

CONTRIBUȚII LA MONITORIZAREA CALITĂȚII ENERGIEI ELECTRICE

Teză destinată obținerii
titlului științific de doctor inginer
la
Universitatea "Politehnica" din Timișoara
în domeniul INGINERIE ELECTRONICĂ ȘI
TELECOMUNICAȚII
de către

Ing. Ciprian Ovidiu Miron DUGHIR

Conducător științific: prof.univ.dr.ing. Alimpie Ignea
Referenți științifici: prof.univ.dr.ing. Radu Munteanu
prof.univ.dr.ing. Cornelia Gordan
prof.univ.dr.ing. Viorel Popescu

Ziua susținerii tezei: 28.10.2010

Seriile Teze de doctorat ale UPT sunt:

- | | |
|------------------------|---|
| 1. Automatică | 7. Inginerie Electronică și Telecomunicații |
| 2. Chimie | 8. Inginerie Industrială |
| 3. Energetică | 9. Inginerie Mecanică |
| 4. Ingineria Chimică | 10. Știința Calculatoarelor |
| 5. Inginerie Civilă | 11. Știința și Ingineria Materialelor |
| 6. Inginerie Electrică | |

Universitatea „Politehnica” din Timișoara a inițiat seriile de mai sus în scopul diseminării expertizei, cunoștințelor și rezultatelor cercetărilor întreprinse în cadrul școlii doctorale a universității. Seriile conțin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susținute în universitate începând cu 1 octombrie 2006.

Copyright © Editura Politehnica – Timișoara, 2010

Această publicație este supusă prevederilor legii dreptului de autor. Multiplicarea acestei publicații, în mod integral sau în parte, traducerea, tipărirea, reutilizarea ilustrațiilor, expunerea, radiodifuzarea, reproducerea pe microfilme sau în orice altă formă este permisă numai cu respectarea prevederilor Legii române a dreptului de autor în vigoare și permisiunea pentru utilizare obținută în scris din partea Universității „Politehnica” din Timișoara. Toate încălcările acestor drepturi vor fi penalizate potrivit Legii române a drepturilor de autor.

România, 300159 Timișoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Cuvânt înainte

Prezenta lucrare a fost elaborată pe parcursul programului de doctorat fără frecvență susținut în Departamentul de Măsurări și Electronică Optică al Facultății de Electronică și Telecomunicații din cadrul Universității „Politehnica” din Timișoara. Perioada desfășurării programului doctoral a fost noiembrie 2000 – octombrie 2010.

Teza dezvoltă o cercetare în domeniul monitorizării calității energiei electrice. Detectarea și clasificarea perturbațiilor care apar în rețelele de distribuție a energiei electrice reprezintă o problemă de interes pe plan mondial. Detectarea perturbațiilor utilizând algoritmi de analiză în domeniul timp reprezintă o soluție modernă și care necesită relativ puține resurse de calcul.

Programul de doctorat s-a desfășurat sub conducerea domnului Prof. Dr. Ing. Alimpie Ignea. În calitate de conducător de doctorat, domnul profesor mi-a oferit cu profesionalism sprijin pentru a finaliza lucrarea. Am beneficiat de o îndrumare deosebită în ceea ce privește selectarea literaturii de specialitate adecvată, realizarea referatelor din cadrul doctoratului, scrierea și publicarea lucrărilor științifice și parcurgerea etapelor de cercetare care au dus la atingerea scopurilor propuse. Pentru toate acestea îi aduc calde mulțumiri.

Sincere mulțumiri aduc membrilor comisiei pentru modul competent de parcurgere a tezei de doctorat și pentru întocmirea referatelor.

Doresc, de asemenea, să aduc mulțumiri domnului Prof. Dr. Ing. Traian Jurca, ale cărui sfaturi au contribuit la soluționarea unora dintre problemele apărute pe parcursul elaborării tezei de doctorat.

Sunt recunoscător domnilor Prof. Dr. Ing. Dan Stoiciu și Prof. Dr. Ing. Franz Quint pentru ajutorul neprețuit în obținerea unei burse de studii Erasmus. De asemenea, aduc mulțumiri agenției ERASMUS pentru finanțarea stagiului de documentare la Universitatea de Științe Aplicate din Karlsruhe.

Adresez mulțumiri firmei Texas Instruments pentru sprijinul material prompt și dezinteresat, fără de care partea hardware a tezei nu putea fi realizată practic.

Mulțumesc conducerii Facultății de Electronică și Telecomunicații pentru facilitarea deplasării la conferințe, în vederea susținerii unor lucrări științifice. Le sunt recunoscător colegilor din Departamentul Măsurări și Electronică Optică pentru atmosfera de lucru plăcută și sprijinul primit pe parcursul elaborării tezei.

Nu în ultimul rând, recunoștința mea se îndreaptă spre familia mea care a dat dovadă de înțelegere, suport și dragoste în timpul desfășurării programului doctoral.

Timișoara, octombrie 2010

Ing. Ciprian Ovidiu Miron DUGHIR

Dughir, Ciprian Ovidiu Miron

Contribuții la monitorizarea calității energiei electrice

Teze de doctorat ale UPT, Seria 7, Nr. 27, Editura Politehnica, 2010, 182 pagini, 101 figuri, 17 tabele.

ISSN: 1842-7014

ISBN (10): 606-554-181-8; ISBN (13): 978-606-554-181-8

Cuvinte cheie:

Monitorizarea calității energiei electrice, perturbații electrice, algoritmi pentru detectarea perturbațiilor din rețelele electrice, circuite de compresie dinamică, divizor secvențial de tensiune, algoritmi de prelucrare a semnalelor în domeniul timp, server web.

Rezumat:

Lucrarea abordează problema monitorizării calității energiei electrice în rețelele electrice de joasă tensiune. Prezența perturbațiilor în rețelele electrice afectează o serie întreagă de consumatori (atât casnici cât și industriali), perturbând buna funcționare a acestora și putând conduce, în cazurile cele mai defavorabile, la defectarea acestora. Prin îmbunătățirea unor algoritmi prezentați în literatura de specialitate dar și prin dezvoltarea unor algoritmi noi de detectare a perturbațiilor (bazați pe analiza semnalelor în domeniul timp), s-a reușit implementarea unui sistem (hardware și software) care detectează existența perturbațiilor din rețelele de distribuție a energiei electrice. Prin utilizarea unui divizor secvențial de tensiune, se pot prelua tensiuni electrice cu o dinamică mare (amplitudini de până la 10 kV), fără a pierde detaliile fine din jurul valorii nominale a tensiunii rețelei. Aplicația dezvoltată este extrem de avantajoasă din punct de vedere al prețului, al ușurinței de utilizare și a acurateței detectării perturbațiilor.

Cuprins

Lista de figuri	7
Lista de tabele	10
1. Introducere	11
1.1 Problematika tezei	11
1.2 Structura tezei	12
2. Tipuri de perturbații și standarde pentru detectarea și monitorizarea acestora ...	14
2.1 Caracterizarea surselor de perturbații	14
2.2 Standarde privind asigurarea calității energiei electrice.....	17
2.2.1 Istoric.....	17
2.2.2 Standardul IEC 61000-4-30	21
2.2.3 Standardul EN 50160.....	27
2.3 Concluzii	29
3. Circuite de condiționare a semnalelor	30
3.1 Introducere.....	30
3.2 Circuit de condiționare cu transformator funcțional	31
3.2.1 Prezentarea circuitului.....	31
3.2.2 Proiectarea circuitului de compresie cu transformator funcțional	32
3.2.3 Simularea funcționării transformatorului funcțional.....	35
3.2.4 Algoritm pentru reducerea erorilor circuitului de precondiționare cu transformator funcțional	44
3.3 Circuit de compresie dinamică folosind trei canale ale convertorului analog-numeric (divizor secvențial de tensiune)	48
3.3.1 Prezentarea circuitului.....	48
3.3.2 Analiza funcționării circuitului în Matlab	52
3.3.3 Aplicații ale divizorului secvențial de tensiune în detectarea perturbațiilor tranzitorii cu amplitudini mari.....	54
3.4 Concluzii	56
4. Sisteme de achiziție de date	58
4.1 Introducere.....	58
4.2 Placa de achiziție NI USB 6259	59
4.3 Biblioteca de funcții NI-DAQmx	61
4.4 Concluzii	65
5. Algoritmi software pentru detectarea perturbațiilor din rețelele electrice de joasă tensiune	66
5.1 Introducere.....	66
5.2 Determinarea apariției perturbațiilor tranzitorii.....	66
5.3 Îmbunătățirea metodei de detectare a perturbațiilor tranzitorii	71
5.4 Concluzii	76
6. Sistem de detectare și monitorizare a perturbațiilor din rețelele electrice	77
6.1 Introducere.....	77
6.2 Prezentarea sistemului hardware.....	77
6.3 Aplicație software pentru detectarea perturbațiilor	84
6.3.1 Descrierea aplicației.....	84
6.3.2 Inițializarea aplicației	89
6.3.3 Memoria circulară.....	92
6.3.4 Detectarea și monitorizarea perturbațiilor	93
6.3.5 Interfața cu utilizatorul.....	98

6.3.6 Configurarea aplicației	99
6.3.7 Testarea aplicației software.....	101
6.4 Server web pentru accesarea on-line a datelor înregistrate	105
6.5 Concluzii	109
7. Rezultate experimentale.....	111
8. Concluzii și contribuții	120
Bibliografie	125
Anexa 1. Programe reprezentative elaborate în PSpice	131
Anexa 2. Programe reprezentative elaborate în Matlab	137
Anexa 2.1 Program Matlab pentru verificarea algoritmului de calibrare a sistemului de condiționare a semnalelor cu transformator funcțional.....	137
Anexa 2.2 Program MatLab pentru trasarea caracteristicii de transfer a divizorului secvențial de tensiune	143
Anexa 3. Program Visual C++ pentru monitorizarea calității energiei electrice.....	145
Anexa 3.1 Program pentru inițializarea aplicației și achiziția de date	145
Anexa 3.2 Program pentru detecția perturbațiilor din rețeaua electrică.....	156
Anexa 3.3 Program pentru implementarea memoriei circulare.....	164
Anexa 3.4 Program pentru implementarea controlului grafic care afișează starea memoriei circulare	172
Anexa 4. Programe reprezentative elaborate în HTML și PHP	176

Lista de figuri

Fig. 1.1 Tipuri de perturbații	17
Fig. 3.1 Schema bloc a unui sistem de monitorizare a perturbațiilor din rețeaua electrică de joasă tensiune	30
Fig. 3.2 Transformator funcțional cu amplificator operațional	31
Fig. 3.3 Caracteristica de transfer a transformatorului funcțional în cadranul I....	32
Fig. 3.4 Circuitul echivalent pentru calculul valorii rezistenței R_1	35
Fig. 3.5 Funcția de transfer a circuitului simulată în PSpice	36
Fig. 3.6 Abaterea de la caracteristica reală la 500 V	36
Fig. 3.7 Abaterea de la caracteristica reală la 2 kV	37
Fig. 3.8 Influența temperaturii asupra punctelor de frângere ale caracteristicii circuitului de compresie dinamică a) la 500 V; b) la 2 kV; c) la 10 kV ...	38
Fig. 3.9 Semnalul de intrare triunghiular aplicat circuitului și semnalul de la ieșirea circuitului	39
Fig. 3.10 Semnalul de intrare triunghiular aplicat circuitului și semnalul reconstruit de aplicația software	39
Fig. 3.11 Spectrele semnalelor de intrare, respectiv de ieșire ale circuitului de compresie	40
Fig. 3.12 Semnal biexponențial aplicat la intrare și semnalul de la ieșirea circuitului de compresie	40
Fig. 3.13 a) Semnal biexponențial aplicat la intrarea circuitului de compresie și semnalul de la ieșirea circuitului de decompresie, b) eroarea relativă a semnalului de la ieșirea circuitului de decompresie față de semnalul aplicat la intrarea circuitului de compresie	41
Fig. 3.14 Spectrul semnalului de intrare biexponențial și al semnalului de ieșire după decompresie	41
Fig. 3.15 Semnal de intrare sinus amortizat și semnalul de la ieșirea circuitului de compresie (linie punctată)	42
Fig. 3.16 Semnal de intrare sinus amortizat și semnalul de la ieșirea circuitului de decompresie (linie punctată)	42
Fig. 3.17 Eroarea relativă a amplitudinii maxime a sinusului amortizat după decompresie	43
Fig. 3.18 Spectrul semnalului sinus amortizat de la intrare și după decompresie ..	43
Fig. 3.19 Schema bloc a circuitului propus pentru minimizarea erorilor prin calibrare	44
Fig. 3.20 Semnalul de intrare și semnalul decomprimat la 1.9 kV a) fără corecție; b) cu corecție.....	45
Fig. 3.21 Semnalul de intrare și semnalul decomprimat la 3 kV a) fără corecție; b) cu corecție.....	45
Fig. 3.22 Semnalul de intrare și semnalul decomprimat la 7 kV a) fără corecție; b) cu corecție.....	46
Fig. 3.23 Semnalul de intrare și semnalul decomprimat la 10 kV a) fără corecție; b) cu corecție.....	46
Fig. 3.24 Schema bloc a circuitului de compresie dinamică propus	48
Fig. 3.25 Funcția de transfer a celor trei divizoare de tensiune utilizate	49
Fig. 3.26 Funcția de transfer a divizorului de tensiune secvențial.....	49

Fig. 3.27 Algoritmul software al divizorului de tensiune secvențial	50
Fig. 3.28 Algoritmul de decompresie a semnalului	51
Fig. 3.29 a) Funcția de transfer a divizorului de tensiune secvențial simulată în Matlab; b) Funcția de transfer a divizorului împreună cu circuitul de decompresie ...	52
Fig. 3.30 Perturbație tranzitorie suprapusă peste semnalul din rețeaua electrică	54
Fig. 3.31 Algoritm pentru detecția perturbațiilor tranzitorii de amplitudine mare ...	55
Fig. 3.32 Aplicație Visual C++ pentru testarea funcționării algoritmului de detecție a perturbațiilor tranzitorii de mare amplitudine bazat pe divizorul secvențial de tensiune	56
Fig. 4.1 Componentele unui sistem de achiziție de date	58
Fig. 4.2 Placa de achiziție NI USB 6259	59
Fig. 4.3 Conectarea intrărilor în modul diferențial	60
Fig. 4.4 Conectarea intrărilor în modul unipolar	61
Fig. 4.5 Aplicația DAQ assistant	62
Fig. 5.1 Schema bloc a sistemului pentru detectarea perturbațiilor tranzitorii ..	67
Fig. 5.2 Tehnica segmentării	68
Fig. 5.3 Ordinograma algoritmului	69
Fig. 5.4 Aplicație software pentru împărțirea pe segmente și calculul indicilor de regularitate pe fiecare segment	70
Fig. 5.5 Schema bloc a sistemului pentru detectarea perturbațiilor tranzitorii modificat	71
Fig. 5.6 Ordinograma algoritmului modificat	72
Fig. 5.7 Variația indicelui de regularitate cu frecvența	73
Fig. 5.8 Semnal sinusoidal cu perioada de 20 ms peste care s-a suprapus o perturbație de tip impuls	74
Fig. 5.9 Semnal sinusoidal cu perioada de 20 ms peste care s-a suprapus o perturbație de tip impuls dublu	74
Fig. 5.10 Semnal sinusoidal cu perioada de 20 ms peste care s-a suprapus o perturbație de tip impuls dublu	75
Fig. 5.11 Semnal sinusoidal cu perioada de 20 ms peste care s-a suprapus un zgomot	75
Fig. 6.1 Diagrama sistemului de monitorizare al perturbațiilor	77
Fig. 6.2 Circuitul de condiționare a semnalelor	78
Fig. 6.3 Modul de intrare izolat	79
Fig. 6.4 Modulul de intrare	79
Fig. 6.5 Diagrama de conectare a modulelor de intrare	80
Fig. 6.6 Circuitul de condiționarea semnalelor	81
Fig. 6.7 Conectarea circuitului de intrare în modul masă comună	82
Fig. 6.8 Conectarea circuitului de intrare în modul diferențial	82
Fig. 6.9 Conectarea a două rețele electrice diferite la circuitul de intrare	83
Fig. 6.10 Conectarea unei rețele electrice trifazice la circuitul de intrare	83
Fig. 6.11 Conectarea unei rețele electrice trifazice la circuitul de intrare în mod diferențial	84
Fig. 6.12 Ierarhia de clase MFC	86
Fig. 6.13 Fereastra principală a aplicației	89
Fig. 6.14 Secvența de inițializare a aplicației	90
Fig. 6.15 Diagrama rutinei soft de detectare a perturbațiilor	91
Fig. 6.16 Organizarea memoriei circulare	92
Fig. 6.17 Afișarea grafică a stării memoriei circulare	92
Fig. 6.18 Algoritmul de detecție a perturbațiilor	93

Fig. 6.19	Diagrame temporale pentru explicarea funcționării sistemului de detecție a perturbațiilor	94
Fig. 6.20	Detecția duratei unei semiperioade a semnalului rețelei	95
Fig. 6.21	Algoritm de detecție a perturbațiilor	96
Fig. 6.22	Blocul de detecție a perturbațiilor	97
Fig. 6.23	Bara de instrumente a aplicației	98
Fig. 6.24	Controale pentru afișarea grafică a semnalelor	99
Fig. 6.25	Structura fișierului settings.ini	100
Fig. 6.26	Aplicația de generare a semnalelor	101
Fig. 6.27	Diagrama aplicației de generare a semnalelor de test	103
Fig. 6.28	Aplicația de generare a semnalelor. Generarea unui semnal biexponențial	103
Fig. 6.29	Impuls biexponențial pentru diferite valori ale parametrilor a și b	104
Fig. 6.30	Aplicația de generare a semnalelor. Generarea unui semnal sinusoidal cu frecvența de 50 Hz peste care se suprapune un impuls biexponențial	104
Fig. 6.31	Aplicația de generare a semnalelor. Generarea unui semnal sinusoidal însoțit de zgomot	105
Fig. 6.32	Conectarea serverului web la Internet	106
Fig. 6.33	Panoul de configurare a serverului web xampp	107
Fig. 6.34	Fereastra Perturbații a aplicației web	108
Fig. 6.35	Sistem de monitorizare a calității energiei electrice	109
Fig. 7.1	Perturbație de tip impuls tranzitoriu	112
Fig. 7.2	Perturbație de tip impuls tranzitoriu. a) semnal perturbat, b) detaliu ..	112
Fig. 7.3	Perturbație de tip impuls tranzitoriu	113
Fig. 7.4	Perturbație de tip impuls tranzitoriu. a) semnal perturbat, b) detaliu ..	113
Fig. 7.5	Perturbație de tip impuls tranzitoriu. a) semnal perturbat, b) detaliu ..	113
Fig. 7.6	Perturbații de tip impuls	114
Fig. 7.7	Perturbații a) cădere de tensiune, b) cădere de tensiune și impuls tranzitoriu	114
Fig. 7.8	Variația valorii maxime a tensiunii rețelei la pornirea unui motor asincron de 18 kVA	115
Fig. 7.9	Semnalul din rețeaua electrică a Facultății de Electronică și Telecomunicații și spectrul acestuia	115
Fig. 7.10	Semnalul din rețeaua electrică a Facultății de Mecanică și spectrul acestuia	116
Fig. 7.11	Semnalul rețelei electrice rurale din Germania și spectrul acestuia	116
Fig. 7.12	Cădere de tensiune a) tensiune, b) evoluția în timp a valorii efective a tensiunii rețelei în timpul golului de tensiune	116
Fig. 7.13	Perturbație de scurtă durată	117
Fig. 7.14	Conectarea sistemului de monitorizare a calității energiei electrice la un sistem fotovoltaic	117
Fig. 7.15	Trecerea alimentării laboratorului de pe sistemul energetic național pe sursa de energie regenerabilă	118
Fig. 7.16	Semnalul rețelei energetice naționale (punctat) și semnalul generat de invertor (linie continuă)	119
Fig. 7.17	Spectrul semnalului generat de invertor a) spectrul semnalului la frecvențe joase; b) spectrul semnalului la frecvențe ridicate (detaliu)	119

