

# Identificarea sistemelor – Laborator 7

## Semnalul pseudo-aleator binar

### Organizare

- Acest laborator este parte obligatorie a cursului de Identificarea Sistemelor. Laboratorul se rezolvă independent de către fiecare student.
- Soluția constă din cod Matlab. Codul va fi verificat și rulat de către profesor pentru a vă lua în considerare prezența la laborator. Vom efectua această verificare pe cât posibil în timpul laboratorului, împreună cu dvs. Scrieți în orice caz codul de o manieră clară, adăugând comentarii acolo unde este necesar, pentru a-l face inteligibil și în lipsa explicațiilor verbale. La sfârșitul laboratorului trimiteți codul profesorului prin email (Zoltán Nagy la [zoltan.nagy@aut.utcluj.ro](mailto:zoltan.nagy@aut.utcluj.ro), sau Marius Costandin la [marius.costandin@aut.utcluj.ro](mailto:marius.costandin@aut.utcluj.ro)) sub forma unui fișier .m sau într-o arhivă ZIP, folosind următorul format pentru numele de fișier:  
`is_labN_indexINDEX_NUME`  
unde N este numărul laboratorului, INDEX este indexul setului de date, vezi mai jos; și NUME este numele dvs. de familie. Vă rugăm să *includeți acest nume de fișier și în subiectul emailului*.
- Discutarea ideilor între studenți este permisă și chiar de dorit, dar copierea sau schimbul direct de cod este interzis. Încălcarea acestei reguli va duce la invalidarea soluției.

### Descrierea laboratorului

În acest laborator vom studia generarea și proprietățile semnalelor pseudo-aleatoare binare, SPAB. Vezi materialul de curs, partea 6: *Semnale de intrare*.

Fiecărui student i se alocă un index de către profesor. Apoi, studentul descarcă de pe pagina cursului:

[http://busoniu.net/teaching/sysid2017/index\\_ro.html](http://busoniu.net/teaching/sysid2017/index_ro.html)

funcția `system_simulator`, care aplică sistemului un semnal dat de intrare și obține datele experimentale corespunzătoare. Pentru acest laborator, cum nu avem acces la sistemul real, îl vom înlocui cu această funcție de simulare, atât pentru identificare cât și pentru validare. Funcția este dată în așa-numitul p-code, care ascunde sursa Matlab. În acest fel, vom putea trata simulatorul ca pe un sistem necunoscut, după cum ar fi cazul și în practică. Formatul acestei funcții este `data = system_simulator(index, u)` unde `index` este indexul alocat, `u` este secvența de intrare (în timp discret), și `data` este setul de date experimentale, returnat ca un obiect standard de tip `iddata`.

Se știe de la operator că sistemul de identificat nu are ordin mai mare decât 4, și că perturbația *nu* satisface structura ARX. Cu toate acestea, alegem structura ARX datorită simplității sale, și pentru a compensa perturbația non-standard vom alege ordine mari pentru model:  $na = nb = 15$ . De notat că pentru a identifica un model ARX, semnalul de intrare trebuie să aibă un ordin de persistență a excitației suficient de mare, vezi cursul.

### Cerințe:

- Generați mai întâi un set de date de validare cu `system_simulator`, folosind la intrare de ex. o secvență de semnale treaptă cu amplitudini de ordinul  $0.5 - 2$ , și cu o lungime în jur de 50 pași discreți pentru fiecare treaptă. Vom folosi acest set de date pentru a valida toate modelele identificate mai jos.
- Scrieți o funcție care generează un semnal de intrare de lungime  $N$  de tip SPAB care comută între valori date  $a$  și  $b$ , folosind un registru LSFR cu  $m$  biți. Parametrii  $N, m, a, b$  sunt argumente ale funcției, iar  $m$  este limitat la valorile  $3, 4, \dots, 10$ . SPAB-ul trebuie să aibă perioada  $P$  maximală, și dacă  $N > P$ , atunci semnalul de intrare va consta din mai multe repetări ale SPABului. Testați această funcție pentru câteva valori ale argumentelor  $N, m, a$  și  $b$ . Indicii: Folosind reprezentarea în spațiul stărilor a registrului de tip LSFR, se poate crea ușor o implementare vectorizată în Matlab. Puteți folosi funcția `mod` pentru a implementa suma modulo 2.
- Generați o intrare SPAB de identificare de lungime suficientă (de ex. 300 eșantioane) cu  $m = 3$ , luând valori între  $a = 0.5$  and  $b = 1$ . Aplicați această intrare sistemului cu `system_simulator`.
- Identificați un model ARX cu datele de identificare astfel obținute, folosind funcția Matlab `arx` pentru simplitate. Calculați ordinul de PE pentru intrare. Este acesta suficient pentru a identifica parametrii ARX? Verificați calitatea modelului ARX pentru a vedea dacă vă susține concluziile.
- Repetați procedura de mai sus, dar de data aceasta cu  $m = 10$ . Au noile date un ordin de PE suficient? Verificați calitatea modelului ARX.
- Opțional, repetați experimentul, dar folosind acum generatorul de semnal deja implementat în toolboxul de identificare al Matlab, `idinput`. Studiați interfața acestei funcții și generați un SPAB (en. *PRBS*) cu valori între  $a = 0.5$  și  $b = 1$ , fără a-i limita conținutul de frecvențe; de notat că lungimea  $m$  a registrului LSFR este aleasă automat. Studiați și posibilitățile de a genera alte tipuri de intrări (zgomot alb, sumă de sinusuri).

Funcții relevante din toolboxul de identificare a sistemelor: `arx`, `idinput`, `plot`, `compare`.