

# Editor grafic pentru realizarea diagramelor fazoriale ale circuitelor electrice în regim permanent sinusoidal

**Marinel Popescu**

I.C.M.E.T. CRAIOVA

Bd. Decebal, 118A

marinel\_popescu@yahoo.com

**Claudiu-Mihai Bălan**

DASSAULT SYSTEMS

France

claudiu6891@yahoo.com

## REZUMAT

Lucrarea va prezenta elementele necesare realizării unui editor grafic care poate să fie utilizat pentru crearea diagramelor vectoriale și fazoriale. Realizarea interfeței editorului și a programului care administrează funcționarea implică utilizarea simplă a unor primitive grafice și a unor elemente de control uzuale. Prezentarea este accesibilă tuturor utilizatorilor de medii de programare vizuală dar, în mod special, utilizatorilor mediului de dezvoltare Delphi.

## Cuvinte cheie

Editor grafic, diagramă fazorială, regim sinusoidal, fazor.

## Clasificare ACM

H5.2. Information interfaces and presentation (e.g., HCI): Miscellaneous.

## INTRODUCERE

Dezvoltarea spectaculoasă a mijloacelor hardware și software care sunt oferite utilizatorilor a contribuit la extinderea fără precedent a posibilităților de analiză și investigare a comportării sistemelor tehnice complexe (și nu numai) folosind modele elaborate pe baza ecuațiilor de funcționare. Acestea stabilesc relații de legătură între mărimile de intrare și cele de ieșire ale sistemului analizat și pot oferi informații utile și complete despre evoluția uneia sau mai multor mărimi de ieșire atunci când sistemul este excitat de o mărime de intrare cu o evoluție standard.

Pentru sistemele electrice, în special, este cunoscut faptul că analiza comportării în regim tranzitoriu se face pe baza evoluției mărimilor de ieșire atunci când mărimile de intrare sunt de tip treaptă unitară, rampă unitară sau impuls unitar. Reprezentarea grafică a evoluției mărimii de ieșire oferă informații despre durata regimului tranzitoriu, timpul de creștere al mărimii de ieșire, stabilitatea sistemului și valoarea erorii după stabilizare, eventuale supracreșteri, etc. [1], [2], [3].

De asemenea, sistemul poate să fie analizat din punct de vedere al comportării în domeniul frecvențelor. În acest caz este urmărită evoluția raportului dintre amplitudinea unei mărimi de ieșire și cea a unei mărimi de intrare sinusoidale precum și evoluția defazajului dintre cele două mărimi când frecvența semnalului de intrare parcurge tot domeniul de interes [4], [5]. În cele două situații descrise mai sus pot fi foarte sugestive reprezentările grafice ale formelor de undă ale mărimilor analizate, respectiv, reprezentările grafice ale diagramelor Bode.

Pentru sistemele electrice monofazate sau trifazate, care funcționează în regim permanent sinusoidal, este utilizat

adesea un instrument grafic de analiză numit diagramă fazorială. Se știe că orice mărime fizică ce are evoluție sinusoidală în timp, cu o frecvență dată, este complet cunoscută dacă se precizează valoarea efectivă și faza inițială a mărimii respective.

Unei astfel de mărimi  $i$  se poate asocia un vector liber, reprezentat într-un plan, care, la rândul său, este complet cunoscut dacă îi sunt precizate modulul și argumentul (unghiul pe care direcția vectorului îl face cu direcția axei orizontale a sistemului de referință în acel plan). Când vectorul este reprezentat în planul complex poartă numele de fazor. Extremităților lui  $i$  se pot asocia numere complexe. Utilizarea numerelor complexe în operații matematice cu mărimi sinusoidale conduce la un volum redus de calcule simple [6], [7], [8].

În literatura tehnică din domeniul electric sunt cunoscute mai multe referințe în care unele mărimi electrice apar în reprezentări grafice prin diagrame fazoriale.

În [9] este prezentat un instrument de măsurare virtual destinat efectuării lucrărilor de laborator care au ca scop analiza rețelelor trifazate și a parametrilor specifici în diverse regimuri de funcționare. Pe panoul frontal al aplicației, realizată în mediul LabView, apare o reprezentare în coordonate polare a fazorilor. Nu se menționează dacă reprezentarea respectivă permite editarea și măsurarea în regim grafic a unor mărimi suplimentare (de exemplu, mărimile de linie, când în diagramă sunt reprezentate mărimile de fază).

