

Identificarea sistemelor

Ingineria sistemelor, anul 3
Universitatea Tehnică din Cluj-Napoca

Lucian Buşoniu



Partea V

Metoda ARX

Conținut

- 1 Derivare analitică
- 2 Exemplu Matlab
- 3 Garanție de performanță
- 4 Intrări și ieșiri multiple

Vom rămâne în cazul cu intrări și ieșiri scalare, cu excepția ultimei secțiuni.

Clasificare

Reamintim clasificarea modelelor din Partea I:

- 1 Modele mentale sau verbale
- 2 Grafice și tabele (neparametrice)
- 3 Modele matematice, cu două subtipuri:
 - Modele analitice, din principii de bază
 - **Modele din identificarea sistemelor**

Metoda ARX produce modele *parametrice*, de tip polinomial.

De ce ARX?

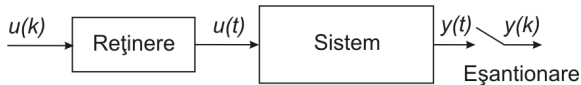
- Metodă pentru orice ordin, implementabilă programatic, cu garanții – ca și analiza de corelație
- Spre deosebire de analiza de corelație, furnizează un model *compact*, numărul de parametri fiind proporțional cu ordinul sistemului

Conținut

- 1 Derivare analitică
- 2 Exemplu Matlab
- 3 Garanție de performanță
- 4 Intrări și ieșiri multiple

Reamintim: Timp discret

Rămânem în cazul de timp discret:



Structura modelului ARX

În structura **ARX**, ieșirea $y(k)$ la pasul curent este calculată din intrările și ieșirile la pași precedenți:

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

echivalent to

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

$e(k)$ este zgomotul la pasul k .

Parametrii modelului: a_1, a_2, \dots, a_{na} și b_1, b_2, \dots, b_{nb} .

Nume: Model **AutoRegresiv** ($y(k)$ depinde de valorile y precedente)
cu intrare eXogenă (dependență de u)

Reprezentare polinomială

Operatorul de deplasare q^{-1} :

$$q^{-1}z(k) = z(k - 1)$$

unde $z(k)$ este orice semnal în timp discret.

Folosind q^{-1} , avem:

$$\begin{aligned} y(k) + a_1y(k - 1) + a_2y(k - 2) + \dots + a_nay(k - na) \\ = (1 + a_1q^{-1} + a_2q^{-2} + \dots + a_naq^{-na})y(k) =: A(q^{-1})y(k) \end{aligned}$$

și:

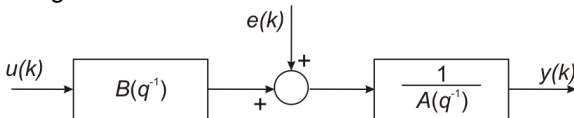
$$\begin{aligned} b_1u(k - 1) + b_2u(k - 2) + \dots + b_nbu(k - nb) \\ = (b_1q^{-1} + b_2q^{-2} + \dots + b_nbq^{-nb})u(k) =: B(q^{-1})u(k) \end{aligned}$$

Modelul ARX în formă polinomială

Ca urmare, modelul ARX se poate scrie compact:

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

Reprezentare grafică:



fiindcă:

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

Observație: Modelul ARX este general, poate descrie orice relație liniară între intrări și ieșiri. Zgomotul participă însă în model într-o manieră restrictivă, și mai târziu vom descrie modele care generalizează acest element.

Model în forma de regresie liniară

Revenind la reprezentarea explicită recursivă:

$$\begin{aligned}y(k) &= -a_1 y(k-1) - a_2 y(k-2) - \dots - a_{na} y(k-na) \\ &\quad b_1 u(k-1) + b_2 u(k-2) + \dots + b_{nb} u(k-nb) + e(k) \\ &= [-y(k-1) \quad \dots \quad -y(k-na) \quad u(k-1) \quad \dots \quad u(k-nb)] \\ &\quad \cdot [a_1 \quad \dots \quad a_{na} \quad b_1 \quad \dots \quad b_{nb}]^T + e(k) \\ &=: \varphi^T(k) \theta + e(k)\end{aligned}$$

ARX se supune așadar formei standard de model din regresia liniară!

