ElectroTools – Software Educaţional 2.0

Budişteanu Ionuţ – elev clasa a XI-a, CN „Mircea cel Bătrân”,
ibudisteanu@acm.org
Mlisan Mirela – profesor îndumător, CN „Mircea cel Bătrân”,
mirela_mlisan@yahoo.com

Abstract

Software educațional încearcă să studieze comportamentul circuitelor elementare în curent continu și alternativ. În cazul curentului alternativ singurele încercări de simulări au fost în regim staționar și mai puțin modul tranzitoriu. Softul face parte din categoria de softuri educaționale 2.0 deoarece folosește paradigme de Inteligență Artificială pentru a crea interfețe om-calculator cât mai ușor de utilizat de către elevi și cadrele didactice.

1. Introducere

Lucrarea face parte din categoria software educaţional 2.0 ce încearcă să studieze comportamentul circuitelor elementare în curent continu şi alternativ. În cazul curentului alternativ singurele încercări de simulări au fost în regim staţionar, adică pentru valori a le lui t mai mari decât constanta de timp a circuitelor τ (pentru valori mici ale timpului $t<3*\tau$ – constanta de timp există un regim al circuitelor electronice numit regim tranzitoriu, pentru modelarea căruia se utilizează ecuaţii diferenţiale).

Scopul principal a fost ca utilizatorul să poate edita orice instanţă a problemei şi nu doar câteva animaţii flash şi lecţii. Utilitatatea educaţională a unui astfel de soft este relevantă faţă de situaţiile: lecţie sub formă de text şi animaţie flash.

2. Software educaţional 2.0

Majoritatatea aplicaţiilor educaţionale încearcă să fie o simplă lecţie pentru elev, cu o anumită teorie şi anumite exemple (imagini, animaţii, figuri), dar nu uşurează munca elevului. Aplicaţia oferă astfel multe exemple (fiind limitate de către creatorul aplicaţiei). În aplicaţia creată se pot crea multe de exemple, în funcţie de cum şi-ar dori elevul, să aprofundeze materialul. Prin convenţie software-ul reprezintă o aplicaţie care presupune un GUI (Graphic User Interface), iar utilizatorul poate “comunica” cu aplicaţia prin interfaţa grafică propusă de către programator.

Software 2.0 încearcă adăugarea unor paradigme de inteligenţă artificială şi interfeţe noi pentru aplicația respectivă: *Vocal User Interface* (VUI), *Brain Computer User Interface* (BCUI), *Natural Language User Interface* (NLUI), etc.

Pentru aplicaţia **ElectroTools**, există paradigma perceptronului (include reţele neuronale artificiale Multi-Layer-Perceptron, pentru Optical Character Recognition, Face Recognition, Face Claassification, Speech Recognition), programare logică (interfaţă în limbaj natural), şi reprezentarea bazelor de cunoştinţe cu ajutorul predicatelor de ordinul I (pentru reprezentarea teoriei), inferențierea lor cu ajutorul compilatorului de programare logica SWI Prolog.

Softul conține un mecanism de natural language processing și transformarea textului scris în consolă în clauze de predicate de ordinul I și interpretarea acestuia. Corpusul lingvistic momentan este redus, dar gramatica acceptată este destul de largă, sunt acceptate foarte multe forme gramaticale din limba română (substantive, articole hot./neh., verbe, legături, adj., adv., etc.). Componenta de NLP este concepută pentru a putea fi utilizată și în alte softuri, de asemenea și elementul de text-to-speech, speech recognition și *OCR* pot fi usor exportate și în alte aplicații.

Acestea sunt scrise cu ajutorul produselor **AILab** şi **NeurosLab** ce permit exportul de aplicaţii cu inteligenţă artificială în alte softuri. Soft-ul prezintă *Vocal User Interface*, *Natural Language User Interface*.