În [10] este prezentat un exemplu de utilizare a limbajului grafic LabView pentru reprezentarea caracteristicilor de funcționare ale unui generator sincron trifazat. De asemenea, nu se menționează dacă diagramele fazoriale reprezentate pot fi editate cu scopul vizualizării și măsurării unor mărimi suplimentare pe diagrama existentă.

În [11] este prezentat un instrument virtual pentru instruirea interactivă prin prezentarea funcționării unui generator trifazat cu impedență de sarcină cu configurație stea și/sau triunghi. Nici în acest caz nu se precizează dacă se pot efectua măsurări în mod grafic pe o diagramă existentă.

Editorul grafic, prezentat în lucrare, este o aplicație software care folosește câteva primitive geometrice (punct, segment, semidreaptă, poligon) pentru a crea, edita și salva în format text sau în format grafic diagrame vectoriale sau fazoriale trasate într-un sistem de referință bidimensional.

Scopul lucrării este acela de a pune la dispoziția utilizatorului interesat o aplicație, realizată în mediul de dezvoltare Delphi, care să facă posibilă:

- ▶ crearea și editarea diagramelor fazoriale (utilizarea ca editor grafic);
- ▶ efectuarea de măsurări ale modulelor, fazelor inițiale și defazajelor direct în planul de reprezentare;
- ▶ efectuarea de operații cu mărimi vectoriale: adunări, scăderi, produse scalare, etc.;
- ▶ efectuarea unor transformări geometrice în plan: translații, rotații, reflexii sau oglindiri, reprezentări simetrice, înmulțirea cu un scalar, etc.;
- ▶ utilizarea controalelor, procedurilor și funcțiilor grafice oferite de mediul de dezvoltare integrat Delphi;
- ▶ aplicarea combinată a cunoștințelor de programare, de aritmetica numerelor complexe, de electrotehnică, de grafică computerizată;
- ▶ integrarea elementelor esențiale ale aplicației în proiecte ale unor sisteme de măsurare și monitorizare a funcționării echipamentelor electrice.

Lucrarea se adresează deopotrivă elevilor, studenților, care în perioada studiilor au dobândit cunoștințe despre utilizarea limbajului Pascal sau Delphi, și cadrelor ce se ocupă cu instruirea în programare, care o pot folosi pentru exemplificări în timpul predării acestor limbaje sau ca bază pentru dezvoltarea unor noi aplicații.

În prima secțiune este prezentat panoul de control al aplicației cu elementele grafice și controalele prezente pe suprafața lui și funcțiile asociate acestora.

În secțiunea a doua sunt prezentate structura și modul de partajare ale suprafeței de desenare, elementele grafice inițiale prezente la lansarea aplicației.

Secțiunea a treia prezintă mărimile caracteristice vectorilor și fazorilor, modul de reprezentare a acestora pe suprafața de lucru și de referire folosind elemente de sintaxă specifice mediului Delphi.

Secțiunea a patra conține o scurtă referire la operațiile ce pot fi efectuate în mod grafic cu vectori și fazori.

La final sunt prezentate exemple utilizare și de integrare a aplicației în proiecte tehnice de măsurare, monitorizare și reprezentare grafică prin fazori a unor mărimi specifice sistemelor electrice mono sau trifazate.

### PANOU DE CONTROL AL APLICAȚIEI

Panoul de control al editorului a fost realizat pe suprafața unei forme (fig.1.). În partea stângă a panoului sunt plasate două componente:

- una de tip *Image* pe al cărei tablou (*Canvas*) este reprezentat sistemul de axe de coordonate în cadrul căruia se face reprezentarea diagramei;
- una de tip *StringGrid* (tabel cu șir de caractere) în care sunt afișate date referitoare la mărimile vectoriale sau fazoriale reprezentate: notarea, modulul, faza, proiecția pe axa absciselor, proiecția pe axa ordonatei, abscisa și ordonata punctului inițial (originii) P1 și abscisa și ordonata punctului final (vârfului) P2.

În partea dreaptă sunt plasate minipanouri pentru comenzi și afișare. Minipanoul "Axe" conține o componentă de tip

*RadioGroup* cu patru *RadioButton* (butoane cu menținerea poziției) care servesc la alegerea notării sistemului de axe utilizat. La realizarea editorului grafic s-a ținut seama de similitudinea reprezentării numerelor complexe cu cea a reprezentării vectorilor și fazorilor în planul cu diverse coordonate. Sistemul de coordonate d-q este frecvent utilizat în cadrul transformării modelului trifazat al mașinilor de curent alternativ în model bifazat cu evidențierea axelor câmpului și cuplului electromagnetic.