Vector de regresori: $\varphi \in \mathbb{R}^{na+nb}$, ieșirile și intrările precedente.

Vector de parametri: $\theta \in \mathbb{R}^{na+nb}$, coeficienții polinoamelor.

Problema de identificare

Considerăm un set de date $u(k), y(k), k = 1, \dots, N$, din care trebuie aflați parametri θ ai modelului.

Pentru orice k :

$$y(k) = \varphi^T(k)\theta + \varepsilon(k)$$

unde $\varepsilon(k)$ este interpretat acum ca o eroare în ecuație (de unde și notația schimbată).

Obiectiv: minimizarea erorii medii pătratice:

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

Observație: Când $k \leq na, nb$, valori ale u și y la momente de timp negative sau zero sunt necesare pentru construirea φ . Aceste valori pot fi luate 0 (presupunând condiții inițiale nule).

Sistem liniar de ecuații

$$y(1) = [-y(0) \quad \cdots \quad -y(1-na) \quad u(0) \quad \cdots \quad u(1-nb)] \theta$$

$$y(2) = [-y(1) \quad \cdots \quad -y(2-na) \quad u(1) \quad \cdots \quad u(2-nb)] \theta$$

...

$$y(N) = [-y(N-1) \quad \cdots \quad -y(N-na) \quad u(N-1) \quad \cdots \quad u(N-nb)] \theta$$

Forma matriceală:

$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} -y(0) & \cdots & -y(1-na) & u(0) & \cdots & u(1-nb) \\ -y(1) & \cdots & -y(2-na) & u(1) & \cdots & u(2-nb) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -y(N-1) & \cdots & -y(N-na) & u(N-1) & \cdots & u(N-nb) \end{bmatrix} \cdot \theta$$

$$Y = \Phi \theta$$

cu notațiile $Y \in \mathbb{R}^N$ și $\Phi \in \mathbb{R}^{N \times (na+nb)}$.

Soluția ARX

Din regresia liniară, parametrii care minimizează $\frac{1}{2} \sum_{k=1}^N \varepsilon(k)^2$ sunt:

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T Y$$

Cum noua funcție obiectiv $V(\theta) = \frac{1}{N} \sum_{k=1}^N \varepsilon(k)^2$ este proporțională cu criteriul de mai sus, aceeași soluție minimizează și $V(\theta)$.

Numărul N de date poate fi foarte mare, în care caz forma de mai sus este impractică pentru identificare. O formă mai bună este:

$$\Phi^T \Phi = \sum_{k=1}^N \varphi(k) \varphi^T(k), \quad \Phi^T Y = \sum_{k=1}^N \varphi(k) y(k)$$
$$\Rightarrow \hat{\theta} = \left[\sum_{k=1}^N \varphi(k) \varphi^T(k) \right]^{-1} \left[\sum_{k=1}^N \varphi(k) y(k) \right]$$

(Reamintim ideea similară din regresia liniară.)

Soluția ARX (continuare)

O problemă rămasă: suma celor N termeni poate fi mare, ducând la probleme numerice: (matrice de numere foarte mari) $^{-1}$ · vector de numere foarte mari.

