Identificarea Sistemelor – Laborator 10
Identificarea recursivă

Organizare

- Acest laborator se rezolvă independent de către fiecare student. Doar dacă există mai mulţi studenţi decât calculatoare, studenţii se pot grupa câte doi la un calculator.
- Soluţia constă din cod Matlab. Codul va fi verificat şi rulat de către profesor, în timpul laboratorului, şi prezenţa la laborator va fi validată doar dacă aveţi o soluţie funcţională şi originală. Prezenţele la toate laboratoarele trebuiesc validate pentru eligibilitate la examen. Cum cel mult două laboratoare se pot recupera la sfârşitul semestrului, acumularea a trei sau mai multe laboratoare lipsă la orice moment de timp duce automat la pierderea finală a eligibilităţii.
- Discutarea ideilor între studenţi este permisă şi chiar de dorit, dar copierea sau schimbul direct de cod este interzis. Încălcarea acestei reguli va duce la invalidarea soluţiei.

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:
`http://busoniu.net/teaching/sysid2018/index_ro.html`

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.

În prima parte a laboratorului, obiectivul dvs. este să 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 implementate, aşadar rezultatele celor două funcţii vor fi mai uşor de comparat).

Cum sistemul este de tip OE, pentru a compensa nepotrivirea cu structura ARX vom lua ordine mai mari pentru modelele ARX pe care le vom căuta. Recomandarea este să alegeţi $na = nb = 3 \cdot n$. Folosind aceste ordine, pentru a doua parte a laboratorului:

- Rulaţi identificarea ARX recursivă folosind funcţia dvs., pe datele de identificare, pornind de la o inversă iniţială $P^{-1}(0) = \frac{1}{\delta} I_{na+nb} = 100 I_{na+nb}$ (deci $\delta = 0.01$). 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. Explicaţi diferenţele.
- Repetaţi experimentul, dar de această dată cu $P^{-1}(0) = \frac{1}{\delta} I_{na+nb} = 0.01 I_{na+nb}$ (so $\delta = 100$). Gândiţi-vă la rezultate. Pentru care valoare a lui δ este mai rău modelul iniţial, şi de ce?
- 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:

- 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.
- Matricea numită P în documentaţia funcţiei `rarx` este de fapt matricea *inversă* P^{-1} din curs, fiţi aşadar atenţi când o alegeţi. Nu uitaţi să configuraţi algoritmul în `rarx` folosind argumentele `'ff', 1`.
- `rarx` returnează o matrice de parametri cu exact acelaşi format cerut mai sus pentru funcţia dvs.