Identificarea sistemelor

Ingineria sistemelor, anul 3
Universitatea Tehnică din Cluj-Napoca

Lucian Buşoniu

Partea VII

Metoda minimizării erorii de predicţie

Conţinut

1. Structuri de model
2. Metoda generală a minimizării erorii de predicţie
3. Rezolvarea problemei de optimizare
4. Garanţii de performanţă

Clasificare

Reamintim taxonomia modelelor din Partea I:

După numărul de parametri:

1. Modele parametrice: au formă fixă (formulă matematică), număr cunoscut şi de obicei mic de parametri
2. Modele neparametrice: nu pot fi descrise cu un număr fix, mic de parametri
   Adesea reprezentate prin grafice sau tabele

După cunoştinţele disponibile în avans ("culoare"):

1. Modele din principii de bază, cutie albă: complet cunoscute în avans
2. Modele cutie neagră: complet necunoscute în avans
3. Modele cutie gri: parţial cunoscute

Ca şi ARX, metoda generală a minimizării erorii de predicţie (MEP) produce modele *cutie neagră*, *parametrice* (polinomiale).

Conţinut

1. Structuri de model
2. Metoda generală a minimizării erorii de predicţie
3. Rezolvarea problemei de optimizare
4. Garanţii de performanţă

Motivaţie

MEP poate fi văzută ca o extensie a metodei ARX la structuri de model mult mai generale, fiind aşadar capabilă să identifice mult mai multe clase de sisteme.

Pentru claritate, vom prezenta întâi structura generală de model căreia i se aplică MEP.

Structura generală de model

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

unde G şi H sunt *funcţii de transfer* în timp discret – fracţii de polinoame. Semnalul $e(t)$ este zgomot alb de medie zero.

- G este funcţia de transfer intrare-ieşire.
- H este funcţia de transfer a perturbaţiei.

Structura generală: Formă explicită

Prin gruparea factorilor comuni ai numitoarelor din G şi H în $A(q^{-1})$, obţinem forma mai detaliată:

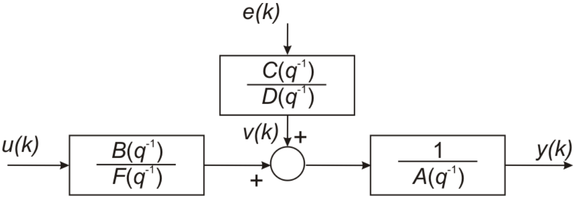
$$y(k) = \frac{B(q^{-1})}{A(q^{-1})F(q^{-1})}u(k) + \frac{C(q^{-1})}{A(q^{-1})D(q^{-1})}e(k)$$
$$A(q^{-1})y(k) = \frac{B(q^{-1})}{F(q^{-1})}u(k) + \frac{C(q^{-1})}{D(q^{-1})}e(k)$$

unde A, B, C, D, F sunt toate polinoame, de ordin na, nb, nc, nd, nf:

$$A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_{na} q^{-na}$$
$$B(q^{-1}) = b_1 q^{-1} + \ldots + b_{nb} q^{-nb} \qquad F(q^{-1}) = 1 + f_1 q^{-1} + \ldots + f_{nf} q^{-nf}$$
$$C(q^{-1}) = 1 + c_1 q^{-1} + \ldots + c_{nc} q^{-nc} \quad D(q^{-1}) = 1 + d_1 q^{-1} + \ldots + d_{nd} q^{-nd}$$

Structura generală: Formă grafică

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



Această formă este foarte generală. Nu se va folosi în practică, ci pentru descrierea algoritmilor într-un mod generic care funcţionează pentru orice formă particulară. În practică, vom folosi una dintre aceste forme particulare, exemplificate în cele ce urmează.

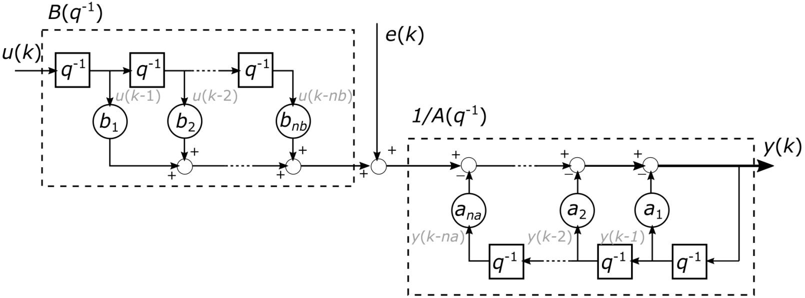
Structura ARMAX

Impunând $F = D = 1$ (adică ordinele $nf = nd = 0$), obţinem:

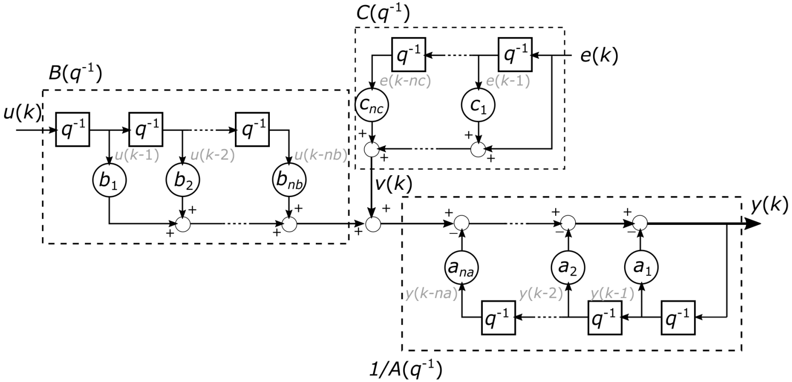
$$\boxed{A(q^{-1})y(k) = B(q^{-1})u(k) + C(q^{-1})e(k)}$$