Soluție: Normalizarea valorilor prin împărțirea fiecărui element cu N . În ecuații, N se simplifică deci nu are nici un efect asupra dezvoltării analitice, dar în practică această împărțire menține numerele la valori rezonabile.

$$\hat{\theta} = \left[\frac{1}{N} \sum_{k=1}^N \varphi(k) \varphi^T(k) \right]^{-1} \left[\frac{1}{N} \sum_{k=1}^N \varphi(k) y(k) \right]$$

Utilizarea modelului ARX

Predicție cu un pas înainte: Secvența de ieșiri reale este cunoscută, deci toate valorile precedente sunt disponibile și pot fi înlocuite în formulă, pe lângă coeficienții extrași din θ :

$$\hat{y}(k) = -a_1 y(k-1) - a_2 y(k-2) - \dots - a_{na} y(k-na) \\ b_1 u(k-1) + b_2 u(k-2) + \dots + b_{nb} u(k-nb)$$

Semnalele la momente negative și zero de timp pot fi luate 0.

Exemplu: În ziua $k-1$, predicția meteo pentru ziua k .

Simulare: Ieșirile reale sunt necunoscute, trebuiesc înlocuite cu ieșirile simulate la iterații anterioare: $y(k-i)$ înlocuit cu $\hat{y}(k-i)$:

$$\hat{y}(k) = -a_1 \hat{y}(k-1) - a_2 \hat{y}(k-2) - \dots - a_{na} \hat{y}(k-na) \\ b_1 u(k-1) + b_2 u(k-2) + \dots + b_{nb} u(k-nb)$$

(ieșirile simulate la pași negativi și zero pot fi și ele luate 0.)

Exemplu: Simularea răspunsului unui avion la comenzi ale pilotului într-o situație de urgență, care ar fi periculoasă pentru sistemul real.

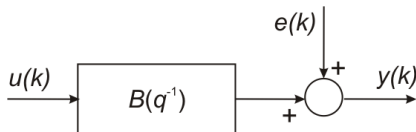
FIR este un caz particular de ARX

Alegând $A = 1$ ($na = 0$) 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 din analiza de corelație!

Mai detaliat, luăm $nb = M - 1$ și $b_j = h(j)$. De notat că $h(0)$, răspunsul la impuls la momentul 0, se presupune egal cu 0 – adică sistemul nu răspunde instantaneu la schimbări ale intrării.



Diferența fundamentală între ARX și FIR

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

$$\text{FIR: } y(k) = B(q^{-1})u(k) + e(k)$$

Cum ARX include o relație recursivă între ieșirea curentă și cele precedente, este suficient să luăm ordinele na și nb egale cu ordinul sistemului dinamic.

Modelul FIR are nevoie de un ordin nb (lungime M) suficient de mare pentru a modela întregul regim tranzitoriu al răspunsului la impuls (în mod ideal recuperăm modelul corect doar dacă $M \rightarrow \infty$).

⇒ mai mulți parametri ⇒ mai multe date necesare pentru identificare.

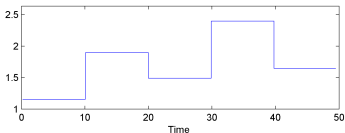
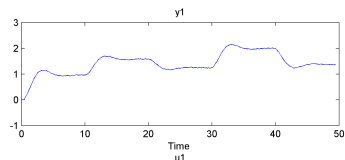
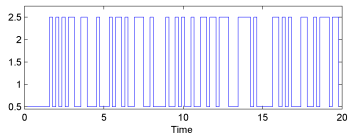
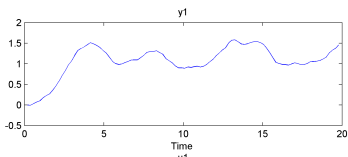
Conținut

- 1 Derivare analitică
- 2 Exemplu Matlab**
- 3 Garanție de performanță
- 4 Intrări și ieșiri multiple

Date experimentale

Se dau două seturi de date, unul pentru identificare, celălalt pentru validare.

```
plot(id); și plot(val);
```



Observații: Intrarea de identificare: *semnal pseudo-aleator binar*.
Intrarea de validare: o secvență de semnale treaptă.

Identificarea unui model ARX

```
model = arx(id, [na, nb, nk]);
```

Argumente funcție:

- 1 Datele de identificare.
- 2 Vector conținând ordinele A și B , și întârzierea nk .

