cu cît procesul este stimulat să reacționeze la mai multe armonice, cu atît semnalul de ieșire va codifica mai multă informație despre comportamentul său.

Semnalul de intrare ideal este zgomotul alb, care are persistență infinită. Din păcate, acest semnal nu poate fi generat pe cale artificială. Mai precis, semnalele *artefacte* (adică produse artificiale) nu pot avea ordin infinit de persistență. Există însă semnale artefacte cu ordin finit de persistență care “aproximează” zgomotul alb, în sensul auto-covarianței. Acestea se numesc *Semnale Pseudo-Aleatoare Binare* (SPAB) sau, mai simplu, *Semnale Pseudo-Aleatoare* (SPA). Ele sunt periodice, deoarece algoritmi folosiți pentru generarea lor utilizează precizia finită de reprezentare a valorilor numerice pe un mijloc automat de calcul. Interesant însă, ordinul lor de persistență este proporțional cu perioada. Mai mult, pe măsură ce perioada crește, funcția de auto-covarianță se apropie de cea a zgomotului alb, adică valorile semnalelor pseudo-aleatoare devin tot mai necorelate.

Utilizarea SPAB sau SPA în IS este foarte frecventă ori de cîte ori procesul de identificat poate fi stimulat în buclă deschisă. Modelele obținute folosind aceste semnale au precizie ridicată și sunt foarte versatile, putînd fi utilizate pentru o gamă largă de puncte de funcționare, semnale de stimul și/sau configurații de sistem.

În fine, **metodele de identificare** au drept obiectiv determinarea parametrilor necunoscuți ai unui model, propunînd fie relații directe de calcul, fie proceduri iterative. În orice caz, necunoașterea nu numai a valorilor parametrilor, ci și a numărului lor atrage după sine adoptarea unei strategii iterative în care complexitatea structurală a modelului este crescută treptat, pînă la nivelul la care precizia sa nu mai este

ameliorată semnificativ. Mai precis, se pleacă de la modelul cel mai simplu, adică *parsimonios*<sup>\*</sup>. Pentru fiecare model de structură dată, se determină parametrii săi și se evaluează eroarea față de proces (cu ajutorul unui criteriu predefinit). Dacă eroarea scade în mod semnificativ, se reia procedeul iterativ, adică se crește numărul de parametri, se re-evaluează aceștia și eroarea față de proces. Altfel, procedeul iterativ este stopat și se reține ultimul model determinat. Acest model trebuie să fie validat în final, folosind teste specifice. De exemplu, un model este valid dacă eroarea dintre datele măsurate și cele simulate are caracteristicile unui zgomot alb Gaussian.

Determinarea parametrilor necunoscuți ai unui model matematic se poate realiza în principal folosind metode extrase din Teoria Optimizărilor și/sau din Teoria Estimației (Statistice). O privire rapidă dar obiectivă aruncată asupra acestor metode ar pune în evidență avantajele și dezavantajele lor. Astfel, metodele de optimizare oferă algoritmi iterativi (implementabili) de estimare a parametrilor, dar estimațiile nu pot fi caracterizate din punct de vedere statistic. Ele asigură *convergența* către punctul de optim, dar nu garantează *consistența* estimației din punct de vedere statistic. (În acest context, o estimație a unui parametru este *consistentă* dacă tinde la valoarea adevărată a acelui parametru, pe măsură ce numărul de date achiziționate din proces tinde la infinit, oricare ar fi setul de date utilizat.) Din cealaltă perspectivă, a Teoriei Estimației, consistența parametrilor poate fi testată, însă metodele efective de evaluare suferă în general de ne-implementabilitate, reprezentând mai degrabă un suport teoretic pentru alte metode. În plus, aceste metode se bazează pe ipoteze adesea restrictive, în scopul asigurării consistenței. Cele două teorii se intersectează, din fericire. Metodele de identificare cele mai interesante și utile sunt cele rezultate din combinația optimizării cu estimarea. Ele sunt implementabile (eventual iterative) și permit caracterizarea statistică a parametrilor estimați. Prototipul îl constituie *Metoda Celor Mai Mici Pătrate* ([MCMMP](#)), care va fi succint prezentată în continuare.

Prin multiplicarea la stînga cu vectorul  $\varphi[n]$  a ecuației de regresie liniară (4) și aplicarea operatorului de mediere statistică  $E$ , se obține:

$$E\{\varphi[n]y[n]\} = E\{\varphi[n]\varphi^T[n]\}\theta + E\{\varphi[n]e[n]\}, \quad \forall n \in \mathbb{N}. \quad (13)$$

Ipoteza ergodică permite eliminarea timpului normalizat (discret) în ecuația (13). Mai mult, procesul furnizor de date are o evoluție observabilă, iar matricea pătrată  $E\{\varphi\varphi^T\}$  este inversabilă. Rezultă că valorile adevărate ale parametrilor sunt:

$$\theta^* = \left(E\{\varphi[n]\varphi^T[n]\}\right)^{-1} (E\{\varphi[n]y[n]\} - E\{\varphi[n]e[n]\}), \quad (14)$$

ele nedepinzînd de timpul normalizat  $n \in \mathbb{N}$ . Să presupunem acum că vectorul regresorilor  $\varphi$  nu este corelat cu valoarea curentă a zgomotului alb, adică  $E\{\varphi[n]e[n]\} = 0$ . Foarte probabil, această ipoteză se verifică de exemplu în cazul modelului ARX, deoarece semnalul de intrare este produs artificial de către utilizator sau furnizat de către un alt (sub-)sistem, în timp ce semnalul de ieșire apare întârziat cu un pas (așa cum arată ecuațiile (6)). Astfel, termenul care depinde direct de zgomotul alb în (14) poate fi eliminat, ecuația parametrilor adevărați simplificîndu-se:

<sup>\*</sup> Temenul provine din limba engleză, unde *parsimonious* înseamnă "sărac" sau "zgîrcit". În contextul IS, el înseamnă "sărac/zgîrcit" în informație sau complexitate.

$$\theta^* = \left( E \{ \varphi[n] \varphi^T[n] \} \right)^{-1} E \{ \varphi[n] y[n] \}. \quad (15)$$

Odată ce valorile adevărate ale parametrilor au fost exprimate (ca în (14) sau, simplificat, în (15)), valoarea adevărată a varianței zgomotului alb,  $(\mathcal{X}^*)^2$ , poate fi determinată în 2 pași:

1. Se evaluează media zgomotului alb folosind ecuația de regresie liniară (4):

$$E \{ e[n] \} = E \{ y[n] \} - E \{ \varphi^T[n] \} \theta^*. \quad (16)$$

2. Se aplică definiția varianței, folosind încă o dată ecuația (4):

$$(\mathcal{X}^*)^2 = E \left\{ \left( y[n] - \varphi^T[n] \theta^* - E \{ e[n] \} \right)^2 \right\}. \quad (17)$$

Cu excepția informației structurale, ecuațiile (14) sau (15) (parametri) și (16)-(17) (varianța zgomotului alb) conduc la determinarea exactă a modelului asociat procesului de identificat atunci când s-ar dispune de un număr infinit de realizări sau, cel puțin, de date măsurate (cerut de operatorul de mediere statistică).

Cum achiziționarea unui set infinit de date măsurate nu este posibilă, media statistică trebuie aproximată folosind încă o dată ipoteza ergodică. În general, media statistică a unui semnal discret  $f$  poate fi aproximată cu ajutorul mediei aritmetice evaluate pe un orizont finit (dar suficient de larg) de măsură:

$$E \{ f[n] \} \cong \bar{f} \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^N f[n]. \quad (18)$$

Consecința directă a ecuației (18) o constituie relațiile aproximative de estimare a parametrilor necunoscuți, derivate din (15), (16) și (17):

$$\hat{\theta}_N \stackrel{\text{def}}{=} \underbrace{\left( \frac{1}{N} \sum_{n=1}^N \varphi[n] \varphi^T[n] \right)^{-1}}_{R_N^{-1}} \underbrace{\left( \frac{1}{N} \sum_{n=1}^N \varphi[n] y[n] \right)}_{r_N}. \quad (19)$$

$$\bar{e} \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^N (y[n] - \varphi^T[n] \hat{\theta}_N); \quad \hat{\lambda}_N^2 \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^N (y[n] - \varphi^T[n] \hat{\theta}_N - \bar{e})^2. \quad (20)$$

Se poate demonstra că estimția (19) minimizează funcția cost definită prin însumarea tuturor pătratelor erorilor dintre datele de ieșire măsurate din proces și cele simulate cu ajutorul modelului estimat, centrate pe mediile lor. Mai precis:

$$\hat{\theta}_N = \arg \min_{\theta \in \mathcal{S}} \mathcal{V}_N(\theta), \quad \text{unde: } \mathcal{V}_N(\theta) \stackrel{\text{def}}{=} \sum_{n=1}^N (\tilde{y}[n] - \tilde{\varphi}[n] \theta)^2, \quad \forall \theta \in \mathcal{S}, \quad (21)$$

unde notațiile  $\tilde{y}$  și  $\tilde{\varphi}$  indică centrarea datelor pe medie (adică  $\tilde{y} \equiv y - \bar{y}$  și  $\tilde{\varphi} \equiv \varphi - \bar{\varphi}$ ), în timp ce  $\mathcal{S}$  indică domeniul de stabilitate al modelului matematic. Forma funcției cost  $\mathcal{V}_N$  a condus la conceptul de identificare/estimare folosind MCMMP. De notat că funcția cost constituie de asemenea o măsură a preciziei

modelului de identificare propus și, în consecință, poate fi folosită pentru a determina indicii structurali ai acestuia după strategia iterativă amintită.

Se poate arăta că estimăția oferită de MCMMP este consistentă (adică  $\hat{\theta}_N$  și  $\hat{\lambda}_N^2$  converg la valorile adevărate  $\theta^*$ , respectiv  $(\lambda^*)^2$ , pentru  $N \rightarrow \infty$ ) dacă matricea  $R_N$  este perfect deterministă și inversabilă, iar  $e$  este efectiv un zgomot alb. Aceste condiții relativ restrictive pot fi relaxate astfel încât consistența să se conserve.

MCMMP constituie un fel de “metodă-mamă” din care au luat naștere numeroase alte metode de identificare prin adaptări inspirate de tipul de model utilizat (Metoda Variabilelor Instrumentale pentru modele ARX, Metoda Minimizării Erorii de Predicție pentru modele ARMAX, Metoda Predicției Optimale pentru modele AR, etc.). Cu toate acestea, domeniul IS nu se reduce doar la familia de metode generate de MCMMP. Există, de exemplu, metode de estimare a stărilor prin filtrare Kalman sau metode de identificare robustă în care funcția cost  $\mathcal{V}_N$  este exprimată în mod diferit față de definiția (21). Majoritatea covârșitoare a acestor metode sunt descrise în [SoSt89].

Metodologia IS nu poate fi însă un panaceu universal, ci are limitele sale. Mai mult, utilizatorul trebuie să o utilizeze cu precauție și știință. Cele mai importante probleme practice care apar în identificarea unui proces sunt enumerate mai jos.

- *Selectarea mărimilor ce trebuie măsurate.* Există situații în care mărimi de importanță capitală pentru identificarea unui proces nu pot fi măsurate în mod direct, fiind inaccesibile. De exemplu, dacă se urmărește determinarea unui model al vibrațiilor unui rulment integrat într-un sistem mecanic în vederea detecției defectelor sale, este foarte posibil ca senzorii de vibrație (accelerometrele) să nu poată fi amplasați direct pe carcasa rulmentului. Amplasarea lor în alte locații poate conduce la combinarea și/sau interferența semnalului măsurat cu semnale de vibrație produse de alte componente ale sistemului mecanic, deci la posibile modele matematice inadecvate. Pentru soluționarea acestei probleme dificile, utilizatorul este nevoit să formuleze o problemă alternativă de identificare sau să extragă informația despre procesul studiat din datele măsurate în contextul în care se află amplasat acesta, dacă sunt cunoscute interacțiunile dintre diferitele subsisteme ale sistemului global. Revenind la exemplul rulmentului, o metodă de extragere a vibrației dorite constă în utilizarea unor senzori direcționali, orientarea lor către rulment și folosirea unei metode de atenuare a interferențelor. (O astfel de metodă a fost de exemplu patentată în SUA [CaDL96].) Costul unei astfel de soluții poate fi însă ridicat, astfel că utilizatorul va fi restricționat de mijloacele de care dispune.
- *Achiziția și prelucrarea primară a datelor.* Procesele identificabile se caracterizează prin seturi de date achiziționate pentru care raportul semnal-zgomot (SNR – *Signal-to-Noise Ratio*) are valori rezonabil de mari. Cu alte cuvinte, zgomotul nu trebuie să domine datele utile. Cu cât SNR este mai mic, cu atât modelul asociat procesului riscă să fie mai imprecis și procesul este mai puțin identificabil. Creșterea SNR (adică a dominanței semnalului util în fața zgomotului) se poate realiza într-o oarecare măsură și printr-o prelucrare primară a datelor. Aceasta constă în principal într-o tehnică de atenuare a zgomotelor (*denoising*) bazată pe filtrare. Utilizatorul este confruntat aici cu problema distorsionării datelor prin alegerea inadecvată a filtrului sau a metodei de atenuare de zgomot. Din păcate, între datele utile și cele parazite nu se poate trasa o linie de demarcație clară, astfel că, indiferent de metoda de

prelucrare primară utilizată, o parte din datele utile riscă să fie eliminate, în timp ce o parte din datele parazite riscă să fie interpretate ca date utile. Pentru adîncirea diferenței dintre datele utile și cele parazite, sunt necesare metode de prelucrare sofisticate, care complică în mod nedorit algoritmul de identificare. În consecință, utilizatorul trebuie să proiecteze cu grijă experimentul de achiziție a datelor, astfel încît SNR să aibă valori suficient de mari.

- *Selectarea unui model matematic adecvat.* Aceasta poate fi o problemă dificilă, în special cînd utilizatorul este confruntat cu un proces avînd comportament neliniar pronunțat. Modelele uzuale de identificare sunt liniare. O manieră de a aborda neliniaritățile constă desigur în selectarea de modele neliniare, cu condiția ca neliniaritățile să poată fi caracterizate din punct de vedere matematic. O altă abordare ar fi bazată pe adaptarea și implementarea unei *rețele neuronale*. (La baza Teoriei Rețelelor Neuronale [DuHa96], [Ta97] se află tot MCMMP ca tehnică de optimizare în faza de instruire a rețelei.) În fine, o a treia strategie, mai apropiată de domeniul IS, este utilizarea modelelor liniare, dar cu parametri variabili în timp, care se auto-adaptează sistematic, în funcție de datele achiziționate.
- *Variabilitatea proceselor în timp.* Această caracteristică rezultă pur și simplu din faptul că valorile adevărate ale parametrilor variază în timp. Astfel, se impune folosirea de modele matematice cu parametri variabili în timp (ca în cazul neliniarităților). Problema principală care apare acum este legată de consistența estimațiilor. De această dată, estimațiile parametrilor trebuie nu numai să tindă statistic (adică odată cu mărirea orizontului de măsură) la valorile lor adevărate, ci să le și urmărească evoluția în timp cu precizie suficient de mare. Cele două cerințe sunt în mod evident opuse, astfel încît principalul obiectiv al metodei de identificare utilizate (care nu poate fi decît iterativă) este să asigure un bun compromis între capacitatea de urmărire a estimațiilor și precizia lor. Un alt compromis care trebuie realizat este cel dintre adaptabilitatea modelului matematic și robustețea sa ca sistem dinamic. Este binecunoscut faptul că adaptabilitatea excesivă conduce la pierderea robusteții sistemelor (adică a capacității lor de a rejecta cu anumite performanțe perturbațiile ce conțin șocuri și de a rămîne stabile). La rîndul ei, robustețea excesivă conduce la slabe performanțe de urmărire (adică de adaptabilitate).

Deși succinta prezentare din această introducere a focalizat discuția asupra domeniului IS, ar trebui totuși precizat că unele dintre tehnicile de identificare pot fi întrebuintate și în scopul prelucrării semnalelor. În special în cazul în care procesului studiat nu i se pot pune în evidență semnalele de intrare, informația despre evoluția sa se află codificată în setul de date de ieșire, care este o serie de timp. Modelele seriilor de timp sunt frecvent utilizate în estimarea spectrală [OpSc85], [OpWi85], [PrMa96], predicție [TeSt85], [StD96] sau filtrarea adaptivă [HaS86] – aplicații mai degrabă de PS decît de IS. Însă, între IS și PS nu se poate trage o linie clară de demarcație, la intersecția lor aflîndu-se metode și tehnici extrem de moderne și eficiente care sîervesc scopurilor ambelor domenii.

Aplicațiile descrise în continuare oferă exemple practice sugestive care să ajute înțelegerea noțiunilor teoretice prezentate în diferitele cursuri amintite (în special de IS și PS) și să sugereze cititorului că, în pofida aparențelor date de aparatul matematic utilizat, domeniul IS este unul aplicativ.

## Capitolul 1

### Caracterizări în timp și frecvență ale proceselor stocastice

#### 1.1. Analize de proces prin metode ne-parametrice

Obiectivul acestui capitol este de a prezenta câteva aspecte practice legate de operarea cu modele de identificare ne-parametrice. Așa cum am amintit în Introducere, modelele ne-parametrice oferă caracterizări (de regulă calitative) ale unui proces stocastic atât în domeniul timpului, cât și în cel al frecvenței. În domeniul timpului, se poate efectua o *analiză tranzitorie* și/sau o *analiză statistică bazată pe corelație*. În domeniul frecvenței, se poate realiza direct o *analiză în frecvență* (de tip Fourier) și/sau o *analiză spectrală* (statistică). În cadrul capitolului, se pune accentul pe analizele statistice (de corelație și spectrale). Vom descrie însă pe scurt și celelalte tipuri de analize.

##### A. Analiza tranzitorie

Aceast tip de analiză este specific aplicațiilor de Teoria Sistemelor (TS) și are ca obiectiv evaluarea performanțelor de stabilitate și robustețe ale unui sistem plecând de la răspunsurile sale la o intrare de tip treaptă unitară (*răspunsul indicial*) sau impuls unitar (*răspunsul cauzal la impuls* sau *funcția pondere*) [IoV85], [StF00]. Graficele din zona tranzitorie a acestor răspunsuri oferă însă și posibilitatea de a identifica unii dintre parametrii sistemului liniar asociat (în special pentru sistemele de ordin I sau II). Este vorba despre constantele de timp dominante ale sistemului și, eventual, câștigul său (sau factorul de amplificare).

În cazul în care ieșirea sistemului este perturbată în mod sensibil de zgomot nedeterminist, graficul din zona tranzitorie nu mai poate pune în evidență cu ușurință caracteristicile sistemului, fiind necesară evaluarea curbei sale mediane în acest scop.

Modelul tranzitoriu este unul de precizie scăzută, chiar și în cazul în care SNR are valori ridicate, deoarece determinarea caracteristicilor procesului se efectuează prin metode grafice. Acest model poate fi utilizat totuși ca instrument auxiliar în alegerea unui model parametric adecvat, deoarece el furnizează informații grosiere preliminare despre evoluția procesului.

##### B. Analiza în frecvență

În afara răspunsului indicial sau a răspunsului cauzal la impuls, un sistem dinamic mai poate fi stimulat să răspundă “în frecvență”. Aceasta înseamnă că semnalul de intrare este o armonică elementară de pulsație  $\omega_0$ :

$$u[n] = u_0 \sin(\omega_0 n), \quad \forall n \in \mathbb{N}. \quad (22)$$

Este binecunoscut faptul că răspunsul unui sistem liniar (discret) asimptotic stabil la intrarea armonică (22) are aceeași pulsație  $\omega_0$ , dar amplitudinea și faza pot fi diferite (cu fază negativă, datorită întârzierii intrinseci provocate de sistem):

$$y[n] = y_0 \sin(\omega_0 n + \phi), \quad \forall n \in \mathbb{N}. \quad (23)$$

## Aspecte Practice în Modelarea și Identificarea Sistemelor

În acest context, analiza în frecvență se bazează pe determinarea *răspunsului în frecvență* al sistemului, care, prin definiție, este TF a funcției pondere  $h$ :

$$H(e^{j\omega}) \stackrel{\text{def}}{=} \mathcal{F}(h)(e^{j\omega}) = \sum_{n \in \mathbb{Z}} h[n] e^{-j\omega n}, \quad \forall \omega \in \mathbb{R}. \quad (24)$$

Notația utilizată în definiția (24) nu este întâmplătoare. Dacă  $U$  și  $Y$  sunt Transformatele Z (TZ) ale semnalelor  $u$ , respectiv  $y$ , atunci funcția de transfer a sistemului se obține fie cu ajutorul Teoremei de Convoluție a TZ, fie aplicând TZ asupra funcției pondere a sistemului:

$$y \equiv h * u \Leftrightarrow H(z) = \frac{Y(z)}{U(z)} = \mathcal{X}(h)(z), \quad \forall z \in \mathcal{C}_h = \mathcal{C}_u \cap \mathcal{C}_y. \quad (25)$$

În (25),  $\mathcal{C}_x$  denotă zona de convergență a TZ (o coroană circulară centrată în originea planului complex) determinată de semnalul discret  $x$  (oricare ar fi el). Dacă cercul unitar aparține zonei de convergență  $\mathcal{C}_h$ , atunci răspunsul în frecvență al sistemului corespunde cu TZ a funcției pondere evaluată pe cercul unitar (adică pentru  $z = e^{j\omega}$ ,  $\forall \omega \in \mathbb{R}$ ).

Este evident că parametrii  $y_0$  și  $\phi$  din ecuația (23) pot fi exprimați prin:

$$y_0 = u_0 |H(e^{j\omega_0})| \quad \text{și} \quad \phi = \arg H(e^{j\omega_0}). \quad (26)$$

Atunci se pot măsura amplitudinile  $u_0$  și  $y_0$  împreună cu defazajul  $\phi$  pentru diferite pulsații  $\omega_0$ , astfel încât răspunsul în frecvență să fie trasat grafic folosind egalitățile (26). Măsurarea defazajului nu este întotdeauna o operație simplă, mai ales în situația în care amplitudinile  $u_0$  și  $y_0$  sunt diferite. Din fericire, defazajul se poate determina și pe altă cale, în cazul pulsațiilor  $2\pi$ -raționale  $\omega_0 = 2\pi m_0 / n_0$  (cu  $m_0, n_0 \in \mathbb{N}^*$ ), folosind următorul algoritm:

1. Se alege un orizont de măsură a ieșirii pe o durată finită și întreagă, proporțională cu perioada armonică de intrare:  $N = 2\pi k m_0 / \omega_0 = k n_0$ , unde  $k \in \mathbb{N}^*$  este un factor de proporționalitate arbitrar ales.
2. Se multiplică semnalul de ieșire  $y$  cu  $\sin(\omega_0 n)$ , respectiv  $\cos(\omega_0 n)$ , pentru  $n \in \overline{0, N-1}$ . Se obțin 2 semnale:

$$\begin{cases} y_s[n] \stackrel{\text{def}}{=} y[n] \sin(\omega_0 n) = y_0 \sin(\omega_0 n + \phi) \sin(\omega_0 n) = \frac{y_0}{2} \cos \phi - \frac{y_0}{2} \cos(2\omega_0 n + \phi) \\ y_c[n] \stackrel{\text{def}}{=} y[n] \cos(\omega_0 n) = y_0 \sin(\omega_0 n + \phi) \cos(\omega_0 n) = \frac{y_0}{2} \sin \phi + \frac{y_0}{2} \sin(2\omega_0 n + \phi) \end{cases}. \quad (27)$$

3. Se evaluează media celor 2 semnale din (27) (sau se integrează pe durata  $\overline{0, N-1}$ ):

## 1. Caracterizări în timp și frecvență ale proceselor stocastice

$$\begin{cases} \bar{y}_s = \frac{1}{N} \sum_{n=0}^{N-1} y_s[n] = \frac{y_0}{2} \cos \phi \\ \bar{y}_c = \frac{1}{N} \sum_{n=0}^{N-1} y_c[n] = \frac{y_0}{2} \sin \phi \end{cases} \quad (28)$$

Rezultatul (28) s-a obținut simplu, ținând cont că media unei armonice calculată pe o durată proporțională cu perioada sa este nulă.

4. Se evaluează defazajul direct din (28):

$$\phi = \text{atan2}\left(\frac{\bar{y}_c}{\bar{y}_s}\right) = \text{atan2}\left(\frac{\sum_{n=0}^{N-1} y[n] \cos(\omega_0 n)}{\sum_{n=0}^{N-1} y[n] \sin(\omega_0 n)}\right), \quad (29)$$

unde prin “atan2” am notat funcția arc-tangentă extinsă la cele 4 cadrane ale planului complex (adică ținând cont de semnele număratorului și numitorului).

Această tehnică este similară cu trasarea diagramelor Bode sau Nyquist (adică a hodografului) din TS [StF00].

Din păcate, procedeul anterior (în special algoritmul de mai sus) este extrem de sensibil la perturbații nedeterminate. Deîndată ce măsurătorile sunt afectate de un zgomot, răspunsul în frecvență al procesului poate fi puternic distorsionat.

Atît analiza tranzitorie, cît și analiza în frecvență sunt tehnici de identificare ne-parametrică utile în cazul proceselor cu o bună rejecție a perturbațiilor sau funcționînd în condiții de izolare față de sursele de perturbații. Deîndată ce perturbațiile joacă un rol important în comportamentul unui proces (adică SNR nu poate depăși un anumit prag – de exemplu 4, adică semnal de 4 ori mai puternic decît zgomotul), mai potrivite ar fi următoarele 2 tipuri de analiză.

### C. Analiza bazată pe corelație

Am amintit în Introducere despre ecuația lui Wiener-Hopf (ecuația (10)). Ea reprezintă exemplul tipic de eliminare a zgomotului alb din datele măsurate, prin înlocuirea acestora cu secvențe de covarianță (sau corelație<sup>1</sup>) corespunzătoare. Definiția practică a auto-covarianței este dată de ecuația (9). În mod similar, se poate formula definiția practică a covarianței încrucișate.

În general, analiza bazată pe corelație se desfășoară prin evaluarea secvențelor de auto-covarianță și covarianță încrucișată ale intrării și ieșirii. Astfel, în cazul modelelor ARMAX, o ecuație echivalentă exprimată cu ajutorul acestor secvențe se poate obține prin multiplicarea ecuației (1) cu  $u[n+k]$  pentru  $k \geq 0$  și aplicarea operatorului de mediere statistică:

$$A(q^{-1})r_{wy}[k] = B(q^{-1})r_u[k] + C(q^{-1})r_{ue}[k], \quad \forall k \in \mathbb{N}. \quad (30)$$

<sup>1</sup> Secvența de corelație se obține prin normalizarea secvenței de covarianță în gama [-1,+1]. Se poate arăta că  $|r_{wy}[k]| \leq \sqrt{r_u[0]r_y[0]}$ ,  $\forall k \in \mathbb{Z}$  și  $|r_y[k]| \leq r_y[0]$ ,  $\forall k \in \mathbb{Z}$ , folosind o inegalitate de tip Cauchy-Buniakowski-Schwarz [StD9605].



De altfel, se poate verifica ușor că ecuația Wiener-Hopf este un caz particular al ecuației (30) pentru modelul de sistem cu răspuns finit la impuls (FIR – *Finite Impulse Response*), unde  $A \equiv 1$  și  $C \equiv 1$ , în condițiile în care intrarea nu este corelată cu zgomotul alb.

Ecuația (30) (sau oricare dintre cazurile particulare ale ei) constituie punctul de plecare în analiza bazată pe corelație. De regulă, covarianța încrucișată dintre intrare și zgomotul alb este nulă (intrare necorelată cu zgomotul), dar, în special în buclă închisă, această proprietate s-ar putea să nu se verifice. Menționăm totuși că obiectivul din acest context nu este determinarea parametrilor modelului de lucru, ci evaluarea efectivă a secvențelor de covarianță și reprezentarea lor grafică. Determinarea parametrilor modelului face obiectul metodelor de identificare parametrică.

#### D. Analiza spectrală

Elementul cheie din desfășurarea analizei spectrale îl constituie densitatea spectrală de putere, definită în (11). Este cunoscut faptul că densitatea spectrală a ieșirii unui sistem dinamic liniar având funcția de transfer  $H$  poate fi evaluată printr-o relație asemănătoare ecuației (25) (obținută cu ajutorul Teoremei de convoluție, vezi de exemplu, [OpSc85], [SoSt89], [PrMa96]):

$$\phi_y(\omega) = |H(e^{j\omega})|^2 \phi_u(\omega), \quad \forall \omega \in \mathbb{R}. \quad (31)$$

Practic, (31) arată că spectrul sistemului (amplitudinea răspunsului în frecvență) este factorul care modulează densitatea spectrală a intrării. De asemenea, el poate fi estimat cu ajutorul celor 2 densități spectrale tot din ecuația (31).

Insuficiența ecuației (31) constă în faptul că nu permite și determinarea argumentului/fazei răspunsului în frecvență al sistemului. Pentru aceasta, ar trebui utilizată densitatea spectrală încrucișată dintre intrare și ieșire,  $\phi_{uy}$ , definită similar cu  $\phi_u$  sau  $\phi_y$  prin aplicarea TF asupra covarianței încrucișate  $r_{uy}$ . Astfel, se poate arăta că [StD9605]:

$$\phi_{uy}(\omega) = H(e^{j\omega}) \phi_u(\omega), \quad \forall \omega \in \mathbb{R}, \quad (32)$$

ceea ce conduce la determinarea completă a răspunsului în frecvență al sistemului.

Determinarea răspunsului în frecvență al sistemului din ecuația (32) se numește *estimare spectrală* și necesită estimarea celor 2 densități spectrale  $\phi_u$  și  $\phi_{uy}$ . Folosirea definițiilor în vederea estimării densităților spectrale este o abordare în care rezultatul suferă 3 tipuri de erori: prima datorată estimării secvențelor de covarianță prin formule aproximative, a doua datorată implementării definițiilor densităților spectrale, care apelează la versiunea discretă a TF, numită și *Transformata Fourier Discretă* (TFD) [OpSc85], [PrMa96] și a treia datorată unor efecte numerice marginale cauzate de orizontul finit de măsură a datelor. Dacă primele două surse de eroare nu pot fi atenuate decât prin metode numerice, legat de a treia există o soluție alternativă. Astfel, utilizarea datelor de pe un orizont finit de măsură este echivalentă cu extragerea unei mulțimi finite dintr-un set infinit de date, prin modularea acestuia cu o fereastră dreptunghiulară având deschiderea corespunzătoare. Efectele numerice marginale sunt cauzate de flancurile abrupte ale ferestrei dreptunghiulare. Utilizarea unor ferestre cu flancuri netede poate conduce la atenuarea erorilor marginale, deși

## 1. Caracterizări în timp și frecvență ale proceselor stocastice

orice altă fereastră diferită de cea dreptunghiulară introduce distorsiuni la nivelul datelor măsurate.

În PS de asemenea se vorbește despre “estimarea spectrală”, dar prin metode specifice acestui domeniu (în general bazate pe TFD), fără a apela la conceptul de sistem. De fapt, estimarea spectrală este una dintre cele mai vechi probleme de PS, numai că în contextul acestui domeniu, se operează cu datele măsurate în mod direct și nu cu secvențe de covarianță. Utilizarea covarianței și a densității spectrale este specifică domeniului IS, deoarece, în acest context, semnalele cu care se operează sunt în mod aprioric considerate ne-deterministe/stocastice.

Revenind la ecuațiile de transformare (31) și (32), este util să fie reaminit că demonstrarea lor se bazează pe relații de transformare similare convoluției, dar în care intervin secvențe de covarianță (vezi [SoSt89] și [StD9605]):

$$r_y[k] = \sum_{p \in \mathbb{Z}} \sum_{q \in \mathbb{Z}} h[p] h[q] r_u[k + p - q], \quad \forall k \in \mathbb{Z}. \quad (33)$$

$$r_{uy}[k] = \sum_{m \in \mathbb{Z}} h[m] r_u[k - m], \quad \forall k \in \mathbb{Z}. \quad (34)$$

În ecuațiile (33) și (34),  $h$  este secvența pondere (răspunsul cauzal la impuls) al sistemului liniar.

### 1.2. Aspecte practice în analiza proceselor stocastice

#### A. Procese total neautocorelate – zgomotul alb

Cea mai importantă caracteristică a unei perturbații stocastice constă în faptul că valorile ei nu pot fi cunoscute sau măsurate în mod direct. Este însă posibilă estimarea lor folosind modele matematice adecvate. Un model deterministic precum  $v(t) = \sin(\omega t)$  este arareori potrivit pentru a caracteriza sau estima valorile unei perturbații stocastice. Este mai naturală folosirea modelelor statistice pentru descrierea acestui tip de perturbații.

Un exemplu simplu de proces stocastic îl reprezintă aruncarea unei monede. Ieșirile generate de acest proces pot fi asociate mulțimii  $\{-1, +1\}$  (de exemplu,  $-1$  pentru cap și  $+1$  pentru pajură). De fiecare dată când se efectuează un experiment de aruncare a monedei, se obține un set de date de ieșire diferit. Secvența de ieșire provenită de la un astfel de experiment se numește *realizare* a procesului aleator.

Deoarece un proces aleator reprezintă o întreagă familie de realizări, descrierea acestuia prin intermediul modelelor deterministe nu este realistă, aceste modele fiind capabile să descrie doar comportamentul unei anumite realizări și nu a ansamblului de realizări. De aceea s-a apelat, într-o primă fază, la modele și tehnici de Statistică, ele având avantajul de a extrage și transfera în domeniul determinist informația cu caracter nedeterminist. Așa cum s-a arătat în secțiunea precedentă, două tipuri de analize ne-parametrice pot fi utilizate cu succes în descrierea proceselor stocastice: analiza bazată pe corelație și analiza spectrală. De notat că atât funcția de covarianță cât și funcția de densitate spectrală sunt entități deterministe evaluate folosind o realizare a unui proces nedeterminist.

În cazul procesului de aruncare a monedei, caracteristica fundamentală constă în faptul că fiecare realizare este independentă de celelalte, adică nu există nici o corelație între evenimente (adică aruncări ale monedei). În termeni matematici, acest proces se poate descrie ca o secvență de variabile aleatoare independente  $\{y[n]\}_{n \in \mathbb{N}}$ , identic distribuite, de medie nulă  $E\{y[n]\} = 0$  și varianță unitară  $r_y[0] = \sigma_y^2 = 1$ . Secvența de auto-covarianță este deci:

$$r_y[k] \stackrel{\text{def}}{=} E\{y[n]y[n-k]\} = \delta_0[k] = \begin{cases} 1, & \text{pentru } k = 0 \\ 0, & \text{pentru } k \neq 0 \end{cases}, \quad \forall k \in \mathbb{Z}. \quad (35)$$

Secvența de auto-corelație corespunzătoare aruncării monedei (35) arată că acest proces este *total neautocorelat* (adică obținerea unei valori “cap” sau “pajură” în cursul aruncării curente nu depinde de valoarea obținută la aruncarea precedentă).

Aplicînd TF asupra secvenței de auto-covarianță (35) se obține o densitate spectrală  $\phi_y$  constantă și unitară (vezi definiția (11)), ceea ce arată că procesul conține toate componentele de frecvențe. Datorită acestui fapt, procesul se mai numește și *zgomot alb*, prin analogie cu următoarea experiență de Fizică elementară. Un disc este împărțit în 7 sectoare egale, fiecare fiind colorat cu una din culorile fundamentale ale spectrului luminos: R-roșu, O-oranj, G-galben, V-verde, A-albastru, I-indigo, V-violet, ca în Figura 2.

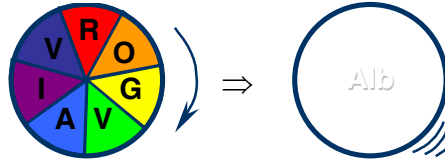


Figura 2. Experimentul obținerii culorii albe din spectrul ROGVAIV.

Culorile sunt ordonate în ordinea descrescătoare a lungimii de undă caracteristice din spectrul vizibil. Rotirea discului cu o anumită viteză conduce totuși la o singură culoare: albă. Efectul se datorează recombinației culorilor fundamentale care sunt în mod egal cantitativ prezente pe disc. În mod analog, unui proces cu densitate spectrală constantă (adică în care frecvențele – “culorile” – sunt în mod egal reprezentate) i s-a asociat sintagma de “zgomot alb”.

Dacă unaia dintre sectoarele discului i se modifică aria, atunci culoarea discului rotit nu mai rămîne albă. În Figura 3, sectorului de culoare roșie I s-a mărit suprafața.

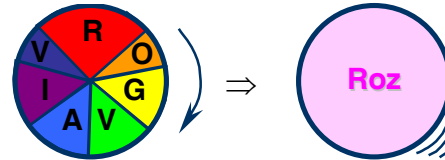


Figura 3. Experimentul obținerii unei nuanțe de roz din spectrul ROGVAIV.

Rezultatul este o nuanță de roz pentru discul în rotație, paleta nuanței depinzînd de proporțiile în care sunt reprezentate culorile fundamentale pe disc. Prin analogie, unui proces stocastic al cărei densitate spectrală posedă cel puțin o armonică dominantă i

## 1. Caracterizări în timp și frecvență ale proceselor stocastice

se asociază conceptul de *zgomot colorat*. Zgomotele colorate se obțin în general prin filtrarea zgomotului alb.

Revenind la densitatea spectrală de putere, aria de sub curba acesteia calculată peste o anumită bandă de pulsații/frecvențe reprezintă energia procesului în acea bandă. În particular, varianța procesului este proporțională cu energia globală a acestuia (vezi relația de inversiune (12)):

$$r_y[0] = \sigma_y^2 = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \phi_u(\omega) d\omega. \quad (36)$$

Atunci cînd s-a făcut referire la procesul de aruncare a monedei ca un exemplu de proces generator de zgomot alb (adică al unor secvențe aleatoare total neautocorelate), s-a presupus de asemenea că el generează datele respectînd o anumită distribuție de probabilitate (uniformă, în acest caz). Cunoașterea apriorică a distribuției de probabilitate asociate unui proces stocastic este însă foarte dificilă, dacă nu imposibilă. Din fericire, o mare categorie de procese stocastice sunt *normal distribuite*, adică după o densitate de probabilitate Gaussiană:

$$p(y[n]) \stackrel{\text{def}}{=} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(y[n] - \bar{y})^2}{2\sigma^2}\right], \quad \forall n \in \mathbb{Z}, \quad (37)$$

unde  $\bar{y}$  este media densității de probabilitate și  $\sigma^2$  este dispersia sa (care măsoară deschiderea *clopotului lui Gauss*). Pentru procesul avînd distribuția de probabilitate (37) se mai scrie:  $y \in \mathcal{N}(\bar{y}, \sigma^2)$  (adică  $y$  aparține clasei de procese normal distribuite de medie  $\bar{y}$  și dispersie  $\sigma^2$ ). Procesul reprezentat de aruncarea monedei nu poate fi considerat normal (ci uniform) distribuit, decît în cazul în care moneda utilizată sau mediul înconjurător prezintă imperfecțiuni ce favorizează apariția mai frecventă a unei fețe. În acest caz, distribuția este Gaussiană, de medie  $+1$  sau  $-1$  (în funcție de fața monedei care iese mai frecvent în urma aruncărilor). În clasa  $\mathcal{N}(\bar{y}, \sigma^2)$  se încadrează adesea procese cu distribuție de probabilitate necunoscută, pe baza *Teoremei Limită Centrală* (TLC) din Statistică<sup>2</sup>.

### Observație

- Două procese stocastice total neautocorelate și normal distribuite sunt și independente. Invers, două procese independente de medie nulă sunt total neautocorelate. Aceste implicații arată ce legătură există între 2 concepte statistice diferite: *neautocorelare* și *independență statistică*. Independența statistică arată doar că probabilitatea apariției simultane a 2 evenimente independente este egală cu produsul probabilităților de apariție separată a lor. Procesele independente pot fi corelate dacă mediile lor sunt nenule.

### B. Zgomote colorate

Identificarea Sistemelor folosind modele neparametrice urmărește să specifice caracteristicile unui proces aleator *cvasi-staționar* (adică avînd densitatea spectrală

<sup>2</sup> Potrivit TLC, un ansamblu cel puțin numărabil de procese statistice cu densități de probabilitate arbitrare constituie un proces aleator normal distribuit.

## Aspecte Practice în Modelarea și Identificarea Sistemelor

aproximativ constantă în timp) la trecerea printr-un sistem liniar sau filtru causal și stabil, a cărui funcție de sistem este rațională:

$$H(q^{-1}) \stackrel{\text{def}}{=} \frac{B(q^{-1})}{A(q^{-1})} = \sum_{n \geq 0} h[n] q^{-n} . \quad (38)$$

În definiția (38),  $A$  și  $B$  sunt polinoame (exprimate ca în definiția (3)), iar  $h$  este secvența pondere a filtrului, ca de obicei. Cauzalitatea și stabilitatea se exprimă simplu astfel [OpSc85]:

$$\bullet \text{Cauzalitate: } h[n] = 0, \dots, \forall n < 0; \quad \bullet \text{Stabilitate: } \sum_{n \geq 0} |h[n]| < \infty . \quad (39)$$

*Cauzalitatea* este o proprietate ce determină în mod unic funcția de auto-covarianță a ieșirii (adică a zgomotului colorat) atunci când filtrul este stimulat la intrare cu un zgomot alb. *Stabilitatea* permite evaluarea răspunsului în frecvență al filtrului (adică asigură existența cercului unitar în zona de convergență a funcției de transfer – TZ a secvenței pondere).

Caracteristicile de interes ale zgomotului colorat sunt media, secvența de auto-covarianță și densitatea spectrală. Am reamintit în paragraful 1.1 relațiile de transformare ale secvențelor de covarianță și densităților spectrale la trecerea unui semnal stocastic printr-un sistem liniar (ecuațiile (31)-(34)). Ele permit evaluarea caracteristicilor semnalului de ieșire dacă se cunosc deja caracteristicile filtrului. Pentru identificarea aceasui, însă, se procedează invers: se observă caracteristicile semnalelor de intrare și ieșire, urmînd ca, pe baza lor, să se determine filtrul ce a condus la aceste caracteristici.

O clasă mare de perturbații poate fi descrisă prin filtrarea unui zgomot alb. Aceasta este echivalentă cu utilizarea unui model de identificare de tip ARMA[na,nb], unde pentru ușurința exprimării, polinomul  $C$  al modelului a fost rennotat cu  $B$ . Următoarele exerciții și probleme de simulare se concentrează pe cîteva exemple de zgomote colorate produse cu ajutorul unor modele ARMA.

### 1.3. Exerciții

#### Exercițiul 1.1

Fie  $e$  un zgomot alb de medie nulă și varianță unitară care stimulează intrarea unui model AR[1]. Să se determine secvența de auto-covarianță a zgomotului colorat rezultat,  $y$ .

#### Exercițiul 1.2

Fie  $e$  un zgomot alb de medie nulă și varianță unitară care stimulează intrarea unui model MA[1]. Să se determine secvența de auto-covarianță a zgomotului colorat rezultat,  $y$ . Dacă modelul MA are ordinul  $nc$ , să se arate că secvența de auto-covarianță a ieșirii are suport finit (adică are un număr finit de valori nenule) și să se determine dimensiunea maximă a suportului.

**Exercițiul 1.3**

Prin filtrarea unui zgomot alb  $e$  de medie nulă și varianță unitară se obține un zgomot colorat  $y$  cu densitatea spectrală:

$$\phi_y(\omega) = \frac{0.75}{1.25 - \cos \omega}, \quad \forall \omega \in \mathbb{R}. \quad (40)$$

Considerînd că filtrul utilizat are funcția de sistem:

$$H(q^{-1}) = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}}, \quad (41)$$

să se determine cei 2 parametri ai acestuia ( $a_1$  și  $b_1$ ). Pot fi ei determinați în mod unic folosind numai analiza spectrală? Evaluați de asemenea varianța  $\sigma_y^2$  a zgomotului colorat.

**Exercițiul 1.4**

Analiza spectrală permite și exprimarea echivalentă a modelelor proceselor stocastice, în scopul simplificării lor. Această tehnică este utilă de exemplu în cazul proceselor afectate de mai multe surse de zgomot. Prin definiție, două modele de procese stocastice sunt *echivalente* dacă densitățile spectrale de putere ale ieșirilor lor sunt identice. Identitatea are loc dacă și numai dacă secvențele lor de auto-covarianță sunt egale.

Fie un proces ARMA[na,nc]:

$$A(q^{-1})x[n] = C(q^{-1})e[n], \quad \forall n \in \mathbb{N}^*, \quad (42)$$

unde ieșirea este  $x$ , iar varianța zgomotului alb  $e$  se notează prin  $\lambda_e^2$ . Să presupunem că ieșirea modelului (42) este la rîndul ei afectată de un zgomot alb aditiv,  $v$ , neocrelat cu  $e$ , avînd varianța  $\lambda_v^2$ . Mai precis, ieșirea observabilă a procesului stocastic este:

$$y[n] = x[n] + v[n], \quad \forall n \in \mathbb{N}^*. \quad (43)$$

Exprimarea modelului cu două surse de zgomot este incomodă. De aceea, se caută echivalarea sa cu un model de filtrare exprimat astfel:

$$y[n] = \frac{B(q^{-1})}{A(q^{-1})}w[n], \quad \forall n \in \mathbb{N}^*, \quad (44)$$

unde  $w$  este un unic zgomot alb de varianță  $\lambda_w^2$ . Această echivalență este ilustrată în **Figura 4**. Să se determine coeficienții și gradul polinomului necunoscut  $B(q^{-1}) = b_0 + b_1 q^{-1} + \dots + b_{nb} q^{-nb}$ , precum și varianța  $\lambda_w^2$  în funcție de polinoamele  $A$ ,  $C$  și varianțele  $\lambda_e^2$ ,  $\lambda_v^2$ , prin echivalarea celor două modele, în cazul  $na = nc = 1$ .

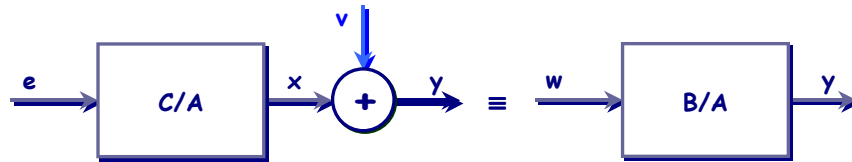


Figura 4. Două modele de procese stocastice echivalente.

Este modelul echivalent (44) unic determinat? Generalizați rezultatul pentru valori arbitrare ale indicilor structurali  $na$  și  $nc$ .

#### Indicație

- Se vor determina și apoi egala secvențele de auto-covarianță ale ieșirilor celor 2 modele, plecând de la ecuațiile nedeterminate ale acestora și exploatând necorelarea zgomotelor.

### 1.4. Probleme de simulare

Se consideră următoarele 2 filtre de zgomot, cu funcții de sistem de ordin 1 și de ordin 2 (respectiv):

$$H_1(q^{-1}) = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}}; \quad H_2(q^{-1}) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}}, \quad (45)$$

Cu ajutorul simulărilor care urmează, se vor analiza influențele polilor și zerourilor asupra funcției de covarianță, densității spectrale și caracteristicilor diferitelor realizări obținute prin filtrarea unui zgomot alb cu filtrele de tipul (45). Simulările se bazează pe următoarele rutine disponibile, scrise în limbajul MATLAB (și înregistrate pe Discul Compact atașat):

#### # ISLAB\_1A

- Apel: `islab_1a(C,A,N,tau_max,nr)` ;
- Modul de calcul al valorilor adevărate și estimate pentru secvențe de auto-covarianță obținute cu ajutorul unui proces ARMA[1,1]. Sunt trasate graficele secvențelor obținute. Este de asemenea trasată o realizare a zgomotului colorat rezultat. Argumentele funcției sunt următoarele:
  - C** polinomul MA (vector [1 c]);
  - A** polinomul AR (vector [1 a]);
  - tau\_max** pivotul maxim al secvențelor de auto-covarianță (implicit: 50);
  - nr** numărul realizărilor de generat (implicit: 1).
- Fereastra grafică tipică: Figura 5.

#### # ISLAB\_1B

- Apel: `islab_1b(x,y,SNR)` ;
- Modul care simulează dependența de SNR a polilor și zerourilor unui model ARMA[2,2], determinat prin echivalarea sa cu un model AR afectat de 2 zgomote necorelate (ca în **Exercițiul 4**). Argumentele funcției sunt următoarele:
  - x** partea reală a polilor modelului AR (implicit: 0.5);
  - y** partea imaginară a polilor modelului AR (implicit: 0.5);
  - SNR** raportul semnal-zgomot (implicit: 3).
- Fereastra grafică tipică: Figura 6.

# 1. Caracterizări în timp și frecvență ale proceselor stocastice

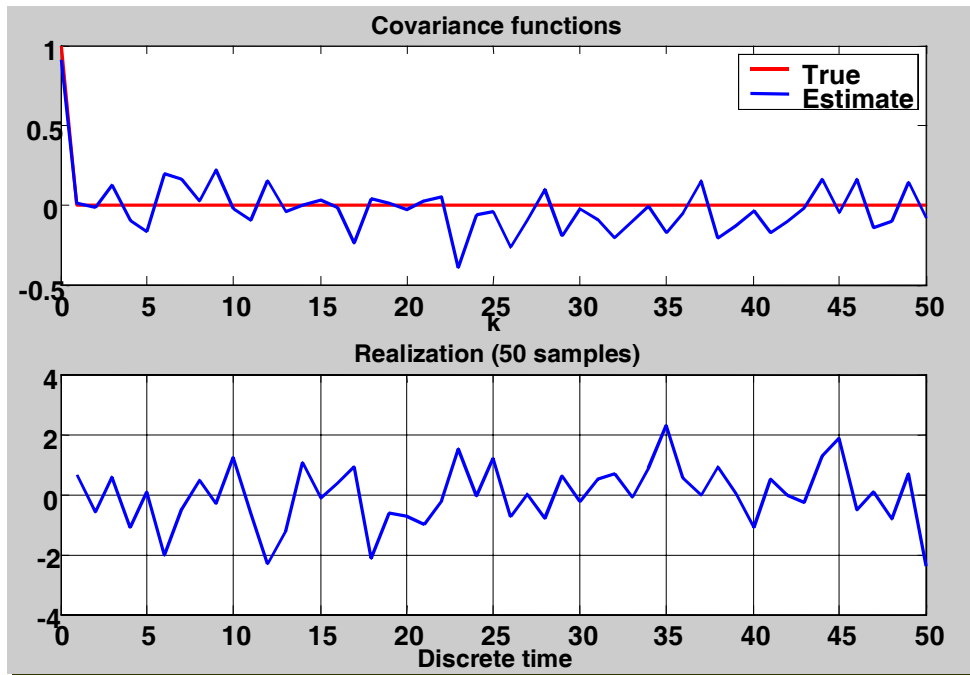


Figura 5. Fereastra grafică tipică a rutinei ISLAB\_1A.

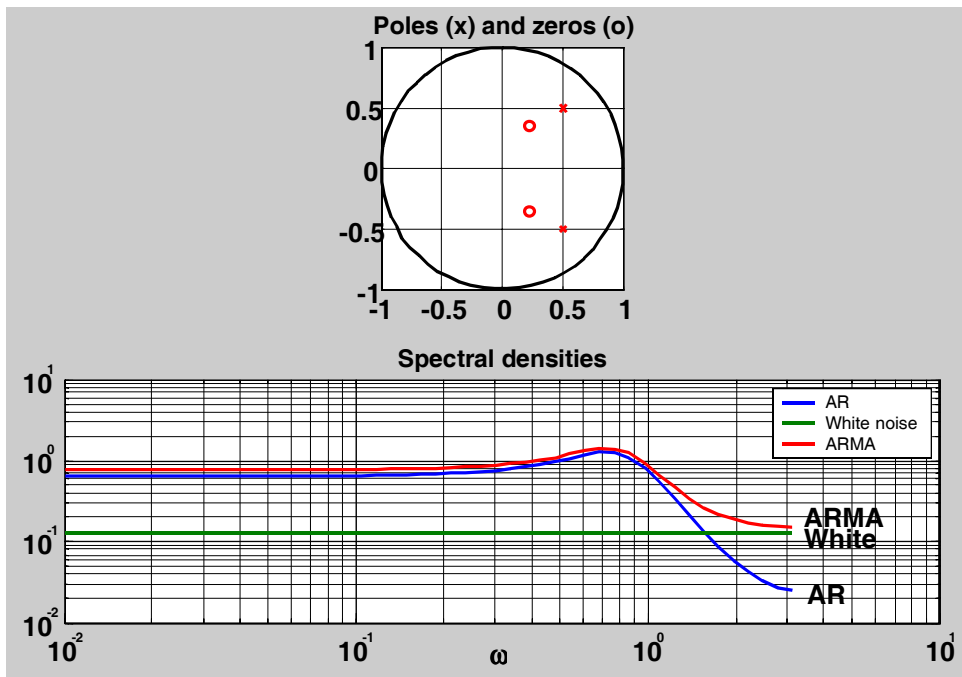


Figura 6. Fereastra grafică tipică a rutinei ISLAB\_1B.