Lista de tabele

Tabelul 1.1 Evoluția conceptului de calitate a energiei electrice de-a lungul timpului .	19
Tabelul 1.2 Standarde pentru monitorizarea calității energiei electrice	20
Tabelul 1.3 Condițiile de referință în raport cu tensiunea măsurată	21
Tabelul 1.4 Plaja de funcționare a echipamentelor de clasă A	22
Tabelul 1.5 Plaja de funcționare a echipamentelor de clasă B	22
Tabelul 1.6 Incertitudinea de măsurare	26
Tabelul 1.7 Condiții de referință relative la influențele externe	27
Tabelul 1.8 Domeniul de operare al instrumentelor	27
Tabelul 1.9 Norme EN 50160: Caracteristici ale calității energiei electrice în JT și MT	28
Tabelul 3.1 Erorile relative ale semnalului comprimat și decomprimat	47
Tabelul 3.2 Energiile semnalelor	47
Tabelul 3.3 Valorile și numărul cuantelor de tensiune pentru diferite intervale de valori ale semnalului de intrare	53
Tabelul 3.4 Numărul efectiv de biți utilizați pe fiecare interval al tensiunilor de intrare .	54
Tabelul 4.1 Principalii parametri ai plăcii de achiziție NI USB 6259	60
Tabelul 6.1 Parametrii amplificatorului izolator ISO 122U	78
Tabelul 6.2 Parametrii circuitului integrat DCH010515N7	79
Tabelul 6.3 Funcțiile matematice admise de aplicație și sintaxa acestora	102

1. INTRODUCERE

1.1 Problematika tezei

Cele mai multe domenii ale industriei, dar și de altă natură, se confruntă cu problema calității energiei electrice, în special cele care trebuie să ofere un standard ridicat al serviciilor (internet, comunicații de date, fabrici de componente semiconductoare etc.). Calitatea energiei electrice poate fi definită în multe moduri, fiecare individ având propria sa percepție asupra acestei noțiuni, motiv pentru care a fost nevoie de definirea clară și concisă a noțiunii de calitate a energiei electrice și de standardizarea parametrilor care intervin în definirea acestei noțiuni. Tensiunea stabilă, forma de undă nedistorsionată (lipsa armonicilor), abateri minime de la frecvența nominală, sunt doar câteva dintre caracteristicile pe care trebuie să le aibă un sistem electric de calitate.

Diagnosticarea promptă a problemelor care apar în ceea ce privește sistemul de distribuție a energiei electrice conduce la corectarea promptă a acestora [1]. Folosirea unor tehnici de procesare avansată a semnalelor și monitorizarea în timp real a perturbațiilor facilitează depanarea acestor sisteme și evitarea evenimentelor neprevăzute. Monitorizarea liniilor de alimentare cu energie electrică poate conduce la adoptarea de măsuri fiabile pentru prevenirea producerii de perturbații. Sistemele de măsurare digitale pot măsura tensiune, curent, dezechilibre de fază, flux magnetic, temperatură cu o acuratețe foarte mare. Se pot amplasa un număr mare de astfel de sisteme de monitorizare în diverse puncte ale sistemului energetic și astfel, se poate realiza monitorizarea acestuia prin intermediul unei rețele Ethernet [2], [3], [4], [5], [6]. Instrumente software puternice stau la dispoziția utilizatorului permițând crearea de jurnale cu datele măsurate, analiza armonică a semnalelor achiziționate în diferite puncte ale sistemului și activarea unor alarme atunci când parametrii depășesc limitele normale [7], [8]. Mai mult, măsurările multiple, în diferite puncte ale rețelei, sincronizate temporal prin GPS, permit o analiză complexă și o localizare mai exactă a surselor [9].

Circuitele de monitorizare prelucrează cantități imense de date, aceste date provenind din diferitele puncte ale sistemului supus monitorizării. Sunt utilizate tehnici speciale de prelucrare a semnalelor pentru a extrage informațiile utile din acest ocean uriaș de date [10]. Procedeele de prelucrare digitală a semnalelor joacă un rol esențial în cadrul algoritmilor utilizați, noi tehnici matematice sunt folosite pentru a îmbunătăți calitatea sistemelor de monitorizare [11].

Odată detectată și clasificată perturbația, este necesar să se cunoască sursa acesteia pentru a se putea lua măsuri în vederea înlăturării ei. Defectarea anumitor componente, funcționarea parțială a unora dintre ele, introducerea sau scoaterea din rețea a unor echipamente electrice de putere sau variații mari ale sarcinii sunt cauze des întâlnite. După ce informațiile au fost clasificate, a fost evaluată severitatea problemei și identificată sursa acesteia, înlăturarea ei poate fi făcută de un sistem de decizie automat sau manual. De obicei, dacă defecțiunea este în amonte, problema trebuie semnalată distribuitorului de energie electrică. Dacă sursa

perturbației este la consumator, operatorii pot ignora defectiunea sau pot repara echipamentul defect.

Foarte multe țări au adoptat standarde pentru asigurarea calității energiei electrice [12], [13], [14], [15], [16], [17].

România a adoptat în 2004, odată cu ratificarea Acordului de Asociere a României la Uniunea Europeană, directiva 89/336 EEC – Compatibilitate electromagnetică [18], directivă prin care Ministerul Economiei și Comerțului a fost abilitat printre altele să emită ordine privind lista standardelor aplicabile, listă ce trebuie actualizată periodic [19]. În acest sens, în cursul anului 2003 ASRO a preluat în întregime standardele din domeniul compatibilității electromagnetice existente la acel moment și publicate în Jurnalul Oficial al Uniunii Europene. Noua listă prezentată o înlocuiește pe cea publicată în Monitorul Oficial 62/230/2002 în conformitate cu Ordinul comun MCTI 555/2001/MIR 409/2001. Experiența dobândită în aplicarea Directivei CEM 89/336/CE, limitele de aplicare, precum și evoluția domeniului electric, în special electronica și telecomunicațiile, au condus la apariția unei noi directive CEM și anume: DIRECTIVA 2004/108/CE (DIRECTIVE 2004/108/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 15 December 2004). Această directivă a fost preluată și adoptată în legislația României prin intermediul Hotărârii Guvernului nr. 982/2007 privind compatibilitatea electromagnetică, fiind aplicată începând cu data de 20 iulie 2007.

Prin ordinul ANRDE nr. 128 din 2008 este aprobat Codul tehnic al rețelelor electrice de distribuție.

Odată cu introducerea standardizării și în țara noastră, mulți cercetători români și-au îndreptat atenția înspre acest domeniu [20], [21], [22], [23].

1.2 Structura tezei

Lucrarea cuprinde 8 capitole structurate după cum urmează.

Capitolul 1 prezintă problematica și structura tezei, trecându-se în revistă principalele probleme ce vor fi tratate pe larg în capitolele următoare.

În capitolul 2 este prezentată o clasificare succintă a perturbațiilor care apar în rețelele de alimentare cu energie electrică. Se prezintă evoluția noțiunii de calitate a energiei electrice de-a lungul timpului și sunt prezentate principalele standarde pentru asigurarea calității energiei electrice, cu descrierea mai amănunțită a două dintre ele: standardul IEC 61000-4-30 și standardul EN 50160.

Capitolul 3, intitulat *Circuite de condiționare a semnalelor*, prezintă structura unui sistem de monitorizare a calității energiei electrice și este orientat asupra circuitului de condiționare a semnalelor, circuit care are rolul de a prelua semnalele din rețeaua electrică și a le modifica nivelul astfel încât să fie compatibil cu intrările convertoarelor analog-numeric folosite pentru preluarea informației în formă numerică. Tot aici este prezentată problematica achiziționării semnalelor cu o dinamică foarte mare, soluția problemei constând în utilizarea unui circuit de compresie pentru a micșora gama dinamică a semnalului de la intrarea convertorului analog-numeric, astfel încât să poată fi înregistrate atât semnale cu amplitudini mari (10 kV) cât și variații mici (de ordinul volților) în jurul valorii nominale a tensiunii rețelei. Din categoria circuitelor de compresie "clasice" este prezentat un circuit de condiționare a semnalelor realizat cu ajutorul unui transformator funcțional, circuit care prezintă o serie de probleme. Autorul propune o soluție pentru rezolvarea acestor probleme, bazată în principal pe introducerea unui modul de calibrare în componența circuitului. Deoarece o soluție care implică circuite integrate este în

general mai scumpă și nu este întotdeauna fiabilă, autorul propune utilizarea unui circuit de compresie a semnalului mult mai simplu, utilizând trei divizoare rezistive de tensiune și procesarea software a semnalelor achiziționate pentru obținerea unei caracteristici asemănătoare unui circuit de compresie, aceasta fiind una dintre contribuțiile originale ale autorului.

În Capitolul 4 sunt prezentate sistemele de achiziție de date, modul de realizare și funcționare al acestora. Este prezentată placa de achiziție utilizată și sunt descrise principalele funcții din biblioteca NI-DAQmx care însoțește placa de achiziție, funcții indispensabile în dezvoltarea aplicației.

În capitolul 5 sunt prezentați câțiva algoritmi originali pentru detectarea perturbațiilor din rețelele electrice bazați pe analiza semnalelor în domeniul timp. S-a ales analiza în domeniul timp, deoarece analiza semnalelor cu ajutorul transformatei Fourier sau Wavelet este consumatoare de timp și de resurse, autorul dorind implementarea sistemului de monitorizare a calității energiei electrice pe sisteme ieftine și cu performanțe modeste din punct de vedere al puterii de calcul. De asemenea, în capitolul 5 este prezentat un algoritm de detectare a perturbațiilor bazat pe media nulă pe o perioadă a unui semnal sinusoidal, algoritm preluat din literatură și îmbunătățit de autor prin introducerea unei memorii circulare, putându-se stoca astfel un număr predefinit de eșantioane ale semnalului. Pornind de la această idee, autorul propune în capitolul 6 o metodă originală de detectare a perturbațiilor bazată pe analiza unei semiperioade a semnalului rețelei. Metoda de detectare se bazează, în esență, pe compararea valorilor instantanee obținute pe o semiperioadă a semnalului rețelei (valoare efectivă, valoare de vârf, factor de creastă și frecvență) cu mediile lor pe 10 perioade. În urma comparării și interpretării valorilor obținute se ia decizia dacă semnalul este sau nu perturbat. În acest capitol este prezentată și aplicația software care implementează algoritmul de detectare a perturbațiilor. Pentru testarea acestui algoritm a fost elaborată o aplicație software care generează pe placa de sunet a calculatorului un semnal de 50 Hz având suprapuse diverse tipuri de perturbații. Sistemul de monitorizare poate fi plasat chiar și în locuri mai puțin accesibile, existând de asemenea și cazuri în care se dorește monitorizarea de la distanță a unui site, motiv pentru care a fost implementat un server web ce permite accesul de la distanță la datele măsurate.

Capitolul 7 prezintă *Rezultatele experimentale* obținute cu sistemul de monitorizare a calității energiei electrice propus și realizat de autor, având ca surse diferite locații ale rețelei de alimentare de joasă tensiune.

Capitolul 8 prezintă cele mai importante concluzii generale și contribuțiile originale ale autorului în cadrul tezei de doctorat.

În anexe sunt prezentate, din programele soft elaborate de autor, secvențe reprezentative din programele elaborate în C++, PSpice, Matlab, HTML și PHP.

2. TIPURI DE PERTURBAȚII ȘI STANDARDE PENTRU DETECTAREA ȘI MONITORIZAREA ACESTORA

2.1 Caracterizarea surselor de perturbații

Există mai multe criterii pentru caracterizarea perturbațiilor care apar în rețelele de distribuție a energiei electrice [24], criteriile ce caracterizează perturbațiile fie în domeniul timp (durată, amplitudine, repetitivitate), fie în domeniul frecvențe (analiză Fourier sau Wavelet) [25], [26], [27].

Din punctul de vedere al duratei, perturbațiile pot fi:

1. Perturbații tranzitorii

- a. *Impulsurile tranzitorii* sunt de regulă unidirecționale ca și polaritate [28], [29].
- b. *Oscilațiile tranzitorii* constau în variații rapide ale tensiunii sau curentului cu schimbări bruște de polaritate. Cauzele posibile ale perturbațiilor tranzitorii de tip impuls sunt: fulgerele, fire defecte, contacte ale releelor, conectarea sau deconectarea unei sarcini sau activarea dispozitivelor pentru corecția factorului de putere. Cea mai comună cauză a producerii oscilațiilor tranzitorii este introducerea în rețea a unor capacități pentru corecția factorului de putere [30].

2. Perturbații de scurtă durată [31]

- a. *Golurile de tensiune* includ scăderile cu 10 % ale tensiunii nominale pentru o perioadă de timp care nu depășește un minut. Sunt cauzate de regulă de defecte ale sistemului de distribuție a energiei electrice sau echipamente defecte.
- b. *Căderile de tensiune* sunt scăderi ale valorilor tensiunii sau ale curentului cuprinse în intervalul 2 ms – 1 minut. Ele pot fi cauzate de defecte în sistemul de alimentare cu energie electrică (o linie căzută la pământ - defect care va persista până nu va fi înlăturat), pornirea-oprirea unor sarcini foarte mari (motoare), sarcini care își modifică impedanța cu tensiunea de alimentare.
- c. *Supracreșterile de tensiune* sunt creșteri ale valorii efective ale tensiunii sau ale curentului pentru intervale de timp cuprinse între 0.5 cicluri ale frecvenței fundamentale și un minut. Amplitudinea tipică a acestor perturbații este cuprinsă între 110 și 180 % din valoarea tensiunii nominale a rețelei. Cuplarea și decuplarea de la rețeaua electrică a unei sarcini mari sau a unei baterii de condensatoare poate cauza astfel de perturbații.

3. **Perturbațiile de lungă durată** se referă la variații ale valorii efective pe o perioadă de timp ce depășește un minut.
 - a. *Supratensiunile* sunt creșteri ale valorii efective (la frecvența nominală) pentru intervale mai lungi de un minut. Amplitudinile tipice ale acestor perturbații sunt cuprinse între 110 % și 120 % din valoarea tensiunii nominale a rețelei. Printre cauzele care generează astfel de perturbații se numără: decuplarea unor sarcini mari, variații în compensarea reactivă a sistemului sau reglarea defectuoasă a reguletoarelor de tensiune.
 - b. *Subtensiunile* sunt scăderi ale valorii efective nominale (la frecvența nominală) pentru intervale mai lungi de un minut. Amplitudinile tipice ale acestor perturbații sunt cuprinse între 80 % și 90 % din valoarea nominală a tensiunii rețelei. Sunt produse de conectări ale unor sarcini mari, variații ale compensării reactive a sistemului, alegerea neinspirată a unui anumit tip de transformator, reguletoare de tensiune ajustate incorect, supraîncărcarea liniei de transmisie a energiei electrice.
 - c. *Înteruperile susținute de tensiune (golurile de tensiune)* sunt definite ca fiind scăderi la zero a valorii tensiunii de alimentare pentru o durată mai mare de un minut. Sunt provocate (cel mai frecvent) de accidente care afectează liniile de transmisie, transformatoarele sau sursele de tensiune alternativă.
4. **Dezechilibrele** se produc atunci când valoarea efectivă a diferitelor tensiuni de fază sau unghiuri de fază între faze consecutive nu sunt egale [32]. Aceste anomalii sunt considerate severe dacă valoarea lor depășește 5 %. Sunt generate de sarcini dezechilibrate sau de pierderile excesive din conductoarele electrice ale liniei de alimentare cu energie electrică.
5. **Distorsiunile de formă** sunt deviații de la forma de undă ideală (sinusoidală) caracterizate de prezența în spectrul semnalului a componentelor spectrale ale acestor distorsiuni. Există cinci tipuri de astfel de perturbații [33]:
 - a. *Componenta continuă* constă în prezența unei tensiuni suprapuse peste tensiunea liniei de alimentare. Astfel de distorsiuni pot fi cauzate de modificările geometrice ale componentelor generatoarelor (stator, rotor), pornirea și oprirea unor mașini sincrone sau prezența unor surse de alimentare în comutație. Amplitudinea componentei continue este destul de mare pe perioada pornirii - opririi mașinilor sincrone.
 - b. *Distorsiunile armonice* sunt perturbații sinusoidale cu frecvența multiplu întreg al frecvenței sistemului de alimentare cu energie electrică. Orice dispozitiv sau sarcină neliniară produc astfel de perturbații [34], [13]. Consumatorii neliniari pasivi produc, de regulă, armonici impare, armonicele pare fiind produse de către dispozitivele

- active, dar și de către transformatoarele saturate cu un curent continuu. În afara componentelor armonice, în rețea pot să apară și componente nearmonice: subarmonici sau interarmonici produse de convertizoarele de frecvență sau de motoarele asincrone prin fenomenul de alunecare [35].
- c. *Intermodulațiile* sunt perturbații ale tensiunii sau curentului care au frecvențe discrete, diferite de frecvența componentelor armonice, sau un spectru bogat. Sursele principale ale acestor perturbații sunt convertoarele statice de frecvență, ciclo-convertoarele, motoarele de inducție și arcul electric [7].
 - d. *Vârfurile de tensiune* sunt distorsiuni ale tensiunii de alimentare cauzate de operarea normală a dispozitivelor electrice în momentul comutării curentului de la o fază la alta [36], [37].
 - e. *Zgomotul* include orice semnal nedorit cu componente spectrale mai mici de 200 kHz suprapuse peste tensiunea sau curentul din linia de alimentare [38]. Zgomotele pot fi produse de echipamentele care produc arc electric, circuitele de control, surse de alimentare în comutație.
6. **Fluctuațiile de tensiune** (flicker) [39] sunt variații sistematice ale anvelopei semnalului sau o serie de schimbări aleatoare ale tensiunii, magnitudinea acestora nedepășind limita de 105 % din valoarea nominală a tensiunii rețelei. Orice sarcină care prezintă variații semnificative ale curentului, respectiv a valorii elementului reactiv cum ar fi arcurile electrice, poate provoca astfel de fluctuații [40], [41]. Flickerul produce variații ale iluminării în cazul utilizării surselor de iluminat incandescente, efectul fiind foarte deranjant pentru ochi dacă frecvența fluctuațiilor este de 8,8 Hz [42].
7. **Variațiile frecvenței** sunt în directă legătură cu schimbările vitezei generatoarelor rotative datorită dezechilibrului între sarcină și capacitatea respectivului generator. Printre cauze poate fi amintită cuplarea / decuplarea unor sarcini foarte mari. Scăderea frecvenței rețelei are ca efect creșterea pierderilor în transformatoare și creșterea uzurii generatoarelor, ca efect al apropierii de frecvența de rezonanță a acestora.
8. **Evenimentele repetitive** reprezintă o serie de evenimente care se produc la intervale regulate. Printre cauzele apariției acestor perturbații se numără și dispozitivele care produc evenimente repetitive cum ar fi: dispozitive cu viteză variabilă, startere, dispozitive de sudură cu arc.