Structura diferită de cea teoretică: include o întârziere minimă nk între intrări și ieșiri, utilă pentru a modela sistemele cu timp mort.

$$y(k) + a_1y(k-1) + a_2y(k-2) + \dots + a_nay(k-na) \\ = b_1u(k-nk) + b_2u(k-nk-1) + \dots + b_nbu(k-nk-nb+1) + e(k)$$

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

$$A(q^{-1}) = (1 + a_1q^{-1} + a_2q^{-2} + \dots + a_naq^{-na})$$

$$B(q^{-1}) = (b_1 + b_2q^{-1} + b_nbq^{-nb+1})$$

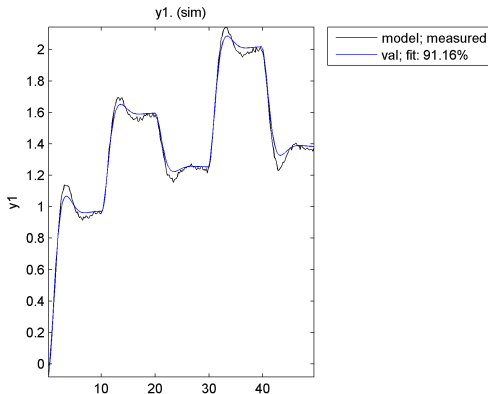
Structura teoretică se obține luând $nk = 1$. Pentru $nk > 1$, noua structură se poate transforma în cea teoretică alegând un polinom B de ordinul $nk + nb - 1$, cu primii $nk - 1$ coeficienți nuli:

$$B_{\text{theor}}(q^{-1}) = 0q^{-1} + \dots + 0q^{-nk+1} + b_1q^{-nk} + \dots + b_nbq^{-nk-nb+1}$$

Validarea modelului

Presupunând că sistemul este de ordinul 2, *în forma ARX*, și fără întârzieri, alegem $na = 2$, $nb = 2$, $nk = 1$. Validare:

```
compare(model, val);
```



Rezultatele nu sunt bune.

Selecția structurii

Alternativă: încercăm mai multe structuri diferite și o alegem pe cea mai bună.

```
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 din primul argument), îl simulează (pe datele din al doilea argument), și returnează toate valorile MSE pe prima linie din `V` (vezi `help arxstruc` pentru formatul variabilei `V`).

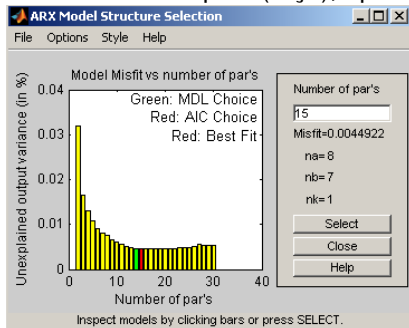
Selecția structurii (continuare)

Pentru a alege structura cu valoarea MSE minimă:

```
N = selstruc(V, 0);
```

Pentru datele noastre, $N = [8, 7, 1]$.

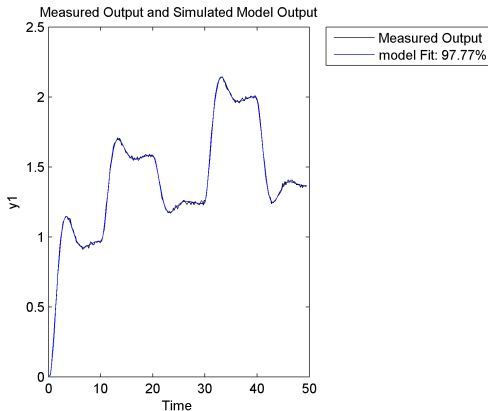
Alternativ, selecție grafică: `N = selstruc(V, 'plot');` Click pe bara corespunzătoare modelului optim (roșu), apoi “Select”, “Close”.



(Mai târziu vom discuta selecția între modele alternative cu alte criterii decât MSE.)

Validarea celui mai bun model ARX

