Identificarea Sistemelor – Laborator 9
Metoda variabilelor instrumentale

Organizare

Recitiţi regulile de organizare din lab 2, ele se vor aplica şi acestui laborator. Singurul lucru care se schimbă este link-ul de dropbox, care pentru acest laborator este:
`https://www.dropbox.com/request/GQ5FCQGHENxm3LnCdET3`

Descrierea laboratorului

În acest laborator, vom implementa metoda variabilelor instrumentale (VI) folosind instrumente bazate pe ieşirile unui model ARX, şi o vom aplica pentru identificarea motorului de curent continuu. Ştim că motorul are întârzieri, deci vom identifica modele VI cu ordinele na, nb şi cu întârzierea nk.

Pentru a rezolva problema de identificare eficient în Matlab, vom folosi următoarea formă a sistemului de ecuaţii din metoda VI, potrivită pentru împărţirea la stânga matriceală (operatorul $\backslash$):

$$\left[\frac{1}{N}\sum_{k=1}^{N} Z(k)\varphi^T(k)\right]\theta = \frac{1}{N}\sum_{k=1}^{N} Z(k)y(k)$$

sau echivalent: $\tilde{\Phi}\theta = \tilde{Y}$

unde $\tilde{\Phi} = \frac{1}{N}\sum_{k=1}^{N} Z(k)\varphi^T(k)$ este o matrice $(na + nb) \times (na + nb)$ şi $\tilde{Y} = \frac{1}{N}\sum_{k=1}^{N} Z(k)y(k)$ este un vector $(na+nb) \times 1$. De notat tildele, care înseamnă că aceste elemente sunt variante ale regresorilor şi ieşirilor "modificate" de către VI.

Fiindcă avem întârzieri, în ecuaţia de mai sus vectorul de regresori este:

$$\varphi(k) = [-y(k-1), \ldots, -y(k-na), u(k-nk), \ldots, u(k-nb-nk+1)]^T$$

iar vectorul de variabile instrumentale este:

$$Z(k) = [-\hat{y}(k-1), \ldots, -\hat{y}(k-na), u(k-nk), \ldots, u(k-nb-nk+1)]^T$$

unde ieşirile $\hat{y}$ sunt cele simulate cu modelul ARX. De notat că nu putem folosi predicţii fiindcă acestea depind de ieşirile reale şi sunt aşadar corelate cu zgomotul! Reamintim de asemenea că $\theta = [a_1, \ldots, a_{na}, b_1, \ldots, b_{nb}]^T$.

Fiecare student va obţine un set de date folosind motorul de curent continuu şi va identifica sistemul, conform instrucţiunilor următoare.

1. Pentru a simplifica lucrurile, vom crea o singura secvenţă de date mai lungă care va conţine atât datele de identificare, cât şi cele de validare. Vom utiliza o perioadă de eşantionare 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 300 de eşantioane şi cu valori între -0.8 şi 0.8, urmat de un alt interval de intrări zero şi apoi de un semnal treaptă cu magnitudinea de aproximativ 0.3 şi o lungime de aproximativ 70 eşantioane. Pentru a genera semnalul SPAB, folosiţi fie `idinput`, fie codul dvs. de la laboratoarele anterioare, dar în acest al doilea caz folosiţi suficienţi biţi pentru ca semnalul să nu se repete.

2. Aplicaţi semnalul de intrare generat sistemului. Ieşirea este viteza de rotaţie. Izolaţi subsecvenţa corespunzătoare intrărilor SPAB şi copiaţi-o în noi vectori de intrare şi ieşire; acestea vor fi datele noastre de identificare. **Observaţie importantă**: pentru a minimiza uzura sistemului, separaţi codul de generare a datelor de cel care efectuează restul paşilor de mai jos (cel mai simplu folosind secţiuni diferite, vezi *Code Sections* în documentaţia Matlab), şi regeneraţi datele doar când este necesar.

3. Reprezentaţi grafic şi examinaţi datele obţinute.

4. Identificaţi un model ARX cu ordinele $na = nb$ şi întârzierea nk. Pentru acest pas se recomandă folosirea funcţiei Matlab `arx`, pentru care va trebui să creaţi un set de date standard de identificare folosind `iddata`. Puteţi folosi în loc de aceasta şi codul dvs. de la laboratorul de ARX, dar trebuie adaptat pentru a permite folosirea întârzierii nk şi riscaţi să nu puteţi termina laboratorul la timp.

5. Aplicaţi metoda VI cu aceleaşi ordine ca şi în modelul ARX, folosind ecuaţiile de mai sus şi ieşirile modelului ARX ca variabile instrumentale. Indicii: (i) Dacă aţi utilizat `arx`, folosind codul `arxmodel.A` şi `arxmodel.B` puteţi accesa polinoamele A şi B pentru iniţializarea polinoamelor D şi C care generează VI. Dacă nu, probabil va fi mai simplu să simulaţi modelul ARX şi să completaţi Z direct cu valorile $\hat{y}$ şi u. (ii) Construiţi $\tilde{\Phi}$ şi $\tilde{Y}$ eficient, prin adunarea de termeni calculaţi matriceal în Matlab. (iii) Nu uitaţi să completaţi cu zerouri poziţiile din φ şi Z corespunzătoare paşilor negativi şi zero.

6. Verificaţi performanţa modelului VI (în simulare). Indiciu: Puteţi folosi `idpoly(A,B,C,D,F,0,Ts)` pentru a crea un model din parametrii calculaţi, după care puteţi simula folosind `compare` pe datele de validare în format `iddata`. În `idpoly`, trebuie specificate nk zerouri iniţiale în B, constanta 1 iniţială în A, şi perioada de eşantionare. Polinoamele pe care nu le folosiţi pot fi egale cu 1.

7. Acordaţi $na = nb$ şi nk pentru a obţine performanţe bune. Indicii: Folosirea unui model de ordinul 2 va fi de ajutor, chiar dacă sistemul este de ordinul 1. Ţineţi cont şi că întârzierea se poate schimba de la o rulare la alta.