

Pachet educațional de programe pentru analiza și proiectarea sistemelor automate

Prof. dr. ing. Sorin Larionescu - UTCB

Standardul *de facto*, atât în țară cât și în străinătate, pentru Proiectarea Asistată de Calculator a Sistemelor Automate (PACSA) [2,3,4,5,12] este limbajul MATLAB (MATrix LABoratory) [13] elaborat în 1980 în scopuri educaționale și devenind accesibil comercial în 1985. La început acest limbaj asigura accesul interactiv cu rutinele algebrice EISPACK și LINPACK. În limbajul MATLAB 5 au fost elaborate foarte multe aplicații de către firma sa producătoare MathWorks și alți producători de produse software. Exemple de aplicații sunt cele 24 cutii de scule (toolboxes), mediul de dezvoltare pentru simulare Simulink 2.1, Blocksets, CtrlLab, etc.

Matlab este folosit și la Facultatea de Instalații UTCB, în special de către doctoranzi. Utilitatea sa pentru aplicațiile cursului de Automatizarea instalațiilor, pentru elaborarea proiectelor de diplomă sau disertație este însă îndoielnică. În continuare voi face o critică a Matlab din acest punct de vedere și voi prezenta pe scurt soluția alternativă prezentată.

Avantajele remarcabile ale limbajului Matlab împreună cu întreg mediul său de dezvoltare al aplicațiilor PACSA nu îl fac totuși potrivit pentru studenții facultății de Instalații și, cred eu, pentru studenții altor facultăți din România care nu se specializează direct în ingineria sistemelor automate. Argumente pentru această poziție sunt următoarele.

- Mulți studenți au calculatoare acasă sau au acces la alte calculatoare decât cele ale universității. Cumpărarea de către licențe pentru Matlab, cutiile de scule și Simulink nu este posibilă din motive financiare.
- Matlab permite elaborarea algoritmilor de conducere pentru regulatoare și controlere. Totuși, la facultatea noastră se acordă o atenție deosebită automatelor discrete logice implementate prin scheme cu contacte și relee sau automate programabile logice. Analiza și proiectarea acestor automate în mediul Matlab nu este avantajoasă.
- Proiectarea unui sistem automat nu înseamnă doar elaborarea algoritmului de conducere. Proiectul de automatizare se referă și la traductoare, elemente de execuție, produse software pentru automate programabile sau calculatoare de

proces, protocoale de comunicație, dulapul sau cutia de automatizare. Proiectarea asistată de calculator a acestor componente înseamnă mai mult decât folosirea Matlab.

- Utilizatorii începători sau ocazionali, cum sunt studenții noștri, au dificultăți în folosirea Matlab datorită modului în care se manipulează concepte de bază, cum ar fi de exemplu funcția de transfer. De exemplu, o funcție de transfer poate fi creată cu programul din lista 1. Se observa ca $H(s)$ este definit prin două polinoame numărătorul și numitorul, care la rândul său sunt definite prin matricea coeficienților. Înmulțirea factorilor numitorului este efectuată prin operația de convoluție *conv* a matricilor coeficienților. Pentru tipărirea lui $H(s)$ și calculul locului rădăcinilor este necesar să se specifice două structuri de date, numărătorul și numitorul. Programul din lista 1 necesită instalarea în Matlab a Control System Toolbox. În cazul folosirii modelului de stare sistemul este identificat de fiecare dată prin cele patru matrici A, B, C, D. De exemplu tipărirea modelului se face cu comanda `printsys(A,B,C,D)`. Trebuie să recunoaștem că pentru un utilizator începător sau ocazional acest prim contact cu PACSA prin intermediul Matlab nu este prea încurajator.