```
model = arx(id, N); compare(model, val);
```



Rezultatele sunt mai bune. Sistemele de ordinul 8 sunt însă destul de rare în practică, probabil se întâmplă ceva mai complicat... vom relua ideea în cursurile următoare.

Conținut

- 1 Derivare analitică
- 2 Exemplu Matlab
- 3 Garanție de performanță**
- 4 Intrări și ieșiri multiple

Rezultat

Ipoteze

- 1 Există un vector corect de parametri θ_0 pentru care:

$$y(k) = \varphi(k)\theta_0 + v(k)$$

unde $v(k)$ este un proces stohastic staționar independent de $u(k)$.

- 2 $E \{ \varphi(k)\varphi^T(k) \}$ este o matrice inversabilă.
- 3 $E \{ \varphi(k)v(k) \} = 0$.

Teoremă

Identificarea ARX este **consistentă**: parametrii estimați $\hat{\theta}$ converg la cei corecți θ_0 , la limită când numărul de date crește la infinit $N \rightarrow \infty$.

Discuție ipoteze

- 1 Ipoteza 1 este echivalentă cu existența unor polinoame corecte $A_0(q^{-1})$, $B_0(q^{-1})$ pentru care:

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

Pentru a motiva Ipoteza 2, reamintim

$$\hat{\theta} = \left[\frac{1}{N} \sum_{k=1}^N \varphi(k)\varphi^T(k) \right]^{-1} \left[\frac{1}{N} \sum_{k=1}^N \varphi(k)y(k) \right]$$

Când $N \rightarrow \infty$, $\frac{1}{N} \sum_{k=1}^N \varphi(k)\varphi^T(k) \rightarrow E \{ \varphi(k)\varphi^T(k) \}$.

- 2 $E \{ \varphi(k)\varphi^T(k) \}$ este inversabilă dacă datele sunt “suficient de informative” (de ex. $u(k)$ nu trebuie să fie un feedback simplu de la $y(k)$; vezi Söderström & Stoica pentru discuții adiționale).
- 3 $E \{ \varphi(k)v(k) \} = 0$ de ex. dacă $v(k)$ este zgomot alb. Mai târziu, vom rediscuta Ipoteza 3 și rolul condiției $E \{ \varphi(k)v(k) \} = 0$.

Conținut

- 1 Derivare analitică
- 2 Exemplu Matlab
- 3 Garanție de performanță
- 4 Intrări și ieșiri multiple**

Sistem MIMO

Până acum am considerat $y(k) \in \mathbb{R}$, $u(k) \in \mathbb{R}$,
sisteme *Single-Input, Single-Output (SISO)*

Multe sisteme sunt *Multiple-Input, Multiple-Output (MIMO)*.
De ex., avion. Intrări: putere motoare, eleron, elevator, cârmă.
Ieșiri: viteză, deviații unghiulare în jurul celor trei axe.



ARX MIMO

Mai departe considerăm $y(k), e(k) \in \mathbb{R}^{ny}$, $u(k) \in \mathbb{R}^{nu}$.
Model ARX MIMO:

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

$$A(q^{-1}) = I + A_1 q^{-1} + \dots + A_{na} q^{-na}$$

$$B(q^{-1}) = B_1 q^{-1} + \dots + B_{nb} q^{-nb}$$

unde I este matricea identitate $ny \times ny$, $A_1, \dots, A_{na} \in \mathbb{R}^{ny \times ny}$,
 $B_1, \dots, B_{nb} \in \mathbb{R}^{ny \times nu}$.

Exemplu concret

Luăm $na = 1$, $nb = 2$, $ny = 2$, $nu = 3$. Atunci:

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

$$A(q^{-1}) = I + A_1q^{-1}$$

