

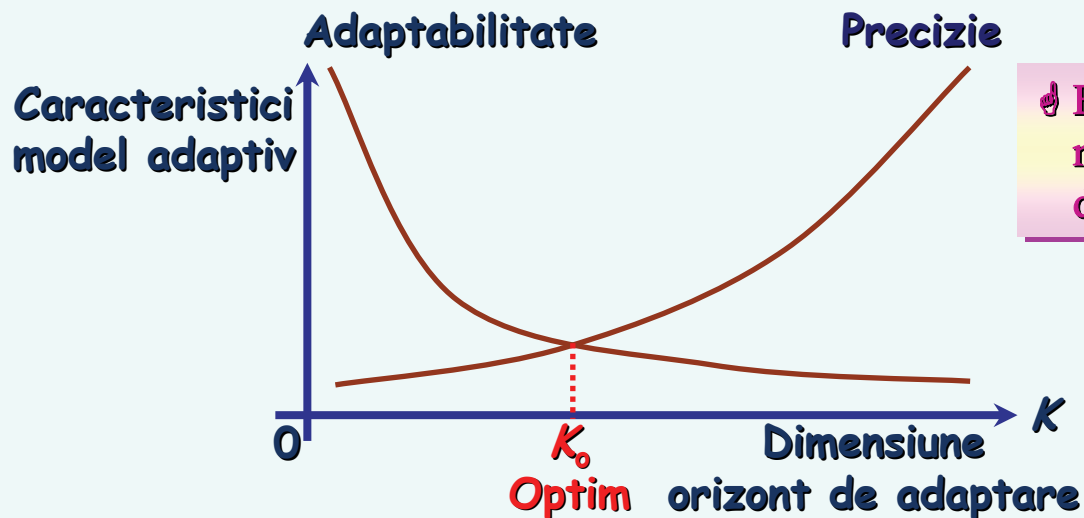


# ⑥ Algoritmi rapizi CMMP-R și VI-R fără fereastră

## Contextul de lucru

10 p

- Majoritatea proceselor furnizoare de date **sunt neliniare** și/sau **posedă parametri variabili în timp**.
- Identificarea proceselor cu parametri variabili în timp se realizează cu ajutorul **modelelor și metodelor adaptive (recursive)**.
- Prin identificare recursivă, se urmărește **asigurarea unui compromis între două caracteristici opuse** ale estimației parametrilor necunoscuți (variabili pe orizontul de măsură):



Estimația vectorului parametrilor necunoscuți se reactualizează folosind datele măsurate pe orizontul de adaptare.

$$\hat{\theta}_{K \lfloor \frac{k}{K} \rfloor} = \hat{\theta}_{K \lfloor \frac{k-1}{K} \rfloor} + \Delta_{K \lfloor \frac{k}{K} \rfloor} \quad \forall k \in \mathbb{N}^*$$

Corecție

Adaptabilitatea **scade**, în timp ce **precizia crește** odată cu dimensiunea orizontului de adaptare.

Cu cât se achiziționează **mai multe date** între momentele de reactualizare, cu atât **adaptarea se efectuează mai rar**, modelul fiind incapabil să surprindă variațiile caracteristicilor procesului între aceste momente.

În schimb, **precizia modelului crește**, deoarece parametrii săi sunt determinați cu ajutorul unui set mai bogat de date.





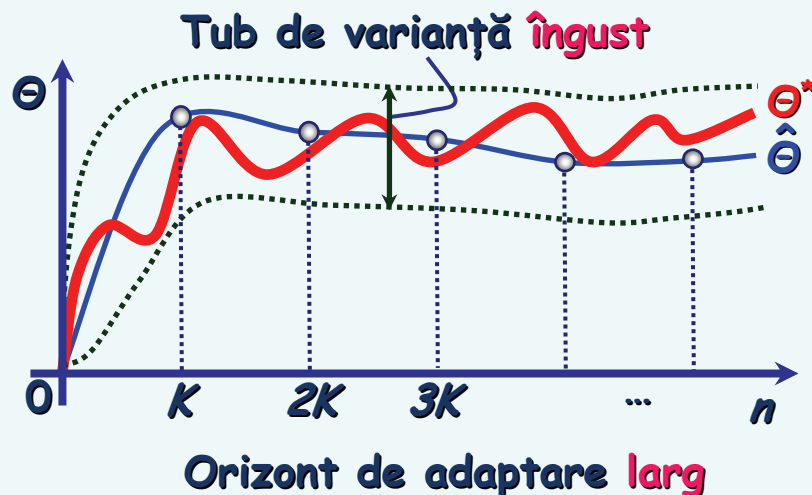
# Algoritmi rapizi CMMP-R și VI-R fără fereastră

## Contextul de lucru

☹️ Asigurarea compromisului precizie-adaptabilitate este dificilă.