Lista 1. Program simplu MATLAB

```
% Se creeaza functia de transfer Hf(s)=1/(s+1)(s+0.5)(s+0.33)
% si se determina locul radacinilor
%
num=[1] % Numaratorul lui Hf
den=conv(conv([1 1], [1 0.5]), [1 0.33]) % Numitorul lui Hf
printsys(num,den, 's') % Tiparire Hf
rlocus(num,den); % Loculradacinilor lui Hf
```

Pachetul de programe educațional folosit de mine propune studenților folosirea programului Scilab [17] pentru proiectarea asistată de calculator a sistemelor automate. Acesta seamănă foarte mult cu Matlab, este freeware și nu are neajunsurile menționate mai înainte. De exemplu, funcția de transfer $H(s)$ este construită natural, asemănător cu modul de lucru manual (lista 2). Mai mult, algebra funcțiilor de transfer și identificarea blocului este intuitivă și simplă. Conectarea blocurilor în serie este echivalentă cu înmulțirea funcțiilor de transfer, conectarea în paralel este echivalentă cu adunarea, iar conectarea cu reacție negativă este descrisă de un operator special. Față de construirea manuală obișnuită a funcțiilor de transfer se fac două operații suplimentare: definirea variabilei s și definirea tipului sistemului. Aceasta este necesar deoarece funcțiile de transfer pot reprezenta atât sisteme continue cât și sisteme discrete. Sistemul implicit este continuu. Această performanță SCILAB a realizat-o prin introducerea unei noi structuri de date: lista. Operațiile cu funcții de transfer sunt de fapt operații cu liste.

Lista 2. Program simplu SCILAB

```
// Programul scilab2.5/my/simplu.sci
// Se creeaza functiile de transfer Hf, Hc, Ht
// ale sistemului automat H0 si se analizeaza
// in domeniul frecventa si in domeniul timp.
//
// Se executa cu comanda exec 'sci/my/simplu.sci'
//
// Dupa halt se apasa o tasta!!!
//
s=poly(0, 's') // Definire s
Hf=1/((s+1)*(s+0.5)*(s+0.33)) // Partea fixa
Hf=syslin('c', Hf); // Sistem tip continuu
Hc=0.05*(1+1/s) // Compensatorul PI
Ht=1 // Traductorul
Hd=Hc*Hf // Conectarea in serie
xbasc();evans(Hd, 1);halt(); // Locul radacinilor
xbasc();bode(Hd);halt(); // Raspunsul in frecventa
xbasc();nyquist(Hd);halt(); // Hodograful
H0=Hd/.Ht // Reactia negativa
xbasc();bode(H0);halt(); // Raspunsul in frecventa
t=0:0.1:70; // Domeniul timpului
y1=csm('step', t, H0); // Raspunsul indicial y1
xbasc();plot(t,y1); // Graficul lui y1
```

Alegerea Scilab în locul Matlab rezolvă pentru noi problema financiară a licențelor și permite prezentarea conceptelor teoriei sistemelor în maniera din lista 2. Când este vorba însă de proiectare cu date reale și utilizatori începători sau ocazionali multe dificultăți rămân. Am încercat să rezolvăm aceste probleme prin elaborarea a două programe: unul de proiectare și unul de simulare.

Programul de proiectare a sistemelor automate prin metoda locului rădăcinilor a fost elaborat în Scilab și în mare rezolvă aceleași sarcini ca și programul din lista 2. Interfața cu utilizatorul este total schimbată, figura 1. Se pot face alegeri multiple, parametrii solicitați utilizatorului sunt explicați în limba română, au valori implicite rezonabile, se pot edita erorile de introducere și cel mai important, au semnificații fizice conform practicii curente. De exemplu, se solicită constante de timp și nu poli și zerouri. Proiectarea asistată de calculator a sistemelor automate prin metoda locului rădăcinilor întâmpină greutăți în cazul sistemelor cu timp mort, care sunt totuși întâlnite frecvent în practică. Programul rezolvă această problemă și alte diferite dificultăți legate de faptul că sistemele reale sunt caracterizate de constante de timp cu valori foarte diferite și de faptul că limbajul Scilab, la fel ca și Matlab,

are drept tip de bază pentru date matricea cu numere complexe. În realitate, evitarea acestor dificultăți a impus maniera de programare Matlab din lista 1 pe care am criticat-o și nu stilul compact și intuitiv posibil în Scilab și ilustrat în lista 2. În sfârșit, pentru proiectare este necesară prelucrarea unor grafice, de exemplu locul rădăcinilor și extragerea unor înfirmații numerice cu o precizie satisfăcătoare. Programul elaborat permite utilizatorilor începători sau ocazionali să realizeze rapid și ușor și această sarcină.