## # NOISE

- Apel: `noise(operation)` ;
- Modul de generare și simulare a zgomotelor colorate produse de modelele stocastice (45). Argumentul funcției (`operation`) este un șir de caractere din mulțimea următoare:

```
close_noise
close_noise_def
init_noise
move_p
move_z
moved_p
moved_z
moving_p
moving_z
noiseclear
show (implicit)
system
winit_noise
```

- Fereastra grafică tipică: **Figura 7** (interfață grafică prietenoasă, care permite varierea în timp real a polilor și zerourilor).

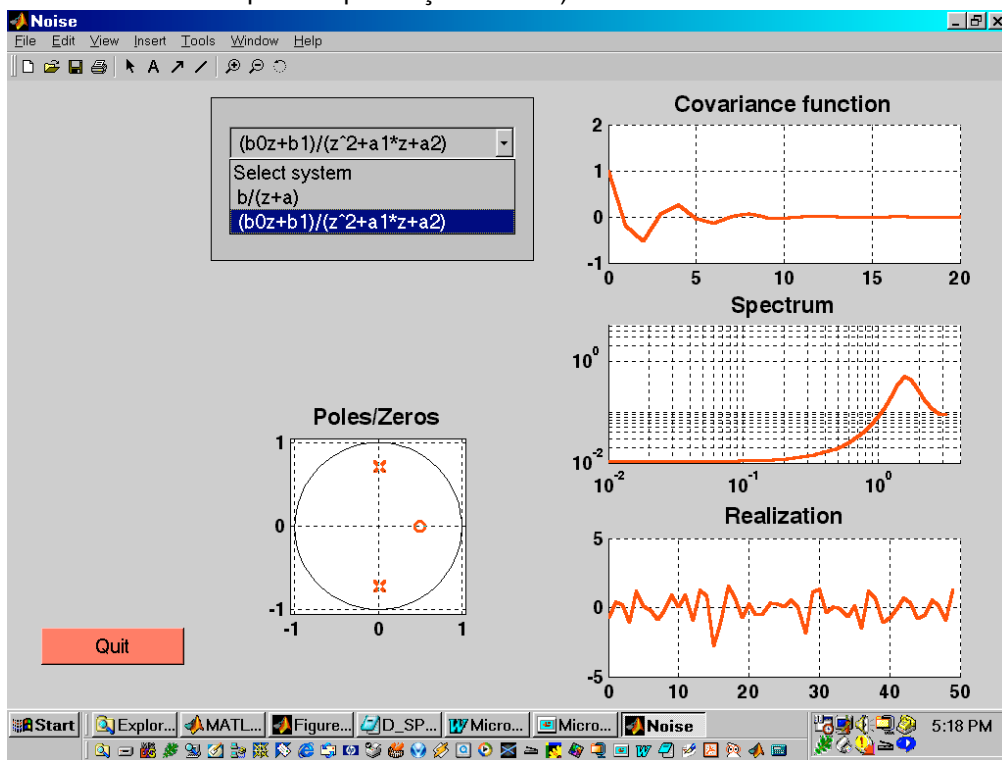


Figura 7. Fereastra grafică tipică a rutinei NOISE.

#### # D\_SPEKTR

- Apel: `[w, fi]=d_spektrum(A,B,sigma2)` ;
  - Rutină auxiliară de evaluare a spectrului ieșirii unui filtru liniar discret stimulat cu un zgomot alb. Argumentele funcției sunt următoarele:
    - A** numitorul funcției de transfer a filtrului (polinom);
    - B** numărătorul funcției de transfer a filtrului (polinom);
    - Sigma2** varianța zgomotului alb de la intrare.
- Funcția returnează:
- w** axa pulsațiilor ( $\omega$ );
  - fi** densitatea spectrală  $\phi_y$  a zgomotului colorat (de ieșire).

#### # SPEFAC

- Apel: `[a, l2]=spefac(r)` ;
- Rutină auxiliară de rezolvare a *Problemei factorizării spectrale*. Aceasta constă în determinarea unui polinom:

$$A(z) \stackrel{def}{=} z^n + a_1 z^{n-1} + \dots + a_n$$

și a varianței  $\lambda^2$  cu proprietatea:

$$\lambda^2 A(z) A(z^{-1}) = \frac{1}{2} \sum_{k=0}^n r[k] (z^k + z^{-k}), \quad (46)$$

pentru o secvență de covarianță  $\{r[0], r[1], \dots, r[n]\}$ . În mod normal, această problemă se poate formula pentru orice secvență de numere  $\{r[0], r[1], \dots, r[n]\}$ , cu condiția să fie *pozitiv definită*, adică verificând inegalitatea:

$$|r[k]| \leq r[0], \quad \forall k \in \overline{0, n}. \quad (47)$$

Problema factorizării spectrale (46) este rezolvată în cazul determinării unui model AR[n] atunci când este stimulat de un zgomot alb și se cunoaște densitatea spectrală de putere a ieșirii (deci și secvența de auto-covarianță a ieșirii, cu ajutorul formulei de inversiune (12)).

Argumentul funcției `spefac` este **r** – secvența de (auto-)covarianță (vector).

Funcția returnează:

- a** coeficienții polinomului AR (vector);
- l2** varianța zgomotului alb  $\lambda^2$  cu care trebuie stimulat modelul AR pentru a obține la ieșire exact secvența de auto-covarianță **r**.

#### **Problema 1.1**

Pentru a rezolva punctele următoare, se va utiliza funcția `NOISE`.

1. Să se testeze grafic dacă filtrul obținut în **Exercițiul 1.3** (de tipul lui  $H_1$  din definiția (45)) este corect.

2. Să se varieze polii filtrului  $H_2$  din definiția (45) și să se comenteze rezultatele obținute cu ajutorul funcției **NOISE**.
3. Unde trebuie amplasați polii filtrului  $H_2$  pentru a obține un filtru trece jos?
4. Unde trebuie amplasați polii filtrului  $H_2$  pentru a obține un vîrf de rezonanță la  $\omega=1$ ? Ce se poate spune despre conținutul în frecvență al semnalului analizînd realizările procesului?
5. Ce efect observați atunci cînd filtrul  $H_2$  are zeroul în vecinătatea cercului unitar?

### Problema 1.2

Să se utilizeze modulul de simulare **ISLAB\_1A** pentru a simula un proces stocastic de model ARMA[1,1]. De exemplu, pentru a genera un proces de tip AR[1] cu un singur pol în 0.9, se folosește sintaxa:

```
islab_1a(1, [1 0.9]) ;
```

În mod implicit, modulul de simulare alege: **N=100**, **tau\_max=50** și **nr=1**.

1. Să se analizeze maniera în care estimațiile funcțiilor de covarianță variază cu **N** (numărul de eșantioane) și **tau\_max** (pivotalul maximal al secvenței de auto-covarianță) pentru diferite locații ale polilor.
2. Să se verifice faptul că estimațiile funcțiilor de covarianță tind către valorile adevărate pentru procese de tip AR[1] și MA[1], pe măsură ce **N** tinde către infinit.
3. Să se verifice corectitudinea rezultatelor obținute la **Exercițiile 1.1** și **1.2**.

### Problema 1.3

Se consideră un proces stocastic asociat unui model AR[2] cu două surse de zgomot (ca în contextul **Exercițiului 1.4**), pe care dorim să îl echivalăm cu un proces descris de un model ARMA[2,2], avînd o singură sursă de zgomot. Pentru simulările care urmează, se va utiliza modulul **ISLAB\_1B**.

1. Să se analizeze maniera în care variază polii și zerourile modelului ARMA atunci cînd variază SNR. În acest context, SNR este definit prin raportul dintre varianța semnalului util  $x$  și varianța zgomotului aditiv  $v$  (cu notațiile din **Exercițiul 1.4**).
2. Să se studieze cazurile în care SNR tinde la infinit (semnalul domină zgomotul) și SNR tinde la zero (zgomotul domină semnalul). Să se comenteze modificările înregistrate de densitățile spectrale.

## Capitolul 2

### Identificarea modelelor ne-parametrice

#### 2.1. Contextul general de lucru

O succintă descriere a modelelor ne-parametrice a fost prezentată în Capitolul 1. La rîndul ei, descrierea face referire la cadrul de lucru conturat în Introducere. Obiectivul acestui capitol este de a ilustra metodologia uzuală folosită în identificarea ne-parametrică. Problemele de simulare propuse pot fi abordate în cadrul mediului de programare MATLAB, cu ajutorul unor funcții dedicate, aparținînd bibliotecii specializate în tehnici de IS (numită *System Identification toolbox*).

Aplicațiile studiate utilizează două modele parametrice pentru a genera datele utilizate în identificarea neparametrică. În acest fel, rezultatele de identificare obținute (adică diagramele rezultate în urma analizelor ne-parametrice) pot fi ușor verificate. Cele 2 modele sunt: ARX[na,nb] și OE[na,nb] (*Output Error model* – model de tip “eroare de ieșire”). Ambele aparțin clasei ARMAX definită prin ecuația (1). Mai precis, ecuațiile celor 2 modele sunt următoarele:

$$\text{ARX[na,nb]: } A(q^{-1})y[n] = B(q^{-1})u[n] + e[n], \quad \forall n \in \mathbb{N}. \quad (48)$$

$$\text{OE[na,nb]: } y[n] = \frac{B(q^{-1})}{A(q^{-1})}u[n] + e[n], \quad \forall n \in \mathbb{N}. \quad (49)$$

Polinoamele  $A$  și  $B$  din ecuațiile (48) și (49) sunt definite în relațiile (3). În ambele modele, perturbația  $e$  este considerată un zgomot alb Gaussian de medie nulă și varianță  $\lambda^2$ , necorelat cu intrarea  $u$ . Se observă că zgomotul afectează ieșirile celor două sisteme în mod diferit. Pentru sistemul descris de modelul ARX, perturbația  $e$  apare ca un zgomot de proces, în timp ce pentru sistemul descris de modelul OE, perturbația  $e$  apare ca un zgomot de măsură (i.e. care distorsionează ieșirea măsurată  $y$ ).

Următoarele polinoame particulare pot fi utilizate în cadrul simulărilor:

$$A(q^{-1}) = 1 - 0.8q^{-1}, \quad B(q^{-1}) = q^{-1}. \quad (50)$$

$$A(q^{-1}) = 1 - 0.1q^{-1} - 0.56q^{-2}, \quad B(q^{-1}) = 0.5q^{-1} + 0.3q^{-2}. \quad (51)$$

De asemenea, zgomotul alb va fi generat cu ajutorul funcției MATLAB **randn**, dispersia fiind fixată la valoarea  $\lambda^2 = 1$ . Pentru a pune în evidență erorile sistematice și diferențele dintre diferitele metode aplicate, vor fi inițiate 100 de experimente, în urma cărora se vor produce 100 de realizări pentru fiecare model. Dacă s-ar lucra cu o singură realizare, unele rezultate ar putea fi dependente într-o măsură prea mare de caracterul aleator al datelor generate.

În aceast capitol, se vor studia trei metode de identificare ne-parametrică, pe bază de: analiză tranzitorie, analiză de corelație și analiză spectrală.

## 2.2. Exerciții

### Exercițiul 2.1

Verificați dacă cele 2 modele generale (48) și (49) pot fi echivalate, în sensul definiției din **Exercițiul 1.4**.

### Exercițiul 2.2

Determinați funcțiile pondere ale celor 2 sisteme liniare modelate de ecuațiile (48) și (49) pentru cazul general ARX[1,1] și OE[1,1]. Particularizare: definițiile (50).

### Exercițiul 2.3

Deduceți relațiile recurente verificate de funcțiile de auto-covarianță ale ieșirii în fiecare din cele 2 modele (48) și (49) pentru cazul particular în care polinoamele sunt definite ca în ecuațiile (50). Evaluați SNR al celor 2 modele în cazul în care sunt stimulate cu treapta unitară și comentați rezultatele obținute.

### Exercițiul 2.4

Deduceți relațiile generale ale densităților spectrale de putere ale ieșirilor celor 2 modele (48) și (49) pentru cazul particular în care polinoamele sunt definite ca în ecuațiile (50). În acest caz particular, ca și în cazul particular (51), deduceți răspunsurile în frecvență ideale ale celor 2 modele (i.e. în absența zgomotului).

## 2.3. Probleme de simulare

### Problema 2.1

În cadrul acestei probleme, se va studia analiza tranzitorie. Modelele ARX (48) și OE (49) vor fi simulate de 100 de ori cu intrarea treaptă:

$$u[n] = \begin{cases} 0 & , n \leq 9 \\ 1 & , n \in \overline{10,100} \end{cases} \quad (52)$$

(timp de cel cel mult 100 de perioade de eșantionare).

- Să se reprezinte grafic, într-o primă fereastră, răspunsul indicial ideal al modelului ARX (48) & (50) (adică în absența zgomotului) plus prima realizare a ieșirii. Într-o a doua fereastră, să se traseze media răspunsurilor obținute (în prezența zgomotului), împreună cu răspunsul indicial ideal și tubul de amplitudine a ieșirii oferit de deviația standard, ca în **Figura 8**. În acest scop, se vor folosi funcțiile MATLAB: **filter**, **mean** și **std**. Observați că deviația standard trebuie calculată luând în considerare ansamblul statistic al realizărilor și nu media acestor realizări. Ce rol credeți ca are tubul de deviație standard astfel ilustrat? Denumiți mini-simulatorul pe care l-ați proiectat prin **ISLAB\_2A**.
- Studiați convergența ieșirii la răspunsul indicial ideal variind numărul de realizări ale procesului ARX în diferite rulări ale mini-simulatorului **ISLAB\_2A**. Este aparent verificată ipoteza ergodică? (Oferiți toate explicațiile necesare.)

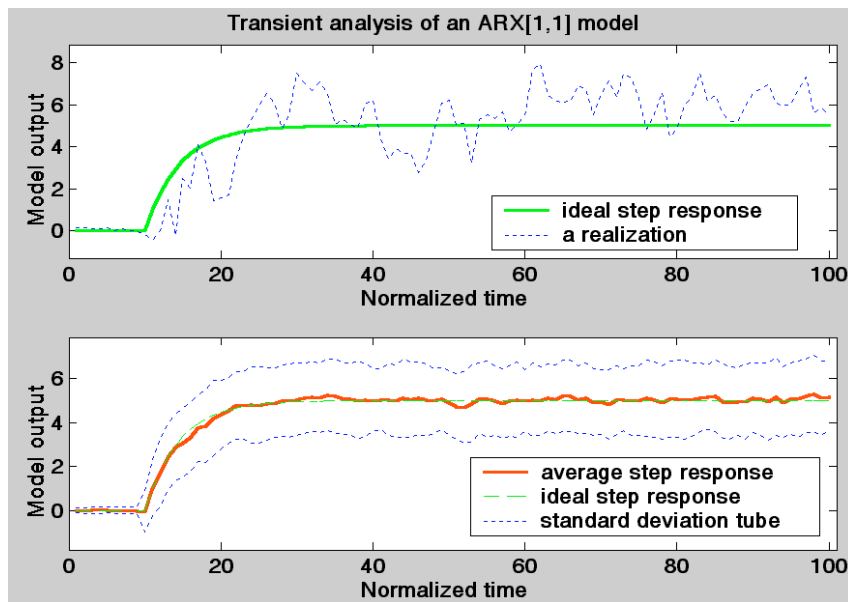


Figura 8. Exemplu de analiză tranzitorie.

- c. Imaginați o tehnică de estimare pe cale grafică a celor 2 parametri  $a$  și  $b$  ai modelului ARX (48) & (50), folosind realizările ieșirii (mai precis zona tranzitorie a acestora).
- d. Reluați simulările pentru modelul OE (49) & (50) (proiectați mini-simulatorul ISLAB\_2B) și comparați rezultatele cu cele ale simulatorului precedent.
- e. Generalizați mini-simulatoarele ISLAB\_2A și ISLAB\_2B pentru cazul unui model ARMAX[na,nb,nc] (adică proiectați mini-simulatorul general ISLAB\_2C) utilizând aceeași intrare (52) și un zgomot alb de dispersie unitară. Rulați simulatorul în cazul modelelor ARX și OE particularizate prin definițiile (51). Comparați rezultatele celor 2 modele. Pot fi determinați parametrii unui sistem de ordin 2 (amplificare, supra-reglaj, pulsație de rezonanță), afectat de zgomot, prin analiză tranzitorie, ca în cazul sistemelor de ordin 1? Dacă nu, argumentați de ce. Dacă da, explicați în ce constă tehnica de identificare.

### Problema 2.2

Această problemă se referă la analiza pe bază de corelație. Nucleul acestui tip de analiză îl constituie ecuația Wiener-Hopf (10) descrisă în Introducere. Cu ajutorul acestei ecuații, se poate determina o mulțime finită de valori ale funcției pondere (răspunsul cauzal la impuls) asociate unui model de sistem cu ieșiri corupte de zgomot. Dorim să determinăm primele  $M = 50$  de valori ale funcției pondere folosind cele 2 modele (48) și (49). Acestea vor fi stimulate cu un SPAB bipolar de lungime  $N = 100$ . Valorile semnalului de intrare sunt doar  $-1$  și  $+1$ . Se vor efectua 100 de experimente.

- a. Să se reprezinte grafic, într-o primă fereastră, funcția pondere ideală a modelului ARX (48) & (50) (adică în absența zgomotului) plus funcția pondere

estimată rezolvînd ecuația Wiener-Hopf, cu ajutorul datelor de intrare-ieșire corespunzătoare primei realizări a procesului. (Folosiți **Exercițiul 2.2** pentru a implementa ecuația generală a funcției pondere ideale.) Într-o a doua fereastră, să se traseze media estimațiilor obținute (în prezența zgomotului), împreună cu secvența pondere ideală și tubul de deviație standard din jurul mediei, ca în **Figura 9**. Și în acest caz se vor folosi funcțiile MATLAB: **filter**, **mean** și **std**. Denumiți mini-simulatorul pe care l-ați proiectat prin **ISLAB\_2D**.

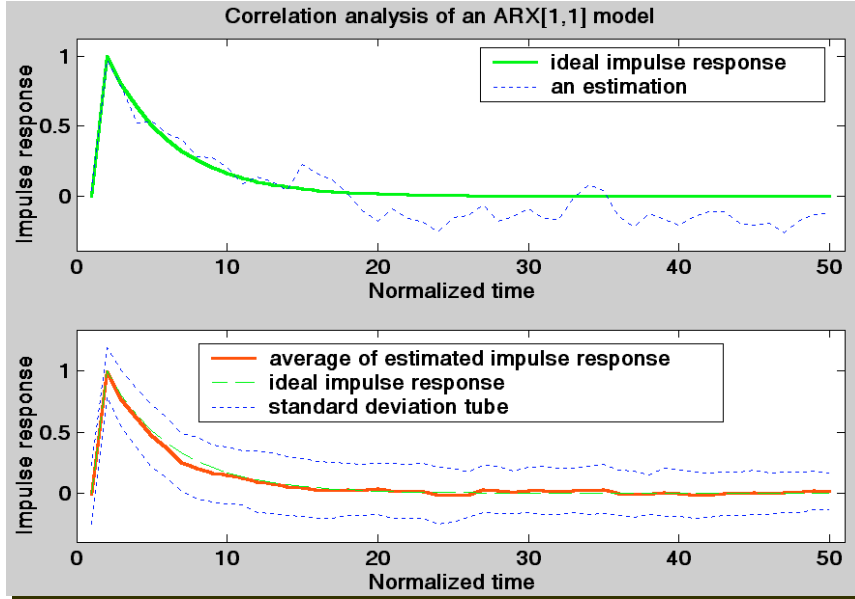


Figura 9. Exemplu de analiză pe bază de corelație.

- Reluați simulările pentru modelul OE (49) & (50) (proiectați mini-simulatorul **ISLAB\_2E**) și comparați rezultatele cu cele ale simulatorului precedent.
- Modificați mini-simulatorele **ISLAB\_2D** și **ISLAB\_2E** astfel încât intrarea de stimul să fie egală cu:

$$u_f[n] \stackrel{\text{def}}{=} \frac{u_0}{1-0.8q^{-1}} u[n], \quad \forall n \in \mathbb{N}, \quad (53)$$

unde  $u_0 = \sqrt{1-(0.8)^2} = 0.6$  este un factor de normare menit să egaleze varianțele lui  $u$  (semnalul SPAB original) și  $u_f$  (versiunea filtrată a semnalului SPAB). Denumiți noile mini-simulatore prin **ISLAB\_2F** și **ISLAB\_2G**, respectiv. Observați că, de această dată, estimația secvenței pondere pare a fi deviată în ambele cazuri. Care credeți că este cauza acestei proprietăți nedorite?

- Pentru a diminua deviația estimației secvenței pondere, se pot aplica 2 tehnici de bază: pre-albire de date sau idealizarea matricii de auto-covarianță a intrării.

## 2. Identificarea modelelor ne-parametrice

- *Pre-albirea datelor.* Funcția MATLAB **cra** efectuează analiza pe bază de corelație însoțită de pre-albirea datelor, dacă utilizatorul o dorește. Această operație constă în filtrarea datelor de intrare-ieșire cu ajutorul unui filtru IIR de tip AR[na]. Implicit, ordinul filtrului este  $na = 10$ , dar utilizatorul poate specifica propria sa opțiune în acest scop. Albirea datelor (mai ales de intrare) conduce la diminuarea deviației estimației. În general, această tehnică este utilizată atunci când nu se cunosc suficiente informații despre maniera în care a fost generată intrarea.

Apelul tipic al funcției **cra** este:

```
[ir,R,cl] = cra(data,M,na,plot) ;
```

unde: **data** este blocul de date măsurate (2 coloane: **[y u]**);

**M** este numărul de valori ale secvenței pondere ce trebuie estimate (implicit: **M=20**);

**na** este ordinul filtrului de albire IIR-AR (implicit: **na=10**); dacă nu se dorește pre-albirea datelor, se poate seta **na=0**;

**plot** este un parametru de afișare grafică; implicit: **plot=1**, care indică trasarea graficului funcției pondere estimate; alte opțiuni recunoscute sunt: **plot=0** (trasarea de grafice este inhibată) și **plot=2** (se trasează graficele tuturor funcțiilor de corelație implicate);

**ir** este răspunsul cauzal la impuls (funcția pondere) estimat(ă);

**R** este o matrice care conține următoarele informații de corelație: pe prima coloană se află pivoții funcțiilor de covarianță; pe coloana a doua se află valorile secvenței de covarianță a ieșirii (după pre-albire, dacă a fost cazul); pe coloana a treia se află valorile secvenței de covarianță a intrării (după pre-albire, dacă este cazul); aceste secvențe pot fi și direct trasate grafic prin apelul: **cra(R)**;

**cl** este nivelul de încredere al estimației funcției pondere.

- *Idealizarea matricii de auto-covarianță a intrării.* Dacă utilizatorul este la curent cu metoda de generare a intrării și poate evalua secvența sa de auto-covarianță, atunci matricea ecuației Wiener-Hopf (adică matricea (8) din Introducere) poate fi implementată direct. În acest context, rezolvarea ecuației (care presupune totuși estimarea corelației încrucișate dintre intrare și ieșire) conduce la estimații cu deviație diminuată.

Folosind fiecare dintre cele 2 tehnici anterioare, să se modifice mini-simulatoarele **ISLAB\_2F** și **ISLAB\_2G** pentru a testa diminuarea deviației estimației în cazul intrării (53). În cazul celei de-a doua tehnici, se va evalua întâi secvența de auto-covarianță a intrării în formă completă. Pentru a construi matricea de auto-covarianță a intrării, se poate folosi funcția MATLAB **toeplitz** (avînd în vedere că această matrice este de tip Toeplitz simetrică). Denumiți mini-simulatoarele obținute prin **ISLAB\_2H** și **ISLAB\_2I** (pentru modelul ARX) și **ISLAB\_2J** și **ISLAB\_2K** (pentru modelul OE).



### Problema 2.3

Ultimul tip de analiză, cea spectrală, va fi ilustrat în contextul acestei probleme. Prin analiza spectrală se urmărește estimarea răspunsului în frecvență al unui proces furnizor de date, folosind ecuația (32) din Introducere. Ecuația poate fi rezolvată dacă se estimează mai întâi densitatea spectrală a intrării ( $\phi_u$ ) și densitatea spectrală încrucișată a intrării și ieșirii ( $\phi_{uy}$ ). Funcția MATLAB care efectuează analiza spectrală plecând de la date măsurate este **spa**. Apelul tipic al acesteia este:

**H = spa(data,M,w) ;**

unde: **data** este blocul de date măsurate (2 coloane: [**y u**]);

**M** este dimensiunea ferestrei Hamming aplicate datelor (implicit: **M = min(length(data)/10,30)**);

**w** este vectorul nodurilor de frecvență unde se dorește estimat răspunsul în frecvență al sistemului;

**H** este răspunsul în frecvență al sistemului.

De notat că fereastra Hamming este una dintre cele mai convenabile pentru estimarea spectrală, avînd expresia:

$$W[n] = 0.54 - 0.46 \cos \frac{2n\pi}{M-1}, \quad \forall n \in \mathbb{N}, \quad (54)$$

unde  $M$  este deschiderea ferestrei, așa cum se poate vedea în Figura 10.

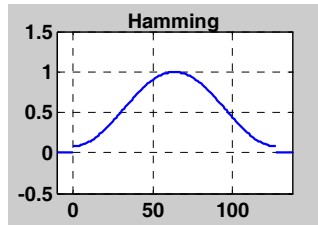


Figura 10. Fereastra spectrală a lui Hamming.

Mini-simulatorul **ISLAB\_2L** (al cărui listing este prezentat în secțiunea următoare) a fost proiectat pentru efectuarea analizei spectrale a modelului ARX (48) & (50), în cazul în care sistemul este stimulat cu intrarea  $u_f$  (definiția (53) din problema precedentă). Diagrama Bode afișată de funcția **ISLAB\_2L** este comparată cu răspunsul în frecvență ideal dedus direct din ecuația modelului (48) (vezi **Exercițiul 2.4**), ca în Figura 11.

- Efectuați cîteva simulări cu diferite valori ale deschiderii ferestrei ( $M$ ) pentru a observa influența acestui parametru asupra calității estimației și a propune o valoare rezonabilă a lui.
- Înlocuiți semnalul de stimul  $u_f$  din mini-simulatorul **ISLAB\_2L** cu un zgomot alb (sau un SPAB). Denumiți noul simulator prin **ISLAB\_2M** și repetați

## 2. Identificarea modelelor ne-parametrice

experimentul de la punctul precedent. Comparați rezultatele celor 2 mini-simulatoare **ISLAB\_2L** și **ISLAB\_2M** pentru cele mai bune valori ale deschiderii ferestrei Hamming găsite în fiecare caz.

- c. Proiectați mini-simulatoarele **ISLAB\_2N** și **ISLAB\_2O** inspirate de cele 2 mini-simulatoare anterioare, dar pentru modelul OE (49) & (50). Efectuați din nou o analiză comparativă.
- d. Proiectați mini-simulatoarele **ISLAB\_2P**, **ISLAB\_2Q**, **ISLAB\_2R** și **ISLAB\_2S** inspirate de cele 4 mini-simulatoare anterioare, dar pentru modelele ARX (48) & (51) și OE (49) & (51). Repetați analiza comparativă.

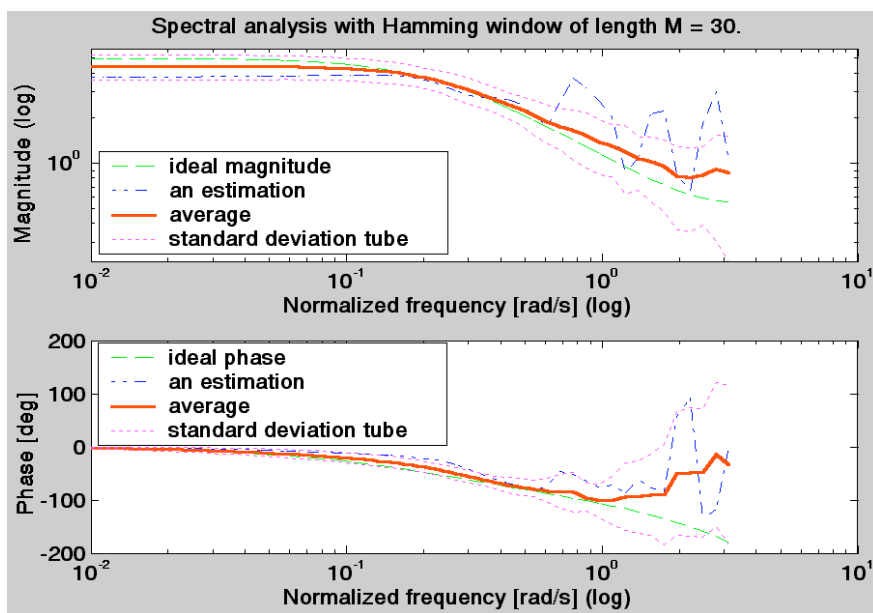


Figura 11. Exemplu de analiză spectrală.

### • Comentarii privind proiectarea mini-simulatorului **ISLAB\_2L**.

Rutina **ISLAB\_2L** utilizează atât câteva funcții MATLAB (versiunea 6.\*) dedicate în special domeniilor IS și TS, cât și tipuri de structuri de date specifice domeniului IS (definite ca obiecte în biblioteca *System Identification* a mediului de programare).

Două astfel de structuri au fost create și exploatate în cadrul rutinei, după cum este explicat în continuare.

- D** este structura datelor de intrare-ieșire generate folosind ecuațiile modelului ales. Structura este creată cu ajutorul funcției (metodei) constructor **iddata** asociată obiectului **IDDATA** (date de identificare). Cele 2 matrici de date **u** (de intrare) și **y** (de ieșire) au dimensiunile identice: **N** linii și **nr** coloane. Fiecare din cele **nr** experimente (realizări) ocupă câte o coloană cu **N** perechi de date intrare-ieșire). Datele se regăsesc în câmpurile **D.u** și **D.y** ale structurii **D**. Structura este mult mai complexă și se compune din următoarele câmpuri:

```

        Domain: 'Time'/'Frequency'
        Name: 'String'
        OutputData: [1x37 char]
        y: 'Same as OutputData'
        OutputName: 'Ny-by-1 cell array of strings'
        OutputUnit: 'Ny-by-1 cell array of strings'
        InputData: [1x36 char]
        u: 'Same as InputData'
        InputName: 'Nu-by-1 cell array of strings'
        InputUnit: 'Nu-by-1 cell array of strings'
        Period: [1x51 char]
        InterSample: [1x36 char]
        Ts: [1x54 char]
        Tstart: 'Scalar (Starting time)'
        SamplingInstants: [1x51 char]
        TimeUnit: 'String'
        ExperimentName: [1x43 char]
        Notes: 'Cell array of strings'
        UserData: 'Arbitrary'

```

**H** este structura datelor reprezentînd răspunsul în frecvență estimat folosind funcția MATLAB **spa** (despre care am amintit). Răspunsurile în frecvență estimate folosind datele **D** se regăsesc în blocul tri-dimensional **H.ResponseData**, astfel: prima estimăție se află în vectorul **H.ResponseData(1,1,:)**, a doua estimăție se află în vectorul **H.ResponseData(2,2,:)**, etc. Sunt **nr** estimății în total. Fiecare estimăție conține **K** date cu valori complexe (partea reală și partea imaginară a răspunsului în frecvență). Numărul **K** figurează printre argumentele funcției **ISLAB\_2L** și reprezintă numărul de noduri (echidistante) de frecvență în care se doresc estimate valorile răspunsului în frecvență. Ca și în cazul structurii de date, structura răspunsului în frecvență este mai complexă, incluzînd următoarele cîmpuri (din care se constată că funcția specializată **spa** poate oferi numeroase informații spectrale sau de corelație referitoare la datele măsurate și zgomet):

```

        Name: 'string'
        Frequency: [1x48 char]
        ResponseData: [1x40 char]
        SpectrumData: [1x38 char]
        CovarianceData: [1x62 char]
        NoiseCovariance: [1x57 char]
        Units: '['rad/s'|'Hz']'
        Ts: 'scalar'
        InputDelay: 'Nu-by-1 vector'
        EstimationInfo: 'structure'
        InputName: 'Nu-by-1 cell array of strings'
        OutputName: 'Ny-by-1 cell array of strings'

```

```

InputUnit: 'Nu-by-1 cell array of strings'
OutputUnit: 'Ny-by-1 cell array of strings'
Notes: 'Array or cell array of strings'
UserData: 'Arbitrary'
Version: 'Internal Use'
Utility: 'Internal Use'

```

Axa celor **K** noduri de frecvență (notată cu **f** în cadrul rutinei **ISLAB\_2L**) a fost generată cu ajutorul funcției Matlab **logspace**, în scară logaritmică, între  $10^{-2}$  și  $\pi$ . Răspunsul ideal în frecvență al modelului de identificare a fost generat folosind funcția MATLAB **dbode** din cadrul bibliotecii *Control* (adică dedicată domeniului TS). Aceasta returnează magnitudinea și faza răspunsului în frecvență (cu care se poate trasa diagrama Bode a sistemului – de unde și denumirea funcției). În acest scop, se folosesc axa de frecvențe **f** și perioada de eșantionare **Ts** setată la valoarea 1 în cadrul rutinei **ISLAB\_2L**. De notat că atât axa de frecvențe cât perioada de eșantionare se regăsesc în cele 2 structuri de date descrise mai sus.

Dacă numărul de realizări considerat (**nr**) este mare, funcția **spa** devine (mare) consumatoare de timp, deoarece ea a fost proiectată să estimeze răspunsul în frecvență pentru fiecare pereche de seturi date de intrare și de ieșire, nu numai pentru intrările și ieșirile care corespund între ele. De exemplu, pentru 100 de realizări, funcția **spa** returnează 10000 de estimări ale răspunsului în frecvență (adică  $100 \times 100$ ), în loc de 100. Cele 100 estimări corecte se regăsesc totuși pe diagonală structurii **H**, cum am explicat mai sus. Rutina **ISLAB\_2L** evită acest efect prin specificarea clară a fiecărei perechi de seturi de date intrare-ieșire corespondente.

Graficele magnitudinii și fazei au fost trasate în axe logaritmice, pentru a scoate în evidență erorile de estimare. Așa cum și **Figura 11** o ilustrează, estimările sunt mai puțin precise la frecvențe înalte, deoarece semnalul de intrare a fost generat prin filtrarea de joasă frecvență a unui SPAB. Mini-simulatorul **ISLAB\_2M** (proiectat să lucreze cu un zgomot alb) ar trebui să corecteze acest defect.

## Capitolul 3

### Identificare parametrică prin Metoda Celor Mai Mici Pătrate

#### 3.1. Contextul general de lucru

Obiectivul acestui capitol este de a familiariza utilizatorii cu metoda fundamentală a domeniului IS, adică MCMMP (descrisă succint în Introducere). Pentru atingerea acestui scop, se pleacă de la contextul de lucru definit în Capitolul 2. Mai precis, modelele de proces ARX (48) și OE (49) vor fi determinate cu ajutorul MCMMP în cazurile particulare (50) și:

$$A(q^{-1}) = 1 - 0.4q^{-1} - 0.32q^{-2}, \quad B(q^{-1}) = 0.5q^{-1} + 0.03q^{-2}. \quad (55)$$

În cazul particular (55), se poate constata cu ușurință că fiecare polinom posedă câte o rădăcină parazită (adică de magnitudine sensibil mai mică decât cealaltă rădăcină sau decât unitatea).

Datele experimentale necesare estimării parametrilor necunoscuți, adică  $\mathcal{D} = \{u[n]\}_{n=1, \overline{N}} \cup \{y[n]\}_{n=1, \overline{N}}$ , vor fi generate cu ajutorul modelelor avînd parametri adevărați și diferite intrări (în principal  $u$  și  $u_f$  din **Problema 2.2**). Zgomotul de proces,  $e$ , este, ca de obicei, alb Gaussian de dispersie unitară ( $\lambda^2 = 1$ ). Dispersia zgomotului va fi de asemenea estimată folosind MCMMP. În cadrul problemelor de simulare, se vor efectua câte 100 de experimente de identificare (ca în cazul problemelor din Capitolul 2), în timp ce dimensiunea orizontului de măsură va fi  $N = 100$ .

#### 3.2. Exerciții

##### Exercițiul 3.1

Determinați ecuațiile de estimare a parametrilor necunoscuți (coeficienți și dispersie zgomot alb) pentru modelului ARX (48) & (50), în formă completă, folosind relațiile (19) și (20) caracteristice MCMMP. Evaluați limitele teoretice ale parametrilor pentru o colecție infinită de date. În ce condiții parametrii estimați sunt consistenți (adică tind la valorile adevărate)?

##### Exercițiul 3.2

Determinați ecuațiile de estimare a parametrilor necunoscuți (coeficienți și dispersie zgomot alb) pentru modelului ARX (48) & (55), folosind MCMMP.

##### Exercițiul 3.3

Este posibilă utilizarea MCMMP pentru a determina parametrii modelului OE (49) & (50)? Dacă nu, argumentați răspunsul. Dacă da, determinați ecuațiile de estimare a parametrilor necunoscuți (coeficienți și dispersie zgomot alb), folosind MCMMP. Tot în acest caz, studiați consistența estimațiilor.

**Exercițiul 3.4**

Este posibilă utilizarea MCMMP pentru a determina parametrii modelului OE (49) & (55)? Dacă nu, argumentați răspunsul. Dacă da, determinați ecuațiile de estimare a parametrilor necunoscuți (coeficienți și dispersie zgomot alb), folosind MCMMP.

**Indicație**

- Pentru a testa identificabilitatea modelelor OE, se recomandă exprimarea ecuației (49) într-o formă echivalentă de ecuație cu diferențe, în care fiecare parametru necunoscut apare ca factor într-un singur termen. În acest fel, se poate observa cum ieșirea modelului este direct afectată de zgomotul alb.

**3.3. Probleme de simulare****Problema 3.1**

Se studiază influența semnalului de intrare asupra calității estimății oferite de MCMMP pentru modelele ARX (48) & (50), respectiv (48) & (55). Aceste modele vor fi stimulate de câte 100 de ori cu fiecare din cele 2 intrări ale **Problemei 2.2** (adică  $u$  – un SPAB bipolar de lungime 100, avînd doar valorile  $-1$  sau  $+1$  și  $u_f$  – un semnal de joasă frecvență generat ca în (53), prin filtrarea semnalului  $u$ ). După achiziția datelor de intrare-ieșire  $\mathcal{D} = \{u[n]\}_{n=1,100} \cup \{y[n]\}_{n=1,100}$ , se vor implementa relațiile de calcul ale estimățiilor parametrilor necunoscuți din **Exercițiile 3.1** și **3.2**. Estimățiile parametrilor vor fi mediate peste ansamblul celor 100 de realizări și li se va calcula deviația standard. Cele 4 mini-simulatoare obținute vor fi denumite prin: **ISLAB\_3A** (model ARX[1,1] & intrare  $u$ ), **ISLAB\_3B** (model ARX[1,1] & intrare  $u_f$ ), **ISLAB\_3C** (model ARX[2,2] & intrare  $u$ ) și **ISLAB\_3D** (model ARX[2,2] & intrare  $u_f$ ).

- Pentru fiecare mini-simulator, să se reprezinte grafic într-o figură erorile de estimare a răspunsului în frecvență după cum urmează.
  - În prima fereastră va fi trasat graficul erorii de estimare a amplitudinii răspunsului în frecvență, adică media amplitudinii diferenței dintre răspunsul în frecvență ideal (în absența zgomotului) și răspunsurile în frecvență obținute din cele 100 de realizări (după estimarea parametrilor necunoscuți). Tubul de dispersie a amplitudinii se va evalua ca în problemele din Capitolul 2 pentru fiecare eroare de estimare și se va trasa pe același grafic.
  - În a doua fereastră va fi trasat graficul erorii de estimare a fazei răspunsului în frecvență, adică media fazei diferenței dintre răspunsul în frecvență ideal (în absența zgomotului) și răspunsurile în frecvență obținute din cele 100 de realizări (după estimarea parametrilor necunoscuți). Se va evalua tubul de dispersie a fazei pentru fiecare eroare de estimare și se va trasa pe același grafic.

Într-o a doua figură vor fi trasate graficul dispersiei estimate a zgomotului, (care este obținută în fiecare realizare a procesului) și graficul valorii adevărate a

dispersiei zgomotului ( $\lambda^2 = 1$ ). Afișați în cadrul figurii valorile parametrilor adevărați și media valorilor parametrilor estimați (calculată peste ansamblul realizărilor). Un exemplu este ilustrat în [Figurile 12 și 13](#).

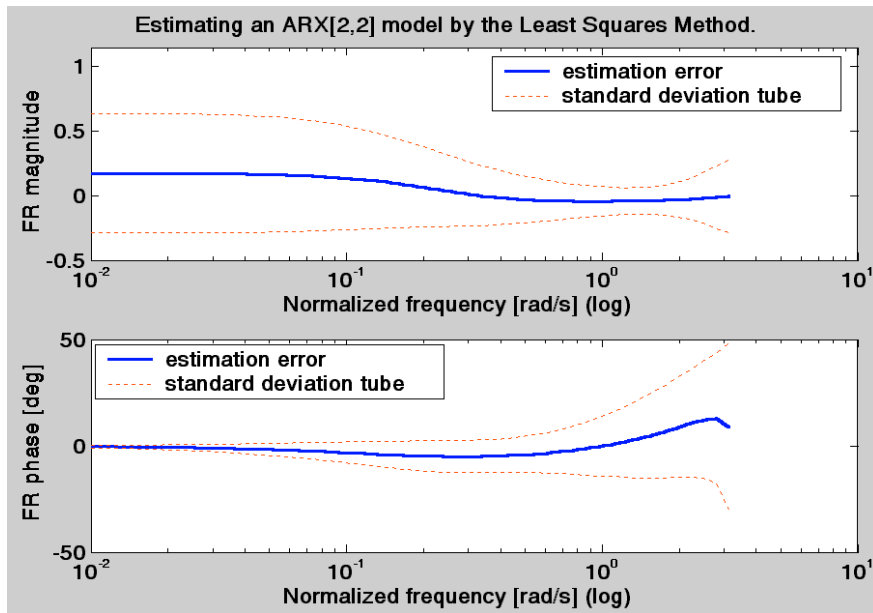


Figura 12. Exemplu de afișare a erorii de estimare cu MCMMP (răspuns în frecvență).

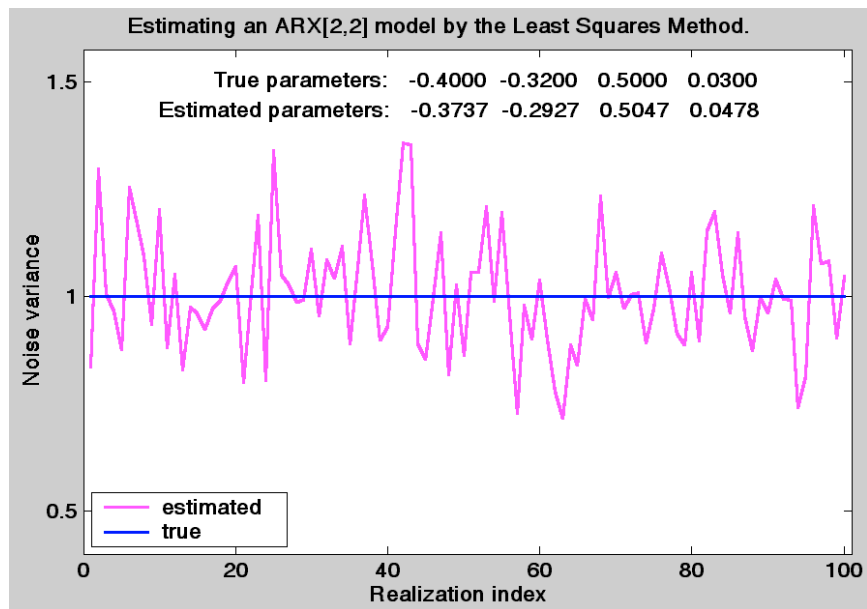


Figura 13. Exemplu de afișare a erorii de estimare cu MCMMP (dispersie zgomot).

### 3. Identificare parametrică prin MCMMP

Pentru determinarea răspunsurilor în frecvență se va utiliza funcția MATLAB **dbode** (ca în cadrul problemelor de simulare din Capitolul 2). Nu va fi în nici un caz utilizată funcția de analiză spectrală **spa**, deoarece răspunsul în frecvență estimat se va obține prin combinația dintre MCMMP și **dbode**. De asemenea, în cazul modelului ARX[2,2], funcțiile de covarianță implicate de relațiile de estimare ale MCMMP pot fi evaluate cu precizie ridicată folosind funcția MATLAB **xcov**, dacă este utilizată cu atenție.

- b. Comentați rezultatele obținute la punctul precedent. Observați influența tipului de intrare asupra estimării rădăcinilor parazite din modelul particular (48) & (55). Dacă acest proces nu va putea fi stimulat decât cu intrări de joasă frecvență, cum credeți că s-ar putea estima (fie și imprecis) rădăcinile parazite?

#### Problema 3.2

Dacă ați ajuns la concluzia că modelele OE (49) & (50), respectiv (49) & (55) ar putea fi identificate cu ajutorul MCMMP, reluați **Problema 3.1** pentru cazul acestor modele. Denumiți mini-simulatoarele obținute prin **ISLAB\_3E** (model OE[1,1] & intrare  $u$ ), **ISLAB\_3F** (model OE[1,1] & intrare  $u_f$ ), **ISLAB\_3G** (model OE[2,2] & intrare  $u$ ) și **ISLAB\_3H** (model OE[2,2] & intrare  $u_f$ ).

#### Problema 3.3

Generalizați mini-simulatoarele anterioare și denumiți noile rutine prin **ISLAB\_3I** (pentru modele ARX[na,nb]) și, dacă este cazul, **ISLAB\_3J** (pentru modele OE[na,nb]). În acest scop, se poate utiliza funcția de bibliotecă IS MATLAB numită **arx**. Apelul tipic al acestei rutine este următorul:

```
theta = arx(D,si) ;
```

unde: **D** este structura datelor de intrare-ieșire, de regulă creată cu ajutorul funcției (metodei) constructor asociată obiectului **IDDATA** (vezi comentariile privind proiectarea mini-simulatorului **ISLAB\_2L** din finalul Capitolului 2);

**si** este vectorul indicilor structurali și al întârzierii modelului:

```
si = [na nb nk] ,
```

unde **na** este ordinul componentei AR, iar **nb+nk** este ordinul componentei X; practic, **nk** este numărul de coeficienți nuli ai polinomului  $B$ , pînă la primul coeficient nenul de grad minim (adică întârzierea intrinsecă a modelului); urmează cei **nb** coeficienți nenuli.

Argumentul de ieșire **theta** este la rîndul său un obiect de tip **IDPOLY** (polinom de identificare – în cazul modelelor SISO) sau **IDMODEL** (model general de identificare în cazul modelelor MIMO). Un obiect **IDMODEL** conține cîmpurile:

```
a: 'A-polynomial (row vector) '  
b: 'B-polynomial (row vector) '  
c: 'C-polynomial (row vector) '  
d: 'D-polynomial (row vector) '  
f: 'F-polynomial (row vector) '  
da: 'standard deviation of a (scalar) '
```



```

db: 'standard deviation of b (scalar)'
dc: 'standard deviation of c (scalar)'
dd: 'standard deviation of d (scalar)'
df: 'standard deviation of f (scalar)'
na: 'order of A-polynomial (scalar)'
nb: 'order of B-polynomial (scalar)'
nc: 'order of C-polynomial (scalar)'
nd: 'order of D-polynomial (scalar)'
nf: 'order of F-polynomial (scalar)'
nk: 'delay of B-polynomial (scalar)'
InitialState: [1x45 char]
  Name: 'string'
  Ts: 'sample time in seconds (scalar)'
  InputName: 'Nu-by-1 cell array of strings'
  InputUnit: 'Nu-by-1 cell array of strings'
  OutputName: 'Ny-by-1 cell array of strings'
  OutputUnit: 'Ny-by-1 cell array of strings'
  TimeUnit: 'string'
ParameterVector: 'Np-by-1 vector'
  PName: 'Np-by-1 cell array of strings'
CovarianceMatrix: 'Np-by-Np matrix'
  NoiseVariance: 'Ny-by-Ny matrix'
  InputDelay: 'Nu-by-1 vector'
  Algorithm: [1x38 char]
  EstimationInfo: [1x39 char]
  Notes: 'Array or cell array of strings'
  UserData: 'Arbitrary'

```

Evident, polinoamele  $A$  și  $B$  se regăsesc în câmpurile: `theta.a`, respectiv `theta.b`. În `theta.b` sunt salvați atât coeficienții nenuli cât și cei nuli (datorați întârzierii intrinseci) ai polinomului  $B$ . Ordinele polinoamelor sunt memorate în `theta.na`, respectiv `theta.nb`, iar întârzierea intrinsecă – în `theta.nk`.

## Capitolul 4

### Identificare parametrică prin Metoda Variabilelor Instrumentale

#### 4.1. Contextul general de lucru

##### A. Metoda Variabilelor Instrumentale

Una dintre primele metode de identificare concepute plecând de la MCMMP a fost *Metoda Variabilelor Instrumentale* (MVI). Aceasta este de regulă folosită în contextul modelelor ARX, dar poate fi adaptată și pentru alte modele (de exemplu OE sau chiar ARMAX). În forma ei generală, estimația oferită de MVI este următoarea (sugerată de estimația dată de MCMMP – vezi ecuația (19) din Introducere):

$$\hat{\theta}_N \stackrel{\text{def}}{=} \left( \frac{1}{N} \sum_{n=1}^N z[n] \varphi^T[n] \right)^{-1} \left( \frac{1}{N} \sum_{n=1}^N z[n] y[n] \right), \quad (56)$$

unde vectorul instrumentelor (sau al variabilelor instrumentale)  $z[n] \in \mathbb{R}^{n_\theta}$  poate fi construit de către utilizator la fiecare moment de timp normalizat  $n \in \mathbb{N}^*$  (în funcție de tipul modelului de identificare cu care se operează). Ecuația (56) trebuie completată cu ecuațiile (20) referitoare la zgomot.

Diferența fundamentală dintre ecuațiile (19) și (56) constă în faptul că estimația (19) oferită de MCMMP a rezultat în urma rezolvării problemei de optimizare (21), în timp ce estimația (56) oferită de MVI este pur și simplu o definiție. În mod evident, se pune atunci problema corectitudinii definiției și consistenței estimației rezultate. Se poate arăta că estimația (56) este bine definită și consistentă dacă vectorul instrumentelor este ales astfel încât:

$$E\{z[n] \varphi^T[n]\} \text{ este inversabilă și } E\{z[n] e[n]\} = 0. \quad (57)$$

De notat că, spre deosebire de MCMMP, (57) arată că în MVI nu este absolut necesar ca zgomotul  $e$  să fie alb (poate fi și colorat), cu condiția ca el să nu fie corelat cu instrumentele vectorului  $z$ . De altfel, în cazul modelelor ARX, se poate arăta că estimația oferită de MVI este consistentă în următoarele condiții: modelul este parsimonios (adică polinoamele  $A$  și  $B$  sunt coprime) și intrarea  $u$  este un zgomot alb necorelat cu zgomotul (nu neapărat alb)  $e$ . Acest rezultat arată superioritatea MVI asupra MCMMP, în cazul modelelor ARX: nu e necesar să presupunem că zgomotul de proces este alb, în schimb procesul trebuie stimulat cu o intrare fabricată artificial care să aproximeze zgomotul alb.

Vectorul instrumentelor poate fi ales în mai multe moduri, în funcție de modelul de lucru. În cazul modelelor ARX, două alegeri sunt frecvente:

$$z[n] \stackrel{\text{def}}{=} [u[n-1] \ u[n-2] \ \cdots \ u[n-na-nb]]^T. \quad (58)$$

și:

$$z[n] \stackrel{def}{=} \begin{bmatrix} u_f[n-1] & u_f[n-2] & \cdots & u_f[n-na] & | & u[n-1] & u[n-2] & \cdots & u[n-nb] \end{bmatrix}^T, \quad (59)$$

unde  $u_f$  este un semnal obținut prin filtrarea intrării, adică:

$$u_f[n] \stackrel{def}{=} \frac{D(q^{-1})}{C(q^{-1})} u[n], \quad \forall n \in \mathbb{N}^*. \quad (60)$$

În definiția (60), polinoamele  $C$  și  $D$  se aleg fie simplu, de forma:  $C(q^{-1})=1$ ,  $D(q^{-1})=q^{-nd}$  (cu  $nd \in \mathbb{N}$  o întârziere fixată), fie, mai sofisticat, de exemplu identice cu polinoamele  $A$ , respectiv  $B$  estimate folosind MCMMP.

### B. Criterii de alegere a structurii modelelor

Condiția de parsimonie exprimată printre cerințele suficiente de consistență a estimației oferite de MVI se poate îndeplini printr-un proces iterativ de îmbogățire a structurii modelului (plecînd de la modelul cel mai simplu), așa cum a fost descris în Introducere. Criteriile uzuale de evaluare a erorii dintre model și proces (care conduc la determinarea indicelui structural optim) sunt descrise în continuare.

