

Identificarea sistemelor

Ingineria sistemelor, anul 3
Universitatea Tehnică din Cluj-Napoca

Lucian Buşoniu



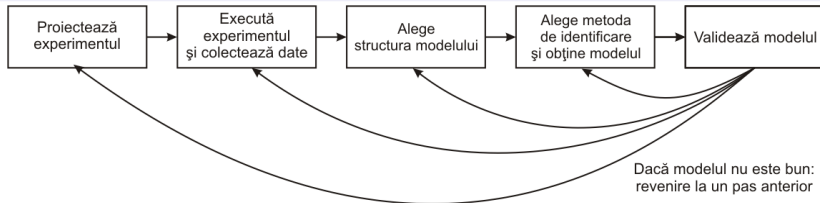
Partea X

Validarea modelelor și probleme practice

Conținut

- 1 Validarea modelelor cu teste de corelație
 - Introducere
 - Teste de corelație
 - Exemplu Matlab
- 2 Selecția structurii și evitarea supraparametrizării
- 3 Alte probleme practice

Reamintim: Importanța validării



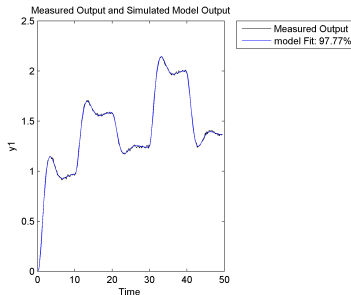
Validarea modelului este un pas esențial: modelul trebuie să fie suficient de bun (pentru scopurile stabilite).

Dacă validarea eșuează, unii dintre pașii precedenți din fluxul de lucru trebuie refăcuți, de exemplu:

- Rerulăm algoritmul de identificare cu parametri diferiți (de ex. δ în metodele recursive).
- Schimbăm structura modelului: de ex. ordinele polinoamelor na, nb în ARX, sau chiar tipul de model, de ex. IV în loc de ARX
- Proiectăm și executăm un nou experiment (de ex. mai multe eșantioane, alt semnal de intrare)

Motivare

Până acum, am validat și selectat modelele informal, examinând graficele ieșirii sau comparând erori – folosind *bunul simț*.



În cele ce urmează, vom introduce o serie de teste fundamentate matematic.

Bunul simț rămâne însă indispensabil – testele matematice funcționează date fiind ipoteze ce pot fi invalidate în practică.

Focus: Metodele de minimizare a erorii de predicție

Ne vom concentra pe modele cu o singură intrare și ieșire, obținute prin *minimizarea erorii de predicție*.

Anumite teste se pot extinde și la alte cazuri.

Conținut

- 1 Validarea modelelor cu teste de corelație
 - Introducere
 - Teste de corelație
 - Exemplu Matlab
- 2 Selecția structurii și evitarea supraparametrizării
- 3 Alte probleme practice

Zgomot alb: Intuiție

Reamintim structura generală de modelul folosită în MEP:

$$y(k) = G(q^{-1})u(k) + H(q^{-1})e(k)$$

unde *se presupune* că $e(k)$ este zgomot alb.

MEP sunt dezvoltate în așa fel încât eroarea de predicție

$\varepsilon(k) = y(k) - \hat{y}(k)$ să fie egală cu $e(k)$. Dacă sistemul satisface structura de model aleasă (pentru a satisface ipoteza de mai sus), și dacă în plus modelul este corect, atunci $\varepsilon(k)$ este și ea zgomot alb.

Ipoteza de zgomot alb

(A) Erorile de predicție $\varepsilon(k)$ sunt zgomot alb de medie zero.

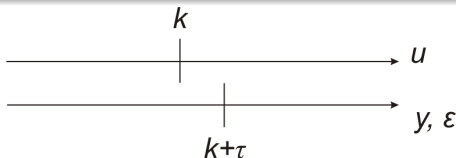
Independența de intrări anterioare: Intuiție

$$y(k) = G(q^{-1})u(k) + v(k)$$

Dacă modelul G este corect, el explică în întregime influența intrării $u(k)$ asupra ieșirilor curente și viitoare $y(k + \tau)$. În consecință, erorile $\varepsilon(k + \tau) = y(k + \tau) - \hat{y}(k + \tau)$ sunt influențate doar de perturbația v , fiind *independente* de intrarea $u(k)$.

