Identificarea sistemelor – Laborator 6
Metoda ARX

Organizare – ca şi până acum, vezi celelalte laboratoare.

La acest laborator vom identifica modele ARX (autoregresive cu intrare exogenă) folosind regresia liniară, vezi materialul de curs, Partea V: *Metoda ARX*.

Veţi dezvolta o funcţie cu următoarea semnătură:

```
[index, PHI, theta, ypred, ysim] = findarx
```

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 `lab6_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`.

Se ştie de la operatorul sistemului că sistemul nu are timp mort (întârziere) şi are ordin cel mult 3.

Cerinţe:

- Reprezentaţi grafic şi examinaţi datele furnizate.
- Implementaţi explicit identificarea ARX folosind regresia liniară, urmărind descrierea din curs. Reamintim că regresorii sunt $-y(k-1), \dots, -y(k-na), u(k-1), \dots, u(k-nb)$. De preferat, codul dvs. trebuie să funcţioneze pentru orice valori na şi nb. Folosiţi ordinea indicată pentru regresori (parametrii vor fi aşadar în ordinea $a_1, \dots, a_{na}, b_1, \dots, b_{nb}$).
- Pentru a permite perturbaţii care nu satisfac forma ARX, vom lua $na = nb = 9$, de trei ori mai mare decât cel mai mare ordin posibil. Identificaţi un model ARX cu aceste ordine. Returnaţi matricea de regresori în `PHI` şi vectorul de parametri rezultat în `theta`. (Validarea lui `PHI` vă ajută să confirmaţi că sistemul de ecuaţii este construit corect.) Afişaţi variabila `theta` înainte de a o returna, după ce aplicaţi instrucţiunea `format long`.
- Implementaţi predicţia şi simularea modelului calculat cu intrările de validare. Ţineţi cont că pentru simulare, nu se cunosc ieşirile reale ale sistemului, deci se pot folosi doar ieşirile anterioare ale modelului însuşi; mai exact $y(k-i)$ în formula modelului trebuie înlocuit cu valoarea sa simulată anterior $\hat{y}(k-i)$, pentru $i = 1, \dots, na$. Comparaţi ieşirile prezise şi simulate cu cele reale ale sistemului, atât grafic cât şi prin calculul MSEului. Returnaţi ieşirile prezise şi de simulare respectiv în `ypred`, `ysim`.
- Opţional, dacă mai aveţi timp – sau dacă aveţi greşeli în cod şi doriţi să comparaţie cu o soluţie garantat bună – identificaţi modele şi cu funcţia Matlab `arx`, pentru aceleaşi valori na, nb ca şi în codul dvs. (cu $nk = 1$ fiindcă ştim deja că sistemul nu are întârzieri). Verificaţi că obţineţi rezultate similare.

Asiguraţi-vă că funcţia dvs. produce graficele cerute, chiar dacă acestea nu sunt validate în mod explicit de către Grader (le vom folosi oricum pentru a evalua soluţiile!).

Important: Semnalele la momente negative sau zero de timp trebuie luate egale cu 0. **Indicii:** Rulaţi predicţia şi simularea pe datele de validare iterativ, pentru fiecare pas de timp. Pentru a reduce probabilitatea de time-out-uri, salvaţi vectorii din obiecte şi structuri (de ex. `id.y`) în array-uri (de ex. `yid`) şi manipulaţi în cod aceste array-uri, Matlab-ul este mult mai rapid în acest mod.

Funcţii relevante din toolbox-ul de identificare a sistemelor: `arx`, `plot`, `compare`. Când o funcţie din toolbox-ul de identificare are acelaşi nume cu o funcţie din alt toolbox – de exemplu `compare`, care suprascrie implementarea din toolbox-ul de MPC – scrieţi de ex. `doc ident/compare` pentru a obţine documentaţia variantei din `ident`. Vezi şi `doc ident` pentru documentaţia completă a toolbox-ului.