Identificarea Sistemelor – Laborator 10
Identificarea ARX recursivă

Organizare

Recitiţi regulile de organizare din lab 2, ele se vor aplica şi acestui laborator. Singurul lucru care se schimbă este link-ul de dropbox, care pentru acest laborator este:

`https://www.dropbox.com/request/zHPO7mqLvekya8InLi9p`

Descrierea laboratorului

Vom studia în acest laborator varianta recursivă a metodei ARX, vezi cursul *Identificarea recursivă*.

Fiecărui student i se alocă de către profesor un index pentru setul de date. Apoi, studentul descarcă fişierul Matlab ce formează baza laboratorului de pe pagina cursului. Fişierul conţine datele de identificare în variabila `id`, şi separat datele de validare în variabila `val`.

Se ştie în avans că ordinul sistemului este n, dat în variabila `n` din fişierul de date; că sistemul are o structură de tip eroare de ieşire, OE; şi că nu are timp mort. Pentru a compensa nepotrivirea cu structura ARX vom lua ordine mai mari pentru modelele ARX pe care le vom căuta. Recomandarea este să alegeţi $na = nb = 3 \cdot n$.

- Implementaţi algoritmul ARX recursiv într-o funcţie care primeşte la intrare setul de date de identificare, ordinele modelului na şi nb, vectorul iniţial de parametri $\theta(0)$, şi matricea inversă iniţială $P^{-1}(0)$. Funcţia trebuie să producă la ieşire o matrice $\Theta \in \mathbb{R}^{N\times(na+nb)}$ conţinând pe fiecare linie k vectorul de parametri $\theta(k)$: întâi coeficienţii $a_1, \dots, a_{na}$ ai polinomului A, şi apoi coeficienţii $b_1, \dots, b_{nb}$ din B (acest format este compatibil cu cel al funcţiei Matlab deja existente, aşadar rezultatele celor două funcţii vor fi mai uşor de comparat). Un pseudocod cu mai multe detalii decât cel prezentat la curs este inclus mai jos.
- Rulaţi identificarea ARX recursivă folosind funcţia dvs., pe datele de identificare, pornind de la o inversă iniţială $P^{-1}(0) = 100 I_{na+nb}$. Comparaţi *pe datele de validare* calitatea celor două modele: unul cu parametrii finali găsiţi după procesarea întregului set de date; şi altul după 10% din date. Care model este mai bun, şi de ce?
- Opţional, dacă aveţi timp, repetaţi experimentul iniţial, dar de această dată cu funcţia `rarx` deja existentă în Matlab. Comparaţi rezultatele produse de această funcţie (de ex., modelele de la 10% şi 100% din date) cu cele ale funcţiei dvs., verificând dacă sunt la fel sau similare.

Funcţii relevante din toolbox-ul de identificare a sistemelor: `rarx`, `idpoly`, `compare`. Indicii adiţionale:

- După ce aveţi polinoamele A şi B ca vectori de coeficienţi în ordinea crescătoare a puterilor lui q^{-1}, folosiţi `idpoly(A, B, [], [], [], 0, Ts)` pentru a genera modelul ARX, unde `Ts` este perioada de eşantionare. Nu uitaţi că toţi vectorii de coeficienţi din polinoame trebuie să conţină întotdeauna coeficienţii constanţi (puterea 0 a argumentului q^{-1}), care trebuie să fie 1 în A, şi 0 în B. Ţineţi cont că matricea de parametri returnată de algoritm *nu* conţine aceşti coeficienţi constanţi.

inițializează $\widehat{\theta}(0)$ (vector coloană $na + nb$), $P^{-1}(0)$ (matrice $(na + nb) \times (na + nb)$)
loop la fiecare pas $k = 1, 2, \dots$
  extrage $u(k)$, $y(k)$
  formează vectorul de regresori ARX:
  $$\varphi(k) = [-y(k-1), \cdots, -y(k-na), u(k-1), \cdots, u(k-nb)]^\top$$
  găsește eroarea de predicție $\varepsilon(k) = y(k) - \varphi^\top(k)\widehat{\theta}(k-1)$ (un scalar)
  actualizează inversa: $P^{-1}(k) = P^{-1}(k-1) - \frac{P^{-1}(k-1)\varphi(k)\varphi^\top(k)P^{-1}(k-1)}{1+\varphi^\top(k)P^{-1}(k-1)\varphi(k)}$
  calculează ponderile: $W(k) = P^{-1}(k)\varphi(k)$ (vector coloană $(na + nb)$)
  actualizează parametrii: $\widehat{\theta}(k) = \widehat{\theta}(k-1) + W(k)\varepsilon(k)$
end loop

- Funcția existentă `rarx` preia la intrare setul de date de identificare, ordinele modelului na și nb și întârzierea nk sub formă de vector, argumentele `'ff', 1` care configurează algoritmul la fel cu cel din curs, vectorul inițial de parametri $\theta(0)$, și matricea inversă inițială $P^{-1}(0)$. Matricea numită P în documentația funcției Matlab `rarx` este de fapt matricea *inversă* P^{-1} din curs, fiți așadar atenți când o alegeți. Funcția produce la ieșire o matrice Θ în același format cu cel explicat mai sus.