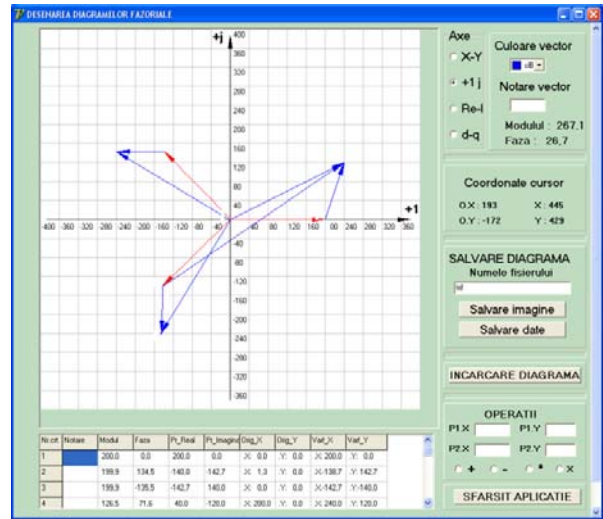


Figura 1. Panoul de control al editorului grafic

Tot în partea dreaptă (sus) se găsește un alt minipanou care conține:

- o componentă de tip *ColorBox* (casetă cu culori) care servește la selectarea culorii cu care va fi reprezentat următorul vector în diagramă;
- o componentă de tip *Edit* (casetă pentru editare de text) în care utilizatorul notează cu doua caractere originea, respectiv extremitatea vectorului ce urmează a fi desenat;
- o componentă de tip *Label* (Etichetă) în care, la trasarea unui vector, în timpul mișcării mouse-ului pe componenta imagine, sunt afișate modulul și faza pentru poziția curentă a cursorului.

Pe minipanoul "Coordonatele cursorului", la mișcarea cursorului pe imagine, sunt afișate coordonatele cursorului atât în sistemul de axe cât și în raport cu colțul stânga-sus al imaginii. Prima pereche de coordonate este foarte utilă la marcarea punctului inițial (originii) vectorului ce urmează a fi desenat.

Următorul minipanou servește la precizarea numelui fișierului și la transmiterea comenzilor de salvare a diagramei desenate într-un dosar cu calea 'C:\Diagrame vectoriale și fazoriale'.

În format grafic, ca bitmap, fișierul este salvat, la apăsarea butonului "Salvare imagine", cu comanda:

```
Image1.Picture.SaveToFile(cale+nume
fisier+'.bmp');
```

În format text, prin apăsarea butonului "Salvare date", cu succesiunea de proceduri cunoscută, se înscriu date în fișierul creat:

```
numefis:=Edit4.Text;
```

```
AssignFile(Fis,director+Edit4.Text+'.txt');
Rewrite(Fis);
WriteLn(Fis,'FISIER : '+Edit4.Text+'.txt');
data:=Now;
WriteLn(Fis,' '+DateToStr(data));
WriteLn(Fis,' '+TimeToStr(data));
WriteLn(Fis,' ');
WriteLn(Fis,' NC '+' Notare '+' Modul
 '+' Faza '+' Pr_X '+' Pr_Y '+' P1_X
 '+' P1_Y '+' P2_X '+' P2_Y ');
CloseFile(Fis);
```

Fis - variabila de tip *TextFile*;

director - calea până la directorul în care urmează să fie salvat fișierul al cărui nume (numefis, variabilă de tip string) este completat de utilizator în componenta *Edit4*. Conținutul minimal al antetului fișierului text este prezentat în fig.2.

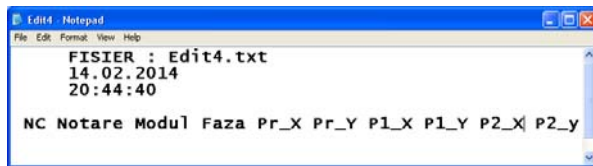


Figura 2. Antetul fișierului text utilizat pentru salvarea diagramei

Este util ca în antet să fie conținute și valorile scării de reprezentare pe axa absciselor ( $S_x$ ) și pe axa ordonatelor ( $S_y$ ), precum și sistemul de axe utilizat pentru reprezentare. Acestea servesc la refacerea condițiilor în care a fost creată și salvată diagrama.

Acest antet este urmat de linii cu conținutul celulelor din *StringGrid*, de la linia 2 până la ultima linie completată în acesta.