- a. *Criteriul aplatizării erorii pătratice.* În mod normal, modelul de identificare cu indicele structural  $n\theta$  este un caz particular de model cu indicele structural  $n\theta + 1$ . În acest fel, se poate utiliza funcția criteriu din (21), adică:

$$\mathcal{V}_N(\theta) \stackrel{def}{=} \sum_{n=1}^N (\tilde{y}[n] - \tilde{\varphi}[n]\theta)^2, \quad \forall \theta \in \mathcal{S}, \quad (61)$$

unde:  $\tilde{y} \equiv y - \bar{y}$ ,  $\tilde{\varphi} \equiv \varphi - \bar{\varphi}$  (datele se centrează pe medie), iar  $\mathcal{S}$  este domeniul de stabilitate al modelului matematic. Teoretic, funcția criteriu:

$$\mathcal{A}_N[n\theta] \stackrel{def}{=} \mathcal{V}_N(\hat{\theta}_N) = N\hat{\lambda}_N^2[n\theta], \quad \forall n\theta \in \mathbb{N}^*, \quad (62)$$

descrește odată cu mărirea dimensiunii estimației  $\hat{\theta}_N$  (adică  $n\theta$ ). În definiția (62), am rennotat dispersia estimată a zgomotului  $\hat{\lambda}_N^2$  prin  $\hat{\lambda}_N^2[n\theta]$ , pentru a pune în evidență dependența de indicele structural  $n\theta$ .

Practic, însă, așa cum este sugerat în Figura 14(a), valoarea lui  $\mathcal{A}_N[n\theta]$  descrește pînă la un anumit indice structural  $n\theta_{\max}$ , după care începe să crească, în principal din două cauze: acumularea erorilor de calcul și particularizarea prea accentuată a modelului la cazul datelor măsurate. În aceste condiții, indicele structural optim  $n\theta_{\text{opt}}$  este indicat de intrarea în zona de aplatizare (adică de palier) a graficului lui  $\mathcal{A}_N$ . Astfel, indicele structural optim este selectat în funcție de dispersia zgomotului, care este invers

#### 4. Identificare parametrică prin MVI

proporțională cu precizia modelului. Deîndată ce nu se mai obține o scădere semnificativă a acestei dispersii, adică o creștere semnificativă a preciziei modelului, este inutilă mărirea complexității acestuia.

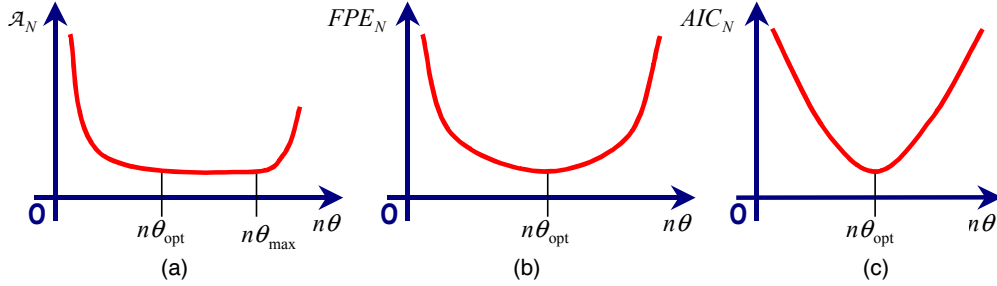


Figura 14. (a) Criteriul aplatizării erorii pătratice. (b) Criteriul FPE. (c) Criteriul AIC.

Un criteriu înrudit cu  $\mathcal{A}_N$  este  $\mathcal{L}_N$ , definit ca determinant al matricii de covarianță a erorilor de predicție cu un pas (exprimate mai jos, în ecuația (70)). În literatură,  $\mathcal{L}_N$  se mai numește și *funcție de pierdere (loss function)*.

- b. *Criteriul descreșterii relative normalizate*. Un criteriu alternativ se bazează pe *Testul F* din Statistică, care exprimă câștigul relativ de precizie înregistrat odată cu creșterea complexității modelului:

$$\mathcal{F}_N[n\theta] \stackrel{\text{def}}{=} \frac{\hat{\lambda}_N^2[n\theta] - \hat{\lambda}_N^2[n\theta + 1]}{\hat{\lambda}_N^2[n\theta + 1]}, \quad \forall n\theta \in \mathbb{N}^*, \quad (63)$$

Astfel, indicele structural optim este cel mai mic indice pentru care  $\mathcal{F}_N[n\theta] \leq 4/N$ . Acest test arată că precizia trebuie să înregistreze un câștig suficient de mic în raport cu o valoare stabilită în mod adaptiv, în funcție de dimensiunea orizontului de măsură, pentru a stopa creșterea complexității modelului. Criteriul este util mai ales în cazul în care variația dispersiei zgomotului cu  $n\theta$  prezintă un palier (ca în Figura 14(a)). În acest fel, spre deosebire de cazul criteriului precedent, aici alegerea indicelui structural optim se poate efectua într-o manieră automată, nesubiectivă.

- c. *Criteriul de penalizare FPE (Final Prediction Error)*. Existența unui palier la nivelul dispersiei de zgomot este în general nedorită. Mai precisă ar fi determinarea indicelui structural optim dintr-o gamă îngustă de valori. Îngustarea intervalului în care se află indicele structural optim se poate realiza prin aplicarea unei penalizări asupra dispersiei zgomotului. Criteriul FPE (adică al *erorii finale de predicție*) propune o penalizare cu factorul  $(N + n\theta)/(N - n\theta)$ , ca mai jos:

$$FPE_N[n\theta] \stackrel{\text{def}}{=} \hat{\lambda}_N^2[n\theta] \frac{N + n\theta}{N - n\theta}, \quad \forall n\theta \in \mathbb{N}^*, \quad (64)$$

Efectul penalizării este ilustrat în Figura 14(b), unde se observă îngustarea suportului palierului zonei de optim, deși, în general, o ușoară aplatizare

persistă. În acest caz indicele structural optim rezultă efectiv prin rezolvarea următoarei probleme de minimizare:

$$n\theta_{\text{opt}} = \arg \min_{n\theta \in \mathbb{N}^*} FPE_N[n\theta]. \quad (65)$$

- d. *Criteriile lui Akaike-Rissanen.* Pentru a pune în evidență și mai precis indicele structural optim al modelului matematic ales, cercetătorul japonez H. Akaike a propus în [AkH69] aplicarea unei penalizări de tip logaritm asupra dispersiei zgomotului:

$$AIC_N[n\theta] \stackrel{\text{def}}{=} \ln(\hat{\chi}_N^2[n\theta]) + \frac{2n\theta}{N}, \quad \forall n\theta \in \mathbb{N}^*. \quad (66)$$

Este bine știut faptul că aplicarea logaritmului conduce la ascuțirea extremelor unei funcții (dar și la apariția unor extreme locale “ascunse”). Figura 14(c) sugerează acest efect. Între criteriile FPE (64) și AIC (66) există o legătură, așa cum o demonstrează **Exercițiul 4.1**. (De altfel, criteriul FPE a fost propus tot de către Akaike.) Ambele criterii tind totuși să supra-parametrizeze modelul ales, în timp ce criteriul aplatizării și Testul F tind să îl sub-parametrizeze. Akaike propune o generalizare a criteriului său, menită să corecteze efectul supra-parametrizării:

$$GAIC_N[n\theta] \stackrel{\text{def}}{=} \ln(\hat{\chi}_N^2[n\theta]) + \frac{2n\theta}{N} \alpha_N, \quad \forall n\theta \in \mathbb{N}^*. \quad (67)$$

În definiția (67), factorul de corecție  $\alpha_N > 1$  poate fi ales astfel:  $\alpha_N \in [2, 4]$  (independent de  $N$ ) sau  $\alpha_N \in \{\ln(N), \ln(\ln(N))\}$  (adaptiv, în funcție de  $N$ ). O altă generalizare a criteriului AIC, tot de tip adaptiv, a fost propusă de către cercetătorul finlandez J. Rissanen în [RiJ78]:  $\alpha_N = \ln(\sqrt{N})$ . Această generalizare conduce la o descriere de proces bazată pe un așa numit *model de lungime minimală*, descriere conformă Principiului parsimoniei.

O majorare excesivă a factorului corector poate conduce însă la sub-parametrizare. De notat totuși că sub-parametrizarea este mai puțin dezirabilă decât supra-parametrizarea, deoarece nu se dorește în nici un caz pierderea de informație printr-o modelare inadecvată a procesului. Și în acest caz, indicele structural optim rezultă prin rezolvarea unei probleme de minimizare:

$$n\theta_{\text{opt}} = \arg \min_{n\theta \in \mathbb{N}^*} (G)AIC_N[n\theta]. \quad (68)$$

### Notă

- J. Rissanen este autorul unei metode universale de compresie de date bazată pe utilizarea contextelor [RiJ83]. Tot el introdus conceptul de “descriere de lungime minimală” (*Minimum Description Length*) [RiJ78], care constituie o exprimare a Principiului parsimoniei și care este utilizată și în compresia datelor.

- e. *Criteriul gradului de potrivire.* În practică, un criteriu extrem de utilizat atât pentru verificarea Principiului parsimoniei cât și pentru validarea modelelor este cel bazat pe evaluarea “potrivirii” (*fitness*) dintre model și proces după următoarea relație:

$$\mathcal{E}_N[n\theta] \stackrel{def}{=} 100 \left( 1 - \frac{\sum_{n=1}^N |\varepsilon[n, \hat{\theta}_N]|^2}{\sum_{n=1}^N \left| y[n] - \frac{1}{N} \sum_{n=1}^N y[n] \right|^2} \right) [\%], \quad \forall n\theta \in \mathbb{N}^*, \quad (69)$$

unde prin  $\varepsilon[n, \hat{\theta}_N]$  s-a notat eroarea de predicție cu un pas, definită astfel:

$$\varepsilon[n, \hat{\theta}_N] \stackrel{def}{=} \underbrace{y[n]}_{\text{date reale}} - \underbrace{\phi^T[n] \hat{\theta}_N}_{\text{date simulate}}, \quad \forall n \in \mathbb{N}^*. \quad (70)$$

Valoarea funcției de potrivire  $\mathcal{E}_N$  este exprimată în procente. Cu cât aceasta este mai apropiată de 100%, cu atât modelul decodifică mai bine informația despre comportamentul procesului furnizor de date și va fi mai bine validat (în sensul criteriilor de validare descrise în paragraful C). Adesea,  $\mathcal{E}_N[n\theta]$  este interpretată ca procentaj al procesului care a fost “explicat” de către model sau valoarea cuantificată a gradului de validare a modelului. Totuși, limita superioară a funcției de potrivire  $\mathcal{E}_N$  depinde de cantitatea de zgomot de proces care corupe datele măsurate, adică de SNR. Mai mult,  $\mathcal{E}_N$  poate avea și valori negative, în cazul erorilor de predicție importante. În fine, o ultimă proprietate nedorită a acestui criteriu este cea a existenței palierului de potrivire, similar cu palierul dispersiei zgomotului din [Figura 14\(a\)](#), cu deosebirea că graficul funcției de potrivire este concav și nu convex. În aceste condiții, indicele structural optim trebuie ales la intrarea în palierul de potrivire.

- f. *Criteriul reprezentării poli-zeroouri.* Un alt criteriu de verificare a Principiului parsimoniei (dar mai mult de tip subiectiv) se bazează pe reprezentarea poli-zeroouri. Mai precis, polii și zerourile modelului determinat sunt reprezentați în planul complex, împreună cu zonele aferente de încredere. O zonă de încredere este reprezentată ca un disc circular de rază proporțională cu deviația standard a polului sau zeroului, centrat în acesta. Deviația standard se evaluează cu ajutorul diagonalei matricii de covarianță a erorii de estimare (vezi **Exercițiul 4.2**). Pe harta localizării polilor și zerourilor se poate observa relativ ușor orice pol și zero care ar trebui eliminați prin simplificare, datorită amplasării lor aproximativ în aceeași vecinătate. Se poate stabili chiar o distanță minimă între un pol și un zero, distanță sub care polul și zeroul pot fi considerați identici. Simplificarea polilor și zerourilor este necesară în spiritul Principiului parsimoniei. Adecvanța modelului se poate testa prin verificarea a 3 proprietăți: stabilitate (polii modelului trebuie să se situeze în interiorul discului unitar), suprafețele discurilor de încredere (care trebuie să fie cât mai mici) și apropiere de harta poli-zeroouri a procesului (dacă această hartă este

disponibilă). Mărimea suprafeței unui disc de încredere este totuși informația esențială: cu cât aceasta este mai redusă, cu atât modelul este mai precis, avînd șanse mai mari de validare.

În pofida naturaleții lor, nici unul din criteriile descrise mai sus nu este “perfect”. De aceea, se recomandă testarea tuturor criteriilor, înainte de a decide valoarea indicelui structural optim. Este de dorit ca el să fie indicat de majoritatea criteriilor. Dar dacă fiecare criteriu indică o altă valoare, se va recurge doar la Testul F (care este criteriul dominant) și, eventual, la unul dintre criteriile GAIC. O inspectare a reprezentării polierourii este de asemenea recomandată.

### C. Criterii de validare a modelelor

Odată determinat, orice model matematic trebuie validat. Validarea constă practic în comparația dintre un set de date achiziționate și setul de date simulate cu ajutorul modelului, ambele fiind generate prin stimularea cu același semnal de intrare. În această secțiune, vom prezenta pe scurt doar 2 dintre metodele de validare corespunzătoare MCMMP și MVI, în cazul în care datele achiziționate din proces au o distribuție Gaussiană. (Alte distribuții atrag după sine metode de validare diferite.) Ambele metode sunt bazate pe așa numitul *Test de albire*, care va fi descris în continuare.

#### ➤ Validarea modelelor determinate cu ajutorul MCMMP.

Principiul care stă la baza criteriului de validare este următorul: dacă modelul determinat este adecvat, eroarea de predicție dintre datele simulate și cele achiziționate tinde să fie un zgomot alb Gaussian pe măsură ce orizontul de măsură crește (de unde și numele de *Test de albire* atribuit criteriului de validare). Proprietatea de necorelare se exprimă prin:

$$\lim_{N \rightarrow \infty} E\{\varepsilon[n, \hat{\theta}_N] \varepsilon[n-k, \hat{\theta}_N]\} = 0, \quad \forall k \in \mathbb{N}^*, \quad (71)$$

unde  $\varepsilon[n, \hat{\theta}_N]$  este eroarea de predicție cu un pas, definită în (70).

Condiția (71) are 2 dezavantaje majore: este greu (dacă nu imposibil) de verificat în practică și nu face referire la tipul de distribuție a erorii de predicție. De aceea, o formă practică a testului de albire se bazează pe proprietățile distribuției Gaussiene de medie nulă și deschidere  $\sigma = \sigma_0 / \sqrt{N}$  (adaptiv, în funcție de dimensiunea orizontului de măsură). Parametrul  $\sigma_0$  este determinat de gradul de corelație existent între valorile variabilelor aleatoare avînd această distribuție. Orice astfel de variabilă aleatoare produce valori într-un interval oarecare  $[-\rho, +\rho]$  cu un nivel de încredere  $\mathcal{N}(\rho)$ . Nivelul de încredere exprimă de fapt probabilitatea ca variabila aleatoare să producă valori în intervalul specificat, deci este egal cu aria de sub graficul distribuției peste acel interval. Zgomotului alb Gaussian îi corespund intervale și nivele de încredere tipice asociate din Tabelul 1. Informația din tabel poate fi fructificată în proiectarea versiunii practice a Testului de albire. Pentru aceasta, se evaluează mai întîi un număr de valori ale secvenței de auto-corelație  $\rho_\varepsilon$  asociate erorii de predicție:

Tabelul 1. Intervale și nivele de încredere tipice pentru validarea modelelor.

$[-\rho, +\rho]$	$\left[-\frac{2.17}{\sqrt{N}}, +\frac{2.17}{\sqrt{N}}\right]$	$\left[-\frac{1.96}{\sqrt{N}}, +\frac{1.96}{\sqrt{N}}\right]$	$\left[-\frac{1.808}{\sqrt{N}}, +\frac{1.808}{\sqrt{N}}\right]$
$\mathcal{N}(\rho)$	97%	95%	93%

$$r_\varepsilon[k] \stackrel{\text{def}}{=} \frac{1}{N-k} \sum_{n=k+1}^N \varepsilon[n, \hat{\theta}_N] \varepsilon[n-k, \hat{\theta}_N], \quad \forall k \in 0, \left\lceil \frac{N}{4} \right\rceil \text{ (auto-covarianță);}$$

$$\rho_\varepsilon[k] \stackrel{\text{def}}{=} \frac{r_\varepsilon[k]}{r_\varepsilon[0]}, \quad \forall k \in 0, \left\lceil \frac{N}{4} \right\rceil \text{ (auto-corelație).} \quad (72)$$

Evaluarea din (72) se oprește la aproximativ un sfert din dimensiunea orizontului de măsură, deoarece, dincolo de acest prag, erorile de calcul acumulate devin importante. (În suma de definiție a funcției de auto-covarianță rămân din ce în ce mai puțini termeni.) Pasul următor este să se contorizeze numărul de valori ale secvenței de auto-corelație  $\rho_\varepsilon$  ce aparțin fiecăruia din intervalele de încredere ale tabelului anterior. Acestea se normalizează apoi cu numărul total de valori calculate ale  $\rho_\varepsilon$  (adică  $\lceil N/4 \rceil + 1$ ) și se exprimă în procente. În final, se compară procentajele obținute cu nivelele de încredere ale tabelului. Pentru un interval de încredere ales, Testul de albire este *pozitiv* (adică modelul este validat) dacă procentul de valori ale lui  $\rho_\varepsilon$  din interval este cel puțin egal nivelul de încredere corespunzător. Astfel, criteriul oferă 4 nivele de validare:

Nivel 0: nici unul din cele 3 Teste de albire nu este pozitiv (model și/sau metodă de identificare invalide).

Nivel 1: doar unul din cele 3 Teste de albire este pozitiv (model și/sau metodă de identificare la limita de validitate).

Nivel 2: două din cele 3 Teste de albire sunt pozitive (model și/sau metodă de identificare valide, dar cu validitate limitată; pentru anumite tipuri de intrări, modelul s-ar putea să nu funcționeze corect).

Nivel 3: toate cele 3 Teste de albire sunt pozitive (model și/sau metodă de identificare valide, cu validitate extinsă la majoritatea covârșitoare a tipurilor de intrări).

➤ **Validarea modelelor determinate cu ajutorul MVI.**

Spre deosebire de modelele determinate cu ajutorul MCMMP, pentru modelele estimate prin MVI principiul de validare este următorul: dacă modelul determinat este adecvat, eroarea de predicție este asimptotic necorelată cu ieșirea predictată centrată (adică obținută prin simularea modelului, după ce s-a scăzut media), avînd totodată distribuție Gaussiană. Necorelarea valorilor erorii de predicție cu cele ale ieșirii simulate centrate înseamnă:

$$\lim_{N \rightarrow \infty} E\left\{ \varepsilon[n, \hat{\theta}_N] y_N[n-k] \right\} = 0, \quad \forall k \in \mathbb{Z}, \quad (73)$$



unde:

$$y_N[n] \stackrel{def}{=} \varphi^T[n] \hat{\theta}_N - E\{\varphi^T[n] \hat{\theta}_N\}, \quad \forall n \in \mathbb{N}^*. \quad (74)$$

Condiția (73) are aceleași dezavantaje ca și condiția (71), astfel încât Testul practic de albire este conceput în mod similar (folosind tot Tabelul 1). Singura deosebire constă în evaluarea corelației încrucișate dintre  $\varepsilon$  și  $y_N$  în loc de auto-corelația lui  $\varepsilon$ . Astfel relațiile (72) trebuie înlocuite cu relațiile următoare:

$$\begin{aligned} r_{\varepsilon, y_N}[k] &\stackrel{def}{=} \frac{1}{N-k} \sum_{n=k+1}^N \varepsilon[n, \hat{\theta}_N] y_N[n-k], \quad \forall k \in 0, \left\lceil \frac{N}{4} \right\rceil \text{ (covarianță încrucișată);} \\ \rho_{\varepsilon, y_N}[k] &\stackrel{def}{=} \frac{r_{\varepsilon, y_N}[k]}{\sqrt{r_{\varepsilon}[0]} \sqrt{r_{y_N}[0]}}, \quad \forall k \in 0, \left\lceil \frac{N}{4} \right\rceil \text{ (corelație încrucișată).} \end{aligned} \quad (75)$$

### Observație

- În practica IS, se obișnuiește ca mulțimea de date achiziționate din proces să fie împărțită în două seturi: unul destinat estimării parametrilor și altul destinat validării modelului obținut. Este bine să nu se utilizeze același set de date atât pentru estimare cât și pentru validare, deoarece, în acest fel, se elimină posibilitatea validării unor modele care sunt mult prea acordate la setul de date de identificare achiziționate. Datele simulate trebuie însă generate cu aceleași intrări cu care au fost produse datele de validare, altfel modelul riscă să fie declarat invalid, deși el este în realitate valid.

Obiectivul acestui capitol este de a realiza o comparație între metodele MCMMP și MVI, plecând de la identificarea unor modele ARX.

## 4.2. Exerciții

### Exercițiul 4.1

Arătați că între criteriile FPE și AIC există următoarea corelație, pentru  $N \gg n\theta$ :

$$AIC_N[n\theta] \cong \ln(FPE_N[n\theta]), \quad \forall n\theta \in \mathbb{N}^*. \quad (76)$$

### Exercițiul 4.2

Fie procesul stocastic descris de următoarea ecuație (de tip AR[1]):

$$\mathcal{P}: y[n] + ay[n-1] = v[n], \quad \forall n \in \mathbb{N}^*, \quad (77)$$

unde  $v$  este un zgomot alb de medie nulă și dispersie  $\lambda^2$ . Procesul furnizează datele de ieșire  $\mathcal{D} = \{y[n]\}_{n=1, \overline{N}}$ .

- Să se estimeze parametrii necunoscuți (coeficienți și dispersie de zgomot) pentru următoarele modele, folosind MCMMP și setul de date măsurate:

#### 4. Identificare parametrică prin MVI

$$\mathcal{M}_1 : y[n] + a_{11}y[n-1] = \varepsilon[n, a_{11}], \quad \forall n \in \mathbb{N}^*; \quad (78)$$

$$\mathcal{M}_2 : y[n] + a_{21}y[n-1] + a_{22}y[n-2] = \varepsilon[n, a_{21}, a_{22}], \quad \forall n \in \mathbb{N}^*. \quad (79)$$

În ecuațiile (78) și (79),  $\varepsilon$  este eroarea dintre model și proces, cu proprietatea:  $\varepsilon[n, a] = v[n]$ , respectiv  $\varepsilon[n, a, 0] = v[n]$ .

- b. Testați consistența estimațiilor obținute la punctul precedent (pentru coeficienți și dispersii de zgomot).
- c. Potrivit Teoremei fundamentale a MCMMP, dispersia erorii de estimare a coeficienților necunoscuți este dată în general de:

$$E\left\{(\hat{\theta}_N - \theta^*)(\hat{\theta}_N - \theta^*)^T\right\} = \lambda^2 \left[ \sum_{n=1}^N \varphi[n] \varphi^T[n] \right]^{-1}. \quad (80)$$

Folosind această proprietate, evaluați dispersiile erorilor de estimare ale parametrului  $a$  din cele 2 modele, notate cu  $\sigma_N^2[1]$ , respectiv  $\sigma_N^2[2]$ . (Pentru modelul  $\mathcal{M}_2$ , vectorul parametrilor adevărați este  $\theta^* = [a \ 0]^T$ .) Arătați că:

$$\lim_{N \rightarrow \infty} (N\sigma_N^2[1]) \leq \lim_{N \rightarrow \infty} (N\sigma_N^2[2]). \quad (81)$$

Ce semnificație are inegalitatea (81)?

- d. Notați estimațiile dispersiei prin  $\lambda_N^2[1]$ , respectiv  $\lambda_N^2[2]$  (după indicele structural al modelului utilizat). Evaluați criteriile  $\mathcal{F}_N$  (63) și FPE (64) de determinare a indicelui structural optim pentru fiecare din cele 2 estimații. Rezultă din comparația lor că indicele structural optim este  $n\theta_{\text{opt}} = 1$ ? Argumentați răspunsul.

#### Exercițiul 4.3

Deduceți expresiile estimațiilor oferite de MVI pentru un model ARX[1,1] și un vector al instrumentelor de tipul (58). Studiați consistența lor și precizați un set de condiții suficiente pentru verificarea acestei proprietăți. Determinați condițiile generale de consistență în cazul în care nici intrarea nici zgomotul nu sunt neapărat albe.

#### Exercițiul 4.4

Reluați exercițiul precedent pentru un vector al instrumentelor de tipul (59), unde filtrul aplicat intrării este determinat de estimațiile coeficienților evaluate cu MCMMP. Dacă, prin șansă, MCMMP ar conduce chiar la valorile adevărate ale parametrilor necunoscuți, arătați că estimațiile oferite de MVI pentru cele 2 tipuri de vectori ai instrumentelor (din acest exercițiu și din exercițiul precedent) sunt identice. Care credeți că este semnificația acestui rezultat interesant? Cum poate fi el exploatat?

### Indicație

- Identitatea a 2 estimații oferite de MVI se poate arăta pe 2 căi. Prima cale, mai laborioasă (și mai puțin elegantă), presupune calculul efectiv al estimațiilor. A doua cale, mai elegantă, se bazează pe o proprietate interesantă a estimației MVI: invarianța la transformări liniare ale vectorului instrumentelor. Încercați să demonstrați această proprietate și apoi găsiți transformarea liniară dintre cei 2 vectori ai instrumentelor din cadrul exercițiului.

## 4.3. Probleme de simulare

Mini-simulatoarele propuse pentru a fi proiectate în cadrul acestui capitol sunt focalizate în jurul unui model ARX[2,2]:

$$(1 - 1.5q^{-1} + 0.7q^{-2})y[n] = (q^{-1} + 0.5q^{-2})u[n] + v[n], \quad \forall n \in \mathbb{N}^*, \quad (82)$$

unde  $v$  este zgomotul de proces. Acesta este obținut prin filtrarea zgomotului alb Gaussian  $e$  de medie nulă și dispersie  $\lambda^2 = 1$ , cu ajutorul următorului sistem cu răspuns finit la impuls (FIR sau MA):

$$v[n] = (1 - q^{-1} + 0.2q^{-2})e[n], \quad \forall n \in \mathbb{N}^*. \quad (83)$$

Practic, (82) și (83) sunt ecuațiile procesului furnizor de date (un proces de tip ARMAX). Procesul este stimulat cu un SPAB bipolar având valorile +1 și -1. Intrarea  $u$  este necorelată cu zgomotul alb  $e$ . Datele generate  $\mathcal{D} = \{u[n]\}_{n=1, \overline{N}} \cup \{y[n]\}_{n=1, \overline{N}}$  sunt înregistrate pe un orizont de măsură de dimensiune  $N = 250$ . Pentru generarea și stocarea datelor, se recomandă scrierea unei rutine separate, numite **gendata**, al cărei apel general să fie următorul:

**[D,V,P] = gendata(A,B,C,nk,N,sigma,lambda) ;**

unde: **A** este vectorul coeficienților polinomului  $A$  (implicit: **A=[1 -1.5 0.7]**);  
**B** este vectorul coeficienților polinomului  $B$  (implicit: **B=[1 0.5]**);  
**C** este vectorul coeficienților filtrului  $C$  (implicit: **C=[1 -1 0.2]**);  
**nk** este întârzierea intrinsecă a sistemului (implicit: **nk=1**);  
**N** este dimensiunea orizontului de măsură (implicit: **N=250**);  
**sigma** este deviația standard a intrării SPAB (implicit: **sigma=1**);  
**lambda** este deviația standard a zgomotului alb Gaussian (implicit: **lambda=1**);  
**D** este obiectul de tip **IDDATA** (vezi **Problema 2.3**) corespunzător datelor generate (intrarea se regăsește în **D.u**, iar ieșirea în **D.y**);  
**V** este obiectul de tip **IDDATA** corespunzător zgomotelor generate (zgomotul alb se regăsește în **V.u**, iar zgomotul colorat (adică MA-filtrat) în **V.y**);  
**P** este obiectul de tip **IDMODEL** (vezi **Problema 3.3**) corespunzător modelului de proces furnizor de date.

#### 4. Identificare parametrică prin MVI

Pentru ușurința proiectării acestei rutine, se poate apela la 2 funcții Matlab dedicate, existente în biblioteca destinată domeniului IS: **idpoly** și **sim** (descrise în continuare)

##### # **IDPOLY**

- Apel: **Mid = idpoly(A,B,C,D,F,lambda2,Ts)** ;
- Generează un obiect de tip model de identificare (**IDPOLY** sau **IDMODEL**) **Mid**, avînd structura descrisă în cadrul **Problemei 3.3**. Modelul corespunde ecuației generale:

$$A(q^{-1})y[n] = \frac{B(q^{-1})}{F(q^{-1})}u[n] + \frac{C(q^{-1})}{D(q^{-1})}e[n], \quad \forall n \in \mathbb{N}, \quad (84)$$

unde  $A, \dots, F$  sunt polinoame corespunzătoare. (Restul notațiilor din (84) sunt cunoscute.) În consecință, argumentele de intrare ale funcției sunt:

**A ... F** polinoamele modelului (exprimate sub formă de vectori cu coeficienții ordonați după puterile crescătoare ale lui  $q^{-1}$ ); de notat că, în funcție de tipul de model adoptat, unele dintre aceste polinoame pot lipsi, lor fiindu-le atribuite valori implicite; implicit, polinomul  $B$  este nul, în timp ce restul polinomelor sunt unitare; cu toate acestea, dacă, de exemplu, se dorește generarea unui model de tip OE, apelul tipic al funcției este:

**Mid = idpoly(1,B,1,1,F,lambda2,Ts)** ;

(adică toate polinoamele trebuie specificate explicit);

**lambda2** varianța zgomotului alb, adică  $\lambda^2$  (implicit: **lambda2=1**);

**Ts** perioada de eșantionare (implicit: **Ts=1**).

##### # **SIM**

- Apel: **[y,ystd] = sim(Mid,ue)** ;
- Rutină care simulează comportamentul unui model de identificare **Mid** pentru intrări și zgomote specificate în **ue**. Argumentul **Mid** este un obiect de tip model de identificare (**IDPOLY** sau **IDMODEL**), returnat, de exemplu, de rutina **idpoly**. Argumentul **ue** este fie un obiect de tip date de identificare (**IDDATA**, descris în contextul **Problemei 2.3**), fie o matrice formată din blocurile **[u e]**, unde **u** este matricea/vectorul intrărilor iar **e** este matricea/vectorul zgomotelor. În cazul modelelor MIMO, fiecare coloană a matricilor **u** sau **e** reprezintă un canal de intrare sau ieșire, după caz. Pentru sistemele SISO, **u** și **e** sunt vectori. Rezultatul simulării este returnat în **y** (ieșirea sistemului), care are aceeași natură ca și **ue** (obiect **IDDATA** sau matrice/vector). Utilizatorul are posibilitatea de a cere calcularea deviației standard a ieșirilor, care va fi returnată în **ystd**.

#### **Observație**

- Rutina MATLAB **sim** are 2 exprimări (permise de filozofia programării orientate obiect). În nucleul de funcții generale, ea are rolul de a lansa în execuție, prin program, simulatorul SIMULINK. Aceasta este definiția de bază. În biblioteca de funcții de IS, ea are rolul de a

## Aspecte practice în Modelarea și Identificarea Sistemelor

simula funcționarea unui model de identificare. Aceasta este forma supra-definită. Pentru a obține o informație ajutătoare mai completă referitoare la **sim** ca funcție de bibliotecă IS, se poate executa comanda: **help idmodel/sim**.

Rutina **gendata** va fi practic apelată de 2 ori: prima dată pentru generarea datelor de identificare și a doua oară pentru generarea datelor de validare. De fiecare dată se vor utiliza aceleași tipuri de intrări și zgomote (adică SPAB bipolar și zgomot alb Gaussian de dispersie unitară).

Estimarea modelelor pe baza datelor astfel generate se poate efectua cu ajutorul următoarelor rutine din cadrul bibliotecii de IS: **arx** (pentru MCMMP, descrisă în contextul **Problemei 3.3**), **iv** (pentru MVI și modele ARX-SISO) sau **iv4** (pentru MVI și modele ARX-MIMO). În general, se recomandă utilizarea rutinei **iv4**. Dacă modelul este de tip SISO, rutina **iv** poate fi însă mai rapidă. În cadrul MATLAB 6.\*, rutina **iv** nu funcționează corect, fiind înlocuită de **iv4**. De aceea, vom descrie pe scurt această rutină ultimă. Numele ei provine de la faptul că estimăția este evaluată în 4 etape de calcul:

1. Se identifică modelul în mod grosier, cu ajutorul MCMMP (rutina **arx**).
2. Modelul anterior este folosit pentru a genera vectorul instrumentelor plecând de la intrarea specificată în cadrul datelor măsurate, prin filtrare. Cu acest vector, se estimează un nou model ARX, folosind MVI.
3. Reziduurile modelului obținut (adică erorile de predicție) sunt asociate unui model AR de ordin foarte mare, care este identificat folosind din nou MCMMP.
4. Datele de intrare-ieșire originale sunt filtrate folosind modelul AR anterior. Parametrii modelului sunt estimați în final folosind datele rezultate (filtrate) și același tip de vector al instrumentelor ca la pasul 2.

Această strategie (fundamentată teoretic în [LJL99]) și implementată în cadrul rutinei **iv4** conduce la estimații de precizie ridicată, cu condiția ca datele de intrare inițiale să aproximeze zgomotul alb.

# **IV4**

- Apel: **Mid = iv4(D, si)** ;
- Estimează parametrii unui model ARX folosind MVI. Modelul identificat rezultat, **Mid**, este returnat ca obiect **IDMODEL**. Estimarea se efectuează pe baza datelor **D** (obiect **IDDATA**) și a informației de structură **si = [na nb nk]**, unde **na** și **nb** sunt indicii structurali ai modelului, iar **nk** este întârzierea intrinsecă.

În cadrul mini-simulatoarelor care urmează, se vor utiliza 2 dintre criteriile de determinare a structurii optime: descreșterea relativă normalizată (Testul F), adică  $\mathcal{F}_N$  definit în (63) și Akaike generalizat de către Rissanen, adică GAIC definit în (67), cu factorul corector adaptiv  $\alpha_N = \ln(\sqrt{N})$ . De asemenea, se vor reprezenta grafic criteriul aplatizării  $\mathcal{A}_N$ , criteriul potrivirii  $\mathcal{E}_N$  și localizarea poli-zero-uri.

Particularitatea cea mai importantă a evaluării criteriilor de determinare a indicelui structural optim în cazul modelelor cu cel puțin 2 polinoame (cum este și ARX) constă în faptul că argumentul funcției criteriu este vectorial. Aceasta deoarece indicele structural global  $n\theta$  se exprimă ca o sumă de indici structurali parțiali – gradele polinoamelor. În cazul modelului ARX:  $n\theta = na + nb$ . Pentru fiecare valoare a lui  $n\theta$

#### 4. Identificare parametrică prin MVI

există un număr (finit) de posibilități de alegere a gradelor  $na$  și  $nb$ . Pentru  $n\theta > 1$ , numărul de posibilități este superior lui 1, deci  $na$  și  $nb$  trebuie selectați dintr-o matrice de valori. Dacă  $Na$  și  $Nb$  sunt valorile maxime ale gradelor polinoamelor, numărul total de indici structurali testați ajunge la  $(Na+1) \cdot (Nb+1) - 1$  (incluzând și valorile nule, dar nu simultan nule ale indicilor), deci matricea poate avea de exemplu  $Na+1$  linii și  $Nb+1$  coloane, cu elementul (1,1) virtual (deoarece corespunde valorilor nule ale celor 2 indici structurali  $na$  și  $nb$ ). Indicele generic al matricii este  $[na+1, nb+1]$ , cu  $na \in \overline{0, Na}$  și  $nb \in \overline{0, Nb}$ .

Se recomandă proiectarea a 2 rutine de calcul pentru criteriile de determinare a structurii optimale: **F\_test2** și **GAIC\_R2**. Argumentele de intrare ale fiecărei rutine sunt: o matrice  $\Lambda_N \in \mathbb{R}^{Na \times Nb}$  cu elementul generic:  $\Lambda_N[na+1, nb+1] = \lambda_N^2[na, nb]$  (folosind notații naturale) și  $N$ . Practic  $\Lambda_N$  oferă valorile criteriului aplatizării. Ele se obțin folosind direct modelul identificat **Mid**:  $\lambda_N^2[na, nb] \leftarrow \mathbf{Mid.NoiseVariance}$ . (În mod asemănător, valoarea funcției de pierdere se obține din **Mid.es.LossFcn**.)

În acest context, exprimarea criteriului  $\mathcal{F}_N$  va fi ușor diferită de definiția originală (63), deși respectînd același principiu. Vecinii imediați ai lui  $\Lambda_N[na+1, nb+1]$  sunt  $\Lambda_N[na+2, nb+1]$  și  $\Lambda_N[na+1, nb+2]$ . Aceasta sugerează, potrivit definiției (63), scăderea liniilor și coloanelor adiacente ale matricii  $\Lambda_N$  pentru evaluarea criteriului  $\mathcal{F}_N$ . Optimul va fi selectat din 2 matrici de valori astfel obținute, suma indicilor săi trebuind să fie minimă. Exprimarea criteriului GAIC în versiunea Rissanen se poate face însă direct cu ajutorul definiției (67):

$$GAIC_N[na, nb] \stackrel{def}{=} \ln(\hat{\lambda}_N^2[na, nb]) + \frac{\sqrt{N}}{N}(na + nb), \quad \forall na \in \overline{0, Na}, \quad \forall nb \in \overline{0, Nb}, \quad (85)$$

urmînd ca optimul să fie selectat prin căutare directă în matricea de valori rezultată (care are aceleași dimensiuni ca și  $\Lambda_N$ ). Printr-o adaptare inspirată, valoarea oricărui criteriu GAIC se poate obține direct folosind modelul identificat **Mid**. Astfel, **fpe(Mid)** returnează valoarea criteriului FPE (definiția (64)).

Funcția de potrivire  $\mathcal{E}_N$  poate fi evaluată indirect, cu ajutorul rutinei **resid** din biblioteca de IS, descrisă mai jos.

# **RESID**

- Apel: **E = resid(Mid, D)** ;
- Rutină care evaluează *reziduurile* (adică erorile de predicție ale) modelului **Mid** (obiect **IDMODEL**) plecînd de la datele **D** (obiect **IDDATA**). Rezultatul, **E**, este tot un obiect **IDDATA**. Erorile de predicție se regăsesc în **E.y**, în timp ce **E.u** este identic cu **D.u**. Dacă rutina este apelată fără argument de ieșire, graficele auto-covarianței erorii de predicție și al covarianței încrucișate dintre erorile de predicție și intrări sunt trasate (adică este efectuată o analiză bazată pe corelație).

Odată ce erorile de predicție au fost estimate, se poate implementa direct definiția (69), unde datele măsurate la ieșire se preiau din obiectul de date **D** (mai precis, ele sunt salvate în **D.y**).

O alternativă de evaluare a funcției de potrivire constă în utilizarea funcției **compare** din biblioteca de IS.

### # COMPARE

- Apel: **[ym,EN] = compare(Mid,D) ;**
- Rutină care efectuează o comparație între datele de ieșire obținute prin simularea modelului **Mid** (obiect de tip **IDMODEL**) și datele de ieșire măsurate salvate în obiectul **D** (de tip **IDDATA**), adică **D.y**. Pentru comparație, modelul este stimulată cu aceeași intrare **D.u** cu care au fost generate datele **D**. Funcția returnează valorile ieșirii simulate **ym** și, dacă se dorește, valoarea de potrivire dintre model și proces **EN** (adică  $\mathcal{E}_N$  din definiția (69)). Între ieșirea simulată a unui model evaluată cu ajutorul acestei funcții și cea evaluată cu ajutorul funcției **sim** există o ușoară deosebire: în contextul funcției **sim**, condițiile inițiale ale ecuației cu diferențe asociate modelului sunt nule; în contextul funcției **compare**, valoarea inițială (în origine) a ieșirii este unitară. Astfel, de exemplu, ieșirea simulată a unui model AR fără zgomot este nulă pentru **sim** și egală cu răspunsul cauzal la impuls pentru **compare**.

Dacă rutina este apelată fără argumente de ieșire, graficele ieșirii măsurate și ieșirii simulate sunt trasate, iar gradul de potrivire dintre ele este afișat.

Reprezentarea poli-zerouri (împreună cu discurile de încredere) poate fi efectuată folosind funcția de bibliotecă IS numită **PZMAP**.

### # PZMAP

- Apel: **pzmap(Mid,'SD',alpha) ;**
- Rutină de reprezentare poli-zerouri pentru modelul **Mid** (obiect de tip **IDMODEL**). Dacă argumentele de intrare **'SD'** și **alpha** sunt precizate, discurile de încredere asociate polilor și zerourilor sunt de asemenea trasate. Razele lor sunt egale cu deviațiile standard multiplicare de valoarea lui **alpha** (care trebuie să fie un număr nenegativ). Dacă **alpha=0** (care este și valoarea implicită), trasarea discurilor de încredere este inhibată. De regulă, pentru date cu distribuție Gaussiană, **alpha=3**.

În biblioteca de IS din MATLAB, este propusă și o altă abordare de selectare a indicilor structurali optimi, care are avantajul că poate fi generalizată la orice model din clasa generată de ecuația (84), dar dezavantajul că doar criteriile lui Akaike-Rissanen sunt evaluate. Testul F implică o manieră de implementare relativ complicată în acest caz. Dacă este interesat, utilizatorul poate studia grupul de funcții: **arxstruc**, **ivstruc**, **selstruc** și **struc**.

Pentru validarea, modelelor, se recomandă proiectarea rutinelor **valid\_LS** (MCMMP) și **valid\_IV** (MVI). Oricare din cele 2 rutine va returna un întreg între 0 și 3 care indică gradul de validare (după cum a fost explicat în paragraful C al primei secțiuni). Rutinele pot folosi funcția MATLAB **xcorr** pentru evaluarea secvențelor de corelație.

În cadrul problemelor de simulare, se va considera că  $Na = Nb = 8$ .

**Problema 4.1**

Să se proiecteze mini-simulatorul **ISLAB\_4A** care evaluează estimația (parsimonioasă a) modelului ARX asociat procesului (82) & (83), folosind MCMMP. Pentru aceasta, se vor parcurge următorii pași:

- a. Se generează 2 seturi de date: unul pentru identificare și altul pentru validare, folosind rutina **gendata**.
- b. Pentru fiecare model identificat cu ajutorul MCMMP (funcția **arx**), model obținut variind indicii  $na$  și  $nb$ , se vor afișa 2 ferestre grafice: una pentru analiza modelului folosind datele de identificare și de validare, alta pentru reprezentarea poli-zerouri cu discuri de încredere corespunzătoare unei raze de 3 ori mai mari decât deviațiile standard aferente (ca în **Figurile 15** și **16**). După fiecare fereastră se inserează o pauză de așteptare pentru a permite utilizatorului să analizeze informațiile afișate. Fiecare sub-fereastră a primei ferestre include 3 grafice aranjate pe verticală:
  - ieșirile măsurate și simulate cu ajutorul modelului, grafic pe care se indică și valoarea funcției de potrivire,  $\mathcal{E}_N$ ;
  - eroarea de predicție (reziduurile modelului), grafic pe care se indică și dispersia estimată a zgomotului,  $\lambda_N^2$ ;
  - secvența de auto-covarianță a erorii de predicție, grafic pe care se indică și indexul de validare.

Modelele obținute vor fi memorate în vederea selectării unuia dintre ele, în urma aplicării testelor de determinare a indicilor structurali optimi și de validare.

- c. Se reprezintă grafic (în ferestre consecutive):
  - suprafața dispersiei zgomotului în decibeli ( $10\lg(\lambda_N^2)$ ) și optimul selectat folosind Testul F (vezi **Figura 17**);
  - suprafața funcției de potrivire ( $\mathcal{E}_N$ ) pentru datele de identificare și optimul selectat folosind tot Testul F, dar adaptat corespunzător (vezi **Figura 18**);
  - suprafața funcției de potrivire ( $\mathcal{E}_N$ ) pentru datele de validare și optimul selectat folosind Testul F adaptat (vezi **Figura 19**);
  - suprafața criteriului GAIC în versiunea Rissanen și optimul indicat de aceasta (vezi **Figura 20**).
- d. Se solicită utilizatorului să aleagă indicii structurali pe care îi consideră optimi.
- e. Pentru modelul ales, se afișează cele 2 ferestre grafice de la b. Modelul este returnat de către mini-simulator, în vederea unei utilizări ulterioare. Se recomandă returnarea și a seturilor de date de identificare și validare.

După proiectarea mini-simulatorului **ISLAB\_4A**, se vor iniția câteva rulări.

Rezultă mereu aceiași indici structurali optimi sau ei diferă de la o rulare la alta? Justificați răspunsul. Observați simplificarea polilor și zerourilor apropiate din diagrama poli-zerouri, pentru indici structurali mari. Care dintre criteriile de determinare a structurii optime are tendința de a sub-parametriza modelul și care – de a supra-parametriza modelul?



### **Problema 4.2**

Problema anterioară, 4.1, se va relua pentru cazul MVI cu instrumentele (58). Mini-simulatorul rezultat va fi denumit **ISLAB\_4B**. În acest scop, se pot utiliza majoritatea funcțiilor mini-simulatorului **ISLAB\_4A**. Excepție face, de exemplu, testul de validare, care trebuie schimbat (se va proiecta rutina **valid\_IV**). Comparați rezultatele de estimare obținute cu performanțele estimației evaluate folosind MCMMP.

### **Problema 4.3**

Generalizați mini-simulatoarele anterioare astfel încât utilizatorului să i se permită să își aleagă metoda de identificare (MCMMP sau MVI) și instrumentele în cazul MVI. Tipul de model identificat rămâne același: ARX. Denumiți noul mini-simulator prin **ISLAB\_4C**. Testați mini-simulatorul cu datele de intrare ale mini-simulatoarelor precedente. Rulați apoi mini-simulatorul cu opțiunile: MVI și instrumentele (59) & (60), unde, în prealabil, trebuie produs un model estimat folosind MCMMP. Comparați performanțele estimațiilor obținute cu MVI pentru cele 2 tipuri de instrumente: (58) și (59)-(60). Arătați avantajele și dezavantajele fiecărei strategii de estimare.

#### 4. Identificare parametrică prin MVI

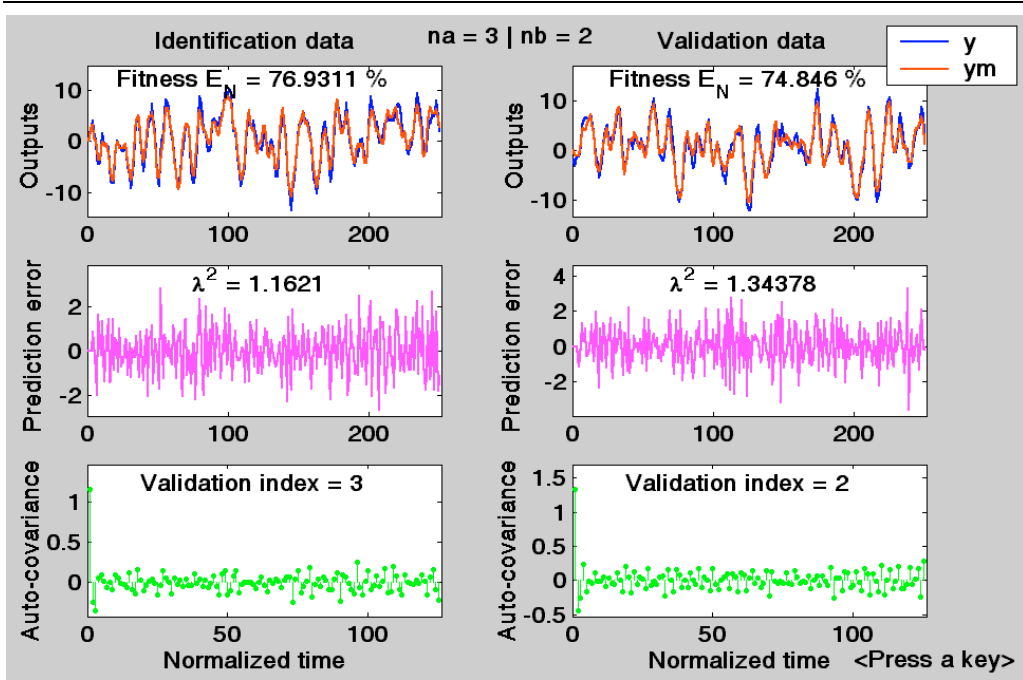


Figura 15. Performanțele unui model estimat cu MCMMP.

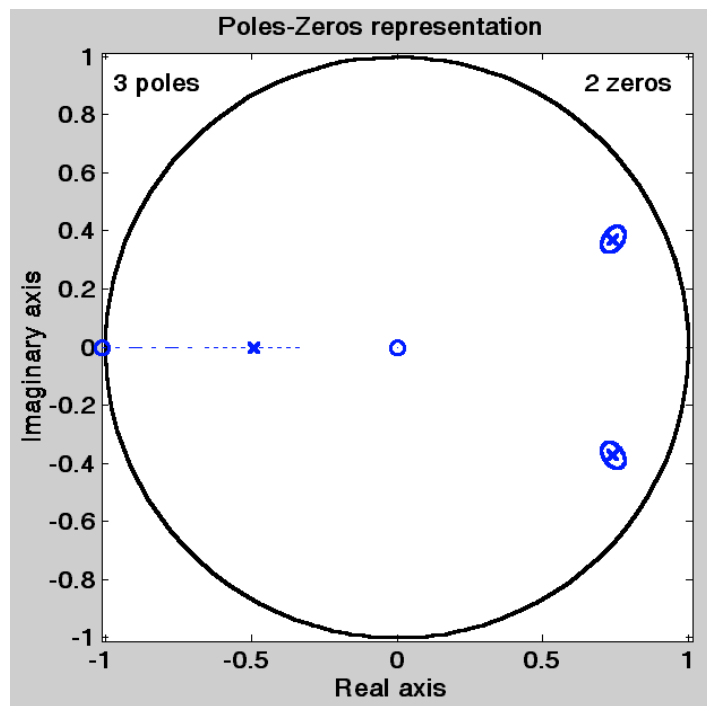


Figura 16. Reprezentarea poli-zerouri a unui model estimat cu MCMMP.

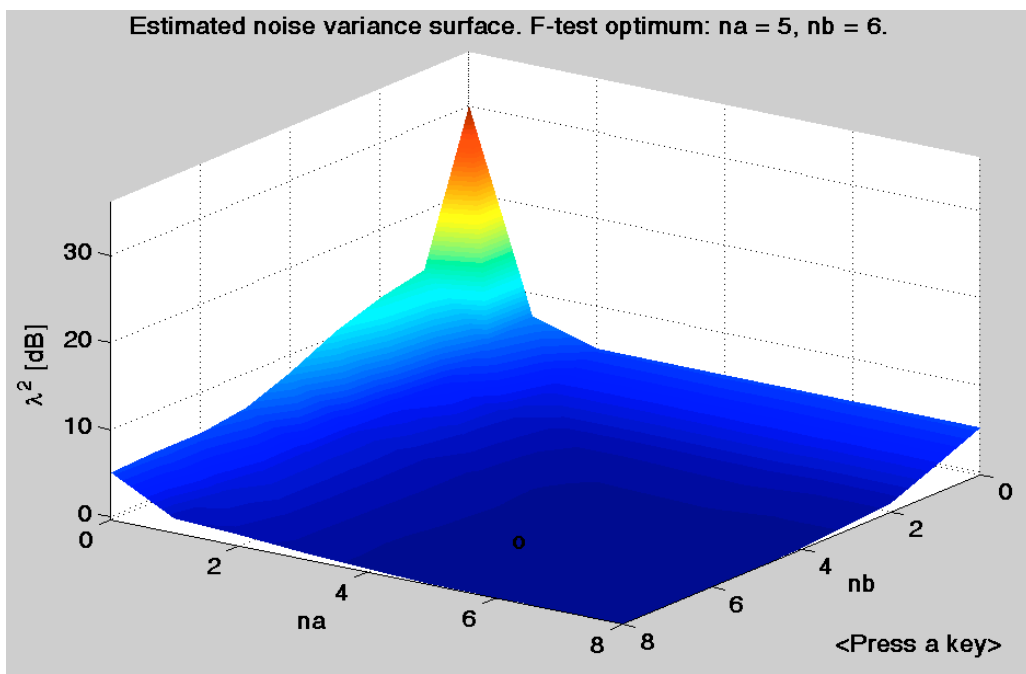


Figura 17. Dispersia estimată a zgomotului. Criteriul aplatizării și Testul F.