Ipoteza de independență 1

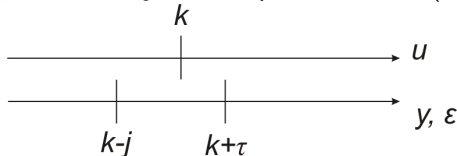
(I1) Erorile de predicție $\varepsilon(k + \tau)$ sunt independente de intrarea $u(k)$ pentru $\tau \geq 0$ (erorile curente și viitoare sunt independente de intrarea curentă).



Independența de toate intrările: Intuiție

$$y(k) = G(q^{-1})u(k) + v(k)$$

Dacă experimentul este în buclă închisă, $u(k)$ depinde de ieșirile precedente și acest lucru va duce la o corelație a erorilor precedente $\varepsilon(k + \tau)$, $\tau < 0$ cu $u(k)$ (de notat că independența nu este afectată pentru $\tau \geq 0$). Dacă experimentul este în buclă deschisă, atunci $\varepsilon(k + \tau)$, $\tau < 0$ sunt și ele independente de $u(k)$.



Ipoteza de independență 2

(I2) Erorile de predicție $\varepsilon(k + \tau)$ sunt independente de $u(k)$ pentru *oricare* τ (toate erorile sunt independente de toate intrările).

Toate ipotezele

- (A) Erorile de predicție $\varepsilon(k)$ sunt zgomot alb de medie zero.
- (I1) Erorile de predicție $\varepsilon(k + \tau)$ sunt independente de intrarea $u(k)$ pentru $\tau \geq 0$ (erorile curente și viitoare sunt independente de intrarea curentă).
- (I2) Erorile de predicție $\varepsilon(k + \tau)$ sunt independente de $u(k)$ pentru *oricare* τ (toate erorile sunt independente de toate intrările).

Un model bun ar trebui să satisfacă (A) și (I1), și dacă bucla este deschisă, și (I2).

Vom dezvolta teste implementabile care fie acceptă, fie resping aceste ipoteze pentru un model dat, și ca atare acceptă sau resping modelul.

Zgomot alb: Corelații

Reamintim funcția de corelație (egală cu covarianța când mediile sunt zero):

$$r_{\varepsilon}(\tau) = \text{E} \{ \varepsilon(k + \tau) \varepsilon(k) \}$$

Dacă $\varepsilon(k)$ este zgomot alb de medie zero:

- Funcția de corelație este zero, $r_{\varepsilon}(\tau) = 0$, pentru orice τ nenul.
- Pentru τ nul, $r_{\varepsilon}(0)$ este varianța σ^2 a zgomotului alb.

Zgomot alb: Corelații din date

Corelațiile se estimează din date, și se normalizează cu varianța (estimată):

$$\hat{r}_\varepsilon(\tau) = \frac{1}{N} \sum_{k=1}^{N-\tau} \varepsilon(k+\tau)\varepsilon(k)$$
$$x(\tau) = \frac{\hat{r}_\varepsilon(\tau)}{\hat{r}_\varepsilon(0)}$$

Test de zgomot alb

În practică, funcția $x(\tau)$ nu va fi niciodată zero pentru seturi finite de date, vom verifica așadar dacă este mică pentru τ nenul. Din motive statistice, impunem un prag la $\frac{1.96}{\sqrt{N}}$

Test de zgomot alb

Dacă $|x(\tau)| \leq \frac{1.96}{\sqrt{N}}$ pentru toate valorile $\tau \neq 0$ suportate de date, atunci ipoteza de zgomot alb (A) este acceptată. Altfel, (A) este respinsă.

Independență: Corelații

Pentru a verifica independența erorilor ε de u , vom folosi funcția de corelație între intrări și erori:

$$r_{\varepsilon u}(\tau) = E \{ \varepsilon(k + \tau) u(k) \}$$

- ❶ Dacă (I1) este adevărată, atunci $r_{\varepsilon u}(\tau) = 0$ pentru $\tau \geq 0$.
- ❷ Dacă (I2) este adevărată, atunci $r_{\varepsilon u}(\tau) = 0$ pentru orice τ .