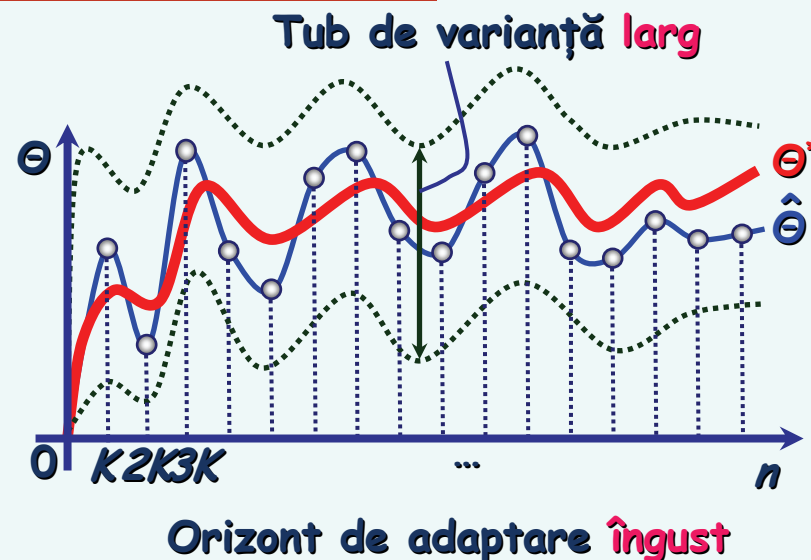
### Exemplu

Cazul parametrului scalar, variabil în timp.



😊 Valorile estimate ale parametrului sunt relativ apropiate de cele adevărate și tubul de varianță este relativ îngust.

☹️ Graficul parametrului estimat este neted, deci modelul sesizează mai puțin variațiile locale ale parametrului adevărat.



☹️ Valorile estimate ale parametrului sunt relativ depărtate de cele adevărate și tubul de varianță este relativ larg.

😊 Graficul parametrului estimat urmărește variațiile locale ale parametrului adevărat, cu o anumită acuratețe.



$K = ?$



Se sacrifică precizia în favoarea adaptabilității.

$K = 1$

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \Delta_k \quad \forall k \in \mathbb{N}^*$$



# ⑥ Algoritmi rapizi CMMP-R și VI-R fără fereastră

## ☞ Contextul de lucru

### Algoritmul recursiv CMMP/VI de bază în IS

#### ➤ Date de intrare:

a. ordinele modelului de identificare:  $n_a$ ,  $n_b$ ,  $n_c$ ,  $n_d$  și  $n_f$ ;

b. o colecție redusă de date intrare-ieșire măsurate (dacă este posibil):

$$\mathcal{D}_{N_0} = \{u[n]\}_{n \in \overline{1, N_0}} \cup \{y[n]\}_{n \in \overline{1, N_0}} \text{ (cu } N_0 \text{ de ordinul zecilor cel mult);}$$

c. un semnal instrumental extern:  $\{f[n]\}_{n \geq 1}$  (eventual).

1. Dacă nu a fost specificat nici un semnal instrumental, vectorul variabilelor instrumentale,  $\zeta$ , este identic cu vectorul regresorilor  $\varphi$ . Altfel,  $\zeta$  este definit ca în cazul MVI, dar folosind în general semnalul instrumental extern  $f$  în locul intrării  $u$  (în particular, este posibil ca  $f \equiv u$ ).

2. 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).

3. Pentru  $k \geq 1$ :

3.1. Se evaluează eroarea de predicție curentă:  $\varepsilon[k] = y[k] - \varphi^T[k] \hat{\theta}_{k-1}$ .

3.2. Se evaluează vectorul auxiliar:  $\xi_k = P_{k-1} \zeta[k]$ .

3.3. Se evaluează câștigul de senzitivitate:  $\gamma_k = \frac{\xi_k}{1 + \varphi^T[k] \xi_k}$ .

3.4. 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).