Acționarea butonului "Încărcare diagramă" are ca efect:

- încărcarea bitmap-ului în imagine pentru vizualizare și continuarea editării cu comanda:

```
Image1.Picture.LoadFromFile(cale+nume
fisier+'.bmp');
```

- preluarea valorilor pentru refacerea condițiilor de reprezentare inițiale;
- încărcarea datelor din fișier în celulele *StringGrid* folosind procedurile *Reset()*, *ReadLn()* și *StringGrid1.Cells[i,j]:=sir\_ij*, unde *sir\_ij* este șirul înscris în fișierul text pentru valoarea înscrisă în *StringGrid* la coloana *i* și linia *j*.

### SUPRAFAȚA DE DESENARE

Ca suport grafic pentru desenarea diagramei este folosit un control tip *Image* cu lățimea de  $IW$  pixeli și înălțimea de  $IH$  pixeli. În planul imaginii pixelul din colțul stânga-sus are coordonatele  $(0,0)$ , iar pentru cel din colțul din dreapta-jos coordonatele sunt  $(IW,IH)$ .

Originea sistemului de axe, punctul  $O$ , este fixată în centrul imaginii și are coordonatele  $(O.X, O.Y)$ :

$$O.X := Round(0,5 IW); \quad (1)$$

$$O.Y := Round(0,5 IH); \quad (2)$$

Funcția *Round()* întoarce partea întregă a valorii raportului. Prin acest punct vor trece cele două axe ale sistemului de coordonate rectangulare. Pentru desenarea axei  $Ox$  se poate parcurge succesiunea de operații:

- mutarea stiloului de desenare în punctul limită-stânga al axei,  $(0, Round(IH/2))$ , folosind procedura:

```
Image1.Canvas.MoveTo(0, Round(IH/2));
```

- trasarea unei linii până în punctul limită dreapta,  $(IW, Round(IH/2))$ , folosind procedura:

```
Image1.Canvas.LineTo(0, Round(IH/2));
```

- trasarea triunghiului care indică sensul axei în punctul limită-dreapta al axei, folosind procedura:

```
Image1.Canvas.Polygon([...]);
```

În locul punctelor din paranteza dreaptă se specifică coordonatele celor trei puncte care definesc triunghiul. Sistemul de axe poate fi trasat în cadrul procedurii de creare a formei aplicației.

Un punct  $A$  cu coordonatele  $(A.X, A.Y)$  în sistemul  $xOy$  va avea coordonatele  $(O.X+A.X, O.Y-A.Y)$  în planul imaginii. Dacă punctul  $A$  are coordonatele  $(-X, -Y)$  în  $xOy$  atunci în planul imaginii coordonatele vor fi  $(O.X-A.X, O.Y+A.Y)$ .

Pe suprafața componenteii vor fi desenate segmente orientate (săgeți) care vor reprezenta vectorii sau favorii din diagramă. Stabilirea mărimii, direcției și sensului segmentului este realizată cu ajutorul cursorului *mouse*-lui. Pentru a desena un segment orientat în planul imaginii, se plasează cursorul în punctul în care se dorește fixarea originii vectorului. Pe panoul de comenzi al editorului sunt afișate coordonatele în raport cu sistemul de axe cu originea în  $O$ . La apăsarea butonului (eveniment *OnMouseDown*) sunt memorate coordonatele punctului  $A$ .

Deplasarea cursorului pe imagine corespunde unui eveniment *MouseMove*. Mișcând cursorul, cu butonul din stânga apăsat, se caută poziția punctului  $B$  pentru care modulul segmentului  $AB$  și faza acestuia sunt cele dorite. În timpul mișcării cursorului editorul afișează coordonatele, fapt ce ușurează găsirea poziției punctului  $B$ . După aceasta se poate elibera butonul mouse-ului (eveniment *OnMouseUp*) și se memorează coordonatele punctului  $B$ . În mod asemănător cu trasarea axelor, este desenat apoi un segment orientat cu originea în  $A$  și extremitatea (vârful săgeții) în  $B$ .

Coordonatele punctelor  $A$  și  $B$  sunt înscrise în celulele corespunzătoare ale controlului *StringGrid*.