$$= I + \begin{bmatrix} a_1^{11} & a_1^{12} \\ a_1^{21} & a_1^{22} \end{bmatrix} q^{-1}$$

$$B(q^{-1}) = B_1q^{-1} + B_2q^{-2}$$

$$= \begin{bmatrix} b_1^{11} & b_1^{12} & b_1^{13} \\ b_1^{21} & b_1^{22} & b_1^{23} \end{bmatrix} q^{-1} + \begin{bmatrix} b_2^{11} & b_2^{12} & b_2^{13} \\ b_2^{21} & b_2^{22} & b_2^{23} \end{bmatrix} q^{-2}$$

Exemplu concret (continuare)

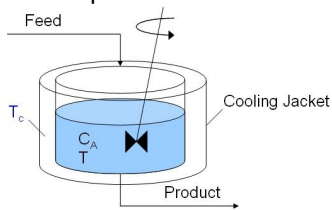
$$\begin{aligned} & \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} a_1^{11} & a_1^{12} \\ a_1^{21} & a_1^{22} \end{bmatrix} q^{-1} \right) \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} \\ &= \left(\begin{bmatrix} b_1^{11} & b_1^{12} & b_1^{13} \\ b_1^{21} & b_1^{22} & b_1^{23} \end{bmatrix} q^{-1} + \begin{bmatrix} b_2^{11} & b_2^{12} & b_2^{13} \\ b_2^{21} & b_2^{22} & b_2^{23} \end{bmatrix} q^{-2} \right) \begin{bmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{bmatrix} + \begin{bmatrix} e_1(k) \\ e_2(k) \end{bmatrix} \end{aligned}$$

Relație explicită:

$$\begin{aligned} y_1(k) &+ a_1^{11} y_1(k-1) + a_1^{12} y_2(k-1) \\ &= b_1^{11} u_1(k-1) + b_1^{12} u_2(k-1) + b_1^{13} u_3(k-1) \\ &+ b_2^{11} u_1(k-2) + b_2^{12} u_2(k-2) + b_2^{13} u_3(k-2) + e_1(k) \\ y_2(k) &+ a_1^{21} y_1(k-1) + a_1^{22} y_2(k-1) \\ &= b_1^{21} u_1(k-1) + b_1^{22} u_2(k-1) + b_1^{23} u_3(k-1) \\ &+ b_2^{21} u_1(k-2) + b_2^{22} u_2(k-2) + b_2^{23} u_3(k-2) + e_2(k) \end{aligned}$$

Exemplu Matlab

Considerăm un reactor de tip *continuous stirred-tank reactor*, CSTR:



Credit imagine: mathworks.com

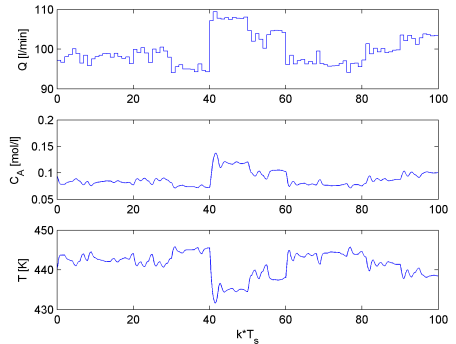
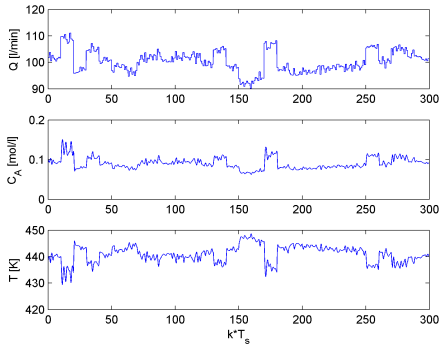
Intrare: debit Q al agentului de răcire

Ieșiri:

- Concentrația C_A a substanței A în amestec
- Temperatura T a amestecului

Matlab: Date experimentale

Stânga: identificare, Dreapta: validare



Matlab: ARX MIMO, diferit de teorie

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

$$A(q^{-1}) = \begin{bmatrix} a^{11}(q^{-1}) & a^{12}(q^{-1}) & \dots & a^{1ny}(q^{-1}) \\ a^{21}(q^{-1}) & a^{22}(q^{-1}) & \dots & a^{2ny}(q^{-1}) \\ \dots & \dots & \dots & \dots \\ a^{ny1}(q^{-1}) & a^{ny2}(q^{-1}) & \dots & a^{nyny}(q^{-1}) \end{bmatrix}$$

$$a^{ij}(q^{-1}) = \begin{cases} 1 & \text{dacă } i = j \\ 0 & \text{altfel} \end{cases} + a_1^{ij}q^{-1} + \dots + a_{na_{ij}}^{ij}q^{-na_{ij}}$$

$$B = \begin{bmatrix} b^{11}(q^{-1}) & b^{12}(q^{-1}) & \dots & b^{1nu}(q^{-1}) \\ b^{21}(q^{-1}) & b^{22}(q^{-1}) & \dots & b^{2nu}(q^{-1}) \\ \dots & \dots & \dots & \dots \\ b^{ny1}(q^{-1}) & b^{ny2}(q^{-1}) & \dots & b^{nynu}(q^{-1}) \end{bmatrix}$$

$$b^{ij}(q^{-1}) = b_1^{ij}q^{-nk_{ij}} + \dots + b_{nb_{ij}}^{ij}q^{-nk_{ij} - nb_{ij} + 1}$$

Matlab: Identificarea modelului