Nume: AutoRegresiv (dependenţa de ieşirile anterioare), cu Medie Alunecătoare (se referă la modelul perturbaţiei) cu intrare eXogenă (dependenţa de u)

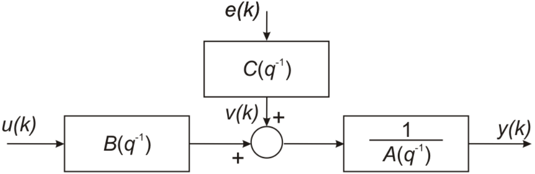
Reamintim forma grafică detaliată pentru ARX



ARMAX: formă grafică



Formă compactă:



ARMAX: formă explicită

$$
\begin{aligned}
A(q^{-1})y(k) &= B(q^{-1})u(k) + C(q^{-1})e(k) \\
A(q^{-1}) &= 1 + a_1 q^{-1} + \ldots + a_{na} q^{-na} \\
B(q^{-1}) &= b_1 q^{-1} + \ldots + b_{nb} q^{-nb} \\
C(q^{-1}) &= 1 + c_1 q^{-1} + \ldots + c_{nc} q^{-nc}
\end{aligned}
$$

$$
\begin{aligned}
y(k) &+ a_1 y(k-1) + \ldots + a_{na} y(k-na) \\
&= b_1 u(k-1) + \ldots + b_{nb} u(k-nb) \\
&+ e(k) + c_1 e(k-1) + \ldots + c_{nc} e(k-nc)
\end{aligned}
$$

cu vectorul de parametri:

$$
\theta = [a_1, \ldots, a_{na}, b_1, \ldots, b_{nb}, c_1, \ldots, c_{nc}]^\top \in \mathbb{R}^{na+nb+nc}
$$

Exemplu recurent: ARMAX de ordinul 1

Alegem $na = nb = nc = 1$, obţinând un model ARMAX de ordinul întâi:

$$\begin{aligned}(1 + aq^{-1})y(k) &= bq^{-1}u(k) + (1 + cq^{-1})e(k)\\ y(k) &= -ay(k-1) + bu(k-1) + ce(k-1) + e(k)\end{aligned}$$

unde am omis indecşii pentru a, b, c fiindcă există un singur parametru în fiecare polinom.

Vom reveni la acest exemplu de-a lungul părţii curente pentru a ilustra toate etapele MEP.

Caz special de ARMAX: ARX

Impunând $C = 1$ în ARMAX ($nc = 0$), obţinem:

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

exact modelul ARX pe care l-am studiat deja.

În comparaţie cu ARX, ARMAX poate modela perturbaţii mai complicate ($C(q^{-1})e(k)$ în loc de $e(k)$, care de obicei se presupune că este zgomot alb de medie zero).

Reamintim: FIR este un caz particular al ARX

Impunând mai departe $A = 1$ ($na = 0$) în ARX, obţinem:

$$y(k) = B(q^{-1})u(k) + e(k) = \sum_{j=1}^{nb} b_j u(k-j) + e(k)$$
$$= \sum_{j=0}^{M-1} h(j)u(k-j) + e(k)$$

modelul FIR.

Reamintim însă că FIR şi ARX sunt modele de natură diferită (răspuns la impuls neparametric, respectiv funcţie de transfer parametrică).

Structura de tip eroare de ieşire

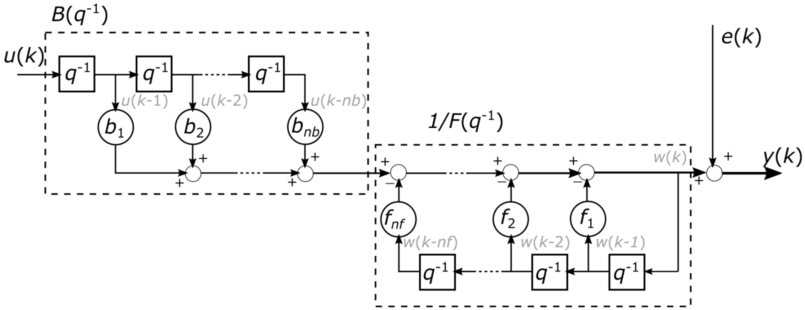
Model de tip eroare de ieşire, OE (en. *Output Error*):

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

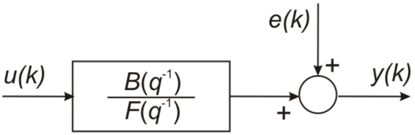
obţinut pentru $na = nc = nd = 0$, adică $A = C = D = 1$.

Structura corespunde unui zgomot simplu, aditiv la ieşire (de unde reiese şi numele).

OE: formă grafică

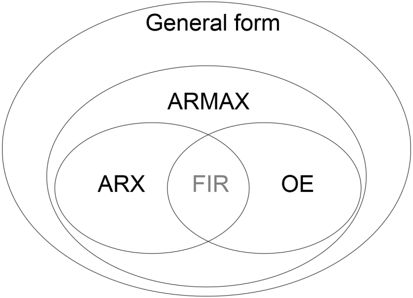


Formă compactă:



Rezumat si relaţie globală între modele

- Forma generală a modelelor utilizate în MEP
- Structuri specifice, inclusiv unele pe care le-am întâlnit deja
- Relaţii:



- ARMAX vs. ARX vs. OE: perturbaţii diferite.
- ARX vs. FIR: modele de natură diferită.