3.5. Se reactualizează vectorul parametrilor:  $\hat{\theta}_k = \hat{\theta}_{k-1} + \gamma_k \varepsilon[k]$ .

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



# ⑥ Algoritmi rapizi CMMP-R și VI-R fără fereastră

## ☞ Contextul de lucru

### Algoritmul recursiv CMMP/VI de tip gradient

#### ➤ Date de intrare:

a. ordinele modelului de identificare:  $n_a$  ,  $n_b$  ,  $n_c$  ,  $n_d$  și  $n_f$  ;

b. câștigul de gradient:  $\gamma \in \mathbb{R}_+^*$  ;

c. o colecție redusă de date intrare-ieșire măsurate (dacă este posibil):

$$\mathcal{D}_{N_0} = \{u[n]\}_{n \in \overline{1, N_0}} \cup \{y[n]\}_{n \in \overline{1, N_0}} \text{ (cu } N_0 \text{ de ordinul zecilor, cel mult);}$$

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

1. Dacă nu a fost specificat nici un semnal instrumental, vectorul variabilelor instrumentale,  $\zeta$  , este identic cu vectorul regresorilor  $\phi$  . Altfel,  $\zeta$  este definit ca în cazul MVI, dar folosind în general semnalul instrumental extern  $f$  în locul intrării  $u$  (în particular, este posibil ca  $f \equiv u$  ).

2. 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  $P_0 = \gamma I_{n_\theta}$  .

3. Pentru  $k \geq 1$  :

3.1. Se evaluează eroarea de predicție:  $\varepsilon[k] = y[k] - \phi^T[k] \hat{\theta}_{k-1}$  .

3.2. Se reactualizează matricea  $P_k$  astfel:  $P_k = P_0$  (gradient ne-normalizat) sau

$$P_k = P_0 / \|\phi[k]\|^2 \text{ (gradient normalizat).}$$

3.3. Se reactualizează vectorul parametrilor:  $\hat{\theta}_k = \hat{\theta}_{k-1} + P_k \phi[k] \varepsilon[k]$  .

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





# ⑥ Algoritmi rapizi CMMP-R și VI-R fără fereastră

## 👉 Contextul de lucru

### Algoritmul recursiv CMMP/VI cu filtrare de tip Kalman-Bucy

#### ➤ Date de intrare:

- ordinele modelului de identificare:  $n_a$  ,  $n_b$  ,  $n_c$  ,  $n_d$  și  $n_f$  ;
- matricea de răspîndire:  $R_v > 0$  ;
- dispersia estimată a zgomotului alb de proces:  $\lambda^2$  ;
- 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:  $\{f[n]\}_{n \geq 1}$  (eventual).

1. Dacă nu a fost specificat nici un semnal instrumental, vectorul variabilelor instrumentale,  $\zeta$  , este identic cu vectorul regresorilor  $\phi$  . Altfel,  $\zeta$  este definit ca în cazul MVI, dar folosind în general semnalul instrumental extern  $f$  în locul intrării  $u$  (în particular, este posibil ca  $f \equiv u$  ).

2. 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).

3. Pentru  $k \geq 1$  :

3.1. Se evaluează eroarea de predicție:  $\varepsilon[k] = y[k] - \phi^T[k] \hat{\theta}_{k-1}$  .

3.2. Se evaluează vectorul auxiliar:  $\xi_k = P_{k-1} \zeta[k]$  .

3.3. Se evaluează câștigul de senzitivitate:  $\gamma_k = \frac{\xi_k}{\lambda^2 + \phi^T[k] \xi_k}$  .

3.4. Se reactualizează matricea  $P_k$  , adică:  $P_k = P_{k-1} + R_v - \gamma_k \phi^T[k] P_{k-1}$  (cu evitarea inversării explicite a matricilor).

3.5. Se reactualizează vectorul parametrilor:  $\hat{\theta}_k = \hat{\theta}_{k-1} + \gamma_k \varepsilon[k]$  .

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





# Algoritmi rapizi CMMP-R și VI-R fără fereastră

## Contextul de lucru

### Procese generatoare de date

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

#### Parametri constanți

$$a_1[n] = a_{10} = -0.7$$

$$b_1[n] = b_{10} = 0.6$$

$$c_1[n] = c_{10} = -0.9$$

$$\forall n \in \mathbb{N}^*$$

#### Parametri variabili

$$a_1[n] \stackrel{\text{def}}{=} a_{10} \cos\left(\frac{10\pi n}{N}\right) \quad b_1[n] \stackrel{\text{def}}{=} b_{10} \operatorname{sgn}\left[\cos\left(\frac{4\pi n}{N}\right)\right]$$

$$c_1[n] \stackrel{\text{def}}{=} c_{10} \operatorname{Sc}\left(\frac{18\pi n}{N}\right)$$

$$\forall n \in \mathbb{N}^*$$

$e$  → SPAB Gaussian sau bipolar de medie nulă și dispersie unitară

Date generate  $\mathcal{D} = \{u[n]\}_{n=1, \overline{N}} \cup \{y[n]\}_{n=1, \overline{N}}$

ales liber de către utilizator →  $N \geq 200$

Pe un orizont mare de timp, modelul ARMAX tinde să devină un model ARX.

