

Identificarea sistemelor

Ingineria sistemelor, anul 3
Universitatea Tehnică din Cluj-Napoca

Lucian Buşoniu



Partea VII

Metoda minimizării erorii de predicție

Conținut

- 1 Structuri de model
- 2 Metoda generală a minimizării erorii de predicție
- 3 Rezolvarea problemei de optimizare
- 4 Garanții de performanță

Clasificare

Reamintim taxonomia modelelor din Partea I:

După numărul de parametri:

- 1 **Modele parametrice**: au formă fixă (formulă matematică), număr cunoscut și 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 cunoștințele 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: parțial cunoscute

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

Conținut

- 1 Structuri de model
- 2 Metoda generală a minimizării erorii de predicție
- 3 Rezolvarea problemei de optimizare
- 4 Garanții de performanță

Motivație

MEP poate fi văzută ca o extensie a metodei ARX la structuri de model mult mai generale, fiind așadar 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 și H sunt *funcții de transfer* în timp discret – fracții de polinoame. Semnalul $e(t)$ este zgomot alb de medie zero.

- G este funcția de transfer intrare-ieșire.
- H este funcția de transfer a perturbației.

Structura generală: Formă explicită

Prin gruparea factorilor comuni ai numitoarelor din G și H în $A(q^{-1})$, obținem 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 :

$$A(q^{-1}) = 1 + a_1 q^{-1} + \dots + a_{na} q^{-na}$$

$$B(q^{-1}) = b_1 q^{-1} + \dots + b_{nb} q^{-nb}$$

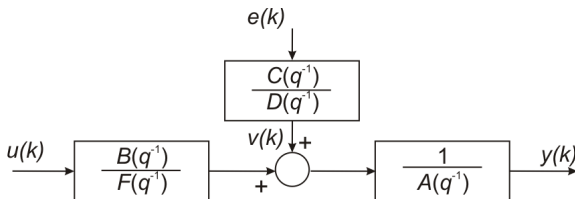
$$C(q^{-1}) = 1 + c_1 q^{-1} + \dots + c_{nc} q^{-nc}$$

$$F(q^{-1}) = 1 + f_1 q^{-1} + \dots + f_{nf} q^{-nf}$$

$$D(q^{-1}) = 1 + d_1 q^{-1} + \dots + d_{nd} q^{-nd}$$

Structura generală: Formă grafică

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



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

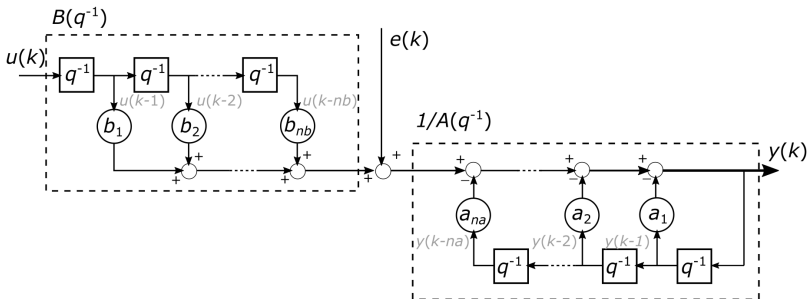
Structura ARMAX

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

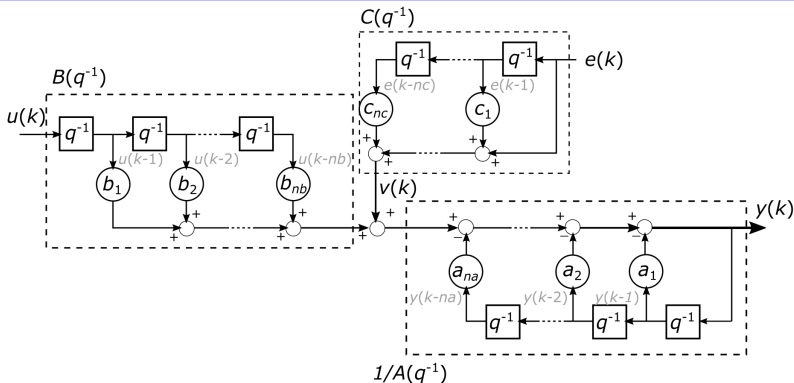
$$A(q^{-1})y(k) = B(q^{-1})u(k) + C(q^{-1})e(k)$$

