Identificarea sistemelor

Ingineria sistemelor, anul 3
Universitatea Tehnică din Cluj-Napoca

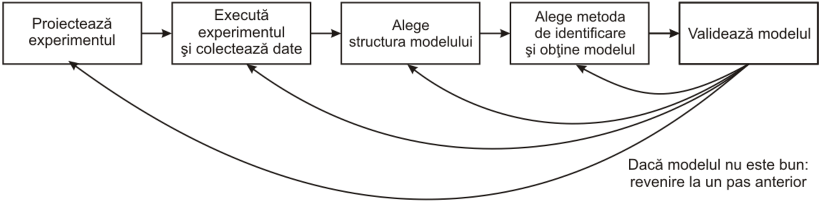
Lucian Buşoniu

Partea X

Validarea modelelor şi probleme practice

Conţinut

Reamintim: Importanţa validării



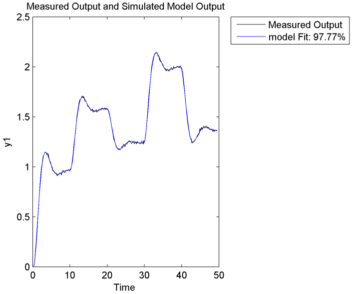
Validarea modelului este un pas esenţial: modelul trebuie să fie suficient de bun (pentru scopurile stabilite).

Dacă validarea eşuează, unii dintre paşii precedenţi din fluxul de lucru trebuie refăcuţi, de exemplu:

- Rerulăm algoritmul de identificare cu parametri diferiţi (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 şi executăm un nou experiment (de ex. mai multe eşantioane, alt semnal de intrare)

Motivare

Până acum, am validat şi selectat modelele informal, examinând graficele ieşirii sau comparând erori – folosind *bunul simţ*.



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

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

Focus: Metodele de minimizare a erorii de predicţie

Ne vom concentra pe modele cu o singură intrare şi ieşire, obţinute prin *minimizarea erorii de predicţie*.

Anumite teste se pot extinde şi la alte cazuri.

Conţinut

Zgomot alb: Intuiţie

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 aşa fel încât eroarea de predicţie $\varepsilon(k) = y(k) - \widehat{y}(k)$ să fie egală cu $e(k)$. Dacă sistemul satisface structura de model aleasă (pentru a satisface ipoteza de zgomot alb), şi dacă în plus modelul este corect, atunci $\varepsilon(k)$ este şi ea zgomot alb.

Ipoteza de zgomot alb

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

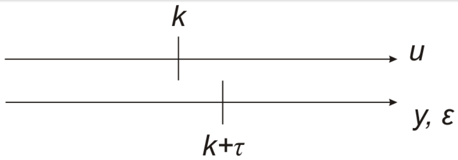
Independenţa de intrări anterioare: Intuiţie

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

Dacă modelul G este corect, el explică în întregime influenţa intrării $u(k)$ asupra ieşirilor curente şi viitoare $y(k+\tau)$. În consecinţă, erorile $\varepsilon(k+\tau) = y(k+\tau) - \widehat{y}(k+\tau)$ sunt influenţate doar de perturbaţia v, fiind *independente* de intrarea $u(k)$. Acest raţionament funcţionează independent de valabilitatea ipotezei de zgomot alb.

Ipoteza de independenţă 1

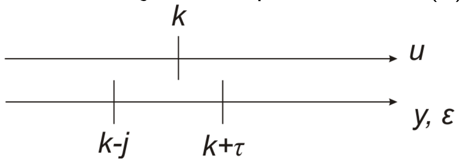
(I1) Erorile de predicţie $\varepsilon(k+\tau)$ sunt independente de intrarea $u(k)$ pentru $\tau \geq 0$ (erorile curente şi viitoare sunt independente de intrarea curentă).