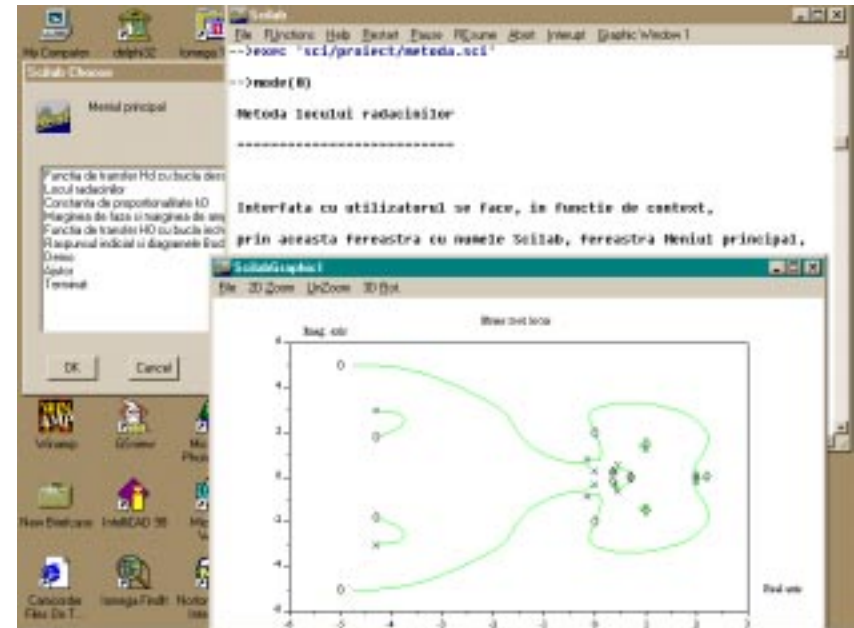


Fig. 1. Interfața cu utilizatorul a programului de proiectare elaborat în Scilab.

Am prezentat până acum avantajele programului elaborat. El are însă și dezavantaje, unele dintre ele datorate chiar faptului că a fost elaborat într-un mediu de dezvoltare tip Matlab. Utilizatorul trebuie să lucreze tot timpul cu multe ferestre, chiar la început apar trei, figura 1. Pe parcursul execuției programului se comunică cu utilizatorul prin diferite ferestre, unele dintre ele ascunse! Pentru un începător sau utilizator ocazional aceasta este foarte supărător. Poate că printr-o programare mai atentă acest neajuns ar putea fi micșorat, dar necazul cel mai mare îl constituie faptul că fereastra principală (fig. 1), care nu poate fi înlăturată, este fereastra Scilab, în care comunicarea se face, la fel ca și la Matlab, în mod text.

Comenzile de simulare și afișare grafică Scilab, lista 2, nu satisfac cerințele unui pachet PACSA. Ca și Matlabul cu al său Simulink și limbajul Scilab are un mediu de dezvoltare grafic al aplicațiilor de simulare numit Scicos (Scilab Connected Object

Simulator). Acesta este un sistem modern orientat spre obiecte și condus de evenimente, dar destinat în special experților, după părerea mea. Pentru utilizatorii începători și ocazionali am elaborat un program de simulare care se găsește în sistemul de programe numit KitSAS (Kit de Simulare și Analiză a Sistemelor) [11]. Obiectivele principale pe care le urmărește acest kit sunt necesitățile educaționale ale studenților facultății noastre. Deci fără să urmărească performanțe deosebite acesta are totuși câteva trăsături care îl pot face atractiv.

- Programele componente sunt elaborate în Delphi 4.
- Kitul conține un program de simulare DOS și un program de simulare pentru Win 95, Win 98 și Win NT.
- Programul de simulare este orientat spre obiecte conduse de evenimente.
- Printre obiectele simulatorului există și un bloc care realizează întârzierea cu timp mort fără aproximații.