Nume: **AutoRegresiv** (dependența de ieșirile anterioare), **cu Medie Alunecătoare** (se referă la modelul perturbației) **cu intrare eXogenă** (dependența de u)

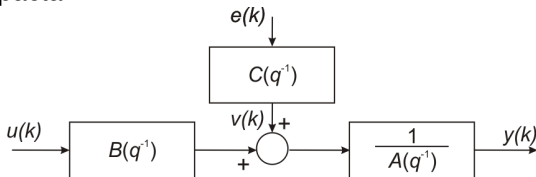
Reamintim forma grafică detaliată pentru ARX



ARMAX: formă grafică



Formă compactă:



ARMAX: formă explicită

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

$$A(q^{-1}) = 1 + a_1 q^{-1} + \dots + a_{na} q^{-na}$$

$$B(q^{-1}) = b_1 q^{-1} + \dots + b_{nb} q^{-nb}$$

$$C(q^{-1}) = 1 + c_1 q^{-1} + \dots + c_{nc} q^{-nc}$$

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

cu vectorul de parametri:

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

Exemplu recurent: ARMAX de ordinul 1

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

$$(1 + aq^{-1})y(k) = bq^{-1}u(k) + (1 + cq^{-1})e(k)$$

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

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

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

Caz special de ARMAX: ARX

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

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

exact modelul ARX pe care l-am studiat deja.

În comparație cu ARX, ARMAX poate modela perturbații 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, obținem:

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

$$= \sum_{j=0}^{M-1} h(j)u(k-j) + e(k)$$

modelul FIR.

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

Structura de tip eroare de ieșire

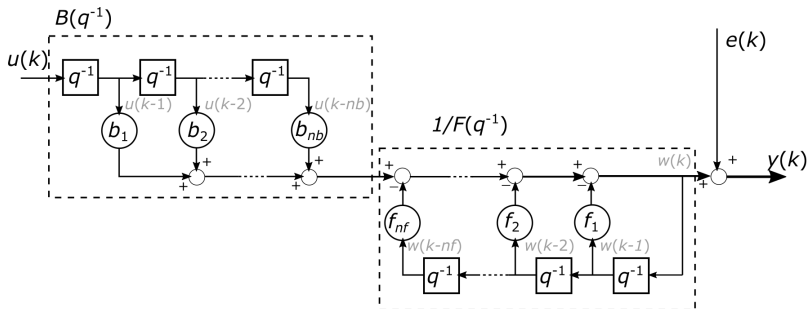
Model de tip **eroare de ieșire, OE** (en. *Output Error*):

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

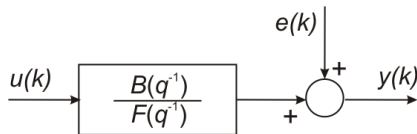
obținut pentru $na = nc = nd = 0$, adică $A = C = D = 1$.

Structura corespunde unui zgomot simplu, aditiv la ieșire (de unde reiese și numele).

