# **Identificarea sistemelor** – Laborator 8
# Identificarea modelelor de tip Output Error folosind metoda Gauss-Newton

## Organizare

Acest laborator se rezolvă independent de către fiecare student. Doar în situaţia în care există mai mulţi studenţi decât calculatoare, *cu acordul explicit al profesorului de laborator pentru fiecare grup*, studenţii se pot grupa câte doi la un calculator.

Soluţia constă din cod Matlab. Dezvoltaţi acest cod într-un singur script Matlab. Dacă aveţi nevoie de funcţii, acestea pot fi locale în script, vezi funcţii locale în scripturi Matlab.

Regulile generale pentru laboratoare sunt descrise pe site. Pentru fiecare laborator, prezenţa dvs. va fi înregistrată numai dacă aveţi o soluţie originală şi funcţională. Profesorul va verifica **funcţionalitatea** codului dvs. timpul orei de laborator. Doar după aceea, pentru verificarea **originalităţii**, încărcaţi soluţia dvs. aici:

`https://www.dropbox.com/request/ApI3XWD1aQmX0c38xYYz`

Încărcaţi *o singură data, un singur fişier .m, denumit exact după următorul model*:

`L8_ROgs_NumePrenume.m`

unde g este grupa, s semigrupa, urmate de numele şi prenumele dvs. De exemplu, `L8_RO31_PopAlex.m`. Dacă aţi lucrat în grup cf. procedurii de mai sus, încărcaţi un singur fişier cu ambele nume ale studenţilor din grup. Fişierele duplicate, nonstandard, denumite în mod necorespunzător sau care corespund unor soluţii necontrolate încă de profesorul de laborator nu vor fi luate în considerare. Fişierele vor fi testate automat pentru plagiat, iar orice soluţie care nu trece acest test va fi marcată copiată; doar soluţiile care trec atât testul de funcţionalitate, cât şi pe cel de originalitate, sunt validate definitiv. Prin urmare, chiar dacă sunteţi încurajaţi să discutaţi idei şi algoritmi între colegi, trimiterea şi împrumutarea unor pasaje de cod este strict interzisă.

## Descrierea laboratorului

Fiecărui student i se alocă de către profesor un index pentru setul de date. Apoi, studentul descarcă fişierul Matlab ce formează baza laboratorului de pe pagina cursului. Fişierul conţine datele de identificare în variabila `id`, şi 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`.

Se ştie în avans că sistemul este de ordinul 1, fără timp mort, şi este afectat doar de zgomot de măsurare $e(k)$ pe ieşire. Ca atare, următoarea formă de tip Output Error este potrivită pentru modelarea acestui sistem:

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

cu parametrii $\theta = [f, b]^\top$. Obiectivul nostru va fi implementarea metodei erorii de predicţie pentru această structură particulară de model, folosind metoda de optimizare Gauss-Newton. Algoritmul este rezumat pe pagina următoare, într-un mod mai direct decât felul în care a fost explicat în curs, pentru a ajuta cu implementarea.

Cerinţe:

- Calculaţi formulele recursive necesare în algoritm, pe hârtie sau la tablă. Indiciu: vezi cazul ARMAX de ordinul 1 exemplificat în curs.
- Implementaţi algoritmul şi rulaţi-l pe datele de identificare, pornind de la valori *nenule* ale parametrilor iniţiali.
- Pentru valorile (aproape) optime ale $f$ şi $b$ obţinute, creaţi un model de tip OE în formatul toolboxului de identificiare, folosind `idpoly`. De notat că sintaxa funcţiei este `idpoly(A,B,C,D,F,0,Ts)` unde trebuie specificat zeroul iniţial în $B$, constanta 1 iniţială în $F$, şi perioada de eşantionare se poate găsi de ex. în setul de date de identificare. Polinoamele pe care nu le folosiţi pot fi egale cu 1. Folosiţi `compare` pentru a verifica performanţa modelului pe datele de validare.
- Dacă modelul nu este suficient de bun, acordaţi $\alpha$, $\delta$ şi $\ell_{\max}$ (eventual împreună cu $\theta_0$), pentru a îmbunătăţi performanţa.

---

alege pasul $\alpha$, parametrii iniţiali $\theta_0$, pragul de convergenţă $\delta$, şi numărul maxim de iteraţii $\ell_{\max}$
iniţializează counterul de iteraţii $\ell = 0$
calculează formulele recursive pentru $\varepsilon(k)$, $\frac{d\varepsilon(k)}{d\theta} = [\frac{d\varepsilon(k)}{df}, \frac{d\varepsilon(k)}{db}]^\top$
**repeat**
  cu parametrii curenţi $\theta_\ell$:
    simulează (aplică formulele recursive) pentru a calcula $\varepsilon(k)$, $\frac{d\varepsilon(k)}{d\theta}$, pentru $k = 1, \ldots, N$
    calculează gradientul funcţiei obiectiv cu $\frac{dV}{d\theta} = \frac{2}{N}\sum_{k=1}^{N} \varepsilon(k)\frac{d\varepsilon(k)}{d\theta}$
    calculează Hessianul aproximat al funcţiei obiectiv, cu $\mathcal{H} = \frac{2}{N}\sum_{k=1}^{N} \frac{d\varepsilon(k)}{d\theta}\left[\frac{d\varepsilon(k)}{d\theta}\right]^\top$
  aplică formula de actualizare Gauss-Newton: $\theta_{\ell+1} = \theta_\ell - \alpha\mathcal{H}^{-1}\frac{dV}{d\theta}$
  incrementează counterul: $\ell = \ell + 1$
**until** $\|\theta_\ell - \theta_{\ell-1}\| \leq \delta$, sau $\ell_{\max}$ a fost atins

---