Indici structurali sunt cunoscuți  $na = nb = nc = 1$

Test de stop → Epuizarea datelor de pe orizontul de măsură.

### Obiectiv

- Compararea performanțelor metodelor recursive de identificare fără fereastră, în cazul modelelor din clasa ARMAX.







# ⑥ Algoritmi rapizi CMMP-R și VI-R fără fereastră

## Probleme de simulare

### Contextul de lucru

#### Rutine preliminare [●●○○○]

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

**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 estimăția 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).







# ⑥ Algoritmi rapizi CMMP-R și VI-R fără fereastră

## ➔ Probleme de simulare

### Contextul de lucru

#### Rutine preliminare [●●●○○]

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

**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.

**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  $\phi[N]$ .





# ⑥ Algoritmi rapizi CMMP-R și VI-R fără fereastră

## ➡ Probleme de simulare

### Contextul de lucru

⚠ Inexistentă în biblioteca  
MATLAB-IS. Disponibilă pe site.

### Rutine preliminare [●●●●○]

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

MVI-R

**f** este semnalul instrumental (implicit:  $f=D \cdot u$ );  
**lambda** este factorul de uitare ( $\lambda \in (0,1]$ ) (implicit: **lambda=1**);  
**z0** este vectorul inițial al instrumentelor  $\zeta[0]$  ;  
**z** este vectorul final al instrumentelor  $\zeta[N]$ .

Restul argumentelor funcției au fost explicitate mai înainte, iar indicii structurali **si** conțin **numai ordinele modelului ARX**.

Cu toate acestea, algoritmul este implementat **numai în varianta cu fereastră exponențială** (de aceea argumentul **ma** lipsește).

`[D,V,P] = gdata_vp(cv,N,sigma,lambda,bin) ;` (generează date)

**cv** este un comutator care arată tipul de proces: cu parametri constanți (**cv=0**) sau cu parametri variabili (**cv~=0**); (implicit: **cv=0**);  
**N** este orizontul de măsură (implicit: **N=250**);  
**sigma** este deviația standard a intrării SPA (implicit: **sigma=1**).



# ⑥ Algoritmi rapizi CMMP-R și VI-R fără fereastră

## ➡ Probleme de simulare

### Contextul de lucru

#### Rutine preliminare [●●●●●]

`[D,V,P] = gdata_vp(cv,N,sigma,lambda,bin) ;` (generează date)

**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** 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; în cazul parametrilor constanți: **P.a=[1 a0]**, **P.b=[0 b0]**, **P.c=[1 c0]**; în cazul parametrilor variabili: **P.a=[1 a]**, **P.b=[0 b]**, **P.c=[1 c]** (unde **a**, **b** și **c** sunt vectorii de variație).

♣ Se vor genera 2 seturi de date: unul provenit de la procesul cu parametri constanți și altul – de la procesul cu parametri variabili.





# ⑥ Algoritmi rapizi CMMP-R și VI-R fără fereastră

## 👉 Probleme de simulare

### Problema 7.1 (Identificare recursivă comparativă cu algoritmul de bază)

Mini-simulatorul **ISLAB\_7A** 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  $\mathcal{D}_c$  (generate de procesul cu parametri constanți). Primele două metode operează cu modelul ARX[1,1] (extras din modelul ARMAX prin anularea componentei MA), în timp ce ultimele două – 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.

- Să se comenteze rezultatele obținute cu ajutorul mini-simulatorului **ISLAB\_7A**. Care ar fi explicațiile performanțelor mai slabe ale MCMMP-R în estimarea parametrului părții AR?
- Să se proiecteze mini-simulatorul **ISLAB\_7B**, similar ca structură cu **ISLAB\_7A**, dar care operează cu datele  $\mathcal{D}_v$  (generate de procesul cu parametri variabili). Comentați rezultatele obținute.