```
m = arx(id, [Na, Nb, Nk]);
```

Argumente funcție:

- 1 Datele de identificare.
- 2 Matrici cu ordinele polinoamelor din A , B , și întârzieri nk :

$$Na = \begin{bmatrix} na_{11} & \dots & na_{1ny} \\ \dots & & \\ na_{ny1} & \dots & na_{nyny} \end{bmatrix}$$

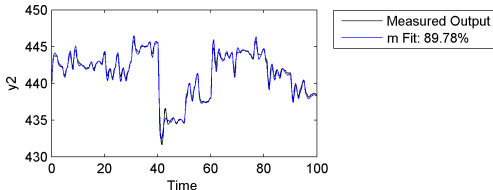
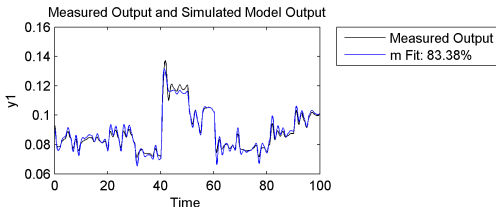
$$Nb = \begin{bmatrix} nb_{11} & \dots & nb_{1nu} \\ \dots & & \\ nb_{ny1} & \dots & nb_{nynu} \end{bmatrix}$$

$$Nk = \begin{bmatrix} nk_{11} & \dots & nk_{1nu} \\ \dots & & \\ nk_{ny1} & \dots & nk_{nynu} \end{bmatrix}$$

Matlab: Rezultate

Luăm $na = 2$, $nb = 2$, și $nk = 1$ peste tot in elementele matricilor:

```
Na = [2 2; 2 2]; Nb = [2; 2]; Nk = [1; 1];  
m = arx(id, [Na Nb Nk]);  
compare(m, val);
```



Anexă: ARX neliniar (pentru proiect)

Structura ARX neliniară

Reamintim ARX standard:

$$y(k) = -a_1 y(k-1) - a_2 y(k-2) - \dots - a_{na} y(k-na) \\ + b_1 u(k-1) + b_2 u(k-2) + \dots + b_{nb} u(k-nb) + e(k)$$

Dependență liniară de ieșirile precedente $y(k-1), \dots, y(k-na)$ și intrările precedente $u(k-1), \dots, u(k-nb)$.

ARX neliniar (NARX) generalizează la orice dependență neliniară:

$$y(k) = g(y(k-1), y(k-2), \dots, y(k-na), \\ u(k-1), u(k-2), \dots, u(k-nb); \theta) + e(k)$$

Funcția g este parametrizată de $\theta \in \mathbb{R}^n$, și acești parametri pot fi aleși pentru a recupera datele de identificare, identificând astfel un sistem dinamic neliniar.

NARX polinomial

În cazul nostru particular, fiecare coeficient constant a, b este înlocuit de către un **polinom în ieșirile și intrările precedente**:

$$y(k) = -p_1(d(k))y(k-1) - p_2(d(k))y(k-2) - \dots - p_{na}(d(k))y(k-na) \\ + z_1(d(k))u(k-1) + z_2(d(k))u(k-2) + \dots + z_{nb}(d(k))u(k-nb) \\ + e(k)$$

unde $d(k) = [y(k-1), \dots, y(k-na), u(k-1), \dots, u(k-nb)]^T$

Observații:

- A nu se confunda cu forma polinomială liniară $A(q^{-1})y(k) = B(q^{-1})u(k) + e(k)$
- Parametrii sunt acum coeficienții tuturor polinoamelor
- Dacă toate polinoamele au grad 0, obținem ARX liniar
- y și u la momente zero și negative pot fi luate 0, presupunând că sistemul este în condiții inițiale nule
- Calculând explicit expresia obținem un polinom global $g(d(k); \theta)$

Exemplu

Luăm $na = nb = 1$, atunci $d(k) = [y(k-1), u(k-1)]^T$ și modelul este:

$$y(k) = -p_1(d(k))y(k-1) + z_1(d(k))u(k-1) + e(k)$$

Luăm gradul 1 pentru toate polinoamele:

$$\begin{aligned}y(k) &= -[a_1 + a_2y(k-1) + a_3u(k-1)]y(k-1) \\ &\quad + [b_1 + b_2y(k-1) + b_3u(k-1)]u(k-1) + e(k) \\ &= -a_1y(k-1) - a_2y(k-1)^2 - a_3u(k-1)y(k-1) \\ &\quad + b_1u(k-1) + b_2y(k-1)u(k-1) + b_3u(k-1)^2 + e(k)\end{aligned}$$

Regresia liniară funcționează ca de obicei, găsește parametrii a_i , b_j care minimizează valoarea MSE!

Exemplu: Reducerea polinoamelor

Atenție însă: **regresori duplicați**, $y(k-1)u(k-1)$ vor duce la sistem singular!

⇒ grupăm duplicatele împreună:

$$\begin{aligned}y(k) &= -a_1y(k-1) - a_2y(k-1)^2 + b_1u(k-1) + b_3u(k-1)^2 \\ &\quad + cy(k-1)u(k-1) + e(k) \\ &= g(d(k); \theta) + e(k)\end{aligned}$$

unde $c = -a_3 + b_2$, și vectorul redus de parametri este:

$$\theta = [a_1, a_2, b_1, b_3, c]^T$$

Reamintim predicție versus simulare

Predicție cu un pas înainte: ieșirea reală a sistemului este cunoscută, deci vectorul de semnale întârziate $d(k)$ este disponibil:

$$x(k) = [y(k-1), \dots, y(k-na), u(k-1), \dots, u(k-nb)]^T$$
$$\hat{y}(k) = g(d(k); \hat{\theta})$$

Simulare: ieșirea reală este necunoscută, folosim ieșirile simulate anterior pentru a construi o *aproximare* a lui $d(k)$:

$$\hat{x}(k) = [\hat{y}(k-1), \dots, \hat{y}(k-na), u(k-1), \dots, u(k-nb)]^T$$
$$\hat{y}(k) = g(\hat{d}(k); \hat{\theta})$$