Independenţa de toate intrările: Intuiţie

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

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



Ipoteza de independenţă 2

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

Toate ipotezele

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

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

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

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

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

Zgomot alb: Corelaţii

Reamintim funcţia de corelaţie (egală cu covarianţa când mediile sunt zero):

$$r_\varepsilon(\tau) = \mathrm{E}\left\{\varepsilon(k+\tau)\varepsilon(k)\right\}$$

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

- Funcţia de corelaţie este zero, $r_\varepsilon(\tau) = 0$, pentru orice τ nenul.
- Pentru τ nul, $r_\varepsilon(0)$ este varianţa σ^2 a zgomotului alb.

Zgomot alb: Corelaţii din date

Corelaţiile se estimează din date, şi se normalizează cu varianţa (estimată):

$$\boxed{\begin{aligned} \widehat{r}_\varepsilon(\tau) &= \frac{1}{N}\sum_{k=1}^{N-\tau} \varepsilon(k+\tau)\varepsilon(k) \\ x(\tau) &= \frac{\widehat{r}_\varepsilon(\tau)}{\widehat{r}_\varepsilon(0)} \end{aligned}}$$

Normalizarea ajută deoarece ne putem gândi la valorile normalizate independent de detaliile despre sistem, în timp ce mărimile nenormalizate depind de natura sistemului şi a semnalului (mV, celule per mililitru, m, km, etc. duc la numere diferite).

Test de zgomot alb

În practică, funcţia $x(\tau)$ nu va fi niciodată zero pentru seturi finite de date, vom verifica aşadar 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.

Independenţă: Corelaţii şi calculul lor din date

Pentru a verifica independenţa erorilor ε de u, vom folosi funcţia de corelaţie între intrări şi erori:

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

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

Estimare din date şi normalizare:

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

Teste de independenţă

Teste de independenţă

Dacă $|x(\tau)| \leq \frac{1.96}{\sqrt{N}}$, $\forall \tau \geq 0$ suportate de date, atunci ipoteza de independenţă (I1) este acceptată.
Dacă aceeaşi condiţie este adevărată pentru $\forall \tau$ suportate de date (inclusiv τ negativ), atunci (I2) este şi ea acceptată.

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

Teste de corelaţie: Interpretare generală

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

- Dacă A şi I1 sunt valide, atunci întregul model (G şi H) este corect
- Dacă I1 este validă dar A respinsă, atunci G este corect dar H este incorect
- Dacă I1 este validă şi I2 respinsă, există feedback în setul de date. Dacă I2 este şi ea validă atunci nu există feedback
- Dacă I1 este respinsă, atunci G este incorect şi nu se mai pot trage alte concluzii

Conţinut

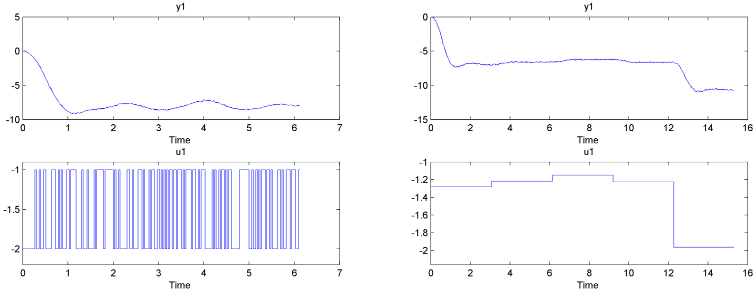
Exemplu Matlab: Date experimentale

Sistemul real este în forma eroare de ieşire, OE:

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

şi are ordinul $n = 3$.

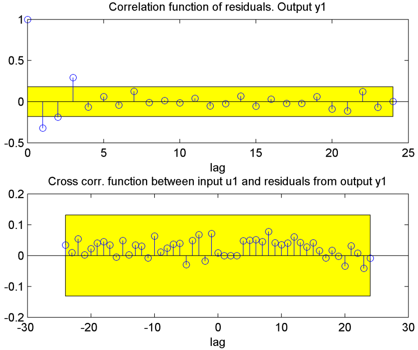
`plot(id);` şi `plot(val);`



Matlab: Model ARX

Încercăm întâi un model ARX:

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

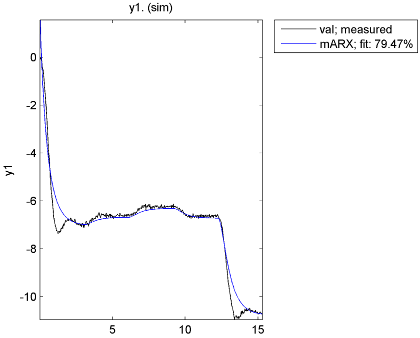


Testul de zgomot alb (A) eşuează, şi modelul este respins. Motivul este că sistemul nu este în clasa de modele considerată.