3. Rezolvarea aplicaţiei

Se creează un graf de tensiune, aplicând teoremele 1 si 2 ale lui Kirchhoff, şi se obţine un sistem liniar cu n ecuaţii şi m necunoscute. Dificultatea este că numărul de ecuaţii este mult mai mare decât numărul de necunoscute, adica este vorba de un sistem supradeterminat (over determinated system). Pentru rezolvarea lui într-o primă versiune s-au utilizat teoremele lui Kirchhoff. Rezolvarea sistemului liniar nu poate fi realizata cu algoritmul lui Cramer sau cu teoremele lui Rouche şi Kronecker-Capelli. Avem n ecuaţii cu m necunoscute, unde n este numărul teoremelor Kirchhoff 1 şi Kirchhoff 2 , întotdeauna n va fi mai mare decât m şi întotdeauna sistemului este rezorvabil. Teorema lui Cramer se aplică pentru sisteme liniare omogene, unde n=m, iar teormele lui Kronecker-Capelli şi Rouche se aplică atunci când numărul de necunoscute este mai mare decât numărul de ecuaţii.

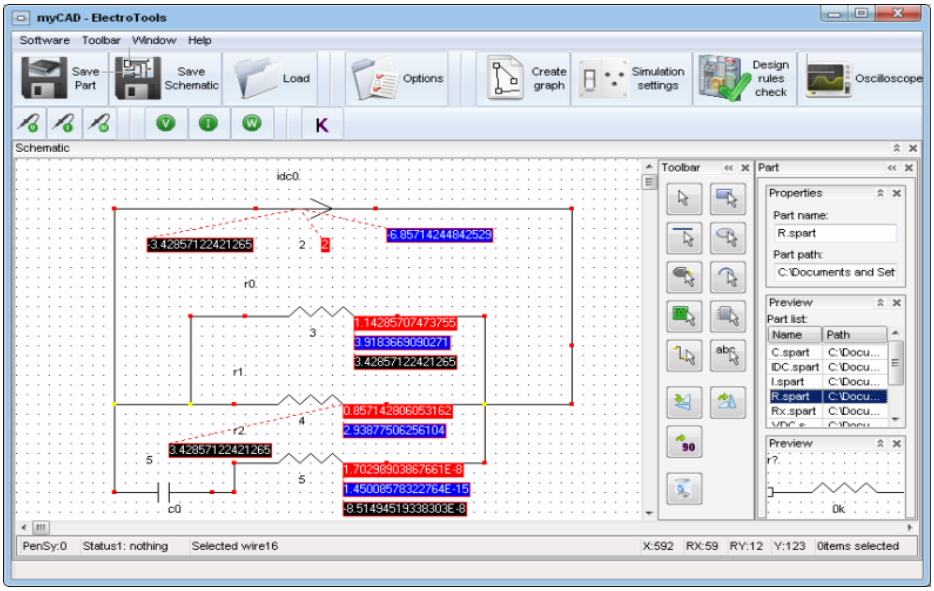


Figura 1. **Imagine cu o schemă a unui circuit în curent continu**

Marea problemă este alegerea celor m ecuaţii din mulţimea celor n ecuaţii (m<n), cele m ecuaţii trebuie să fie întotdeauna liniar independente, pentru a asigura că nu este soluţia banală (toate necunoscutele =0). De asemenea, cele n ecuaţii trebuie să conţină cele m necunoscute), alegerea se poate face printr-un algoritm polinomia. Astfel, se aleg combinări de n luate m

De exemplu, pentru 8 componente în schemă avem 19 ecuaţii şi astfel se realizează combinări de 19 luate câte 8. De fiecare dată se calculeaza un determinant de ordin 8, şi se verifica cu teorema lui Rouche dacă tot sistemul verifică soluţiile găsite. Dacă tot sistemului de 19 ecuaţii verifică soluţiile găsite, pentru acest caz particular executia a durat un sfert de oră pe un Pentium 4.