Program  
existent

**ISLAB\_7A**

Program ce  
trebuie proiectat

**ISLAB\_7B**



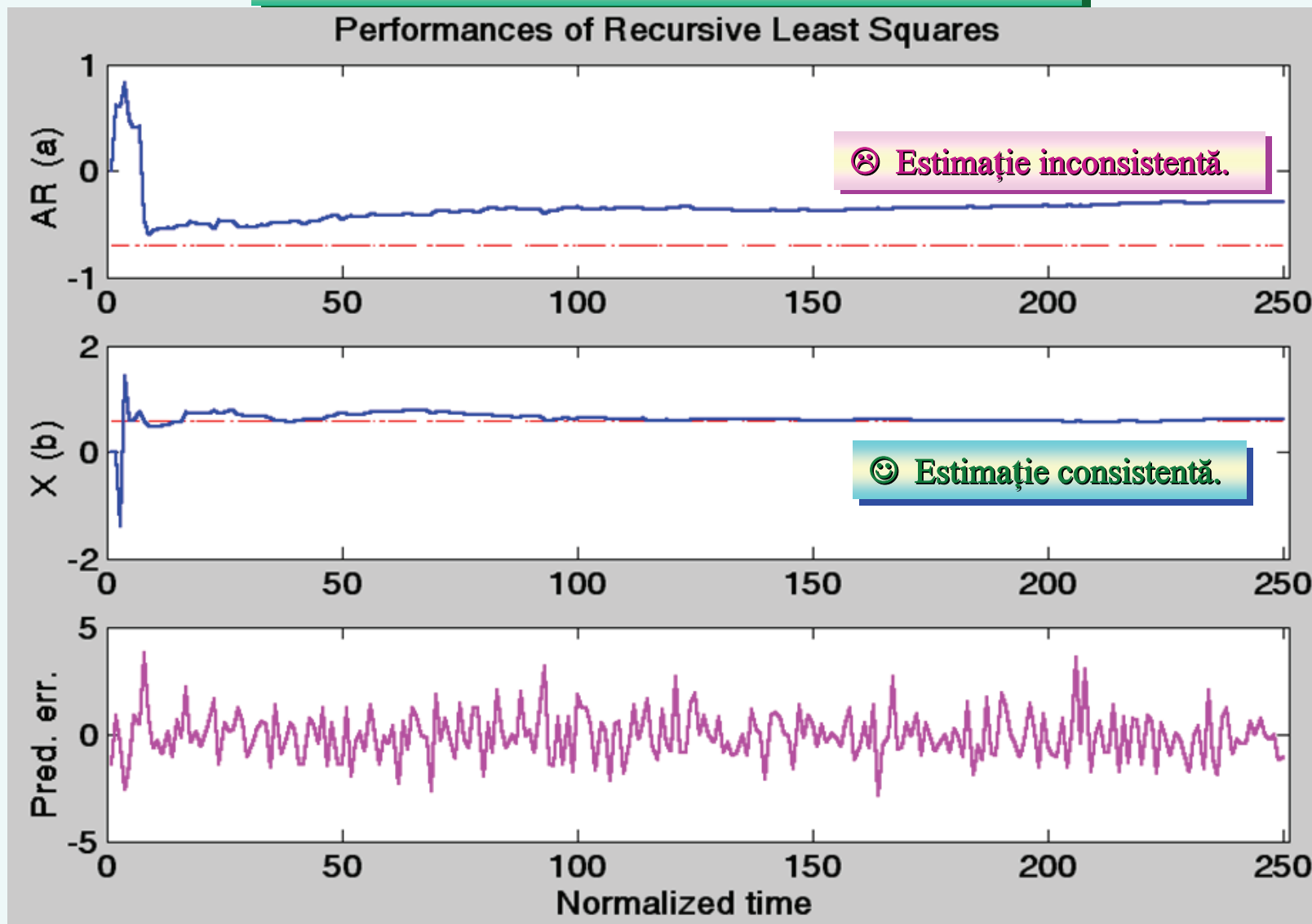
**L.98**





# ⑥ Algoritmi rapizi CMMP-R și VI-R fără fereastră

Ce afișează mini-simulatorul ISLAB\_7A [●○○○]