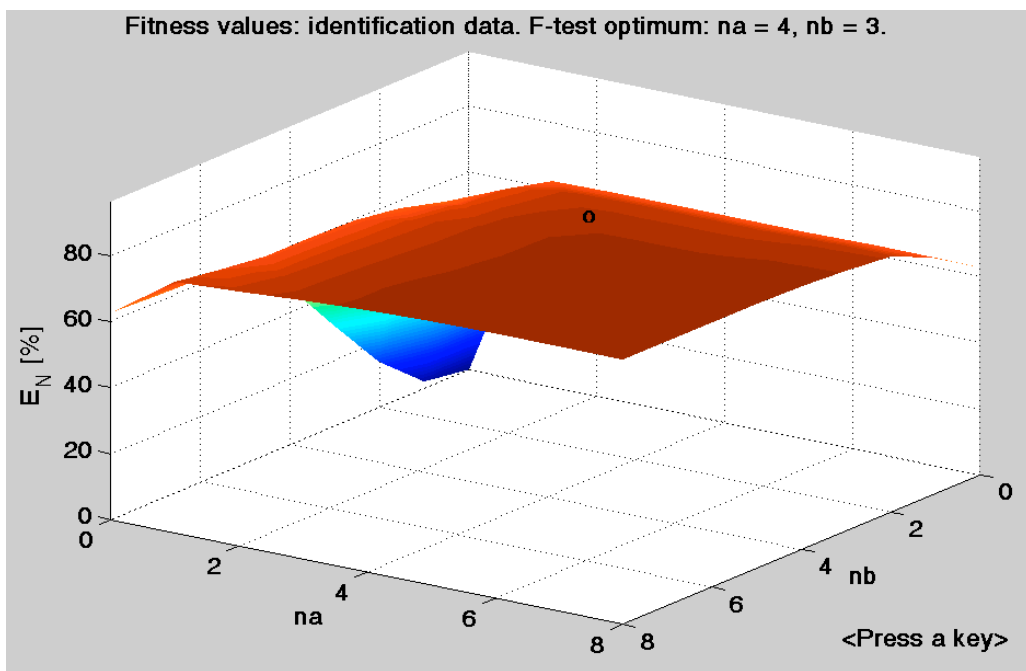


Figura 18. Potrivirea cu datele de identificare.

#### 4. Identificare parametrică prin MVI

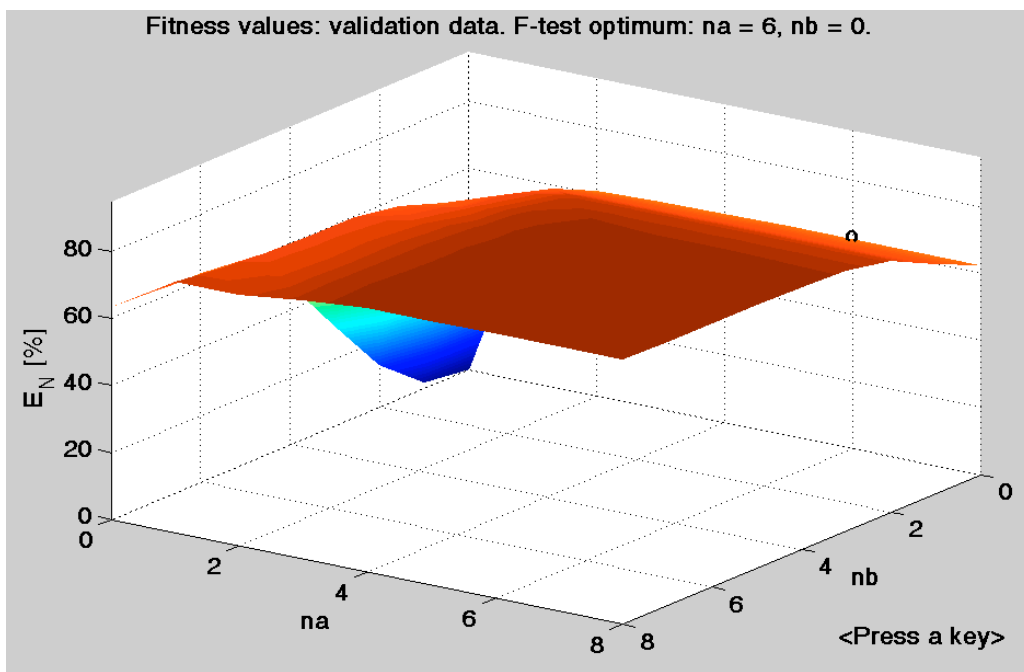


Figura 19. Potrivirea cu datele de validare.

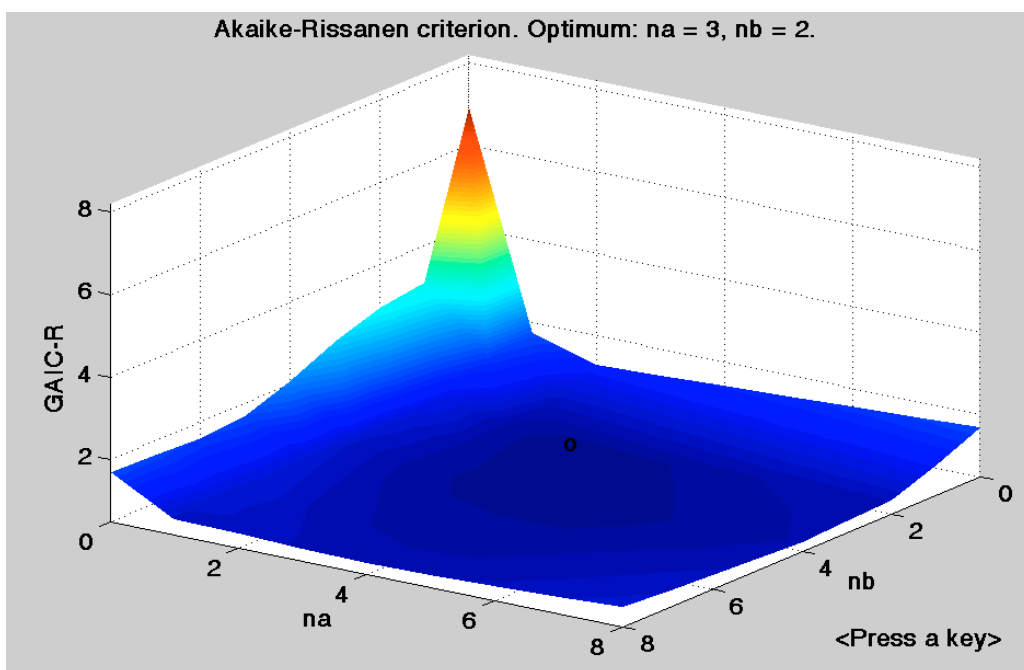


Figura 20. Criteriul Akaike-Rissanen.

## Capitolul 5

### Identificare parametrică prin Metoda Minimizării Erorii de Predicție

#### 5.1. Contextul general de lucru

Modelele de tip ARX, deși extrem de utilizate în aplicațiile de control automat prezintă dezavantajul că nu oferă posibilitatea reprezenta comportamentul zgomotelor perturbatoare. Pentru aceasta, cel mai frecvent se operează cu modele ARMAX generale sau de tip *Box-Jenkins* (BJ). Ecuația unui model ARMAX este descrisă în Introducere (definiția (1)), în timp ce modelul BJ constituie un caz particular al clasei generale de modele de identificare liniare (84), în care polinomul  $A$  este unitar (filtru de intrare independent de filtrul de zgomot):

$$\text{BJ: } y[n] = \frac{B(q^{-1})}{F(q^{-1})} u[n] + \frac{C(q^{-1})}{D(q^{-1})} e[n], \quad \forall n \in \mathbb{N}. \quad (86)$$

Două metode pot fi utilizate în principal pentru a estima parametrii modelelor ARMAX sau BJ: *Metoda celor Mai Mici Pătrate Extinsă* (MCMMP) și *Metoda Minimizării Erorii de Predicție* (MMEP, care include MCMMP în faza de inițializare). Acestea vor fi descrise succint în continuare.

#### A. Metoda Celor Mai Mici Pătrate Extinsă

Aplicarea MCMMP pentru estimarea unui model ARMAX/BJ nu este posibilă fără o adaptare corespunzătoare, deoarece vectorul regresorilor conține valori ale zgomotului (care nu pot fi măsurate) – a se vedea definiția (5) din Introducere. Adaptarea se bazează pe o strategie de identificare cu 2 etape: estimarea valorilor zgomotului alb folosind un model ARX și estimarea parametrilor modelului ARMAX/BJ. Metoda rezultată este chiar MCMMP.

##### Etapa 1. Estimarea valorilor zgomotului alb.

Datele de intrare-ieșire  $\mathcal{D} = \{u[n]\}_{n=1, \overline{N}} \cup \{y[n]\}_{n=1, \overline{N}}$  sunt utilizate pentru a identifica un model ARX de forma:

$$A_\alpha(q^{-1})y \equiv B_\beta(q^{-1})u + e, \quad (87)$$

unde polinoamele  $A_\alpha$  și  $B_\beta$  au grade suficient de mari (până la câteva zeci de coeficienți) și sunt obținute prin operația de împărțire infinită trunchiată:

$$\begin{aligned} \text{ARMAX: } A_\alpha(q^{-1}) &= \frac{A(q^{-1})}{C(q^{-1})} = 1 + \alpha_1 q^{-1} + \dots + \alpha_{n_\alpha} q^{-n_\alpha} \\ B_\beta(q^{-1}) &= \frac{B(q^{-1})}{C(q^{-1})} = 1 + \beta_1 q^{-1} + \dots + \beta_{n_\beta} q^{-n_\beta}, \end{aligned} \quad (88)$$

$$\begin{aligned} \text{BJ: } A_\alpha(q^{-1}) &= \frac{D(q^{-1})}{C(q^{-1})} = 1 + \alpha_1 q^{-1} + \dots + \alpha_{n\alpha} q^{-n\alpha} \\ B_\beta(q^{-1}) &= \frac{B(q^{-1})D(q^{-1})}{C(q^{-1})F(q^{-1})} = 1 + \beta_1 q^{-1} + \dots + \beta_{n\beta} q^{-n\beta} \end{aligned} \quad (89)$$

Vectorul regresorilor și vectorul parametrilor necunoscuți sunt, în acest caz:

$$\begin{cases} \psi^T[n] \stackrel{\text{def}}{=} [-y[n-1] \dots -y[n-n\alpha] \mid u[n-1] \dots u[n-n\beta]], \forall n \in \overline{1, N} \\ \theta_{\alpha\beta}^T \stackrel{\text{def}}{=} [\alpha_1 \dots \alpha_{n\alpha} \mid \beta_1 \dots \beta_{n\beta}] \end{cases} \quad (90)$$

Pentru estimarea parametrilor necunoscuți din datele măsurate, se poate folosi fie MCMMP, fie (mai indicat) MVI. Odată obținută, estimația  $\hat{\theta}_{\alpha\beta}$  poate fi utilizată pentru evaluarea valorilor zgomotului alb prin simularea modelului ARX identificat:

$$\varepsilon[n, \hat{\theta}_{\alpha\beta}] \stackrel{\text{def}}{=} y[n] - \psi^T[n] \hat{\theta}_{\alpha\beta}, \quad \forall n \in \overline{1, N}. \quad (91)$$

#### Etapă 2. Estimarea parametrilor modelului ARMAX/BJ.

Pentru modelul ARMAX, vectorul regresorilor  $\varphi[n]$  din ecuația (5) se înlocuiește cu un vector în care apar valorile estimate ale zgomotului alb:

$$\varphi_{\alpha\beta}[n] \stackrel{\text{def}}{=} [-y[n-1] \dots -y[n-na] \mid u[n-1] \dots u[n-nb] \mid \dots \mid \varepsilon[n-1, \hat{\theta}_{\alpha\beta}] \dots \varepsilon[n-nc, \hat{\theta}_{\alpha\beta}]], \quad \forall n \in \overline{1, N}. \quad (92)$$

Definiția (91) poate fi utilizată și pentru modelul BJ, cu modificările următoare:  $na = nd + nf$ ;  $nb \leftarrow nb + nd$ ;  $nc \leftarrow nc + nf$  (datorate exprimării modelului ca un model ARMAX, prin aducerea la același numitor). Aplicând MCMMP pentru noile notații, se obține o estimație a vectorului parametrilor necunoscuți și a dispersiei zgomotului alb:

$$\hat{\theta}_N \stackrel{\text{def}}{=} \left( \frac{1}{N} \sum_{n=1}^N \varphi_{\alpha\beta}[n] \varphi_{\alpha\beta}^T[n] \right)^{-1} \left( \frac{1}{N} \sum_{n=1}^N \varphi_{\alpha\beta}[n] y[n] \right); \quad (93)$$

$$\hat{\lambda}_N^2 \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^N (y[n] - \varphi_{\alpha\beta}^T[n] \hat{\theta}_N)^2. \quad (94)$$

În cazul modelului ARMAX, ecuațiile (93) și (94) rezolvă complet problema de identificare (vectorul parametrilor necunoscuți estimați  $\hat{\theta}_N$  are 3 componente, câte una pentru fiecare polinom al modelului, conform definiției (5)).

În cazul modelului BJ, parametrii estimați se obțin din vectorul  $\hat{\theta}_N$  după o serie de operații. Mai întâi, se determină rădăcinile polinoamelor corespunzătoare

celor 3 componente ale lui  $\hat{\theta}_N$ , notate, de exemplu, prin:  $A_{DF}$ ,  $B_{BD}$  și  $C_{CF}$  (după factorii care contribuie la formarea lor – vezi ecuația (86)). Rădăcinile comune (sau apropiate) ale polinoamelor  $A_{DF}$ ,  $B_{BD}$  sunt rădăcinile polinomului  $D$ , restul rădăcinilor lui  $B_{BD}$  aparținând polinomului  $B$ . Similar, rădăcinile comune (sau apropiate) ale polinoamelor  $A_{DF}$ ,  $C_{CF}$  sunt rădăcinile polinomului  $F$ , restul rădăcinilor lui  $C_{CF}$  aparținând polinomului  $C$ . Coeficienții celor 4 polinoame se evaluează apoi din rădăcinile lor.

Estimația oferită de MCMPE are totuși o precizie limitată, datorită mai multor surse de eroare, principalele fiind aproximarea modelului cu un model ARX în prima etapă și utilizarea valorilor estimate ale zgomotului în etapa a doua.

### B. Metoda Minimizării Erorii de Predicție

Erorile dintre model și proces (notate prin  $\{\varepsilon[n, \theta]\}_{n \in \overline{1, N}}$  pentru fiecare vector al parametrilor  $\theta$ ) constituie totodată și *erori de predicție* (cu un pas) a ieșirii procesului stocastic. Estimarea parametrilor se poate realiza atunci prin minimizarea următorului criteriu pătratic exprimat cu ajutorul erorilor de predicție pe orizontul de măsură:

$$\mathcal{J}_N(\theta) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^N \varepsilon^2[n, \theta], \quad \forall \theta \in \mathcal{S}, \quad (95)$$

unde  $\mathcal{S}$  este domeniul de stabilitate al modelului. Datorită acestui fapt, metoda de identificare se numește MMEP și estimația parametrilor oferită de ea rezultă prin rezolvarea unei probleme de optimizare:

$$\hat{\theta}_N = \arg \min_{\theta \in \mathcal{S}} \mathcal{J}_N(\theta). \quad (96)$$

Pentru a rezolva problema (96), se folosește o metodă mai generală de optimizare cu criterii pătratice: *Metoda Gauss-Newton* ([MGN](#)). Potrivit acestei metode, optimul este evaluat în mod recursiv cu precizie din ce în ce mai mare, după următoarea relație:

$$\hat{\theta}_N[k+1] = \hat{\theta}_N[k] - R_N^{-1}[k] r_N[k], \quad \forall k \geq 0, \quad (97)$$

unde:

$$\begin{cases} R_N[k] \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^N \nabla_{\theta} \varepsilon[n, \hat{\theta}_N[k]] (\nabla_{\theta} \varepsilon[n, \hat{\theta}_N[k]])^T \\ r_N[k] \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^N \varepsilon[n, \hat{\theta}_N[k]] \nabla_{\theta} \varepsilon[n, \hat{\theta}_N[k]] \end{cases}, \quad \forall k \geq 0. \quad (98)$$

Calculul efectiv al corecției  $R_{N,k}^{-1} r_{N,k}$  din ecuația (97) necesită evaluarea erorii curente de predicție  $\varepsilon[n, \hat{\theta}_{N,k}]$  și a gradientului acesteia  $\nabla_{\theta} \varepsilon[n, \hat{\theta}_{N,k}]$  pentru fiecare moment  $n \in \overline{1, N}$ . Ambele se calculează iterativ, folosind ecuațiile modelului matematic ales.

a. Modelul ARMAX:

$$\begin{aligned} \varepsilon[n, \hat{\theta}_N[k]] = & y[n] + \hat{a}_{1,k}y[n-1] + \dots + \hat{a}_{na,k}y[n-na] - \\ & - \hat{b}_{1,k}u[n-1] - \dots - \hat{b}_{nb,k}u[n-nb] - \\ & - \hat{c}_{1,k}\varepsilon[n-1, \hat{\theta}_N[k]] - \dots - \hat{c}_{nc,k}\varepsilon[n-nc, \hat{\theta}_N[k]], \quad \forall k \geq 0, \end{aligned} \quad (99)$$

$$\begin{cases} \nabla_{\theta_a} \varepsilon[n, \hat{\theta}_N[k]] = -\varphi_y[n] - \\ \quad - \hat{c}_{1,k} \nabla_{\theta_a} \varepsilon[n-1, \hat{\theta}_N[k]] - \dots - \hat{c}_{nc,k} \nabla_{\theta_a} \varepsilon[n-nc, \hat{\theta}_N[k]] \\ \nabla_{\theta_b} \varepsilon[n, \hat{\theta}_N[k]] = -\varphi_u[n] - \\ \quad - \hat{c}_{1,k} \nabla_{\theta_b} \varepsilon[n-1, \hat{\theta}_N[k]] - \dots - \hat{c}_{nc,k} \nabla_{\theta_b} \varepsilon[n-nc, \hat{\theta}_N[k]] \\ \nabla_{\theta_c} \varepsilon[n, \hat{\theta}_N[k]] = -\varphi_\varepsilon[n, \hat{\theta}_N[k]] \end{cases} \quad \forall k \geq 0, \quad (100)$$

unde:

$$\theta \stackrel{def}{=} \begin{bmatrix} \theta_a \\ \theta_b \\ \theta_c \end{bmatrix} = \begin{bmatrix} [a_i]_{i=1,na} \\ [b_j]_{j=1,nb} \\ [c_l]_{l=1,nc} \end{bmatrix} \quad \text{și} \quad \varphi[n, \theta] \stackrel{def}{=} \begin{bmatrix} \varphi_y[n] \\ \varphi_u[n] \\ \varphi_\varepsilon[n] \end{bmatrix} = \begin{bmatrix} [-y[n-i]]_{i=1,na} \\ [u[n-j]]_{j=1,nb} \\ [\varepsilon[n-1, \theta]]_{l=1,nc} \end{bmatrix}. \quad (101)$$

Ecuțiile (90) și (100) se inițializează de regulă cu valori nule.

b. Modelul BJ:

În acest caz, vectorul  $\hat{\theta}_{N,k}$  are aceleași 3 componente ca în definiția lui  $\theta$  din (101). Ca urmare, mai întâi se evaluează coeficienții polinoamelor  $\hat{A}_{DF,k} \equiv \hat{D}_k \hat{F}_k$  (de grad  $nd + nf$ ),  $\hat{B}_{BD,k} \equiv \hat{B}_k \hat{D}_k$  (de grad  $nb + nd$ ) și  $\hat{C}_{CF,k} \equiv \hat{C}_k \hat{F}_k$  (de grad  $nc + nf$ ). Se folosesc apoi ecuațiile iterative (90) și (100) pentru polinoamele anterioare (ecuații inițializate de regulă cu valori nule). După aplicarea corecției, parametrii lui  $\hat{\theta}_N[k+1]$  sunt folosiți pentru a produce valorile curente ale parametrilor modelului ca în Etapa 2 a MCMPE (adică prin identificarea zerourilor comune). Se observă că, de fapt, modelul BJ se poate determina ca și modelul ARMAX în cursul iterațiilor (adică folosind doar 3 polinoame în loc de 4), urmînd ca toate cele 4 polinoame să fie explicitate doar în final prin tehnica identificării zerourilor comune.

În mod evident, odată ce estimația  $\hat{\theta}_N[k]$  a fost obținută, dispersia estimată a zgomotului alb este:

$$\hat{\lambda}_N^2[k] \stackrel{def}{=} \frac{1}{N} \sum_{n=1}^N (y[n] - \varphi^T[n, \hat{\theta}_N[k]] \hat{\theta}_N[k])^2 = \frac{1}{N} \sum_{n=1}^N \varepsilon^2[n, \hat{\theta}_N[k]] = \mathcal{V}_N(\hat{\theta}_N[k]). \quad (102)$$

Testul de stop al iterațiilor (97) este unul din următoarele:



$$\begin{aligned} & \left\| \hat{\theta}_N[k+1] - \hat{\theta}_N[k] \right\| < \eta \quad \text{sau} \quad \max_{i \in 1, n\theta} \left| \hat{\theta}_{N,i}[k+1] - \hat{\theta}_{N,i}[k] \right| < \eta \\ & \text{sau} \quad \left\| \mathbf{v}_N(\hat{\theta}_N[k+1]) - \mathbf{v}_N(\hat{\theta}_N[k]) \right\| = \left\| \hat{\lambda}_N^2[k+1] - \hat{\lambda}_N^2[k] \right\| < \eta \\ & \text{sau} \quad k+1 \leq K_{\max}, \end{aligned} \quad (103)$$

unde  $\eta > 0$  (numit și *toleranță*) controlează precizia modelului,  $n\theta$  este lungimea vectorului  $\theta$ , indicele  $i$  denotă componenta  $i$  a vectorului, iar  $K_{\max}$  este numărul maxim de iterații impus (inițial,  $k=0$ ). Dacă inegalitatea aleasă din (103) este verificată,  $\hat{\theta}_N[k+1]$  este considerată estimăția “optimă”  $\hat{\theta}_N$ .

Inițializarea calculului iterativ (97) se poate efectua plecând de la un model determinat cu ajutorul MCMPE, eventual mai puțin precis.

Se poate arăta că dacă modelul ales este parsimonios (adică polinoamele sunt coprime între ele – nu se mai poate simplifica nici o rădăcină comună) și intrarea  $u$  este un semnal persistent suficient de mare (cel puțin egal cu numărul parametrilor necunoscuți), atunci estimăția oferită de MMEP este consistentă.

Obiectivul acestui capitol este de a studia comparativ performanțele MCMPE și MMEP folosind modele ARMAX și BJ de mai jos:

$$\underbrace{(1 - 1.5q^{-1} + 0.7q^{-2})}_{A(q^{-1})} y[n] = \underbrace{(q^{-1} + 0.5q^{-2})}_{B(q^{-1})} u[n] + \underbrace{(1 - q^{-1} + 0.2q^{-2})}_{C(q^{-1})} e[n], \quad \forall n \in \mathbb{N}^*; \quad (104)$$

$$y[n] = \underbrace{\frac{q^{-1} + 0.5q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}}}_{B(q^{-1})/F(q^{-1})} u[n] + \underbrace{\frac{1 - q^{-1} + 0.2q^{-2}}{1 + 1.5q^{-1} + 0.7q^{-2}}}_{C(q^{-1})/D(q^{-1})} e[n], \quad \forall n \in \mathbb{N}^*. \quad (105)$$

## 5.2. Exerciții

### Exercițiul 5.1

Exprimați primele 4 iterații ( $n \in \overline{1,4}$ ) și ultima iterație ( $n = N \gg n\theta$ ) în evaluarea erorii de predicție pentru un model ARMAX în general. Particularizare în cazul modelelor ARMAX (104) și BJ (105).

### Exercițiul 5.2

Exprimați primele 4 iterații ( $n \in \overline{1,4}$ ) și ultima iterație ( $n = N \gg n\theta$ ) în evaluarea gradientului erorii de predicție pentru un model ARMAX în general. Particularizare în cazul modelelor ARMAX (104) și BJ (105).

### Exercițiul 5.3

Descrieți algoritmul implicat de MCMPE în cazul unui model ARMA.

**Exercițiul 5.4**

Descrieți algoritmul implicat de MMEP în cazul unui model ARMA.

**5.3. Probleme de simulare**

Pentru problemele de simulare care urmează, se va urmări strategia adoptată în cadrul simulărilor efectuate în capitolul precedent. Modelele de bază ale proceselor furnizoare de date sunt (104) și (105). Se recomandă proiectarea unei rutine de generare a datelor mai generală decât `gendata` din cadrul capitolului precedent. Dacă se denumește această rutină prin `gen_data`, apelul tipic al acesteia ar trebui să fie următorul:

```
[D,V,P] = gen_data(DP,N,sigma,lambda,bin) ;
```

unde: **DP** este obiectul de tip **IDMODEL** (vezi **Problema 3.3**) corespunzător modelului de proces furnizor de date; obiectul poate fi construit de exemplu cu ajutorul funcției `idpoly`, care a fost descrisă în secțiunea 4.3; implicit, acest model este identic cu cel din definiția (104) (ARMAX);

**N** este dimensiunea orizontului de măsură (implicit: **N=250**);

**sigma** este deviația standard a intrării SPAB (implicit: **sigma=1**);

**lambda** este deviația standard a zgomotului alb Gaussian (implicit: **lambda=1**);

**bin** este un parametru care arată tipul de intrări dorit: **bin=0** (intrare SPAB Gaussiană); **bin~=0** (implicit, intrare SPAB Gaussiană bipolară);

**D** este obiectul de tip **IDDATA** (vezi **Problema 2.3**) corespunzător datelor generate (intrarea se regăsește în **D.u**, iar ieșirea în **D.y**);

**V** este obiectul de tip **IDDATA** corespunzător zgomotelor generate (zgomotul alb se regăsește în **V.u**, iar zgomotul colorat (adică MA-filtrat) în **V.y**).

**P** este obiectul de tip **IDMODEL** corespunzător modelului de proces furnizor de date.

Biblioteca de rutine dedicate IS conține 2 rutine referitoare la ansamblul de metode MCMPE-MMEP, după modelul de identificare ales: **armax** (pentru modele ARMAX) și **bj** (pentru modele BJ).

**# ARMAX**

- Apel: **Mid = armax(D,si) ;**
- Estimează parametrii unui model ARMAX folosind MMEP. Modelul identificat rezultat, **Mid**, este returnat ca obiect **IDMODEL**. Estimarea se efectuează pe baza datelor **D** (obiect **IDDATA**) și a informației de structură **si = [na nb nc nk]**, unde **na**, **nb** și **nc** sunt indicii structurali ai modelului, iar **nk** este întârzierea intrinsecă. Cu ajutorul acestei rutine se pot identifica atât modele AR cât și modele ARMA unidimensionale (însă nu și multi-dimensionale). Apelul rutinei este ușor diferit în acest caz:
  - pentru modele AR: **Mid = armax(D.y,na) ;**
  - pentru modele ARMA: **Mid = armax(D.y,[na nc]) ;**

## Aspecte practice în Modelarea și Identificarea Sistemelor

Observați că datele de identificare sunt specificate acum doar sub forma unei *serii de timp* (**D.y**). Pentru identificarea modelelor AR, rutina apelează intern o funcție specializată numită **ar**, care este disponibilă și utilizatorului (cu apel similar lui **armax**).

O altă modalitate de a identifica modele AR și ARMA este de a folosi obiectul **D** împreună cu o informație de structură de forma: **si = [na 0 0 0]** (AR) sau **si = [na 0 nc 0]** (ARMA). Rutina nu funcționează însă decât dacă **na>1** (nu și pentru **na=1**). De aceea, se recomandă utilizarea rutinei cu argument serie de timp, pentru aceste modele.

# **BJ**

- Apel: **Mid = bj(D,si) ;**
- Estimează parametrii unui model BJ folosind MMEP. Modelul identificat rezultat, **Mid**, este returnat ca obiect **IDMODEL**. Estimarea se efectuează pe baza datelor **D** (obiect **IDDATA**) și a informației de structură **si = [na nb nc nd nf nk]**, unde **na**, **nb**, **nc**, **nd** și **nf** sunt indicii structurali ai modelului, iar **nk** este întârzierea intrinsecă. În principiu, algoritmul implementat în cadrul acestei rutine este similar cu cel al rutinei **armax**, cu adaptările de rigoare impuse de utilizarea modelului BJ.

Metoda implementată în cadrul acestor rutine este MMEP (dar cu inițializare oferită de MCMMPPE). Ambele rutine permit și o serie de specificații mai tehnice privind performanțele dorite ale algoritmilor implementați, cum ar fi: setarea toleranței de precizie, specificarea unui număr maxim de iterații, precizarea unei direcții preferențiale de căutare a optimului, etc.

Alegerea structurii în cazul metodelor MCMMPPE-MMEP ridică, în general, probleme de complexitate. În cadrul simulatoarelor, se vor utiliza criteriul applatizării, Testul F, funcția de potrivire și criteriul GAIC-Rissanen, ca în cazul simulărilor din Capitolul 4. Testul F depinde acum de cel puțin 3 indici structurali, astfel că evaluarea sa ar trebui efectuată în același ciclu în care a fost determinat modelul curent.

Validarea modelelor se bazează pe aceleași criterii descrise în secțiunea 4.1 (paragraful C). Indicii structurali maximi sunt:  $N_a = N_b = N_c = 5$ .

### Problema 5.1

Biblioteca MATLAB dedicată domeniului IS nu dispune de funcții explicite care implementează MCMMPPE. Să se proiecteze două astfel de funcții: **armax\_e** pentru identificarea modelelor ARMAX și **bj\_e** pentru identificarea modelelor BJ. Apelul tipic al lor ar trebui să fie similar altor funcții cu obiectiv asemănător (estimarea parametrilor unui model cu structură dată; vezi de exemplu funcțiile **armax** și **bj**):

```
Mid = armax_e(D,si) ;  
Mid = bj_e(D,si) ;
```

Informația de structură are forma: **si = [na nb nc nk]** pentru modelul ARMAX și **si = [na nb nc nd nf nk]** pentru modelul BJ. Încercați să folosiți funcția **armax\_e** în cadrul funcției **bj\_e**. Testați cele 2 rutine în cazul modelelor (104) și (105) pentru câteva seturi de indici structurali (inclusiv cei adevărați). Comentați precizia de estimare a parametrilor.

**Problema 5.2**

Să se proiecteze mini-simulatorul **ISLAB\_5A** care evaluează estimăția (parsimonioasă a) modelului ARMAX asociat procesului (104), folosind MMEP. Pentru aceasta, se vor parcurge următorii pași:

- a. Se generează 2 seturi de date: unul pentru identificare și altul pentru validare, folosind rutina **gen\_data**.
- b. Pentru fiecare model identificat cu ajutorul MMEP (funcția **armax**), model obținut variind indicii  $na$ ,  $nb$  și  $nc$ , se vor afișa 3 ferestre grafice: una pentru analiza modelului folosind datele de identificare și de validare și alte două pentru reprezentările poli-zero-uri (filtru sistem și filtru zgomot) cu discuri de încredere corespunzătoare unei raze de 3 ori mai mari decât deviațiile standard aferente (ca în **Figurile 21 și 22**). După fiecare fereastră se inserează o pauză de așteptare pentru a permite utilizatorului să analizeze informațiile afișate. Fiecare sub-fereastră a primei ferestre include 3 grafice aranjate pe verticală:
  - ieșirile măsurate și simulate cu ajutorul modelului, grafic pe care se indică și valoarea funcției de potrivire,  $\mathcal{E}_N$ ;
  - eroarea de predicție (reziduurile modelului), grafic pe care se indică și dispersia estimată a zgomotului,  $\hat{\sigma}_N^2$ ;
  - secvența de auto-covarianță a erorii de predicție, grafic pe care se indică și indexul de validare.

Modelele obținute vor fi memorate în vederea selectării unuia dintre ele, în urma aplicării testelor de determinare a indicilor structurali optimi și de validare.

- c. Se afișează indicii structurali optimi selectați folosind:
  - Testul F aplicat dispersiei estimate a zgomotului (adică erorii de predicție);
  - Testul F aplicat funcției de potrivire pentru datele de identificare;
  - Testul F aplicat funcției de potrivire pentru datele de validare;
  - criteriului GAIC în versiunea Rissanen.
- d. Se solicită utilizatorului să aleagă indicii structurali pe care îi consideră optimi.
- e. Pentru modelul ales, se afișează cele 3 ferestre grafice de la b. Modelul este returnat de către mini-simulator, în vederea unei utilizări ulterioare. Se recomandă returnarea și a seturilor de date de identificare și validare.

După proiectarea mini-simulatorului **ISLAB\_5A**, se vor iniția câteva rulări.

Sunt indicii structurali adevărați indicați de către majoritatea criteriilor utilizate sau ei diferă de la o rulare la alta? Justificați răspunsul.

**Problema 5.3**

Problema anterioară, **5.3**, se va relua pentru modelul BJ (105). Mini-simulatorul rezultat va fi denumit **ISLAB\_5B**. Testați mini-simulatorul și comentați rezultatele de estimare obținute.

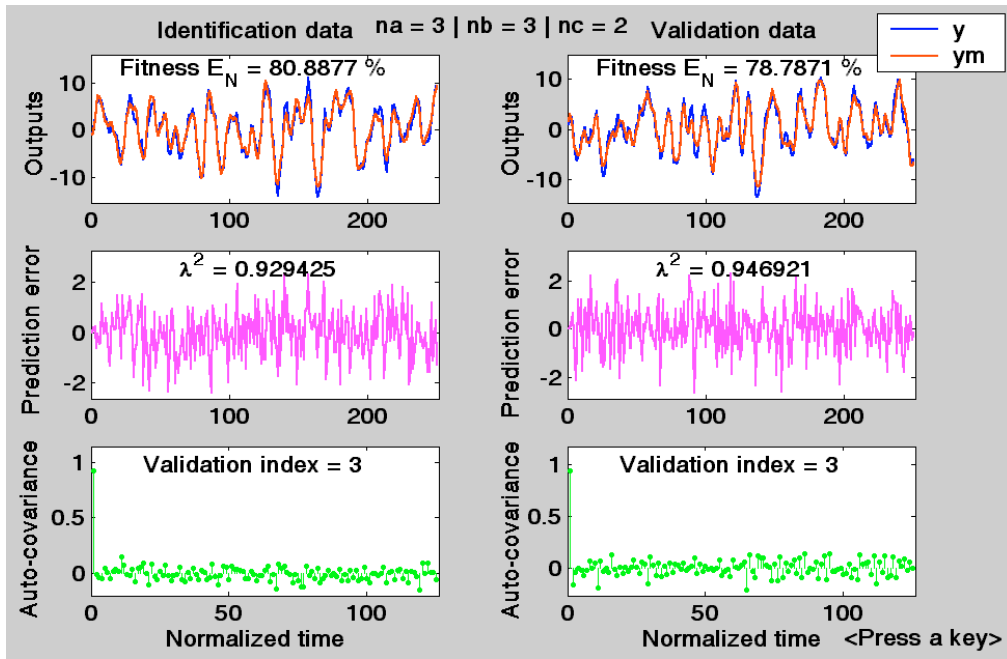


Figura 21. Performanțele unui model estimat cu MMEP.

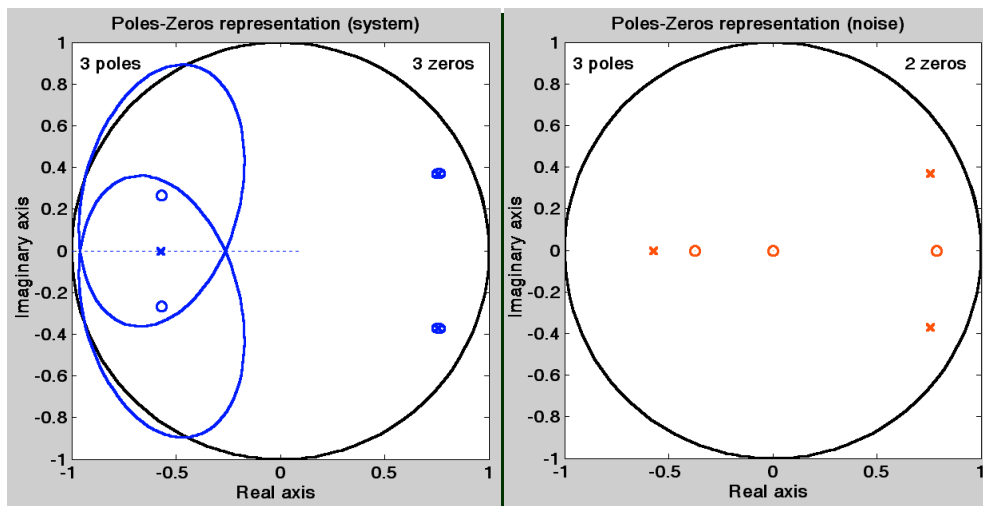


Figura 22. Reprezentarea poli-zeroruri a unui model estimat cu MMEP.

## Capitolul 6

### Identificare recursivă

#### 6.1. Contextul general de lucru

##### A. Algoritmi recursivi de identificare

Ipoteza parametrilor constanți ai unui model (menținuți la aceleași valori pe întreaga durată a experimentului de identificare) se dovedește nerealistă în multe cazuri de procese concrete analizate. Modelele intrare-ieșire ale unui mare număr de procese ar trebui să evolueze la rîndul lor în timp. În multe situații, este necesară estimarea și reactualizarea parametrilor unui model simultan cu achiziția de date. Modelul poate fi apoi utilizat pentru luarea unor decizii în timp real, ca în cazul aplicațiilor de control adaptiv, filtrare adaptivă, detecție de defecte sau predicție adaptivă. De aceea, terminologia IS a fost îmbogățită cu noi concepte ca: *identificare recursivă*, *estimare adaptivă de parametri*, *estimare secvențială* sau *algoritm de identificare on-line*. Aceste concepte sunt pe larg descrise în [LjL99] și [SoSt89].

Identificarea recursivă a parametrilor trebuie să ia în considerare variația lor în timp, pentru a asigura nu numai precizia modelului, ci și capacitatea de urmărire a acestor variații. Cele 2 proprietăți sunt de fapt contradictorii: o precizie excesivă implică rigiditate în urmărirea parametrilor, în timp ce o flexibilitate prea mare în urmărirea parametrilor atrage după sine imprecizia de estimare a lor. În general, compromisul dintre aceste două proprietăți este controlat prin mărimea perioadei de reactualizare, pe a cărei durată parametrul sunt presupuși constanți. În contextul descris mai jos, reactualizarea parametrilor se efectuează după fiecare perioadă de eșantionare, pentru a favoriza capacitatea de urmărire. De notat totuși că precizia modelului poate fi mai mult decît satisfăcătoare chiar și în acest caz, dacă este utilizată o metodă de identificare adecvată modelului. De exemplu, MCMMP în variantă recursivă ([MCMMP-R](#)) va fi mai puțin precisă decît MVI în variantă recursivă ([MVI-R](#)) în cazul modelului ARX.

Revenim la modelul ARMAX (1), care poate fi exprimat echivalent sub formă de ecuație de regresie liniară (4). Metodele recursive pleacă de fapt de la ecuația (4) și pot fi utilizate pentru orice model care poate fi exprimat echivalent în acest mod. Pentru estimarea și reactualizarea parametrilor la fiecare pas de eșantionare, se poate folosi orice variantă a algoritmului descris în [Figura 23](#). Semnalul instrumental specificat în datele de intrare permite utilizatorului să particularizeze algoritmul într-o procedură de tip MVI-R. În afara procedurilor MCMMP-R și MVI-R, utilizatorul poate de asemenea să particularizeze acest algoritm într-o procedură sugerată de MMEP în variantă recursivă ([MMEP-R](#)).

O versiune a MMEP-R este de asemenea utilizată în aplicații: *Metoda de Regresie Pseudo-Liniară Recursivă* ([MRPL-R](#)). Așa cum am amintit în capitolul precedent, MMEP apelează la *Metoda Gauss-Newton* pentru reactualizarea recursivă a parametrilor estimați. Aceasta se bazează pe o anumită manieră de aproximare (fie prin liniarizarea reziduurilor (adică a erorii de predicție), fie prin aproximarea matricii Hessian corespunzătoare erorii de predicție).

Figura 23. Algoritmul recursiv de bază în IS.

- Date de intrare:
- ordinele modelului de identificare:  $na, nb, nc, nd$  și  $nf$ ;
  - o colecție redusă de date intrare-ieșire măsurate (dacă este posibil):  
 $\mathcal{D}_{N_0} = \{u[n]\}_{n \in \overline{1, N_0}} \cup \{y[n]\}_{n \in \overline{1, N_0}}$  (cu  $N_0$  de ordinul zecilor cel mult);
  - un semnal instrumental extern:  $\{f[n]\}_{n \in \overline{1, N}}$  (eventual).
- Centrarea datelor pe medie:  $y \leftarrow y - \bar{y}, u \leftarrow u - \bar{u}$  (și  $f \leftarrow f - \bar{f}$ , dacă a fost specificat).
  - Dacă nu a fost specificat nici un semnal instrumental, vectorul variabilelor instrumentale,  $z$ , este identic cu vectorul regresorilor  $\varphi$ . Altfel,  $z$  este definit ca în (58) sau (59)-(60), dar folosind în general semnalul instrumental extern  $f$  în locul intrării  $u$  (în particular, este posibil ca  $f \equiv u$ ).
  - Inițializare. Fie se setează arbitrar vectorul parametrilor  $\hat{\theta}_0$  și matricea  $P_0 = \alpha I_{n_\theta}$  (cu  $\alpha \in \mathbb{R}_+^*$ ) (în cazul în care nu se dispune de setul de date redus  $\mathcal{D}_{N_0}$ ), fie se estimează valoarea inițială a parametrilor ( $\hat{\theta}_0$ ) folosind o metodă off-line adecvată modelului particular utilizat (din clasa MCMMP-MVI) și se egalează matricea  $P_0$  cu inversa matricii de covarianță  $R_0$  folosită în calculul lui  $\hat{\theta}_0$  (în cazul în care setul de date redus  $\mathcal{D}_{N_0}$  este disponibil).
  - Pentru  $k \geq 1$ :
    - Se evaluează eroarea de predicție curentă:  $\varepsilon[k] = y[k] - \varphi^T[k] \hat{\theta}_{k-1}$ .
    - Se evaluează vectorul auxiliar:  $\xi_k = P_{k-1} z[k]$ .
    - Se evaluează câștigul de sensibilitate:  $\gamma_k = \frac{\xi_k}{1 + \varphi^T[k] \xi_k}$ .
    - Se reactualizează inversa matricii  $R_k$ , adică:  $P_k = P_{k-1} - \gamma_k \varphi^T[k] P_{k-1}$  (cu evitarea inversării explicite a matricilor).
    - Se reactualizează vectorul parametrilor:  $\hat{\theta}_k = \hat{\theta}_{k-1} + \gamma_k \varepsilon[k]$ .
- Date de ieșire: parametrii modelului ( $\hat{\theta}_k$ ) la fiecare pas de reactualizare  $k \geq 0$ .

Metoda Gauss-Newton este înrudită cu familia de metode de optimizare de tip *Newton-Raphson* (din care fac parte și metodele de gradient). În cazul MRPL-R, este utilizată o metodă de optimizare din această clasă [LjL99], [SoSt89].

Evitarea inversării matricii  $R_k$  a fost posibilă grație unei leme de inversiune din Teoria Matricilor:

$$(A + bc^T)^{-1} = A^{-1} - \frac{A^{-1}c^T b A^{-1}}{1 + c^T A^{-1}b}, \quad (106)$$

unde  $A$  este o matrice inversabilă, iar  $b$  și  $c$  sunt vectori de lungimi corespunzătoare dimensiunii matricii  $A$ .

O variantă generalizată a algoritmului de identificare recursiv anterior este obținută prin aplicarea principiului ponderării datelor. În general, pentru a reactualiza parametrii estimați, nu este indicat să se țină cont în egală măsură de toată istoria evoluției procesului pînă la momentul curent, deoarece datele de intrare-ieșire foarte vechi reprezintă comportamente neactuale, eventual eronate pentru momentul curent. De aceea, erorile de predicție sunt ponderate cu ajutorul unei ferestre care aplică penalități datelor situate în trecut. Fereastra este de regulă dreptunghiulară (cu lungime mai mică decît orizontul de măsură) sau exponențială. Fie  $w_k$  fereastra aplicată la momentul curent al achiziției de date  $k \geq 1$ . Atunci criteriul de optimizare din ecuațiile (21) devine:

$$J_k(\theta) \stackrel{\text{def}}{=} \sum_{n=1}^k w_k[n](y[n] - \varphi[n]\theta)^2, \quad \forall \theta \in \mathcal{S}, \quad (107)$$

unde  $\mathcal{S}$  este domeniul de stabilitate al modelului. Procedura rezultată prin minimizarea criteriului (107) depinde intim de tipul de fereastră utilizat. Vom descrie cei 2 algoritmi rezultați prin utilizarea ferestrelor dreptunghiulară și exponențială.

#### ➤ Fereastra dreptunghiulară

Fereastra dreptunghiulară are lungimea  $M$ , cu  $M < N$  (unde  $N$  este lungimea orizontului de măsură). Considerăm, că momentul curent al achiziției de date este  $k$ . Pentru simplitate și eficacitate, algoritmul recursiv descris mai jos poate fi rulat începînd cu  $k = M + 1$ . Cît timp  $k < M$ , se efectuează doar achiziția datelor, fără estimarea parametrilor. Cînd  $k = M$ , un algoritm de identificare de tip off-line este utilizat pentru estimarea parametrilor inițiali.

Fereastra, rennotată prin  $w_{M,k}$ , aplică o penalizare dură vechilor date prin uitarea totală a lor. Mai exact, erorile de predicție evaluate pentru momente de timp inferioare lui  $k - M + 1$  sunt complet înlăturate și nu mai participă la reactualizarea parametrilor curenți. Numai datele achiziționate între momentele  $k - M + 1$  și  $k$  sunt considerate (multiplicate de ponderi unitare). Vectorii  $\varphi$  și  $z$  din cadrul algoritmului de bază din Figura 23 își păstrează totuși definițiile originale, independent de tipul de fereastră considerat. Fereastra dreptunghiulară afectează matricea  $P_k^{-1}$ , exprimată recursiv astfel:

$$P_k^{-1} \stackrel{\text{def}}{=} \sum_{n=k-M+1}^k z[n]\varphi^T[n] = P_{k-1}^{-1} - z[k-M]\varphi^T[k-M] + z[k]\varphi^T[k]. \quad (108)$$

Pentru a inversa matricea (108), lema de inversiune (106) trebuie aplicată de 2 ori succesiv. În consecință, se obține algoritmul descris în Figura 24.

Spre deosebire de algoritmul de bază din Figura 23, în acest caz, fereastra dreptunghiulară necesită calcularea a 2 erori de predicție, cîte una pentru fiecare latură a ferestrei. Complexitatea acestui algoritm este mai ridicată decît în cazul precedent, dar atît precizia cît și capacitatea de urmărire sunt îmbunătățite.



Figura 24. Algoritmul recursiv cu fereastră dreptunghiulară.

➤ Date de intrare:

a. ordinele modelului de identificare:  $na, nb, nc, nd$  și  $nf$  ;

b. lungimea ferestrei dreptunghiulare:  $M$  ;

c. o colecție redusă de date intrare-ieșire măsurate:

$$\mathcal{D}_M = \{u[n]\}_{n \in \overline{1, M}} \cup \{y[n]\}_{n \in \overline{1, M}} ;$$

d. un semnal instrumental extern:  $\{f[n]\}_{n \in \overline{1, N}}$  (eventual).

1. Centrarea datelor pe medie:  $y \leftarrow y - \bar{y}, u \leftarrow u - \bar{u}$  (și  $f \leftarrow f - \bar{f}$ , dacă a fost specificat).

2. Dacă nu a fost specificat nici un semnal instrumental, vectorul variabilelor instrumentale,  $z$ , este identic cu vectorul regresorilor  $\varphi$ . Altfel,  $z$  este definit ca în (58) sau (59)-(60), dar folosind în general semnalul instrumental extern  $f$  în locul intrării  $u$  (în particular, este posibil ca  $f \equiv u$ ).

3. Inițializare. Se estimează valoarea inițială a parametrilor ( $\hat{\theta}_0$ ) folosind o metodă off-line adecvată modelului particular utilizat (din clasa MCMMP-MVI, cu datele  $\mathcal{D}_M$ ) și se egalează matricea  $P_0$  cu inversa matricii de covarianță  $R_0$  folosită în calculul lui  $\hat{\theta}_0$ .

4. Pentru  $k \geq 1$ :

4.1. Se evaluează eroarea de predicție la dreapta:  $\varepsilon_d[k] = y[k] - \varphi^T[k] \hat{\theta}_{k-1}$ .

4.2. Se evaluează eroarea de predicție la stînga:

$$\varepsilon_s[k-M] = y[k-M] - \varphi^T[k-M] \hat{\theta}_{k-1}.$$

4.3. Se reactualizează matricea  $P_k$  în 2 pași:

$$\begin{aligned} \bullet \xi_k &= P_{k-1} z[k] \text{ și } P_{k-1} \leftarrow P_{k-1} - \frac{\xi_k \varphi^T[k] P_{k-1}}{1 + \varphi^T[k] \xi_k}; \\ \bullet \xi_k &= P_{k-1} z[k-M] \text{ și } P_k = P_{k-1} + \frac{\xi_k \varphi^T[k-M] P_{k-1}}{1 - \varphi^T[k-M] \xi_k}. \end{aligned}$$

4.4. Se reactualizează vectorul parametrilor:

$$\hat{\theta}_k = \hat{\theta}_{k-1} + P_k (z[k] \varepsilon_d[k] - z[k-M] \varepsilon_s[k-M]).$$

➤ Date de ieșire: parametrii modelului ( $\hat{\theta}_k$ ) la fiecare pas de reactualizare  $k \geq 0$ .

➤ **Fereastra exponențială**

Înlăturarea bruscă a erorilor de predicție produse de datele mai vechi poate provoca erori marginale, în special în cazul proceselor cu dinamică rapidă și bogat conținut în frecvențe. Pentru astfel de sisteme (și nu numai), anihilarea contribuției datelor mai vechi poate fi realizată prin intermediul unei ferestre exponențiale, care aplică o penalizare treptată. Variația exponențială este simulată cu ajutorul unui parametru controlabil notat cu  $\lambda \in (0,1]$  și denumit

*factor de uitare*. La momentul curent de achiziție a datelor  $k \geq 1$ , fereastra exponențială are următoarea exprimare:

$$w_k[n] \stackrel{\text{def}}{=} \lambda^{k-n}, \quad \forall n \in \overline{1, k}. \quad (109)$$

Potrivit definiției (109), criteriul pătratic (107) devine:

$$\mathcal{J}_k(\theta) = \sum_{n=1}^k \lambda^{k-n} (y[n] - \varphi[n]\theta)^2, \quad \forall \theta \in \mathcal{S}. \quad (110)$$

Aceasta implică următoarea relație recursivă verificată de matricea  $P_k^{-1}$ :

$$P_k^{-1} \stackrel{\text{def}}{=} \sum_{n=1}^k \lambda^{k-n} z[n] \varphi^T[n] = \lambda P_{k-1}^{-1} + z[k] \varphi^T[k]. \quad (111)$$

Lema de inversiune (106) poate fi acum aplicată direct asupra relației (111). În consecință, algoritmul recursiv corespunzător ferestrei exponențiale este cel descris în [Figura 25](#).

Evident, în acest caz, utilizatorul are posibilitatea de a controla mai fin procesul de penalizare a datelor învechite, prin stabilirea factorului de uitare dorit (de regulă între 0.95 și 0.995). Factorii de uitare de valori din ce în ce mai mici conduc la o atenuare din ce în ce mai severă a erorilor produse de datele vechi. Se poate observa cu ușurință diferențele dintre pașii 4.3, respectiv 4.4 ai algoritmilor din [Figurile 23 și 25](#). Dacă factorul de uitare este stabilit la valoarea unitară, se obține chiar algoritmul de bază din [Figura 23](#).

O altă semnificație practică a factorului de uitare este dată de următoarea interpretare: măsurători mai vechi de  $T_\lambda = \lambda/(1-\lambda)$  față de momentul curent sunt incluse în criteriul pătratic cu o pondere de aproximativ 36% din ponderea măsurătorilor celor mai recente ( $T_\lambda$  se numește *constantă de timp* a memoriei datelor sau *orizont de memorare a datelor*).

Majoritatea algoritmilor de tip off-line pot fi transformați (exact sau aproximativ) în algoritmi de tip on-line sau recursiv, de o manieră directă. Scopul principal al unei astfel de operații este de a oferi capacitatea de urmărire în timp a caracteristicilor unor procese variabile și sau neliniare. Modelarea neliniarităților unui proces prin această tehnică este posibilă în cazul în care clasa de modele (de exemplu, ARMAX) nu este părăsită în cursul funcționării. În caz contrar, este mai indicată o abordare orientată pe neliniarități sau de tip multi-model (cu baleierea mai multor clase de modele).