Estimare din date și normalizare:

$$\hat{r}_{\varepsilon u}(\tau) = \begin{cases} \frac{1}{N} \sum_{k=1}^{N-\tau} \varepsilon(k + \tau) u(k) & \text{if } \tau \geq 0 \\ \frac{1}{N} \sum_{k=1-\tau}^N \varepsilon(k + \tau) u(k) & \text{if } \tau < 0 \end{cases}$$

$$x(\tau) = \frac{\hat{r}_{\varepsilon u}(\tau)}{\sqrt{\hat{r}_{\varepsilon}(\tau) \hat{r}_u(\tau)}}$$

Teste de independență

Teste de independență

Dacă $|x(\tau)| \leq \frac{1.96}{\sqrt{N}}$, $\forall \tau \geq 0$ suportate de date, atunci ipoteza de independență (I1) este acceptată.

Dacă aceeași condiție este adevărată pentru $\forall \tau$ suportate de date (inclusiv τ negativ), atunci (I2) este și ea acceptată.

Dacă modelul este corect (I1 validă), atunci verificarea condiției pentru $\tau < 0$ (I2) testează prezența feedback-ului (buclei de reacție închise).

Teste de corelație: Interpretare generală

$$y(k) = G(q^{-1})u(k) + H(q^{-1})e(k)$$

- Dacă A și I1 sunt valide, atunci întregul model (G și H) este corect
- Dacă I1 este validă dar A respinsă, atunci G este corect dar H este incorect
- Dacă I1 este validă și I2 respinsă, există feedback în setul de date. Dacă I2 este și ea validă atunci nu există feedback

Conținut

- 1 Validarea modelelor cu teste de corelație
 - Introducere
 - Teste de corelație
 - Exemplu Matlab
- 2 Selecția structurii și evitarea supraparametrizării
- 3 Alte probleme practice

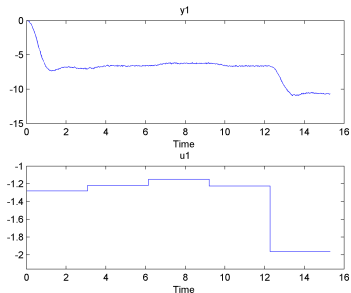
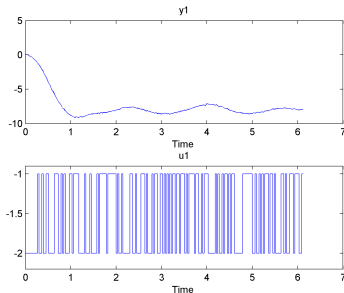
Exemplu Matlab: Date experimentale

Sistemul real este în forma eroare de ieșire, OE:

$$y(k) = \frac{B(q^{-1})}{F(q^{-1})}u(k) + e(k)$$

și are ordinul $n = 3$.

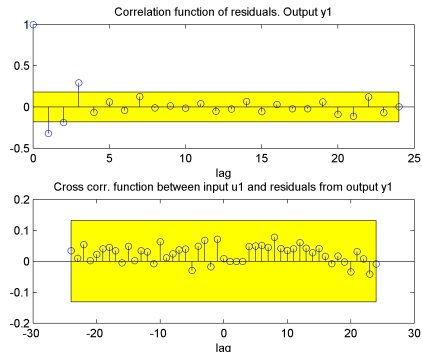
```
plot(id); și plot(val);
```



Matlab: Model ARX

Încercăm întâi un model ARX:

```
mARX = arx(id, [3, 3, 1]); resid(mARX, id);
```

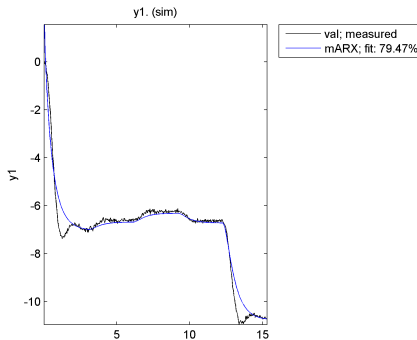


Testul de zgomot alb (A) eșuează, și modelul este respins. Motivul este că sistemul nu este în clasa de modele considerată.