Dacă domeniul de variație al variabilei de pe axa absciselor este  $Dx=[-dx, +dx]$  (de ex.,  $dx=500$ ), atunci scara de reprezentare pe axa  $Ox$  va fi:

$$S_x := D_x / IW = 2 dx / IW; \quad (3)$$

Când codomeniul funcției, sau variabilei reprezentată pe axa ordonatelor, este  $Dy=[-dy,+dy]$  atunci scara de reprezentare pe axa  $OY$  va avea valoarea:

$$S_y := D_y / IH = 2 dy / IH; \quad (4)$$

Pentru exemplificare, presupunem că dorim să trasăm un vector cu originea în punctul  $A (A.X, A.Y)$  și cu vârful în punctul  $B$  aflat, față de punctul  $A$ , la 250 unități pe

orizontală și la 200 unități pe verticală. Pentru scala curentă de reprezentare, coordonatele punctului B se pot calcula cu relațiile:

$$B.X := A.X + Round(250/S_x); \quad (5)$$

$$B.Y := A.Y + Round(200/S_y); \quad (6)$$

Pentru o reprezentare cât mai precisă a elementelor grafice în planul imaginii este necesar să se utilizeze cea mai bună rezoluție posibilă pentru terminalul grafic. De asemenea, se poate selecta o viteză de mișcare adecvată pentru cursorul mouse-lui.

### MĂRIMI CARACTERISTICE VECTORILOR ȘI FAZORILOR

Atât mărimile vectoriale cât și cele fazoriale sunt reprezentate prin segmente orientate care încep într-un punct numit origine și sfârșesc într-un punct numit extremitate sau vârf. Distanța dintre cele două puncte reprezintă, la alta scară, norma sau modulul mărimii reprezentate, iar unghiul pe care direcția segmentului îl face cu axa absciselor se numește faza inițială sau argument. Segmentul orientat AB din fig.3. are proiecțiile pe cele două axe cuprinse între A.X și B.X, respectiv între A.Y și B.Y.

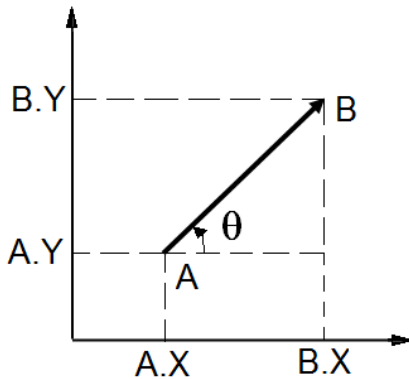


Figura 3. Reprezentarea segmentului orientat AB

Se poate considera că AB este suma vectorială a celor două proiecții reprezentate la scările curente ale celor două axe.  $\theta$  este unghiul pe care segmentul orientat îl face cu axa absciselor și poate fi exprimat ca arctangenta raportului dintre proiecția pe axa ordonatelor și proiecția pe axa absciselor.

Pentru determinarea modulului segmentului AB se folosește relația cunoscută:

$$AB = Sqrt(Sqr(B.X - A.X) + Sqr(B.Y - A.Y)); \quad (7)$$

în timp ce argumentul sau faza rezultă din:

$$\theta := arctan2(B.Y - A.Y, B.X - A.X); \quad (8)$$

Funcția  $arctan2(Y, X)$ , specifică limbajului Delphi, întoarce valoarea arctangentei raportului Y/X.

Reprezentarea vectorului sau fazorului prin modul și fază constituie reprezentarea în coordonate polare.

Reciproc, pentru reprezentarea în coordonate rectangulare, cunoscând modulul |AB| și faza  $\theta$  se pot calcula componentele:

$$B.X - A.X := AB * cos(\theta) \quad (9)$$

$$B.Y - A.Y := AB * sin(\theta) \quad (10)$$

Editorul grafic permite măsurarea modulului și fazei unui segment orientat reprezentat în planul imaginii. Pentru măsurarea modulului se plasează cursorul mouse-lui în punctul A și se apasă butonul din partea stângă (se contorizează ca primă apăsare). Procedura care tratează acest eveniment, *OnMouseDown*, trebuie să aibă ca rezultat memorarea coordonatelor cursorului, X și Y, ca și coordonate ale punctului A:

$$A.X := X; \quad (11)$$

$$A.Y := Y; \quad (12)$$

Asemănător se procedează pentru memorarea coordonatelor punctului B: se plasează cursorul în punctul B și se apasă butonul din partea stângă. Procedura care tratează cel de-al doilea eveniment *OnMouseDown* va memora coordonatele:

$$B.X := X; \quad (13)$$

$$B.Y := Y; \quad (15)$$

Evenimentul poate să fie urmat de calcularea și afișarea valorilor modulului, fazei și proiecțiilor calculate cu relațiile prezentate anterior. Pentru fiecare segment orientat desenat în planul imaginii sunt memorate și afișate într-un *StringGrid* numărul de ordine, notația, valoarea modulului și fazei, valorile proiecțiilor pe cele două axe.

