

Identificarea sistemelor – Laborator 8

Minimizarea erorii de predicție

Organizare

- Acest laborator este parte obligatorie a cursului de Identificarea Sistemelor. Laboratorul se rezolvă independent de către fiecare student.
- Soluția constă din cod Matlab. Codul va fi verificat și rulat de către profesor pentru a vă lua în considerare prezența la laborator. Vom efectua această verificare pe cât posibil în timpul laboratorului, împreună cu dvs. Scrieți în orice caz codul de o manieră clară, adăugând comentarii acolo unde este necesar, pentru a-l face inteligibil și în lipsa explicațiilor verbale. La sfârșitul laboratorului trimiteți codul profesorului prin email (Zoltán Nagy la zoltan.nagy@aut.utcluj.ro, sau Marius Costandin la marius.costandin@aut.utcluj.ro) sub forma unui fișier .m sau într-o arhivă ZIP, folosind următorul format pentru numele de fișier:
`is_labN_indexINDEX_NUME`
unde N este numărul laboratorului, INDEX este indexul setului de date, vezi mai jos; și NUME este numele dvs. de familie. Vă rugăm să *includeți acest nume de fișier și în subiectul emailului*.
- Discutarea ideilor între studenți este permisă și chiar de dorit, dar copierea sau schimbul direct de cod este interzis. Încălcarea acestei reguli va duce la invalidarea soluției.

Descrierea laboratorului

În acest laborator vom identifica și compara modele de tip OE, ARMAX, și ARX, folosind metoda minimizării erorii de predicție. Vezi materialul de curs, Partea VII: *Metoda minimizării erorii de predicție, ARMAX, și OE*.

Fiecărui student i se alocă de către profesor un index pentru setul de date. Apoi, studentul descarcă fișierele Matlab ce formează baza laboratorului de pe pagina cursului:

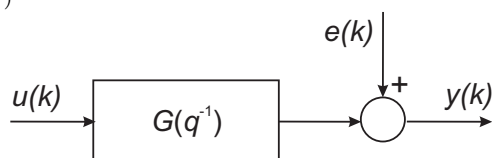
http://busoniu.net/teaching/sysid2017/index_ro.html

Există un fișier de date care conține datele de identificare în variabila `id`, și 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`. Este de asemenea furnizată funcția `oeidentify`.

Se știe în avans că sistemul este în forma eroare de ieșire (output error, OE):

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

vezi și figura, unde $G = \frac{B(q^{-1})}{F(q^{-1})}$.



Așadar, sistemul este afectat doar de zgomot alb de măsurare pe ieșire. Se știe de asemenea că sistemul nu are timp mort. **Ordinul n al sistemului este dat în variabila Matlab `n` din fișierul de date.**

Prima parte a laboratorului va consta din calculul erorii de predicție pentru o astfel de structură OE. Reamintim de la curs că formula generală pentru calculul erorii de predicție este:

$$\varepsilon(k) = H^{-1}(q^{-1})(y(k) - G(q^{-1})u(k))$$

Pentru structura OE, $G(q^{-1}) = \frac{B(q^{-1})}{F(q^{-1})}$ și $H(q^{-1}) = 1$. Ținând cont de aceste lucruri, scrieți o funcție cu următorul format:

$$V = \text{prederr}(\text{theta}, \text{id}, \text{nb}, \text{nf})$$

Această funcție trebuie să calculeze eroarea de predicție $\varepsilon(k)$ pentru $k = 1, \dots, N$, și să returneze în V eroarea medie pătratică, $V = \frac{1}{N} \sum_{k=1}^N \varepsilon^2(k)$. Calculul se face pentru un vector dat de parametri, $\theta = [f_1, \dots, f_{nf}, b_1, \dots, b_{nb}]^T$, pe orice set de date id , și orice ordine nb și nf ale modelului. Indicii: scrieți întâi formula pentru eroarea de predicție pe hârtie; eroarea de predicție va fi rezultatul simulării unui sistem dinamic. Nu trebuie să denumiți funcția și variabilele cu numele indicate, dar argumentele de intrare și ieșire trebuie să aibă semnificațiile de mai sus.

Odată ce ați scris și testat funcția, o puteți folosi cu funcția deja furnizată:

$$\text{oem} = \text{oeidentify}(\text{'prederr'}, \text{id}, \text{nb}, \text{nf})$$

care va găsi un vector de parametri ce minimizează (aproximativ) eroarea medie pătratică, și returnează direct un model de tip OE numit `oem` cu parametrii găsiți. Înlocuiți aici 'prederr' cu numele funcției dvs.

În a doua parte a laboratorului, vom folosi codul de mai sus pentru a identifica un model OE și a-l compara cu ARMAX și ARX.

- Dacă fiind structura cunoscută a sistemului, alegeți valori potrivite pentru nf și nb și apelați funcția de optimizare cu aceste valori pentru a găsi un model de tip OE. Validați acest model pe datele de validare.
- Repetați experimentul dar acum cu funcția standard Matlab `oe`. Comparați cele două modele, ținând cont că algoritmi de optimizare folosiți de `oe` și `oeidentify` sunt diferiți și pot produce așadar rezultate diferite. Dacă se întâmplă așa, folosiți la pașii următori modelul mai bun dintre cele două.
- Alegeți ordinele $\text{na}, \text{nb}, \text{nc}$ pentru un model ARX și unul ARMAX, și apelați funcțiile standard `arx` și `armax` pentru a găsi modelele. Nu folosiți `arxstruc` sau alte funcții de selecție automată a ordinului sistemului, selectați în schimb ordinele manual bazându-vă pe experiența de la laboratoarele precedente și pe cunoștințele indicate mai sus despre sistem.
- Comparați modelele OE, ARMAX, și ARX. Se supune sistemul și formei ARMAX? Dar celei ARX? Comparați scorurile celor trei modele date de funcția `compare`. Dacă două scoruri sunt similare, luați în considerare și complexitatea modelului. Care este cel mai bun model, când se ține cont de toate elementele incluzând calitatea ieșirii, complexitatea modelului, și structura sistemului?

Funcții relevante din toolbox-ul de identificare a sistemelor: `arx`, `armax`, `oe`, `plot`, `compare`. Dacă aveți un model numit de ex. `mod`, puteți scrie `mod` în linia de comandă pentru a-i investiga structura, iar polinoamele sale componente sunt accesibile via câmpuri cu numele adecvat, de ex. `mod.A`, `mod.B`. Vezi și `doc ident` pentru documentația completă a toolbox-ului.