Cum I1 este acceptată, modelul de intrare-ieșire G este bun, dar modelul perturbației H este incorect și trebuie îmbunătățit.

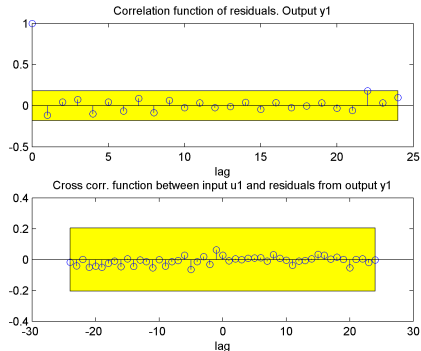
Matlab: Model ARX (continuare)

Simularea pe datele de validare confirmă faptul că modelul este incorect.



Matlab: Model OE

```
mOE = oe(id, [3, 3, 1]); resid(mOE, id);
```

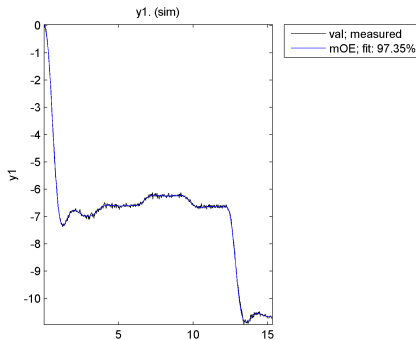


Modelul OE trece toate testele – după cum era de așteptat, sistemul fiind în clasa de modele OE. Așadar, atât G cât și H sunt corecte.

Observație importantă: Funcția Matlab impune un prag mai mic pentru corelații, deci are o probabilitate mai mică de a respinge un model corect.

Matlab: Model OE (continuare)

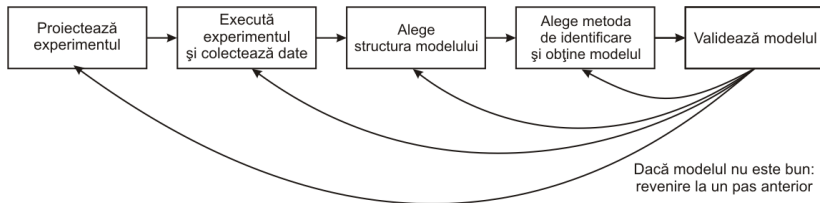
Simularea modelului pe datele de validare confirmă corectitudinea sa.



Conținut

- 1 Validarea modelelor cu teste de corelație
- 2 **Selecția structurii și evitarea supraparametrizării**
 - Selecția structurii
 - Evitarea supraparametrizării
- 3 Alte probleme practice

Selecția structurii în fluxul de lucru



Chiar dacă în majoritatea cazurilor am selectat atent structura modelului (de ex. tip, ordine, lungime), criteriile folosite au fost de obicei informale.

În cele ce urmează, discutăm selecția structurii într-un mod mai precis.

Selecția structurii: Complexitatea modelului

Considerăm că se dau mai multe structuri de modele

$\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_\ell$.

Exemplu: Structuri ARX de mai ordin variabil.

Cum alegem între ele?

O primă idee: alegem $\mathcal{M}_i$ cu eroarea medie pătratică minimă:

$$V(\hat{\theta}) = \frac{1}{N} \sum_{k=1}^N \varepsilon(k)^2$$

Nu ține cont de complexitatea modelului, ignorând așadar:

- efortul de calcul necesar identificare și simulare
- cantitatea datelor necesară pentru identificare
- riscul de supraantrenare

Explorăm alte opțiuni care țin cont de complexitatea modelului (fără a intra în derivarea lor matematică).

Criteriul informației al lui Akaike (AIC)

$$W_{\text{AIC}} = N \log V(\hat{\theta}) + 2p, \text{ or equivalently: } \log V(\hat{\theta}) + \frac{2p}{N}$$

unde N este numărul de puncte și p numărul de parametri (de ex. $na + nb$ în ARX).

Selecție: Modelul cu valoarea W_{AIC} minimă.

Intuiție:

- Termenul $2p$ penalizează complexitatea modelului (numărul de parametri).
- Împărțirea la numărul de date N din $2p/N$ ține cont de faptul că un număr mai mare de date permite identificarea mai multor parametri.
- Aplicarea logaritmului asupra MSE permite o diferențiere mai bună între valori mici ale MSE.