### OPERATII CU VECTORI ȘI FAZORI

La întocmirea diagramelor fazoriale apare necesitatea de a realiza operații de adunare, scădere, înmulțire cu un scalar sau oglindire (obținerea simetricului) în raport cu originea [12], [13]. Aceste operații se pot realiza atât în mod aritmetic, prin calcul, folosind componentele fazorilor, cât și în mod grafic prin trasarea fazorului rezultat direct pe diagramă.

#### Adunarea fazorilor

În fig.4. este reprezentat modul în care se poate realiza însumarea grafică a doi fazori reprezentați în planul sistemului de coordonate rectangulare.

Pentru a determina vectorul sumă, AD, a vectorilor AB și AC cu ajutorul editorului grafic se va proceda astfel:

- ▶ Se marchează punctual A cu cursorul mouse-lui, prin apăsarea butonului stâng și se memorează coordonatele A.X și A.Y;
- ▶ Se marchează punctual B cu cursorul mouse-lui, prin apăsarea butonului stâng și se memorează coordonatele B.X și B.Y;
- ▶ Se marchează punctual C cu cursorul mouse-lui, prin apăsarea butonului stâng și se memorează coordonatele C.X și C.Y.

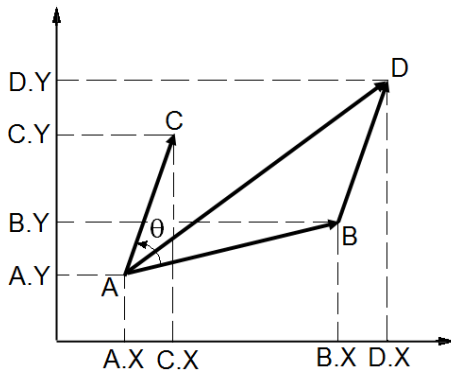


Figura 4. Reprezentarea însumării grafice a doi fazori

► Se calculează faza  $\theta_1$  a lui AB:

$$\theta_1 := \arctan 2 \frac{B.Y - A.Y}{B.X - A.X} \quad (16)$$

► Se calculează faza  $\theta_2$  a lui AC:

$$\theta_2 := \arctan 2 \frac{C.Y - A.Y}{C.X - A.X} \quad (17)$$

► Se calculează unghiul  $\theta$  dintre cei doi vectori:

$$\theta := \theta_2 - \theta_1 \quad (18)$$

► Se calculează modulele vectorilor AB și AC cu relația cunoscută, după care se poate calcula modulul vectorului sumă AD:

$$|\overrightarrow{AD}| := \sqrt{(AB)^2 + (AC)^2 + 2 AB AC \cos \theta} \quad (19)$$

**Observație:** dacă se ține seama că:

$$(B.X - A.X) + (C.X - A.X) := D.X - A.X \quad (20)$$

$$(B.Y - A.Y) + (C.Y - A.Y) := D.Y - A.Y \quad (21)$$

atunci,

$$|\overrightarrow{AD}| := \sqrt{(D.X - A.X)^2 + (D.Y - A.Y)^2} \quad (22)$$

și faza  $\theta_s$  a vectorului sumă

$$\theta_s := \arctan 2 \frac{D.Y - A.Y}{D.X - A.X} \quad (23)$$

► Se trasează automat, prin software, fazorul cu modulul  $|\overrightarrow{AD}|$  și faza  $\theta_s$ .

Se poate proceda în mod asemănător pentru efectuarea operațiilor de scădere, înmulțire cu un scalar, oglindire sau reprezentare a simetricului în raport cu o axă sau cu originea sistemului de axe.

### EXEMPLE. CONCLUZII

Ca exemplu se prezintă diagrama fazorială a unui transformator monofazat care are în secundar conectată o impedanță de sarcina inductivă (fig.5.).

Modulele și fazele mărimilor care apar în diagramă se obțin din ecuațiile de funcționare scrise pentru schema echivalentă T a transformatorului [15]. În aceste ecuații intervin parametrii de circuit ai transformatorului (rezistențe, reactanțe, impedanțe) și ai sarcinii, tensiunea de alimentare, tensiunile electromotoare induse, curenții din circuitul primar și secundar.

În cadrul aplicației, fazorii și vectorii sunt “săgeți” desenată cu ajutorul unei proceduri numită Vector().