Exerciţii

1. Care este forma explicită (cu formule în semnale întârziate) a modelului OE?
2. Desenaţi forma grafică a modelului FIR şi a modelului din cazul general.
3. Căutaţi în literatură structura Box-Jenkins şi derivaţi-i forma explicită şi cea grafică. Care este legătura acestei structuri cu restul tipurilor discutate?

Conţinut

1. Structuri de model
2. Metoda generală a minimizării erorii de predicţie
   - Pas preliminar: ARX revizitat
   - Cazul general
   - Exemplu: ARMAX de ordinul 1
   - Exemplu Matlab
3. Rezolvarea problemei de optimizare
4. Garanţii de performanţă

ARX reinterpretat ca MEP

1. Dat fiind vectorul de parametri θ, calculăm predicţiile la fiecare pas, $\widehat{y}(k) = \varphi^\top(k)\theta$.
2. Calculăm erorile de predicţie la fiecare pas, $\varepsilon(k) = y(k) - \widehat{y}(k)$.
3. Găsim vectorul de parametri θ care minimizează media erorilor pătratice $V(\theta) = \frac{1}{N}\sum_{k=1}^{N} \varepsilon^2(k)$.

Procedura de mai sus este doar o reinterpretare, echivalentă cu algoritmul deja studiat la cursul de ARX.

MEP se obţine extinzând procedura la structura generală de model.

ARX reinterpretat ca MEP (continuare)

Observaţii:

- Pentru ARX, ştim deja cum să minimizăm MSE (cu regresia liniară); pentru modele mai generale vom prezenta metode noi de minimizare.
- Predictorul ARX $\widehat{y}(k)$ este ales pentru a obţine eroarea $\varepsilon(k) = e(k)$, egală cu zgomotul. Acesta va fi şi scopul MEP generale, intuitiv fiindcă nu putem spera la mai mult.
  De notat semnificaţiile diferite ale $\varepsilon(k)$ (eroare de predicţie) şi $e(k)$ (zgomot)
- Eroarea de predicţie este doar o rearanjare a ecuaţiei $y(k) = \varphi^\top(k)\theta + \varepsilon(k) = \widehat{y}(k) + \varepsilon(k)$.

Checklist

Pentru fiecare structură de model, va trebui să trecem prin aceşti trei paşi: predictor, eroare de predicţie, şi minimizarea MSEului.

Vom menţine o listă de verificare (pe scurt, checklist):

	ARX	MEP generală	ARMAX ordinul 1
predicţie $\widehat{y}(k)$	$\checkmark$		
eroare de predicţie $\varepsilon(k)$	$\checkmark$		
minimizare $V(\theta)$	$\checkmark$		

Conţinut

1. Structuri de model
2. Metoda generală a minimizării erorii de predicţie
   - Pas preliminar: ARX revizitat
   - Cazul general
   - Exemplu: ARMAX de ordinul 1
   - Exemplu Matlab
3. Rezolvarea problemei de optimizare
4. Garanţii de performanţă

Reamintim: Structura generală de model

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

unde G şi H sunt funcţii de transfer în timp discret:

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

În cele ce urmează, vom omite argumentul q^{-1} pentru a mări lizibilitatea ecuaţiilor, şi vom înţelege explicit că literele mari se referă la funcţii de transfer sau polinoame în variabila q^{-1}.

Eroarea de predicţie

Începem prin calculul zgomotului $e(k)$.

$$y(k) = Gu(k) + He(k)$$
$$\Rightarrow e(k) = H^{-1}(y(k) - Gu(k))$$

unde $H^{-1} = \frac{AD}{C}$ este inversa fracţiei de polinoame H.

Predictorul va fi derivat în aşa fel încât eroarea de predicţie să fie egală cu zgomotul, $\varepsilon(k) = y(k) - \widehat{y}(k) = e(k)$. Aşadar, aceeaşi formulă ca şi mai sus se poate folosi pentru a calcula $\varepsilon(k)$:

$$\boxed{\varepsilon(k) = H^{-1}(y(k) - Gu(k))}$$

Acesta este un sistem dinamic care poate fi simulat pentru a obţine semnalul $\varepsilon(k)$.

Predictor

Pentru a obţine eroarea $e(k)$, dinamica predictorului trebuie să fie:

$$\begin{aligned}\widehat{y}(k) &= y(k) - e(k) \\ &= y(k) - H^{-1}(y(k) - Gu(k)) \\ &= \boxed{(1 - H^{-1})y(k) + H^{-1}Gu(k)}\end{aligned}$$

Observaţie: Pentru a avea un predictor *cauzal*, care depinde doar de valorile anterioare ale ieşirii şi intrării, este necesar ca $G(0) = 0$ şi $H(0) = 1$. Aceste condiţii sunt satisfăcute deoarece $B(0) = 0, A(0) = C(0) = D(0) = 1$.

Preview: Găsirea parametrilor şi folosirea modelului

Găsirea parametrilor: Odată ce procedura pentru calculul erorilor este disponibilă, parametrii θ sunt găsiţi prin minimizarea funcţiei obiectiv $V(\theta) = \frac{1}{N}\sum_{k=1}^{N} \varepsilon^2(k)$. Această minimizare necesită de obicei evaluări multiple ale semnalului de eroare, pentru mai multe valori ale lui θ.

