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

Organizare – ca şi până acum, vezi celelalte laboratoare.

Veţi dezvolta o funcţie cu următoarea semnătură:

```
[index, e1, de1, theta] = oeidentify
```

Fiecărui student i se alocă de către profesor un index în intervalul 1-8, şi acesta trebuie salvat în variabila `index` la începutul funcţiei. Indexul dictează care fişier de date trebuie încărcat. De exemplu, dacă aveţi indexul 3, trebuie să încărcaţi fişierul `lab8_3.mat`. Toate aceste fişiere de date sunt deja accesibile din codul funcţiei dvs. Fiecare fişier conţine datele de identificare în variabila `id`, şi datele de validare în variabila `val`. Indiciu: pentru a creşte viteza de execuţie şi pentru a evita timeouturile, salvaţi vectorii de intrări şi ieşiri în arrayuri separate în loc de a accesa tot timpul obiectul `id`.

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, şi cu indicii adiţionale pentru a ajuta cu implementarea. Triunghiurile semnalează comentarii.

Cerinţe:

- Calculaţi pe hârtie formulele recursive pentru $\varepsilon(k)$, $\frac{d\varepsilon(k)}{d\theta} = [\frac{d\varepsilon(k)}{df}, \frac{d\varepsilon(k)}{db}]^\top$. Indiciu: vezi cazul ARMAX de ordinul 1 exemplificat în curs.

- Pentru valorile parametrilor $f = b = 1$, aplicaţi formulele obţinute pentru calculul semnalelor $\varepsilon(k)$, $d\varepsilon(k)$ (liniile 4 şi 6 din pseudocod). Returnaţi secvenţele rezultate în `e1`, `de1`, unde `de1` este o matrice cu 2 linii şi N coloane, în care coloana k conţine $d\varepsilon(k)$. Observaţie: Grader verifică doar primele 10 elemente ale secvenţelor. Indicii: Nu uitaţi că ε este scalar şi $d\varepsilon$ vector coloană de 2 elemente. Poate fi mai uşor să nu reprezentaţi explicit semnalele la momentul 0, ci să implementaţi un caz special pentru actualizările de la pasul $k = 1$. Pseudocodul deja funcţionează în acest fel.

- Folosind formulele recursive de mai sus la fiecare iteraţie, implementaţi algoritmul OE şi rulaţi-l pe datele identificare. Configuraţi algoritmul după cum urmează: $\theta_1 = [f_1, b_1]^\top = [1, 1]^\top$, $\alpha = 0.5$, $\ell_{\max} = 100$, $\delta = 10^{-4}$. Returnaţi întreaga secvenţă de vectori de parametri calculaţi în `theta`, este o matrice cu 2 linii şi N coloane, în care coloana ℓ conţine θ_ℓ. Observaţie: Grader verifică primii 5 vectori de parametri, şi ultimul (după convergenţă). Indicii: Asiguraţi-vă că înţelegeţi structura variabilelor $\frac{dV}{d\theta}$ şi $\mathcal{H}$ înainte de a programa. Folosiţi inversa reală a matricii (`inv`, nu backslash) pentru a implementa actualizarea.

- 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. Observaţie: Grader verifică dacă aţi validat, dar nu şi ieşirile modelului (verificarea valorilor lui θ asigură deja calitatea modelului).

- Dacă mai aveţi timp, acordaţi α, δ şi $\ell_{\max}$ (eventual împreună cu θ_1), pentru a îmbunătăţi performanţa.

1: alege pasul α, parametrii iniţiali θ_1, pragul de convergenţă δ, şi numărul maxim de iteraţii $\ell_{\max}$
2: iniţializează indexul de iteraţie $\ell = 1$
3: **repeat**
$\triangleright$ liniile 4-9 rulează cu valorile curente ale parametrilor, θ_l
4: calculează direct $\varepsilon(1), d\varepsilon(1)$, folosind $\varepsilon(0) = 0$, $d\varepsilon(0) = [0, 0]^\top$, $y(0) = 0$, $u(0) = 0$
5: **for** $k = 2, \ldots, N$ **do** $\triangleright$ atenţie la indexul de start
6: aplică formulele recursive pentru a calcula $\varepsilon(k)$, $\frac{d\varepsilon(k)}{d\theta}$ $\triangleright$ $\frac{d\varepsilon(k)}{d\theta}$ trebuie să fie un vector 2x1
7: **end for**
$\triangleright$ $\frac{dV}{d\theta}$ este vector 2x1, $\mathcal{H}$ şi termenii sumei sale sunt matrici 2x2
8: calculează gradientul funcţiei obiectiv cu $\frac{dV}{d\theta} = \frac{2}{N} \sum_{k=1}^{N} \varepsilon(k) \frac{d\varepsilon(k)}{d\theta}$
9: calculează Hessianul aproximat al funcţiei obiectiv, cu $\mathcal{H} = \frac{2}{N} \sum_{k=1}^{N} \frac{d\varepsilon(k)}{d\theta} \left[\frac{d\varepsilon(k)}{d\theta}\right]^\top$
10: aplică formula de actualizare Gauss-Newton: $\theta_{\ell+1} = \theta_\ell - \alpha \mathcal{H}^{-1} \frac{dV}{d\theta}$ $\triangleright$ folosiţi `inv`, nu \
11: incrementează counterul: $\ell = \ell + 1$
12: **until** $\|\theta_\ell - \theta_{\ell-1}\| \leq \delta$, sau $\ell > \ell_{\max}$