În figura 1.1 sunt prezentate câteva dintre perturbațiile întâlnite frecvent în rețelele de alimentare cu energie electrică: impulsuri tranzitorii, întreruperi de tensiune, supratensiuni, căderi de tensiune, fluctuații de tensiune (flicker), variații ale frecvenței rețelei.

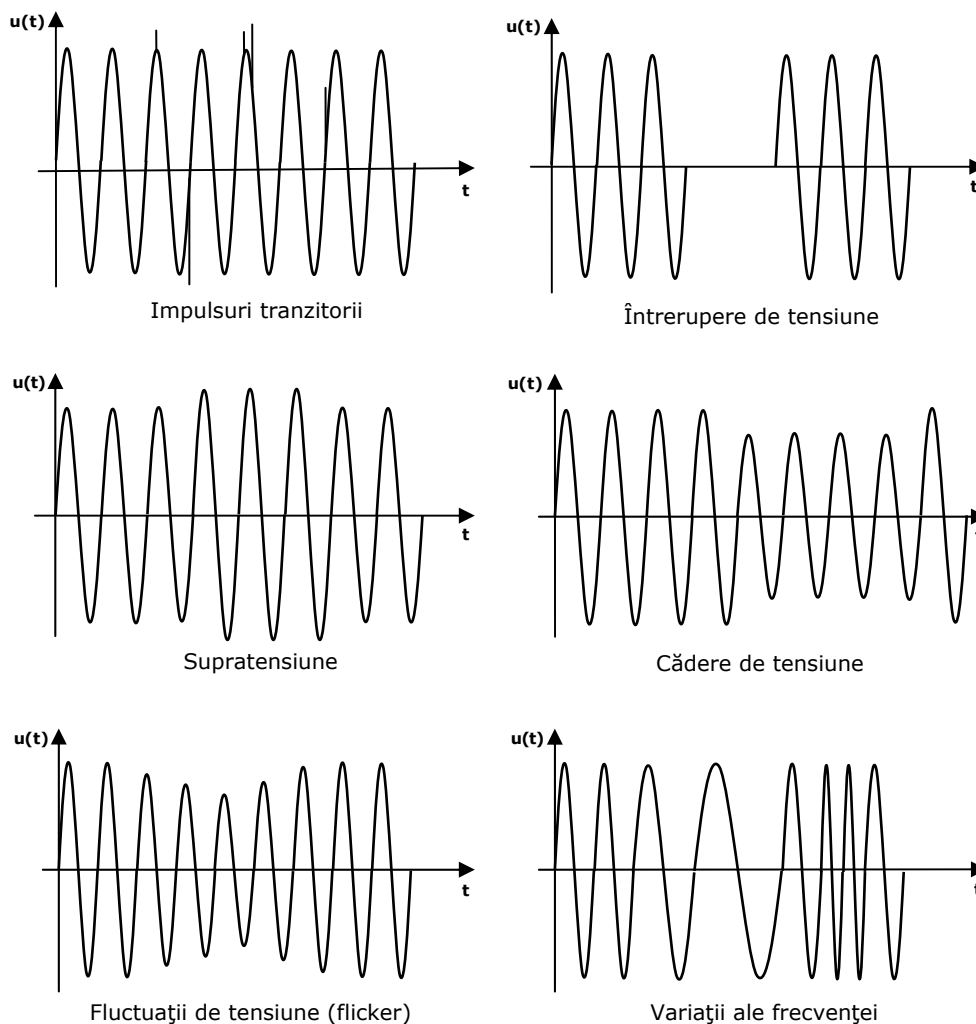


Fig. 1.1 Tipuri de perturbații

2.2 Standarde privind asigurarea calității energiei electrice

2.2.1 Istoric

Încă de la începutul anilor '80 numeroși producători de echipamente electronice au început să producă diverse sisteme electronice pentru monitorizarea calității energiei electrice. În lipsa standardizării procedurilor de măsurare, prelucrare și stocare a datelor, era aproape imposibilă compararea și coroborarea rezultatelor obținute cu diverse tipuri de echipamente.

O primă încercare de standardizare a fost făcută de către Institute of Electrical and Electronics Engineer (IEEE) în 1987, inițiativă ce culminează cu elaborarea și publicarea standardului IEEE 1159 [43].

În 1996, un grup format ad-hoc, alcătuit din specialiști în calitatea energiei electrice din zece țări, au luat în considerare trei cerințe majore în ceea ce privește monitorizarea calității energiei electrice:

- Stabilirea unei ordini în dezordinea existentă la acea dată în ceea ce privește monitorizarea calității energiei electrice;
- Propunerea unei clasificări a nivelurilor de calitate e energiei electrice și definirea așteptărilor consumatorilor;
- Construirea unei punți de legătură între producătorii de energie electrică, utilizatori și constructorii de echipamente.

Se spera ca discuțiile grupului să conducă la dezvoltarea și impunerea de standarde pentru calitatea energiei electrice, scop care a fost atins câțiva ani mai târziu. Un alt grup de studiu a fost înființat în 1997 cu scopul de a defini metodele de măsurare pentru a obține rezultate sigure și repetabile indiferent de instrumentele folosite și indiferent de condițiile de măsurare. După numai doi ani, în 1999 a apărut primul material scris [44], realizat cu ajutorul grupului de lucru IEC (International Electrotechnical Commission) și care prezintă un raport cu progresele înregistrate, identificând aspectele principale ale monitorizării calității energiei electrice, precum și problemele rămase nerezolvate.

Încă de la început, documentele elaborate de IEC cuprindeau două părți: prima parte conținea standardele și normativele care defineau diverși parametri ai perturbațiilor din rețelele electrice, iar cea de-a doua parte cuprindea anexe informative despre când, de ce, unde și cum trebuia monitorizată rețeaua de distribuție a energiei electrice.

Partea normativă din standard cuprinde [45], [46]:

- Definirea scopului, documente de referință și definiții;
- Organizarea măsurărilor (tipuri de instrumente, parametri ce trebuiesc măsurați);
- Acuratețea metodelor de testare, domeniile de operare ale instrumentelor utilizate;
- Metodele specifice de măsurare pentru parametrii rețelelor electrice: frecvență, tensiune, întreruperi de tensiune, supratensiuni, căderi de tensiune, goluri de tensiune etc.;
- Caracteristici tehnice: sunt descrise două clase de instrumente de măsurat cu specificarea limitelor de acuratețe ale fiecărei clase de instrumente în parte.

Partea informativă a standardului cuprinde:

- Aplicații pentru realizarea de măsurări;
- Aplicații pentru diagnosticarea problemelor care pot să apară în timpul exploatarea normală a sistemului de distribuție;
- Aplicații pentru determinări statistice;
- Precauții de instalare;
- Creșteri de curent vs. creșteri de tensiune;
- Bibliografie.

Evoluția conceptului de calitate a energiei electrice de-a lungul timpului este prezentată succint în tabelul 1.1 [47].

Tabelul 1.1 Evoluția conceptului de calitate a energiei electrice de-a lungul timpului

Anul	Sursa	Definiția
1979	T.S. Key, "Diagnosing Power Quality Related Computer Problems" [48]	"În concluzie, răspunsul la întrebarea: ce este calitatea energiei electrice? poate fi găsit doar prin studiul perturbațiilor din liniile electrice. Aceste studii sunt primul pas spre o diagnosticare exactă a problemelor surselor de alimentare ale calculatoarelor."
1985	Clemmensen and Ferraro, "The Emerging Problem of Electric Power Quality" [49]	Calitatea energiei electrice reprezintă lipsa perturbațiilor.
1987	IEC 60050,605-01 -05 International Vocabulary (IEV)	Calitatea sursei de energie electrică: o evaluare a abaterii de la criteriile tehnice, în afara domeniului definit (implicit sau explicit), a sursei de energie electrică sau a unui ansamblu de surse de energie electrică care provin de la mai mulți furnizori.
1988	PCIM Magazine [50]	Calitatea energiei electrice reprezintă nivelul de calitate care trebuie să satisfacă nevoile consumatorilor.
1992 1995	Standardul IEEE 1100-1992 [51] Standardul IEEE 1159-1995 [52]	Calitatea energiei electrice: conceptul alimentării și pământării echipamentelor electronice sensibile într-un mod care este propice bunei funcționări a acestor echipamente.
1995	Standardul IEEE 1250-1995 [53]	Noțiunea de calitate a energiei electrice este foarte des aplicată termenilor ca tensiune, disponibilitate a energiei electrice și chiar armonicilor.
1995	Eskom, Handbook on Quality of Supply [54]	Calitatea energiei electrice: calitatea sursei de energie care afectează distribuția acesteia și totodată include elemente de calitate a tensiunii electrice și măsurări ale disponibilității energiei electrice, precum și a continuității sursei de energie.
1996	Raportul Consiliului Consultativ în privința Compatibilității Electromagnetice al IEC, 1996	Calitatea energiei electrice: un set de parametri ce definesc proprietățile sursei de energie așa cum trebuie ea livrată consumatorilor în condiții normale, în termeni ce definesc continuitatea sursei de energie și caracteristicile tensiunii furnizate de aceasta (amplitudine, frecvență, simetrie, formă de undă).
1999	Standardul IEEE 1100-1999 [55]	Calitatea energiei electrice: conceptul alimentării și pământării echipamentelor electronice sensibile într-un mod care este propice bunei funcționări a acestor echipamente în ipoteza interconectării acestora cu alte echipamente.
2000	Grupul de lucru IEC 77WG09	Calitatea energiei electrice este definită ca un set de parametri tehnici ce descriu compatibilitatea dintre energia furnizată într-o rețea și consumatorii conectați la acea rețea.

În tabelul 1.2 sunt prezentate câteva dintre standardele elaborate de-a lungul timpului pentru monitorizarea calității energiei electrice.

Tabelul 1.2 Standarde pentru monitorizarea calității energiei electrice

Standard	Descriere
IEC 868	Este înlocuit de standardul IEC 61000-4-15
IEEE 519 [56], [57]	Descrie nivelurile de armonici acceptate în punctele de delimitare dintre furnizor și consumator. Este cel mai răspândit standard folosit de furnizorii de energie electrică din SUA;
SR EN 50160	Caracteristici ale tensiunii în rețelele electrice publice;
IEEE 1159 [58]	Este practica recomandată pentru monitorizarea și interpretarea corectă a fenomenelor electromagnetice care cauzează probleme ale calității energiei electrice;
SR EN 61000-2-2	Compatibilitate electromagnetică (CEM) - Partea 2: Mediu înconjurător, Secțiunea 2: Descrierea mediului înconjurător. Niveluri de compatibilitate pentru perturbații de joasă frecvență propagate prin conducție și transmisia de semnale pe rețelele publice de joasă tensiune;
SR EN 61000-2-4	Compatibilitate electromagnetică (CEM) - Partea 2: Mediu înconjurător, Secțiunea 4: Niveluri de compatibilitate în medii industriale pentru perturbații de joasă frecvență propagate prin conducție;
CEI TR 61000-2-8	Compatibilitate electromagnetică (CEM) - Partea 2: Mediu înconjurător, Secțiunea 8: Goluri de tensiune și întreruperi de scurtă durată în rețelele de distribuție a energiei electrice cu rezultate statistice ale măsurărilor;
SR EN 61000-4-7	Descrie o tehnică de măsurare a armonicilor și interarmonicilor astfel încât măsurările efectuate cu aparate diferite pot fi comparate în mod corespunzător;
SR EN 61000-4-15	Standard de măsurare a flickerului, care include specificații ce trebuie respectate de aparatele de măsurare a flickerului (flickermetre), cele mai multe flickermetre bazându-și funcționarea pe acest standard;
SR EN 61000-4-30	Compatibilitate electromagnetică (CEM). Partea 4, Secțiunea 30: Tehnici de încercare și de măsurare. Metode de măsurare a calității energiei electrice;
CEI TR 61000-2-8	Compatibilitate electromagnetică (CEM). Partea 2, Secțiunea 8: Mediu înconjurător – goluri de tensiune și întreruperi de scurtă durată în rețelele de distribuție a energiei electrice;
SR EN 50065	Transmisia semnalelor prin rețele electrice de joasă tensiune în banda de frecvențe de la 3 kHz până la 148,5 kHz. Cerințe generale, benzi de frecvențe și perturbații electromagnetice;
SR CEI 60364-4-44	Instalații electrice în construcții. Partea 4, Secțiunea 44: Protecție pentru asigurarea securității. Protecție împotriva perturbațiilor de tensiune și perturbațiilor electromagnetice;
SR EN 60664-1	Coordonarea izolației echipamentelor din rețelele de joasă tensiune. Partea 1: Principii, prescripții și încercări;
SR HD 472 S1	Tensiuni nominale ale rețelelor electrice de distribuție publică de joasă tensiune.

În continuare se vor prezenta succint standardele IEC 61000-4-30 și EN 50160, standarde ce definesc metodele de măsurare a perturbațiilor din rețelele electrice, precum și specificațiile instrumentelor utilizate.

2.2.2 Standardul IEC 61000-4-30

Standardul IEC 61000-4-30 [59] definește metodele de măsurare ale parametrilor rețelei de alimentare cu energie electrică, astfel încât să se obțină rezultate repetabile și comparabile, indiferent de instrumentele folosite. Acest standard se referă strict la instrumente și metode de măsurare "in-situ" (la fața locului).

Standardul definește două clase de performanță pentru instrumentele de măsurare:

Clasa A – se referă la instrumentele de măsurare pentru evaluarea calității energiei electrice folosite în rețelele companiilor de producție a energiei electrice și în rețelele de distribuție a energiei către clienți.

Clasa B – instrumentele din această clasă pot fi folosite pentru supraveghere generală, diagnostic inițial, sau alte aplicații care nu necesită acuratețe foarte mare. Instrumentele din clasa B produc rezultate mai puțin exacte decât cele din clasa A de performanță.

Producătorii de instrumente de măsurare trebuie să precizeze care dintre parametrii definiți de acest standard sunt măsurați de către instrumentele lor și în ce clasă de exactitate se încadrează instrumentul respectiv.

Standardul definește noțiunea de **tensiune nominală** U_n ca tensiunea cu care un sistem este denumit sau identificat.

Condițiile de referință în raport cu tensiunea măsurată sunt prezentate în tabelul 1.3.

Tabelul 1.3 Condițiile de referință în raport cu tensiunea măsurată

Mărimea	Toleranța permisă
Frecvența	± 50 Hz
Amplitudinea tensiunii	± 1 % din tensiunea nominală
Flicker	10 %
Supratensiuni tranzitorii	0,5 kV
Asimetrii	0,5 %
Armonici (THD)	3 % din tensiunea nominală
Interarmonici (la orice frecvență)	1 % din tensiunea nominală

Aceste condiții de referință sunt valabile atât pentru echipamente ce funcționează în clasă A, cât și pentru cele ce funcționează în clasă B. Ca și tensiune de intrare, tensiunea nominală U_n poate lua valori între 60 V și 600 V.

În tabelul 1.4 este prezentată plaja de funcționare a instrumentelor de măsurare de clasă A.

Tabelul 1.4 Plaja de funcționare a echipamentelor de clasă A

Mărimea de influență	Domeniul
Frecvența	42,5 Hz – 57,5 Hz 51 Hz – 69 Hz
Amplitudinea tensiunii	60 – 180 % din U_n
Flicker	0 – 4 %
Asimetrii	0 – 5 %
Armonici (THD)	De două ori mai mari decât IEC 61000-2-4, tab. 2, 3, 4, 5
Interarmonici (la orice frecvență)	De două ori mai mari decât IEC 61000-2-4, tab. 6
Ripluri ale semnalelor de control	0 – 9 % din U_n
Supratensiuni tranzitorii	6 kV
Perturbații tranzitorii rapide	4 kV

Plaja de funcționare a instrumentelor de măsură de clasă B este prezentată în tabelul 1.5.

Tabelul 1.5 Plaja de funcționare a echipamentelor de clasă B

Mărimea de influență	Domeniul
Frecvența	42,5 Hz – 57,5 Hz 51 Hz – 69 Hz
Amplitudinea tensiunii	80 – 120 % din U_n
Flicker	0 – 4 %
Asimetrii	0 – 5 %
Armonici (THD)	De două ori mai mari decât IEC 61000-2-4, tab. 2, 3, 4, 5
Interarmonici (la orice frecvență)	De două ori mai mari decât IEC 61000-2-4, tab. 6
Ripluri ale semnalelor de control	0 – 9 % din U_n
Supratensiuni tranzitorii	6 kV

Acolo unde sunt necesare măsurări cu o acuratețe ridicată, de exemplu, pentru aplicații contractuale care pot cere să se soluționeze dispute, să se verifice conformitatea cu standardele existente etc., trebuie să fie utilizate instrumente de măsurare realizate să măsoare în conformitate în clasa A de performanță. În acest mod, este asigurată cerința ca orice măsurare a unui parametru efectuată cu două instrumente diferite, care sunt conforme cu cerințele clasei A, atunci când măsoară aceleași semnale, va produce rezultate care corespund, în interiorul intervalului de incertitudine specificat.

În funcție de durata pe care se face măsurarea, sunt definite trei intervale de timp:

- timp foarte scurt: 3 secunde;
- timp scurt: 10 minute;
- timp lung: 2 ore.

Măsurarea frecvenței rețelei

Pentru instrumentele de clasă A, măsurarea frecvenței se face pe un interval de timp de până la 10 secunde și care trebuie să cuprindă un număr întreg de perioade. Frecvența poate fi determinată prin numărarea trecerilor printr-o valoare constantă apropiată de zero într-un interval de 10 secunde. Standardul admite și

alte tehnici de măsurare care dau rezultate echivalente. Instrumentul de măsurat trebuie să afișeze o nouă valoare a frecvenței la fiecare 10 secunde. Intervalele de măsurare trebuie să fie continue și să nu se suprapună.

Pentru instrumentele de clasă B, se permit pauze între intervalele de măsurare. Producătorii trebuie să precizeze pauza maximă dintre două intervale de măsurare.

Instrumentele clasei A trebuie să fie capabile să măsoare frecvența cu o incertitudine $\Delta f = \pm 10$ mHz, iar cele de clasă B cu o incertitudine $\Delta f = \pm 100$ mHz.

Dacă în timpul măsurării frecvenței apare un eveniment de tipul cădere de tensiune, întrerupere de tensiune, supratensiune, frecvența măsurată pe acest interval de timp va fi marcată ca fiind neconformă, iar următoarea valoare conformă ce va fi înregistrată va fi prima valoare a frecvenței obținută în primul interval de 10 secunde de după încetarea perturbației. Prima semiperioadă pozitivă a semnalului de după momentul încetării perturbației va marca începutul unei noi măsurări a frecvenței.