Nu discutăm încă metodele prin care rezolvăm problema de minimizare. Le vom studia în detaliu în secţiunea următoare.

Folosirea modelului: În final, odată ce o estimare $\widehat{\theta}$ a optimului este găsită, formula de predicţie poate fi aplicată pentru calculul ieşirii modelului $\hat{y}(k)$. Modelul se poate adapta şi pentru simulare.

Checklist

	ARX	MEP generală	ARMAX ordinul 1
predictor $\widehat{y}(k)$	✓	✓	
eroare de predicţie $\varepsilon(k)$	✓	✓	
minimizare $V(\theta)$	✓	?	

Table of contents

1 Structuri de model

2 Metoda generală a minimizării erorii de predicţie
- Pas preliminar: ARX revizitat
- Cazul general
- Exemplu: ARMAX de ordinul 1
- Exemplu Matlab

3 Rezolvarea problemei de optimizare

4 Garanţii de performanţă

Exemplu recurent: ARMAX ordinul 1

Reamintim ARMAX:

$$Ay(k) = Bu(k) + Ce(k)$$

Prin plasare în forma standard, obţinem:

$$\begin{aligned} y(k) &= \frac{B}{A}u(k) + \frac{C}{A}e(k) \\ &= Gu(k) + He(k) \end{aligned}$$

Aşadar, $G = \frac{B}{A}$, $H = \frac{C}{A}$

Pentru ordinul 1: $A = 1 + aq^{-1}$, $B = bq^{-1}$, $C = 1 + cq^{-1}$.

ARMAX ordinul 1: Eroare de predicţie

Reamintim formula generală a erorii de predicţie:

$$\varepsilon(k) = H^{-1}(y(k) - Gu(k))$$

În cazul nostru, fiindcă $G = \frac{B}{A}$, $H = \frac{C}{A}$:

$$\begin{aligned}
\varepsilon(k) &= \frac{A}{C}\left(y(k) - \frac{B}{A}u(k)\right) \\
C\varepsilon(k) &= Ay(k) - Bu(k) \\
(1 + cq^{-1})\varepsilon(k) &= (1 + aq^{-1})y(k) - bq^{-1}u(k) \\
\varepsilon(k) + c\varepsilon(k-1) &= y(k) + ay(k-1) - bu(k-1) \\
\varepsilon(k) &= -c\varepsilon(k-1) + y(k) + ay(k-1) - bu(k-1)
\end{aligned}$$

Formula este una *dinamică*, recursivă ce trebuie simulată! Necesită iniţializare pentru $\varepsilon(0)$, de obicei luată 0.

ARMAX ordinul 1: Predictor

Reamintim formula generală a predictorului:

$$\widehat{y}(k) = (1 - H^{-1})y(k) + H^{-1}Gu(k)$$

În cazul nostru, fiindcă $G = \frac{B}{A}$, $H = \frac{C}{A}$:

$$\begin{aligned}
\widehat{y}(k) &= \left(1 - \frac{A}{C}\right) y(k) + \frac{\cancel{A}}{C}\frac{B}{\cancel{A}}u(k) \\
C\widehat{y}(k) &= (C - A)y(k) + Bu(k) \\
(1 + cq^{-1})\widehat{y}(k) &= (\cancel{1} + cq^{-1} - \cancel{1} - aq^{-1})y(k) + bq^{-1}u(k) \\
\widehat{y}(k) + c\widehat{y}(k-1) &= (c-a)y(k-1) + bu(k-1) \\
\widehat{y}(k) &= -c\widehat{y}(k-1) + (c-a)y(k-1) + bu(k-1)
\end{aligned}$$

Din nou, o formulă *dinamică*, recursivă. Necesită iniţializare pentru $\widehat{y}(0)$; această valoare iniţială se ia de obicei 0.

Checklist

	ARX	MEP generală	ARMAX ordinul 1
predictor $\widehat{y}(k)$	$\checkmark$	$\checkmark$	$\checkmark$
eroare de predicţie $\varepsilon(k)$	$\checkmark$	$\checkmark$	$\checkmark$
minimizare $V(\theta)$	$\checkmark$	?	?

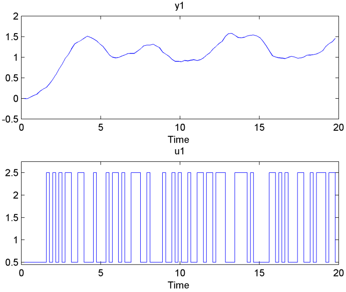
Conţinut

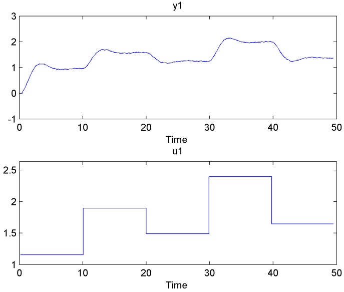
1. Structuri de model
2. Metoda generală a minimizării erorii de predicţie
   - Pas preliminar: ARX revizitat
   - Cazul general
   - Exemplu: ARMAX de ordinul 1
   - Exemplu Matlab
3. Rezolvarea problemei de optimizare
4. Garanţii de performanţă

Date experimentale

Folosim din nou datele experimentale pe care am identificat modelul ARX mai demult.