Transformarea algoritmilor off-line în algoritmi on-line se realizează de regulă prin 2 tehnici: modificarea algoritmului de bază (de exemplu, ca în definiția (107)) sau trecerea la o abordare pe stare cu predicția stărilor prin *filtrare Kalman*.

În afara utilizării ferestrelor, algoritmul de bază din [Figura 23](#) mai poate fi modificat în sensul utilizării metodelor de gradient (de tip Newton-Raphson) pentru reactualizarea vectorului parametrilor necunoscuți. Există 2 abordări practice: cu gradient ne-normalizat și cu gradient normalizat. În ambele cazuri, matricea  $P_k$  este forțată să fie proporțională cu matricea unitară.

Figura 25. Algoritmul recursiv cu fereastră exponențială.

➤ Date de intrare:

- ordinele modelului de identificare:  $na, nb, nc, nd$  și  $nf$ ;
- factorul de uitare:  $\lambda \in [0,1]$  (de regulă,  $\lambda \in [0.95, 0.995]$ );
- o colecție redusă de date intrare-ieșire măsurate (dacă este posibil):  
 $\mathcal{D}_{N_0} = \{u[n]\}_{n \in \overline{1, N_0}} \cup \{y[n]\}_{n \in \overline{1, N_0}}$  (cu  $N_0$  de ordinul zecilor, cel mult);
- un semnal instrumental extern:  $\{f[n]\}_{n \in \overline{1, N}}$  (eventual).

- Centrarea datelor pe medie:  $y \leftarrow y - \bar{y}$ ,  $u \leftarrow u - \bar{u}$  (și  $f \leftarrow f - \bar{f}$ , dacă a fost specificat).
  - Dacă nu a fost specificat nici un semnal instrumental, vectorul variabilelor instrumentale,  $z$ , este identic cu vectorul regresorilor  $\varphi$ . Altfel,  $z$  este definit ca în (58) sau (59)-(60), dar folosind în general semnalul instrumental extern  $f$  în locul intrării  $u$  (în particular, este posibil ca  $f \equiv u$ ).
  - Inițializare. Fie se setează arbitrar vectorul parametrilor  $\hat{\theta}_0$  și matricea  $P_0 = \alpha I_{n_\theta}$  (cu  $\alpha \in \mathbb{R}_+^*$ ) (în cazul în care nu se dispune de setul de date redus  $\mathcal{D}_{N_0}$ ), fie se estimează valoarea inițială a parametrilor ( $\hat{\theta}_0$ ) folosind o metodă off-line adecvată modelului particular utilizat (din clasa MCMMP-MVI) și se egalează matricea  $P_0$  cu inversa matricii de covarianță  $R_0$  folosită în calculul lui  $\hat{\theta}_0$  (în cazul în care setul de date redus  $\mathcal{D}_{N_0}$  este disponibil).
  - Pentru  $k \geq 1$ :
    - Se evaluează eroarea de predicție:  $\varepsilon[k] = y[k] - \varphi^T[k] \hat{\theta}_{k-1}$ .
    - Se vectorul auxiliar:  $\xi_k = P_{k-1} z[k]$ .
    - Se evaluează câștigul de senzitivitate:  $\gamma_k = \frac{\xi_k}{\lambda + \varphi^T[k] \xi_k}$ .
    - Se reactualizează inversa matricii  $R_k$ , adică:  $P_k = \frac{1}{\lambda} (P_{k-1} - \gamma_k \varphi^T[k] P_{k-1})$  (cu evitarea inversării explicite a matricilor).
    - Se reactualizează vectorul parametrilor:  $\hat{\theta}_k = \hat{\theta}_{k-1} + \gamma_k \varepsilon[k]$ .
- Date de ieșire: parametrii modelului ( $\hat{\theta}_k$ ) la fiecare pas de reactualizare  $k \geq 0$ .

În cazul algoritmului cu gradient ne-normalizat, matricea  $P_k$  este proporțională cu matricea unitară printr-o constantă  $\gamma \in \mathbb{R}_+^*$ , numită *câștig de gradient*. În cazul algoritmului cu gradient normalizat, câștigul  $\gamma$  este împărțit la fiecare pas cu un factor proporțional cu norma vectorului regresorilor. În al doilea caz, factorul de proporționalitate variază în funcție de momentul curent de reactualizare. Ambele tipuri de algoritmi sunt descrise în Figura 26. Mai multe detalii legate de raționamentul care a condus la aceștia se pot găsi în [LjL99].

Figura 26. Algoritmi recursivi de tip gradient.

- Date de intrare:
- ordinele modelului de identificare:  $na, nb, nc, nd$  și  $nf$ ;
  - cîștigul de gradient:  $\gamma \in \mathbb{R}_+^*$ ;
  - o colecție redusă de date intrare-ieșire măsurate (dacă este posibil):  
 $\mathcal{D}_{N_0} = \{u[n]\}_{n \in \overline{1, N_0}} \cup \{y[n]\}_{n \in \overline{1, N_0}}$  (cu  $N_0$  de ordinul zecilor, cel mult);
  - un semnal instrumental extern:  $\{f[n]\}_{n \in \overline{1, N}}$  (eventual).
- Centrarea datelor pe medie:  $y \leftarrow y - \bar{y}$ ,  $u \leftarrow u - \bar{u}$  (și  $f \leftarrow f - \bar{f}$ , dacă a fost specificat).
  - Dacă nu a fost specificat nici un semnal instrumental, vectorul variabilelor instrumentale,  $z$ , este identic cu vectorul regresorilor  $\varphi$ . Altfel,  $z$  este definit ca în (58) sau (59)-(60), dar folosind în general semnalul instrumental extern  $f$  în locul intrării  $u$  (în particular, este posibil ca  $f \equiv u$ ).
  - Inițializare. Fie se setează arbitrar vectorul parametrilor  $\hat{\theta}_0$  (în cazul în care nu se dispune de setul de date redus  $\mathcal{D}_{N_0}$ ), fie se estimează valoarea inițială a parametrilor ( $\hat{\theta}_0$ ) folosind o metodă off-line adecvată modelului particular utilizat (din clasa MCMMP-MVI) (în cazul în care setul de date redus  $\mathcal{D}_{N_0}$  este disponibil). Se setează  $P_0 = \mathcal{I}_{n_\theta}$ .
  - Pentru  $k \geq 1$ :
    - Se evaluează eroarea de predicție:  $\varepsilon[k] = y[k] - \varphi^T[k] \hat{\theta}_{k-1}$ .
    - Se reactualizează matricea  $P_k$  astfel:  $P_k = P_0$  (gradient ne-normalizat) sau  $P_k = P_0 / \|\varphi[k]\|^2$  (gradient normalizat).
    - Se reactualizează vectorul parametrilor:  $\hat{\theta}_k = \hat{\theta}_{k-1} - P_k \varphi[k] \varepsilon[k]$ .
- Date de ieșire: parametrii modelului ( $\hat{\theta}_k$ ) la fiecare pas de reactualizare  $k \geq 0$ .

Se constată cu ușurință că algoritmi din Figurile 23 (de bază) și 26 (de gradient) sunt sensibil diferiți prin maniera de evaluare a matricii  $P_k$ . De altfel, MRPL-R este bazată în principal pe algoritmi de gradient ca în Figura 26.

În cazul trecerii la reprezentarea pe stare cu utilizarea filtrării Kalman, abordarea pleacă de la ipoteza ca parametrii adevărați ai procesului,  $\theta^*$ , variază în timp după o regulă cunoscută sub numele de “plimbare la întâmplare” (*random walk*). Mai precis, regula de variație este următoarea:

$$\theta^*[n] = \theta^*[n-1] + v[n], \quad \forall n \geq 1, \quad (112)$$

unde  $v$  este un vector de zgomote albe Gaussiene, avînd matricea de auto-covarianță  $R_1 = E\{v[n]v^T[n]\}$  (numită și *matrice de răspîndire*).

Dacă  $\lambda^2$  este dispersia zgomotului alb  $e$  care apare în ecuația (84) a modelului general de identificare, atunci prin utilizarea filtrului Kalman se obține algoritmul recursiv din Figura 27. (Pentru detalii privind filtrarea Kalman se pot consulta [LjL99] și/sau [SoSt89].) Acest algoritm, deși oferă estimări de precizie ridicată este totuși mai complex decât predecesorii săi.

Figura 27. Algoritmul recursiv cu filtrare Kalman.

➤ Date de intrare:

- a. ordinele modelului de identificare:  $na, nb, nc, nd$  și  $nf$ ;
- b. matricea de răspîndire:  $R_v > 0$ ;
- c. dispersia estimată a zgomotului alb de proces:  $\lambda^2$ ;
- d. o colecție redusă de date intrare-ieșire măsurate (dacă este posibil):  
 $\mathcal{D}_{N_0} = \{u[n]\}_{n \in \overline{1, N_0}} \cup \{y[n]\}_{n \in \overline{1, N_0}}$  (cu  $N_0$  de ordinul zecilor, cel mult);
- e. un semnal instrumental extern:  $\{f[n]\}_{n \in \overline{1, N}}$  (eventual).

1. Centrarea datelor pe medie:  $y \leftarrow y - \bar{y}$ ,  $u \leftarrow u - \bar{u}$  (și  $f \leftarrow f - \bar{f}$ , dacă a fost specificat).
  2. Dacă nu a fost specificat nici un semnal instrumental, vectorul variabilelor instrumentale,  $z$ , este identic cu vectorul regresorilor  $\varphi$ . Altfel,  $z$  este definit ca în (58) sau (59)-(60), dar folosind în general semnalul instrumental extern  $f$  în locul intrării  $u$  (în particular, este posibil ca  $f \equiv u$ ).
  3. Inițializare. Fie se setează arbitrar vectorul parametrilor  $\hat{\theta}_0$  și matricea  $P_0 = \alpha I_{n_\theta}$  (cu  $\alpha \in \mathbb{R}_+^*$ ) (în cazul în care nu se dispune de setul de date redus  $\mathcal{D}_{N_0}$ ), fie se estimează valoarea inițială a parametrilor ( $\hat{\theta}_0$ ) folosind o metodă off-line adecvată modelului particular utilizat (din clasa MCMMP-MVI) și se egalează matricea  $P_0$  cu inversa matricii de covarianță  $R_0$  folosită în calculul lui  $\hat{\theta}_0$  (în cazul în care setul de date redus  $\mathcal{D}_{N_0}$  este disponibil).
  4. Pentru  $k \geq 1$ :
    - 4.1. Se evaluează eroarea de predicție:  $\varepsilon[k] = y[k] - \varphi^T[k] \hat{\theta}_{k-1}$ .
    - 4.2. Se evaluează vectorul auxiliar:  $\xi_k = P_{k-1} z[k]$ .
    - 4.3. Se evaluează câștigul de sensibilitate:  $\gamma_k = \frac{\xi_k}{\lambda^2 + \varphi^T[k] \xi_k}$ .
    - 4.4. Se reactualizează matricea  $P_k$ , adică:  $P_k = P_{k-1} + R_v - \gamma_k \varphi^T[k] P_{k-1}$  (cu evitarea inversării explicite a matricilor).
    - 4.5. Se reactualizează vectorul parametrilor:  $\hat{\theta}_k = \hat{\theta}_{k-1} + \gamma_k \varepsilon[k]$ .
- Date de ieșire: parametrii modelului ( $\hat{\theta}_k$ ) la fiecare pas de reactualizare  $k \geq 0$ .

### B. Rutine MATLAB pentru identificare recursivă

Biblioteca de rutine dedicată IS conține un număr de 6 funcții care implementează algoritmi recursivi: **rarmax** (pentru ARMAX și MMEP-R), **rarx** (pentru ARX și MCMMP-R sau MVI-R), **rbj** (pentru BJ și MMEP-R), **rpem** (pentru modele generale și MMEP-R), **rplr** (pentru ARMAX și MPRL-R) și **roe** (pentru OE și MMEP-R). Oricare dintre aceste funcții poate fi apelată cu ajutorul unei comenzi cu următoarea sintaxă tipică:

```
[theta,ypred] = rnume(D,si,ma,pa) ;
```

unde: **D** este obiectul de tip **IDDATA** (vezi **Problema 2.3**) corespunzător datelor generate (intrarea se regăsește în **D.u**, iar ieșirea în **D.y**);  
**si** este vectorul indicilor structurali și al întârzierii modelului (ca în cazul rutinei **pem**):

```
si = [na nb nc nd nf nk];
```

**ma** este un argument care indică metoda de adaptare a algoritmului off-line la algoritmul on-line (șir de 2 caractere):

'ff' se va opera cu criteriu pătratic afectat de fereastră exponențială (i.e. cu factor de uitare; 'ff' = *forgetting factor*);

'ug' se va utiliza o metodă de gradient (Newton-Raphson) ne-normalizată ('ug' = *unnormalized gradient*);

'ng' se va utiliza o metodă de gradient (Newton-Raphson) normalizată ('ng' = *normalized gradient*);

'kf' se va utiliza reprezentarea pe stare și filtrarea Kalman (aici: 'kf' = *Kalman filtering*);

**pa** este un parametru de adaptare corespunzător metodei de adaptare a algoritmului off-line la algoritmul on-line, adică argumentului **ma** (scalar sau matrice):

$\lambda$  factorul de uitare (scalar) pentru fereastra exponențială (când argumentul **ma** este setat cu 'ff');

$\gamma$  câștigul dorit (scalar) în cazul utilizării algoritmilor de gradient (când argumentul **ma** este setat cu 'ug' sau 'ng');

$R_v$  matricea de răspîndire în cazul utilizării algoritmului bazat pe filtrare Kalman (când argumentul **ma** este setat cu 'kf'); în acest caz, parametrul  $\lambda^2$  (dispersia zgomotului alb) este considerat implicit egal cu 1; dacă, în realitate,  $\lambda^2$  nu este unitară, se poate demonstra că estimația parametrilor nu este afectată dacă se scalează matricile  $R_v$  și  $P_0$  cu valoarea estimată a sa (urmînd să se lucreze tot cu  $\lambda^2 = 1$ , ca în cazul implicit).

**theta** este matricea parametrilor estimați variabili în timp; fiecare linie a matricii memorează valoarea parametrilor la un anumit moment de timp; numărul de linii este egal cu lungimea orizontului de măsură (adică a vectorilor **D.u** și **D.y**); pe fiecare linie, parametrii sunt precizați

în ordinea alfabetică a numelor polinoamelor pe care le reprezintă ( $A$ ,  $B$ ,  $C$ ,  $D$ ,  $F$ );

**ypred** este vectorul ieșirii predictate a procesului la fiecare moment de timp, folosind modelul matematic reactualizat; lungimea sa este egală cu lungimea orizontului de măsură.

Evident, **nume** este unul din următoarele: **armax**, **arx**, **bj**, **pem**, **plr** sau **oe**. Toate rutinele permit și precizarea unor parametri suplimentari de intrare prin care să se poată specifica o inițializare dorită ( $\theta_0$ ,  $P_0$  și chiar  $\phi[0]$ ). Inițializarea poate fi construită și din valorile corespunzătoare obținute prin întreruperea unui algoritm recursiv. Apelul funcțiilor cu setul complet de argumente este următorul:

```
[theta,ypred,P,phi] = rnume(D,si,ma,pa,theta0,P0,phi0) ;
```

unde: **theta0** este vectorul inițial al parametrilor;

**P0** este matricea inițială  $P_0$ ;

**phi0** este vectorul inițial al regresorilor  $\phi[0]$ .

**P** este matricea finală  $P_N$ ;

**phi** este vectorul final al regresorilor  $\phi[N]$ .

Restul argumentelor funcției au fost deja explicați mai sus.

Biblioteca nu include și versiuni ale algoritmului bazat pe MVI-R. Utilizatorul este așadar invitat să proiecteze o rutină de tipul celor de mai sus, numită sugestiv **riv**. Pentru a nu complica proiectarea rutinei, se poate considera doar varianta de algoritm din **Figura 25** (cu fereastră exponențială), utilizatorul avînd posibilitatea de a rula algoritmul de bază din **Figura 23** prin specificarea factorului unitar cu valoarea unitară. Apelul tipic al rutinei ar trebui să fie următorul:

```
[theta,ypred,P,phi,z] = riv(D,si,f,lambda,theta0,P0,phi0,z0) ;
```

unde: **f** este semnalul instrumental (implicit: **f=D.u**);

**lambda** este factorul de uitare ( $\lambda \in (0,1]$ ) (implicit: **lambda=1**);

**z0** este vectorul inițial al instrumentelor  $z[0]$ ;

**z** este vectorul final al instrumentelor  $z[N]$ .

Restul argumentelor funcției au fost explicate mai sus. De notat că parametrii structurali **si** conțin numai ordinele modelului ARX.

Obiectivul acestui capitol este de a efectua o comparație între 4 metode recursive de identificare: MCMMP-R, MVI-R, MMEP-R și MRLP-R, în identificarea parametrilor unor modele din clasa ARMAX.

## 6.2. Exerciții

### Exercițiul 6.1

Descrieți algoritmul MMEP-R folosind un model ARMAX și algoritmul general prezentat în **Figura 23**.

**Exercițiul 6.2**

Justificați prin calcule adecvate algoritmul din Figura 24.

**Exercițiul 6.3**

Justificați prin calcule adecvate algoritmul din Figura 25.

**Exercițiul 6.4**

Să se demonstreze că, în cazul utilizării ferestrei exponențiale (109), măsurători mai vechi de  $T_\lambda = \lambda/(1-\lambda)$  față de momentul curent sunt incluse în criteriul pătratic cu o pondere de aproximativ 36% din ponderea măsurătorilor celor mai recente, dacă factorul de uitare  $\lambda$  este situat într-o vecinătate a lui 1.

**6.3. Probleme de simulare**

Datele pe care toate mini-simulatoarele din cadrul capitolului le vor analiza sunt generate de următorul model ARMAX[1,1,1], cu parametri constanți sau variabili în timp:

$$(1 + a_1[n]q^{-1})y[n] = b_1[n]q^{-1}u[n] + (1 + c_1[n]q^{-1})e[n], \quad \forall n \in \mathbb{N}^*, \quad (113)$$

unde  $u$  și  $e$  sunt zgomote albe Gaussiene, de medie nulă și dispersii unitare ( $\lambda_u^2 = \lambda_e^2 = 1$ ).

În cazul în care parametrii sunt constanți, ei au următoarele valori:

$$a_1[n] = a_{10} = -0.7, \quad b_1[n] = b_{10} = 0.6, \quad c_1[n] = c_{10} = -0.9, \quad \forall n \in \mathbb{N}^*. \quad (114)$$

Un set de parametri variabili în timp poate fi considerat următorul:

$$a_1[n] \stackrel{def}{=} a_{10} \cos\left(\frac{10\pi n}{N}\right), \quad b_1[n] \stackrel{def}{=} b_{10} \operatorname{sgn}\left[\cos\left(\frac{4\pi n}{N}\right)\right], \quad c_1[n] \stackrel{def}{=} c_{10} \operatorname{Sc}\left(\frac{18\pi n}{N}\right), \quad \forall n \in \mathbb{N}^*, \quad (115)$$

unde “sgn” este notația pentru operatorul de luare a semnului, iar “Sc” este funcția:

$\operatorname{Sc}(t) \stackrel{def}{=} \sin(t)/t$  (*sinus cardinal* sau *sinus atenuat*). Practic,  $a_1$  variază după o lege armonică,  $b_1$  – după o lege de tip “tren de impulsuri”, iar  $c_1$  – după o lege hiperbolică oscilatorie. Pe termen lung, modelul ARMAX (113) tinde să devină un model ARX.

Orizontul de măsură,  $N$ , care intervine și în legile de variație a parametrilor (115), este specificat de utilizator, dar nu se poate situa sub 200.

Pentru generarea și stocarea datelor, se recomandă scrierea unei rutine separate, numite **gdata\_vp**, al cărei apel general să fie următorul:

**[D,V,P] = gdata\_vp(cv,N,sigma,lambda,bin) ;**

unde: **cv** este un comutator care arată tipul de proces: cu parametri constanți dați de ecuațiile (114) (**cv=0**) sau cu parametri variabili dați de ecuațiile (115) (**cv~=0**); (implicit: **cv=0**);



## Aspecte practice în Modelarea și Identificarea Sistemelor

<b>N</b>	este orizontul de măsură (implicit: <b>N=250</b> );
<b>sigma</b>	este deviația standard a intrării SPA (implicit: <b>sigma=1</b> );
<b>lambda</b>	este deviația standard a zgomotului alb Gaussian (implicit: <b>lambda=1</b> );
<b>bin</b>	este un parametru care arată tipul de intrări dorit: <b>bin=0</b> (intrare SPAB Gaussiană); <b>bin~=0</b> (implicit, intrare SPAB Gaussiană bipolară);
<b>D</b>	este obiectul de tip <b>IDDATA</b> (vezi <b>Problema 2.3</b> ) corespunzător datelor generate (intrarea se regăsește în <b>D.u</b> , iar ieșirea în <b>D.y</b> );
<b>V</b>	este obiectul de tip <b>IDDATA</b> corespunzător zgomotelor generate (zgomotul alb se regăsește în <b>V.u</b> , iar zgomotul colorat (adică MA-filtrat) în <b>V.y</b> );
<b>P</b>	este obiectul de tip <b>IDMODEL</b> (vezi <b>Problema 3.3</b> ) corespunzător modelului de proces furnizor de date; în cazul parametrilor constanți (114): <b>P.a=[1 a0]</b> , <b>P.b=[0 b0]</b> , <b>P.c=[1 c0]</b> ; în cazul parametrilor variabili (115): <b>P.a=[1 a]</b> , <b>P.b=[0 b]</b> , <b>P.c=[1 c]</b> (unde <b>a</b> , <b>b</b> și <b>c</b> sunt vectorii de variație).

Se vor genera 2 seturi de date de dimensiune  $N$ : unul provenit de la procesul cu parametri constanți ( $\mathcal{D}_c$ ) și altul – de la procesul cu parametri variabili ( $\mathcal{D}_v$ ).

### Problema 6.1

Mini-simulatorul **ISLAB\_6A** efectuează o comparație între cele 4 metode de identificare recursive menționate, adică: MCMMP-R, MVI-R, MMEP-R și MRPL-R, folosind setul de date  $\mathcal{D}_c$ . Primele 2 metode operează cu modelul ARX[1,1], în timp ce ultimele 2 – cu modelul ARMAX[1,1,1]. Pentru aprecierea performanțelor lor, sunt afișate 4 ferestre grafice care includ variațiile parametrilor reali (aici constanți) suprapuse peste variațiile parametrilor estimați și variația ieșirii simulate suprapuse peste ieșirea reală (măsurată) a procesului (**Figurile 28-31**).

- Să se comenteze rezultatele obținute cu ajutorul mini-simulatorului **ISLAB\_6A**. Care ar fi explicațiile performanțelor mai slabe ale MCMMP-R în estimarea parametrului părții AR?
- Să se proiecteze mini-simulatorul **ISLAB\_6B**, similar ca structură cu **ISLAB\_6A**, dar care operează cu datele  $\mathcal{D}_v$ . Comentați rezultatele obținute.

### Problema 6.2

Să se proiecteze mini-simulatorul **ISLAB\_6C**, care să afișeze performanțele MCMMP-R (sau MVI-R) pentru inițializările  $P_0 = \alpha I$ , cu  $\alpha \in \{0.01, 0.1, 1, 10, 100\}$ .

### Problema 6.3

Rutinele recursive folosite în mini-simulatoarele din **Problema 6.1** au posibilitatea de a opera cu fereastra exponențială aplicată erorii de predicție. Să se proiecteze mini-simulatorul **ISLAB\_6D**, care să afișeze performanțele MCMMP-R (sau MVI-R) pentru următoarele valori ale factorului de uitare:  $\lambda \in \{0.95, 0.96, 0.97, 0.98, 0.99\}$ . Comentați rezultatele obținute în ultimele 2 probleme.

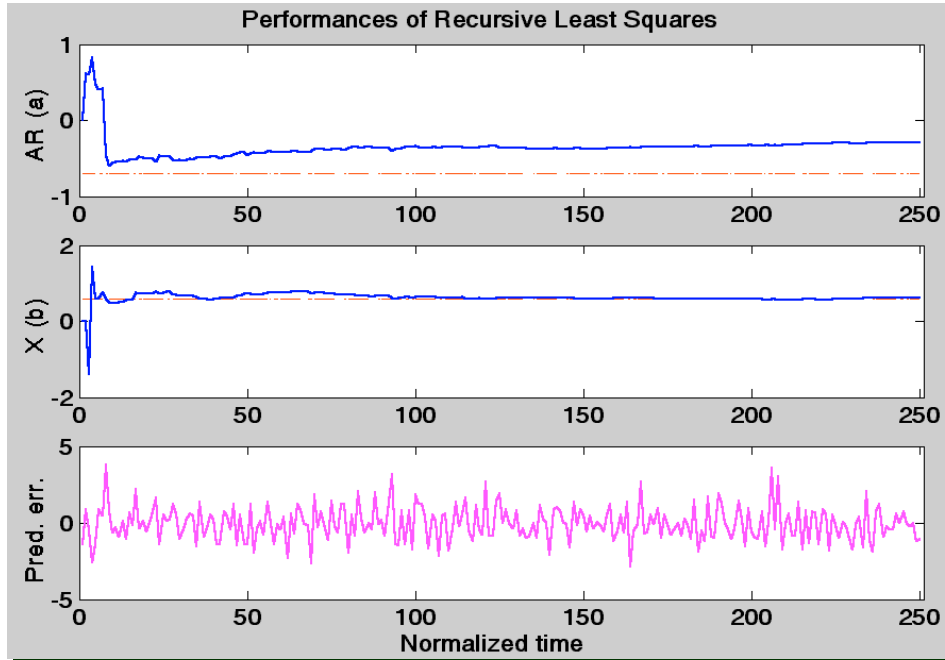


Figura 28. Performanțele MCMMP-R.

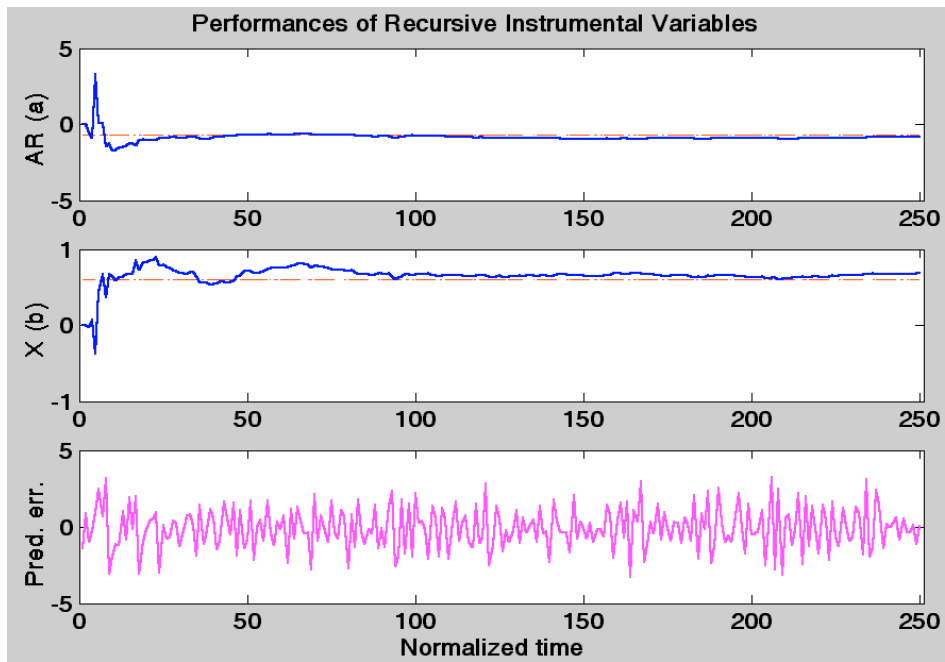


Figura 29. Performanțele MVI-R.

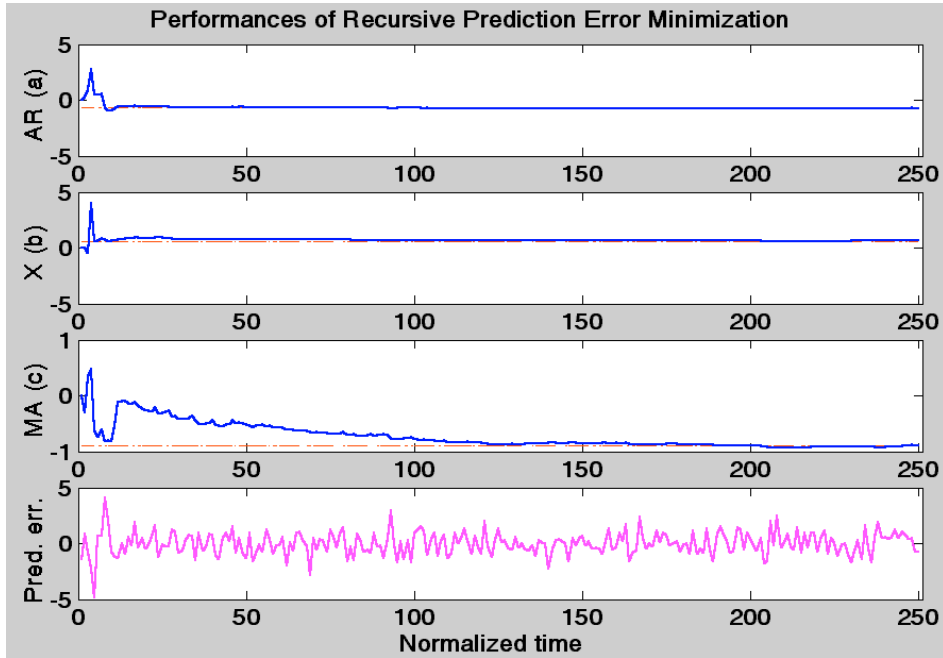


Figura 30. Performanțele MMEP-R.

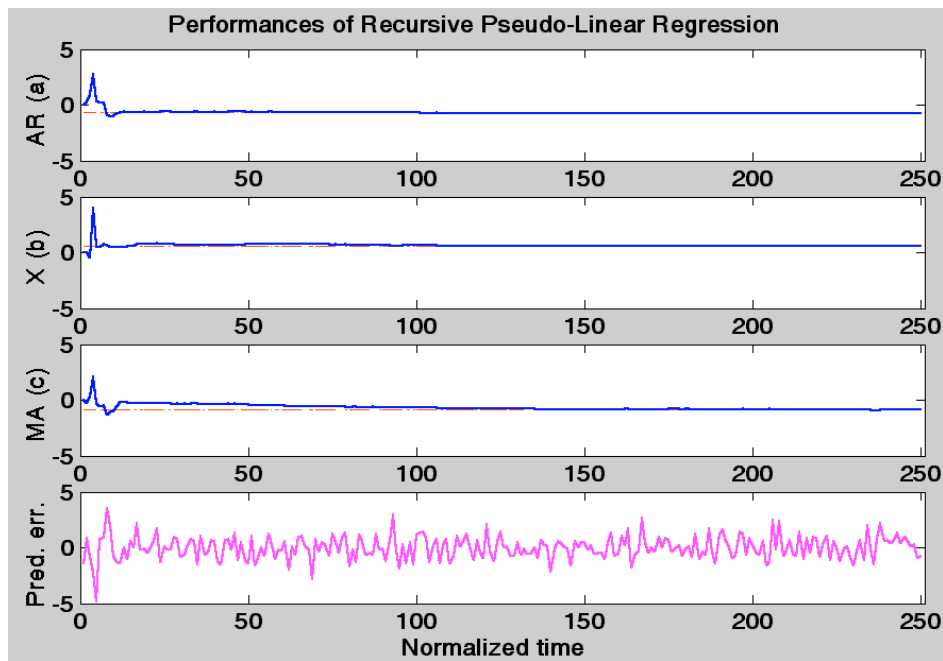


Figura 31. Performanțele MRPL-R.

## Capitolul 7

### Aplicații de identificare recursivă

#### 7.1. Contextul general de lucru

Identificarea recursivă este utilizată în numeroase aplicații moderne (în special de control adaptiv, filtrare adaptivă sau estimare spectrală). În cadrul acestui capitol, discuția este axată pe două astfel de aplicații: aproximarea modelelor de complexitate ridicată prin modele mai simple și identificarea parametrilor fizici ai unui proces. Contextul de lucru al capitolului precedent este îmbogățit aici cu câteva noi abordări de identificare.

##### A. Aproximarea modelelor complexe

Există multe situații în care dinamica procesului analizat este prea complexă pentru a fi reprezentată de un model matematic precis. De cele mai multe ori, precizia este sacrificată în scopul atingerii unui anumit grad de eficiență a algoritmilor de estimare parametrică. Modelele aproximative adoptate în aceste cazuri fac parte fie din clasa generală (84), fie, mai des, din clasa ARMAX (1).

Pentru simplificarea discuției, să considerăm numai partea utilă a unui model ARMAX, descrisă de funcția de sistem rațională  $H \equiv B/A$ , ca în Figura 1. Aceasta aproximează cu o anumită precizie partea utilă din procesul de complexitate ridicată, notată prin  $H^*$ . De regulă, modelul ARMAX nu poate reprezenta procesul pe întreaga bandă de frecvențe a răspunsului său în frecvență  $H^*(e^{j\omega})$ , datorită complexității acestuia și a variabilității în timp, dar pot exista frecvențe în jurul cărora precizia de aproximare este satisfăcătoare. Dacă aceste frecvențe sunt cunoscute, procesul poate fi stimulat cu un SPA(B) filtrat în așa fel încât setul de date intrare-ieșire astfel generat să reflecte comportamentul procesului în jurul lor. Vom nota filtrul aplicat intrării prin  $\mathcal{F}$  (de regulă un filtru de tip trece-bandă), în timp ce intrarea filtrată va fi referită prin  $u_f$ . Astfel:

$$u_f \equiv \mathcal{F}(q^{-1})u. \quad (116)$$

Convenim să re-notăm funcția de sistem  $H(q^{-1})$  prin  $H(q^{-1}, \theta)$ , pentru a pune mai bine în evidență dependența acesteia de parametrii modelului. O schimbare similară va interveni și în notația răspunsului în frecvență.

În aceste condiții, determinarea parametrilor modelului se poate efectua rezolvînd următoarea problemă de minimizare a erorii pătratice de estimare spectrală:

$$\hat{\theta} = \arg \min_{\theta \in \mathcal{S}} \mathcal{V}(\theta), \quad \text{unde:}$$
$$\mathcal{V}(\theta) \stackrel{\text{def}}{=} \int_{-\pi}^{+\pi} \left| \mathcal{F}(e^{j\omega}) \right|^2 \left| H^*(e^{j\omega}) - H(e^{j\omega}, \theta) \right|^2 \phi_u(\omega) d\omega, \quad \forall \theta \in \mathcal{S}. \quad (117)$$

Ca de obicei,  $\mathcal{S}$  este domeniul de stabilitate al modelului ARMAX.

Recunoaștem în definiția (117) notația densității spectrale de putere a intrării. Evident, această definiție este conformă cu definiția (116). Pentru rezolvarea problemei de optimizare anterioare poate fi utilizată o variantă a MMEP-R. Aceasta corespunde rutinei Matlab `rarmax`, din biblioteca IS (rutină descrisă indirect în cadrul capitolului precedent).

Problema cea mai dificilă rămâne doar proiectarea adecvată a filtrului  $\mathcal{F}$ , deoarece, de regulă, nu se cunosc frecvențele în jurul cărora modelul ARMAX aproximează satisfăcător comportamentul procesului. Folosirea frecvențelor de rezonanță ale procesului constituie adesea o alegere inspirată, cu condiția ca măcar aceste frecvențe să poată fi cunoscute a priori, chiar și imprecis. Determinarea grosieră a frecvențelor de rezonanță (sau a frecvențelor naturale de oscilație) ale unui proces se poate efectua, de exemplu, plecând de la ecuațiile diferențiale rezultate din aplicarea legilor Fizicii care guvernează funcționarea aceluși proces. În acest caz, se apelează la metode de identificare a parametrilor fizici semnificativi pentru proces (cum va fi descris în paragraful următor). O altă abordare, mai directă, este cea în care se trasează spectrul răspunsului în frecvență al procesului determinat grosier din datele de intrare-ieșire obținute cu intrarea ne-filtrată. Spectrul poate pune în evidență frecvențele dominante în jurul cărora ar trebui utilizat modelul ARMAX aproximant. Mai precis, din datele măsurate se obține:

$$\left| H^*(e^{j\omega}) \right|^2 \equiv \frac{\phi_y(\omega)}{\phi_u(\omega)}, \quad \forall \omega \in [0, \pi], \quad (118)$$

care se trasează grafic (în dB). Pe grafic se pot pune în evidență frecvențele dominante sau dorite. Se recomandă determinarea cîte unui model ARMAX în jurul fiecărei frecvențe selectate.

Presupunînd că a fost selectată o frecvență în jurul căreia dorim aproximarea modelului procesului complex cu un proces mai simplu, de tip ARMAX, filtrul  $\mathcal{F}$  (de tip trece-bandă) poate fi proiectat cu ajutorul tehnicilor cunoscute din PS. Există numeroase metode de proiectare a filtrelor trece-jos, trece-sus, trece-bandă, stop-bandă, multi-bandă, etc. [PrMa96]. Cea mai mare parte dintre acestea sunt implementate în cadrul bibliotecii MATLAB dedicată domeniului PS. Ne vom opri doar asupra unei metode relativ simple, care permite și proiectarea de filtre trece-bandă (tipul de filtru de interes în această aplicație): *Metoda lui Butterworth*.

Filtrele Butterworth sunt definite în contextul semnalelor analogice. Versiuni numerice ale acestora se pot obține prin discretizare. Spectrul unui astfel de filtru este, prin definiție, următorul:

$$\left| \mathcal{F}(j\Omega) \right|^2 \stackrel{\text{def}}{=} \frac{1}{1 + (\Omega/\Omega_c)^{2K}}, \quad \forall \Omega \in \mathbb{R}, \quad (119)$$

unde:  $K \in \mathbb{N}^*$  este ordinul filtrului, iar  $\Omega_c > 0$  este pulsația de tăiere (adică pulsația pentru care spectrul atinge valoarea de  $-3$  dB în scară logaritmică sau, echivalent,  $1/\sqrt{2}$  în scară liniară). Evident spectrul este monoton descrescător pentru pulsații pozitive. Mai mult, odată cu creșterea ordinului filtrului, caracteristicile sale în frecvență se îmbunătățesc: pulsația de tăiere este abordată mai abrupt, iar lobiile parazitiți din banda de stop sunt atenuați. **Figurile 32 și 33** arată caracteristicile cîtorva filtre.

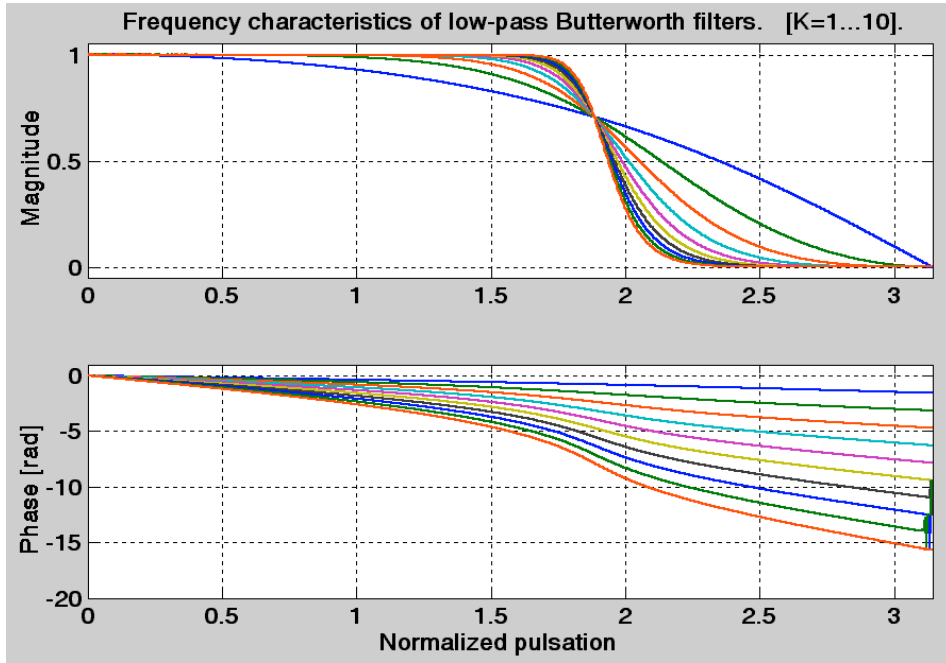


Figura 32. Caracteristicile în frecvență ale filtrelor Butterworth de tip trece-jos.

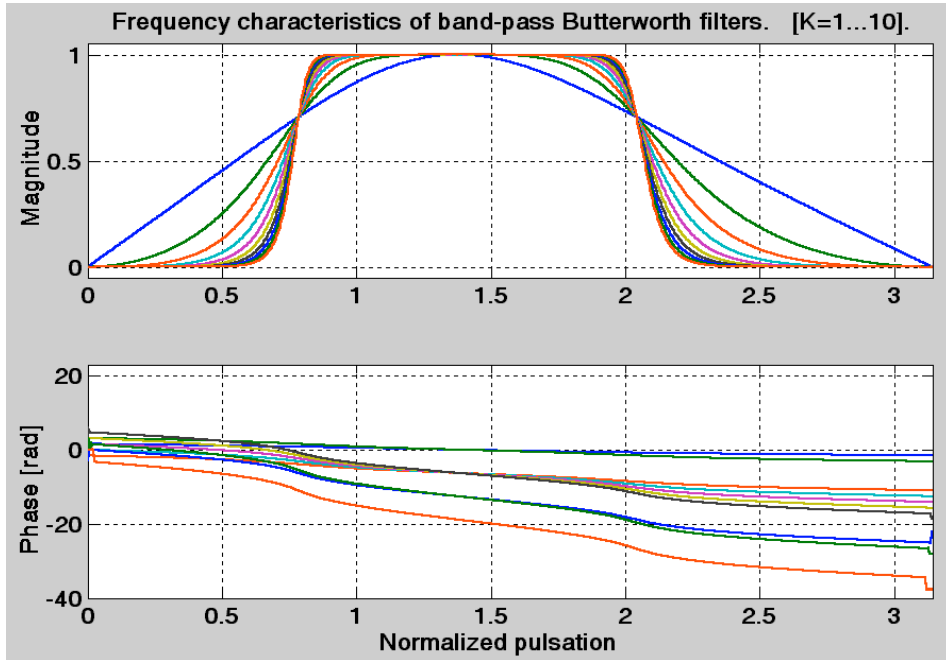


Figura 33. Caracteristicile în frecvență ale filtrelor Butterworth de tip trece-bandă.

Definiția (119) implică următoarea proprietate remarcabilă verificată de funcția de transfer a filtrului:

$$\mathcal{F}(s)\mathcal{F}(-s) = \frac{1}{1 + (-s^2/\Omega_c^2)^K}. \quad (120)$$

Proprietatea (120) arată că polii funcției complexe  $\mathcal{F}(s)\mathcal{F}(-s)$  sunt situați numai pe cercul de rază  $\Omega_c$  în puncte echidistante:

$$s_k^\pm = \pm j\Omega_c \exp\left(\frac{(2k+1)\pi j}{2K}\right), \quad \forall k \in \overline{0, K-1}. \quad (121)$$

Funcția de transfer  $\mathcal{F}(s)$  fiind stabilă, ea extrage din mulțimea (121) numai polii cu partea reală negativă. Ceilalți poli (simetrici) sunt produși de  $\mathcal{F}(-s)$ .

În proiectarea unui filtru Butterworth, se pleacă de la restricția ca valoarea spectrului la o anumită pulsație precizată  $\Omega_a > 0$  să fie  $\delta_a \in (0,1)$  și se cere determinarea ordinului său. Evident, folosind definiția (119), ordinul filtrului rezultă imediat:

$$K = \left\lceil \frac{\log \frac{1-\delta_a^2}{\delta_a^2}}{2 \log \frac{\Omega_a}{\Omega_c}} \right\rceil, \quad (122)$$

ținând cont că atenuarea crește (adică  $\delta_a$  scade) odată cu creșterea ordinului filtrului (vezi Figurile 32 și 33). Pentru a proiecta un filtru Butterworth avem deci nevoie de următoarele date: pulsația de tăiere  $\Omega_c$  și ordinul  $K$  sau pulsația de tăiere  $\Omega_c$  și perechea  $(\Omega_a, |F(j\Omega_a)| = \delta_a)$ .

Practic, filtrul Butterworth este de tip trece-jos. Pentru a proiecta un filtru de tip trece-bandă, este suficient să se proiecteze 2 filtre de tip trece jos (cîte unul pentru fiecare pulsație de tăiere a benzii de trecere) și să se scadă caracteristica filtrului de bandă mai îngustă din cea de bandă mai largă.

Funcția Matlab cu ajutorul căreia se poate proiecta un filtru Butterworth este **butter**, cu apelul tipic:

$$[B,A] = \text{butter}(K,fc,type) ;$$

unde: **K** este ordinul filtrului (întreg strict pozitiv); dacă ordinul trebuie obținut cu ajutorul relației (122), se poate utiliza funcția MATLAB **buttord**;  
**fc** este frecvența de tăiere a filtrului, exprimată ca un număr în intervalul (0,1), cu valoarea unitară corespunzând jumătății frecvenței de eșantionare (conform Teoremei lui Shannon-Nyquist [Std9602a]); de exemplu, dacă  $\nu_s$  este frecvența de eșantionare și **fc=0.7**, atunci frecvența de tăiere este  $f_c = 0.35\nu_s$ ; pe scala pulsației normalizate

- (adică în gama  $[0, \pi]$ , unde  $\pi$  corespunde jumătății frecvenței de eșantionare), pulsația de tăiere va fi  $\omega_c = 0.7\pi$ ; în mod implicit, dacă **fc** este scalar, se proiectează un filtru de tip trece-jos; se poate însă preciza **fc** ca un vector cu 2 elemente subunitare, caz în care se proiectează un filtru de tip trece-bandă, de ordin  $2 \cdot K$ ; frecvențele sale de tăiere sunt alese în funcție de elementele lui **fc**;
- type** dacă se dorește proiectarea de filtre de tip trece-sus sau stop-bandă, se poate preciza aceasta prin argumentul opțional **type**; astfel, **type** poate fi: **'high'** (pentru filtrul trece-sus) sau **'stop'** (pentru filtrul stop-bandă, caz în care **fc** trebuie să fie un vector cu 2 elemente);
- B** polinomul care definește numărătorul funcției de transfer a filtrului; este un vector de lungime  $K+1$  (pentru filtrele trece-jos, trece-sus) sau  $2 \cdot K+1$  (pentru filtrele trece-bandă, stop-bandă), cu coeficienții în ordinea crescătoare a puterilor lui  $z^{-1}$ ;
- A** polinomul care definește numitorul funcției de transfer a filtrului; este un vector de lungime  $K+1$  (pentru filtrele trece-jos, trece-sus) sau  $2 \cdot K+1$  (pentru filtrele trece-bandă, stop-bandă), cu coeficienții în ordinea crescătoare a puterilor lui  $z^{-1}$ .

Funcția **butter** oferă direct versiunea discretă a filtrului Butterworth, dar este posibilă apelarea ei cu parametri suplimentari, în așa fel încît să se obțină versiunea originală (în timp continuu).

În practică, discretizarea unei funcții de transfer continue se efectuează cu ajutorul următoarei transformări omografice (care asociază dreptele verticale din planul complex cu cercurile și axa imaginară cu cercul unitar):

$$s = \frac{2}{T_s} \frac{z-1}{z+1}, \quad (123)$$

unde  $T_s$  este perioada de eșantionare. Cu alte cuvinte, înlocuind variabila Laplace  $s$  cu termenul drept al egalității (123), în funcția de transfer din timp continuu, se obține funcția de transfer din timp discret (normalizat cu perioada de eșantionare). Reamintim că transformarea de discretizare ideală (dar neinvertibilă) este următoarea (care transformă dreptele verticale în cercuri) [LaID93], [LaID97]:

$$z = e^{sT_s}. \quad (124)$$

Utilizarea extrapolatorului de ordin zero (care constă în interpolarea valorilor eșantionate la stînga cu valori constante) permite adoptarea aproximării Padé a transformării (124), adică aproximarea cu un polinom Taylor de ordin 1 a exponențialei complexe:

$$z \cong 1 + sT_s \quad \Leftrightarrow \quad s \cong \frac{z-1}{T_s}. \quad (125)$$

Aproximarea Padé este grosieră (deși interesantă practic) și nu conservă proprietatea de a transforma verticalele în cercuri. Din aceste motive, transformarea (123), care



## Aspecte practice în Modelarea și Identificarea Sistemelor

corespunde de fapt extrapolatorului de ordin 1 (și constă în interpolarea cu drepte oblice valorilor eșantionate la stînga), este considerată mai precisă și adesea adoptată în locul ei.

Teoretic discretizarea unei funcții de transfer continue raționale  $H(s)$  se efectuează cu ajutorul Teoremei Reziduurilor [SeS01], [SeS02]:

$$H_d(z) \stackrel{\text{def}}{=} (z-1) \sum_{\{\text{poli}\}} \text{Rez} \left( \frac{H(s)}{s} \frac{1}{z - e^{sT_s}} \right). \quad (126)$$

Revenind la aplicația de identificare, mai multe etape trebuie parcurse:

1. Se stimulează procesul cu un anumit număr de semnale SPAB (de exemplu 100) și se trasează media răspunsurilor în frecvență determinate folosind relația (118).
2. După ce se stabilește frecvența în jurul căreia se dorește modelarea ARMAX și lărgimea de bandă, se proiectează filtrul Butterworth corespunzător, cu ajutorul căruia se filtrează un semnal SPAB.
3. Semnalul rezultat este folosit pentru a stimula procesul în vederea culegerii datelor intrare-ieșire de identificare (pe un orizont de măsură suficient de larg; de exemplu, cel puțin 200 de perechi de date).
4. Datele sunt întâi folosite într-un experiment de identificare ne-recursivă, în special pentru a determina ordinele modelului ARMAX.
5. În final, cu ordinele din etapa precedentă, se inițiază un experiment de identificare recursivă a modelului ARMAX, plecînd de la datele achiziționate.

Pentru problemele de simulare, se va considera că procesul generator de date este descris de un model general de tip (84), cu următoarele polinoame stabile:

$$\begin{cases} A(q^{-1}) = 1 - 1.5q^{-1} + 0.9q^{-2} \\ B(q^{-1}) = q^{-2} - 1.3q^{-3} + 0.8q^{-4} \\ C(q^{-1}) = 1 - q^{-1} + 0.2q^{-2} - 0.18q^{-4} \\ D(q^{-1}) = 1 - 2q^{-1} + 1.85q^{-2} - 0.65q^{-3} \\ F(q^{-1}) = 1 + 0.9q^{-2} \end{cases}. \quad (127)$$

Se recomandă proiectarea unei rutine separate avînd rolul de a genera datele de identificare folosind modelul (84) cu particularizarea (127). Apelul general al acestei rutine ar putea fi următorul:

**[D,V,P] = gdata\_cp(u,lambda) ;**

unde: **u** este intrarea de stimul a procesului (vector coloană); implicit, **u** este un SPAB Gaussian de lungime  $N = 250$  ;  
**lambda** este deviația standard a zgomotului alb Gaussian (implicit: **lambda=1**);  
**D** este obiectul de tip **IDDATA** (vezi **Problema 2.3**) corespunzător datelor generate (intrarea se regăsește în **D.u**, iar ieșirea în **D.y**);

- V** este obiectul de tip **IDDATA** corespunzător zgomotelor generate (zgomotul alb se regăsește în **V.u**, iar zgomotul colorat (adică filtrat) în **V.y**);
- P** este obiectul de tip **IDMODEL** (vezi **Problema 3.3**) corespunzător modelului de proces furnizor de date; parametrii din definițiile (127) se regăsesc în: **P.a**=[1 a], **P.b**=[0 b], **P.c**=[1 c], **P.d**=[1 d] și **P.f**=[1 f].

### B. Identificarea parametrilor fizici ai unui proces

Există aplicații în care procesul furnizor de date este caracterizat de un număr relativ redus de parametri cu ajutorul cărora se pot exprima ecuațiile diferențiale de funcționare deduse din legile Fizicii. Din acest motiv, parametrii dobîndesc atributul de “fizici”. Folosind în special MMEP, este posibilă estimarea parametrilor fizici din date intrare-ieșire măsurate.

Două abordări sunt în general uzitate în estimarea parametrilor fizici ai unui proces. Ambele pleacă de la exprimarea funcției de transfer continue a procesului,  $H(s)$ , în așa fel încît să se pună în evidență parametrii necunoscuți. O primă abordare constă în discretizarea funcției de transfer (cu perioada de eșantionare cu care sunt achiziționate datele intrare-ieșire) și estimarea coeficienților funcției de transfer discretizate. Parametrii fizici rezultă apoi din coeficienții estimați. A doua abordare constă în reprezentarea pe stare a sistemului continuu, cu precizarea unor parametri de stare necunoscuți corespunzători parametrilor fizici. Estimarea parametrilor de stare se poate efectua direct din datele măsurate, după discretizarea modelului pe stare.