Măsurarea tensiunii rețelei

Instrumentele folosite trebuie să poată măsura valoarea efectivă (rms) a tensiunii rețelei. Valoarea efectivă a tensiunii $u(t)$, măsurată pe un interval T se calculează cu relația:

$$U_{rms} = \sqrt{\frac{1}{T} \int_0^T u^2(t) dt} \quad (2.1)$$

Pentru instrumentele din clasa A, incertitudinea de măsurare trebuie să fie mai bună de $\pm 0,1\%$, iar pentru cele de clasă B incertitudinea de măsurare trebuie să fie mai bună de $\pm 1\%$.

Intervalele de măsurare pentru tensiune, prevăzute de standard, sunt: 200 milisecunde, 3 secunde, 10 minute și 2 ore. Pentru sistemele trifazate, se vor efectua trei măsurări ale valorii efective pentru fiecare interval de măsurare și pe fiecare fază în parte.

Intervalul de timp de 200 milisecunde a fost ales astfel încât să cuprindă un număr întreg de perioade ale semnalului atât în cazul rețelelor cu frecvența de 50 Hz, cât și în cazul celor cu frecvența de 60 Hz. Se obține astfel o fereastră de măsurare ce cuprinde 10 perioade pentru sistemele pe 50 Hz și 12 perioade pentru cele pe 60 Hz. Pe parcursul celor 200 ms, sunt obținute N eșantioane de la convertorul analog-numeric, eșantioane care sunt mediate pe baza relației (2.2) pentru a se obține valoarea efectivă a tensiunii rețelei pe intervalul de 200 ms.

$$u_{rms-200ms} = \sqrt{\frac{\sum_{i=1}^N u^2}{N}} \quad (2.2)$$

Fiecare interval de 200 ms pe care se face măsurarea trebuie să fie continuu și intervalele succesive nu trebuie să se suprapună.

În cazul clasei A de performanță, dacă valoarea efectivă a tensiunii pe 200 ms este mai mare de 150 % sau mai mică de 50 % din valoarea efectivă a tensiunii nominale, intervalul respectiv va fi marcat ca fiind neconform.

Pentru măsurarea valorii efective a tensiunii rețelei pe un interval de 3 secunde se va folosi relația:

$$U_{rms-3-s} = \sqrt{\frac{\sum_{i=1}^{15} U_{rms-200-ms}^2}{15}} \quad (2.3)$$

Dacă opt sau mai multe intervale consecutive de 200 ms sunt marcate ca neconforme, atunci și intervalul de 3 secunde corespunzător este marcat ca fiind neconform. De asemenea, intervalele de 3 secunde trebuie să fie continue și nu trebuie să se suprapună.

În mod similar se pot defini valorile efective ale tensiunii rețelei pe intervale de 10 minute :

$$U_{rms-10-min} = \sqrt{\frac{\sum_{i=1}^{200} U_{rms-3-s}^2}{200}} \quad (2.4)$$

și respectiv 2 ore:

$$U_{rms-2-h} = \sqrt{\frac{\sum_{i=1}^{12} U_{rms-10-min}^2}{12}} \quad (2.5)$$

Pentru clasa B, semnalele aplicate la intrarea instrumentului de măsurare vor fi filtrate cu ajutorul unui filtru trece-jos, astfel încât să fie eliminate armonicile mai mari de ordinul 13.

Măsurarea căderilor și supracreșterilor de tensiune

Pentru măsurarea acestor parametri este nevoie de măsurarea valorii efective a tensiunii rețelei pe fiecare semiperioadă a semnalului, conform relației:

$$u_{rms(\frac{1}{2})} = \sqrt{\frac{\sum_{i=1}^N u^2}{N}} \quad (2.6)$$

unde N reprezintă numărul de eșantioane ale semnalului pe semiperioada pe care se face măsurarea, iar u reprezintă eșantionul curent.

Pentru clasa A de performanță, incertitudinea de măsurare trebuie să fie:

$$\Delta U = \pm 0,2 \% U_n.$$

Pentru clasa B de performanță, incertitudinea de măsurare trebuie specificată de către producător, dar trebuie să fie $\leq 2 \% U_n$.

Se poate introduce un prag x , exprimat în procente din tensiunea nominală, și se pot determina în acest caz supracreșterile de tensiune peste pragul de $x \%$, respectiv căderile de tensiune sub pragul de $x \%$ din valoarea nominală a tensiunii.

Măsurarea întreruperilor de tensiune de scurtă și lungă durată

Măsurarea acestor parametri se face la fel ca și la măsurarea căderilor de tensiune cu un prag de 10 %. În cazul sistemelor monofazate, se consideră întrerupere de tensiune, dacă tensiunea nominală scade sub pragul de 10 %, iar în

cazul sistemelor multifazate se consideră întrerupere de tensiune, dacă tensiunea efectivă a uneia dintre faze scade sub 10 % din tensiunea de referință.

Durata de timp pe care se face observarea (pentru ambele clase) este de două semiperioade ale semnalului rețelei (20 ms pentru 50 Hz și 18 ms pentru 60 Hz).

Măsurarea supratensiunilor tranzitorii

O supratensiune tranzitorie trebuie detectată prin compararea valorii instantanee a tensiunii pe fiecare perioadă cu valoarea instantanee a tensiunii din perioada anterioară, măsurată la același moment de timp față de trecerea prin zero. Dacă există diferențe mai mari de 10 % din U_n , atunci se va incrementa un contor, în caz contrar decrementându-se acest contor, excepție făcând cazul în care acest contor este deja la valoarea zero. Dacă acest contor atinge o valoare egală cu 5% din durata unei perioade a semnalului rețelei, atunci este semnalată prezența unei supratensiuni.

Instrumentele de clasă A trebuie să înregistreze și să prezinte o oscilogramă a supratensiunii.

Instrumentele de clasă B pot prezenta un sumar al supratensiunilor apărute în format text, putând folosi un algoritm aproximativ de detectare a supratensiunilor.

Măsurarea asimetriilor tensiunii rețelei

Instrumentele de clasă A trebuie să măsoare amplitudinea asimetriei tensiunii pe un interval de 10 minute.

Asimetria tensiunii măsurată pe o perioadă de timp T este definită de obicei (folosind metoda componentelor simetrice), prin raportul dintre amplitudinea componentei negative și amplitudinea componentei pozitive, exprimată procentual:

$$u_u = \frac{|V_n|}{|V_d|} \cdot 100\% = \frac{\text{componenta negativă}}{\text{componenta pozitivă}} \cdot 100\% \quad (2.7)$$

Pentru calculul asimetriilor trebuie folosită doar componenta fundamentală. Toate armonicile trebuie eliminate utilizând un filtru sau un algoritm DFT.

Pentru instrumentele de clasă B producătorul trebuie să specifice algoritmi și metodele de determinare utilizate pentru calculul asimetriei.

Pentru clasa A de performanță instrumentele trebuie să aibă o incertitudine de măsurare $\Delta U \leq 0,2 \% U_n$. Pentru indicatoarele de clasă B se impune o incertitudine de măsurare maximă $\Delta U \leq 0,5 \% U_n$.

Măsurarea componentelor armonice

Standardul 61000-4-33 definește noțiunea de frecvență armonică ca fiind un multiplu întreg al frecvenței rețelei electrice. Componenta armonică este valoarea efectivă a oricăreia dintre componentele armonice având o anumită frecvență armonică, în cazul analizei unui regim nesinusoidal prin intermediul Transformatei Fourier Discrete (DFT).

Măsurarea componentelor armonice este o măsurare cvasi-staționară. Pentru măsurarea componentelor armonice, o Transformată Fourier Discretă este aplicată semnalului, pe o fereastră de exact 10 perioade în cazul rețelelor de 50 Hz și 12 perioade la 60 Hz. În ambele cazuri, fereastra de timp este de 200 ms.

Valorile de ieșire ale blocului care realizează Transformata Fourier Discretă trebuie grupate după cum urmează:

$$C_{n-200-ms}^2 = \sum_{k=-1}^1 C_{10n+k}^2 (\text{sisteme pe 50 Hz})$$

$$C_{n-200-ms}^2 = \sum_{k=-1}^1 C_{12n+k}^2 (\text{sisteme pe 60 Hz})$$
(2.8)

În tabelul 1.6 sunt prezentate incertitudinile de măsurare ale instrumentelor folosite pentru măsurarea armonicilor tensiunii rețelei electrice, U_m fiind tensiunea măsurată, iar U_n fiind tensiunea nominală a rețelei.

Tabelul 1.6 Incertitudinea de măsurare

Clasa	Condiții	Eroarea maximă admisă
A	$U_m \geq 1 \% U_n$	5 % U_m
	$U_m < 1 \% U_n$	0,05 % U_m
B	$U_m \geq 3 \% U_n$	5 % U_m
	$U_m < 3 \% U_n$	0,15 % U_m

Măsurarea componentelor interarmonice

Conform standardului, componenta interarmonică reprezintă valoarea efectivă a tensiunii unei linii spectrale cu frecvența între frecvențele armonice, ca rezultat al Transformatei Fourier Discrete aplicată semnalului pe o fereastră de timp dată.

Frecvența componentei interarmonice nu este multiplu întreg al frecvenței rețelei. Un grup de frecvențe interarmonice reprezintă valoarea efectivă a tuturor componentelor interarmonice dintre două frecvențe armonice consecutive. Standardul prevede ca acele componente interarmonice care sunt adiacente frecvențelor armonice să nu fie incluse în grupul interarmonic.

Fereastra de timp pe care se face analiza este de 200 ms, cuprinzând 10 perioade pentru semnalele de 60 Hz și 12 perioade pentru semnalele de 50 Hz. Frecvența de eșantionare trebuie să fie sincronizată cu frecvența rețelei, dar suficient de ridicată pentru a putea face posibilă analiza componentelor cu frecvența de până la 2 kHz (armonica a 40-a la 50 Hz) și până la 2,4 kHz (armonica a 40-a la 60 Hz).

Analiza trebuie să se facă în mod continuu, fără pauze între ferestrele de timp succesive. Standardul admite o pauză de maxim o perioadă între ferestrele de timp succesive, în cazul în care analiza acestora nu este posibilă.

Valorile obținute în urma aplicării Transformatei Fourier Discrete trebuie grupate după cum urmează, pentru a obține valorile reprezentative ale interarmonicilor:

$$C_{n+0,5-200-ms}^2 = \sum_{k=-4}^4 C_{10n+k}^2 (\text{sisteme pe 50 Hz})$$

$$C_{n+0,5-200-ms}^2 = \sum_{k=-5}^5 C_{12n+k}^2 (\text{sisteme pe 60 Hz})$$
(2.9)

Intervalul pe care se face afișarea interarmonicilor este de 3 secunde. Dacă pe durata ferestrei de timp pe care se face analiza componentelor interarmonice are loc un eveniment de tipul gol, supratensiune sau întrerupere de tensiune, atunci acel interval de măsurare a interarmonicilor va fi marcat ca fiind neconform.

Parametrii de funcționare ai instrumentelor de măsurat

În standard sunt definiți parametrii de funcționare ai instrumentelor de măsurat și condițiile în care se efectuează aceste măsurări, parametrii specificați în anumite condiții de referință (tabelul 1.7).

Tabelul 1.7 Condiții de referință relative la influențele externe

Parametrul	Valoarea	Toleranța
Temperatura ambiantă	23 °C	± 2 °C
Umiditatea	50 %	± 10 %
Inducția magnetică a unei surse externe la frecvența de referință (50 sau 60 Hz) în orice direcție	0	0,05 mT
Câmp electric extern la frecvența de referință (50 sau 60 Hz) în orice direcție	0	0,1 kV/m
Aer	Absența vaporilor corozivi	

Domeniul de utilizare al instrumentelor este prezentat în tabelul 1.8:

Tabelul 1.8 Domeniul de operare al instrumentelor

Parametrul	Domeniul Clasa A	Domeniul Clasa B
Temperatura ambiantă	- 10 ÷ + 45 °C	0 – 30 °C
Umiditatea	20 % - 95 %	nespecificat
Inducția magnetică a unei surse externe la frecvența de referință (50 sau 60 Hz) în orice direcție	< 0,5 mT	nespecificat
Câmp electric extern la frecvența de referință (50 sau 60 Hz) în orice direcție	< 1 kV/m	nespecificat
Descărcări electrostatice	15 kV	nespecificat
Câmpuri electromagnetice de înaltă frecvență	10 V/m	nespecificat
Aer	Absența vaporilor corozivi	

2.2.3 Standardul EN 50160

Standardul SR EN 50160 definește și prezintă principalele caracteristici ale tensiunii în punctele de conectare a instalațiilor utilizatorilor la rețelele de distribuție de joasă și de medie tensiune, în condiții normale de funcționare. Unele fenomene care afectează calitatea energiei în rețeaua electrică au caracter imprevizibil și din acest motiv valorile specificate în acest standard pentru astfel de fenomene, cum ar fi golurile de tensiune și întreruperile tensiunii de alimentare, trebuie interpretate în mod corespunzător.

Întrucât există o diversitate foarte mare în structura rețelelor de distribuție a energiei electrice în diferite zone, rezultată din diferențele de densitate de sarcină, din dispersia populației, din topografia locală etc., mulți clienți vor constata variații considerabil mai mici ale caracteristicilor tensiunii decât valorile cuprinse în acest standard.

Standardul prevede că, în general, supratensiunile tranzitorii nu trebuie să depășească 6 kV valoare de vârf, dar ocazional pot să apară valori mai mari. Durata acestora acoperă un interval de la câteva milisecunde până la mai puțin de o microsecundă. Energia conținută de supratensiunea tranzitorie variază considerabil în funcție de locul de origine al perturbației. O supratensiune indusă de trăsnet are, în general, o amplitudine mai mare, dar conține o energie mai mică decât o supratensiune cauzată de comutație, datorită în general, duratei mai mari a supratensiunilor de comutație. Dispozitivele de protecție la supratensiune în instalațiile consumatorilor trebuie alese ținând seama, cu mai multă exigență, de supratensiunile de comutație. În acest fel se acoperă atât domeniul supratensiunilor de trăsnet cât și al celor de comutație.

Caracteristicile parametrilor ce definesc calitatea energiei electrice în rețelele de medie tensiune (MT) și joasă tensiune (JT) sunt prezentați în tabelul 1.9.

Tabelul 1.9 Norme EN 50160: Caracteristici ale calității energiei electrice în JT și MT

Indicator	Prevederi standard EN 50160
Frecvența	50 Hz \pm 1 %, 95 % din săptămână; 50 Hz +4 / -6 %, 100 % din săptămână; (valori medii pe 10 secunde)
Variația amplitudinii tensiunii	\pm 10 %, 95 % din săptămână;
Variații rapide (bruste)	Joasă tensiune: 5 % în general, mai rar: 10 % pentru 95% din săptămână; Medie tensiune: 4 % în general, mai rar: 6 % pentru 95% din săptămână;
Variații rapide (flicker)	$P_{\text{f}} \leq 1$ %; 95 % din săptămână;
Goluri de tensiune	Cea mai mare parte a golurilor au o durată mai mică de 1 s cu o amplitudine sub 60 %
Întreruperi de scurtă durată (mai scurte de 3 minute)	De la câteva zeci la mai multe sute pe an; 70 % dintre întreruperi au o durată mai mică de 1 s;
Întreruperi de lungă durată (mai lungi de 3 minute)	De la 10 la maxim 50 pe an;
Supratensiuni temporare	Joasă tensiune: sub 1.5 kV rms; Medie tensiune: 1,7 din tensiunea contractuală;
Supratensiuni tranzitorii	Joasă tensiune: sub 6 kV, ocazional pot avea și valori mai mari, timpul de creștere fiind de ordinul ms - μ s;
Nesimetrie	2 %, 95 % din săptămână, măsurate ca medie a tensiunii efective pe un interval de 10 minute; 3 % în unele locuri;
Armonici	Armonici până la rangul 25 și factor de distorsiune ≤ 8 %, 95 % din săptămână;
Interarmonici	În studiu.

În afara supratensiunilor temporare la frecvența industrială (50 Hz sau 60 Hz), în rețelele electrice pot să apară și supratensiuni tranzitorii (de impuls) în raport cu pământul, de origine atmosferică sau de comutație. Supratensiunile atmosferice se datorează loviturilor de trăsnet în liniile electrice sau în apropierea

acestora. Supratensiunile de comutație pot să apară din cauza unor manevre (comutații) în rețeaua distribuitorului de energie electrică sau în rețeaua utilizatorului. Se întâlnesc frecvent supratensiuni fază – pământ de 2 sau de 3 ori tensiunea nominală.

Protecția contra supratensiunilor tranzitorii trebuie asigurată încă din faza de proiectare (prevederea de paratrăsnete, descărcătoare, etc.), atât în rețelele de distribuție cât și la utilizator.

2.3 Concluzii

Calitatea energiei electrice a devenit în ultimul timp o preocupare majoră, mai ales în cazul particular al consumatorilor casnici, deoarece buna funcționare a echipamentelor electrice și electronice conectate la rețeaua electrică depinde în mare măsură de calitatea energiei electrice furnizate acestora.

Din punctul de vedere al consumatorilor, calitatea energiei electrice se rezumă la livrarea energiei electrice fără variații ale tensiunii și frecvenței acesteia.

Din diverse motive, energia electrică livrată consumatorilor nu este întotdeauna conformă cu cerințele acestora, existând o serie întreagă de evenimente care pot să apară în sistem: scurt-circuite, cuplarea/decuplarea unor generatoare în/din sistem, conectarea/deconectarea unor motoare sincrone, trăsnete etc., evenimente ce pot conduce la apariția unor perturbații: întreruperi de tensiune, căderi de tensiune, supratensiuni, armonici, impulsuri tranzitorii, zgomot etc.

Odată cu dezvoltarea explozivă a tehnicii de calcul și a utilizării pe scară largă a calculatoarelor personale, rețeaua electrică a devenit din ce în ce mai poluată. Sursele de alimentare în comutație ale calculatoarelor personale introduc în rețeaua electrică perturbații, în special impulsuri singulare de scurtă durată. Prezența acestor perturbații în rețea poate împiedica buna funcționare a anumitor echipamente, în special a celor medicale și a receptoarelor radio.

În trecut, datorită lipsei standardizării instrumentelor și a metodologiei de măsurare, era imposibilă compararea datelor obținute în urma monitorizării calității energiei electrice de către diferite institute de cercetare. Astfel, nu puteau fi coroborate datele obținute în site-uri de măsurare diferite, cu instrumente diferite.

Odată cu apariția standardelor în domeniu, lucrurile s-au schimbat, fiind posibile comparații între măsurările efectuate în locații diferite și analiza coerentă a datelor, indiferent de instrumentele folosite pentru măsurarea și înregistrarea parametrilor energiei electrice.

În cadrul grupului de lucru IEEE pentru monitorizarea calității energiei electrice au fost analizate așteptările consumatorilor privind livrarea energiei electrice, limitele de susceptibilitate ale echipamentelor electrice și electronice conectate la rețeaua electrică, ajungându-se la elaborarea unor standarde în ceea ce privește metodele de evaluare a calității energiei electrice și a parametrilor ce definesc această noțiune. Au fost definite clasele de funcționare ale instrumentelor de măsurat și modul în care trebuie efectuate măsurările parametrilor care definesc calitatea energiei electrice.