Criteriul erorii finale de predicție (FPE)

$$W_{\text{FPE}} = V(\hat{\theta}) \frac{1 + p/N}{1 - p/N}$$

Selecție: Modelul cu valoarea W_{FPE} minimă.

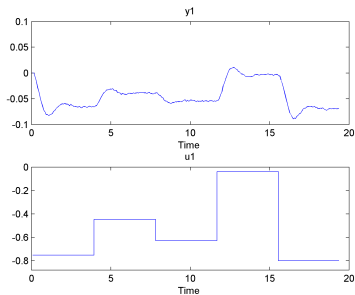
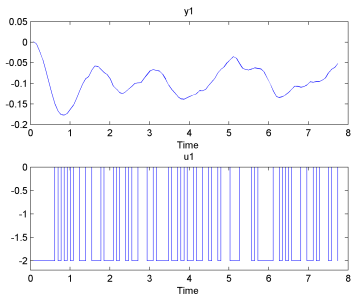
Intuiție: Când N este mare:

$$V(\hat{\theta}) \frac{1 + p/N}{1 - p/N} = V(\hat{\theta}) \left(1 + \frac{2p/N}{1 - p/N}\right) \approx V(\hat{\theta}) \left(1 + \frac{2p}{N}\right)$$

Termenul $\frac{2p}{N}$ funcționează ca și înainte, dar acum aplică o corecție proporțională cu eroarea, mai degrabă decât să i se adune direct

Exemplu Matlab

Un sistem OE cu $n = 2$.



Matlab: selstruc cu AIC

Reamintim arxstruc:

```
Na = 1:15; Nb = 1:15; Nk = 1:5;  
NN = struc(Na, Nb, Nk); V = arxstruc(id, val, NN);
```

- `struc` generează toate combinațiile de ordine în `Na`, `Nb`, `Nk`.
- `arxstruc` identifică pentru fiecare combinație un model ARX pe datele `id`, îl simulează pe datele `val`, și returnează informații despre MSE, ordine etc. în `V`.

Matlab: selstruc cu AIC (continuare)

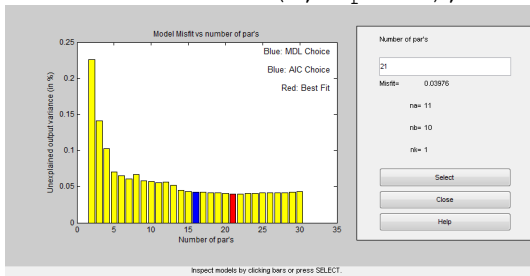
Pentru a alege structura cu cea mai mică valoare a criteriului Akaike:

```
N = selstruc(V, 'aic');
```

Pentru datele noastre, $N = [8, 8, 1]$.

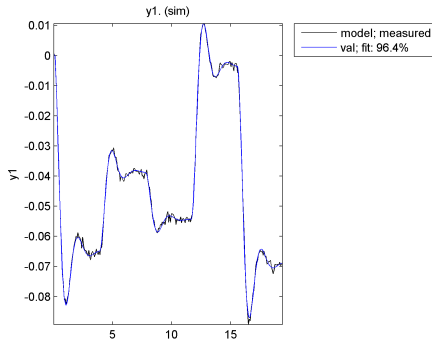
Alternativ, selecția grafică permite și ea folosirea AIC:

```
N = selstruc(V, 'plot');
```



De notat că modelul cu cel mai bun AIC nu este (întotdeauna) același cu modelul cu MSE minimal!

Matlab: Rezultate



Observații

AIC, FPE funcționează și dacă sistemul nu este în clasa de modele considerată.

Matlab oferă funcțiile `aic`, `fpe` care calculează aceste criterii pentru o listă de modele cu orice structură.

Conținut

- 1 Validarea modelelor cu teste de corelație
- 2 **Selecția structurii și evitarea supraparametrizării**
 - Selecția structurii
 - Evitarea supraparametrizării
- 3 Alte probleme practice

Motivare

Considerăm un caz în care *sistemul real* se supune structurii ARMAX:

$$A_0(q^{-1})y(k) = B_0(q^{-1})u(k) + C_0(q^{-1})e(k)$$

unde indicele 0 evidențiază variabilele legate de sistemul real.