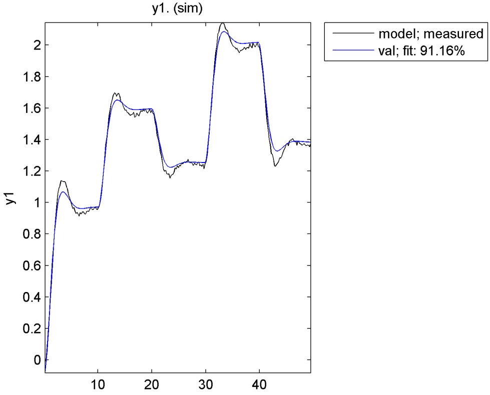
`plot(id);` şi `plot(val);`





Reamintim: rezultat ARX

Presupunând că sistemul este de ordinul 2 fără timp mort, alegem $na = 2, nb = 2, nk = 1$.



Rezultatele sunt destul de proaste.

Identificarea unui model ARMAX

```
mARMAX = armax(id, [na, nb, nc, nk]);
```

Argumente:

1. Setul de date de identificare.
2. Vector conţinând ordinele polinoamelor A, B, C şi *întârzierea* nk.

Ca şi pentru ARX, structura include explicit întârzierea minimă nk între intrări şi ieşiri.

$$
\begin{aligned}
y(k) &+ a_1 y(k-1) + a_2 y(k-2) + \ldots + a_{na} y(k-na) \\
&= b_1 u(k-nk) + b_2 u(k-nk-1) + \ldots + b_{nb} u(k-nk-nb+1) \\
&+ e(k) + c_1 e(k-1) + c_2 e(k-2) + \ldots + c_{nc} e(k-nc)
\end{aligned}
$$

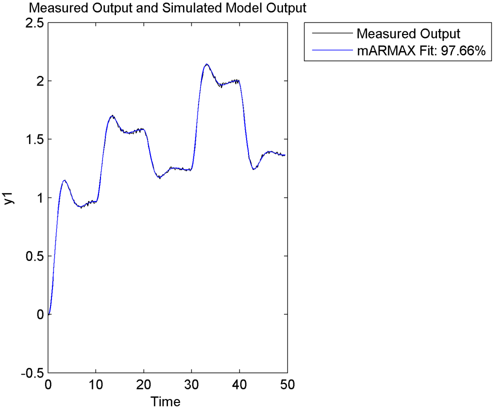
$$A(q^{-1})y(k) = B(q^{-1})u(k-nk) + C(q^{-1})e(k), \text{ where:}$$

$$
\begin{aligned}
A(q^{-1}) &= (1 + a_1 q^{-1} + a_2 q^{-2} + \ldots + a_{na} q^{-na}) \\
B(q^{-1}) &= (b_1 + b_2 q^{-1} + b_{nb} q^{-nb+1}) \\
C(q^{-1}) &= (1 + c_1 q^{-1} + c_2 q^{-2} + \ldots + c_{nc} q^{-nc})
\end{aligned}
$$

Observaţie: Ca pentru ARX, structura teoretică se obţine alegând $nk = 1$ (şi pentru a reprezenta $nk > 1$ în structura teoretică, adăugăm coeficienţi iniţiali zero în B).

Model ARMAX

Considerând că sistemul este de ordinul 2 fără timp mort, alegem $na = nb = nc = 2$, $nk = 1$. Validare: `compare(val, mARMAX);`

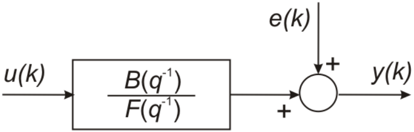


Spre deosebire de ARX, rezultatele sunt bune. Modelul mai flexibil al perturbaţiei îşi arată utilitatea.

Identificarea unui model OE

Reamintim structura OE:

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



Identificarea unui model OE (continuare)

```
mOE = oe(id, [nb, nf, nk]);
```

Argumente:

1. Setul de date de identificare.
2. Vector conţinând ordinele polinoamelor B, F, şi întârzierea nk.

$$y(k) = \frac{B(q^{-1})}{F(q^{-1})}u(k-nk) + e(k), \text{ where:}$$
$$B(q^{-1}) = (b_1 + b_2 q^{-1} + b_{nb} q^{-nb+1})$$
$$F(q^{-1}) = (1 + f_1 q^{-1} + f_2 q^{-2} + \ldots + f_{nf} q^{-nf})$$

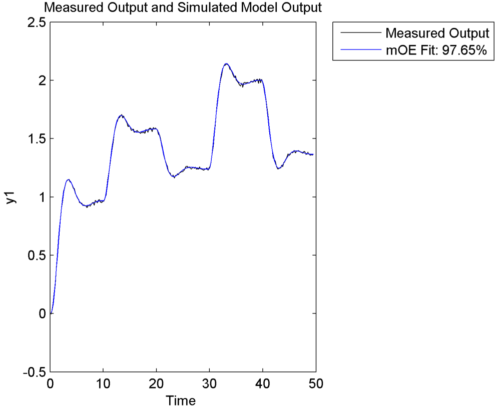
Formula explicită:

$$\begin{aligned} y(k) + f_1 y(k-1) + f_2 y(k-2) + \ldots + f_{nf} y(k-nf) \\ = b_1 u(k-nk) + b_2 u(k-nk-1) + \ldots + b_{nb} u(k-nk-nb+1) \\ + e(k) + f_1 e(k-1) + f_2 e(k-2) + \ldots + f_{nf} e(k-nf) \end{aligned}$$

Observaţie: Ca şi înainte, structura teoretică se obţine alegând $nk = 1$ (sau schimbând B dacă $nk > 1$).