Noțiunea de calitate a energiei electrice, destul de ambiguă cu câțiva zeci de ani în urmă, a fost reanalizată și redefinită în concordanță cu cerințele actuale ale consumatorilor de energie electrică. Standardele adoptate în domeniu definesc o serie întreagă de perturbații ce pot să apară în rețelele electrice și conțin recomandări pentru evitarea apariției acestora.

3. CIRCUITE DE CONDIȚIONARE A SEMNALELOR

3.1 Introducere

Schema tipică a unui sistem de monitorizare a parametrilor rețelelor electrice de joasă tensiune este prezentată în figura 3.1. Tensiunea preluată din rețeaua electrică este aplicată unui bloc de condiționare, este convertită în eșantioane cu ajutorul unui convertor analog-numeric și apoi procesată de către un sistem cu microprocesor (calculator personal sau microcontroler).

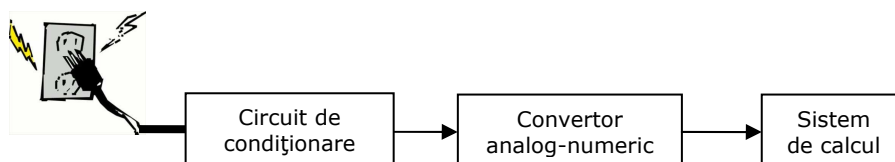


Fig. 3.1. Schema bloc a unui sistem de monitorizare a perturbațiilor din rețeaua electrică de joasă tensiune

Rolul circuitului de condiționare este acela de a adapta nivelurile de tensiune de la intrarea circuitului, la cele acceptate de către circuitul de conversie analog – numeric (în general ± 10 V).

Deși tensiunea nominală a rețelei de distribuție a energiei electrice de joasă tensiune este (în Europa) de 230V, pot apărea, sporadic, perturbații cu amplitudini de până la 10 kV. Se pune problema preluării acestor semnale (perturbate sau nu) din rețeaua electrică și a procesării acestora în bune condiții astfel încât să nu se piardă date pe parcursul acestui lanț de procesare. Este nevoie de o achiziție a semnalelor cu frecvență ridicată și cu o dinamică foarte mare, lucru care necesită convertoare analog-numerice cu un număr foarte mare de biți (minim 16 biți) și rate de eșantionare ridicate (până la 1 – 10 Msps). Astfel de convertoare, pe lângă faptul că sunt foarte scumpe, nici nu au numărul efectiv de biți (ENOB) egal cu numărul de biți declarat, lucru datorat în primul rând faptului că la peste 20 - 24 de biți, cuanta de tensiune se apropie de valori de ordinul 10 – 20 μ V, confundându-se cu zgomotul din sistem.

În marea majoritate a timpului, tensiunea din rețea va avea o valoare apropiată de 230 V, destul de des (1 – 2 ori pe zi), există vârfuri de tensiune cu amplitudini până în 2 kV și sporadic (odată pe săptămână, de 2-3 ori pe lună), vor exista vârfuri de tensiune cu amplitudini de peste 6 kV. Cele mai multe perturbații au amplitudini mici, în jurul valorii de 230 V, nedepășind însă 500 V [60].

Se pune problema realizării unei compresii a semnalului, compresie mai mică pentru semnalele cu amplitudini în jurul tensiunii nominale a rețelei, mai mare pentru tensiuni cu amplitudini cuprinse între 500 V și 2 kV, și o compresie mai puternică pentru tensiuni mai mari de 2 kV. Astfel, se păstrează un număr de biți suficient de mare pentru tensiuni cu amplitudini în jurul tensiunii nominale a rețelei, și se asigură totodată captarea vârfurilor de tensiune de valori foarte mari cu o

acuratețe satisfăcătoare. Este mai importantă detectarea și înregistrarea acestor vârfuri de tensiune decât măsurarea cu acuratețe ridicată a amplitudinii acestora.

Dacă nu s-ar folosi un circuit de compresie a semnalelor în partea de condiționare a semnalelor și s-ar conecta la intrarea convertorului analog – numeric un divizor simplu de tensiune, atunci, pentru semnale de intrare cu amplitudini de până la ± 10 kV, în cazul unui convertor analog – numeric pe 16 biți, cuanta de tensiune ar fi de aproximativ 300 mV, ceea ce conduce la un număr de doar 1000 de valori discrete în intervalul 0 - 300 V, echivalent cu folosirea unui convertor pe 10 biți pentru acest interval de tensiuni de intrare.

3.2 Circuit de condiționare cu transformator funcțional

3.2.1 Prezentarea circuitului

Cel mai vechi și mai simplu circuit electronic de compresie a semnalelor este transformatorul funcțional (figura 3.2) [61], [62]. Acesta utilizează un circuit integrat analogic și câteva diode în circuitul de reacție negativă, diodele având rol de comutatoare comandate de tensiunea de la intrare. Atunci când tensiunea de la intrarea circuitului depășește un anumit prag, se deschide dioda aferentă pragului respectiv și se modifică amplificarea sistemului.

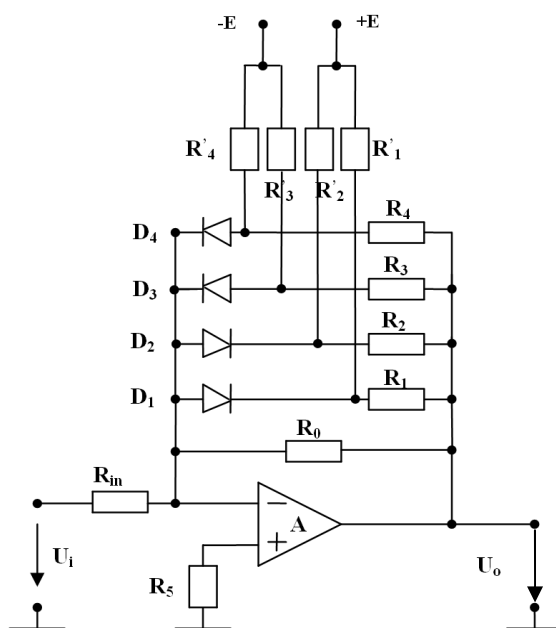


Fig. 3.2. Transformator funcțional cu amplificator operațional

3.2.2 Proiectarea circuitului de compresie cu transformator funcțional

Pentru proiectarea acestui circuit s-au parcurs mai multe etape [63]:

1) Se stabilește numărul de segmente cu care se va aproxima caracteristica de transfer. Se alege un număr de segmente egal cu 3 și se consideră că eroarea relativă maximă este aproximativ aceeași pe toate segmentele (figura 3.3).

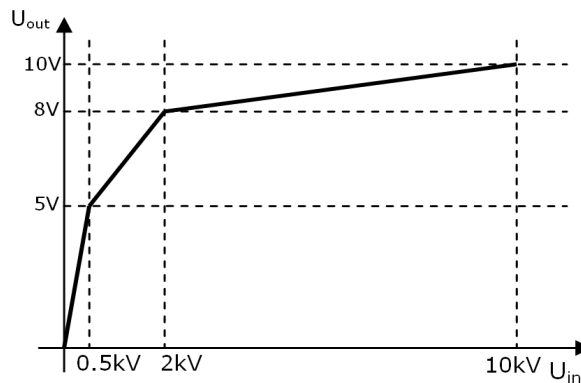


Fig. 3.3. Caracteristica de transfer a transformatorului funcțional în cadranul I

2) Se alege amplificatorul operațional astfel încât curentul de intrare să fie cât mai mic și rezistența de intrare cât mai mare. S-a ales un amplificator operațional AD8616, fabricat de firma Analog Devices care are un curent de intrare de 0.2 pA și o rezistență de intrare de 100 MΩ. Acest curent foarte mic de intrare al amplificatorului operațional asigură o cădere de tensiune neglijabilă pe rezistența de intrare, cădere de tensiune care nu va influența valoarea tensiunii de ieșire. Se alege valoarea rezistenței de intrare astfel încât să satisfacă relația:

$$R_{in} \leq \frac{R_{inAO}}{100} \Rightarrow R_{in} \leq 1M\Omega \quad (3.1)$$

Se alege o rezistență de intrare $R_{in} = 1 M\Omega \pm 1\%$, valoare ce satisface relația (3.1).

3) Se determină pantele tuturor segmentelor caracteristicii de transfer cu relația:

$$a_k = \frac{U_{ek+1} - U_{ek}}{U_{ik+1} - U_{ik}} \quad , \quad k = 0,1,2 \quad (3.2)$$

rezultând

$$a_0 = \frac{5V - 0V}{500V - 0V} = 0.01 \quad (3.3)$$

$$a_1 = \frac{8V - 5V}{2000V - 500V} = \frac{3}{1500} = 0.002 \quad (3.4)$$

$$a_2 = \frac{10V - 8V}{10000V - 2000V} = \frac{2}{8000} = 0.00025 \quad (3.5)$$

4) Pentru tensiuni de intrare cuprinse în intervalul (-500V, +500V) diodele D_1 , D_2 , D_3 și D_4 sunt blocate, iar circuitul este echivalent cu un divizor de tensiune format din rezistoarele R_{in} și R_o . Rezistența R_o se determină pe baza relațiilor de mai jos:

$$U_o = \frac{R_o}{R_{in}} U_i \quad (3.6)$$

$$a_0 = \frac{U_o}{U_i} \quad (3.7)$$

$$R_o = a_0 R_{in} \quad (3.8)$$

rezultând

$$R_o = 0.01 \cdot R_{in} = 10 \text{ k}\Omega \quad (3.9)$$

5) Se adoptă tensiunea de alimentare a transformatorului funcțional ca fiind cu cel puțin 2-3V mai mare decât tensiunea maximă de la ieșirea transformatorului funcțional ($\pm 10V$). Se alege o valoare de $\pm 15V$.

6) Considerăm funcționarea transformatorului funcțional în cadranul I al caracteristicii de transfer și tensiunea de la intrare cuprinsă în intervalul [+0.5 kV, +2 kV). În acest caz dioda D_1 este în conducție, iar circuitul echivalent este format din R_{in} ca și rezistență de intrare și R_o în paralel cu R_1 ca și rezistență de reacție a amplificatorului operațional. Astfel avem:

$$U_o = U_i \frac{R_{1echiv}}{R_{in}} \quad (3.10)$$

$$a_1 = \frac{U_o}{U_i} \quad (3.11)$$

$$R_{1echiv} = a_1 R_{in} \quad (3.12)$$

în care

$$R_{1echiv} = R_o \parallel R_1 \quad (3.13)$$

din relațiile (3.5) – (3.13) se obține:

$$a_1 = \frac{R_{1echiv}}{R_{in}} = \frac{R_0 R_1}{R_0 + R_1} \quad (3.14)$$

$$R_1 = R_{in} \frac{a_0 a_1}{a_0 - a_1} \quad (3.15)$$

Înlocuind valorile numerice obținem $R_1 = 2.5 \text{ k}\Omega$.

Deoarece caracteristica de transfer a circuitului de compresie dinamică este simetrică în cadranul III față de cadranul I, se procedează similar și pentru cazul în care tensiunea de intrare este cuprinsă în intervalul $[-0.5 \text{ kV}, -2 \text{ kV}]$. În acest caz dioda D_3 este în conducție, obținându-se $R_3 = 2.5 \text{ k}\Omega$.

7) Considerăm funcționarea transformatorului funcțional în cadranul I pentru valori ale tensiunii de intrare cuprinse în intervalul $[2 \text{ kV}, 10 \text{ kV}]$. În acest caz diodele D_1 și D_2 sunt ambele în conducție și rezistența din reacția amplificatorului operațional are valoarea:

$$R_{2echiv} = R_0 \parallel R_1 \parallel R_2 \quad (3.16)$$

În acest caz, tensiunea de ieșire urmărește panta a_2 a caracteristicii și putem scrie:

$$U_o = U_i \frac{R_{2echiv}}{R_{in}} \quad (3.17)$$

$$a_2 = \frac{U_o}{U_i} \quad (3.18)$$

$$R_{2echiv} = a_2 R_{in} \quad (3.19)$$

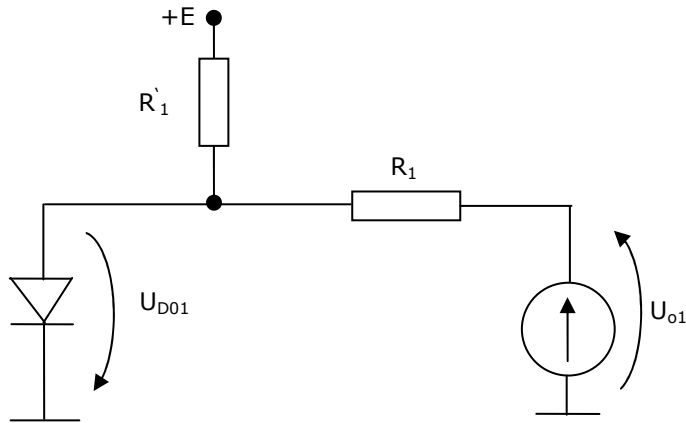
Înlocuind relația (3.11) în relația (3.19) rezultă:

$$R_2 = R_{in} \frac{a_1 a_2}{a_1 - a_2} \quad (2.20)$$

Rezultă $R_2 = 222 \text{ }\Omega$. Se alege pentru R_2 o valoare standardizată : $R_2 = 220 \text{ }\Omega$.

Se procedează identic pentru cadranul III, rezultând $R_4 = 220 \text{ }\Omega$.

8) Se calculează valorile rezistențelor care vor asigura punctele de deschidere ale diodelor D_1 , D_2 , D_3 și D_4 . Pentru cazul în care dioda D_1 intră în conducție vom folosi schema echivalentă prezentată în figura 3.4.

Fig. 3.4. Circuitul echivalent pentru calculul valorii rezistenței R'_1

unde cu U_{D01} s-a notat tensiunea de deschidere a diodei D_1 , iar cu U_{o1} s-a notat valoarea tensiunii de la ieșirea circuitului în primul punct de frângere al caracteristicii de transfer. Acest punct reprezintă momentul intrării în conducție al diodei D_1 . Putem scrie:

$$U_{D01} = E \frac{R_1}{R_1 + R'_1} - U_{o1} \frac{R'_1}{R_1 + R'_1} \quad (3.21)$$

$$R'_1 = R_1 \frac{E - U_{D01}}{U_{o1} - U_{D01}} \quad (3.22)$$

se obține $R'_1 = 8,8 \text{ k}\Omega$.

Procedând similar se poate deduce valoarea rezistorului R'_2 :

$$R'_2 = R_2 \frac{E - U_{D02}}{U_{o2} - U_{D02}} \quad (3.23)$$

se obține $R'_2 = 450 \Omega$.

Pentru determinarea rezistențelor R'_3 și R'_4 se aplică aceeași metodă de calcul. Rezultă $R'_1 = R'_3$ și $R'_2 = R'_4$.

Rezistența R_5 se alege 0 din motive de zgomot minim la ieșire.

3.2.3 Simularea funcționării transformatorului funcțional

Circuitul a fost simulat utilizând aplicația PSpice [64]. În figura 3.5 este prezentată caracteristica de transfer a circuitului simulată în PSpice.

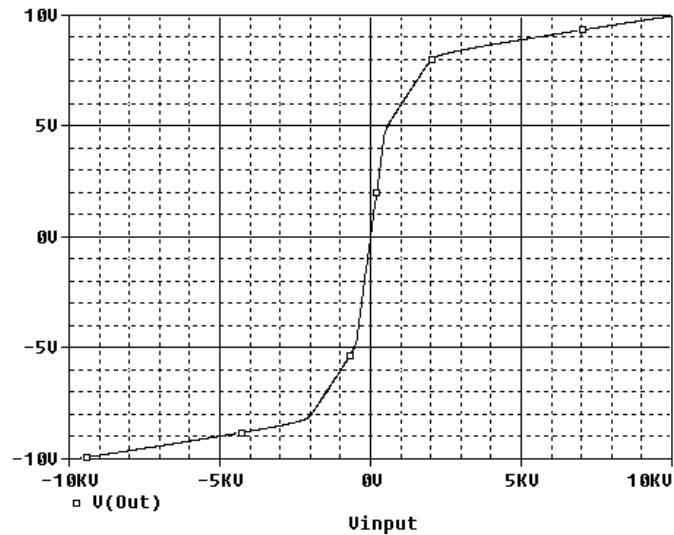


Fig. 3.5. Funcția de transfer a circuitului simulată în PSpice

Este interesant de observat cum se comportă circuitul în punctele de frângere ale caracteristicii de transfer, puncte în care, din cauza rezistenței dinamice, a capacității parazite a diodei și a tensiunilor de comutare diferite ale diodelor, comutarea nu se face brusc (figura 3.6). Acesta este un dezavantaj major la acest tip de circuit.

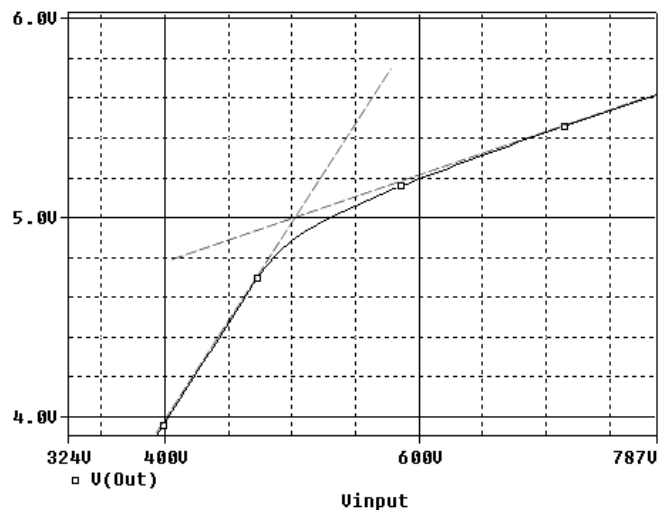


Fig. 3.6. Abaterea de la caracteristica reală la 500 V

După cum se poate observa și din figura 3.6, caracteristica ideală (trasată cu linie întreruptă) este cea proiectată, primul punct de frângere fiind la $U_i = 500$ V și $U_o = 5$ V, așa cum a fost proiectat. Caracteristica reală a circuitului (trasată cu linie continuă) prezintă o rotunjire în jurul punctului de frângere, cauzată de caracteristica reală a diodei. Există un interval de timp în care diodele D_1 și D_2 sunt

simultan în conducție, fapt ce produce abaterea caracteristicii reale de la cea ideală cu aproximativ 1%.

În cazul celui de-al doilea punct de frângere (figura 3.7), se constată că eroarea este sensibil mai mare decât în cazul primului punct de frângere, acest lucru datorându-se și propagării erorii din primul punct. În plus, la calculul valorii rezistorului R_1 s-a aproximat valoarea tensiunii de deschidere a diodei D_2 . Eroarea maximă în acest caz este de aproximativ 1.5 %.