Aplicația propusă în cadrul acestui capitol constă în identificarea parametrilor fizici ai unui motor de curent continuu caracterizat de următoarea funcție de transfer simplificată (de ordin 2):

$$H(s) \stackrel{def}{=} \frac{K}{s(1+Ts)}, \quad (128)$$

unde  $K$  (cîștigul) și  $T$  (constanta de timp) sunt parametrii fizici ce trebuie identificați. Datele de intrare și ieșire măsurate,  $\mathcal{D} = \{u[n]\}_{n=1,N} \cup \{y[n]\}_{n=1,N}$ , sunt achiziționate cu o perioadă de eșantionare  $T_s$ .

Potrivit primei abordări, prin discretizarea funcției de transfer (128), se obține o nouă funcție de transfer de forma:

$$H_d(z) \stackrel{def}{=} \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}, \quad (129)$$

unde coeficienții  $a_1$ ,  $a_2$ ,  $b_1$  și  $b_2$  pot fi exprimați în funcție de  $K$  și  $T$ . Dependența coeficienților de parametrii fizici este neunică, fiind determinată de metoda de discretizare aleasă. Estimarea parametrilor fizici decurge din valorile identificate ale coeficienților. Din punctul de vedere al IS, (129) este considerat un model fie de tip ARX, fie de tip OE.

### Aspecte practice în Modelarea și Identificarea Sistemelor

A doua abordare presupune reprezentarea pe stare a motorului de curent continuu. O astfel de reprezentare (neunică) este următoarea:

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} \theta_1 & 0 \\ \alpha & 0 \end{bmatrix} x(t) + \begin{bmatrix} \theta_2 \\ 0 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 0 & \beta \end{bmatrix} x(t) \end{cases}, \quad \forall t \in \mathbb{R}_+, \quad (130)$$

unde  $\theta_1$  și  $\theta_2$  sunt parametri de stare necunoscuți care trebuie estimați, iar  $\alpha$  și  $\beta$  sunt 2 parametri constanți arbitrar aleși. Reprezentarea (130) corespunde formei canonice a funcției de transfer continue (128), adică:

$$H(s) = \frac{K/T}{s(s+1/T)}. \quad (131)$$

De asemenea, reprezentarea (130) este ideală, în sensul că nu include și perturbațiile interne și/sau externe care corup procesul. Parametrii fizici se pot deduce apoi din parametri de stare (vezi **Exercițiul 7.4**).

Estimarea parametrilor necunoscuți se poate efectua prin discretizarea reprezentării pe stare folosind perioada de eșantionare și aproximarea derivatei stărilor prin:

$$\dot{x}(t) \cong \frac{x(t+T_s) - x(t)}{T_s}, \quad \forall t \in \mathbb{R}_+, \quad (132)$$

care corespunde eșantionorului de ordin zero. Reprezentarea discretă pe stare corespunzătoare este exprimată astfel (inclusiv perturbațiile):

$$\begin{cases} x[n+1] = Ax[n] + Bu[n] + Fw[n], \\ y[n] = Cx[n] + Du[n] + Ev[n] \end{cases}, \quad \forall n \in \mathbb{N}, \quad (133)$$

unde  $n$  este timpul normalizat (discret), iar  $v$  și  $w$  sunt perturbații. Matricea  $A$  și vectorii  $B$ ,  $C$ ,  $D$ ,  $F$ ,  $E$  pot fi deduși direct din reprezentarea în timp continuu, cu aproximarea (132) și definiția:  $x[n] \stackrel{def}{=} x(nT_s)$ .

În ultima problemă de simulare, modelul motorului de curent continuu va fi utilizat pentru a genera 2 seturi de date intrare-ieșire: unul cu parametri fizici constanți și altul cu parametri fizici variabili. Parametrii constanți sunt:

$$K = K_0 = 4 \quad \text{și} \quad T = T_0 = 0.5. \quad (134)$$

Aceștia pot varia după următoarele legi, pe durata orizontului de măsură, de la momentul inițial arbitrar fixat la 0, pînă la momentul final  $T_{\max}$ :

$$K(t) = K_0 \left( 3 \frac{t^2}{T_{\max}^2} - 3 \frac{t}{T_{\max}} + 1 \right) \quad \text{și} \quad T = T_0 \left( 1 + \frac{1}{2} \sin \frac{10\pi t}{T_{\max}} \right), \quad \forall t \in [0, T_{\max}]. \quad (135)$$

În mediul de programare MATLAB, bibliotecile dedicate domeniilor de IS și TS sunt folosite pentru a proiecta programe ce simulează funcționarea sistemelor continue invariante la deplasări temporale ([LTI](#) – *Linear Time Invariant Systems*) și discretizarea acestora. Informații generale privind reprezentarea sistemelor continue se pot obține cu ajutorul comenzii `help ltimodels`. Astfel, există 4 tipuri de obiecte cu care pot fi reprezentate sistemele continue:

- *Funcții de transfer.* Obiectul este creat sau convertit cu ajutorul funcției `tf` (*transfer function*):

`H = tf(B,A,Ts) ;` sau `H = tf(SYS) ;`

unde: **B** este numărătorul funcției de transfer (vector, cu coeficienții aranjați în ordinea descrescătoare a puterilor variabilei complexe Laplace sau  $Z$ );

**A** este numitorul funcției de transfer (vector, cu coeficienții aranjați în ordinea descrescătoare a puterilor variabilei complexe  $s$  sau  $z$ );

**Ts** este perioada de eșantionare; dacă este specificată, perioada de eșantionare indică generarea unei funcții de transfer discrete; implicit, sau dacă **Ts=0**, funcția de transfer este continuă;

**H** este obiectul de tip **TF** corespunzător argumentelor de intrare; câmpurile sale specifice sunt următoarele:

**H.num** numărătorul funcției de transfer (câmp de celule vectoriale; în cazul modelelor SISO, numărătorul este stocat în **H.num{1}**);

**H.den** numitorul funcției de transfer (câmp de celule vectoriale; în cazul modelelor SISO, numitorul este stocat în **H.den{1}**);

**H.Variable** câmp care poate lua următoarele valori: '**s**', '**p**', '**z**', '**z<sup>-1</sup>**', '**q**'; arată tipul de sistem și maniera de reprezentare a funcției de transfer: Laplace, exprimată în variabilele  $s$  sau  $p$ ;  $Z$ , exprimată în variabilele  $z$  sau  $z^{-1}$ ; timp discret, exprimată în funcție de operatorul  $q^{+1}$ ; acest câmp poate fi setat ulterior de către utilizator.

**SYS** este un obiect de tip **ZPK** sau **SS**, care va fi convertit la un obiect de tip **TF**; conversia se poate realiza și cu ajutorul funcțiilor de bibliotecă IS: `zp2tf`, `ss2tf` (cu nume sugestive).

- *Poli-zero-uri-cîștig.* Obiectul este creat sau convertit cu ajutorul funcției `zpk` (*zero-pole-gain*):

`H = zpk(Z,P,K,Ts) ;` sau `H = zpk(SYS) ;`

unde: **Z** este vectorul zerourilor sistemului;

**P** este vectorul polilor sistemului;

**K** este cîștigul sistemului (scalar), obținut pentru  $s=0$  (în cazul continuu) sau  $z=1$  (în cazul discret);

## Aspecte practice în Modelarea și Identificarea Sistemelor

- Ts** este perioada de eșantionare; dacă este specificată, perioada de eșantionare indică generarea unei funcții de transfer discrete; implicit, sau dacă **Ts=0**, funcția de transfer este continuă;
- H** este obiectul de tip **ZPK** corespunzător argumentelor de intrare; câmpurile sale specifice sunt următoarele:
- H.z** zerourile funcției de transfer (câmp de celule vectoriale; în cazul modelelor SISO, zerourile sunt stocate în **H.z{1}**);
  - H.p** polii funcției de transfer (câmp de celule vectoriale; în cazul modelelor SISO, numitorul este stocat în **H.p{1}**);
  - H.k** câștigul funcției de transfer (matrice; în cazul modelelor SISO, numitorul este stocat în **H.k**);
  - H.Variable** câmp care poate lua următoarele valori: '**s**', '**p**', '**z**', '**z<sup>-1</sup>**', '**q**'; arată tipul de sistem și maniera de reprezentare a funcției de transfer: Laplace, exprimată în variabilele  $s$  sau  $p$ ; Z, exprimată în variabilele  $z$  sau  $z^{-1}$ ; timp discret, exprimată în funcție de operatorul  $q^{+1}$ ; acest câmp poate fi setat ulterior de către utilizator.
- SYS** este un obiect de tip **TF** sau **SS**, care va fi convertit la un obiect de tip **ZPK**; conversia se poate realiza și cu ajutorul funcțiilor de bibliotecă IS: **tf2zp**, **ss2zp** (cu nume sugestive).
- *Spațiul stărilor.* Obiectul este creat sau convertit cu ajutorul funcției **ss** (*state space*):

**H = ss(A,B,C,D,Ts) ; sau H = ss(SYS) ;**

- unde: **A** este matricea sistemului;
- B** este vectorul intrare-stare;
- C** este vectorul stare-ieșire;
- D** este factorul intrare-ieșire;
- Ts** este perioada de eșantionare; dacă este specificată, perioada de eșantionare indică generarea unei reprezentări discrete pe stare; implicit, sau dacă **Ts=0**, reprezentarea pe stare este continuă;
- H** este obiectul de tip **SS** corespunzător argumentelor de intrare; câmpurile sale specifice sunt următoarele:
- H.a** matricea sistemului;
  - H.b** vectorul intrare-stare;
  - H.c** vectorul stare-ieșire;
  - H.d** factorul intrare-ieșire;
- SYS** este un obiect de tip **TF** sau **ZPK**, care va fi convertit la un obiect de tip **SS**; conversia se poate realiza și cu ajutorul funcțiilor de bibliotecă IS: **tf2ss**, **zp2ss** (cu nume sugestive).

- *Răspuns în frecvență*. Obiectul este creat cu ajutorul funcției **frd** (*frequency response data*):

**H = frd(FR,omega,Ts) ;** sau **H = frd(SYS,omega) ;**

unde: **FR** este răspunsul în frecvență al sistemului (Transformata Fourier a funcției pondere);

**omega** este vectorul pulsațiilor/frecvențelor unde este evaluat răspunsul în frecvență ([rad/s] sau [Hz]);

**Ts** este perioada de eșantionare; dacă este specificată, perioada de eșantionare indică generarea unui răspuns în frecvență discret; implicit, sau dacă **Ts=0**, răspunsul în frecvență este continuu;

**H** este obiectul de tip **FRD** corespunzător argumentelor de intrare; câmpurile sale specifice sunt următoarele:

**H.Frequency** axa pulsațiilor;

**H.ResponseData** blocul matricial al răspunsului în frecvență; în cazul sistemelor SISO, răspunsul în frecvență este înregistrat în vectorul

**H.ResponseData(1,1,:)**;

**H.Units** unitatea de măsură a axei frecvențelor ('rad/s' sau 'Hz');

**SYS** este un obiect de tip **TF**, **ZPK** sau **SS**, care va fi convertit la un obiect de tip **FRD**.

În afara câmpurilor celor 4 obiecte de tip **LTI** descrise mai sus, există și câmpuri pe care le posedă toate obiectele **LTI**. Cel mai important este **H.Ts** (dacă **H** este un obiect **LTI**), care reprezintă perioada de eșantionare. În cazul sistemelor continue, acest câmp este vid sau are valoarea nulă.

Biblioteca IS este dotată cu o serie de rutine care permit trecerea de la reprezentările continue la cele discrete și reciproc. Cele mai importante sunt: **c2d**, **d2c** și **d2d**, descrise în continuare.

#### # C2D

- Apel: **[Hd,G] = c2d(H,Ts,met) ;**
- Convertește un obiect **LTI** exprimat în timp continuu într-un obiect LTI exprimat în timp discret. Argumentele funcției sunt următoarele:
- H** obiect **LTI** (de regulă de tip **TF**) exprimat în timp continuu;
- Ts** perioada de eșantionare;
- met** metoda de extrapolare, care, în principal, poate fi: '**zoh**' (*zero order holder* – extrapolator de ordin zero, implicit) sau '**foh**' (*first order holder* – extrapolator de ordin unu);
- Hd** obiect **LTI** (de regulă de tip **TF**) exprimat în timp discret;
- G** matricea de transformare a condițiilor inițiale la trecerea din continuu în discret, pentru inițializări nebanale.

#### # D2C

- Apel: **H = d2c(Hd,met) ;**
- Convertește un obiect **LTI** exprimat în timp discret într-un obiect LTI exprimat în timp continuu. Argumentele funcției sunt următoarele:

## Aspecte practice în Modelarea și Identificarea Sistemelor

- Hd** obiect **LTI** (de regulă de tip **TF**) exprimat în timp discret;
- met** metoda de extrapolare, care implicit este **'zoh'** (*zero order holder* – extrapolator de ordin zero);
- H** obiect **LTI** (de regulă de tip **TF**) exprimat în timp continuu.

### # D2D

- Apel: **Hd2 = d2d(Hd1,Ts) ;**
- Re-eșantionează un obiect **LTI** exprimat în timp discret. Argumentele funcției sunt următoarele:
  - Hd1** obiect **LTI** (de regulă de tip **TF**) exprimat în timp discret, la o anumită perioadă de eșantionare;
  - Ts** perioada de re-eșantionare;
  - Hd2** obiect **LTI** (de regulă de tip **TF**) exprimat în timp discret, la o nouă perioadă de eșantionare.

O rutină interesantă care realizează legătura dintre bibliotecile de TS și IS este **idss**. Aceasta crează un model de tip reprezentare pe stare, plecând de la descrierea explicită a matricilor corespunzătoare. Reprezentarea pe stare este similară cu (133), dar, pentru **idss**, zgomotele  $v$  și  $w$  sunt identice (notate cu  $e$ ),  $E$  este nul, iar  $F$  este notat cu  $K$  (deși nu este câștigul funcției de transfer asociate). Apelul tipic al rutinei este următorul:

**M = idss(A,B,C,D,K,X0,Ts) ;**

unde: **A, ..., K** sunt matricile reprezentării pe stare dorite;

- X0** este starea inițială (implicit nulă);
- Ts** este perioada de eșantionare; dacă este specificată, perioada de eșantionare indică o reprezentare pe stare discretă; altfel, dacă **Ts=0**, se construiește o reprezentare pe stare continuă; implicit, **Ts=1** (reprezentare discretă pe stare);
- M** este un obiect de tip **IDSS**, care corespunde reprezentării pe stare dorite; câmpurile sale sunt următoarele:

```
A: 'A-matrix (Nx-by-Nx matrix)'  
B: 'B-matrix (Nx-by-Nu matrix)'  
C: 'C-matrix (Ny-by-Nx matrix)'  
D: 'D-matrix (Nu-by-Ny matrix)'  
K: 'K-matrix (Nx-by-Ny matrix)'  
X0: 'initial state (Nx-length vector)'  
dA: 'standard deviation of A-matrix'  
dB: 'standard deviation of B-matrix'  
dC: 'standard deviation of C-matrix'  
dD: 'standard deviation of D-matrix'  
dK: 'standard deviation of K-matrix'  
dX0: 'standard deviation of X0'  
SSParameterization: 'type of parametrization'  
{'Free', 'Canonical', 'Structured'}
```

```

    As: 'free parameters in A (set by NaN) '
    Bs: 'free parameters in B (set by NaN) '
    Cs: 'free parameters in C (set by NaN) '
    Ds: 'free parameters in D (set by NaN) '
    Ks: 'free parameters in K (set by NaN) '
    X0s: 'free parameters in X0 (set by NaN) '
    StateName: [1x53 char]
    InitialState: [1x50 char]
    nk: [1x53 char]
    DisturbanceModel: [1x32 char]
    CanonicalIndices: 'Row vector or 'Auto''
    Name: 'string'
    Ts: 'sampling period'
    InputName: 'Nu-by-1 cell array of strings'
    InputUnit: 'Nu-by-1 cell array of strings'
    OutputName: 'Ny-by-1 cell array of strings'
    OutputUnit: 'Ny-by-1 cell array of strings'
    TimeUnit: 'string'
    ParameterVector: 'Np-by-1 vector'
    PName: 'Np-by-1 cell array of strings'
    CovarianceMatrix: 'Np-by-Np matrix'
    NoiseVariance: 'Ny-by-Ny matrix'
    InputDelay: 'Nu-by-1 vector'
    Algorithm: [1x38 char]
    EstimationInfo: [1x39 char]
    Notes: [1x30 char]
    UserData: 'Arbitrary'

```

Cîmpurile principale ale obiectului **M** sunt: **M.A**, ..., **M.K**, **M.As**, ..., **M.Ks**, și **M.Ts**. Primele cîmpuri (**M.A**, ..., **M.K**) includ numai valori numerice finite. Spre deosebire de acestea, cîmpurile **M.As**, ..., **M.Ks** pot conține și valori de tipul **NaN**, după cum indică **M.SSPparameterization**. Orice valoare **NaN** arată că acel parametru este liber sau necunoscut. Astfel:

- **M.SSPparameterization='Free'** (implicit) indică faptul că toți parametrii matricilor **M.As**, **M.Bs**, **M.Cs** sunt setați cu **NaN**, adică toți parametrii matricilor **M.A**, **M.B**, **M.C** sunt liberi sau necunoscuți; parametrii liberi/necunoscuți din **M.D**, **M.K** și **M.X0** sunt determinați de cîmpurile: **M.nk**, **M.DisturbanceModel** și **M.InitialState**.
- **M.SSPparameterization='Canonical'** indică faptul că matricile **M.A**, **M.B** și **M.C** sunt exprimate în forma canonică de observabilitate, determinată de **M.CanonicalIndices**; parametrii liberi/necunoscuți din **M.D**, **M.K** și **M.X0**



sunt determinați de câmpurile: **M.nk**, **M.DisturbanceModel** și **M.InitialState**.

- **M.SSPparameterization='Structured'** indică faptul că parametrii liberi/necunoscuți ai câmpurilor **M.A**, ..., **M.X0** sunt indicați numai de pozițiile valorilor **NaN** din cadrul câmpurilor **M.As**, ..., **M.X0s**, stabilite de către utilizator.

De exemplu, pentru a construi reprezentarea pe stare (130) (în timp continuu), se poate executa următoarea secvență de cod:

```
>> A = [-1 0 ; 1 0] ;
>> B = [1 ; 0] ;
>> C = [0 1] ;
>> D = 0 ;
>> K = [0 ; 0] ;
>> X0 = [0 ; 0] ;
>> M = idss(A,B,C,D,K,X0,0) ;
>> M.As = [NaN 0 ; 1 0] ;      % Parameter: theta1.
>> M.Bs = [NaN ; 0] ;          % Parameter: theta2.
>> M.Cs = C ;
>> M.Ds = D ;
>> M.Ks = K ;
>> M.X0 = X0 ;
```

Mai multe informații relative la obiectele de tip **IDSS** se pot obține cu comanda **idprops idss**.

Un model reprezentat pe stare în timp continuu poate fi direct identificat folosind funcțiile **pem** (off-line) sau **rpem** (on-line), ca în exemplul de mai jos:

```
>> M_est = pem(D,M) ;
```

unde **D** este un obiect de tip **IDDATA** (vezi **Problema 2.3**) corespunzător datelor generate de procesul corespunzător reprezentării pe stare, iar **M** este reprezentarea pe stare de mai sus. Deși reprezentarea este în timp continuu, funcția **pem** efectuează o discretizare în vederea identificării, preluând perioada de eșantionare din obiectul **D**.

Pentru problema de simulare din finalul acestui capitol, se recomandă proiectarea unei rutine de generare a datelor care să ofere valori obținute plecând de la funcția de transfer (128) cu parametri fizici constanți (134) sau variabili (135). Rutina ar putea avea următorul apel tipic:

```
[D,V,P] = gdata_fp(cv,K0,T0,Tmax,Ts,U,lambda) ;
```

unde: **cv** este un comutator care arată tipul de proces: cu parametri constanți dați de ecuațiile (134) (**cv=0**) sau cu parametri variabili dați de ecuațiile (135) (**cv~=0**); (implicit: **cv=0**);  
**K0** este valoarea constantă a câștigului (implicit: **K0=4**);  
**T0** este valoarea constantei de timp (implicit: **T0=0.5** [secunde]);  
**Tmax** este durata simulării (implicit: **Tmax=80** [secunde]);  
**Ts** este perioada de eșantionare (implicit: **Ts=0.1** [secunde]);

## 7. Aplicații de identificare recursivă

- U** este amplitudinea intrării; semnalul de intrare va fi o formă de undă dreptunghiulară bipolară (adică avînd numai valori pozitive și negative), cu perioada egală cu  $1/7$  din durata totală a simulării; (implicit: **U=0.5**);
- lambda** este varianța zgomotului alb care corupe datele măsurate la ieșirea din proces (implicit: **lambda=1**);
- D** este obiectul de tip **IDDATA** (vezi **Problema 2.3**) corespunzător datelor generate (intrarea se regăsește în **D.u**, iar ieșirea în **D.y**);
- V** este obiectul de tip **IDDATA** corespunzător zgomotului generat (zgomotul alb se regăsește în **V.u** sau în **V.y**);
- P** este obiectul de tip **TF** corespunzător modelului de proces furnizor de date; în cazul parametrilor constanți (134): **P.num{1}** și **P.den{1}** sunt numărătorul, respectiv numitorul funcției de transfer continue (128); în cazul parametrilor variabili (135): **P.num{1}** este vectorul valorilor cîștigului  $K$ , iar **P.den{1}** este vectorul valorilor constantei de timp  $T$  pe durata simulării.

Rutina poate folosi funcția de bibliotecă TS **lsim** pentru simularea unui sistem continuu cu parametri constanți. (Cititorul este invitat să se informeze singur asupra acestei rutine.) Pentru simularea sistemului cu parametri variabili, se recomandă discretizarea funcției de transfer la fiecare pas de eșantionare și evaluarea ieșirii folosind ecuația cu diferențe asociată funcției de transfer discrete (129). Zgomotul va fi adăugat numai ieșirii, în final.

Se vor genera 2 seturi de date intrare-ieșire eșantionate: unul provenit de la procesul cu parametri constanți ( $\mathcal{D}_c$ ) și altul – de la procesul cu parametri variabili ( $\mathcal{D}_v$ ).

Obiectivul acestui capitol este de a analiza performanțele metodelor de identificare în cele 2 aplicații descrise mai sus.

### 7.2. Exerciții

#### Exercițiul 7.1

Arătați legătura care există între definițiile (116) și (117).

#### Exercițiul 7.2

Arătați că transformarea omografică de discretizare (123) asociază dreptele verticale ale planului complex cu cercuri.

#### Exercițiul 7.3

Folosind aproximarea Padé (125), se constată că elementului integrator continuu (adică avînd funcția de transfer  $1/s$ ) îi corespunde următorul sistem discret(izat):

$$H_d(z) = \frac{T_s z^{-1}}{1 - z^{-1}}. \quad (136)$$

### Aspecte practice în Modelarea și Identificarea Sistemelor

- a. Utilizați aproximația Padé pentru a deduce corespondentul discret al sistemului continuu de ordin 1:

$$H(s) = \frac{K}{1 + Ts}, \quad (137)$$

unde  $K$  este câștigul iar  $T$  este constanta de timp ce caracterizează sistemul.

- b. Folosind corespondența dintre aproximația Padé (125) și transformarea ideală de discretizare (124), arătați că o mai bună exprimare a sistemului discret obținut la punctul precedent este următoarea:

$$H_d(z) = \frac{b_1 z^{-1}}{1 + a_1 z^{-1}}, \quad \text{cu: } \begin{cases} a_1 = -e^{-T_s/T} \\ b_1 = K(1 - e^{-T_s/T}) \end{cases}. \quad (138)$$

Arătați că valorile coeficienților funcției de transfer discrete (138) se pot obține și folosind Teorema Reziduurilor (adică relația (126)).

- c. Evaluați funcția de transfer a sistemului discretizat corespunzător sistemului continuu descris de funcția de transfer (128) (sistem liniar simplificat asociat unui motor electric de curent continuu). Pentru aceasta, se vor utiliza mai întâi separat transformările (123) și (125), apoi Teorema Reziduurilor (126). Din punctul de vedere al IS, care ar fi dezavantajul utilizării transformării (123)? Arătați că este posibilă folosirea corespondenței dintre aproximația Padé și transformarea ideală de discretizare (ca în cazul punctului precedent) pentru a deduce o formă îmbunătățită a funcției de transfer discrete. Pentru aceasta, se vor deduce coeficienții  $a_1$ ,  $a_2$ ,  $b_1$  și  $b_2$  în funcție de constantele fizice  $K$  și  $T$ . (Se recomandă descompunerea funcției de transfer în fracții simple.) Arătați că:

$$a_1 + a_2 = -1 \quad (139)$$

și că parametrii fizici ai procesului se pot recupera direct din parametrii estimați ai funcției de transfer discrete cu ajutorul următoarelor relații:

$$K = \frac{b_1 + b_2}{T_s(1 - a_2)}; \quad T = T_s \frac{a_2 b_1 + b_2}{(1 - a_2)(b_1 + b_2)} = -\frac{T_s}{\ln a_2}. \quad (140)$$

Care dintre cele 2 relații de evaluare a constantei de timp  $T$  ar trebui aleasă în implementarea pe un mijloc automat de calcul? Justificați răspunsul.

- d. Fie sistemul continuu de ordin 2, descris de următoarea funcție de transfer:

$$H(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}, \quad (141)$$

unde  $\omega_0$  este pulsația naturală de oscilație, iar  $\zeta$  este factorul de amortizare.

Dacă  $\zeta \geq 1$ , polii funcției de transfer sunt reali. Dacă  $\zeta < 1$ , polii nu sunt reali.

Folosiți Teorema Reziduurilor (în cazul extrapolatorului de ordin zero) pentru a deduce versiunea discretizată a sistemului. Mai precis, arătați că:

$$H_d(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}, \quad (142)$$

unde:

$$\begin{bmatrix} a_1 = -2\alpha\beta \\ a_2 = \alpha^2 \end{bmatrix}, \quad \begin{bmatrix} b_1 = 1 - \alpha \left( \beta + \frac{\omega_0}{\omega} \zeta \gamma \right) \\ b_2 = \alpha^2 + \alpha \left( \beta + \frac{\omega_0}{\omega} \zeta \gamma \right) \end{bmatrix}, \quad \begin{cases} \stackrel{def}{\omega} = \omega_0 \sqrt{1 - \zeta^2} \\ \stackrel{def}{\alpha} = e^{-\zeta \omega_0 T_s} \\ \stackrel{def}{\beta} = \cos \omega T_s \\ \stackrel{def}{\gamma} = \sin \omega T_s \end{cases}, \text{ pentru } \zeta < 1; \quad (143)$$

$$\begin{bmatrix} a_1 = -2\alpha\beta \\ a_2 = \alpha^2 \end{bmatrix}, \quad \begin{bmatrix} b_1 = 1 - \alpha \left( \beta + \frac{\omega_0}{\omega} \zeta \gamma \right) \\ b_2 = \alpha^2 + \alpha \left( \beta + \frac{\omega_0}{\omega} \zeta \gamma \right) \end{bmatrix}, \quad \begin{cases} \stackrel{def}{\omega} = \omega_0 \sqrt{\zeta^2 - 1} \\ \stackrel{def}{\alpha} = e^{-\zeta \omega_0 T_s} \\ \stackrel{def}{\beta} = \text{ch } \omega T_s \\ \stackrel{def}{\gamma} = \text{sh } \omega T_s \end{cases}, \text{ pentru } \zeta > 1; \quad (144)$$

$$\begin{bmatrix} a_1 = -2e^{-\omega_0 T_s} \\ a_2 = e^{-2\omega_0 T_s} \end{bmatrix}, \quad \begin{bmatrix} b_1 = 1 + (T_s - 1)e^{-\omega_0 T_s} \\ b_2 = e^{-\omega_0 T_s} (e^{-\omega_0 T_s} - T_s - 1) \end{bmatrix}, \text{ pentru } \zeta = 1. \quad (145)$$

Dacă în urma unui experiment de identificare ar fi estimați coeficienții funcției de transfer discrete (142), arătați cum se pot determina parametri fizici  $\omega_0$  și  $\zeta$  din relațiile (143)-(145).

e. Reluați exercițiul de la punctul precedent pentru sistemul de ordin 2:

$$H(s) = \frac{\omega_0^2 s}{s^2 + 2\zeta \omega_0 s + \omega_0^2}, \quad (146)$$

adică arătați că parametri funcției de transfer discrete asociate sunt:

$$\begin{bmatrix} a_1 = -2\alpha\beta \\ a_2 = \alpha^2 \end{bmatrix}, \quad \begin{bmatrix} b_1 = \frac{\omega_0^2}{\omega} \alpha \gamma \\ b_2 = -b_1 \end{bmatrix}, \quad \begin{cases} \stackrel{def}{\omega} = \omega_0 \sqrt{1 - \zeta^2} \\ \stackrel{def}{\alpha} = e^{-\zeta \omega_0 T_s} \\ \stackrel{def}{\beta} = \cos \omega T_s \\ \stackrel{def}{\gamma} = \sin \omega T_s \end{cases}, \text{ pentru } \zeta < 1; \quad (147)$$

$$\begin{bmatrix} a_1 = -2\alpha\beta \\ a_2 = \alpha^2 \end{bmatrix}, \begin{bmatrix} b_1 = \frac{\omega_0^2}{\omega} \alpha \gamma \\ b_2 = -b_1 \end{bmatrix}, \begin{cases} \omega = \omega_0 \sqrt{1 - \zeta^2} \\ \alpha = e^{-\zeta \omega_0 T_s} \\ \beta = \text{ch } \omega T_s \\ \gamma = \text{sh } \omega T_s \end{cases}, \text{ pentru } \zeta < 1; \quad (148)$$

$$\begin{bmatrix} a_1 = -2e^{-\omega_0 T_s} \\ a_2 = e^{-2\omega_0 T_s} \end{bmatrix}, \quad b_1 = -b_2 = \omega_0^2 T_s e^{-\omega_0 T_s}, \text{ pentru } \zeta = 1. \quad (149)$$

#### Notă privind discretizarea sistemelor continue

- Acest exercițiu sugerează maniera practică în care se efectuează discretizarea unei funcții de transfer raționale continue fără poli multipli, cazul utilizării extrapolatorului de ordin zero. Mai întâi se descompune funcția de transfer continuă în fracții simple (de ordin 1 sau 2). Frațiile de ordin 1 sunt asociate fie integratoarelor, fie sistemelor de ordin 1. Integratoarele se discretizează cu ajutorul aproximației Padé (125). Sistemele de ordin 1 se discretizează cu ajutorul relațiilor (129), care sunt mai precise. Sistemele de ordin 2 se discretizează cu ajutorul relațiilor (143)-(145) sau (147)-(149).

#### Exercițiul 7.4

Deduceți valorile parametrilor necunoscuți  $\theta_1$  și  $\theta_2$  din reprezentarea pe stare (130) în funcție de parametri fizici  $K$  și  $T$  din definiția (128). Arătați că:

$$T = -\frac{1}{\theta_1} \quad \text{și} \quad K = -\frac{\alpha\beta\theta_2}{\theta_1}. \quad (150)$$

### 7.3. Probleme de simulare

#### Problema 7.1

Mini-simulatorul **ISLAB 7A** implementează primele 4 etape de identificare descrise în finalul paragrafului A din secțiunea 7.1. Pentru aceasta, au fost utilizate o serie de rutine MATLAB de uz general sau din biblioteca dedicată domeniului IS (în afara rutinei **butter** deja descrise), precum și unele special proiectate de către autori. Practic, mini-simulatorul efectuează o identificare off-line a unui model ARMAX, plecând de la modelul general (84) & (124). După afișarea graficului caracteristicii aproximative în frecvență a procesului generator de date (Figura 34), performanțele modelului sunt ilustrate în următoarele grafice: caracteristicile în frecvență ale filtrului Butterworth ales (Figura 35); spectrul real și cel simulat al ieșirii, împreună cu eroarea spectrală pe întreaga bandă de pulsații normalizate (Figura 36); spectrul real și cel simulat al ieșirii, împreună cu eroarea spectrală pe banda de trecere a filtrului Butterworth (Figura 37).

## 7. Aplicații de identificare recursivă

- a. Să se analizeze cu atenție listingul mini-simulatorului **ISLAB\_7A** și să se enumere rutinele apelate. Apoi, să se determine ce reprezintă fiecare rutină, folosind eventual comanda **help** din mediul MATLAB.
- b. Să se ruleze mini-simulatorul **ISLAB\_7A** alegînd succesiv cîteva frecvențe de interes din spectrul procesului furnizor de date. Pentru fiecare frecvență, să se aleagă întîi un filtru Butterworth de tip trece-bandă (cu o lărgime de bandă corespunzătoare) și apoi un filtru Butterworth de tip trece-jos (cu o frecvență de tăiere corespunzătoare). Comparați performanțele celor 2 alegeri pentru fiecare frecvență aleasă. Observați dacă performanțele diferă mai mult pentru frecvențe înalte decît pentru frecvențe joase și motivați acest fenomen.

### Problema 7.2

Mini-simulatorul **ISLAB\_7A** returnează un set de date destinat identificării recursive a parametrilor (obiect de tip **IDDATA**), filtrul de intrare utilizat cu pulsația (sau pulsațiile) de tăiere (structură specifică) și modelul ARMAX determinat folosind MMEP în variantă off-line (obiect **IDMODEL**). Proiectați mini-simulatorul **ISLAB\_7B** care să efectueze o identificare on-line a modelului ARMAX folosind datele furnizate, dar oferind posibilitatea utilizatorului să aleagă alți indici structurali decît cei propuși de modelul off-line, dacă o dorește. Dacă se aleg aceiași indici structurali, este recomandabil să se utilizeze chiar modelul ARMAX off-line ca inițializare pentru procedura on-line. Performanțele modelului identificat vor fi afișate ca în cazul mini-simulatorului **ISLAB\_7A**. Testați mini-simulatorul pentru diferite frecvențe din banda procesului, ca în problema precedentă (se vor alege aceiași indici structurali pentru o frecvență selectată, indiferent de tipul de filtru). Comentați rezultatele obținute și efectuați o comparație cu cele din problema precedentă.

### Problema 7.3

În cadrul acestei probleme, este propusă proiectarea a 4 mini-simulatoare de identificare a parametrilor fizici: **ISLAB\_7C**, **ISLAB\_7D**, **ISLAB\_7E** și **ISLAB\_7F**. Acestea operează cu seturile de date  $\mathcal{D}_c$  (**ISLAB\_7C** și **ISLAB\_7D**), respectiv  $\mathcal{D}_v$  (**ISLAB\_7E** și **ISLAB\_7F**). Datele pot fi generate folosind rutina **gdata\_fp** descrisă în paragraful B al secțiunii 7.1. După generarea datelor, se efectuează următoarele experimente de identificare:

- Identificare off-line a parametrilor fizici constanți (cu funcțiile **oe** sau **arx**), folosind tehnica discretizării funcției de transfer continue și modelul de tip OE sau ARX (**ISLAB\_7C**).
- Identificare off-line a parametrilor fizici constanți (cu funcția **pem**), folosind tehnica reprezentării pe stare (**ISLAB\_7D**).
- Identificare on-line a parametrilor fizici variabili (cu funcțiile **roe** sau **rarx**), folosind tehnica discretizării funcției de transfer continue și modelul de tip OE sau ARX (**ISLAB\_7E**).
- Identificare on-line a parametrilor fizici variabili (cu funcția **rpem**), folosind tehnica reprezentării pe stare (**ISLAB\_7F**).

## Aspecte practice în Modelarea și Identificarea Sistemelor

---

Toate cele 4 mini-simulatoare vor afișa grafic, în aceeași fereastră, variațiile intrării și ieșirii procesului, ca în **Figurile 38**. A doua figură afișează variația ieșirii simulate folosind modelul în timp discret și a celei măsurate folosind modelul în timp continuu (vezi **Figurile 39**).

În cazul parametrilor fizici constanți, se vor afișa valorile adevărate și cele estimate acestora ca mai jos:

**<ISLAB\_7C>: Physical parameters:**

	<b>True</b>	<b>Estimated</b>
<b>K:</b>	<b>4.0000</b>	<b>4.0597</b>
<b>T:</b>	<b>0.5000</b>	<b>0.4412</b>

În cazul parametrilor fizici variabili, se va ilustra variația acestora împreună cu valorile lor estimate, ca în **Figura 40**.

Pentru a proiecta mini-simulatoarele, este util să se rezolve mai întâi exercițiile de gândire propuse. În cazul reprezentării pe stare, se poate ține cont de observațiile din paragraful B al secțiunii 7.1.

- În urma simulărilor, se constată că modelele de tip OE sunt superioare celor de tip ARX din punctul de vedere al preciziei de identificare a parametrilor necunoscuți. Care credeți că este explicația acestui fapt?
- Reprezentarea pe stare ar trebui să conducă la estimări mai precise ale parametrilor fizici. Justificați această afirmație. Este ea verificată în urma simulărilor?
- Variați constanta de timp  $T_0$  și observați cum este influențată precizia de estimare a parametrilor fizici dar și SNR caracteristic datelor de ieșire măsurate. Comentați rezultatele obținute.
- Variați câștigul  $K_0$  și comentați precizia cu care sunt estimați parametrii fizici în cele 4 mini-simulatoare.
- Variați perioada de eșantionare și comentați rezultatele obținute.

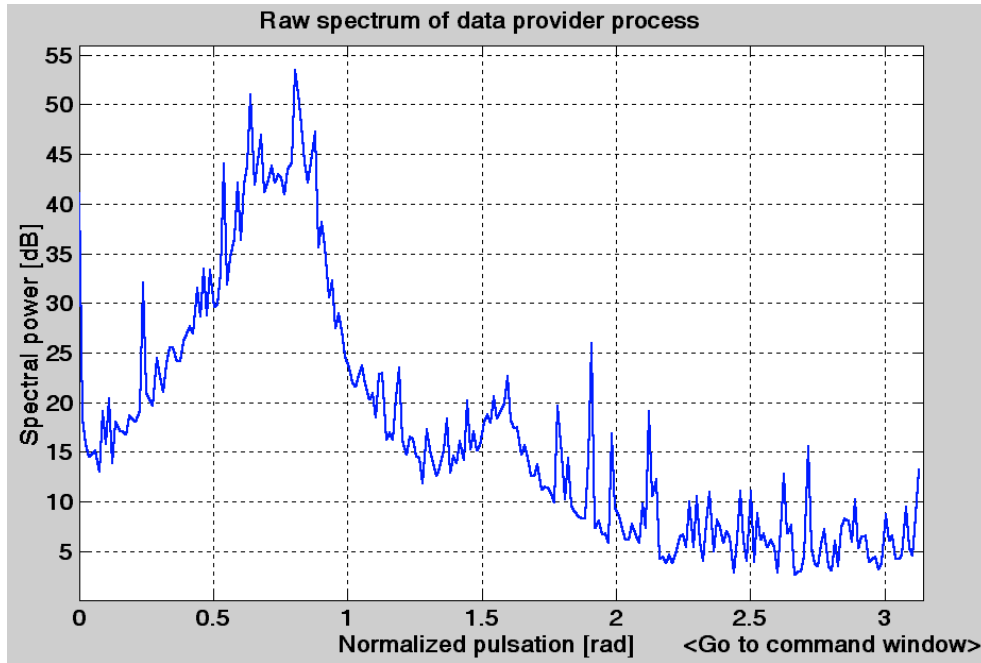


Figura 34. O estimare grosieră a spectrului procesului furnizor de date.

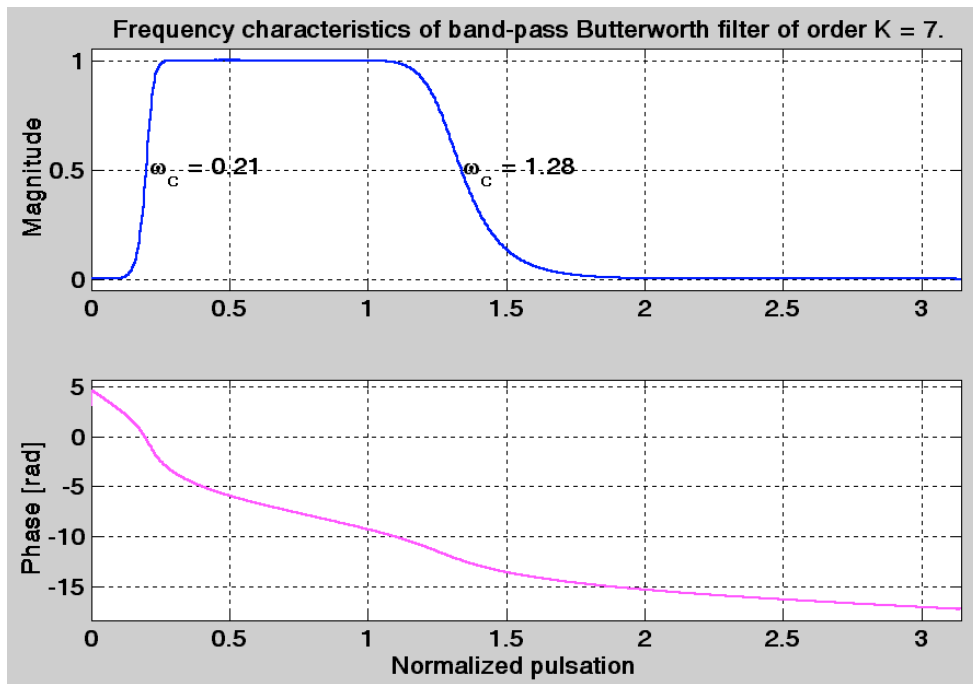


Figura 35. Caracteristicile filtrului Butterworth ales.



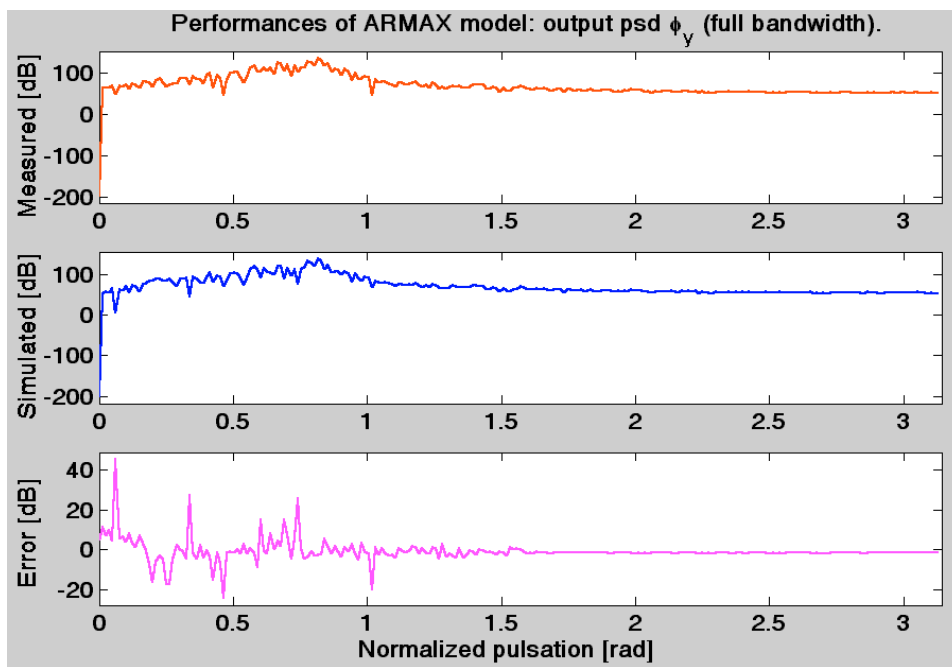


Figura 36. Performanțele modelului ARMAX pe toată lărgimea de bandă.

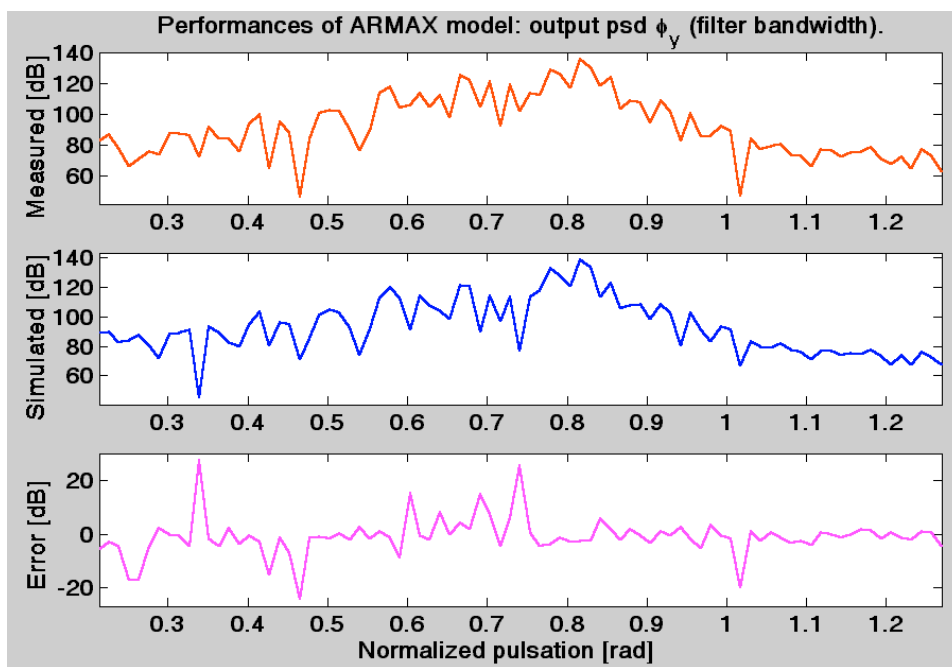


Figura 37. Performanțele modelului ARMAX pe lărgimea de bandă a filtrului.

## 7. Aplicații de identificare recursivă

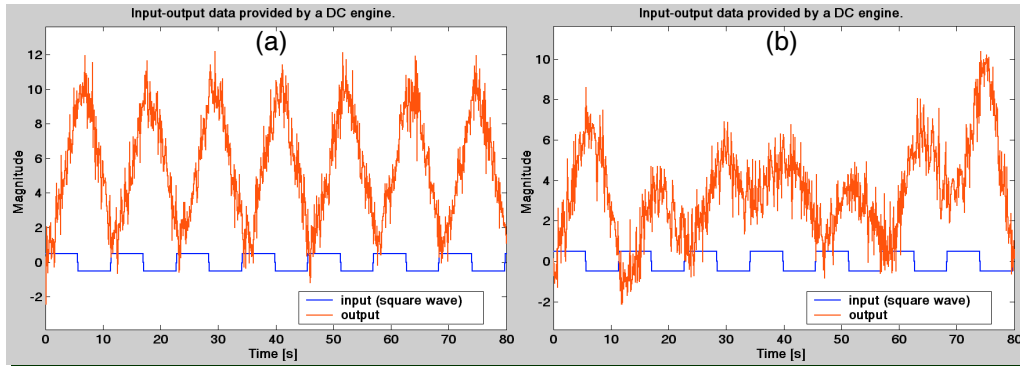


Figura 38. Date de intrare-ieșire furnizate de un motor de curent continuu: (a) cu parametri constanți; (b) cu parametri variabili.

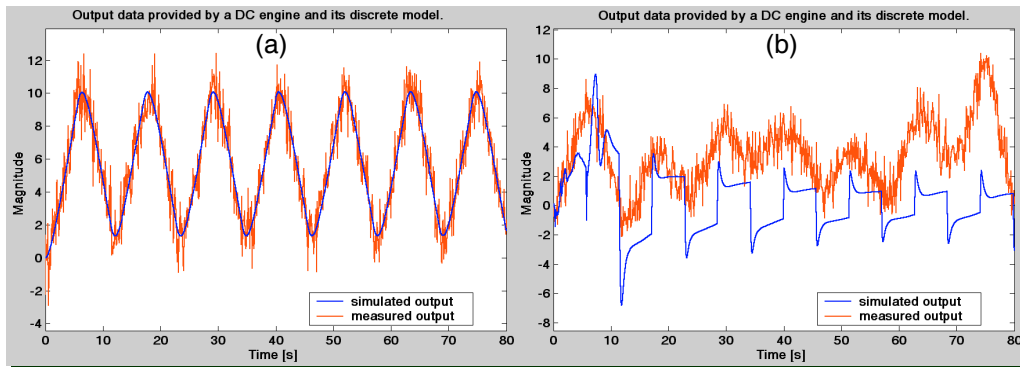


Figura 39. Ieșirea măsurată și cea simulată ale motorului de curent continuu: (a) cu parametri constanți; (b) cu parametri variabili.

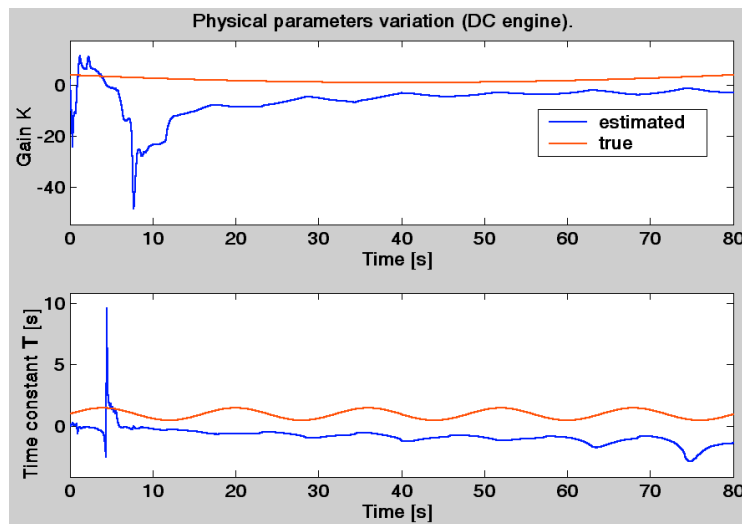


Figura 40. Urmărirea parametrilor fizici ai motorului de curent continuu.

## Capitolul 8

### Modelarea și predicția seriilor de timp

#### 8.1. Contextul general de lucru

Conceptul de *serie de timp* [TeSt85], [StD9603] desemnează un șir de date înregistrat în urma evoluției unui proces, fără a putea cuantifica sau fără a cunoaște cauzele acelei evoluții. Doar ieșirea procesului este monitorizată.

Eșantioanele unei serii de timp sunt achiziționate la momente uniforme sau ne-uniforme de eșantionare. În primul caz, seria de timp este notată prin  $\mathcal{D} = \{y[n] = y(nT_s)\}_{n=1, \overline{N}}$ , unde  $T_s$  este perioada de eșantionare aleasă. În al doilea caz, seria de timp este  $\mathcal{D} = \{y(t_n)\}_{n=1, \overline{N}}$ , unde  $t_1 < t_2 < \dots < t_n \dots < t_N$  sunt momentele de eșantionare considerate. Alegerea unei perioade de eșantionare minime (notată tot cu  $T_s$ ) este necesară și în cazul neuniform (eventual, ea este egală cu durata minimă dintre 2 momente de eșantionare consecutive). Practic, mulțimea momentelor de eșantionare neuniforme poate fi considerată un subșir al șirului  $\{nT_s\}_{n \geq 0}$ , astfel că:

$$t_n \geq nT_s, \quad \forall n \in \mathbb{N}. \quad (151)$$

Orizontul de observabilitate al seriei de timp este întotdeauna finit:  $T_{\max} = NT_s$  sau  $T_{\max} = t_N$ , unde  $N \in \mathbb{N}^*$  este numărul de date măsurate. Obiectivul principal al modelării unei serii de timp este predicția comportamentului procesului furnizor de date dincolo de orizontul de măsură. De regulă, această operație se efectuează pe un orizont de predicție finit, la diferite momente de eșantionare consecutive echidistante  $((N+1)T_s, \dots, (N+K)T_s$ , dacă perioada de eșantionare este constantă) sau neuniforme  $(t_{N+1}, \dots, t_{N+K}$ , dacă perioada de eșantionare este variabilă). De aceea, seriile de timp trebuie să fie *consistente*, adică să conțină un număr suficient de mare de date măsurate ( $N$  este cel puțin de ordinul zecilor). Orizontul de predicție are o dimensiune  $K \in \mathbb{N}^*$  mult mai mică (maxim 3 momente de timp), datorită dispersiei erorii de predicție care, de regulă, crește exponențial. Diminuarea dispersiei erorii de predicție se poate realiza numai cu reactualizarea modelului seriei de timp în funcție de noile date măsurate.

Modelul unei serii de timp ( $y_{\mathcal{M}}$ ) include 3 componente aditive: două de tip determinist (tendința polinomială  $y_T$ , variația sezonieră  $y_S$ ) și una de tip nedeterminist (modelul AR al zgomotului care afectează datele măsurate,  $y_{AR}$ ):

$$y_{\mathcal{M}} \equiv y_T + y_S + y_{AR}. \quad (152)$$

Estimarea celor 3 modele matematice din (152) se bazează pe MCMMP. Algoritmii efectivi de estimare au fost descriși pe larg în [StD9603] sau se pot deduce din [TeSt85]. Vom prezenta succint în continuare numai principalele etape ale acestora.