OE: formă grafică

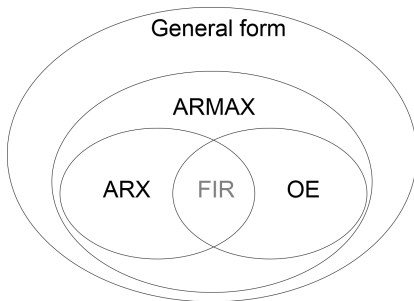


Formă compactă:



Rezumat si relație globală între modele

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



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

Exerciții

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

Conținut

1 Structuri de model

2 Metoda generală a minimizării erorii de predicție

- Pas preliminar: ARX revizitat
- Cazul general
- Exemplu: ARMAX de ordinul 1
- Exemplu Matlab

3 Rezolvarea problemei de optimizare

4 Garanții de performanță

ARX reinterpretat ca MEP

- ① Dat fiind vectorul de parametri θ , calculăm predicțiile la fiecare pas, $\hat{y}(k) = \varphi^\top(k)\theta$.
- ② Calculăm erorile de predicție la fiecare pas, $\varepsilon(k) = y(k) - \hat{y}(k)$.
- 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 obține extinzând procedura la structura generală de model.

ARX reinterpretat ca MEP (continuare)

Observații:

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

Checklist

Pentru fiecare structură de model, va trebui să trecem prin acești trei pași: predictor, eroare de predicție, și minimizarea MSEului.

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

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

Conținut

- 1 Structuri de model
- 2 **Metoda generală a minimizării erorii de predicție**
 - Pas preliminar: ARX revizitat
 - **Cazul general**
 - Exemplu: ARMAX de ordinul 1
 - Exemplu Matlab
- 3 Rezolvarea problemei de optimizare
- 4 Garanții de performanță

Reamintim: Structura generală de model

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

unde G și H sunt funcții 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 ecuațiilor, și vom înțelege explicit că literele mari se referă la funcții de transfer sau polinoame în variabila q^{-1} .

Eroarea de predicție

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 fracției de polinoame H .

Predictorul va fi derivat în așa fel încât eroarea de predicție să fie egală cu zgomotul, $\varepsilon(k) = y(k) - \hat{y}(k) = e(k)$. Așadar, aceeași formulă ca și mai sus se poate folosi pentru a calcula $\varepsilon(k)$:

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

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

Predictor

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

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

Observație: Pentru a avea un predictor *causal*, care depinde doar de valorile anterioare ale ieșirii și intrării, este necesar ca $G(0) = 0$ și $H(0) = 1$. Aceste condiții sunt satisfăcute deoarece $B(0) = 0, A(0) = C(0) = D(0) = 1$.

Preview: Găsirea parametrilor și folosirea modelului

Găsirea parametrilor: Odată ce procedura pentru calculul erorilor este disponibilă, parametrii θ sunt găsiți prin minimizarea funcției 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 secțiunea următoare.

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

Checklist

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

Table of contents

1 Structuri de model

2 Metoda generală a minimizării erorii de predicție

- Pas preliminar: ARX revizitat
- Cazul general
- Exemplu: ARMAX de ordinul 1
- Exemplu Matlab

3 Rezolvarea problemei de optimizare

4 Garanții de performanță

Exemplu recurent: ARMAX ordinul 1

Reamintim ARMAX:

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

Prin plasare în forma standard, obținem:

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

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

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

ARMAX ordinul 1: Eroare de predicție

Reamintim formula generală a erorii de predicție:

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

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

$$\varepsilon(k) = \frac{A}{C} \left(y(k) - \frac{B}{A} u(k) \right)$$

$$C\varepsilon(k) = Ay(k) - Bu(k)$$

$$(1 + cq^{-1})\varepsilon(k) = (1 + aq^{-1})y(k) - bq^{-1}u(k)$$

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

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

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

ARMAX ordinul 1: Predictor

Reamintim formula generală a predictorului:

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

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

$$\hat{y}(k) = \left(1 - \frac{A}{C}\right)y(k) + \frac{A}{C} \frac{B}{A}u(k)$$

$$C\hat{y}(k) = (C - A)y(k) + Bu(k)$$

$$(1 + cq^{-1})\hat{y}(k) = (1 + cq^{-1} - 1 - aq^{-1})y(k) + bq^{-1}u(k)$$

$$\hat{y}(k) + c\hat{y}(k-1) = (c - a)y(k-1) + bu(k-1)$$

$$\hat{y}(k) = -c\hat{y}(k-1) + (c - a)y(k-1) + bu(k-1)$$

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

Checklist

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

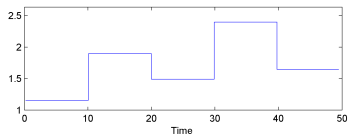
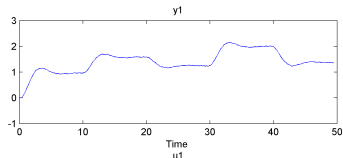
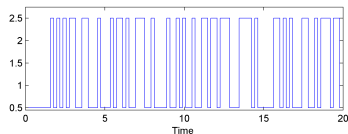
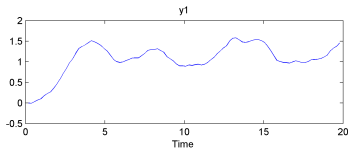
Conținut

- 1 Structuri de model
- 2 **Metoda generală a minimizării erorii de predicție**
 - Pas preliminar: ARX revizitat
 - Cazul general
 - Exemplu: ARMAX de ordinul 1
 - **Exemplu Matlab**
- 3 Rezolvarea problemei de optimizare
- 4 Garanții de performanță