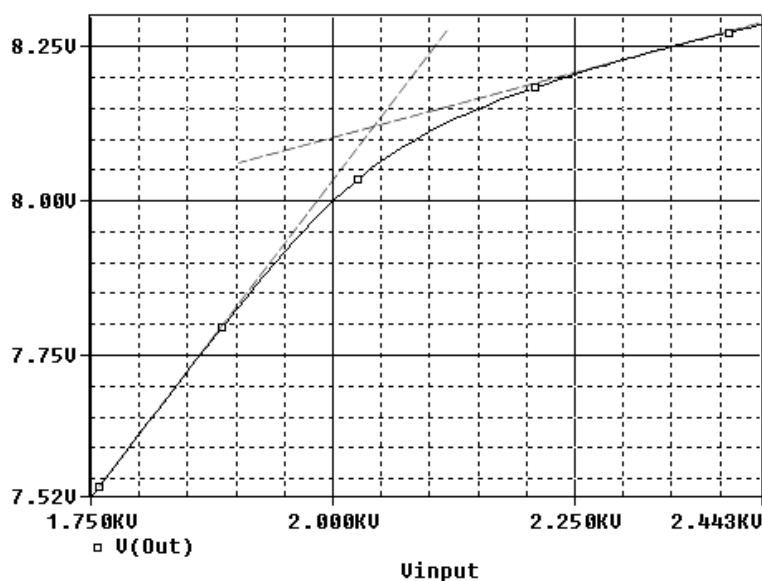


Fig. 3.7. Abaterea de la caracteristica reală la 2 kV

Cel de-al treilea punct important este punctul de 10 kV. Aici lucrurile stau destul de bine, pentru o tensiune de intrare de 10 kV, amplificatorul dă la ieșire o tensiune de 9.998V, eroarea fiind de doar 0.2 %. Această eroare este dată de aproximarea rezistenței dinamice a diodei D_2 , a cărei valoare este de același ordin de mărime ca și valoarea rezistorului R_2 . Pentru a minimiza această eroare, este necesar ca valoarea rezistorului R_2 să fie de cel puțin 5-10 ori mai mare decât rezistența dinamică a diodei D_2 . Din păcate, creșterea valorii lui R_2 , conduce automat la creșterea valorii rezistorului R_{in} (relația 2.20), lucru care nu este de dorit, deoarece devine comparabilă ca valoare cu rezistența de intrare a amplificatorului operațional, ceea ce va genera erori suplimentare datorate divizorului de tensiune de la intrarea montajului format din R_{in} și rezistența de intrare a amplificatorului.

Un alt parametru care influențează funcționarea acestui amplificator este temperatura. Influența temperaturii a fost studiată în intervalul $10\text{ }^{\circ}\text{C} \div 50\text{ }^{\circ}\text{C}$ cu un pas de $10\text{ }^{\circ}\text{C}$.

Temperatura influențează valorile rezistoarelor din reacția amplificatorului operațional, rezistența dinamică și tensiunea de prag U_{D0} a diodelor. Modificarea cu temperatura a punctelor de frângere a caracteristicii de transfer a circuitului este prezentată în figura 3.8. Se poate observa dependența puternică cu temperatura a caracteristicii de transfer a circuitului, fapt ce impune termostatarea acestuia.

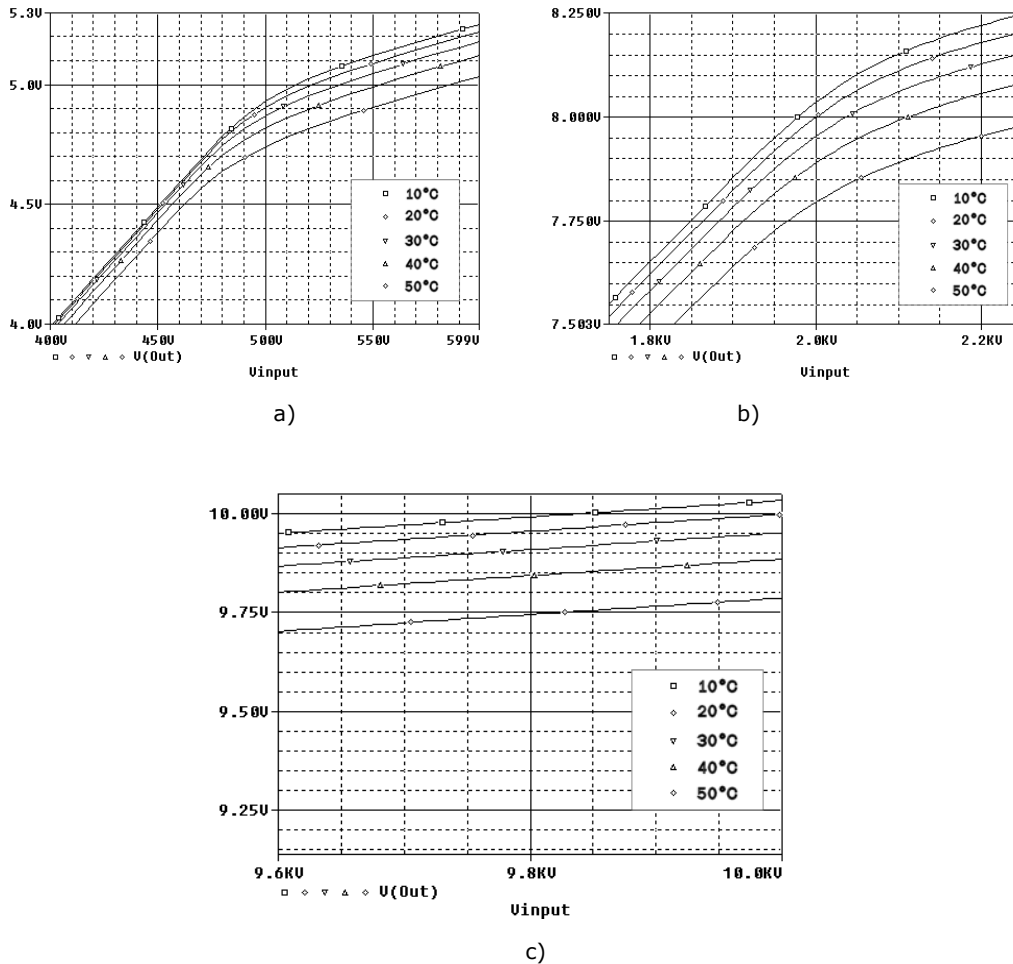


Fig. 3.8. Influența temperaturii asupra punctelor de frângere ale caracteristicii circuitului de compresie dinamică

a) la 500 V; b) la 2 kV; c) la 10 kV

Pentru a studia comportamentul dinamic al circuitului, s-a aplicat la intrarea acestuia un semnal de tip triunghi, acest tip de semnal fiind cel mai potrivit pentru studierea caracteristicii de transfer a circuitului. Semnalul triunghiular are o amplitudine de 10 kV și o frecvență de 1 kHz, având aplicat un offset de 5 kV. S-a studiat doar comportarea în cadranul I, comportarea circuitului în cadranul III fiind similară.

Se observă din figura 3.9, că semnalul de ieșire corespunde caracteristicii de transfer a circuitului. Pentru a decompresa semnalul obținut cu un astfel de circuit, se folosește un modul software ce aplică o funcție inversă (funcție de decompresie) celei aplicate de către circuitul de compresie.

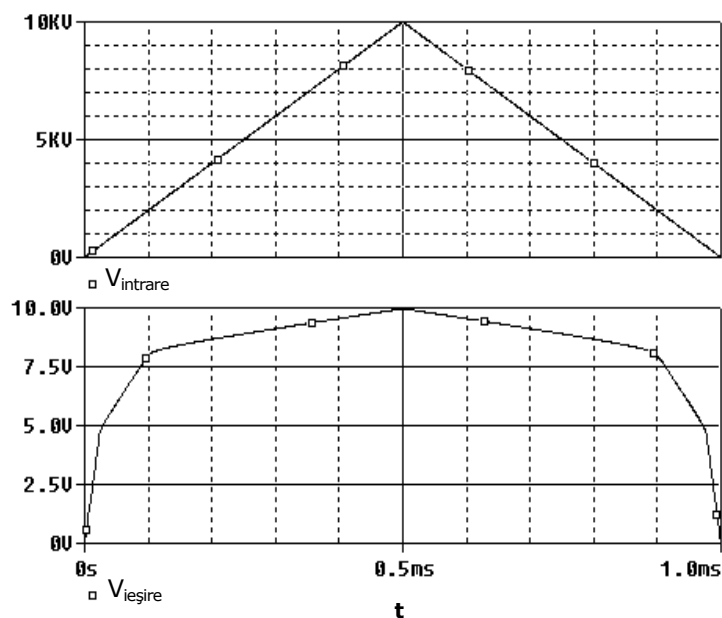


Fig. 3.9. Semnalul de intrare triunghiular aplicat circuitului și semnalul de la ieșirea circuitului

În figura 3.10 este prezentat semnalul de la intrarea circuitului de compresie și semnalul de la ieșirea circuitului de decompresie. Se observă diferențele dintre cele două semnale, diferențe datorate erorilor care apar în punctele de frângere ale caracteristicii circuitului de compresie. Datorită capacității parazite a diodelor, tensiunea de la ieșirea circuitului nu poate crește brusc, capacitățile parazite formând, împreună cu celelalte rezistențe din circuit, un filtru trece-jos.

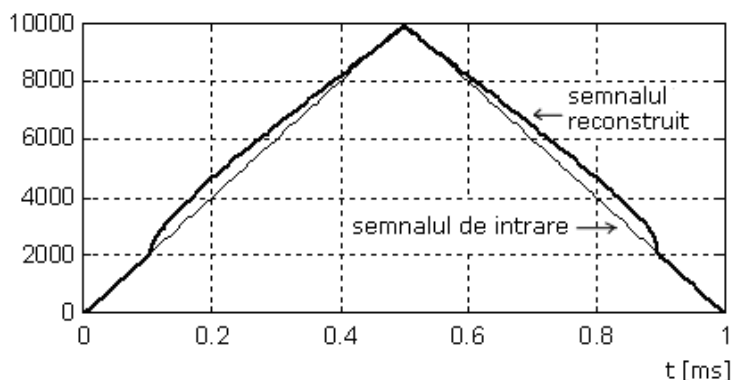


Fig. 3.10. Semnalul de intrare triunghiular aplicat circuitului și semnalul reconstruit de aplicația software

Prin aplicarea unui semnal dreptunghiular la intrarea circuitului de compresie și prin evaluarea semnalului de la ieșirea blocului de decompresie se poate

determina eroarea absolută a circuitului de compresie și se poate interveni în algoritmul sistemului de decompresie pentru a elimina aceste erori.

Analiza spectrală a semnalului de la ieșirea circuitului de compresie prezintă diferențe mari între spectrele semnalelor de intrare și de ieșire (figura 3.11). Astfel, se poate observa că în spectrul semnalului de ieșire apar armonici de ordin impar.

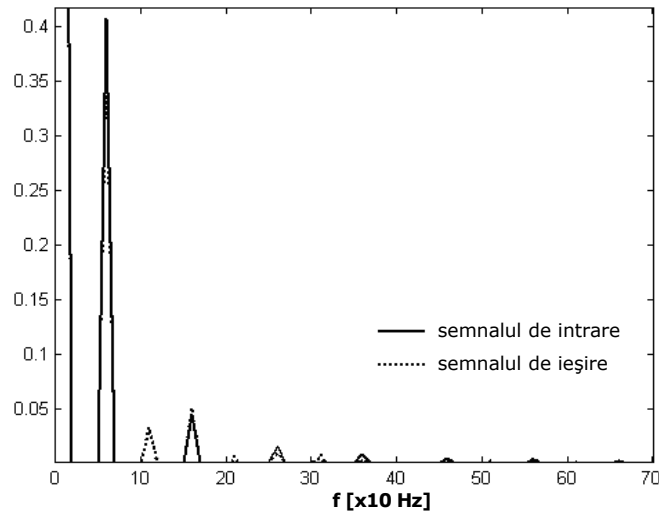


Fig. 3.11. Spectrele semnalelor de intrare, respectiv de ieșire ale circuitului de compresie

Deoarece în practică nu se întâlnesc decât foarte rar semnale triunghiulare, în continuare se va evalua funcționarea circuitului în cazul în care la intrarea sa se aplică semnale de tip biexponențial (figura 3.12) și sinus amortizat, cele mai multe perturbații din rețelele electrice având astfel de forme.

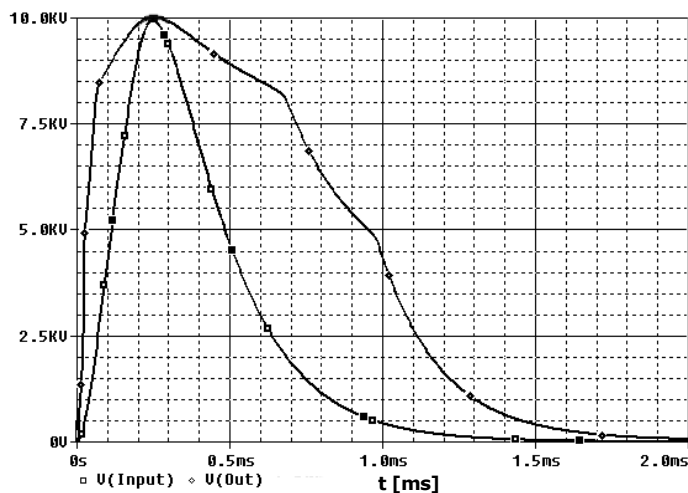


Fig. 3.12. Semnal biexponențial aplicat la intrare și semnalul de la ieșirea circuitului de compresie

Semnalele de tip biexponențial întâlnite ca perturbații în rețelele de distribuție a energiei electrice au un timp de ridicare foarte scurt, de ordinul microsecundelor. Astfel de perturbații sunt adesea cauzate de trăsnete sau de cuplarea/decuplarea unor sarcini inductive mari.

După cum se poate observa din figura 3.13, eroarea este maximă în cel de-al doilea punct de frângere al caracteristicii.

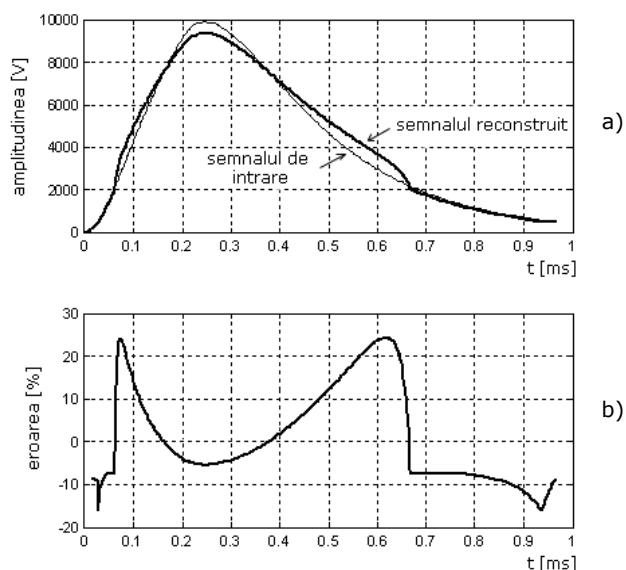


Fig. 3.13. a) Semnal biexponențial aplicat la intrarea circuitului de compresie și semnalul de la ieșirea circuitului de decompresie
 b) eroarea relativă a semnalului de la ieșirea circuitului de decompresie față de semnalul aplicat la intrarea circuitului de compresie

De asemenea, se observă o creștere a numărului armonicilor de ordin superior în spectrul semnalului decomprimat (figura 3.14).

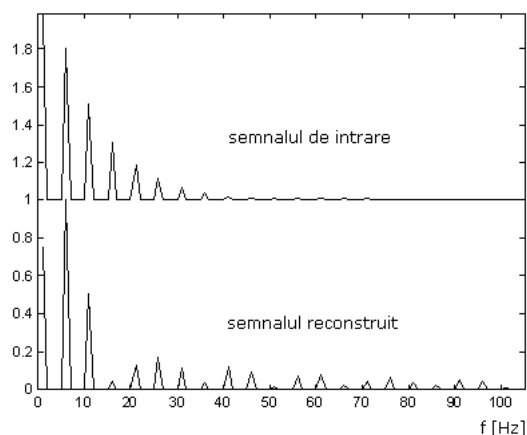


Fig. 3.14. Spectrul semnalului de intrare biexponențial și al semnalului de ieșire după decompresie

Aceste armonici sunt generate, în marea lor majoritate, de către circuitul de compresie la trecerea semnalului prin punctul de frângere corespunzător unei tensiuni de intrare de 2 kV, și datorită slew-rate-ului relativ mic ($1000 \text{ V}/\mu\text{s}$) al amplificatorului operațional utilizat.

Un alt tip de perturbații întâlnite în rețelele de distribuție a energiei electrice sunt perturbațiile de tip sinus amortizat. Pentru testarea comportamentului circuitului la astfel de perturbații, s-a aplicat la intrarea acestuia un semnal de tipul sinus amortizat (uneori căderile de tensiune, dar și conectarea unor baterii de condensatoare, au forma unui sinus amortizat) cu amplitudinea maximă de 10 kV (figura 3.15).

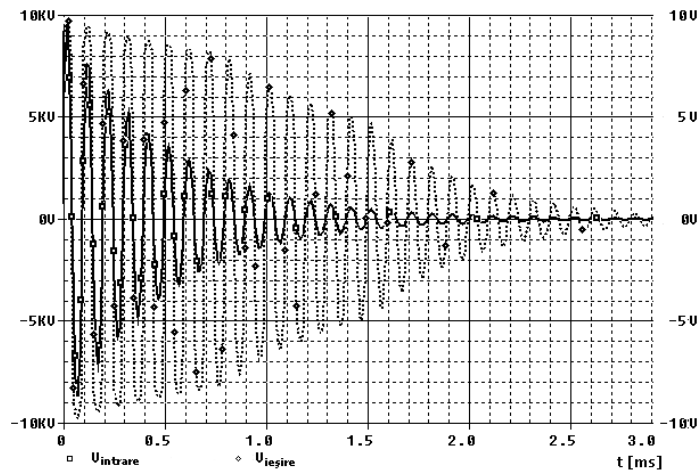


Fig. 3.15. Semnal de intrare sinus amortizat și semnalul de la ieșirea circuitului de compresie (linie punctată)

Perturbația în sine constă într-un semnal sinusoidal a cărei amplitudine scade după o lege exponențială.

Semnalul de la ieșirea circuitului de compresie dinamică este aplicat circuitului de decompresie și se obține imaginea din figura 3.16.

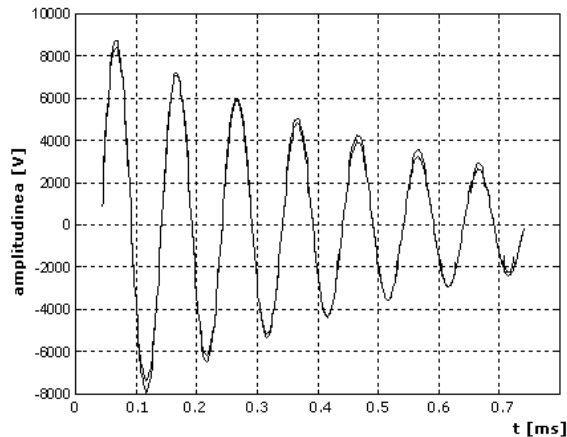


Fig. 3.16. Semnal de intrare sinus amortizat și semnalul de la ieșirea circuitului de decompresie (linie punctată)

În acest caz, erorile sunt mai mici decât în cazul semnalului biexponențial, dar au totuși valori ridicate. Eroarea maximă, la 2 kV este de 12 %. În figura 3.17 este prezentată eroarea relativă a semnalului decomprimat față de semnalul aplicat la intrare.