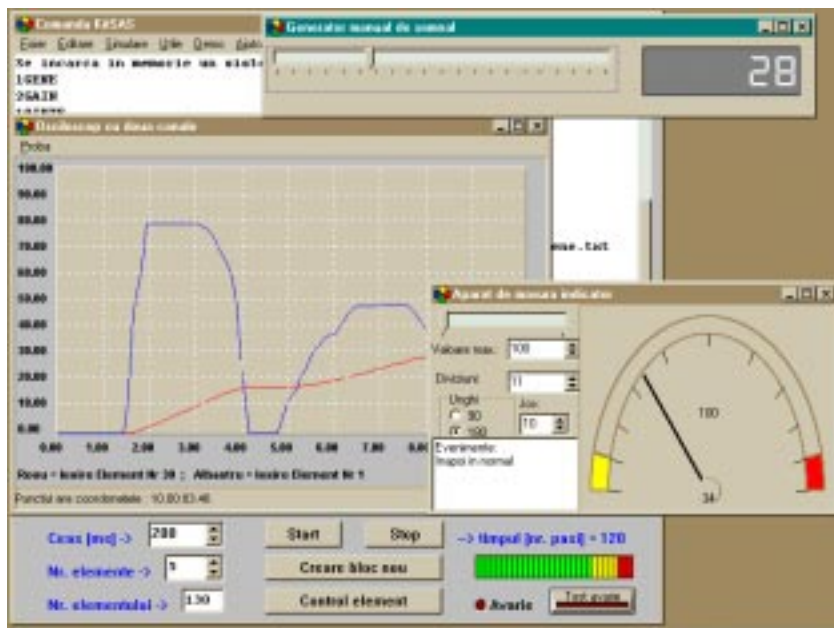


Fig. 2. Simulatorul KitSAS în timpul operării cu ceas de timp real.

- Simularea se poate face la o anumită scară de timp.
- Se pot simula ușor sisteme hibride, de exemplu sisteme automate cu procese continue și regulatoare numerice.
- Măsurările din timpul simulării pot fi efectuate cu câteva instrumente virtuale: indicator analogic, osciloscop cu două canale, generator manual de semnal (fig.2) și un înregistrator numeric și grafic (fig.3) cu o memorie de 8K.

- Rezultatele simulării pot fi prezentate atât în domeniul timp cât și în domeniul frecvență.
- Sistemul simulat poate fi construit prin mai multe metode. Elementele și structura sa pot fi descrise în cadrul unui program neprocedural de tip Matlab sau Scilab sau blocuri tip sunt interconectate și după aceea li se modifică parametrii. Dacă se folosește a doua metodă fișierul text cu descrierea sistemului se obține automat.
- Programele KitSAS de simulare sunt compilate și apoi executate spre deosebire de programele Matlab și Scilab care sunt interpretate. În timpul compilării se semnalizează erorile, de exemplu existența buclor algebrice.
- Există o bibliotecă vastă de blocuri și sisteme tipice care pot fi modificate sau modificate pentru construirea sistemului dorit.
- Există o documentație sub formă de manual [9] sau fișiere tip pdf în română.

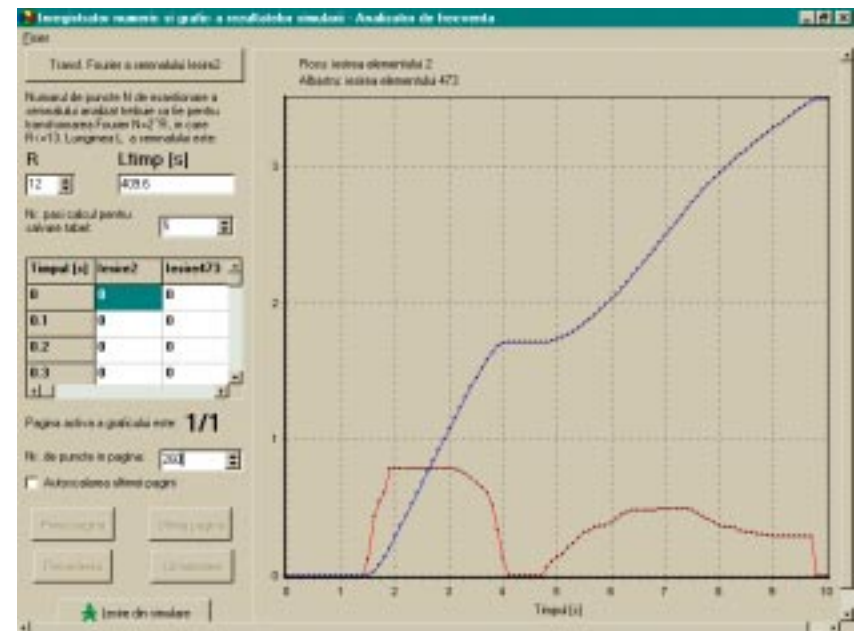


Fig. 3 Înregistratorul numeric și grafic al KitSAS.