### A. Estimarea modelului polinomial al tendinței

Tendința unei serii de timp modelează orientarea sa generală de-a lungul timpului, fără a lua în considerare (pe cât posibil) variațiile periodice ale datelor și zgomotele care le afectează. Media datelor măsurate constituie, de exemplu, un model grosier al tendinței acestora. Dreapta de regresie liniară îmbunătățește aproximarea. Expresia generală a componentei tendință este de tip polinomial:

$$y_T(t) = a_0 + a_1 t + \dots + a_p t^p, \quad \forall t \in \mathbb{R}, \quad (153)$$

unde  $p \in \mathbb{N}$  este gradul polinomului, necunoscut. Tot necunoscuți sunt și coeficienții acestuia,  $\{a_i\}_{i \in \overline{0,p}}$ .

Pentru date eșantionate uniform, modelul (153) se poate particulariza în:

$$y_T(nT_s) = a_0 + a_1 nT_s + \dots + a_p n^p T_s^p, \quad \forall n \in \mathbb{Z}, \quad (154)$$

dacă perioada de eșantionare este cunoscută, sau, mai simplu, în:

$$y_T[n] = a_0 + a_1 n + \dots + a_p n^p, \quad \forall n \in \mathbb{Z}, \quad (155)$$

dacă perioada de eșantionare nu se cunoaște cu precizie sau este considerată unitatea de măsură a timpului.

Eșantionarea neuniformă induce utilizarea modelului în timp continuu (153), ale cărui valori calculate în momentele de eșantionare  $(\{y_T(t_n)\}_{n \in \overline{1,N}})$  sunt puse în corespondență cu datele seriei de timp.

Pentru generalitate, se notează cu  $t_n$  momentul generic de eșantionare, indiferent de maniera de eșantionare (uniformă sau neuniformă). Evident, în cazul eșantionării uniforme,  $t_n = nT_s$  sau chiar  $t_n = n$ . Altfel, are loc inegalitatea (151). În aceste condiții, pentru estimarea modelului (153), trebuie rezolvată următoarea problemă de minimizare pătratică (exprimată în funcție de eroarea dintre proces și model, sau de eroarea de predicție cu un pas):

$$\hat{\theta}_N = \arg \min_{\theta \in \mathcal{S}} \mathcal{V}_N(\theta), \quad \text{unde: } \mathcal{V}_N(\theta) \stackrel{\text{def}}{=} \sum_{n=1}^N (y[n] - y_T(t_n))^2, \quad \forall \theta \in \mathcal{S}, \quad (156)$$

iar  $\mathcal{S} \subseteq \mathbb{R}^{p+1}$  este domeniul de stabilitate al modelului matematic. Evident, prin  $\theta \in \mathbb{R}^{p+1}$  a fost notat vectorul coeficienților necunoscuți  $\{a_i\}_{i \in \overline{0,p}}$ , în ordinea crescătoare a indicilor.

Aplicînd MCMMP pentru rezolvarea problemei (156), se obține:

$$\hat{\theta}_N = R_N^{-1} r_N, \quad (157)$$

unde:

$$R_N \stackrel{\text{def}}{=} \left[ \frac{1}{N} \sum_{n=1}^N t_n^{i+j} \right]_{i,j \in \overline{0,p}} \quad \text{și} \quad r_N \stackrel{\text{def}}{=} \left[ \frac{1}{N} \sum_{n=1}^N t_n^i y[n] \right]_{i \in \overline{0,p}}. \quad (158)$$

## Aspecte practice în Modelarea și Identificarea Sistemelor

Implementarea relațiilor (157) și (158) în forma originală conduce de regulă la erori importante, matricea  $R_N$  fiind dezechilibrată numeric. Datorită inegalității (151), se poate observa că suma generică a matricii verifică următoarea proprietate:

$$\sum_{n=1}^N t_n^{i+j} \geq \sum_{n=1}^N n^{i+j} T_s^{i+j} \sim N^{i+j+1} T_s^{i+j+1}, \quad (159)$$

ceea ce implică faptul că elementele de pe diagonala matricii au ordine de mărime extrem de diferite:

$$NT_s, N^3T_s^3, N^5T_s^5, \dots, N^{2p+1}T_s^{2p+1} \quad (160)$$

și inversarea conduce la valori numerice extrem de dezechilibrate. Pentru a corecta acest fenomen, matricii  $R_N$  i se aplică un operator diagonal de *balansare* (adică de echilibrare numerică) de forma:

$$B_N \stackrel{def}{=} \text{diag} \left[ \frac{1}{\sqrt{NT_s}}, \frac{1}{NT_s \sqrt{NT_s}}, \dots, \frac{1}{N^p T_s^p \sqrt{NT_s}} \right]. \quad (161)$$

Cu definiția (161), ecuația (157) se poate exprima echivalent astfel:

$$\hat{\theta}_N = B_N (B_N R_N B_N)^{-1} B_N r_N. \quad (162)$$

În (162), inversarea matricii dintre paranteze se poate efectua acum cu precizie. Se observă de asemenea că matricea de balansare nu se inversează explicit niciodată.

O altă proprietate interesantă utilă implementării metodei de estimare parametrică este recurența verificată de matricile  $R_N$  și vectorii  $r_N$  pentru diferite grade ale polinomului tendință. Astfel, dacă  $R_N$  este renotată cu  $R_{N,p}$ , iar  $r_N$  cu  $r_{N,p}$  (pentru a pune în evidență gradul polinomului), atunci se constată cu ușurință că:

$$R_{N,p} = \begin{bmatrix} & & & \frac{1}{N} \sum_{n=1}^N t_n^p \\ & & & \vdots \\ & R_{N,p-1} & & \frac{1}{N} \sum_{n=1}^N t_n^{2p-1} \\ \hline \frac{1}{N} \sum_{n=1}^N t_n^p & \dots & \frac{1}{N} \sum_{n=1}^N t_n^{2p-1} & \frac{1}{N} \sum_{n=1}^N t_n^{2p} \end{bmatrix} \quad \text{și} \quad r_{N,p} = \begin{bmatrix} r_{N,p-1} \\ \frac{1}{N} \sum_{n=1}^N t_n^p y[n] \end{bmatrix}, \quad (163)$$

matricile  $R_{N,p}$  fiind, în plus simetrice. Recurențele (163) arată de fapt că efortul de calcul depus pentru a evalua matricea inversabilă și vectorul liber pentru un anumit grad  $p$  poate fi conservat la evaluarea acestora pentru gradul următor,  $p+1$ .

Alegerea gradului polinomului tendință se poate efectua apelînd la criteriile structurale descrise în Capitolul 4. De subliniat că nu se pot trasa linii de demarcație nete între cele 3 componente ale unei serii de timp. Acest lucru este ilustrat din plin de polinomul tendinței. Dacă gradul său este prea mare, componenta sezonieră tinde să fie parțial înglobată în modelul tendinței. Dacă gradul acesteia crește și mai mult,

## 8. Modelarea și predicția seriilor de timp

atunci și zgomotele care afectează datele tind să fie parțial modelate prin tendință. Un grad prea mic conduce la o modelare grosieră a tendinței, o parte din informația ei fiind preluată de celelalte 2 componente. Este deci recomandabil ca modelul să fie parsimonios, adică tendința să aibă un grad mic, dar suficient, pentru a discrimina tendința de celelalte 2 componente cu o bună acuratețe.

După construcția modelului tendinței, valorile simulate ale acestuia se scad din valorile seriei de timp inițiale, rezultatul fiind o serie de timp *staționarizată*:

$$y_{\text{sta}}[n] \stackrel{\text{def}}{=} y[n] - y_T(t_n) = y[n] - a_0 - a_1 t_n - \dots - a_p t_n^p, \quad \forall n \in \overline{1, N}. \quad (164)$$

### B. Estimarea componentei sezoniere

Seria de timp staționarizată constituie punctul de plecare pentru determinarea următoarei componente, cea sezonieră. Componenta sezonieră a unei serii de timp exprimă fenomenul de repetabilitate din evoluția procesului care a furnizat datele măsurate. Ea este modelată cu ajutorul a  $P$  valori succesive numite *coeficienți sezonieri*, care sunt replicați prin periodicitate pe durata orizontului de măsură. Noul semnal discret obținut,  $y_S$ , este periodic, de perioadă  $PT_s$ , unde  $P \in \mathbb{N}^*$ , iar  $T_s$  este perioada de eșantionare stabilită conform convențiilor de la punctul precedent.

Prelungirea prin periodicitate a coeficienților sezonieri se efectuează simplu în cazul eșantionării uniforme. În cazul eșantionării neuniforme, după determinarea coeficienților sezonieri, este necesară o interpolare înaintea prelungirii prin periodicitate. Interpolarea poate fi polinomială (liniară sau cu polinomul lui Lagrange) sau cu funcții spline cubice (de preferat).

În mod convențional, coeficienții sezonieri (parametrii necunoscuți ai modelului) sunt notați prin  $y_{S,1}, \dots, y_{S,P}$ , în timp ce indicele structural este numărul  $P$  (perioada în timp normalizat) – de asemenea necunoscut.

Determinarea componentei sezoniere se bazează pe două abordări (în care intervine MCMMP): una temporală (*Metoda Wittacker-Robinson*) și alta frecvențială (*Metoda periodogramei Schuster*).

#### 1. Metoda Wittacher-Robinson (în timp)

Coeficienții sezonieri se pot obține folosind media temporală a unor submulțimi de date consecutive staționarizate.

Dacă seria de timp este eșantionată neuniform, atunci seria staționarizată poate fi interpolată și apoi re-eșantionată uniform, la momente de timp de tipul  $mT_s$ , unde  $m \in \overline{1, M}$ , cu  $M > N$ . Perioada de eșantionare  $T_s$ , dacă nu este precizată, va fi aleasă egală cu durata minimă dintre momentele de eșantionare adiacente. Pentru ușurința exprimării, se poate considera interpolarea liniară. Astfel,  $y_{\text{sta},1}$  – versiunea interpolată liniar a seriei staționarizate – are următoarea exprimare:

$$y_{\text{sta},1}(t) \stackrel{\text{def}}{=} y_{\text{sta}}[n-1] + \frac{t - t_{n-1}}{t_n - t_{n-1}} (y_{\text{sta}}[n] - y_{\text{sta}}[n-1]),$$

$$\forall t \in [t_{n-1}, t_n], \quad \forall n \in \overline{2, N}. \quad (165)$$

Prin re-eșantionarea semnalului (165), se obține setul de date  $\{y_{sta,1}(mT_s)\}_{m \in \overline{1,M}}$ , unde  $M = \lfloor T_{\max}/T_s + 0.5 \rfloor$  (rotunjire la cel mai apropiat întreg).

În cazul eșantionării uniforme,  $M = N$  și  $y_{sta,1}(mT_s) = y_{sta}[m]$ ,  $\forall m \in \overline{1,M}$ .

Pentru fiecare  $P \in 2, \lfloor M/2 \rfloor$ , setul de date  $\{y_{sta,1}(mT_s)\}_{m \in \overline{1,M}}$  este segmentat într-un număr de  $K = \lfloor M/P \rfloor$  seturi de date consecutive (numite și *cadre*) aranjate într-o matrice, pe linii:

$$Y_{sta} \stackrel{def}{=} \begin{bmatrix} y_{sta,1}(T_s) & y_{sta,1}(2T_s) & \cdots & y_{sta,1}(PT_s) \\ y_{sta,1}((P+1)T_s) & y_{sta,1}((P+2)T_s) & \cdots & y_{sta,1}(2PT_s) \\ \vdots & \vdots & \ddots & \vdots \\ y_{sta,1}(((K-1)P+1)T_s) & y_{sta,1}(((K-1)P+2)T_s) & \cdots & y_{sta,1}(KPT_s) \end{bmatrix}. \quad (166)$$

Ultimele date (care nu pot constitui un segment complet) se pierd.

Calculînd media fiecărei coloane a matricii  $Y_{sta}$ , se obțin coeficienții sezonieri:

$$y_{S,p} \stackrel{def}{=} \frac{1}{K} \sum_{k=0}^{K-1} y_{sta,1}((kP+p)T_s), \quad \forall p \in \overline{1,P}. \quad (167)$$

Urmează interpolarea coeficienților sezonieri în cazul eșantionării neuniforme. Astfel, componenta sezonieră este exprimată pe o perioadă de următorul semnal continuu (obținut prin interpolare liniară):

$$\begin{aligned} \tilde{y}_{S,1}(t) &\stackrel{def}{=} y_{S,p-1} + \frac{t - (p-1)T_s}{T_s} (y_{S,p} - y_{S,p-1}), \\ &\forall t \in [(p-1)T_s, pT_s], \quad \forall p \in \overline{1,P}, \end{aligned} \quad (168)$$

unde, datorită periodicității,  $y_{S,0} = y_{S,P}$ . Pe întregul orizont de măsură, componenta sezonieră  $y_S(\bullet)$  este exprimată în timp continuu prin replicarea consecutivă a semnalului (168) de un număr corespunzător de ori. În timp discret, semnalul sezonier continuu trebuie re-eșantionat la aceleași momente de timp ca și seria de timp originală:  $y_S[n] = y_S(t_n)$ ,  $\forall n \in \overline{1,N}$ .

De notat că interpolarea induce distorsiuni ale semnalului periodic rezultat. În cazul interpolării liniare, distorsiunile pot fi mai importante decît pentru interpolarea cu funcții spline cubice. Operațiile de interpolare și re-eșantionare nu mai sunt necesare în cazul în care seria de timp a fost eșantionată uniform.

Alegerea unei componente sezoniere adecvate se efectuează baleind gama de perioade posibile între 2 și  $\lfloor M/2 \rfloor$ . Fie  $y_{S,\tilde{P}}$  componenta sezonieră discretă de perioadă  $P$  (eventual eșantionată neuniform). Dintre toate componentele sezoniere  $\{y_{S,\tilde{P}}\}_{\tilde{P} \in 2, \lfloor M/2 \rfloor}$  se va alege aceea care conduce la o eroare pătratică minimă față de seria staționară. Mai precis, perioada optimă rezultă prin

## 8. Modelarea și predicția seriilor de timp

rezolvarea următoarei probleme de minimizare a erorii pătratice dintre model și procesul furnizor de date:

$$P_{0,t} = \arg \min_{P \in \overline{2, \lfloor M/2 \rfloor}} \mathcal{V}[P], \text{ unde: } \mathcal{V}[P] \stackrel{\text{def}}{=} \sum_{n=1}^N (y_{\text{sta}}[n] - y_{S,\bar{P}}[n])^2, \forall P \in \overline{2, \lfloor M/2 \rfloor}. \quad (169)$$

Deoarece criteriul  $\mathcal{V}$  este discret, minimizarea acestuia se poate efectua printr-o procedură de căutare exhaustivă, cu condiția ca numărul total al perioadelor posibile,  $\lfloor M/2 \rfloor$ , să fie suficient de mic (de ordinul zecilor de mii, cel mult). Dacă acest număr este prea mare (sute de mii, milioane, etc.), determinarea minimului se poate realiza folosind algoritmi de căutare evoluționisti (algoritmul de anealizare, algoritmi genetici, algoritmi de ascensiune montană, etc.) [RuNo95], [MiM95].

Atunci când numărul perioadelor posibile este mic, alegerea perioadei optime se poate realiza și pe cale grafică, după trasarea variației criteriului  $\mathcal{V}$ , ca în Figura 41.

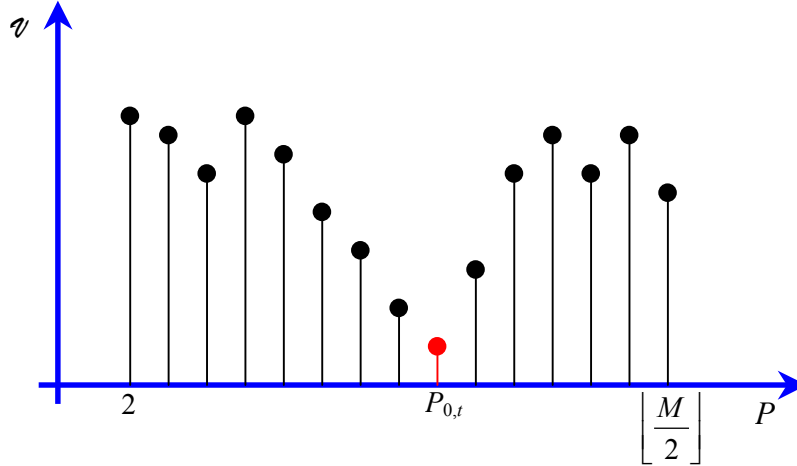


Figura 41. Determinarea perioadei optime cu Metoda Wittacker-Robinson.

Graficul lui  $\mathcal{V}$  poate pune în evidență mai multe minime locale situate la perioade multiple ale unei perioade date. Acestea indică de fapt doar perioada de bază, deoarece un semnal periodic de perioadă  $P$  este periodic și de perioadă  $nP$ , cu  $n \geq 2$ .

Dacă seria staționară nu posedă componentă sezonieră, graficul criteriului  $\mathcal{V}$  este fie aproape constant, fie extrem de oscilant cu numeroase minime locale situate în jurul aceleiași valori.

### 2. Metoda periodogramei Schuster (în frecvență)

Potrivit Teoriei lui J. Fourier, componenta sezonieră fiind un semnal periodic, poate fi aproximată punctual cu o sumă finită de armonice elementare:

$$y_S(t_n) \cong \sum_{m=0}^M (a_m \sin(\omega_m t_n) + b_m \cos(\omega_m t_n)), \quad \forall n \in \overline{1, N}. \quad (170)$$



Numărul maxim al armonicelor,  $M$ , poate fi ales în așa fel încât să fie inferior lui  $\lfloor T_{\max}/(2T_s) \rfloor$  (adică  $\lfloor N/2 \rfloor$  în cazul eșantionării uniforme). Acest număr poate fi ales și mai mare, dar, dincolo de  $\lfloor T_{\max}/(2T_s) \rfloor$ , armonicile au puteri spectrale nesemnificative în raport cu puterile armonicelor anterioare, datorită Teoremei de eșantionare Kotelnikov-Shannon-Nyquist [StD9603], [StD9602a].

Pulsațiile armonicelor sunt alese să fie echidistante (chiar și în cazul eșantionării neuniforme). Se poate arăta că dacă momentele de eșantionare sunt multipli raționali ai perioadei de eșantionare, atunci o reprezentare Fourier corectă se obține alegînd următorul set de pulsații [StD99]:

$$\omega_m \stackrel{\text{def}}{=} \frac{2mM_0\pi}{NT_s}, \quad \forall m \in \overline{0, M}, \quad (171)$$

unde  $M_0$  este restul împărțirii celui mai mic multiplu comun (cmmmc) al numitorilor numerelor raționale  $\{t_n/T_s\}_{n \in \overline{1, N}}$  la  $N$ . În cazul eșantionării uniforme,  $M_0 = 1$ . Pulsația de eșantionare  $\omega_s = 2\pi/N$  controlează rezoluția în frecvență, care este de dorit să fie cît mai bună, deoarece, în acest fel, precizia de determinare a perioadei optime a componentei sezoniere este mai mare. Rezoluția crește odată cu scăderea pulsației de eșantionare, adică odată cu mărirea numărului de date achiziționate,  $N$ . Condiția ca momentele de eșantionare să fie multipli raționali ai perioadei de eșantionare nu este restrictivă, deoarece, în practică, se operează numai cu numere raționale.

Pentru a estima parametrii necunoscuți ai modelului sezonier (170), adică  $\{a_m\}_{m \in \overline{1, M}}$  și  $\{b_m\}_{m \in \overline{0, M}}$  (coeficienții Fourier), trebuie rezolvată o problemă de minimizare a erorii pătratică dintre model și proces:

$$\hat{\theta}_M = \arg \min_{\theta \in \mathbb{R}^{2M+1}} \mathcal{V}_M(\theta), \quad \text{unde: } \mathcal{V}_M(\theta) \stackrel{\text{def}}{=} \sum_{n=1}^N (y[n] - y_S(t_n))^2, \quad \forall \theta \in \mathbb{R}^{2M+1}, \quad (172)$$

unde  $\theta$  este vectorul coeficienților Fourier. De notat că, deoarece seria de timp a fost staționarizată, coeficientul Fourier  $b_0$  este aproximativ nul (el este proporțional cu media datelor, codificînd componenta staționară a acestora). Rezolvarea problemei (169) se poate efectua prin metoda clasică (anularea gradientului criteriului  $\mathcal{V}_M$ ). Rezultă următoarele estimări ale coeficienților Fourier:

$$\hat{\theta}_M = R_M^{-1} r_M, \quad (173)$$

unde:

$$R_M = \begin{bmatrix} \left[ \sum_{n=1}^N \sin(\omega_i t_n) \sin(\omega_j t_n) \right]_{i,j \in \overline{1, M}} & \left[ \sum_{n=1}^N \sin(\omega_i t_n) \cos(\omega_j t_n) \right]_{i,j \in \overline{1, M}} \\ \left[ \sum_{n=1}^N \sin(\omega_i t_n) \cos(\omega_j t_n) \right]_{i,j \in \overline{1, M}} & \left[ \sum_{n=1}^N \cos(\omega_i t_n) \cos(\omega_j t_n) \right]_{i,j \in \overline{1, M}} \end{bmatrix}; \quad (174)$$

$$r_{N,p} = \begin{bmatrix} \sum_{n=1}^N y_{sta}[n] \sin(\omega_i t_n) \\ \vdots \\ \sum_{n=1}^N y_{sta}[n] \cos(\omega_i t_n) \end{bmatrix}_{i \in \overline{1,M}}. \quad (175)$$

Matricea  $R_M$  din (174) devine diagonală în cazul în care seria de timp este uniform eșantionată. Aceasta conduce la exprimarea coeficienților Fourier în formă completă:

$$\begin{cases} \hat{a}_m = \frac{N}{2} \sum_{n=1}^N y_{sta}[n] \sin(\omega_m t_n) \\ \hat{b}_m = \frac{N}{2} \sum_{n=1}^N y_{sta}[n] \cos(\omega_m t_n) \end{cases}, \quad \forall m \in \overline{1,M}. \quad (176)$$

După estimarea coeficienților Fourier, se poate trasa graficul *periodogramei Schuster*, care este definită prin:

$$\mathcal{P}[m] \stackrel{def}{=} \sqrt{\hat{a}_m^2 + \hat{b}_m^2}, \quad \forall m \in \overline{1,M}. \quad (177)$$

Valorile periodogramei aproximează puterile spectrale ale armonicelor din componența seriei de timp staționarizate. Armonica dominantă se poate determina prin evaluarea punctului de maxim al periodogramei, adică prin rezolvarea următoarelor probleme de maxim:

$$m_0 = \arg \max_{m \in \overline{1,M}} \mathcal{P}[m]. \quad (178)$$

Indexul pulsației optime,  $m_0$ , conduce direct la perioada optimă a componentei sezoniere în timp continuu:

$$T_0 = \frac{2\pi}{\omega_{m_0}} = \frac{NT_s}{m_0 M_0}. \quad (179)$$

În timp discret, perioada optimă se obține prin rotunjire la cel mai apropiat întreg:

$$P_{0,f} = \left\lfloor \frac{T_0}{T_s} + \frac{1}{2} \right\rfloor = \left\lfloor \frac{N}{m_0 M_0} + \frac{1}{2} \right\rfloor. \quad (180)$$

Ca și în cazul abordării anterioare (în timp), rezolvarea problemei (178) se poate realiza practic prin căutarea exhaustivă a maximului său direct de pe graficul periodogramei (ca în Figura 42). De altfel, graficul periodogramei conduce la aceleași concluzii ca și cel al erorii pătratice din cazul metodei anterioare.

Abordarea în frecvență completează demersul anterior, bazat pe estimarea componentei sezoniere direct în domeniul timpului, în sensul că, pentru a decide perioada optimă a componentei sezoniere, trebuie comparate perioadele oferite de ambele metode.

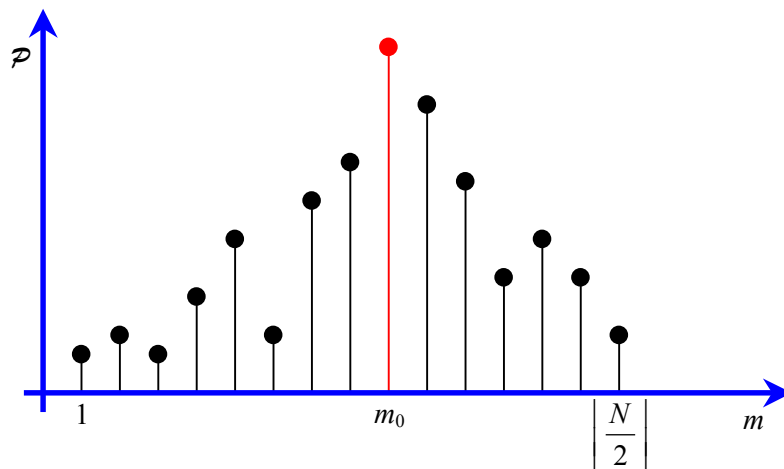


Figura 42. Determinarea perioadei optime cu Metoda periodogramei Schuster.

Astfel:

- Dacă graficul criteriului  $\mathcal{V}$  și cel al periodogramei  $\mathcal{P}$  sunt grupate într-o bandă de  $\pm 10\%$  în jurul mediilor lor, seria de timp nu posedă componentă sezonieră.
- Dacă  $P_{0,t} = P_{0,f}$  (caz destul de rar), atunci perioada în timp discret este  $P_0 = P_{0,t} = P_{0,f}$ .
- Dacă  $P_{0,t} \cong P_{0,f}$  (adică dacă  $P_{0,t}$  și  $P_{0,f}$  diferă cu o valoare mică în raport cu valorile lor, se preferă Metoda Wittacker Robinson (deoarece introduce mai puține erori de calcul). Așadar, în acest caz:  $P_0 = P_{0,t}$ .
- Dacă una din cele două perioade o divide pe cealaltă, se va alege perioada optimă în mod natural:  $P_0 = \min\{P_{0,t}, P_{0,f}\}$ .
- Dacă perioadele  $P_{0,t}$  și  $P_{0,f}$  sunt total diferite, se vor lua în calcul și alte valori rezultate din extremele locale ale criteriului  $\mathcal{V}$  și periodogramei  $\mathcal{P}$ . Se va alege o pereche de valori apropiate ale perioadei (dacă este posibil) și se va selecta perioada oferită de minimul local corespunzător al criteriului  $\mathcal{V}$ . Dacă nu se poate stabili nici o corespondență între valorile posibile ale perioadelor rezultate în urma celor 2 abordări, se poate considera că seria de timp nu posedă componentă sezonieră.

Alegerea perioadei optime a componentei sezoniere constituie punctul cel mai delicat al modelării seriilor de timp. În afara celor 2 abordări, se pot utiliza în acest scop și o serie de informații apriorice legate de procesul furnizor de date, dacă sunt cunoscute. De exemplu, seria de timp a ratei șomajului înregistrat lunar într-o anumită țară va avea probabil o perioadă de 12 luni (considerînd că perioada de eșantionare este de 1 lună).

Odată ce perioada componentei sezoniere a fost stabilită, coeficienții sezonieri se evaluează după procedeul descris în cadrul metodei Wittacher-Robinson (definițiile (165)-(166), cu  $P_0$  în loc de  $P$  și numărul de cadre  $K$  evaluat corespunzător).

Dacă există, semnalul periodic asociat,  $y_S$ , se obține apoi tot în maniera descrisă în cadrul Metodei Wittacker-Robinson (prelungire prin periodicitate și, eventual, re-eșantionare). Dacă seria de date nu posedă componentă sezonieră, aceasta este asimilată cu un semnal nul. Prin scăderea componentei sezoniere din seria de date staționarizată se obține semnalul perturbator al datelor:

$$v[n] \stackrel{def}{=} y_{sta}[n] - y_S(t_n), \quad \forall n \in \overline{1, N}. \quad (181)$$

### C. Estimarea componentei nedeterminate (aleatoare)

Dacă tendința și componenta sezonieră ale seriei de timp au fost corect determinate, setul de date  $\{v[n]\}_{n \in \overline{1, N}}$ , obținut după extragerea lor din seria de timp, are caracteristicile unui zgomot colorat rezidual. Filtrul de zgomot poate fi de tip FIR sau IIR. În primul caz, se poate opera cu un model de identificare de tip MA. În al doilea caz, este utilizat modelul AR. Filtrele de tip FIR sunt de asemenea utilizate pentru a aproxima filtre de tip IIR, dar funcțiile pondere au, în general, o lungime ridicată. Vom adopta în continuare filtrul de tip IIR.

Modelul AR este unul dintre primele modele de identificare utilizate în aplicații, în special datorită simplității și a posibilității de a estima parametrii în manieră recursivă. Reamintim că ecuația modelului AR este următoarea:

$$\begin{cases} v[n] + a_1 v[n-1] + \dots + a_{na} v[n-na] = e[n], \\ E\{e[n]e[m]\} = \lambda^2 \delta_0[n-m] \end{cases}, \quad \forall n \in \overline{1, N}, \quad (182)$$

unde parametri necunoscuți sunt coeficienții  $\{a_i\}_{i \in \overline{1, na}}$  (asamblați într-un vector  $\theta_{na} \in \mathbb{R}^{na}$ ) și dispersia zgomotului alb  $\lambda^2$ . Indicele structural al modelului (necunoscut și el) este  $na \geq 1$ . În general, indicele structural maxim nu depășește valoarea  $Na = 6$  pentru majoritatea seriilor de timp.

Determinarea modelului AR (182) se bazează pe MCMMP. Astfel, pentru fiecare index structural  $na \in \overline{1, Na}$ , parametri necunoscuți estimați din zgomotul colorat sunt următorii:

$$\hat{\theta}_{na} = R_{na}^{-1} r_{na}; \quad \hat{\lambda}_{na}^2 = \frac{1}{N} \sum_{n=1}^N \left( v[n] - \varphi_{na}^T[n] \hat{\theta}_{na} \right)^2, \quad (183)$$

unde:

$$\varphi_{na}^T[n] \stackrel{def}{=} [-v[n-1] \quad \dots \quad -v[n-na]]; \quad (184)$$

$$R_{na} \stackrel{def}{=} \frac{1}{N} \sum_{n=1}^N \varphi_{na}[n] \varphi_{na}^T[n] = \left[ \frac{1}{N} \sum_{n=1}^N v[n-i] v[n-j] \right]_{i, j \in \overline{1, na}}; \quad (185)$$

$$r_{na} \stackrel{def}{=} \frac{1}{N} \sum_{n=1}^N \varphi_{na}[n] v[n] = \left[ -\frac{1}{N} \sum_{n=1}^N v[n] v[n-i] \right]_{i \in \overline{1, na}}. \quad (186)$$

## Aspecte practice în Modelarea și Identificarea Sistemelor

Implementarea relațiilor (183)-(186) se poate realiza mai eficient observînd că matricea  $R_{na}$  este Toeplitz simetrică (adică este generată de prima linie sau coloană), iar majoritatea elementelor vectorului  $r_{na}$  se regăsesc printre elementele matricii  $R_{na}$ . Relații de recurență similare celor de la (163) pot fi de asemenea puse în evidență cu ușurință.

Pentru a evita inversarea matricii  $R_{na}$  se poate utiliza Algoritmul Levinson-Durbin [PrMa96], pe care îl vom prezenta în continuare. Acesta permite estimarea recursivă a coeficienților modelului AR[na], în funcție de coeficienții modelului AR[na-1]. Pentru a ilustra aceste relații recursive, coeficienții modelului AR[na] se notează cu  $a_{na,i}$ ,  $\forall i \in \overline{1, na}$ . De asemenea, se notează cu  $\hat{r}_v$  funcția de auto-covarianță aproximativă estimată din datele reziduale (adică termenul din dreapta al relației aproximative (9)). Algoritmul este descris în Figura 43.

Figura 43. Algoritmul Levinson-Durbin.

➤ Date de intrare: seria de date reziduale:  $\{v[n]\}_{n \in \overline{1, N}}$ .

1. Inițializare: 
$$\begin{cases} \hat{a}_{1,1} = -\frac{\hat{r}_v[1]}{\hat{r}_v[0]} \\ \hat{\lambda}_1^2 = \hat{r}_v[0] + \hat{a}_{1,1}\hat{r}_v[1] = \hat{r}_v[0](1 - \hat{a}_{1,1}^2) \end{cases}$$

2. Pentru  $na \in \overline{2, Na}$ :

$$\begin{cases} \hat{a}_{na,na} = -\frac{1}{\hat{\lambda}_{na-1}^2}(\hat{r}_v[na] + \hat{a}_{na-1,1}\hat{r}_v[na-1] + \dots + \hat{a}_{na-1,na-1}\hat{r}_v[1]) \\ \hat{a}_{na,i} = \hat{a}_{na-1,i} + \hat{a}_{na,na}\hat{a}_{na-1,na-i}, \quad \forall i \in \overline{1, na-1} \\ \hat{\lambda}_{na}^2 = \hat{\lambda}_{na-1}^2(1 - \hat{a}_{na,na}^2) \end{cases}$$

➤ Date de ieșire: coeficienții tuturor modelelor AR[na], cu  $na \in \{1, 2, \dots, Na\}$ .

Relațiile algoritmului au fost deduse prin exploatarea unor proprietăți remarcabile ale matricilor Toeplitz simetrice. De altfel, folosind acest algoritm, se poate proiecta o procedură eficientă de inversare a matricilor Toeplitz simetrice. Se poate arăta că modelul AR determinat cu ajutorul Algoritmului Levinson-Durbin este stabil (rădăcinile polinomului auto-regresiv sunt situate în discul unitar al planului complex) [PrMa96].

De asemenea, coeficienții  $\hat{a}_{na,na}$  se mai notează prin  $\hat{k}_{na}$  și se mai numesc *coeficienți de reflexie*. Ei joacă un rol important în estimarea dispersiei zgomotului alb, care îndeplinesc funcția de eroare de predicție cu un pas. Se poate arăta că  $|\hat{k}_{na}| < 1$ ,

$\forall na \geq 1$ , ceea ce arată că estimația dispersiei zgomotului alb ( $\hat{\lambda}_{na}^2$ ) scade odată cu ordinul modelului. Cu toate acestea, așa cum arată paragraful următor, precizia

## 8. Modelarea și predicția seriilor de timp

predicției cu mai mulți pași a modelului AR nu se îmbunătățește în mod necesar odată cu ordinul acestuia.

Alegerea modelului optimal al componentei nedeterminate se realizează folosind criteriile structurale descrise în Capitolul #4. Se poate utiliza și graficul dispersiei  $\hat{\lambda}_{na}^2$  în acest scop. Odată ce ordinul optim ( $na_0$ ) a fost selectat, modelul matematic al componentei aleatoare este dat de următoarea ecuație omogenă recursivă (cu diferențe):

$$\begin{cases} y_{AR}[n] + \hat{a}_{na_0,1}y_{AR}[n-1] + \dots + \hat{a}_{na_0,na_0}y_{AR}[n-na] = 0 \\ y_{AR}[1] = v[1]; y_{AR}[2] = v[2]; \dots; y_{AR}[na] = v[na] \end{cases}, \quad \forall n \in \overline{1, N}, \quad (187)$$

De notat că zgomotul alb rezidual lipsește din ecuația (187). El va interveni (prin dispersia sa) în faza de predicție.

De asemenea, se poate observa că maniera de eșantionare a seriei de timp nu este importantă în modelarea componentei aleatoare. Cu toate acestea, dacă seria de timp a fost eșantionată neuniform, modelul  $y_{AR}$  ar putea fi determinat după interpolarea zgomotului colorat rezidual și re-eșantionarea sa uniformă. Modelul obținut este la rindul său interpolat și apoi re-eșantionat neuniform. Însă cele 2 interpolări distorsionează, de regulă, rezultatul final.

### D. Predicția seriei de timp

Modelul complet al seriei de timp (152), conține 2 componente deterministe ( $y_T$  și  $y_S$ ) și una nedeterministă ( $y_{AR}$ ). De aceea, prognoza seriei de timp nu se realizează doar prin extrapolarea celor 3 componente pe orizontul de predicție, ci și prin estimarea preciziei valorilor predictate. Aceasta este determinată de dispersia erorii de predicție. Singura componentă care oferă posibilitatea de a estima eroarea de predicție este cea nedeterministă.

În IS se operează cu conceptul de *predictor optimal*, care oferă prognoze cu eroare de predicție minimală. În cazul modelului AR, dacă se notează prin  $\hat{y}_{AR}[N+k|N]$  valoarea predictată la momentul  $t_{N+k}$  din  $N$  date măsurate, predictorul optimal este descris de următoarele relații recursive:

$$\hat{y}_{AR}[N+k|N] \stackrel{def}{=} \begin{cases} v[N+k] & , k \leq 0 \\ -\hat{a}_{na_0,1}\hat{y}_{AR}[N+k-1|N] - \dots - \hat{a}_{na_0,k-1}\hat{y}_{AR}[N+1|N] - \\ -\hat{a}_{na_0,k}v[N] - \dots - \hat{a}_{na_0,na_0}v[N+k-na_0] & , k \in \overline{1, na_0} \\ -\hat{a}_{na_0,1}\hat{y}_{AR}[N+k-1|N] - \dots - \hat{a}_{na_0,na_0}\hat{y}_{AR}[N-na_0|N] & , k \geq na_0 + 1 \end{cases} \quad (188)$$

Pentru a evalua precizia valorii predictate  $\hat{y}_{AR}[N+k|N]$ , se apelează la modelul AR (182), cu ajutorul căruia se poate estima eroarea de predicție:

$$\hat{e}[N+k] = v[N+k] - \hat{y}_{AR}[N+k|N], \quad \forall k \in \overline{0, K}, \quad (189)$$

și dispersia acesteia, notată cu  $\sigma_k^2$ . După o serie de calcule elementare în care intervine definiția (188), se poate arăta că erorile de predicție verifică ecuația:

$$\begin{aligned} \hat{e}[N+k] + \hat{a}_{na_0,1} \hat{e}[N+k-1] u_0[k-1] + \dots + \hat{a}_{na_0,na_0} \hat{e}[N+k-na_0] u_0[k-na_0] = \\ = e[N+k], \quad \forall k \in \overline{1, N}, \end{aligned} \quad (190)$$

unde  $u_0$  este treapta unitară discretă, iar  $\hat{e}[N] = v[N] - v[N] = 0$ .

Plecînd de la ecuația (190), se pot deduce expresiile dispersiilor erorilor de predicție. De exemplu, primele 3 estimări ale dispersiei de predicție sunt următoarele:

$$\sigma_1^2 = \lambda^2 \Rightarrow \hat{\sigma}_1^2 = \hat{\lambda}_{na_0}^2; \quad (191)$$

$$\sigma_2^2 = \lambda^2 (1 + \hat{a}_{na_0,1}^2) \Rightarrow \hat{\sigma}_2^2 = \hat{\lambda}_{na_0}^2 (1 + \hat{a}_{na_0,1}^2) = \hat{\sigma}_1^2 (1 + \hat{a}_{na_0,1}^2); \quad (192)$$

$$\begin{aligned} \sigma_3^2 = \lambda^2 \left( 1 + \hat{a}_{na_0,1}^2 + (\hat{a}_{na_0,1}^2 - \hat{a}_{na_0,2})^2 \right) \Rightarrow \\ \Rightarrow \hat{\sigma}_3^2 = \hat{\lambda}_{na_0}^2 \left( 1 + \hat{a}_{na_0,1}^2 + (\hat{a}_{na_0,1}^2 - \hat{a}_{na_0,2})^2 \right) = \hat{\sigma}_2^2 (1 + \hat{a}_{na_0,1}^2) + \hat{\sigma}_2^2 (\hat{a}_{na_0,1}^2 - \hat{a}_{na_0,2})^2. \end{aligned} \quad (193)$$

Evident, relațiile (191)-(193) arată că dispersia erorii de predicție crește ( $\hat{\sigma}_1^2 \leq \hat{\sigma}_2^2 \leq \hat{\sigma}_3^2 \leq \dots$ ), ceea ce implică o deteriorare a preciziei prognozei pe măsură ce momentele de predicție se îndepărtează de orizontul de măsură.

Valorile predictate ale seriei de timp se obțin astfel:

$$\hat{y}[N+k | N] = y_T[N+k] + y_S[N+k] + \hat{y}_{AR}[N+k | N], \quad \forall k \in \overline{1, K}. \quad (194)$$

Acestea sunt de regulă approximate pe graficul seriei de timp împreună cu precizia de estimare, reprezentată de intervalele (segmentele verticale):

$$[y[N+k] - 3\hat{\sigma}_k, y[N+k] + 3\hat{\sigma}_k], \quad \forall k \in \overline{1, K}, \quad (195)$$

considerînd că zgomotul alb rezidual este și Gaussian. Practic, valoarea predictată se află în acest interval cu probabilitate superioară lui 90%. Evident, cu cît intervalul este mai larg, cu atît precizia de predicție este mai slabă.

Obiectivul acestui capitol este de a determina și utiliza modelele de predicție ale unor serii de timp.

## 8.2. Exerciții

### Exercițiul 8.1

Deduceți estimăția (157)-(158) a parametrilor necunoscuți care exprimă modelul polinomial al tendinței unei serii de timp.

### Exercițiul 8.2

Exprimați relațiile de interpolare cu funcții spline cubice ale seriei staționarizate de date eșantionate neuniform. Reamintim că funcțiile spline cubice sunt polinoame de gradul 3 care verifică proprietățile următoare:

- coeficienții lor trebuie reactualizați pentru fiecare pereche de noduri de eșantionare/interpolare adiacente;

## 8. Modelarea și predicția seriilor de timp

- b. oricare două funcții spline adiacente coincid în nodurile de eșantionare/interpolare;
- c. derivatele de ordin 1 ale oricăror două funcții spline adiacente coincid în nodurile de eșantionare/interpolare;
- d. pentru nodurile de eșantionare/interpolare extreme, funcțiile spline și derivatele lor de ordin 1 sunt obligate să verifice condiții inițiale și finale impuse (de regulă, valori nule ale derivatelor).

### Exercițiul 8.3

Deduceți estimăția (173)-(175) a parametrilor necunoscuți care exprimă modelul componentei sezoniere a unei serii de timp. Demonstrați că, în cazul în care seria este eșantionată uniform, parametrii se pot exprima în forma completă (176).

#### Indicație

- Se poate ține cont de următoarea relație a lui Poisson:

$$\sum_{n=0}^{N-1} \exp\left(\pm \frac{2kn\pi}{N} j\right) = N\delta_0[k - N\mathbb{Z}], \quad \forall k \in \mathbb{Z},$$

unde  $N\mathbb{Z}$  este mulțimea multiplilor întregi ai lui  $N$ , iar  $\delta_0$  este impulsul unitar discret sau simbolul lui Kronecker.

### Exercițiul 8.4

- a. Proiectați o procedură de inversare a matricilor Toeplitz simetrice folosind Algoritmul Levinson-Durbin din Figura 43.
- b. Demonstrați că erorile de predicție verifică ecuația (190).
- c. Deduceți estimățiile dispersiilor erorilor de predicție pentru primii 3 pași de predicție (ecuațiile (191), (192) și (193)).

## 8.3. Probleme de simulare

Un număr de 15 serii de timp sunt puse la dispoziția utilizatorilor. Acestea sunt înregistrate sub forma unor fișiere MATLAB care, odată apelate, încarcă în memorie seria de date în variabila **y** (vector linie) și semnificația datelor în variabila de tip text **tit** (adică *titlul* seriei de timp). Seriile de timp sunt descrise succint în Tabelul 2, care precede variațiile lor grafice din finalul capitolului. Cu excepția ultimelor 3 serii de timp, celelate 12 serii sunt eșantionate uniform. Ultimele 3 serii de timp sunt eșantionate neuniform, vectorul linie **d** conținând momentele de eșantionare.

### Problema 8.1

Să se proiecteze mini-simulatorul **ISLAB\_8A** care returnează modelul parsimonios al tendinței unei serii de timp. Apelul tipic al rutinei va fi următorul:

**[yT,ysta,d,theta\_T] = ISLAB\_8A ;**

unde: **yT** este vectorul linie al valorilor polinomului tendință calculate în momentele de eșantionare (adică  $\{y_T(t_n)\}_{n \in \overline{1,N}}$ ), **ysta** este vectorul linie al datelor staționarizate (**ysta=y-yT**), **d** este vectorul linie al momentelor de eșantionare, iar **theta\_T** este vectorul coloană al coeficienților estimați ai



modelului. Utilizatorul va avea posibilitatea să aleagă pe oricare din cele 15 serii de timp sau o serie de date proprie înregistrată în același format ca seriile de timp disponibile. De asemenea, pentru alegerea indicelui structural optim al modelului (adică a gradului polinomului), se poate folosi funcția MATLAB `ginput`, cu ajutorul căreia se pot citi în mod direct coordonatele oricărui punct al unui grafic de funcție afișat pe ecran. Astfel, de exemplu, dacă se folosește criteriul aplatizării, utilizatorul va indica punctul de început al palierului dispersiei erorii de predicție (adică  $\mathcal{U}_N(\theta)$ ) pe cale grafică, urmînd ca programul să stabilească automat ordinul optim al modelului (prin aproximare la cel mai apropiat întreg). Se vor afișa graficele seriei de timp inițiale, tendinței selectate și seriei de timp staționarizate.

### Problema 8.2

Să se proiecteze mini-simulatorul `ISLAB_8B` care returnează modelul componentei sezoniere a unei serii de timp. Apelul tipic al rutinei va fi următorul:

`[yS,v,P0] = ISLAB_8B(ysta,d) ;`

unde: `ysta` este vectorul linie al datelor staționarizate, `d` este vectorul linie al momentelor de eșantionare, `yS` este vectorul linie al valorilor componentei sezoniere calculate la momentele de eșantionare (adică  $\{y_S(t_n)\}_{n \in \overline{1,N}}$ ), `v` este vectorul linie al zgomotului colorat rezultat după extragerea componentei sezoniere din seria de date staționarizată (`v=ysta-yS`), iar `P0` este perioada discretă a componentei sezoniere. Pentru alegerea perioadei optime a modelului, se poate folosi funcția MATLAB `ginput`. Se vor afișa graficele seriei de timp staționarizate, componentei sezoniere și zgomotului colorat rezidual.

### Problema 8.3

a. Să se proiecteze mini-simulatorul `ISLAB_8C` care returnează modelul componentei aleatoare a unei serii de timp. Apelul tipic al rutinei va fi următorul:

`[theta,lambda2,e] = ISLAB_8C(v,d) ;`

unde: `v` este vectorul linie al zgomotului rezidual colorat, `d` este vectorul linie al momentelor de eșantionare, `theta` este vectorul coloană al coeficienților estimați ai predictorului AR optimal (adică  $\{\hat{a}_{na_0,i}\}_{i \in \overline{1,na_0}}$ ), `lambda2` este dispersia estimată a zgomotului alb rezidual (adică  $\hat{\lambda}_{na_0}^2$ ), iar `e` este vectorul linie al valorilor zgomotului alb rezidual. Pentru alegerea ordinului optim al modelului, se poate folosi funcția MATLAB `ginput`. De asemenea, pentru estimarea parametrilor, se recomandă implementarea Algoritmului Levinson-Durbin. Se vor afișa graficele zgomotului colorat rezidual, al componentei aleatoare și al zgomotului alb rezidual. Indicați raportul semnal-zgomot estimat (SNR) pe graficul zgomotului alb rezidual (în deciBeli).

b. Folosind cele 3 mini-simulatoare anterioare, să se proiecteze mini-simulatorul `ISLAB_8D` care returnează valorile predictate ale unei serii de timp pe un orizont de predicție precizat. Apelul tipic al rutinei va fi următorul:

`[ypred,sigma2,d] = ISLAB_8D(K) ;`

## 8. Modelarea și predicția seriilor de timp

---

unde: **ypred** este vectorul linie al seriei de timp predictate, **sigma2** este vectorul linie al dispersiilor erorilor de predicție corespunzătoare (adică  $\{\hat{\sigma}_k^2\}_{k \in \overline{1, K}}$ ), **d** este vectorul linie al momentelor de predicție, iar **K** este dimensiunea orizontului de predicție. Utilizatorul va avea posibilitatea să aleagă pe oricare din cele 15 serii de timp sau o serie de date proprie înregistrată în același format ca seriile de timp disponibile. Modelul seriei de timp va fi construit restrângând seria de timp la un număr de eșantioane egal cu  $N - K$ , pentru a testa precizia acestuia pe orizontul de predicție. În afara graficelor afișate de către mini-simulatoarele **ISLAB\_8{A,B,C}**, se vor afișa în final graficele seriei de timp măsurate și predictate, împreună cu intervalul de precizie al fiecărei valori predictate.

Tabelul 2. Serii de timp disponibile pe Discul Compact.

Fișier (Figură)	Semnificație
<b>ST01.M</b> (Figura 44)	Rata lunară a numărului de șomeri din SUA între Ianuarie 1973 și iulie 1985 [%].
<b>ST02.M</b> (Figura 45)	Circulația monedei belgiene măsurată lunar, timp de 10 ani, între 1980 și 1990 [miliarde BFr].
<b>ST03.M</b> (Figura 46)	Media lunară a numărului de pete solare observate între 1976 și 1989.
<b>ST04.M</b> (Figura 47)	Distanța lunară parcursă la U.K. Airlines pe cursele interne între 1982 și 1989 [mii km].
<b>ST05.M</b> (Figura 48)	Rata lunară a șomajului în Marea Britanie între 1978 și 1989 [%].
<b>ST06.M</b> (Figura 49)	Rata lunară a șomajului în Franța între 1980 și 1990 [%].
<b>ST07.M</b> (Figura 50)	Rata lunară a șomajului în Canada între 1979 și 1989 [%].
<b>ST08.M</b> (Figura 51)	Veniturile lunare realizate din impozitele pe telefoane, într-o regiune din SUA [milioane USD].
<b>ST09.M</b> (Figura 52)	Media lunară a timpului mediu de lucru săptămînal în SUA între 1979 și 1989 [ore].
<b>ST10.M</b> (Figura 53)	Numărul lunar al bolnavilor operați de amigdalită la Spitalul 23 August din București, între 1982 și 1990.
<b>ST11.M</b> (Figura 54)	Intensitatea conștiinței colective pe Terra măsurată lunar între 2000 și 2004 la Kings College în Londra [mH].
<b>ST12.M</b> (Figura 55)	Intensitatea radio cosmică măsurată la radio-telescopul din Indianapolis (SUA) între 2001 și 2004 [mV DC].
<b>ST13.M</b> (Figura 56)	Rata de conversie între USD și ROL începînd cu 15 octombrie 2001.
<b>ST14.M</b> (Figura 57)	Rata de conversie între EURO și ROL începînd cu 10 ianuarie 2002.
<b>ST15.M</b> (Figura 58)	Rata de conversie între USD și EURO începînd cu 10 ianuarie 2002.

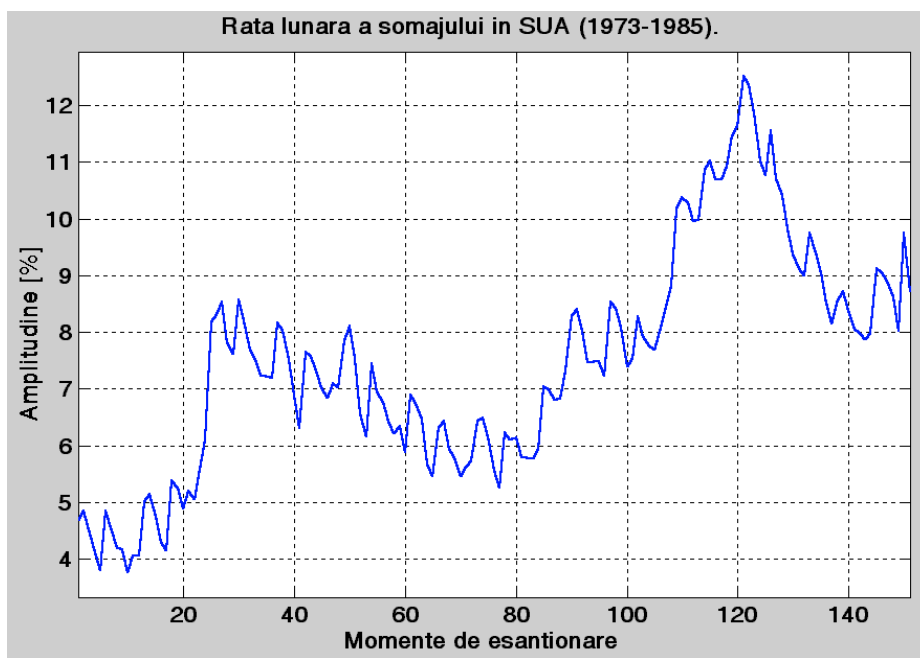


Figura 44. Rata lunară a numărului de șomeri din SUA.