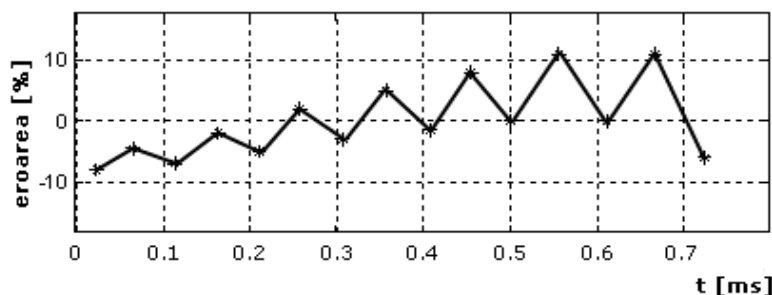


Fig. 3.17. Eroarea relativă a amplitudinii maxime a sinusului amortizat după decompresie

În jurul valorii de 10 kV, eroarea este negativă și egală cu 9 %, fiind minimă pentru tensiuni de intrare în jurul a 6 kV și crescând apoi din nou pentru tensiuni de intrare în jurul a 2 kV. Se observă că erorile pentru amplitudini pozitive ale semnalului de la intrare sunt diferite de cele pentru amplitudinile negative, lucru datorat dispersiei tehnologice a diodelor din circuit (în simularea PSpice s-a introdus și dispersia parametrilor). Pentru a preveni acest lucru, diodele trebuie selectate cu atenție, atât în ceea ce privește tensiunea de deschidere U_{D0} , cât și în ceea ce privește curentul de saturație I_{D0} .

La fel ca și în cazul anterior, spectrul semnalului prezintă armonici de ordin superior suplimentare (figura 3.18).

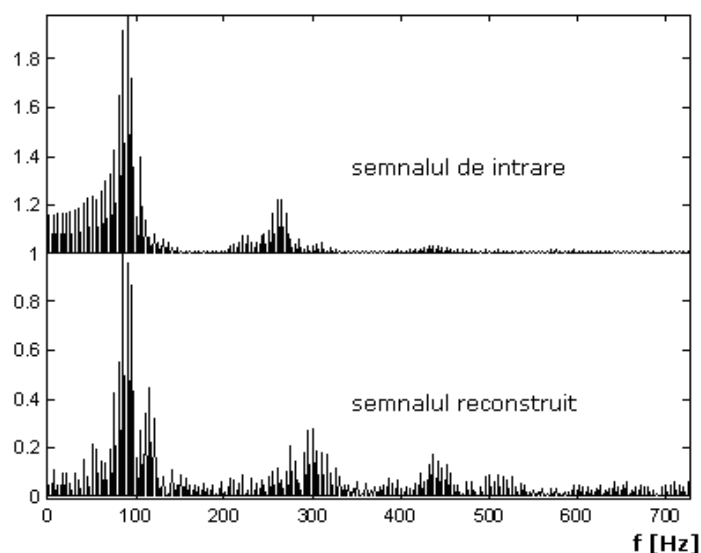


Fig. 3.18. Spectrul semnalului sinus amortizat de la intrare și după decompresie

Se impune de la sine aplicarea unui algoritm de reducere a acestor erori, algoritm ce va fi prezentat în paragraful 3.2.4.

3.2.4 Algoritm pentru reducerea erorilor circuitului de condiționare cu transformator funcțional

Pentru a reduce erorile generate de divizorul de tensiune neliniar cu transformator funcțional, autorul propune introducerea în circuit a unui modul suplimentar de calibrare (figura 3.19) [65]. Rolul acestui modul este de a aplica la intrarea sistemului un impuls triunghiular (sau de altă formă) cu parametri cunoscuți, de a măsura răspunsul sistemului de compresie-decompresie la acest semnal și de a genera o funcție de corecție, astfel încât eroarea relativă a semnalului de la ieșirea blocului de decompresie să fie minimă.

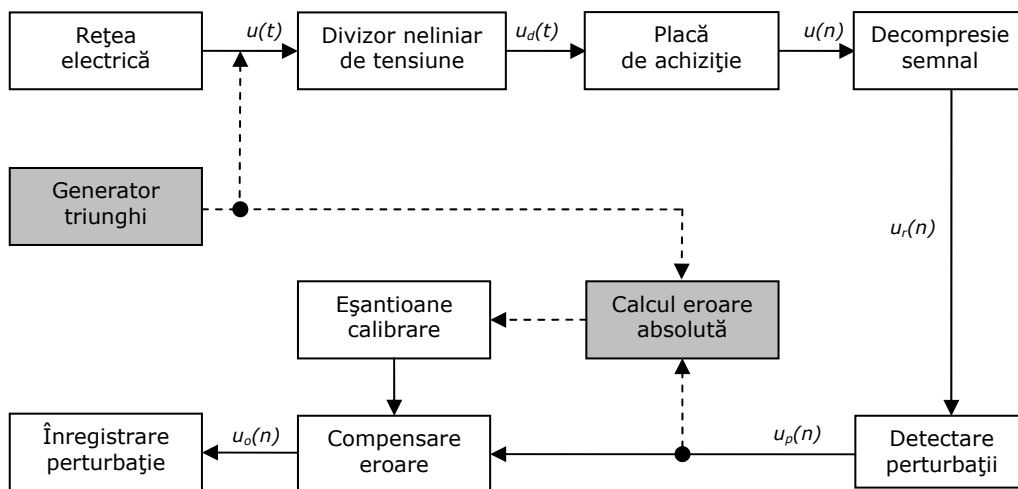


Fig. 3.19. Schema bloc a circuitului propus pentru minimizarea erorilor prin calibrare

Pentru testarea funcționalității soluției propuse s-a folosit o aplicație elaborată în Matlab 7, al cărui cod sursă este prezentat în Anexa 2. Prin aplicarea unui semnal triunghiular la intrarea sistemului, s-a determinat pentru fiecare eșantion al semnalului de intrare valoarea abaterii semnalului de la ieșirea blocului de decompresie și eșantioanele astfel obținute au fost memorate într-un fișier. Aceste eșantioane sunt folosite pentru a corecta caracteristica de transfer a circuitului, astfel încât să fie minimizeze erorile din punctele de frângere ale acesteia.

Pentru testarea sistemului propus, la intrarea sistemului s-a aplicat un semnal de tip biexponențial. S-a urmărit comportarea circuitului pentru diverse tensiuni de intrare, dar în special pentru tensiuni de intrare în jurul punctelor de frângere ale caracteristicii de transfer a divizorului de tensiune neliniar.

Pentru început se va analiza comportamentul circuitului pentru tensiuni de intrare de 2 kV. În intervalul tensiunilor de intrare de la 0 la 500 V, eroarea se menține la o valoare minimă și circuitul de calibrare nu are efect, deoarece între aceste limite, circuitul acționează ca un simplu divizor de tensiune (figura 3.20).

Pentru valori ale tensiunii de intrare mai mari de 500 V efectul circuitului de calibrare devine vizibil, eroarea relativă a tensiunii de ieșire fiind mai mică decât eroarea relativă a tensiunii de ieșire fără circuitul de calibrare.

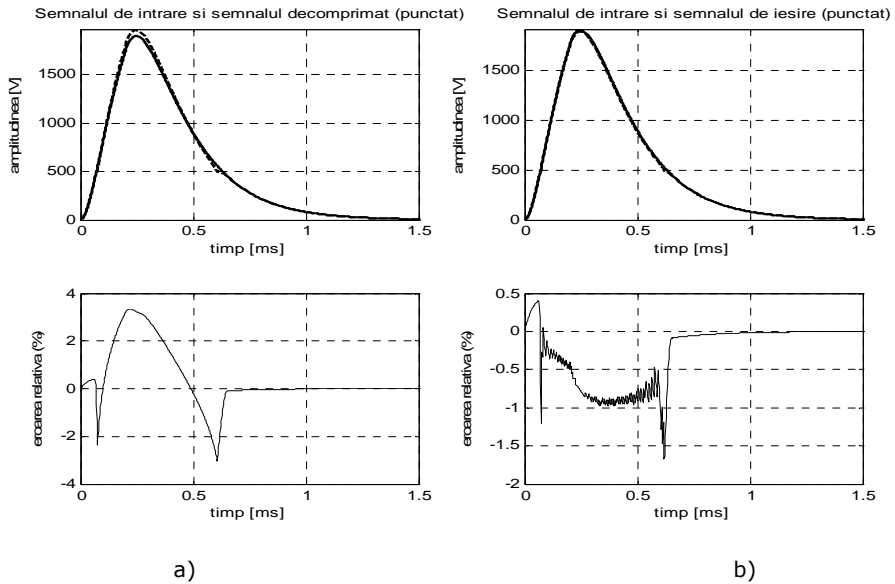


Fig. 3.20. Semnalul de intrare și semnalul decomprimat la 1.9 kV
a) fără corecție; b) cu corecție

În continuare, se aplică la intrarea circuitului o tensiune de intrare de 3 kV (figura 3.21).

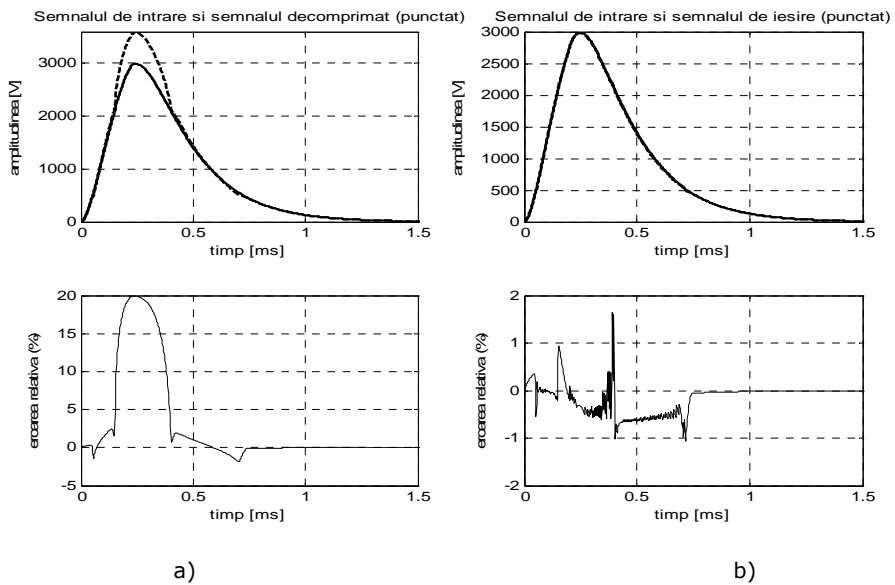


Fig. 3.21. Semnalul de intrare și semnalul decomprimat la 3 kV
a) fără corecție; b) cu corecție

Pentru semnale de intrare cu amplitudini peste 2 kV se observă o scădere a erorii relative a tensiunii de ieșire $u_0(n)$, în cazul folosirii circuitului de calibrare.

În figura 3.22 se prezintă semnalele de intrare, decompimat și de ieșire pentru un semnal de intrare cu amplitudinea de 7 kV.

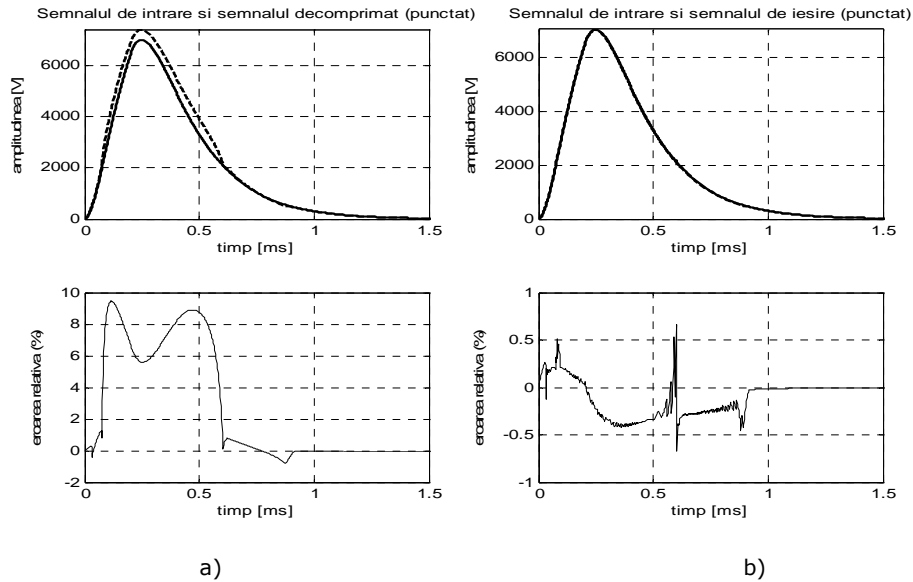


Fig. 3.22. Semnalul de intrare și semnalul decompimat la 7 kV
a) fără corecție; b) cu corecție

În figura 3.23 se prezintă semnalele de intrare, decompimat și de ieșire pentru un semnal de intrare cu amplitudinea de 10 kV.

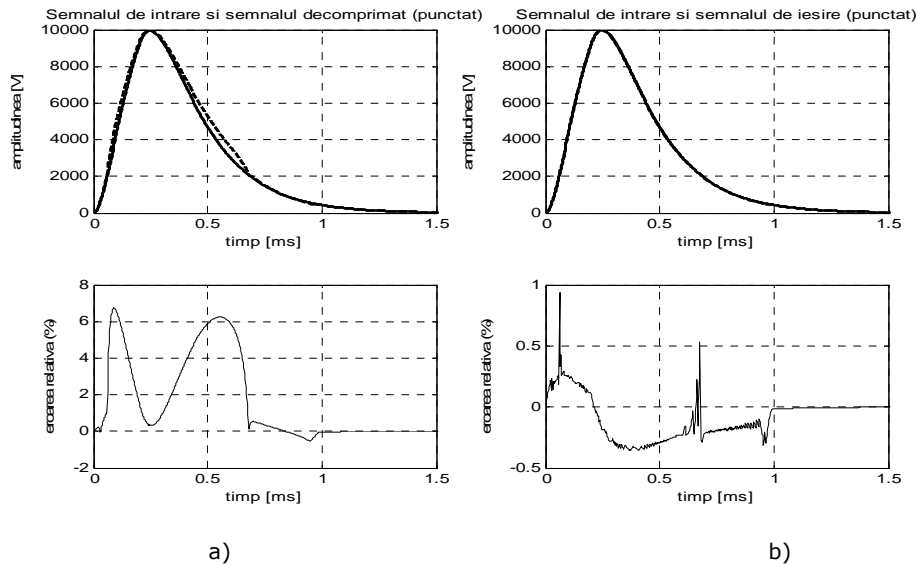


Fig. 3.23. Semnalul de intrare și semnalul decompimat la 10 kV
a) fără corecție; b) cu corecție

În tabelul 3.1 sunt prezentate erorile relative ale semnalului comprimat și decomprimat. Se poate observa că eroarea relativă a semnalului de la ieșirea circuitului, după aplicarea corecției (ca urmare a calibrării circuitului), este mult diminuată față de eroarea relativă a semnalului decomprimat.

Tabelul 3.1 Erorile relative ale semnalului comprimat și decomprimat

Tensiunea semnalului de intrare $u(t)$	Eroarea relativă a amplitudinii semnalului decomprimat ($u_r(n)$)	Eroarea relativă a semnalului de ieșire ($u_c(n)$)
[V]	[%]	[%]
230	0.43	0.43
460	0.43	0.43
650	9	2.5
1900	3.24	1.6
3000	20	1.65
5000	12.50	1.05
7000	9.51	0.70
10000	6.82	0.95

În tabelul 3.2 sunt prezentate energiile semnalului de intrare a semnalului comprimat și a celui decomprimat, precum și erorile relative ale acestor energii raportate la energia semnalului de intrare. Se poate observa că energia semnalului de ieșire în cazul circuitului cu calibrare se apropie de energia semnalului de intrare.

Tabelul 3.2 Energiile semnalelor

Tensiunea semnalului de intrare	Energia semnalului de intrare	Energia semnalului decomprimat	Energia semnalului de ieșire	Eroarea relativă a energiei semnalului decomprimat	Eroarea relativă a semnalului de ieșire
[V]	[mW]	[mW]	[mW]	[%]	[%]
230	47.78	47.76	47.76	-0.03	-0.03
460	88.79	88.77	88.77	-0.02	-0.02
650	132	128	131	-3.22	-1.22
1900	416	422	412	1.33	-0.99
3000	655	722	652	10.19	-0.54
5000	1050	1172	1046	11.57	-0.45
7000	1540	1680	1535	9.10	-0.36
10000	2186	2310	2180	7.61	-0.28

3.3 Circuit de compresie dinamică folosind trei canale ale convertorului analog–numeric (divizor secvențial de tensiune)

3.3.1 Prezentarea circuitului

Circuitele de compresie a semnalelor care utilizează dispozitive neliniare (dispozitive semiconductoare: diode, tranzistoare) sunt predispuse la apariția de distorsiuni, introduc erori la modificarea temperaturii și din aceste motive, autorul a încercat să găsească o altă metodă de realizare a acestui tip de circuit folosind doar componente liniare (rezistoare).

Un astfel de circuit este cel prezentat în figura 3.24. Circuitul de compresie este compus din trei divizoare rezistive de tensiune cu rapoarte de divizare diferite.

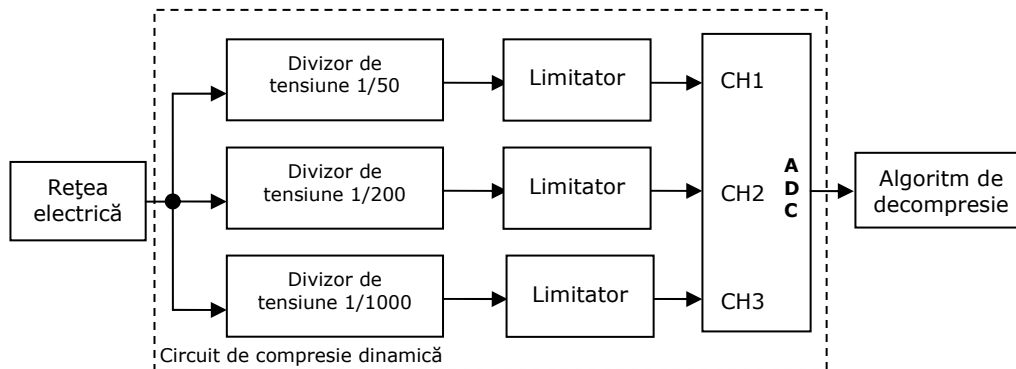


Fig. 3.24. Schema bloc a circuitului de compresie dinamică propus

Semnalul preluat din rețeaua electrică este aplicat simultan la intrările celor trei divizoare rezistive, iar semnalele de la ieșirile divizoarelor rezistive sunt aplicate la intrările a trei canale ale unei plăci de achiziție [66].