Performanțele MCMMP-R în cazul modelului ARX



L.99



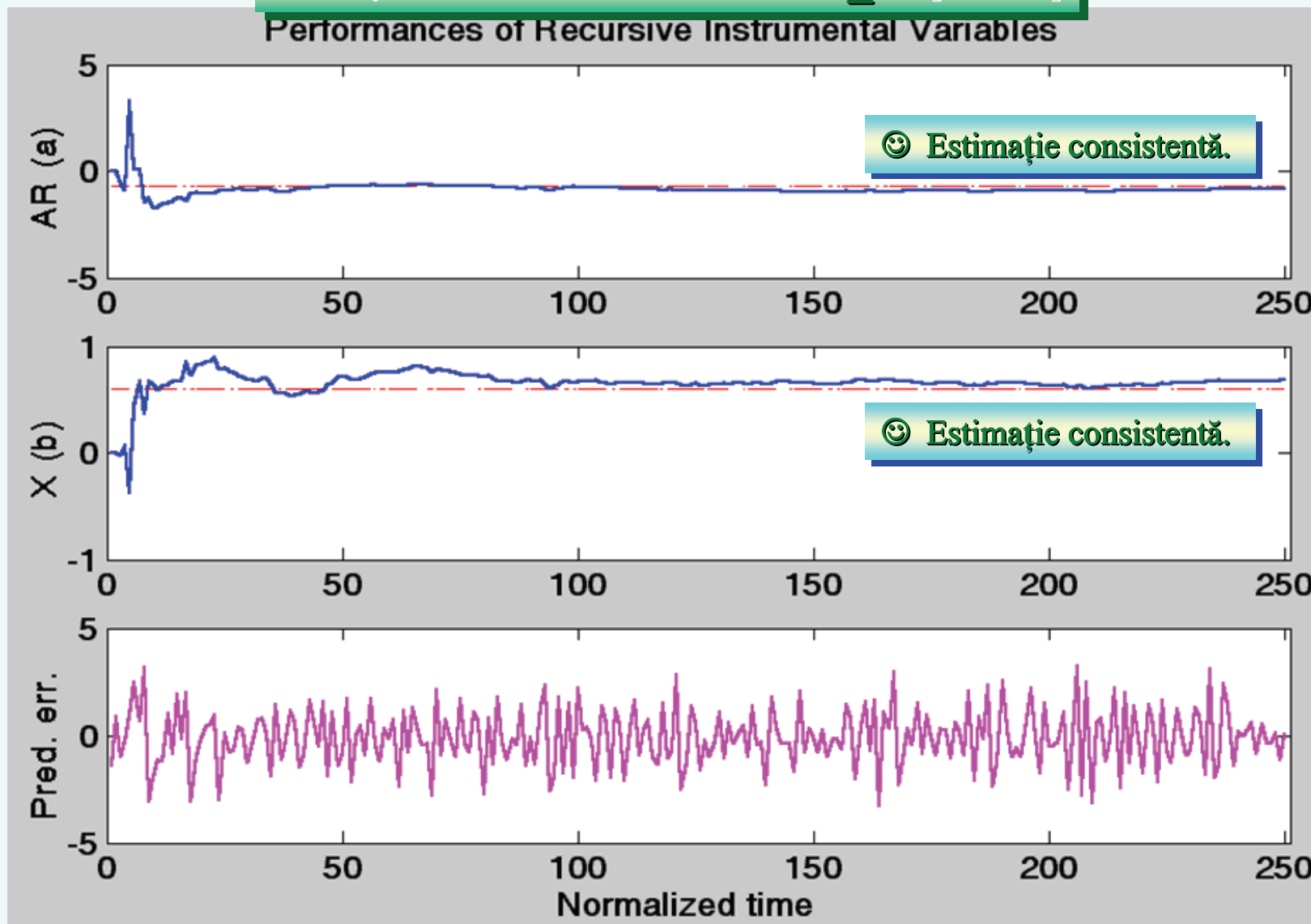




# ⑥ Algoritmi rapizi CMMP-R și VI-R fără fereastră

Ce afișează mini-simulatorul ISLAB\_7A [●●○○]