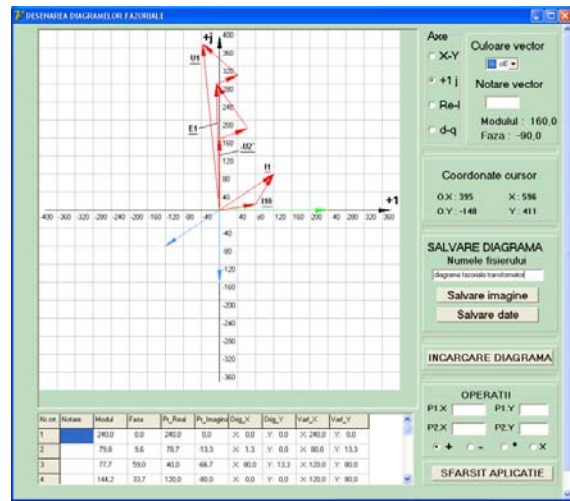


Figura 5. Exemplu de utilizare a editorului grafic

Procedura Vector ( $ACanvas: TCanvas; A, B: TPoint; Color: TColor$ );

în care:

- $ACanvas$  este un tablou al unei componente de tip *Image* (de exemplu);
- A și B sunt originea și extremitatea vectorului, două puncte reprezentate în plan cu componente .X și .Y;
- $Color$  este culoarea cu care se va reprezenta fazorul.

Direcția și modulul “săgeții” sunt determinate din coordonatele punctelor A și B, iar vârful este desenat ca un poligon definit de trei puncte (triunghi).

Dacă se notează cu VA, VB și VC cele trei puncte atunci coordonatele lor pot să fie:

$$VA.X := O.X + Round(IW/2) \quad (40)$$

$$VA.Y := O.Y \quad (41)$$

VB.X, VB.Y, VC.X și VC.Y pot să fie exprimate simplu în funcție de VA.X și VA.Y.

Echipamentele moderne utilizate pentru măsurări și monitorizări în sisteme electrice care includ ansambluri de traductoare pentru mărimi electrice și minicontrolere [16] sau procesoare de semnale digitale [17] utilizate ca dispozitive electronice inteligente (DEI).

Ansamblul traductor-DEI, căruia i se alocă o adresă sau etichetă de identificare, poate fi integrat într-o minirețea de traductoare administrată de un dispozitiv master, care poate fi un calculator personal sau un laptop (fig. 6.). Comunicarea între master și DEI se realizează prin magistrala RS 485 folosind un protocol consacrat sau unul dezvoltat de utilizator.

Traductoarele au rolul de a transforma mărimi alternative măsurate (curenți, tensiuni) în semnale de măsurare standard. DEI convertesc analog-numeric, simultan, aceste semnale și colectează serii de eșantioane pe durata uneia sau mai multor perioade.

Transferul seriilor de date de la DEI la master se realizează la cererea și sub controlul dispozitivului master.



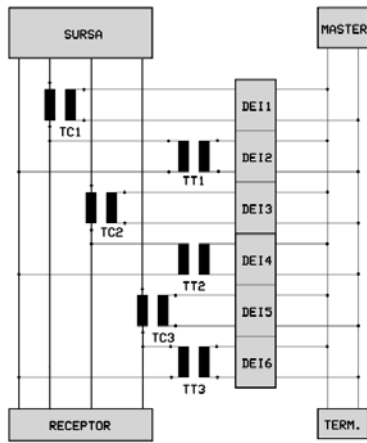


Figura 6. Schema bloc a unui sistem de monitorizare a mărimilor sinusoidale trifazate

Acesta prelucrează, în timp real, seriile de date pentru a determina valorile efective și defazajele dintre mărimile de fază. Aceste rezultate constituie mărimi de intrare pentru editorul grafic descris anterior. Considerând una din mărimile unei faze (de obicei tensiunea) ca referință, toate celelalte cinci pot fi reprezentate prin fazori în planul componentei Image de pe panoul editorului.

Reprezentarea tuturor celor șase fazori se poate face cu o perioadă de repetiție care să permită efectuarea tuturor operațiilor necesare. În acest fel, editorul grafic poate să fie utilizat pentru trasarea fazorilor mărimilor de linie și pentru măsurări în mod grafic folosind procedura descrisă în unul din capitolele anterioare. Pornind de la această descriere simplă, aria aplicațiilor editorului grafic se poate extinde la analiza, în mod grafic, a evoluției mărimilor măsurate în circuitele primare și secundare ale unui transformator trifazat [18]. Aceasta presupune însă cunoașterea parametrilor de circuit ai transformatorului, completarea hardware-lui cu elementele necesare efectuării măsurărilor și adaptarea software-lui la noua structură.