Model OE

Considerând că sistemul este de ordinul 2 fără timp mort, alegem $nb = 2, nf = 2, nk = 1$. Validare: `compare(val, mOE);`



Rezultatele sunt la fel de bune ca şi ARMAX. Sistemul satisface deci ambele structuri de model. Întrebare: Care este structura reală?

Rezumat

- Reinterpretare ARX sub formă de MEP, izolând predicţia şi eroarea
- Derivarea MEP în cazul cu model general: eroare de predicţie, predictor
- Exemplu pentru ARX de ordinul 1: eroare de predicţie, predictor
- Verificare formule predictor şi eroare prin respecializare la ARX
- Exemplu Matlab ilustrând identificarea ARX, ARMAX, şi OE
- Metoda de minimizare a erorii rămâne deschisă

Conţinut

1. Structuri de model
2. Metoda generală a minimizării erorii de predicţie
3. Rezolvarea problemei de optimizare
4. Garanţii de performanţă

Checklist

	ARX	MEP generală	ARMAX ordinul 1
predictor $\widehat{y}(k)$	✓	✓	✓
eroare de predicţie $\varepsilon(k)$	✓	✓	✓
minimizare $V(\theta)$	✓	?	?

Problema de optimizare

Obiectivul procedurii de identificare: Minimizarea erorii medii pătratice

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

unde $\varepsilon(k)$ sunt erorile de predicţie. În cazul general:

$$\varepsilon(k) = H^{-1}(q^{-1})(y(k) - G(q^{-1})u(k))$$

Soluţia: $\widehat{\theta} = \arg\min_{\theta} V(\theta)$

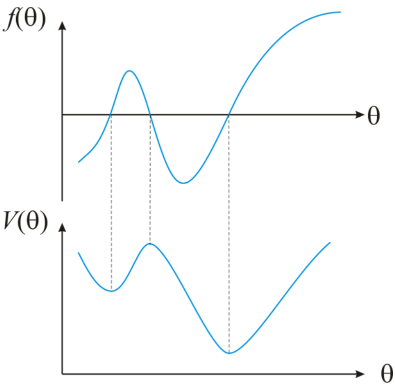
Până acum am presupus că această soluţie a fost deja obţinută, şi i-am investigat proprietăţile. Pe când în ARX puteam obţine soluţia folosind regresia liniară, în general această metodă nu va funcţiona. Principala problemă care apare în implementare este:

Cum se rezolvă problema de optimizare?

Minimizare folosind zeroul derivatei

Considerăm întâi cazul scalar, $\theta \in \mathbb{R}$.

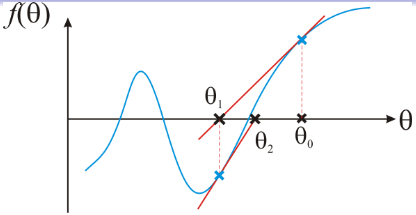
Idee: în orice minim, derivata $f(\theta) = \frac{dV}{d\theta}$ este zero. Aşadar, căutăm un zero al funcţiei $f(\theta)$.



Observaţii:

- Trebuie avut grijă ca metoda să găsească un minim şi nu un maxim sau punct de inflexiune. Acest lucru se poate verifica folosind derivata a doua, $\frac{d^2V}{d\theta^2} = \frac{df}{d\theta} > 0$.
- Este posibil ca metoda să găsească şi un punct de minim local, în care funcţia are valoare mai mare decât în minimul global.

Metoda Newton pentru găsirea rădăcinilor ($f(\theta) = 0$)



- Pornim dintr-un punct iniţial θ_0.
- La iteraţia ℓ, următorul punct $\theta_{\ell+1}$ este intersecţia între abscisă şi tangenta la f în punctul curent θ_ℓ. Geometric:

$$\boxed{\theta_{\ell+1} = \theta_\ell - \frac{f(\theta_\ell)}{\frac{df(\theta_\ell)}{d\theta}}}$$

Observaţii:

- Notaţia $\frac{df(\theta_\ell)}{d\theta}$ înseamnă valoarea derivatei $\frac{df}{d\theta}$ în punctul θ_ℓ.
- Panta tangentei este $\frac{df(\theta_\ell)}{d\theta}$.
- $\theta_{\ell+1}$ este cea mai bună estimare a soluţiei dată fiind informaţia din punctul curent θ_ℓ.

Metoda Newton pentru optimizare

Înlocuim $f(\theta)$ cu $\frac{dV}{d\theta}$ pentru a ne întoarce la problema de optimizare:

$$\theta_{\ell+1} = \theta_\ell - \frac{\frac{dV(\theta_\ell)}{d\theta}}{\frac{d^2V(\theta_\ell)}{d\theta^2}}$$