Datorită celor prezentate mai sus s-a implementat o altă metodă de rezolvare a sistemelor supradeterminate, prin metoda Gauss Elimination (numita şi Row Echelon Form). Prin această metodă aceeaşi problemă se rezolvă sub 0.1 secunde pe acelaşi calculator.

4. Simularea

Se poate ataşa un osciloscop pentru vizualizarea în domeniul timp-amplitudine a tensiunii şi curentului în orice punct al schemei. Pentru modelarea în curentul alternativ s-a simplificat problema în această etapă prin utilizarea doar a surselor cu aceeaşi frecvenţă, neputându-se utiliza surse cu frecvenţe diferite.

Pentru fiecare frecvenţă, în domeniul curent alternativ, bobinele şi condesatoarele se înlocuesc cu rezistenţele lor echivalente (reactanţa inductivă XL, reactanţa capacitivă Xc):

- XL = ω*l = 2*π*f*L
- Xc = 1/(ω*c) = 1/(2* π *f*c)

  Perioada T = 1/f

Pentru plotarea comportamentului în curent alternativ s-a calculat eşantioanele surselor la diverse momente t şi s-a creat pentru fiecare eşantion instanţa ca şi cum ar fi în curent continu. Se poate plota răspunsul circuitului la momentul T, plotându-se toate răspunsurile pentru fiecare eşantion. Se obţine răspunsul circuitului la semnalul sinusoidal iniţial. Precizăm că condesantoarele si bobinele au fost înlocuite cu rezistenţe echivalenţe în funcţie de frecvenţe Xc XL.

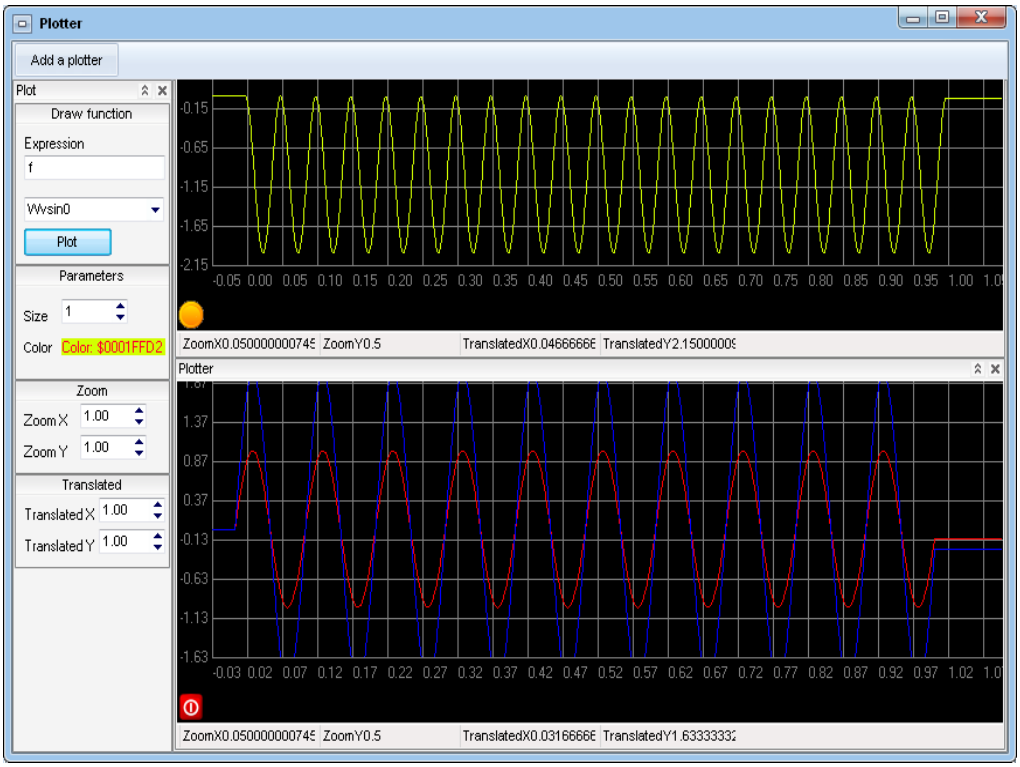
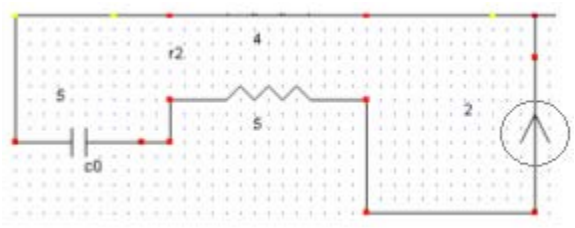


