Identificarea Sistemelor – Laborator 9
Metoda variabilelor instrumentale

Organizare – ca şi până acum, vezi celelalte laboratoare.

Veţi dezvolta o funcţie cu următoarea semnătură:

```
[index, Z10, phi10, theta] = ividentify
```

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 `lab9_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 ordinul sistemului este cel dat în variabila `n` din fişierul de date, şi că perturbaţia nu este zgomot alb, ci este colorată. Pentru toate modelele de mai jos vom alege aşadar $na = nb =$`n`.

Sarcina dvs. este să implementaţi algoritmul de identificare cu variabile instrumentale egale cu ieşirile unui model ARX (vezi mai jos pentru detalii). Pentru a rezolva problema de identificare eficient în Matlab, va fi util să rescriem sistemul de ecuaţii din metoda VI într-o formă potrivită pentru împărţirea la stânga matriceală (operatorul \). În acest scop, pornim de la ecuaţia (8.3) din materialul de curs şi o scriem sub forma:

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

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

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.

În formula de mai sus, $Z(k)$ este vectorul de variabile instrumentale:

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

unde ieşirile $\hat{y}$ sunt cele **simulate** cu modelul ARX găsit anterior. De notat că nu putem folosi predicţii fiindcă acestea depind de ieşirile reale şi sunt aşadar corelate cu zgomotul!

Cerinţe:

- Folosind funcţia Matlab `arx`, identificaţi un model ARX cu ordinele $na = nb =$`n` şi studaţi-i calitatea pe setul de date de validare.
- Implementaţi algoritmul VI descris mai sus, pentru a găsi un model cu aceleaşi valori na, nb. Ca un pas intermediar de verificare, returnaţi $Z(10)$ şi $\varphi(10)$ în variabilele `Z10` şi `phi10`. În final, returnaţi vectorul de parametri θ obţinut în `theta`.
- Creaţi un model VI în formatul `idpoly` folosind θ, şi comparaţi calitatea acestui model cu cea a modelului ARX iniţial, pe setul de date de validare, folosind `compare`.

Indicii:

- Pentru simplitate, completaţi vectorii Z de VI direct din ieşirile simulate ARX $\hat{y}$, fără a defini polinoame C şi D. Puteţi folosi funcţii Matlab pentru a calcula $\hat{y}$.
- Asiguraţi-vă permanent că lucraţi cu vectori şi matrici de dimensiunile corecte. Mai exact, Z, φ, şi $\tilde{Y}$ trebuie să fie vectori de $(na + nb) \times 1$ elemente; iar $\tilde{\Phi}$ şi fiecare din termenii sumei cu care este calculată, $Z(k)\varphi^T(k)$, trebuie să fie matrici de $(na + nb) \times (na + nb)$ elemente.
- Rezolvaţi ecuaţia matriceală folosind $\backslash$.
- După ce aveţi polinoamele A şi B ca vectori de coeficienţi în ordinea crescătoare a puterilor lui q^{-1}, folosiţi `idpoly(A, B, [], [], [], 0, Ts)` pentru a genera modelul VI, unde `Ts` este perioada de eşantionare. Nu uitaţi că toţi vectorii de coeficienţi din polinoame trebuie să conţină întotdeauna coeficienţii constanţi (pentru puterea 0 a argumentului q^{-1}), care trebuie să fie 1 în A şi 0 în B.
- Puteţi compara simultan calitatea mai multor modele cu sintaxa: `compare(val, model1, model2, ...)`.