Gradientul şi Hessianul

Pentru a generaliza de la cazul scalar la $\theta \in \mathbb{R}^n$, avem nevoie de derivatele de ordinul 1 şi 2 ale lui $V(\theta)$, ţinând cont că $V : \mathbb{R}^n \to \mathbb{R}$. Atunci:

$$\frac{dV}{d\theta} = \begin{bmatrix} \frac{\partial V}{\partial \theta_1} \\ \frac{\partial V}{\partial \theta_2} \\ \vdots \\ \frac{\partial V}{\partial \theta_n} \end{bmatrix}, \quad \frac{d^2 V}{d\theta^2} = \begin{bmatrix} \frac{\partial^2 V}{\partial \theta_1^2} & \frac{\partial^2 V}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 V}{\partial \theta_1 \theta_n} \\ \frac{\partial^2 V}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 V}{\partial \theta_2^2} & \cdots & \frac{\partial^2 V}{\partial \theta_2 \theta_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 V}{\partial \theta_n \partial \theta_1} & \frac{\partial^2 V}{\partial \theta_n \theta_2} & \cdots & \frac{\partial^2 V}{\partial \theta_n^2} \end{bmatrix}$$

Gradientul $\frac{dV}{d\theta}$ este un vector de lungimea n, şi Hessianul $\frac{d^2 V}{d\theta^2}$ este o matrice de dimensiunea $n \times n$.

Metoda Newton pentru optimizare

Pornim de la formula din cazul scalar:

$$\theta_{\ell+1} = \theta_\ell - \frac{\frac{dV(\theta_\ell)}{d\theta}}{\frac{d^2V(\theta_\ell)}{d\theta^2}}$$

şi generalizăm folosind vectorul de gradient şi matricea cu Hessianul:

$$\theta_{\ell+1} = \theta_\ell - \left[\frac{d^2V(\theta_\ell)}{d\theta^2}\right]^{-1} \frac{dV(\theta_\ell)}{d\theta}$$

Adăugăm un pas $\alpha_\ell > 0$. Forma finală:

$$\boxed{\theta_{\ell+1} = \theta_\ell - \alpha_\ell \left[\frac{d^2V(\theta_\ell)}{d\theta^2}\right]^{-1} \frac{dV(\theta_\ell)}{d\theta}}$$

Observaţie: Pasul ajută la convergenţa metodei, menţinând actualizările în zona locală în care funcţia este aproximativ pătratică.

Criteriu de oprire

Algoritmul poate fi oprit:

- Când diferenţa între doi vectori consecutivi de parametri este mică, de ex. $\max_{i=1}^{n} |\theta_{i,\ell+1} - \theta_{i,\ell}|$ este mai mic decât un prag prestabilit.

sau

- Când numărul de iteraţii ℓ depăşeşte un maxim prestabilit.

Observaţie: Până acum, nimic din metoda lui Newton nu e specific identificării sistemelor! Metoda funcţionează în general, pentru orice problemă de optimizare. În cele ce urmează, revenim la problema de identificare din MEP.

Calculul derivatelor

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

Ţinânt cont că $\varepsilon(k)$ depinde de θ, avem:

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

$$\frac{d^2 V}{d\theta^2} = \frac{2}{N}\sum_{k=1}^{N} \frac{d\varepsilon(k)}{d\theta}\left[\frac{d\varepsilon(k)}{d\theta}\right]^\top + \frac{2}{N}\sum_{k=1}^{N} \varepsilon(k)\frac{d^2\varepsilon(k)}{d\theta^2}$$

unde:

- $\frac{d\varepsilon(k)}{d\theta}$ este derivata vectorială şi $\frac{d^2\varepsilon(k)}{d\theta^2}$ este Hessianul lui $\varepsilon(k)$.
- $\frac{d\varepsilon(k)}{d\theta}[\frac{d\varepsilon(k)}{d\theta}]^\top$ este o matrice $n \times n$.

Gauss-Newton

Ignorăm cel de-al doilea termen în Hessianul lui V şi îl folosim doar pe primul:

$$\mathcal{H} = \frac{2}{N} \sum_{k=1}^{N} \frac{d\varepsilon(k)}{d\theta} \left[\frac{d\varepsilon(k)}{d\theta} \right]^\top$$

Obţinem metoda Gauss-Newton:

$$\theta_{\ell+1} = \theta_\ell - \alpha_\ell \mathcal{H}^{-1} \frac{dV(\theta_\ell)}{d\theta}$$

Motivare:

- Forma pătratică a lui $\mathcal{H}$ duce la un comportament mai bun al algoritmului.
- Calculele sunt mai simple.

Calculul detaliat al derivatei $\frac{d\varepsilon(k)}{d\theta}$ depinde de structura aleasă pentru model.

Checklist

	ARX	MEP generală	ARMAX ordinul 1
predictor $\widehat{y}(k)$	$\checkmark$	$\checkmark$	$\checkmark$
eroare de predicţie $\varepsilon(k)$	$\checkmark$	$\checkmark$	$\checkmark$
minimizare $V(\theta)$	$\checkmark$	$\checkmark$	**?**

Exemplu: ARMAX de ordinul 1

Reamintim modelul ARMAX de ordinul 1 şi eroarea sa de predicţie:

$$y(k) = -ay(k-1) + bu(k-1) + ce(k-1) + e(k)$$
$$\varepsilon(k) = -c\varepsilon(k-1) + y(k) + ay(k-1) - bu(k-1)$$

Avem nevoie de $\frac{d\varepsilon(k)}{d\theta} = [\frac{\partial\varepsilon(k)}{\partial a}, \frac{\partial\varepsilon(k)}{\partial b}, \frac{\partial\varepsilon(k)}{\partial c}]^\top$. Derivând a doua ecuaţie:

$$\frac{\partial\varepsilon(k)}{\partial a} = -c\frac{\partial\varepsilon(k-1)}{\partial a} + y(k-1)$$
$$\frac{\partial\varepsilon(k)}{\partial b} = -c\frac{\partial\varepsilon(k-1)}{\partial b} - u(k-1)$$
$$\frac{\partial\varepsilon(k)}{\partial c} = -c\frac{\partial\varepsilon(k-1)}{\partial c} - \varepsilon(k-1)$$

$\frac{\partial\varepsilon(k)}{\partial a}, \frac{\partial\varepsilon(k)}{\partial b}, \frac{\partial\varepsilon(k)}{\partial c}$ sunt semnale dinamice! Ele pot fi calculate folosind formulele recursive de mai sus, pornind de ex. de la valori iniţiale 0.