Figura 2. **Reprezentarea în domeniul timp, osciloscop-ul**

Design rules check. Softul încearcă să verifice corectitudinea schemei propuse găsind erori de genul pini neconectaţi, surse de tensiune puse în paralel, sau surse de current în serie etc.

Exemplu:

Dacă s-ar rula o simulare în curentul continu, partul c0, care este un condensator şi se comportă ca un fir întrerupt astfel nu există un drum (lanţ) la partul idc0 de la pin0 la pin1. Deoarece firul este întrerupt, astfel nu se poate simula.



Deasemenea nu se poate simula dacă unul dintre pini la orice part sunt în aer (nu sunt conectaţi).

5. Sistem expert

Softul permite înterpretarea limbajului uman. Conţine şi o foarte mica baza de cunoştinţe. Interpretarea limbajului uman presupune o analiză lexicală şi semantică a unor propoziţi iar apoi o interpretarea a propoziţiilor.

Iniţial s-a scris folosind programare logică în Turbo Prolog, dar din cauza creşterii complexitatii s-a renunţat şi s-a implementat în C++. Momentan permite: *Substantive, Verbe, Prepoziţii, Legături, Adjective, Numerale, Adverbe, Articole*. S-a utilizat metoda de rezolvare cu automate finite recursive.

Se poate face un paralelism între un limbaj uman şi un limbaj de programare.

Limbajele de programare au o gramatică simplă:

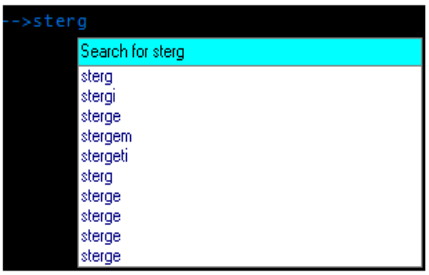
- Pascal – să fie cât mai clară, de exemplu pentru bloc există begin şi end;
- C/C++ - să fie cât mai puţin de scris.

Când se încearcă scrierea unui soft ce permite natural language processing (NLP) trebuie analizate toate cazurile privind sintaxa si semantica. Softul încearca să facă o analiză sintactică şi semantică a constructiiloe lexicale pentru a rezolva dificultatea problemelor ce trebuie depaşite în NLP. Componenta face analiză sintactică a cerinţei (query-ul) şi dacă este corectă din punt de vedere lexical , se verifică din punct de vedere semantic şi eventual se execută o acţiune.

De exemplu "*Eu aş vrea ca tu să salvezi tot* " este corect sintactic, dar semantic nu, întrucât nu s-a specificat unde să salveze. Se poate scrie orice formulare corectă şi semantică pentru orice acţiune pe care să o genereze soft-ul.