Cum I1 este acceptată, modelul de intrare-ieşire G este bun, dar modelul perturbaţiei H este incorect şi trebuie îmbunătăţit.

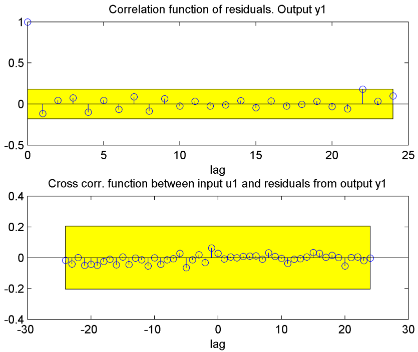
Matlab: Model ARX (continuare)

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



Matlab: Model OE

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

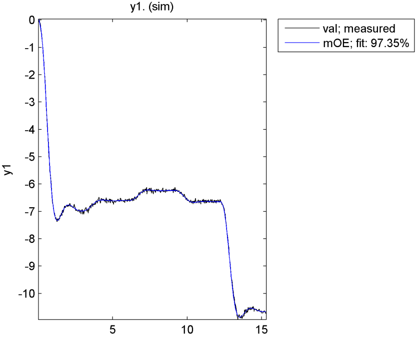


Modelul OE trece toate testele – după cum era de aşteptat, sistemul fiind în clasa de modele OE. Aşadar, atât *G* cât şi *H* sunt corecte.

Observaţie importantă: Funcţia Matlab impune un prag mai mic pentru corelaţii, deci are o probabilitate mai mică de a respinge un model corect.

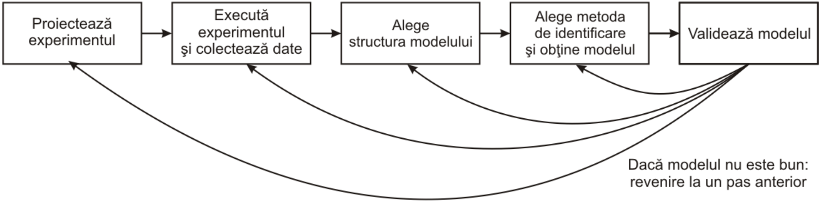
Matlab: Model OE (continuare)

Simularea modelului pe datele de validare confirmă corectitudinea sa.



Conţinut

Selecţia 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 selecţia structurii într-un mod mai precis.

Selecţia 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:

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

Nu ţine cont de complexitatea modelului, ignorând aşadar:

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

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

Criteriul informaţiei al lui Akaike (AIC)

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

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

Selecţie: Modelul cu valoarea W_{AIC} minimă.

Intuiţie:

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

Criteriul erorii finale de predicţie (FPE)

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

Selecţie: Modelul cu valoarea W_{FPE} minimă.

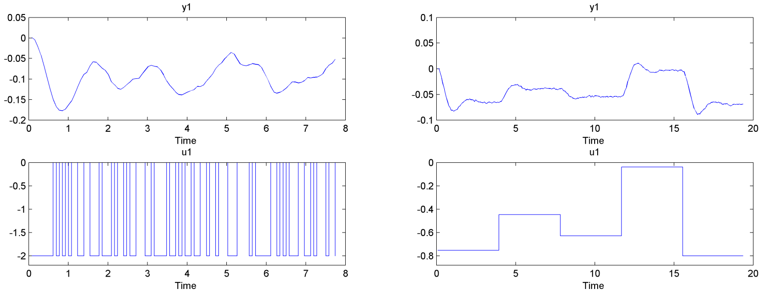
Intuiţie: Când N este mare:

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

şi termenul $\frac{2p}{N}$ funcţionează ca şi înainte, dar acum aplică o corecţie proporţională 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 combinaţiile de ordine în `Na, Nb, Nk`.
- `arxstruc` identifică pentru fiecare combinaţie un model ARX pe datele `id`, îl simulează pe datele `val`, şi returnează informaţii despre MSE, ordine etc. în `V`.

