Identificarea sistemelor – Laborator 8
Identificarea modelelor de tip Output Error folosind metoda Gauss-Newton

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. Dezvoltaţi acest cod într-un singur script 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 duce la pierderea 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

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/sysid2019/index_ro.html`

Fişierul conţine datele de identificare în variabila `id`, şi separat datele de validare în variabila `val`. Ambele variabile sunt obiecte de tip `iddata` din toolbox-ul Matlab de identificare a sistemelor, vezi `doc iddata`.

Se ştie în avans că sistemul este de ordinul 1, fără timp mort, şi este afectat doar de zgomot de măsurare $e(k)$ pe ieşire. Ca atare, următoarea formă de tip Output Error este potrivită pentru modelarea acestui sistem:

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

cu parametrii $\theta = [f, b]^\top$. Obiectivul nostru va fi implementarea metodei erorii de predicţie pentru această structură particulară de model, folosind metoda de optimizare Gauss-Newton. Algoritmul este rezumat pe pagina următoare, într-un mod mai direct decât felul în care a fost explicat în curs, pentru a ajuta cu implementarea.

Cerinţe:

- Calculaţi formulele recursive necesare în algoritm, pe hârtie sau la tablă. Indiciu: vezi cazul ARMAX de ordinul 1 exemplificat în curs.
- Implementaţi algoritmul şi rulaţi-l pe datele de identificare, pornind de la valori *nenule* ale parametrilor iniţiali.
- Pentru valorile (aproape) optime ale f şi b obţinute, creaţi un model de tip OE în formatul toolboxului de identificiare, folosind `idpoly`. De notat că sintaxa funcţiei este `idpoly(A,B,C,D,F,0,Ts)`

unde trebuie specificat zeroul iniţial în B, constanta 1 iniţială în F, şi perioada de eşantionare se poate găsi de ex. în setul de date de identificare. Folosiţi `compare` pentru a verifica performanţa modelului pe datele de validare.

- Dacă modelul nu este suficient de bun, acordaţi α, δ şi $\ell_{\max}$ (eventual împreună cu θ_0), pentru a îmbunătăţi performanţa.

---

alege pasul α, parametrii iniţiali θ_0, pragul de convergenţă δ, şi numărul maxim de iteraţii $\ell_{\max}$
iniţializează counterul de iteraţii $\ell = 0$
calculează formulele recursive pentru $\varepsilon(k)$, $\frac{d\varepsilon(k)}{d\theta} = [\frac{d\varepsilon(k)}{df}, \frac{d\varepsilon(k)}{db}]^\top$
repeat
    cu parametrii curenţi θ_ℓ:
        simulează (aplică formulele recursive) pentru a calcula $\varepsilon(k)$, $\frac{d\varepsilon(k)}{d\theta}$, pentru $k = 1, \dots, N$
        calculează gradientul funcţiei obiectiv cu $\frac{dV}{d\theta} = \frac{2}{N} \sum_{k=1}^{N} \varepsilon(k) \frac{d\varepsilon(k)}{d\theta}$
        calculează Hessianul aproximat al funcţiei obiectiv, cu $\mathcal{H} = \frac{2}{N} \sum_{k=1}^{N} \frac{d\varepsilon(k)}{d\theta} \left[\frac{d\varepsilon(k)}{d\theta}\right]^\top$
    aplică formula de actualizare Gauss-Newton: $\theta_{\ell+1} = \theta_\ell - \alpha \mathcal{H}^{-1} \frac{dV}{d\theta}$
    incrementează counterul: $\ell = \ell + 1$
until $\|\theta_\ell - \theta_{\ell-1}\| \leq \delta$, sau $\ell_{\max}$ a fost atins

---