Performances of Recursive Instrumental Variables

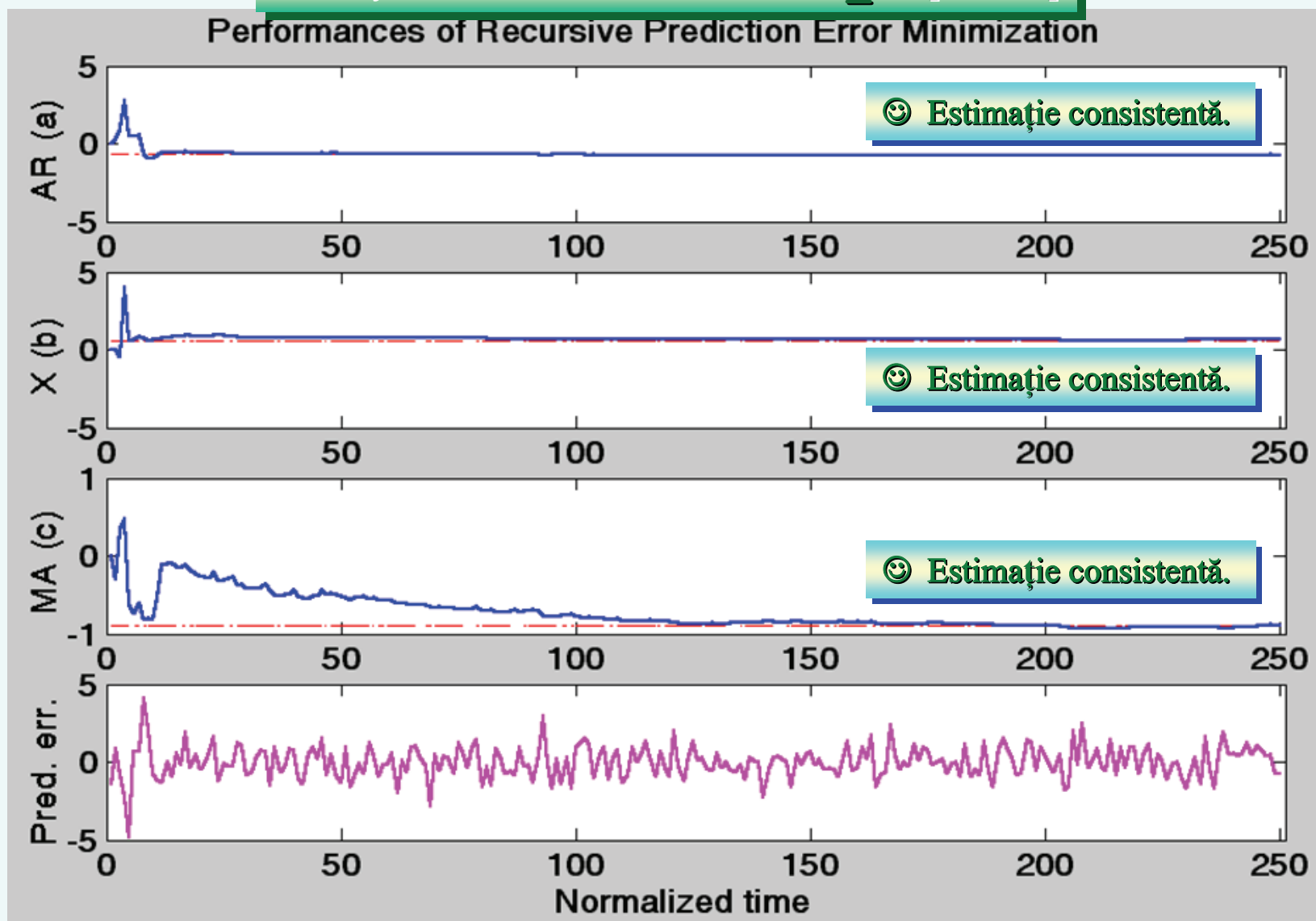


Performanțele MVI-R în cazul modelului ARX



# ⑥ Algoritmi rapizi CMMP-R și VI-R fără fereastră

Ce afișează mini-simulatorul ISLAB\_7A [●●●○]



