Identificarea sistemelor – Laborator 8
Identificarea unui model Output-Error pentru motorul DC cu optimizare Gauss-Newton

Organizare

Recitiți partea de logistică din laboratorul 2, aceleași reguli se vor aplica și pentru acest laborator. Singurele lucruri care se schimbă sunt assignment-ul pe Teams, care pentru acest laborator este "Lab 8 (OE)", și desigur numărul laboratorului în numele fișierului.

Descrierea laboratorului

În acest laborator vom identifica un model de tip eroare de ieșire a motorului DC folosind metoda minimizării erorii de predicție.

Din fizica sistemului știm că este de ordinul 1 (de la voltaj la viteza unghiulară), și datorită unor particularități ale comunicației seriale, este posibil ca sistemul să aibă întârzieri. Ca atare, următoarea formă de tip Output Error este potrivită pentru modelarea acestui sistem:

$$y(k) = \frac{b}{1 + fq^{-1}} u(k - nk) + e(k)$$

cu parametrii $\theta = [f, b]^\top$. Obiectivul nostru va fi implementarea metodei erorii de predicție pentru această structură particulară de model, folosind metoda de optimizare Gauss-Newton. Algoritmul este rezumat pe pagina următoare, pentru a ajuta cu implementarea. Spre deosebire de explicația de la curs, aici va trebui să acordăm atenție întârzierii nk; de exemplu, fiindcă eroarea și derivata acesteia vor depinde de $u(k - nk)$, vom începe actualizările recursive la pasul $k = nk + 1$.

Cerințe:

- Calculați formulele recursive necesare în algoritm, pe hârtie sau la tablă. **Indiciu:** Impuneți $\varepsilon(k) = e(k)$, găsiți $\varepsilon(k)$ folosind ecuația modelului, și apoi calculați derivatele parțiale ale lui $\varepsilon(k)$ în raport cu cei doi parametri.
- Pentru a simplifica lucrurile, vom crea o singură secvență de date mai lungă care va conține atât datele de identificare, cât și cele de validare. Vom utiliza o perioadă de eșantionare de 0.01 s(10 ms). După un interval scurt de intrări zero, vom aplica un semnal de intrare de tip SPAB cu o lungime de aproximativ 200 de eșantioane și cu valori între -0.7 și 0.7, urmat de un alt interval de intrări zero și apoi de un semnal treaptă cu magnitudinea de aproximativ 0.4 și o lungime de aproximativ 70 eșantioane. Pentru a genera semnalul SPAB, folosiți fie `idinput`, fie codul dvs. de la laboratoarele anterioare.
- Aplicați semnalul de intrare generat sistemului. Ieșirea este viteza de rotație. Izolați subsecvența corespunzătoare intrărilor SPAB și copiați-o în noi vectori de intrare și ieșire; acestea vor fi datele noastre de identificare. **Observație importantă**: pentru a minimiza uzura sistemului, separați codul de generare a datelor de cel care efectuează restul pașilor de mai jos (cel mai simplu folosind secțiuni diferite, vezi *Code Sections* în documentația Matlab), și regenerați datele doar când este necesar.

- Reprezentați grafic și examinați datele obținute.
- Implementați algoritmul și rulați-l pe datele de identificare, pornind de la valori *nenule* ale parametrilor inițiali, și acordând nk.
- Pentru valorile (aproape) optime ale f și b obținute, creați un model de tip OE în formatul toolboxului de identificare, folosind `idpoly`. De notat că sintaxa funcției este `idpoly(A,B,C,D,F,0,Ts)` unde trebuie specificate nk zerouri inițiale în B, constanta 1 inițială în F, și perioada de eșantionare. Polinoamele pe care nu le folosiți pot fi egale cu 1. Folosiți `compare` pentru a verifica performanța modelului pe datele de validare.
- Dacă modelul nu este suficient de bun, acordați α, δ și $\ell_{\max}$ (eventual împreună cu θ_0), pentru a îmbunătăți performanța.

---

alege pasul α, parametrii inițiali θ_0, pragul de convergență δ, și numărul maxim de iterații $\ell_{\max}$
inițializează counterul de iterații $\ell = 0$
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 curenți θ_ℓ:
  inițializează $\varepsilon(k) = y(k)$, $\frac{d\varepsilon(k)}{d\theta} = [0,0]^\top$ for $k = 1, \ldots, nk$
  for $k = nk+1$ la N **do**
    aplică formulele recursive pentru calculul $\varepsilon(k)$, $\frac{d\varepsilon(k)}{d\theta}$
  end for
  calculează gradientul funcției obiectiv cu $\frac{dV}{d\theta} = \frac{2}{N-nk}\sum_{k=1}^{N}\varepsilon(k)\frac{d\varepsilon(k)}{d\theta}$
  calculează Hessianul aproximat al funcției obiectiv, cu $\mathcal{H} = \frac{2}{N-nk}\sum_{k=1}^{N}\frac{d\varepsilon(k)}{d\theta}\left[\frac{d\varepsilon(k)}{d\theta}\right]^\top$
  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

---