Această dinamică este echivalentă cu orice model:

$$W(q^{-1})A_0(q^{-1})y(k) = W(q^{-1})B_0(q^{-1})u(k) + W(q^{-1})C_0(q^{-1})e(k)$$

unde $W(q^{-1})$ este un polinom de gradul nw .

Așadar, metoda ARMAX cu $na = na_0 + nw$, $nb = nb_0 + nw$, $nc = nc_0 + nw$ poate să producă un model precis. Acest model este însă **prea complicat** (supraparametrizat), și va avea factori aproape comuni $W(q^{-1})$ în toate polinoamele (doar “aproape” comuni datorită naturii aproximative a identificării).

Poli și zerouri comune

Acest tip de situație se poate identifica verificând dacă există poli și zerouri care se simplifică (aproximativ).

Exemplificăm folosind funcția Matlab `pzmap`, care arată polii și zerourile funcției de transfer G din modelul general:

$$y(k) = G(q^{-1})u(k) + v(k)$$

În exemplul ARMAX, $G(q^{-1}) = \frac{W(q^{-1})B_0(q^{-1})}{W(q^{-1})A_0(q^{-1})}$, deci rădăcinile lui W sunt atât poli cât și zerouri, simplificându-se aproximativ.

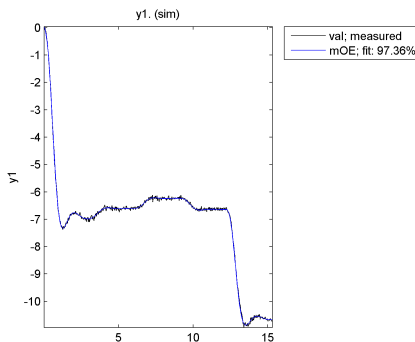
Această idee se aplică și altor tipuri de modele în afară de ARMAX.

Matlab: Model OE supraparametrizat

Pe datele folosite anterior pentru testele de corelație (în care sistemul real are ordinul $n = 3$):

```
mOE = oe(id, [5, 5, 1]);
```

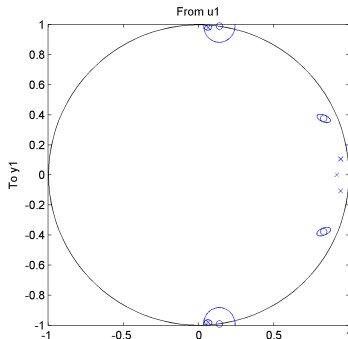
Examinând datele de validare, modelul este precis:



Matlab: Test pentru simplificări între poli și zerouri

```
pzmap(mOE, 'sd', nsd);
```

Argumentele 'sd', nsd impun o regiune de încredere statistică în jurul polilor și zerourilor. Aici alegem nsd=1.96.



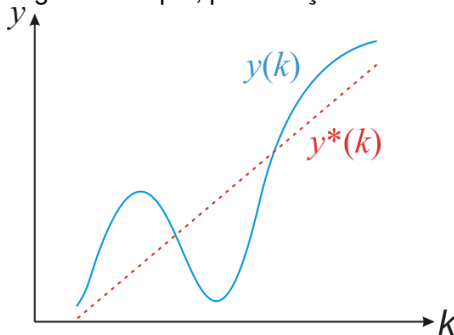
Două perechi de poli și zerouri au regiuni de încredere suprapuse $\Rightarrow$ probabil se simplifică. Acest lucru indică faptul că ordinul ar trebui să fie 3 (ordinul real al sistemului).