Utilizând controalele tip CheckBox sau RadioGroup pe panoul de control al editorului se poate opta pentru reprezentarea mărimilor achiziționate de la una, două sau trei faze. Aceleași elemente pot fi utilizate și pentru emiterea de cereri de date către DEI vizate la un moment dat de către administratorul minirețelei de tractoare.

Editorul grafic prezentat nu ridică dificultăți de utilizare. A fost prezentat unor potențiali utilizatori cu pregătire medie și superioară și a fost estimat gradul de receptare informațiilor referitoare la funcțiile, modul de utilizare și interpretarea datelor prezentate pe panoul de comandă. Concluzia a fost că impactul asupra utilizatorului este rapid, în special asupra utilizatorilor editoarelor grafice cunoscute: Paint, Pixia, PhotoShop, etc.

## REFERINȚE

1. Glenn Vinnicombe – Impulse responses, step responses and transfer functions. <http://www-control.eng.cam.ac.uk/gv/p6/Handout2.pdf>.

6. \*\*\*, Step response of Second-Order Systems. [http://dunsys.uml.edu/tutorials/2nd\\_Order\\_Systems/Step\\_Resp/Second\\_Order\\_Step\\_Response\\_011705.pdf](http://dunsys.uml.edu/tutorials/2nd_Order_Systems/Step_Resp/Second_Order_Step_Response_011705.pdf).
7. Mustafa M Aziz - System Response. [http://people.exeter.ac.uk/mmaziz/ecm2105/ecm2105\\_n4.pdf](http://people.exeter.ac.uk/mmaziz/ecm2105/ecm2105_n4.pdf)
8. R.W.Erikson – Bode Diagrams of Transfer Functions and Impedances. <http://citeseer.ist.psu.edu/viewdoc/download?doi=10.1.1.112.281&rep=rep1&type=pdf>.
9. \*\*\* - Module 7.2: Bode Diagrams.
10. <http://www.see.ed.co.uk/~jwp/control06/controlcourse/course/advanced/module7-2.html>.
11. \*\*\*, Phasors and complex numbers in AC.
12. <http://www.usna.edu/users/cs/vincent/supnotes/EE301Topic18.pdf>.
13. Andrew E. Yagle – Complex Numbers and Phasors. EECS 206, phasor.pdf.
14. \*\*\*, Circuite electrice in curent alternativ sinusoidal. [www.ubm.ro/sites/cee/images/stories/download/erdeiz/Curs\\_6\\_mine.pdf](http://www.ubm.ro/sites/cee/images/stories/download/erdeiz/Curs_6_mine.pdf).
15. Alexandru Baloi, Adrian Pana, A Virtual Measurement Instrument for Three Phase Electrical Networks Analysis, <http://www.wseas.us-e-library/conferences/2011/Corfu/EDUC/EDUC-29.pdf>.
16. M. Usama Sardar, Synchronous Generator Simulation Using LabView, <http://waset.org/publications/11069/synchronous-generator-simulation-using-labview.pdf>.
17. F. Benhamida, A. Ayad, A. Bendaoued, A. Betaallah, A Basic Power System Analysis by Using LabView, [http://ie.utcluj.ro/Contents\\_Acta\\_ET/2011/Number1/Paper07\\_Benhamida.pdf](http://ie.utcluj.ro/Contents_Acta_ET/2011/Number1/Paper07_Benhamida.pdf)
18. \*\*\*, <http://www.c4s.utcluj.ro/Publicatii/Cursuri/Cursuri%20nr%201.pdf>.
19. \*\*\*, [www.ubm.ro/sites/cee/images/stories/download/./Curs\\_6\\_mine.pdf](http://www.ubm.ro/sites/cee/images/stories/download/./Curs_6_mine.pdf).
20. \*\*\*, DelphiLanguageGuide, <http://www.win.tue.nl/~wstomv/edu/delphi/DelphiLanguageGuide.pdf>
21. \*\*\* Raportarea secundarului la primar - schema echivalenta – transformator electric. <http://www.scrigroup.com/tehnologie/electronica-electricitate/Raportarea-secundarului-la-pri63839.php>
22. M. Popescu, C. Cărămidă, Low Cost System for Waveform Continuous Signal Analysis, Conferinta Națională de Acționări Electrice, Craiova, 7-8 octombrie 2010, Sisteme de achiziție și monitorizare, ISSN: 1842-4805
23. \*\*\* - Norma 4000-5000. Power Analyzer. Operators Manual.
24. Valentin Dogaru Ulieru, Electrical Power Transformer-Modeling, Simulation, and Data Acquisiton, <http://www.wseas.us-e-library/conferences/2008/crete/circuits/circuits11.pdf>