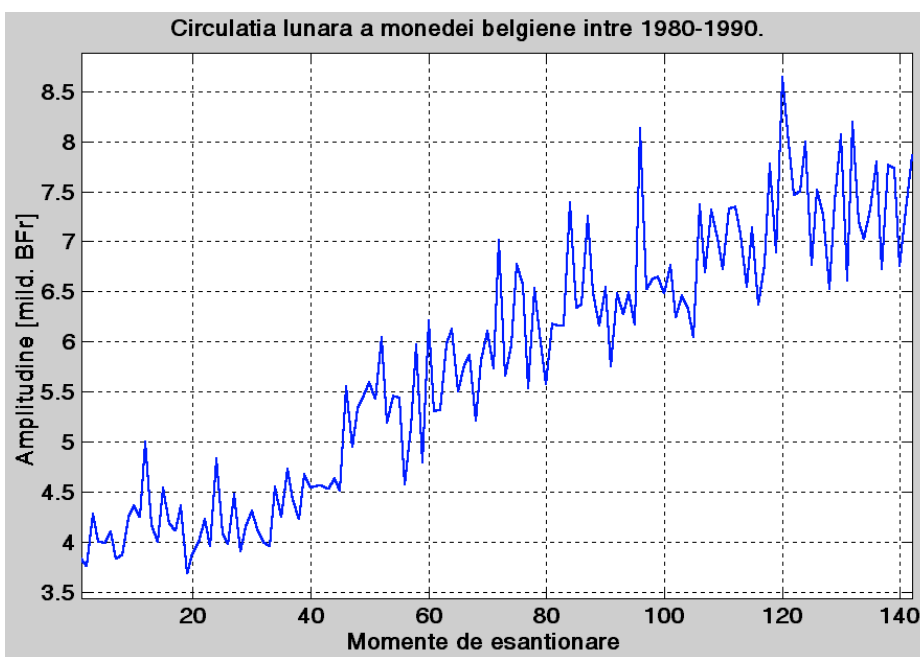


Figura 45. Circulația monedei belgiene măsurată lunar, timp de 10 ani.

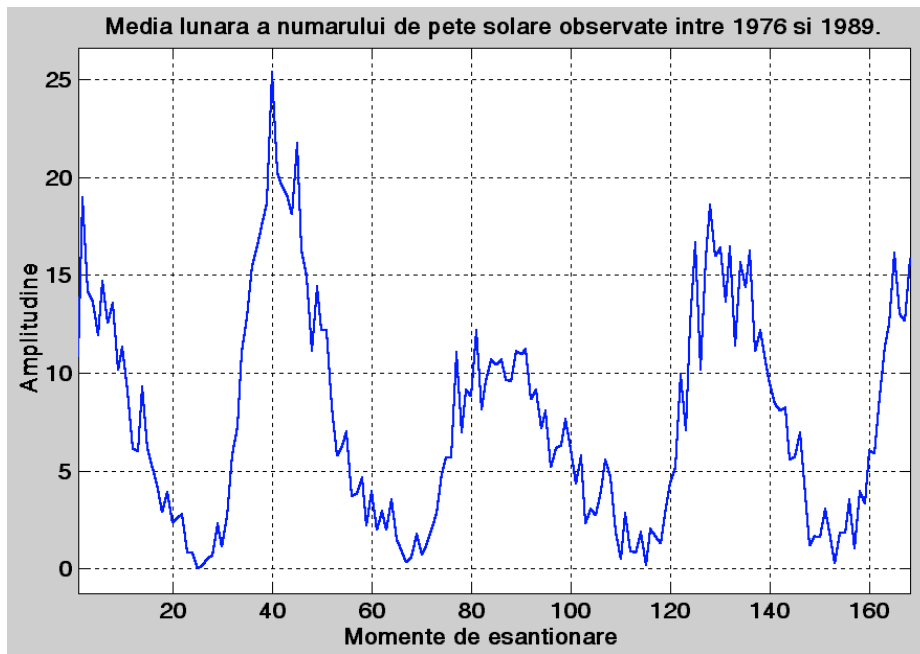


Figura 46. Media lunară a numărului de pete solare observate.

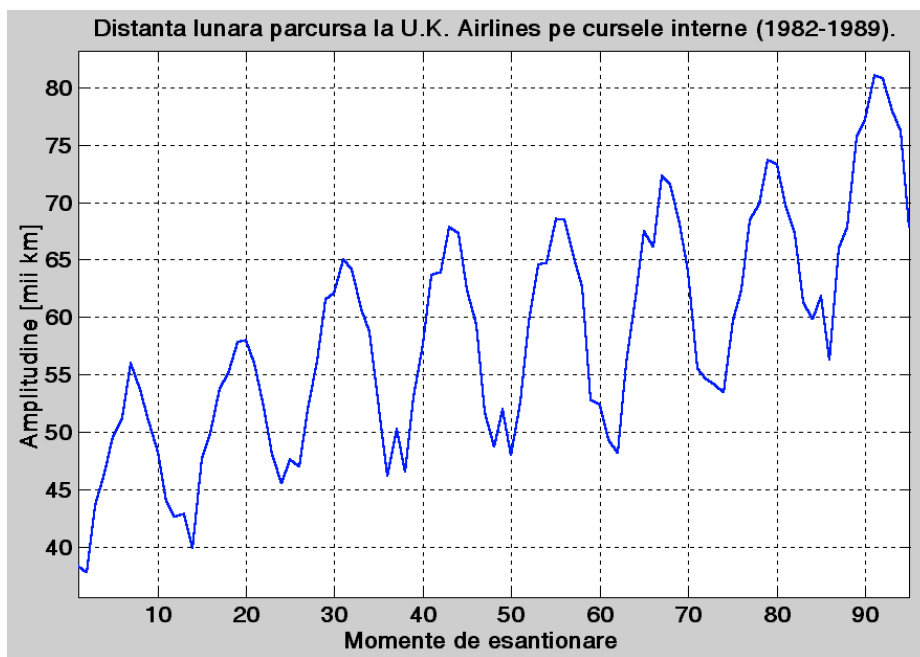


Figura 47. Distanța lunară parcursă la U.K. Airlines pe cursele interne.

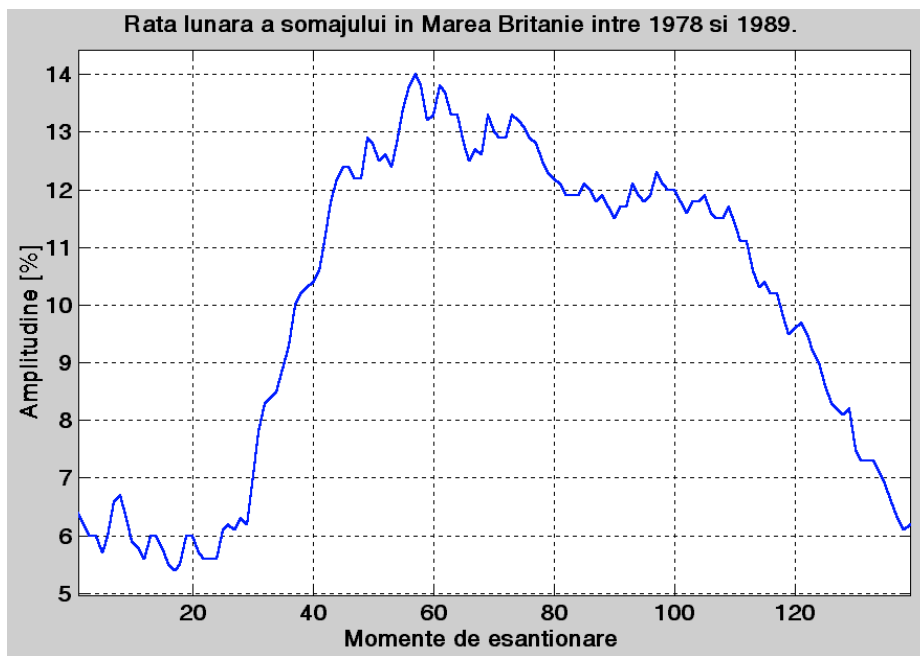


Figura 48. Rata lunară a șomajului în Marea Britanie.

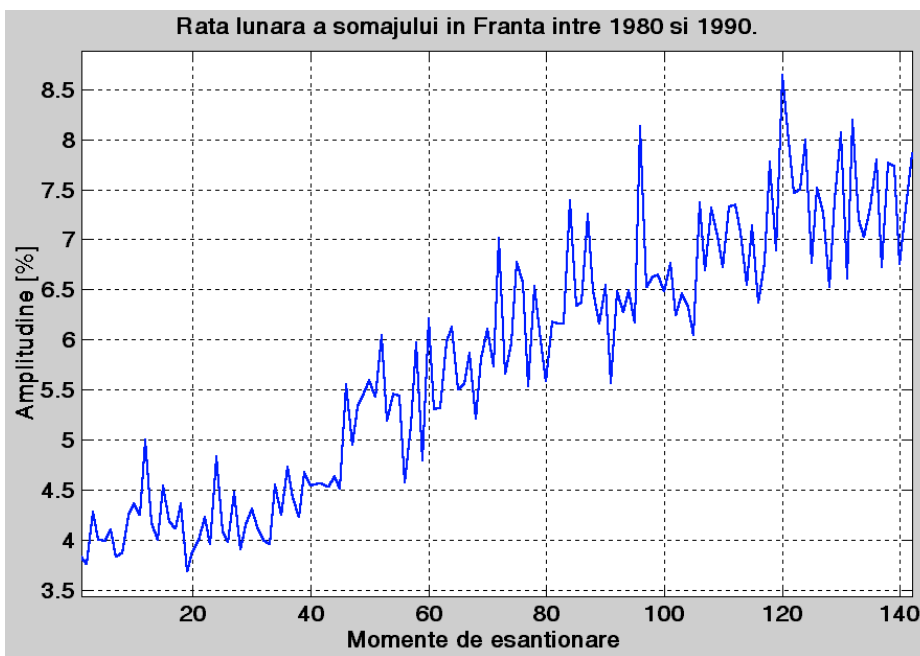


Figura 49. Rata lunară a șomajului în Franța.

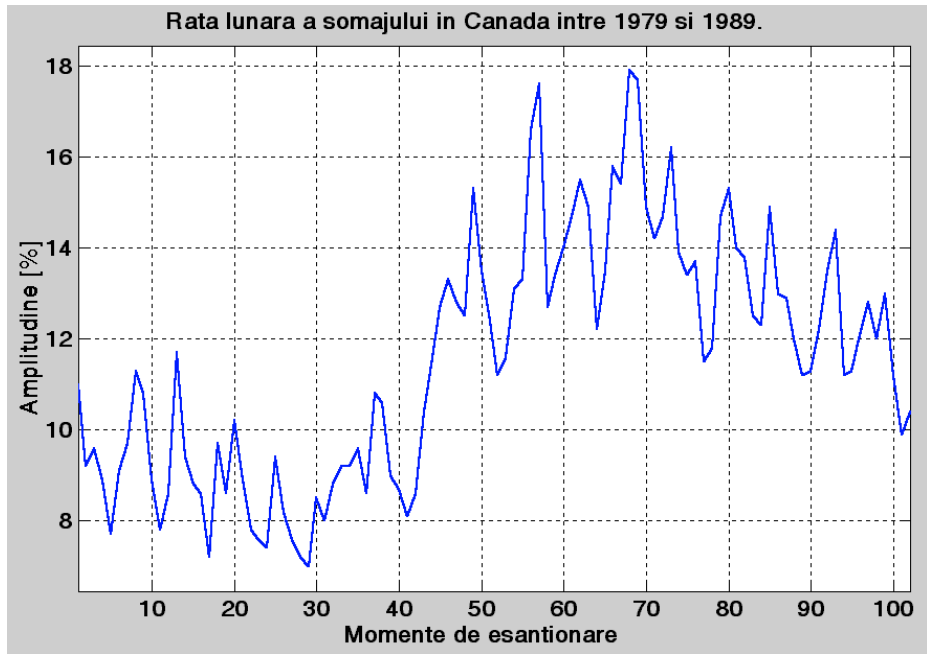


Figura 50. Rata lunară a șomajului în Canada.

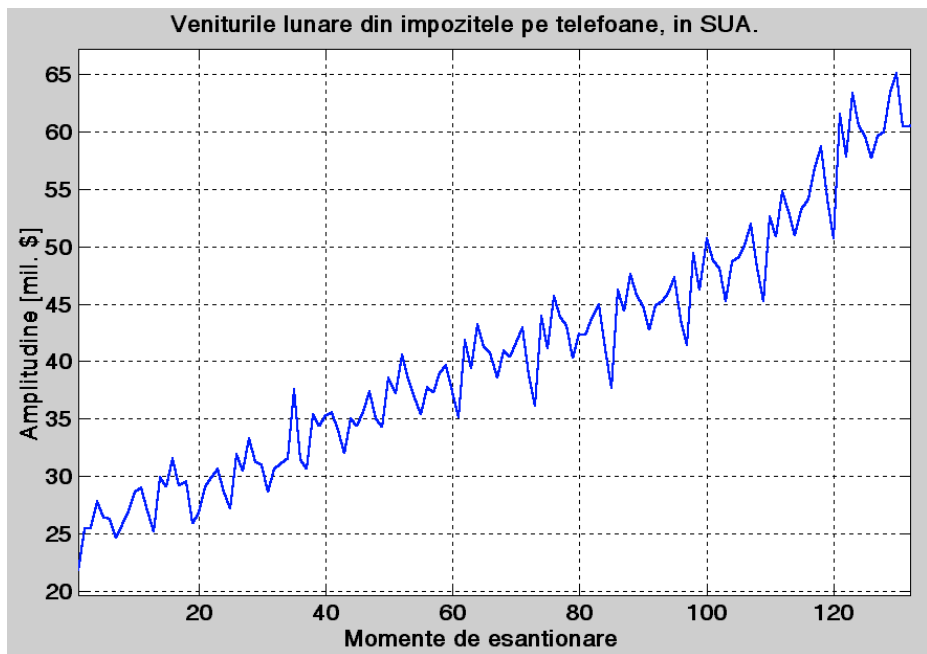


Figura 51. Veniturile lunare din impozitele pe telefoane în SUA.

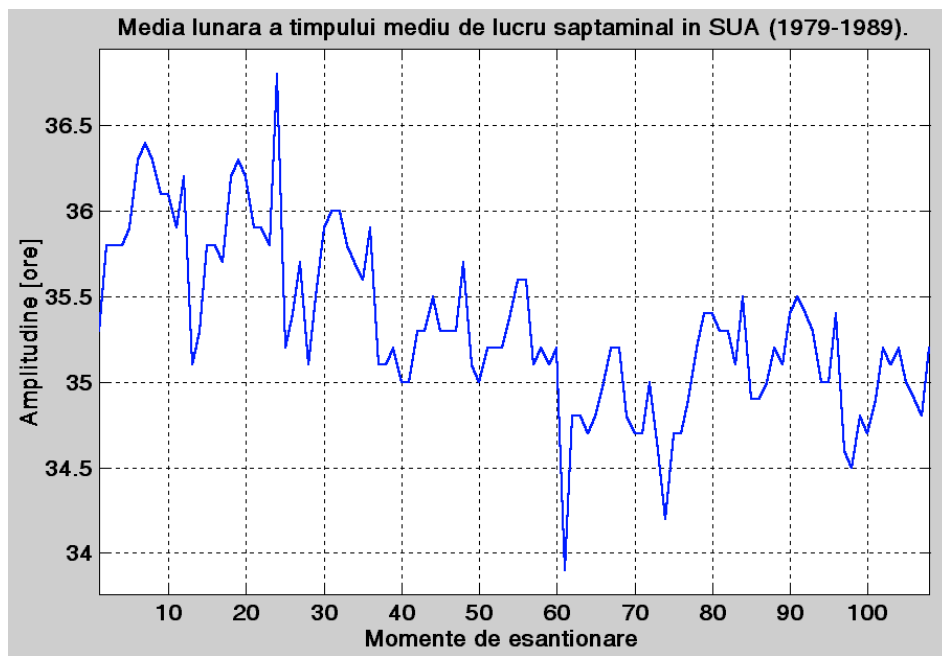


Figura 52. Media lunară a timpului mediu de lucru săptăminal în SUA.

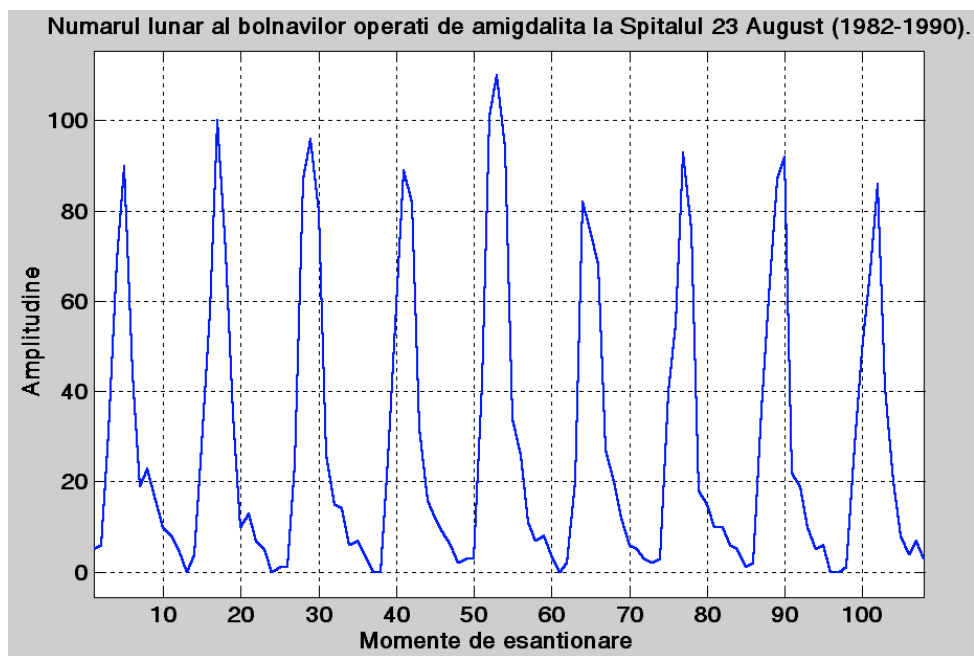


Figura 53. Numărul lunar al bolnavilor operați de amigdalită la Spitalul 23 August.



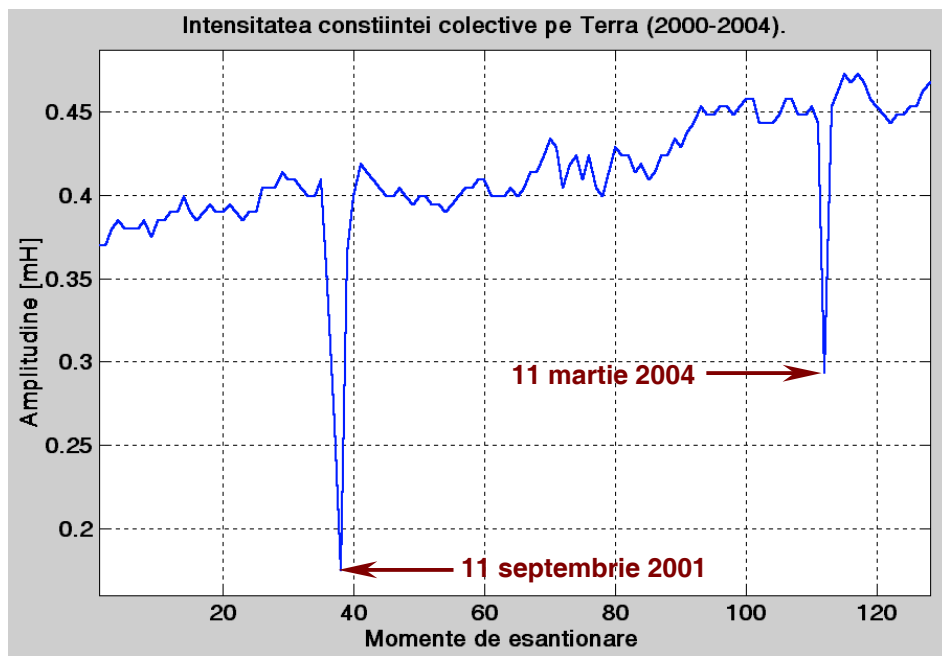


Figura 54. Intensitatea conștiinței colective pe Terra măsurată lunar.

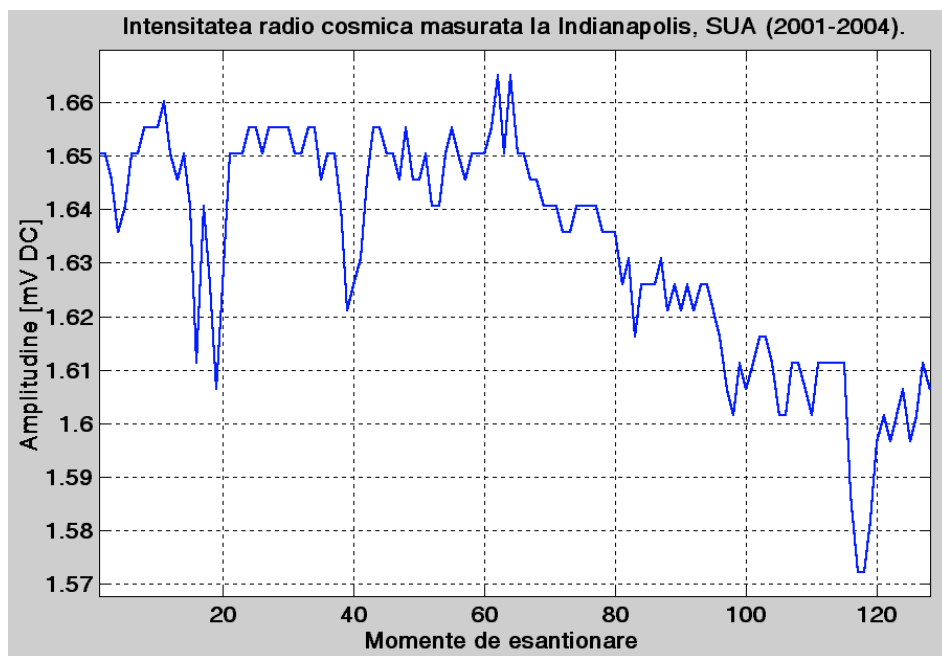


Figura 55. Intensitatea radio cosmică măsurată la radio-telescopul din Indianapolis.

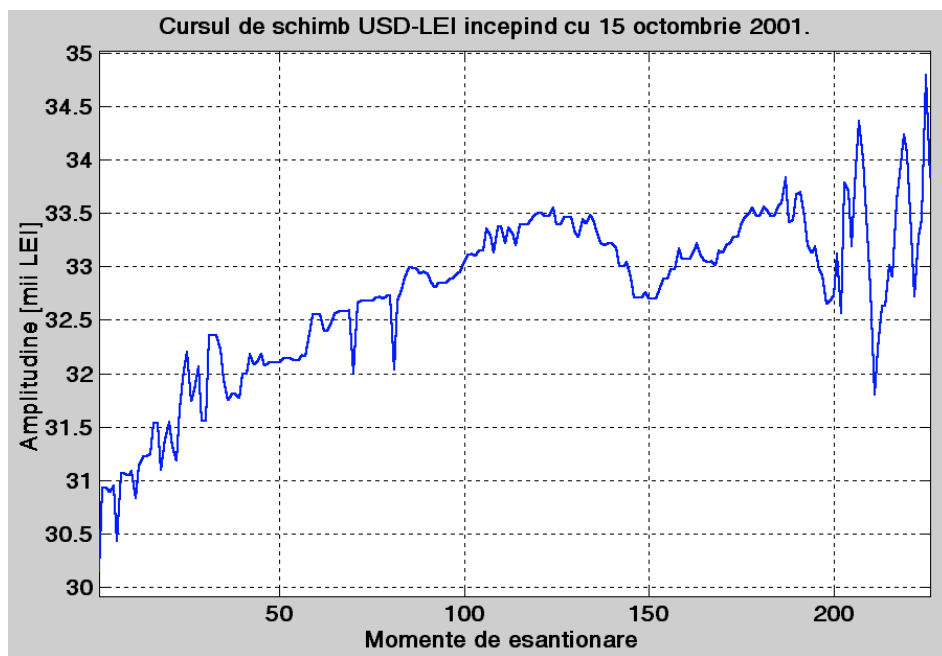


Figura 56. Cursul de schimb USD-LEI (eșantionare neuniformă).

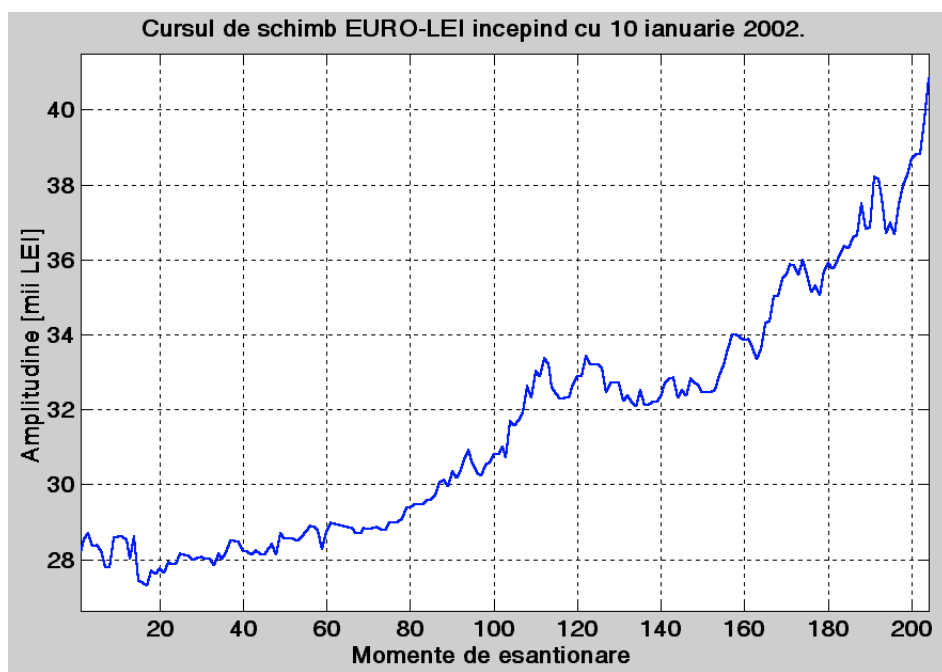


Figura 57. Cursul de schimb EURO-LEI (eșantionare neuniformă).

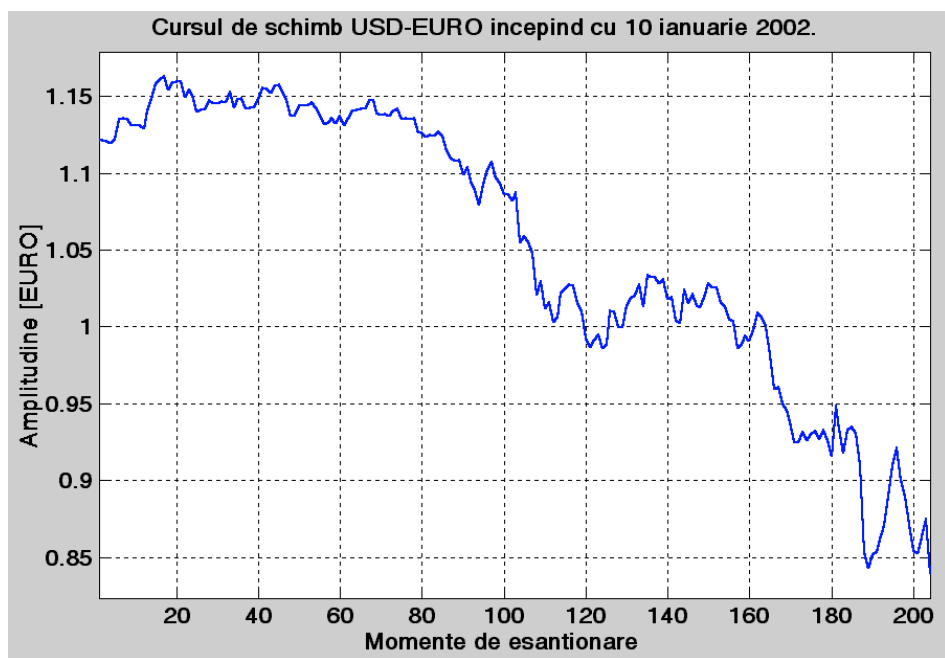


Figura 58. Cursul de schimb USD-EURO (eșantionare neuniformă).

## Anexa A

### Despre biblioteca de rutine MATLAB dedicate Identificării Sistemelor

Biblioteca de Identificare a Sistemelor din cadrul mediului de programare MATLAB 6.\* (adică *System Identification Toolbox*) este scrisă folosind tehnologia programării orientate obiect și conține numeroare rutine extrem de utile care pot fi direct apelate. Rutinele corespund în principiu conceptelor și metodelor specifice domeniului IS, așa cum sunt descrise în principal în [SoSt89] și [LjL99].

Primele informații despre utilizarea bibliotecii se pot obține cu ajutorul comenzilor: **idhelp** (specifică bibliotecii) sau **helpwin** (generală), urmată de selectarea bibliotecii din colecția de biblioteci afișate.

Prima comandă (**idhelp**) afișează un mico-manual al bibliotecii, pe care îl vom reproduce și noi în această anexă. A doua comandă (**helpwin**) este însă recomandată pentru accesarea informațiilor detaliate legate de rutinele bibliotecii. De notat că biblioteca deține și o interfață grafică convivială (*Graphical User Interface – GUI*) care ar putea fi utilizată pentru demonstrarea unor rezultate de simulare.

Comanda generică de estimare a parametrilor unui model este următoarea:

**Mid = id\_function(D,Ms) ;**

unde: **id\_function** este funcția de identificare utilizată (de exemplu: **pem** – folosind Metoda Minimizării Erorii de Predicție; **arx** – folosind un model ARX și MCMMP; **iv** – folosind un model ARX și MVI, etc.).

**D** este un obiect de tip date de identificare (**IDDATA**), descris de exemplu în contextul **Problemei 2.3**; pentru a obține mai multe informații, se poate executa comanda **idhelp data**.

**Ms** este o variabilă care definește structura modelului; comanda **idhelp model** oferă informații despre tipurile de structuri de modele acceptate în cadrul bibliotecii; în principiu, există 4 mari categorii de modele:

1. Cutie neagră de tip intrare ieșire (**idhelp iobb**).
2. Cutie neagră de tip reprezentare pe stare (**idhelp ssbb**).
3. Cutie neagră în timp continuu (**idhelp ssct**).
4. Cutie neagră de tip reprezentare pe stare cu structură internă definită de utilizator fie în timp continuu, fie în timp discret (**idhelp ssstruct**).

**Mid** este modelul rezultat în urma identificării, un obiect de tipul model de identificare (**IDMODEL**, descris de exemplu în contextul **Problemei 3.3**); pentru informații suplimentare, se poate executa comanda **idhelp evaluate** (care va ilustra modul în care poate fi evaluat/comparat modelul).

De notat că biblioteca a fost concepută pentru operarea cu modele MIMO, în cazul unei colecții de experimente de identificare. Aceasta înseamnă că obiectele construite

### Aspecte practice în Modelarea și Identificarea Sistemelor

conțin informații atât despre numărul de intrări și ieșiri, cât și despre experimente (indicele experimentului curent, numele său (dacă este cazul), numărul total de experimente, etc.). Filozofia de identificare a modelelor MIMO este în principiu următoarea: pentru fiecare canal de intrare și de ieșire este propus un model SISO corespunzător datelor achiziționate și informației de structură indicate/determinate. Ansamblul lor este apoi compactat într-o matrice constituind modelul MIMO operațional. Astfel, de exemplu, unui proces ARX cu 3 intrări și 2 ieșiri i se vor propune un număr de  $3 \times 2 = 6$  modele SISO, ansamblul lor fiind integrat într-o matrice cu 2 linii și 3 coloane:

$$H(q^{-1}) = \begin{bmatrix} \frac{B_{11}(q^{-1})}{A_{11}(q^{-1})} & \frac{B_{12}(q^{-1})}{A_{12}(q^{-1})} & \frac{B_{13}(q^{-1})}{A_{13}(q^{-1})} \\ \frac{B_{21}(q^{-1})}{A_{21}(q^{-1})} & \frac{B_{22}(q^{-1})}{A_{22}(q^{-1})} & \frac{B_{23}(q^{-1})}{A_{23}(q^{-1})} \end{bmatrix}. \quad (196)$$

Pentru a simula funcționarea modelului, se apelează la Principiul superpoziției: toate contribuțiile care afectează o anumită intrare sunt adunate. În particular, pentru exemplul anterior, ieșirea simulată este produsă după următoarea relație:

$$y_M \equiv \begin{bmatrix} y_{M,1} \\ y_{M,2} \end{bmatrix} \equiv \begin{bmatrix} \frac{B_{11}(q^{-1})}{A_{11}(q^{-1})} & \frac{B_{12}(q^{-1})}{A_{12}(q^{-1})} & \frac{B_{13}(q^{-1})}{A_{13}(q^{-1})} \\ \frac{B_{21}(q^{-1})}{A_{21}(q^{-1})} & \frac{B_{22}(q^{-1})}{A_{22}(q^{-1})} & \frac{B_{23}(q^{-1})}{A_{23}(q^{-1})} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \quad (197)$$

Ecuția (197) simulează un comportament mai aproape de cel real al procesului furnizor de date dacă elementele matricii  $H$  ar fi determinate simultan din datele de intrare-ieșire măsurate și nu pe fiecare canal în parte. Aceasta conduce însă la proceduri de identificare mai complicate, de complexitate proporțională cu numărul de intrări și ieșiri ale modelului ales. De exemplu, un model MIMO-ARX cu 2 intrări și 2 ieșiri poate fi identificat cu ajutorul a 2 modele BJ:

$$y \equiv \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \equiv \begin{bmatrix} \frac{B_{11}(q^{-1})}{A_{11}(q^{-1})} & \frac{B_{12}(q^{-1})}{A_{12}(q^{-1})} \\ \frac{B_{21}(q^{-1})}{A_{21}(q^{-1})} & \frac{B_{22}(q^{-1})}{A_{22}(q^{-1})} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} A_{11}(q^{-1})A_{12}(q^{-1})e_1 \\ A_{21}(q^{-1})A_{22}(q^{-1})e_2 \end{bmatrix} \Leftrightarrow$$

$$\Leftrightarrow \begin{cases} y_1 \equiv \frac{B_{11}(q^{-1})}{A_{11}(q^{-1})}u_1 + \frac{B_{12}(q^{-1})}{A_{12}(q^{-1})}u_2 + v_1 \\ y_2 \equiv \frac{B_{21}(q^{-1})}{A_{21}(q^{-1})}u_1 + \frac{B_{22}(q^{-1})}{A_{22}(q^{-1})}u_2 + v_2 \end{cases}. \quad (198)$$

Determinarea parametrilor modelelor BJ (198) conduce la un model mai precis decât cel obținut prin identificarea câte unei funcții de transfer pentru fiecare canal separat,

datorită faptului că, în general, modelele MIMO nu prezintă decuplări între canale. Modelul (197) funcționează bine doar în cazul decuplării totale, când, de fapt, matricea (196) este diagonală.

Sub-modele ale unui model MIMO pot fi de asemenea selectate (vezi **idhelp channels** pentru mai multe detalii).

Pentru a începe lucrul cu biblioteca de IS, este recomandat să se țină cont de următoarele sugestii (obținute prin execuția comenzii **idhelp advice**):

- Utilizatorii începători sunt invitați să opereze cu ajutorul interfeței grafice conviviale (GUI). Aceasta se lansează cu comanda **ident**. Fereastra grafică de bază afișată este ilustrată în Figura 59. Înainte de prima utilizare, este bine să se selecteze opțiunea **Demo of the Toolbox** din meniul **Help** al ferestrei.

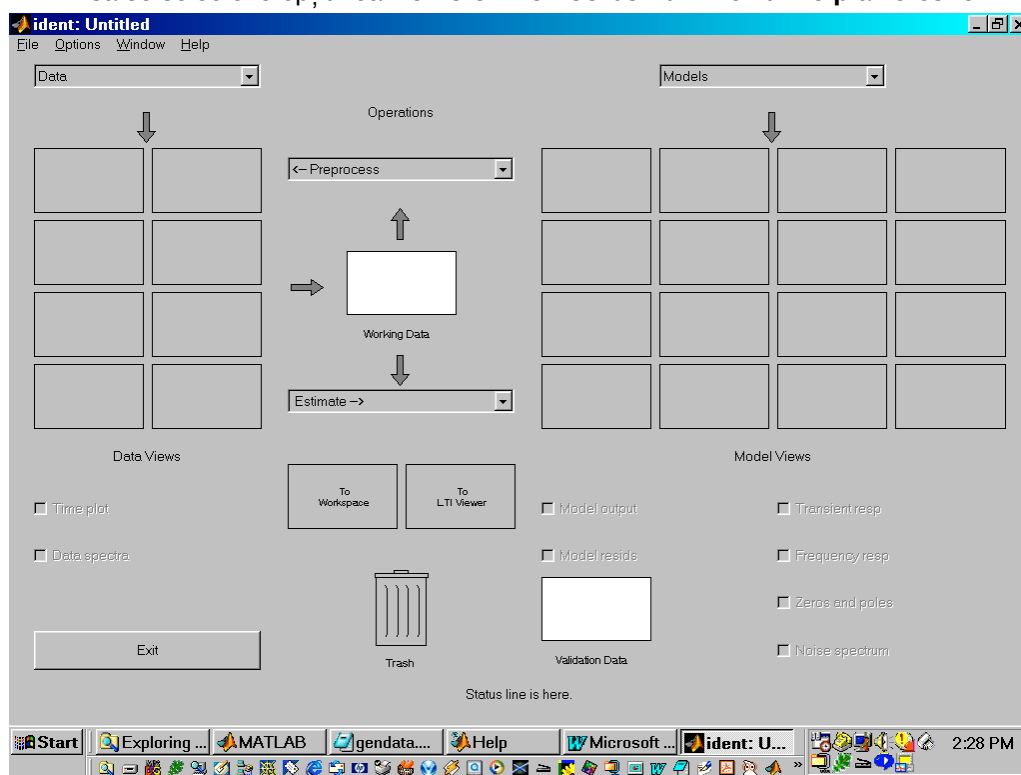


Figura 59. Fereastra grafică tipică a interfeței bibliotecii de IS.

- În cursul utilizării interfeței, se va acorda atenție următoarelor aspecte:
  - a. Inspectarea datelor achiziționate pentru a sesiza și înlătura erorile grosolane.
  - b. Datele ar trebui segmentate în 2 submulțimi. Prima va fi utilizată pentru estimarea parametrilor, în timp ce a doua va fi utilizată pentru validarea modelelor identificate. O rutină de bibliotecă capabilă să efectueze validarea modelelor este **compare**.
  - c. În cazul în care datele au o proveniență necunoscută sau au fost generate de un proces neliniar, este bine ca, la început, să se testeze cât de adecvată

este utilizarea unui model liniar. Acesta se poate realiza prin alegerea unui model reprezentat pe stare, identificat cu ajutorul uneia din comenzile următoare: `Mid = pem(Di)` sau `Mid = n4sid(Di)`, unde `Di` reprezintă segmentul de date dedicat identificării. Celălalt segment de date, `Dv`, dedicat validării poate fi folosit la pasul următor pentru a testa adecvanța modelului (adică maniera în care un sistem liniar poate reproduce datele). Aceasta se realizează cu comanda: `compare(Dv,Mid)`.

- d. Înainte de alegerea structurii modelului este bine să se detecteze întârzierile intrinseci cu ajutorul funcțiilor `impulse` sau `step` aplicate datelor și modelelor identificate cu diferiți indici structurali. Întârzierile detectate pot fi apoi specificate în model prin intermediul variabilei `nk`.
- e. Pentru modelele MIMO suficient de complexe, este mai eficient să se elaboreze o strategie de identificare în care să se testeze mai mulți indici structurali și mai multe întârzieri posibile. Aceasta poate fi mare consumatoare de timp, însă. Pentru a remedia (fie și parțial) acest neajuns, se poate utiliza executa comanda:

```
Mid = n4sid(Di,18,'cov','none') ;
```

O altă comandă, `idhelp note`, oferă câteva informații generale suplimentare, cum ar fi:

- Tehnologia programării orientate obiect permite funcțiilor să partajeze același nume cu funcții din alte biblioteci sau din nucleul mediului de programare MATLAB. Pentru a obține informațiile explicative referitoare la funcțiile din biblioteca de IS prin comanda `help`, este indicat ca numele funcției să fie precedat de numele `idmodel/`. De exemplu: `help step` va oferi informații referitoare la funcția `step` din cadrul bibliotecii de rutine dedicate domeniului TS, în timp ce `help idmodel/step` va conduce la afișarea informației referitoare la funcția `step` din biblioteca de IS.
- Unele proprietăți de estimare sunt moștenite de obiecte pe tot parcursul sesiunii MATLAB, chiar dacă funcția care le-a folosit si-a încheiat execuția. De exemplu, să presupunem că dorim identificarea unui model ARMAX după maximum 5 iterații:

```
Mid = armax(Di,[2 2 2 1],'MaxIter',5) ;
```

Numărul maxim de iterații este în acest caz moștenit și de obiectul `Mid` (modelul de identificare rezultat). Re-identificarea aceluiași model pentru un nou set de date se poate efectua cu comanda:

```
Mid = armax(Di_new,Mid) ;
```

În acest caz, tot maximum 5 iterații sunt efectuate. Dacă se dorește schimbarea acestei proprietăți de la 5 la 20 de iterații se poate proceda în unul din următoarele moduri:

```
Mid = armax(Di_new,Mid,'maxi',20) ; (indirect)
```

```
Mid.Algorithm.MaxIter = 20 ; (direct)
```

```
Mid = armax(Di_new,Mid) ;
```

- Dacă algoritmului de identificare i s-au setat alte opțiuni decât cele implicite și se dorește reutilizarea acestuia cu opțiunile ne-implicite, configurația opțiunilor poate fi salvată în cadrul unei structuri virtuale de opțiuni. Aceasta poate fi ulterior folosită ori de câte ori este necesar. De exemplu, să presupunem că algoritmului de Minimizare a Erorii de leșire i s-au setat următoarele opțiuni:

```
Mid = oe(Di,[2 2 1], 'lim',0, 'maxi',40, 'tol',0.00001) ;
```

Acestea sunt întâi salvate într-o structură virtuală de opțiuni:

```
myalg = Mid.Algorithm ;
```

apoi reutilizate:

```
Mid_new = pem(Di_new,3, 'alg',myalg) ;
```



## Anexa B

### Lista de verificare a mini-simulatoarelor și rutinelor MATLAB

#### Notă importantă

- În vederea afișării grafice în ferestre cu index controlat de utilizator, mini-simulatoarele deja proiectate operează cu variabila globală **FIG**, care trebuie inițializată înainte de rulare, din mediul MATLAB, astfel:

```
>> global FIG ;           % Declarație variabilă globală.  
>> FIG = n ;             % Inițializare (cu n dorit, e.g. 1).
```

**Legendă:** ☒ = mini-simulator (rutină) disponibil(ă)  
☐ = mini-simulator (rutină) ce trebuie proiectat(ă)

#### Capitolul 1

- ☒ **ISLAB\_1A** (mini-simulator)
- ☒ **ISLAB\_1B** (mini-simulator)
- ☒ **D\_SPEKTR** (rutină auxiliară)
- ☒ **NOISE** (rutină auxiliară)
- ☒ **SPEFAC** (rutină auxiliară)

#### Capitolul 2

- ☒ **ISLAB\_2A** (mini-simulator)
- ☐ **ISLAB\_2B** (mini-simulator)
- ☐ **ISLAB\_2C** (mini-simulator)
- ☒ **ISLAB\_2D** (mini-simulator)
- ☐ **ISLAB\_2E** (mini-simulator)
- ☐ **ISLAB\_2F** (mini-simulator)
- ☐ **ISLAB\_2G** (mini-simulator)
- ☐ **ISLAB\_2H** (mini-simulator)
- ☐ **ISLAB\_2I** (mini-simulator)
- ☐ **ISLAB\_2J** (mini-simulator)
- ☐ **ISLAB\_2K** (mini-simulator)

- ☒ **ISLAB\_2L** (mini-simulator)
- ☐ **ISLAB\_2M** (mini-simulator)
- ☐ **ISLAB\_2N** (mini-simulator)
- ☐ **ISLAB\_2O** (mini-simulator)
- ☐ **ISLAB\_2P** (mini-simulator)
- ☐ **ISLAB\_2Q** (mini-simulator)
- ☐ **ISLAB\_2R** (mini-simulator)
- ☐ **ISLAB\_2S** (mini-simulator)

### **Capitolul 3**

- ☐ **ISLAB\_3A** (mini-simulator)
- ☐ **ISLAB\_3B** (mini-simulator)
- ☐ **ISLAB\_3C** (mini-simulator)
- ☒ **ISLAB\_3D** (mini-simulator)
- ☐ **ISLAB\_3E** (mini-simulator)
- ☐ **ISLAB\_3F** (mini-simulator)
- ☐ **ISLAB\_3G** (mini-simulator)
- ☐ **ISLAB\_3H** (mini-simulator)
- ☐ **ISLAB\_3I** (mini-simulator)
- ☐ **ISLAB\_3J** (mini-simulator)

### **Capitolul 4**

- ☒ **ISLAB\_4A** (mini-simulator)
- ☐ **ISLAB\_4B** (mini-simulator)
- ☐ **ISLAB\_4C** (mini-simulator)
- ☒ **F\_TEST2** (rutină auxiliară)
- ☒ **GAIC\_R2** (rutină auxiliară)
- ☒ **GENDATA** (rutină auxiliară)
- ☒ **VALID\_LS** (rutină auxiliară)
- ☐ **VALID\_IV** (rutină auxiliară)

### Capitolul 5

- ☒ **ISLAB\_5A** (mini-simulator)
- ☐ **ISLAB\_5B** (mini-simulator)
- ☐ **ARMAX\_E** (rutină auxiliară)
- ☐ **BJ\_E** (rutină auxiliară)
- ☒ **GAIC\_R3** (rutină auxiliară)
- ☒ **GEN\_DATA** (rutină auxiliară)
- ☒ **VALID\_LS** (rutină auxiliară)

### Capitolul 6

- ☒ **ISLAB\_6A** (mini-simulator)
- ☐ **ISLAB\_6B** (mini-simulator)
- ☐ **ISLAB\_6C** (mini-simulator)
- ☐ **ISLAB\_6D** (mini-simulator)
- ☒ **GDATA\_VP** (rutină auxiliară)
- ☒ **RIV** (rutină auxiliară)

### Capitolul 7

- ☒ **ISLAB\_7A** (mini-simulator)
- ☐ **ISLAB\_7B** (mini-simulator)
- ☒ **ISLAB\_7C** (mini-simulator)
- ☐ **ISLAB\_7D** (mini-simulator)
- ☒ **ISLAB\_7E** (mini-simulator)
- ☐ **ISLAB\_7F** (mini-simulator)
- ☒ **GDATA\_FP** (rutină auxiliară)
- ☒ **GDATA\_CP** (rutină auxiliară)

### Capitolul 8

- ☐ **ISLAB\_8A** (mini-simulator)
- ☐ **ISLAB\_8B** (mini-simulator)
- ☐ **ISLAB\_8C** (mini-simulator)
- ☐ **ISLAB\_8D** (mini-simulator)

## Referințe bibliografice

- [AkH69] Akaike H. – *Fitting Autoregressive Models for Prediction*, Ann. of Institute for Statistical Mathematics, Vol. 21, pp. 243-247, 1969.
- [AsEy71] Åström K.J., Eykhoff P. – *System Identification – A Survey*, Automatica, Vol. 7, pp. 123-167, 1971.
- [CaDL96] Carter D.L. – *Rolling Element Bearing Condition Testing Method and Apparatus*, United States Patent No. 5,477,730, December 26, 1996.  
URL: [www.uspto.gov/go/ptdl](http://www.uspto.gov/go/ptdl)
- [DuHa96] Dumitrescu D., Hariton C. – *Rețele Neuronale – Teorie și Aplicații*, Editura TEORA, București-Sibiu, România, 1996.
- [EyP74] Eykhoff P. – *System Identification: Parameter and State Estimation*, Wiley, London, UK, 1974.
- [EyP81] Eykhoff P. – *Trends and Progress in System Identification*, Pergamon Press, Oxford, UK, 1981.
- [GoPa77] Goodwin G.C., Payne R.L. – *Dynamic System Identification: Experiment Design and Data Analysis*, Academic Press, New York, USA, 1977.
- [HaS86] Haykin S. – *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1986.
- [IFAC80] IFAC – *Tutorial Section on System Identification*, Automatica, Vol. 16, 1980.
- [IFAC82] IFAC – *Special Issue on System Identification*, Automatica, Vol. 18, 1982.
- [IoV85] Ionescu V. – *Teoria Sistemelor. Sisteme Liniare.*, Editura Didactică și Pedagogică, București, România, 1985.
- [KaRa76] Kashyap R.L., Rao A.R. – *Dynamic Stochastic Models from Empirical Data*, Academic Press, New York, USA, 1976.
- [LaID93] Landau I.D. – *Identification et Commande des Systèmes*, Hermès, Paris, France, 1993.
- [LaID97] Landau I.D. – *Identificarea și Comanda Sistemelor* (traducere în limba română), Editura Tehnică, București, România, 1997.
- [LjGI94] Ljung L., Glad T. – *Modeling of Dynamic Systems*, Prentice Hall, Englewood Cliffs, N.J., 1994.
- [LjL99] Ljung L. – *System Identification - Theory for the User*, Prentice Hall, Upper Saddle River, N.J., 2<sup>nd</sup> edition, 1999.
- [MeLa76] Mehra R.K., Lainiotis D.G. – *System Identification – Advances and Case Studies*, Academic Press, New York, USA, 1976.
- [MiM95] Mitchell M. – *An Introduction to Genetic Algorithms*, The MIT Press, Cambridge, Massachusetts, USA, 1995.
- [OpSc85] Oppenheim A.V., Schaffer R. – *Digital Signal Processing*, Prentice Hall, New York, USA, 1985.
- [OpWi85] Oppenheim A.V., Willsky A.S. – *Signals and Systems*, Prentice Hall, Englewood Cliffs, N.J. 1985.

#### Aspecte practice în Modelarea și Identificarea Sistemelor

---

- [PITC71] Penescu C., Ionescu G., Tertișco M., Ceangă E. – *Identificarea Experimentală a Proceselor Automatizate*, Editura Tehnică, București, România, 1971.
- [PrMa96] Proakis J.G., Manolakis D.G. – *Digital Signal Processing. Principles, Algorithms and Applications.*, third edition, Prentice Hall, Upper Saddle River, New Jersey, USA, 1996.
- [RiJ78] Rissanen J. – *Modeling by Shortest Data Description*, Automatica, No. 14, pp. 465-471, 1978.
- [RiJ83] Rissanen J. – *A Universal Data Compression System*, IEEE Transactions on Information Theory, Vol. IT-29, pp. 656-664, 1983.
- [RuNo95] Russel S.J., Norvig P. – *Artificial Intelligence – A Modern Approach*, Prentice Hall, Upper Saddle River, New Jersey, USA, 1995.
- [SoSt89] Söderström T., Stoica P. – *System Identification*, Prentice Hall, London, UK, 1989.
- [StF00] Stratulat F. – *Teoria Sistemelor. Analiza Asistată de Calculator a Sistemelor Liniare.*, MATRIX-ROM, București, România, 2000.
- [SeS01] Șerban S. – *Sisteme Dinamice Liniare – Aplicații Numerice*, Printech, București, România, 2001.
- [SeS02] Șerban S. – *Sisteme Liniare*, Printech, București, România, 2002.
- [StD99] Ștefănoiu D. – *On Non Uniform Sampling of Signals*, CSCS-12 International Conference, Bucharest, Romania, pp. 405-410, May 25-30, 1999.
- [StD9602a] Ștefănoiu D. – *Introducere în Prelucrarea Numerică a Semnalelor* (note de curs), Centrul de multiplicare al Universității “Politehnica” din București, România, Februarie 1996.
- [StD9602b] Ștefănoiu D. – *Tehnici de Calcul în Prelucrarea Numerică a Semnalelor* (note de curs și îndrumar de laborator), Centrul de multiplicare al Universității “Politehnica” din București, România, Februarie 1996.
- [StD9603] Ștefănoiu D. – *Identificarea Experimentală a Sistemelor – Serii de Timp* (îndrumar de laborator), Centrul de multiplicare al Universității “Politehnica” din București, România, Martie 1996.
- [StD9605] Ștefănoiu D. – *Identificarea Experimentală a Sistemelor – Probleme de Seminar*, Centrul de multiplicare al Universității “Politehnica” din București, România, Mai 1996.
- [Tal97] Tăbuș I. ș.a. – *Commande Numérique et Intelligence Artificielle en Automatique* (capitolul: *Réseaux de Neurones*), Editura Tehnică, București, România, 1997.
- [TeSt80] Tertișco M., Stoica P. – *Identificarea și Estimarea Parametrilor Sistemelor*, Editura Didactică și Pedagogică, București, România, 1980.
- [TeSt85] Tertișco M., Stoica P. – *Modelarea și Predicția Seriilor de Timp*, Editura Academiei Române, București, România, 1985.
- [TSP87] Tertișco M., Stoica P., Popescu Th. – *Identificarea Asistată de Calculator a Sistemelor*, Editura Tehnică, București, România, 1987.