Identificarea Sistemelor – Laborator 10
Identificarea recursivă

Organizare: Aceeaşi ca şi la laboratoarele anterioare.

Vom studia în acest laborator varianta recursivă a metodei ARX, vezi cursul *Identificarea recursivă*. Veţi dezvolta o funcţie cu următoarea semnătură:

`[index, Pinv10, theta50, thetaN] = rarxidentify`

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 `lab10_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; că sistemul are o structură de tip eroare de ieşire, OE; şi că nu are timp mort. Pentru a compensa nepotrivirea cu structura ARX vom lua ordine mai mari pentru modelele ARX pe care le vom căuta: $na = nb = 3 \cdot \texttt{n}$.

- Implementaţi algoritmul ARX recursiv, folosind pseudocodul de mai jos care conţine informaţii în plus faţă de curs. Rulaţi algoritmul pe datele de identificare, pornind de la o inversă iniţială $P^{-1}(0) = \frac{1}{\delta} I_{na+nb}$ cu $\delta = 0.01$ şi de la un vector de parametri iniţial $\theta(0)$ identic egal cu zero. Ca o verificare intermediară, returnaţi $P^{-1}(10)$ în ieşirea `Pinv10` a funcţiei.
- Returnaţi în `theta50` vectorul de parameteri $\theta(49)$ obţinut după procesarea a 49 eşantioane,[1] şi în `thetaN` vectorul final de parametri $\theta(N)$ după procesarea întregului set de date.
- Folosind funcţia `compare`, comparaţi *pe datele de validare* calitatea celor două modele. Care model este mai bun, şi de ce?
- Opţional, dacă aveţi timp, repetaţi experimentul, dar de această dată cu $\delta = 100$. Gândiţi-vă la rezultate. Pentru care valoare a lui δ sunt mai proaste modelele, şi de ce?

---

iniţializează $\widehat{\theta}(0)$ (vector coloană $na + nb$), $P^{-1}(0)$ (matrice $(na + nb) \times (na + nb)$)
loop la fiecare pas $k = 1, 2, \ldots$
  extrage $u(k)$, $y(k)$
  formează vectorul de regresori ARX:
$$\varphi(k) = [-y(k-1), \cdots, -y(k-na), u(k-1), \cdots, u(k-nb)]^\top$$
  găseşte eroarea de predicţie $\varepsilon(k) = y(k) - \varphi^\top(k)\widehat{\theta}(k-1)$ (un scalar)
  actualizează inversa: $P^{-1}(k) = P^{-1}(k-1) - \frac{P^{-1}(k-1)\varphi(k)\varphi^\top(k)P^{-1}(k-1)}{1+\varphi^\top(k)P^{-1}(k-1)\varphi(k)}$
  calculează ponderile: $W(k) = P^{-1}(k)\varphi(k)$ (vector coloană $(na + nb)$)
  actualizează parametrii: $\widehat{\theta}(k) = \widehat{\theta}(k-1) + W(k)\varepsilon(k)$
end loop

Funcţii relevante din toolbox-ul de identificare a sistemelor: `rarx`, `idpoly`, `compare`. Indicii adiţionale:

[1] Soluţia de referinţă are un offset de 1 din greşeală, ne cerem scuze.

- 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 ARX, unde `Ts` este perioada de eşantionare. Nu uitaţi că toţi vectorii de coeficienţi din polinoame trebuie să conţină întotdeauna coeficienţii constanţi (puterea 0 a argumentului q^{-1}), care trebuie să fie 1 în A, şi 0 în B. Ţineţi cont că matricea de parametri returnată de algoritm *nu* conţine aceşti coeficienţi constanţi.

- Matlab oferă funcţia `rarx`, pe care o puteţi folosi dacă doriţi ca să vă verificaţi rezultatele. Funcţia preia la intrare setul de date de identificare, ordinele modelului na şi nb şi întârzierea nk sub formă de vector, argumentele `'ff', 1` care configurează algoritmul la fel cu cel din curs, vectorul iniţial de parametri $\theta(0)$, şi matricea inversă iniţială $P^{-1}(0)$. Matricea numită P în documentaţia funcţiei Matlab `rarx` este de fapt matricea *inversă* P^{-1} din curs, fiţi aşadar atenţi când o alegeţi. Funcţia produce la ieşire o matrice $\Theta \in \mathbb{R}^{N\times(na+nb)}$ conţinând pe fiecare linie k vectorul de parametri $\theta(k)$: întâi coeficienţii $a_1, \ldots, a_{na}$ ai polinomului A, şi apoi coeficienţii $b_1, \ldots, b_{nb}$ din B.