Metoda generală de minimizare a erorii de predicţie

iniţializează θ_0, indexul de iteraţie $\ell = 0$
repeat
dat fiind vectorul curent de parametri θ_ℓ,
calculează cu formulele recursive $\frac{d\varepsilon(k)}{d\theta}$, $k = 1, \dots, n$
înlocuieşte $\frac{d\varepsilon(k)}{d\theta}$ în formulele $\frac{dV}{d\theta}$, $\mathcal{H}$:

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

$$\mathcal{H} = \frac{2}{N} \sum_{k=1}^{N} \frac{d\varepsilon(k)}{d\theta} \left[\frac{d\varepsilon(k)}{d\theta} \right]^\top$$

găseşte $\theta_{\ell+1}$ cu actualizarea Gauss-Newton:

$$\theta_{\ell+1} = \theta_\ell - \alpha_\ell \mathcal{H}^{-1} \frac{dV(\theta_\ell)}{d\theta}$$

incrementează indexul: $\ell = \ell + 1$
until $\theta_{\ell+1} - \theta_\ell$ este destul de mic, sau numărul maxim de iteraţii este atins

Checklist

	ARX	MEP generală	ARMAX ordinul 1
predictor $\widehat{y}(k)$	✓	✓	✓
eroare de predicţie $\varepsilon(k)$	✓	✓	✓
minimizare $V(\theta)$	✓	✓	✓

Rezumat

- Nevoia de regresie neliniară
- Metoda lui Newton pentru găsirea rădăcinilor, cazul scalar
- Metoda lui Newton pentru optimizare, cazul scalar
- Generalizare la n dimensiuni; calculul gradientului şi Hessianului
- Optimizarea Gauss-Newton
- Exemplu pentru ARMAX de ordinul 1, cu formule şi algoritm complete

Conţinut

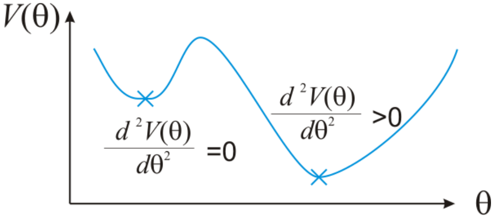
1. Structuri de model
2. Metoda generală a minimizării erorii de predicţie
3. Rezolvarea problemei de optimizare
4. Garanţii de performanţă

Ipoteze

Ipoteze (simplificate)

1. Semnalele $u(k)$ şi $y(k)$ sunt procese stohastice staţionare.
2. Semnalul de intrare $u(k)$ are un ordin de PE suficient de mare.
3. Hessianul $\frac{d^2 V}{d\theta^2}$ este inversabil în punctele de minim ale funcţiei MSE V.

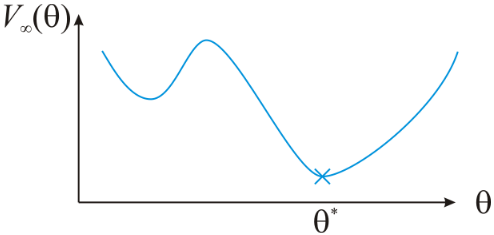
Reamintim funcţia de MSE $V(\theta) = \frac{1}{N}\sum_{k=1}^{N} \varepsilon^2(k)$. Ipoteza 3 garantează că V nu este "plat" în jurul minimelor.



Garanţie

Teorema 1

Definim limita $V_\infty(\theta) = \lim_{N\to\infty} V(\theta)$. Date fiind Ipotezele 1–3, soluţia de identificare $\widehat{\theta} = \arg\min_\theta V(\theta)$ converge la un punct de minim θ^* al $V_\infty(\theta)$ când numărul de date $N \to \infty$.



Observaţie: Rezultatul este unul de consistenţă, în limita unui număr infinit de date.

Ipoteze adiţionale

Ipoteze (simplificate)

4. Sistemul adevărat satisface structura de model aleasă. Anume, există cel puţin un vector corect θ_0 astfel încât pentru orice intrare $u(k)$ şi ieşirea corespunzătoare $y(k)$ a sistemului adevărat, avem:

$$y(k) = G(q^{-1}; \theta_0)u(k) + H(q^{-1}; \theta_0)e(k)$$

   unde $e(k)$ este *zgomot alb*.

5. Intrarea $u(k)$ este independentă de zgomotul $e(k)$ (experimentul este efectuat în buclă deschisă).

Garanţie adiţională

Teorema 2

Date fiind Ipotezele 1-5, $\widehat{\theta}$ converge la vectorul corect de parametri θ_0 când $N \to \infty$.

Observaţie: Şi această garanţie este una de consistenţă. Pe când Teorema 1 garantează o soluţie de eroare minimă, Teorema 2 ne spune că această soluţie corespunde sistemului adevărat, *în condiţiile în care* sistemul satisface structura aleasă pentru model.