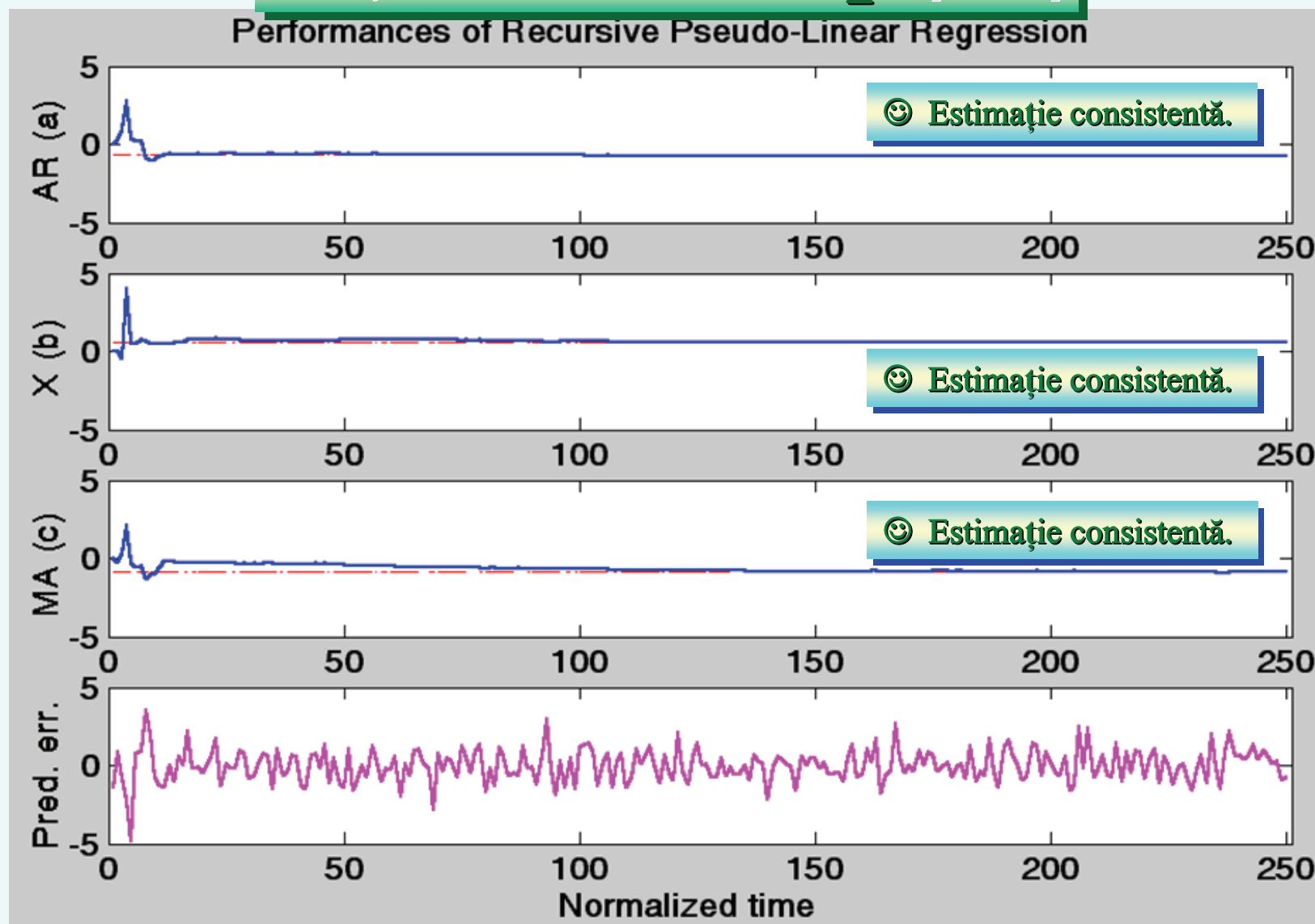
☞ Eficiență superioară.

Performanțele MMEP-R în cazul modelului ARMAX



# ⑥ Algoritmi rapizi CMMP-R și VI-R fără fereastră

Ce afișează mini-simulatorul ISLAB\_7A [●●●●]



☞ Eficiență maximă,  
dar și complexitate ridicată.

Performanțele MPRL-R în cazul modelului ARMAX



# ⑥ Algoritmi rapizi CMMP-R și VI-R fără fereastră

## ➔ Probleme de simulare

### Problema 7.2 (Identificare recursivă cu diferite inițializări)

Să se proiecteze mini-simulatorul **ISLAB\_7C**, care să afișeze performanțele MCMMP-R și MVI-R în cazul modelelor ARX cu parametri constanți și variabili pentru inițializările  $P_0 = \alpha I$ , cu  $\alpha \in \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ . Descrieți influența factorului  $\alpha$  și efectuați o analiză comparativă.

Program ce trebuie proiectat

**ISLAB\_7C**

### Problema 7.3 (Identificare recursivă comparativă)

Se consideră doar procesul ARMAX cu parametri variabili, pentru valori mari ale momentelor de timp. Din acest motiv, procesul va fi aproximat cu unul de tip ARX, adică se va neglija variația coeficientului  $c_1$ . Pentru a genera datele corespunzătoare, se va utiliza tot funcția **gdata\_vp**, dar cu o durată a simulării stabilită la 1000 de eșantioane. Pentru experimentul care urmează, se vor selecta însă doar ultimele 250 de date I/O.

- Să se proiecteze rutinele **arx\_nabla** și **arx\_KB**, care implementează algoritmi recursivi CMMP/VI de tip gradient, respectiv cu filtrare de tip Kalman-Bucy. Rutinele vor avea argumentele principale de intrare și ieșire similare rutinelor **rarx**, **riv**.
- Să se proiecteze mini-simulatorul **ISLAB\_7D**, similar ca structură cu **ISLAB\_7B**, dar care oferă posibilitatea utilizatorului de a selecta oricare dintre cei 6 algoritmi recursivi (cei 4 din cadrul **ISLAB\_7B**, plus cei doi implementați prin rutinele de la punctul anterior). Efectuați diferite simulări cu ajutorul programului **ISLAB\_7D**, care să pună în evidență deosebirile dintre cei 6 algoritmi. Comentați rezultatele obținute.

Program & rutine ce trebuie proiectate

**ISLAB\_7D arx\_nabla arx\_KB**