Matlab: `selstruc` cu AIC (continuare)

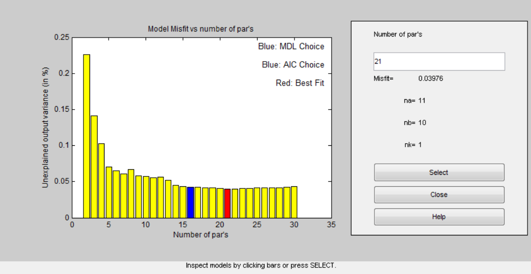
Pentru a alege structura cu cea mai mică valoare a criteriului Akaike:

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

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

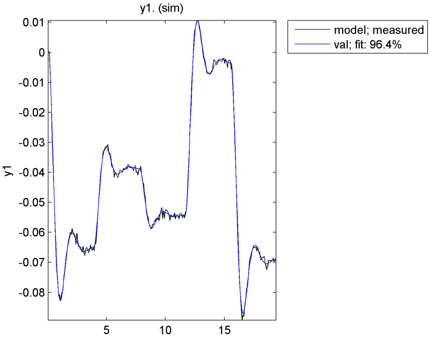
Alternativ, selecţia grafică permite şi ea folosirea AIC:

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



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

Matlab: Rezultate



Observaţii

AIC, FPE funcţionează şi dacă sistemul nu este în clasa de modele considerată.

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

Conţinut

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 evidenţiază 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.

Aşadar, 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), şi va avea factori aproape comuni $W(q^{-1})$ în toate polinoamele (doar "aproape" comuni datorită naturii aproximative a identificării).

Poli şi zerouri comune

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

Exemplificăm folosind funcţia Matlab `pzmap`, care arată polii şi zerourile funcţiei 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 şi zerouri, simplificându-se aproximativ.

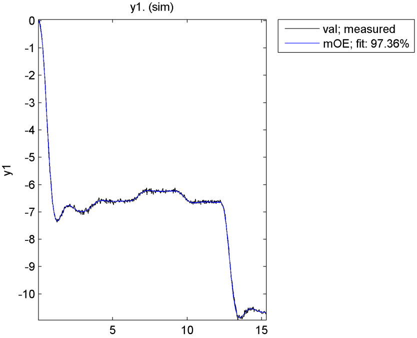
Această idee se aplică şi altor tipuri de modele în afară de ARMAX.

Matlab: Model OE supraparametrizat

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

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

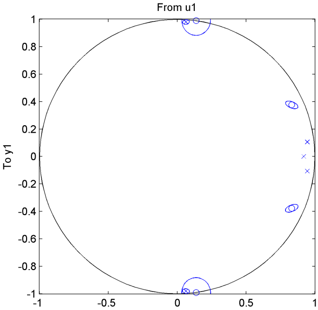
Examinând datele de validare, modelul este precis:



Matlab: Test pentru simplificări între poli şi zerouri

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

Argumentele `'sd', nsd` impun o regiune de încredere statistică în jurul polilor şi zerourilor. Aici alegem `nsd`=1.96, din motive statistice.



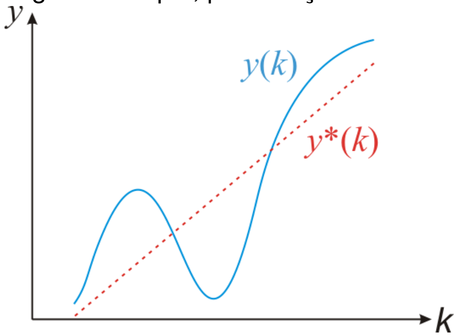
Două perechi de poli şi zerouri au regiuni de încredere suprapuse ⇒ probabil se simplifică. Acest lucru indică faptul că identificarea ar trebui rerulată cu ordinul real al sistemului, 3 (am făcut deja acest lucru în rezultatele anterioare cu OE).