În cadrul cursului de Sisteme automate de la Facultatea de Instalații sistemele automate discrete logice [7] ocupă aproximativ 40%. Pentru proiectarea asistată de calculator a acestor sisteme se folosește un program pentru simplificarea funcțiilor logice cu ajutorul diagramei Karnaugh [16] prezentat în figura 4, un program de simulare a automatelor

programabile ProSys [1], a cărei interfață apare în figura 5, și doua cataloage cu componente specifice domeniului instalațiilor automate din clădirile civile și industriale.

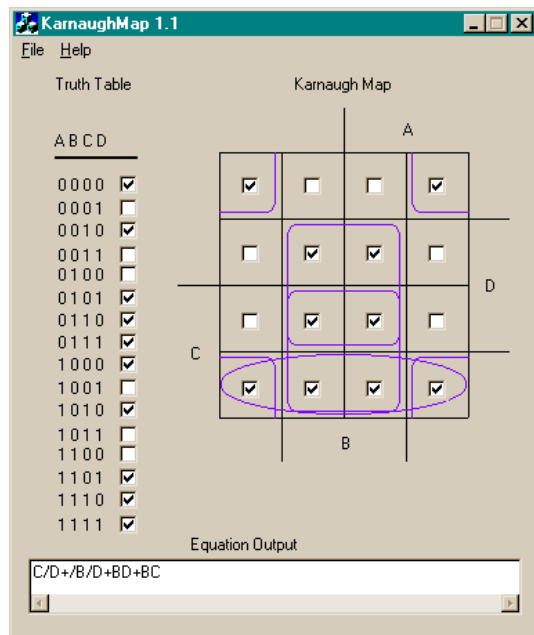


Fig. 4. Rezultatul folosirii programului pentru simplificarea funcțiilor logice.

Programul ProSys permite descrierea automatelor secvențiale în diferite moduri. La facultatea noastră se folosesc reprezentarea sistemelor automate discrete logice sub formă de schemă desfășurată sau Grafcet. Important este ca studenții să vadă că acest sistem PACSA ajută nu numai proiectarea unei componente a sistemului automat, programul Automatului programabil, dar și construiește codul și îl transferă în automat.

Proiectarea sistemelor automate pe bază de catalog este metoda principală de proiectare folosită. Din păcate această metodă nu prea este prezentă în pachetele educaționale PACSA. Ea presupune o seamă de cunoștințe și abilități din diferite domenii care trebuie transmise studenților. Menționăm câteva prezente și în pachetul nostru educațional: generarea automată de documente desenate sau scrise, manevrarea bazelor de date, elaborarea și folosirea unor mici programe, în diferite limbaje, de exemplu Excel, pentru realizarea unor sarcini concrete, de exemplu calculul fiabilității componentelor sau analize economice.

O colecție largă de demonstrații cu instrumente virtuale National Instruments face parte din pachetul educațional PACSA și permite studenților să ia contact cu ultimele realizări ale conducerii proceselor cu calculatorul.

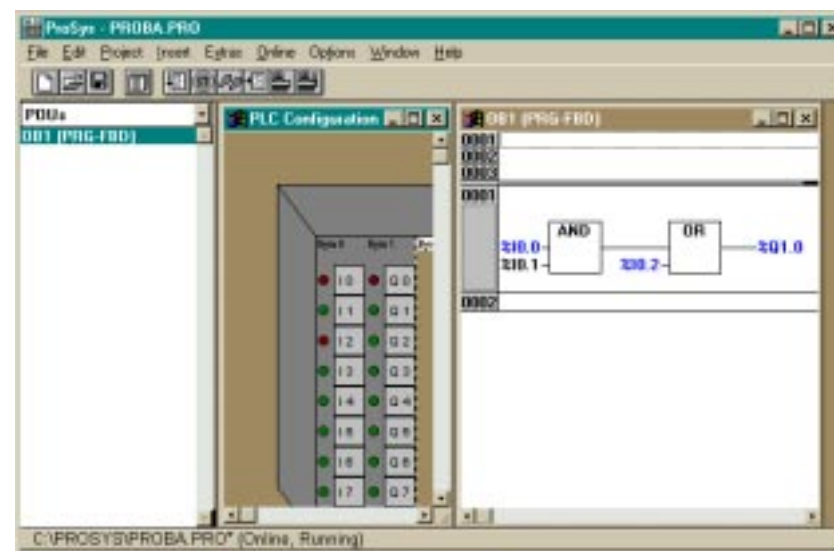


Fig. 5 Simulatorul de automate programabile.