Sistemul expert propus permite următoarele genuri de propoziţii (template-uri, ele pot fi dezvoltate) :

- şterge din schematic "a" şi "b"
- şterge din schematic două parturi "a" şi "b"
- şterge un part din schematic "a" şi "b'
- şterge două parturi "a" şi "b" si "c"
- salvează un schematic în "C:\"
- salvează un fişier in "C:\"
- salvează un schematic curent în "C:\



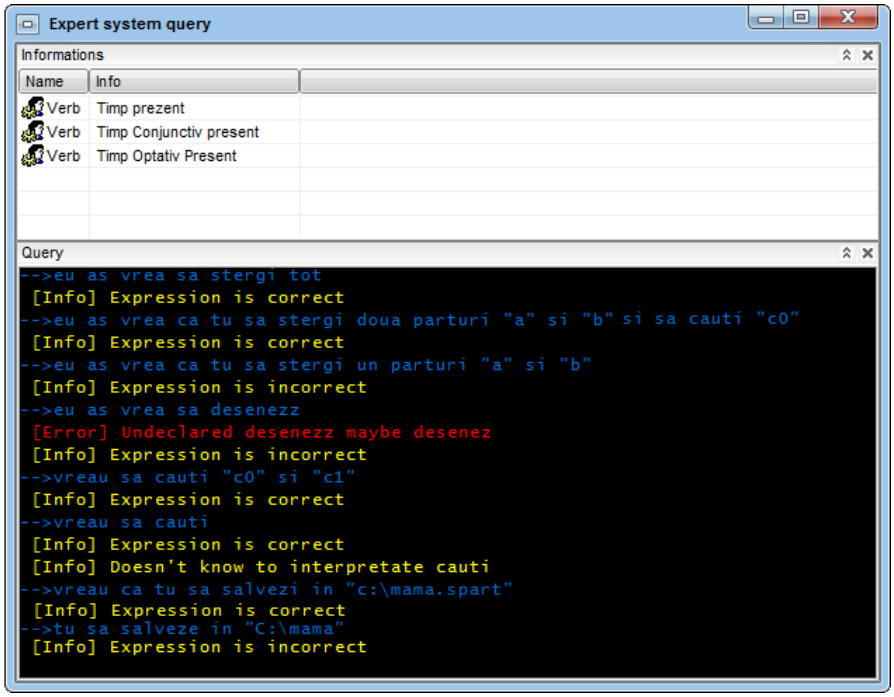
Deasemenea se permite înlănţuirea de construcţii:

- Eu aş vrea ca tu să ştergi două parturi "a" şi "b" şi să cauţi "a"
- Eu aş dori ca să ştergi tot şi aş vrea să deschizi o aplicaţie
- Vreau sa ştergi tot şi tu să închizi o aplicaţie
- Vreau ca el să şteargă tot
- Doresc să cauţi în schematic două parturi "a" şi "b"
- Eu aş dori să deschizi o aplicaţie nouă
- Eu aş dori să ştergi din schematic un part "a"

6. Optical Character recognition OCR

Soft-ul generează câteva teste aleatoare, iar apoi elevul trebuie să răspundă la intrebari (ex: Ce valoare are curentul care trece prin Vdc?). Elevul va trebui sa tasteze raspunsul, deci este nevoie

de un cod ce va face recunoasterea scrisului uman. Folosind reţele neuronale Multi-Layer-Perceptron s-a creat o clasă ce permite OCR. Într-o suprafaţă de dimensiuni 128*128, elevul „desenează” răspunsul.



Dacă vom avea o imagine de 128*128=16384 pixeli, ar trebui ca stratul de intrare să fie format din toţi pixelii. Informaţia fiind foarte mare s-a optat pe o reducere a imaginii. Informaţia pentru calculatoare este aproape aceeaşi (informaţie redundantă) atât în matricea de 128*128 cât şi într-o matrice de 10*10 (pentru calculator se pierde foarte puţină informaţie din imaginea mare în imaginea de 10*10). Din suprafaţa de desenat se face un Regional of Interes (ROI) şi după aceea se scalează la dim. de 10*10, apoi se aplică unui clasificator neuronal.

8. Face recognition

Softul permite logarea în zona de administrare după ce are loc o recunoaştere facială şi o clasificare facială. S-a folosit AILab pentru a scrie codul si a antrena retelele neuronale. S-au folosit reţele MLP şi algoritm de învăţare Back-Propagation cu moment.