Date experimentale

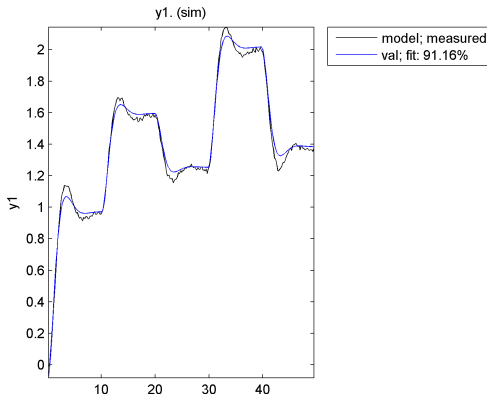
Folosim din nou datele experimentale pe care am identificat modelul ARX mai demult.

```
plot(id); și 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 conținând ordinele polinoamelor A , B , C și întârzierea nk .

Ca și pentru ARX, structura include explicit întârzierea minimă nk între intrări și ieșiri.

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

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

$$A(q^{-1}) = (1 + a_1 q^{-1} + a_2 q^{-2} + \dots + a_{na} q^{-na})$$

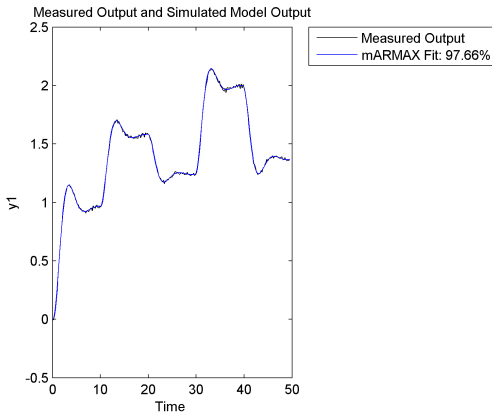
$$B(q^{-1}) = (b_1 + b_2 q^{-1} + b_{nb} q^{-nb+1})$$

$$C(q^{-1}) = (1 + c_1 q^{-1} + c_2 q^{-2} + \dots + c_{nc} q^{-nc})$$

Observație: Ca pentru ARX, structura teoretică se obține alegând $nk = 1$ (și pentru a reprezenta $nk > 1$ în structura teoretică, adăugăm coeficienți inițiali 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)`;

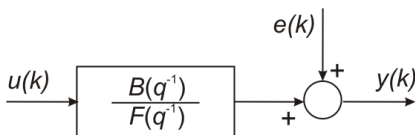


Spre deosebire de ARX, rezultatele sunt bune. Modelul mai flexibil al perturbației își arată utilitatea.

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 conținând ordinele polinoamelor B , F , și întârzierea nk .

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

$$B(q^{-1}) = (b_1 + b_2 q^{-1} + b_{nb} q^{-nb+1})$$

$$F(q^{-1}) = (1 + f_1 q^{-1} + f_2 q^{-2} + \dots + f_{nf} q^{-nf})$$

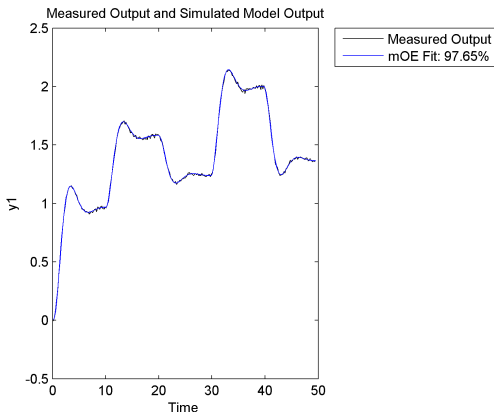
Formula explicită:

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

Observație: Ca și înainte, structura teoretică se obține 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 și ARMAX. Sistemul satisface deci ambele structuri de model. **Întrebare:** Care este structura reală?

Rezumat

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

Conținut

- 1 Structuri de model
- 2 Metoda generală a minimizării erorii de predicție
- 3 Rezolvarea problemei de optimizare**
- 4 Garanții de performanță

Checklist

	ARX	MEP generală	ARMAX ordinul 1
predictor $\hat{y}(k)$	✓	✓	✓
eroare de predicție $\varepsilon(k)$	✓	✓	✓
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 predicție. În cazul general:

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

Soluția: $\hat{\theta} = \arg \min_{\theta} V(\theta)$

Până acum am presupus că această soluție a fost deja obținută, și i-am investigat proprietățile. Pe când în ARX puteam obține soluția folosind regresia liniară, în general această metodă nu va funcționa. Principala problemă care apare în implementare este:

Cum se rezolvă problema de optimizare?

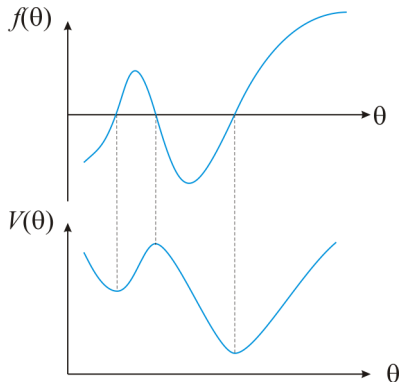
Minimizare folosind zeroul derivatei

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

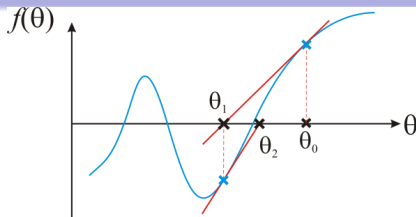
Idee: în orice minim, derivata $f(\theta) = \frac{dV}{d\theta}$ este zero. Așadar, căutăm un zero al funcției $f(\theta)$.

Observații:

- Trebuie avut grijă ca metoda să găsească un minim și 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 și un punct de minim local, în care funcția are valoare mai mare decât în minimul global.



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



- Pornim dintr-un punct inițial θ_0 .
- La iterația ℓ , următorul punct $\theta_{\ell+1}$ este intersecția între abscisă și **tangenta** la f în punctul curent θ_ℓ . Geometric:

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

Observații:

- Notăția $\frac{df(\theta_\ell)}{d\theta}$ 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 soluției dată fiind informația 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:

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

Gradientul și Hessianul

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

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

Gradientul $\frac{dV}{d\theta}$ este un vector de lungimea n , și **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:

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

și generalizăm folosind vectorul de gradient și 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:

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

Observație: Pasul ajută la convergența metodei, menținând actualizările în zona locală în care funcția este aproximativ pătratică.

Criteriu de oprire

Algoritmul poate fi oprit:

- Când diferența 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 iterații ℓ depășește un maxim prestabilit.

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

Calculul derivatelor

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

Ținând 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 și $\frac{d^2 \varepsilon(k)}{d\theta^2}$ este Hessianul lui $\varepsilon(k)$.
- $\frac{d\varepsilon(k)}{d\theta} \left[\frac{d\varepsilon(k)}{d\theta} \right]^\top$ este o matrice $n \times n$.

Gauss-Newton

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

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

Obținem 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 $\hat{y}(k)$	✓	✓	✓
eroare de predicție $\varepsilon(k)$	✓	✓	✓
minimizare $V(\theta)$	✓	✓	?

Exemplu: ARMAX de ordinul 1

Reamintim modelul ARMAX de ordinul 1 și eroarea sa de predicție:

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

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

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

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

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

$$\frac{\partial \varepsilon(k)}{\partial c} = -c \frac{\partial \varepsilon(k-1)}{\partial c} - \varepsilon(k-1)$$

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

Metoda generală de minimizare a erorii de predicție

inițializează θ_0 , indexul de iterație $\ell = 0$

repeat

dat fiind vectorul curent de parametri θ_ℓ ,

calculează cu formulele recursive $\frac{d\varepsilon(k)}{d\theta}$, $k = 1, \dots, N$

înlocuiește $\frac{d\varepsilon(k)}{d\theta}$ în formulele $\frac{dV}{d\theta}$, $\mathcal{H}$:

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

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

găsește $\theta_{\ell+1}$ 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 iterații este atins

Checklist

	ARX	MEP generală	ARMAX ordinul 1
predictor $\hat{y}(k)$	✓	✓	✓
eroare de predicție $\varepsilon(k)$	✓	✓	✓
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 și Hessianului
- Optimizarea Gauss-Newton
- Exemplu pentru ARMAX de ordinul 1, cu formule și algoritmi complete

Conținut

- 1 Structuri de model
- 2 Metoda generală a minimizării erorii de predicție
- 3 Rezolvarea problemei de optimizare
- 4 **Garanții de performanță**

Ipoteze

Ipoteze

- 1 Sistemul adevărat satisface structura de model aleasă. Anume, există cel puțin un vector corect θ_0 astfel încât pentru orice intrare $u(k)$ și ieșirea corespunzătoare $y(k)$ a sistemului adevărat, avem:

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

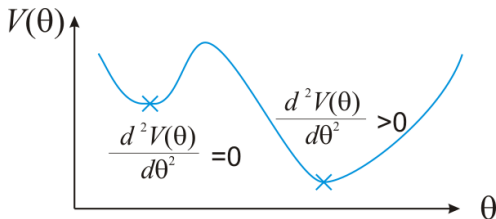
unde $e(k)$ este *zgomot alb*.

- 2 Semnalele $u(k)$ și $y(k)$ sunt procese stohastice staționare.
- 3 Semnalul de intrare $u(k)$ are un ordin de PE suficient de mare, și este independent de zgomotul $e(k)$ (experimentul este efectuat în buclă deschisă).
- 4 Hessianul $\frac{d^2 V}{d\theta^2}$ este inversabil în punctele de minim ale funcției MSE V .

Discuție despre Hessian

- 4 Hessianul $\frac{d^2 V}{d\theta^2}$ este inversabil în punctele de minim ale funcției MSE V .

Reamintim funcția 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.



Garanție

Teorema 2

Date fiind Ipotezele 1-4, $\hat{\theta}$ converge la vectorul corect de parametri θ_0 când $N \rightarrow \infty$.

Observație: Rezultatul este unul de **consistență**, în limita unui număr infinit de date.