O caracteristică modernă a sistemelor PACSA o constituie folosirea interactivă a noilor posibilități multimedia [9, 18]. Și în pachetul educațional elaborat s-a căutat o dezvoltare a intuiției ingineresti și a abilităților de manevrarea a obiectelor prin intermediul unor activități analoge celor de laborator sau de experimentare cu calculatoare analogice, mult folosite acum 25 de ani. Sistemul meu KitSAS conține componente interactive și animate. Sistemele virtuale demonstrative incluse familiarizarea cu o gamă largă de tipuri de aparate existente și experimentarea funcțiilor de bază. O comparație cu sistemele PACSA moderne prezentate în literatură evidențiază multiple similarități, atât în modalitățile de abordare teoretică cât și a mijloacelor de realizare. Deosebirea principală o constituie prezența masivă a sistemelor automate discrete logice, a aparatelor virtuale și folosirea de software propriu sau freeware. În programele elaborate s-a acordat o atenție deosebită folosirii terminologiei de specialitate corecte în limba română. Din păcate există tendința întrebuirii unui jargon româno – englez care provoacă confuzii și ignoră cultura tehnică românească existentă.

În sfârșit, un scop important al pachetului educațional PACSA prezentat îl constituie obișnuirea studentului cu explozia informativă din domeniul nostru. Toate programele și documentele prezentate pe scurt până acum ocupă un CD plin. Instalarea și examinarea lor constituie o sarcină dificilă care presupune folosirea diferitor utilitare. Pachetul educațional PACSA elaborat conține o colecție largă de astfel de utilitare în regim freeware.

Bibliografie

1. ACCON-ProSys 1131 – The complete IEC 1131-3 Programming System for SIMATIC – S5/s7.
2. Dorf R. C., The Electrical Engineering Handbook, CRC Press LLC, 1997, Rimvall C. M., Jobling Ch. P., Computer – Aided Control System Design, p. 2515 – 2535.
3. Dorf R., Bishop R., Modern Control Systems, Addison-Wesley, New York, 1998.
4. Dumitrache I., Dumitru S., Miha I., Munteanu F., Musca Gh., Calcev C., Automatizări electronice, Ed. didactică și pedagogică, Buc., 1993.
5. Dutton K., Thompson S., Barraclough., The art of control engineering, Addison-Wesley, New York, 1997.
6. Ghinea M., Fireșteanu V., MATLAB calcul numeric – grafică – aplicații, Teora, Buc., 1995.
7. Ionescu C., Larionescu S., Automatizări. Sisteme Automate Discrete Logice, UTCB, 1997
8. Ionescu C., Vlădeanu V., Larionescu S., Ionescu D., Automatizări, Ed. didactică și pedagogică, Buc., 1982.
9. Johansson M., Gafvert M., Astrom K. J., Interactive Tools for Education in Automatic Control, IEEE Control System, June 1998.
10. Lange S. T., What's All This About Object Technology, Control Engineering, August, 1997.
11. Larionescu S., KitSAS – Simularea și Analiza Sistemelor, Matrix Rom, Buc., 1999.
12. Leonard N. E., Levine W. S., Using Matlab to analyze and design control system, Addison-Wesley, New York, 1995.
13. MATLAB Product Catalog, MathWorks, 1998.
14. Part-Enander E., Sjoberg A., Melin B., Isaksson P., The MATLAB handbook, Addison-Wesley, New York, 1997.
15. Popescu D., Teoria sistemelor automate, Matrix Rom, București, 2000.
16. Sasmaori R., KarnaughMap Version 1.1, 1999, <http://puz.com/sw/karnaugh/>
17. Scilab Group, Introduction to Scilab, INRIA, Unite de recherche de Rocquencourt, Projeet Meta2, France, <http://www-rocq.inria.fr/scilab/>.
18. Wittenmark B., Haglund H., Johansson M., Dynamic Pictures and Interactive Learning, IEEE Control System, June, 1998.