Conţinut

1. Validarea modelelor cu teste de corelaţie
2. Selecţia structurii şi evitarea supraparametrizării
3. Alte probleme practice
   - Devieri
   - Timp mort
   - Minime locale
   - Valori aberante

Devieri

Uneori, datele vor conţine semnale parazite lente numite *devieri*, provenind de exemplu din perturbaţii lente (spre deosebire de perturbaţiile sau zgomotul rapid, pe care ştim cum să le tratăm)



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

Estimarea devierii

1. Tratăm intrarea şi ieşirea ca două serii temporale separate (aşadar, nu mai avem o problemă de identificare a unui sistem dinamic); scriem modelele devierilor:

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

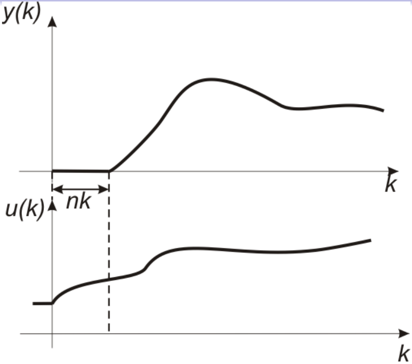
2. Găsim vectorii de parametri θ^u, θ^y prin regresie liniară asupra $u(k), y(k)$ şi calculăm devierile corespunzătoare $u^*(k), y^*(k)$
3. Scădem devierile din date:

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

4. Identificăm ca de obicei, dar pe datele "nedeviate" $\bar{u}, \bar{y}$

Observaţii: Există funcţia Matlab `detrend`; Eliminarea devierii de ordinul zero = eliminarea mediei

Timp mort

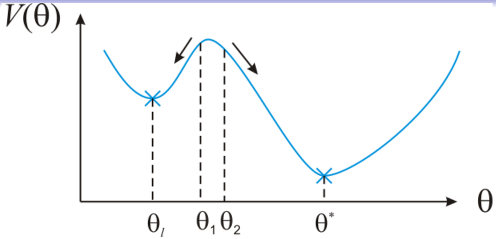


Se citeşte pe grafic, şi se setează *nk* corespunzător în Matlab. Altfel, se adaugă *nk* zerouri iniţiale în $B(q^{-1})$ din model:

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

Observaţie: Subestimarea *nk* nu este gravă (posibil să fie nevoie de mărirea *nb*); dar supraestimarea invalidează modelul!

Minime locale

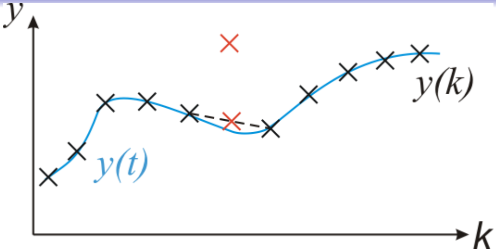


- Optimizarea iterativă (necesară pentru metodele care nu pot fi rezolvate folosind regresia liniară, cum ar fi ARMAX şi OE) se poate bloca în minime locale
- De ex. metoda Newton converge probabil la minimul local θ_ℓ dacă este iniţializată în θ_1. Dar din θ_2 găseşte optimul global θ^*!

$\Rightarrow$ Dacă rezultatele sunt proaste şi se suspectează minime locale, restartăm optimizarea dintr-un alt vector iniţial de parametri

Observaţie: ARMAX converge de obicei la minimul global; OE converge de multe ori la minime locale, cu excepţia 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 predicţie ε, după găsirea unui model iniţial: dacă $\varepsilon(k)$ este anormal de mare la un anumit pas k, măsurătoarea respectivă este probabil aberantă
- Soluţia 1: Înlocuim măsurătoarea folosind de ex. media între $y(k-1)$ şi $y(k+1)$ (ca în figură), sau predicţia modelului $\hat{y}(k)$
- Soluţia 2: Limităm erorile de predicţie la un maximum rezonabil $\varepsilon_{\max}$, aşadar $V(\theta) = \sum_{k=1}^{N} \min\{\varepsilon^2(k), \varepsilon_{\max}\}$