Conținut

- 1 Validarea modelelor cu teste de corelație
- 2 Selecția structurii și evitarea supraparametrizării
- 3 Alte probleme practice
 - Devieri
 - Timp mort
 - Minime locale
 - Valori aberante

Devieri

Uneori, datele vor conține semnale parazite lente numite *devieri*, provenind de exemplu din perturbații lente (spre deosebire de perturbațiile sau zgomotul rapid, pe care știm cum să le tratăm)



Idee: Tratăm devierea ca o serie temporală, o identificăm cu regresia liniară, și o îndepărtăm

Estimarea devierii

1. Tratăm intrarea și ieșirea ca două serii temporale separate, scriem modelele devierilor:

$$u^*(k) = \theta_1^u + \theta_2^u k + \theta_3^u k^2 + \dots + \theta_n^u k^{n-1}$$

$$y^*(k) = \theta_1^y + \theta_2^y k + \theta_3^y k^2 + \dots + \theta_n^y k^{n-1}$$

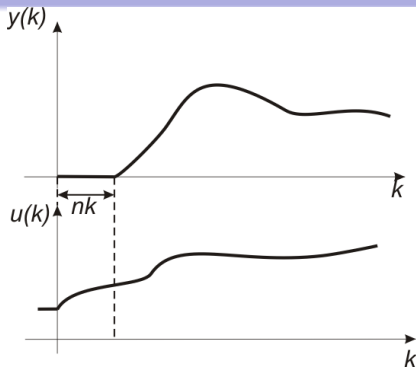
2. Găsim vectorii de parametri θ^u, θ^y prin regresie liniară asupra $u(k), y(k)$ și calculăm devierile corespunzătoare $u^*(k), y^*(k)$
3. Scădem devierile din date:

$$\bar{u}(k) = u(k) - u^*(k), \quad \bar{y}(k) = y(k) - y^*(k)$$

4. Identificăm ca de obicei, dar pe datele “nedeivate” $\bar{u}, \bar{y}$

Observație: Eliminarea devierii de ordinul zero = eliminarea mediei

Timp mort

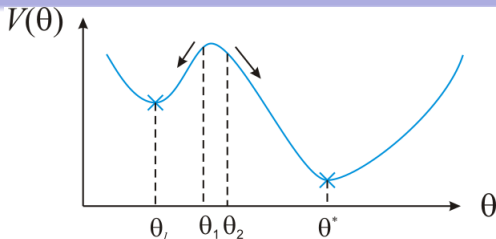


Se citește pe grafic, și se setează nk corespunzător în Matlab. Altfel, se adaugă nk zerouri inițiale în $B(q^{-1})$ din model:

$$\cdots y(k) = \frac{B(q^{-1})}{\cdots} u(k) + \cdots$$

Observație: Subestimarea nk nu este gravă; dar supraestimarea invalidează modelul!

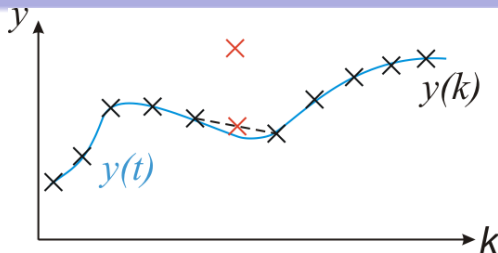
Minime locale



- Optimizarea iterativă (necesară pentru metodele care nu pot fi rezolvate folosind regresia liniară, cum ar fi ARMAX și OE) se poate bloca în minime locale
 - De ex. metoda Newton converge probabil la minimul local θ_l dacă este inițializată în θ_1 . Dar din θ_2 găsește optimul global θ^* !
- ⇒ Dacă rezultatele sunt proaste și se suspectează minime locale, restartăm optimizarea dintr-un alt vector inițial de parametri

Observație: ARMAX converge de obicei la minimul global; OE converge de multe ori la minime locale, cu excepția cazului în care u este zgomot alb

Valori aberante



- Câteodată, anumite măsurători vor avea erori foarte mari, de ex. datorită defectelor tranzitorii. Aceste măsurători se numesc aberante
- Se testează via eroarea de predicție ε , după găsirea unui model inițial: dacă $\varepsilon(k)$ este anormal de mare la un anumit pas k , măsurătoarea respectivă este probabil aberantă
- **Soluția 1:** Înlocuim măsurătoarea folosind de ex. media între $y(k-1)$ și $y(k+1)$ (ca în figură), sau predicția modelului $\hat{y}(k)$
- **Soluția 2:** Limităm erorile de predicție la un maximum rezonabil $\varepsilon_{\max}$, așadar $V(\theta) = \sum_{k=1}^N \min\{\varepsilon^2(k), \varepsilon_{\max}\}$