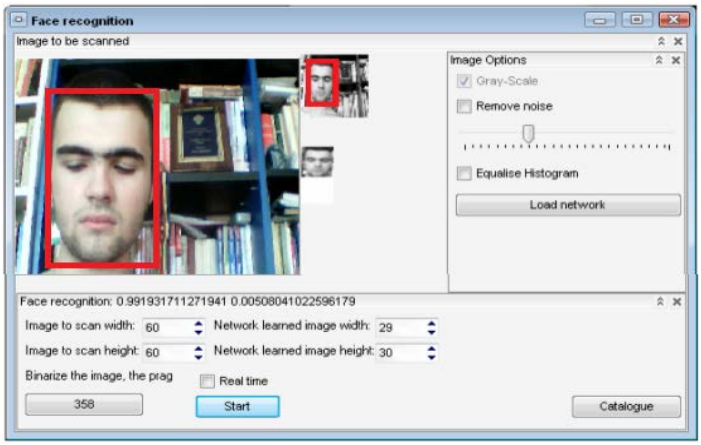
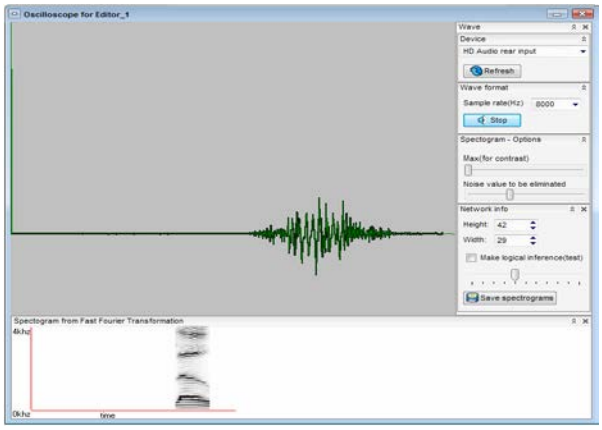


Figura 4. **Face recognition**

9. Vocal User Interface

Interfeţele vocale au început să devină foarte populare în aplicaţiile software-ului uzual. Interfaţa creată recunoaşte anumite cuvinte, reuşind să şi vorbească având înglobat un text-to-speech pentru Limba română. Exemplul permite comenzi vocale utile pentru editarea unor scheme real-time. Softul aduce în evidenţă maturitatea soluţiilor de Inteligenţă Artificială, raportat la puterea de calcul a calculatoarelor uzuale. Softul este realizat cu Reţele Neuronale *Multi-Layer-Perceptron*, sunetul este achiziţionat cu un microfon. Se face analiză spectrală a acestuia, se tranformă semnalul din domeniul timp-amplitudine în domeniul timp-frecvenţă, creându-se o spectrogramă a sunetului. Spectrogramele sunt procesate informatic şi clasificate. Soft-ul recunoaşte *Search, Delete, New, Save, Load Stop.*

Pentru implementare s-a folosit Tranformata Fourier Rapidă cu ajutorul căruia s-a creat spectrograma semnalului. Ca algoritm de învătare pentru reţelele neuronale s-a utilizat *Back-Propagation* cu moment. S-au încercat şi alte arhitecturi de reţele neuronale: *Kohonen şi Carpenter*, dar nu s-au obţinut rezultatele aşteptate.



Bibliografie

[1] Cadence PSpice Help.
[2] Cătălin-Daniel Caleanu si Virgil Tiponut - *Retele neuronale - Aplicatii*
[3] Ştefan Holban - *Retele neuronale - Arhitecturi si algoritmi*
[4] Calin Enăchescu - *Învăţarea reţelelor neuronale pe bază de exemple*
[5] Nicoale Ţăndăreanu - *Reprezentarea Bazelor de cunostiinte si intepretarea limbajelor umane, Craiova.*
[6] Nicoale Ţăndăreanu - *Sisteme Expert*, Craiova
[7] ACM Digital Library
[8] Constatin Sâmbotin - *Sisteme expert cu Prolog.*