Circuitul de limitare (realizat cu diode Zener) are rolul de a limita tensiunea aplicată plăcii de achiziție la ± 10 V. Deși placa de achiziție realizează o limitare hardware a semnalelor de intrare la ± 10 V, deoarece valorile tensiunilor de la ieșirea divizoarelor de tensiune, în cazul în care apar perturbații cu amplitudini de ordinul kV sunt mult mai mari decât tensiunea maximă acceptată de intrările plăcii de achiziție (50 V). Spre exemplu, dacă la intrarea divizorului de tensiune cu raportul de divizare de 1/50, se aplică o tensiune de 10 kV, la ieșirea divizorului avem 200V, tensiune mult prea mare pentru a putea fi preluată de către circuitul de intrare al plăcii de achiziție.

Funcția de transfer a divizorului secvențial de tensiune propus este prezentată în figura 3.25, unde a_1 , a_2 și a_3 sunt pantele caracteristicilor de transfer ale fiecăruia dintre cele trei divizoare de tensiune rezistive. Punctul A este intersecția caracteristicii celui de-al doilea divizor rezistiv cu dreapta $x = 500$ V, iar punctul B este intersecția caracteristicii celui de-al doilea divizor rezistiv cu dreapta $x = 2000$ V.

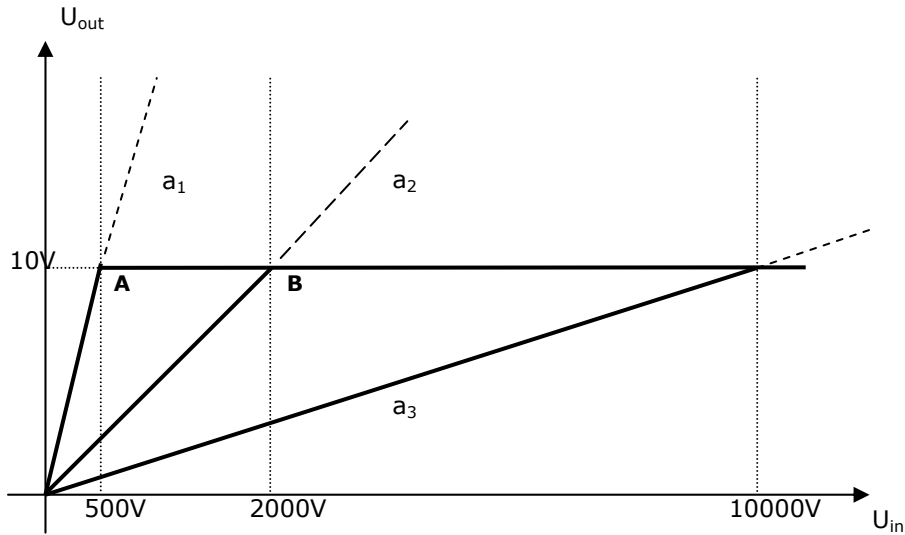


Fig. 3.25. Funcția de transfer a celor trei divizoare de tensiune utilizate

Dacă considerăm punctul A din figura 3.25 ca fiind caracterizat de perechea (x_1, y_1) și punctul B de perechea (x_2, y_2) , putem translaa cea de-a doua caracteristică cu cantitatea $c_1 = 10\text{ V} - y_1$, iar cea de-a treia caracteristică cu cantitatea $c_2 = c_1 + 10\text{ V} - y_2$, rezultând caracteristica din figura 3.26.

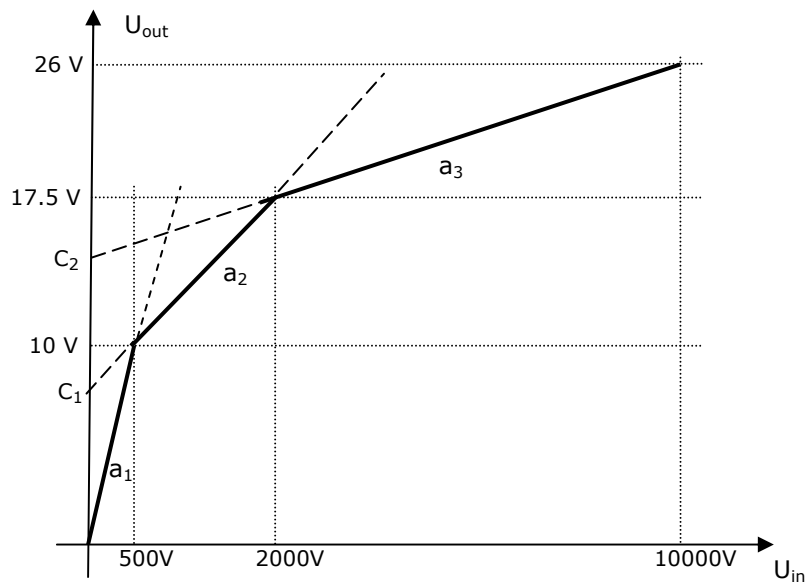


Fig. 3.26. Funcția de transfer a divizorului de tensiune secvențial

Utilizând un algoritm software, cele trei semnale de la ieșirile divizoarelor de tensiune pot fi convertite într-un singur semnal, a cărui funcție de transfer să fie cea din figura 3.26. Algoritm software calculează, la inițializare, constantele c_1 și c_2 ,

achiziționează câte un eșantion de pe fiecare canal al plăcii de achiziție și procesează eșantioanele conform algoritmului prezentat în figura 3.27.

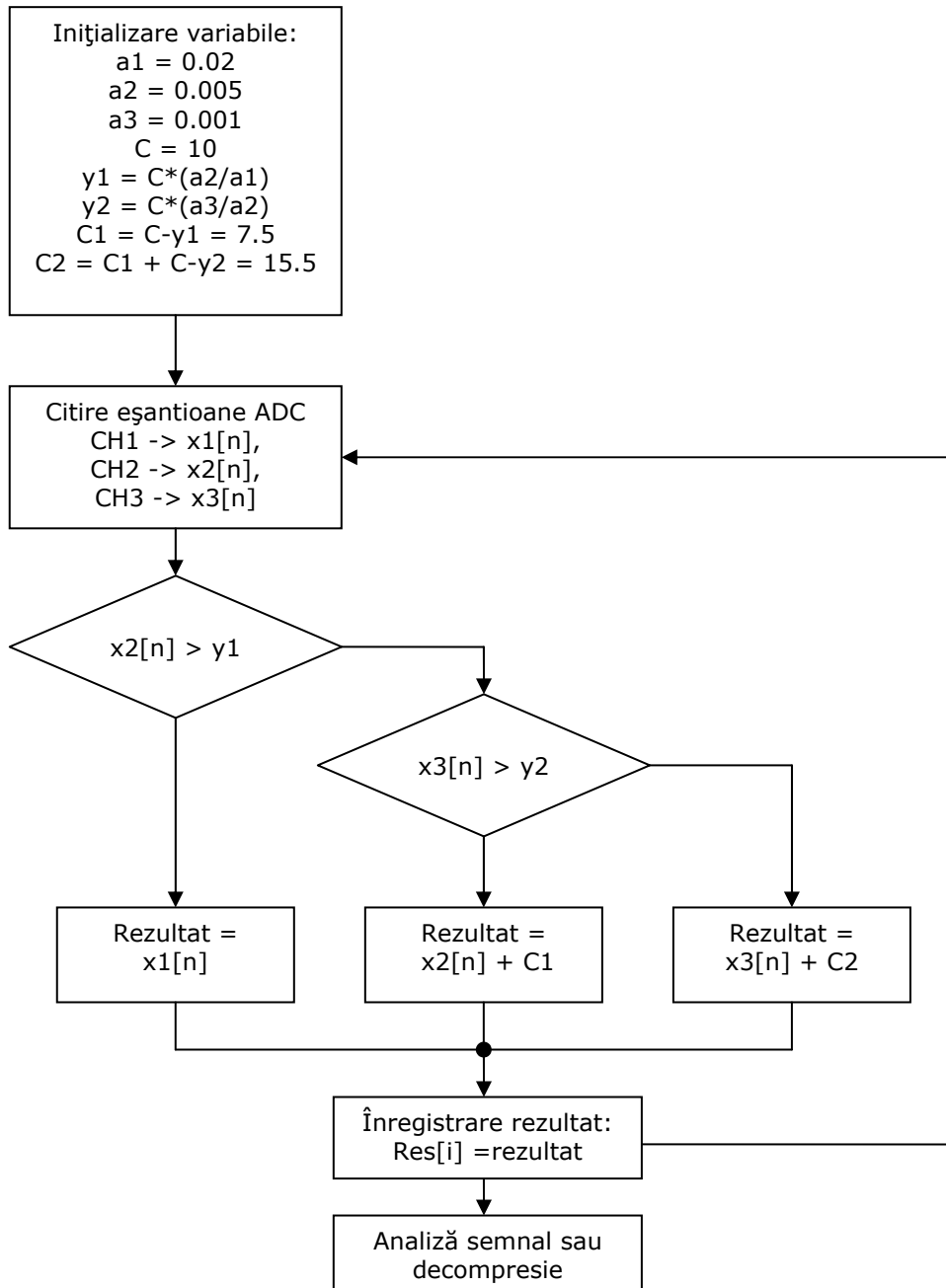


Fig. 3.27. Algoritmul software al divizorului de tensiune secvențial

Eșantioanele semnalului preluat de pe primul canal sunt folosite doar pentru porțiuni ale valorii semnalului de intrare cuprinse între -500 V și 500 V, eșantioanele preluate de pe cel de-al doilea canal fiind folosite doar în intervalul de valori ale semnalului de intrare cuprinse între 0.5 kV la 2 kV, respectiv -0.5 kV la -2 kV, iar eșantioanele preluate de pe cel de-al treilea canal vor fi folosite doar în intervalul valorilor semnalului de intrare ce depășesc ± 2 kV, lucru care aparent face să scadă performanța sistemului.

Se poate observa că dinamica semnalului este diferită de cea a circuitului de compresie cu transformator funcțional, în acest caz semnalul de la ieșirea circuitului de compresie având valori între ± 26 V ($c_1 + c_2 + 10V$). Valorile tensiunii de la intrarea convertorului analog – numeric rămân între ± 10 V, valorile de la ieșirea circuitului modificându-se doar după aplicarea algoritmului software.

Semnalul de la ieșirea divizorului secvențial poate fi folosit ca atare, sau poate fi decomprimat, rezultând un semnal direct proporțional cu tensiunea aplicată la intrarea divizorului neliniar de tensiune. Algoritmul de decompresie (figura 3.28), poate fi realizat de către un bloc de procesare independent, sau, poate fi înglobat în algoritmul de detecție a perturbațiilor.

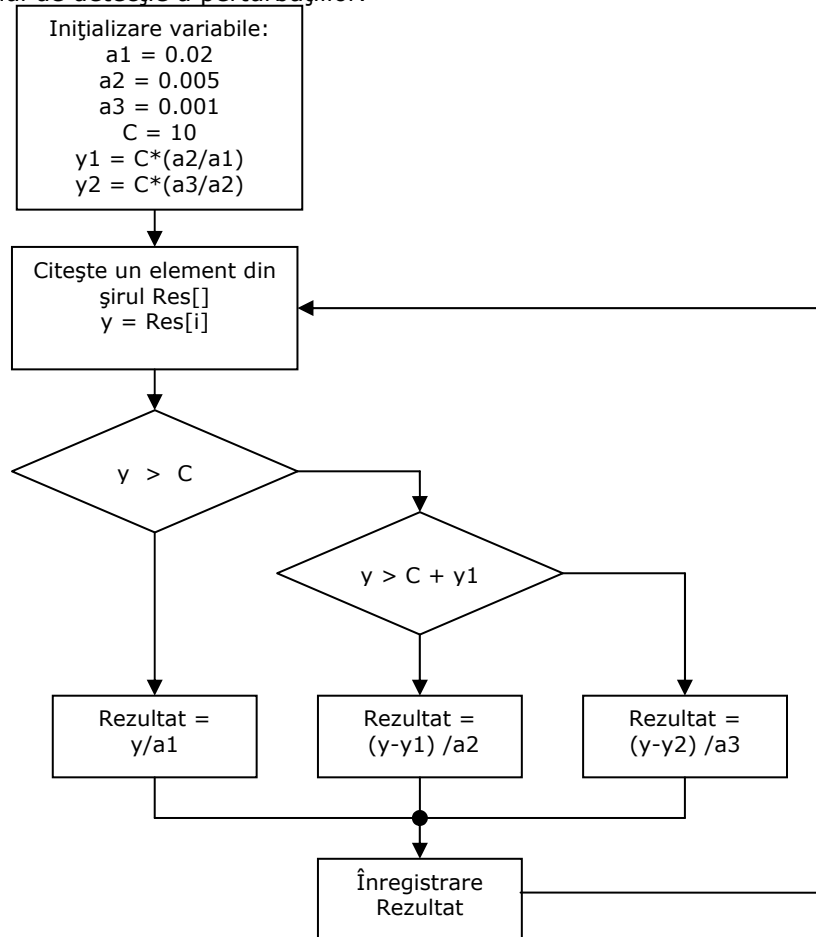


Fig. 3.28 Algoritmul de decompresie a semnalului

Decompresia semnalului se realizează pe semnalul numeric și nu pe semnalul analogic de la intrare, fapt care conduce la eliminarea circuitelor electrice complexe de compresie-decompresie a semnalului.

3.3.2 Analiza funcționării circuitului în Matlab

Funcționarea circuitului a fost analizată cu ajutorul aplicației Matlab 7. Caracteristica de transfer ideală a divizorului secvențial de tensiune simulată în Matlab este prezentată în figura 3.29a, iar caracteristica ideală a circuitului alcătuit din divizorul de tensiune secvențial și blocul de decompresie este prezentată în figura 3.29b.

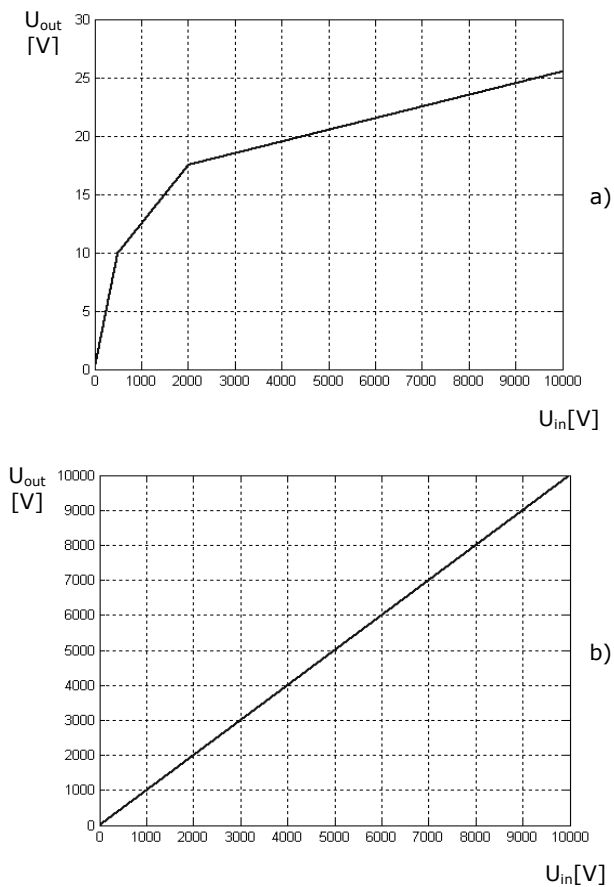


Fig. 3.29. a) Funcția de transfer a divizorului de tensiune secvențial simulată în Matlab
b) Funcția de transfer a divizorului împreună cu circuitul de decompresie

Rezultă că singurele erori care intervin în acest caz sunt erorile de conversie datorate convertorului analog-numeric.

Pentru achiziția semnalelor s-a folosit o placă de achiziție National Instruments NI PCI 6259 cu 32 de canale de intrare pe 16 biți, având o rată maximă de achiziție de 1,25 Msps. Deoarece se dorește înregistrarea cât mai fidelă a formei perturbațiilor, în special a celor tranzitorii cu amplitudini foarte mari, este necesară o rată de achiziție de

1 Msps. Pentru a elimina perturbațiile suplimentare care pot să apară la conectarea divizoarelor rezistive la placa de achiziție, este necesară folosirea modului de intrare diferențial al plăcii de achiziție. Caracteristicile semnalelor achiziționate pe cele trei canale ale plăcii de achiziție sunt prezentate în tabelul 3.3, comparativ cu numărul de biți necesari dacă s-ar utiliza cuanta de tensiune minimă. Trebuie menționat că în această situație, caracteristica de transfer a convertorului analog – numeric echivalent devine neliniară, echivalentă cu o cuantizare neuniformă.

Tabelul 3.3. Valorile și numărul cuantelor de tensiune pentru diferite intervale de valori ale semnalului de intrare

Intervalul de tensiune	0 – 0.5 kV	0 – 2 kV	0 – 10 kV
Cuanta de tensiune pentru intervalul de tensiune dat	7.63 mV (16 biți)	30.5 mV (16 biți)	152.6 mV (16 biți)
Niveluri de cuantizare pentru o cuantă de tensiune de 7.63 mV	65536 (16 biți)	262123 (18 biți)	1310616 (21 biți)

Numărul cuantelor de tensiune N_q pentru fiecare interval al tensiunilor de la intrarea divizorului secvențial, prezentate în tabelul 3.3 a fost calculat cu formula:

$$N_q = \frac{\text{domeniul de tensiune}}{\text{cuantă}} \quad (3.24)$$

unde cuanta este definită prin:

$$q = \frac{\text{domeniul de tensiune}}{2^n - 1} \quad (3.25)$$

n fiind numărul de biți ai convertorului analog – numeric. Numărul de biți n pentru un anumit număr de cuante N_q este dat de formula:

$$n = \text{int}[\log_2(QI) + 0.5] \quad (3.26)$$

Cea mai mică cuantă de tensiune (7.36 mV) se obține pentru intervalul 0 – 0.5 kV, iar cuanta maximă (152.6 mV) se obține pentru intervalul 0 – 10 kV. Prin folosirea unui singur divizor de tensiune cu domeniul de intrare cuprins între 0 și 10 kV, în cazul folosirii unui convertor analog – numeric pe 16 biți semnalele de intrare cu amplitudinea sub 0.5 kV vor avea un număr de 3276 cuante, ceea ce este echivalent cu folosirea unui convertor analog - numeric pe 12 biți.

$$N_q = \frac{0.5 \text{ kV}}{152.6 \text{ mV}} = 3276 \quad (3.27)$$

$$n = \text{int}[\log_2(3276) + 0.5] = 12 \quad (3.28)$$

Pentru a avea aceeași rezoluție ca și în cazul unui divizor simplu de tensiune având intervalul valorilor de intrare cuprinse între 0 și 0.5 kV, este nevoie (conform formulelor de mai jos) de un convertor analog – numeric pe 21 de biți.

$$N_q = \frac{10 \text{ kV}}{7.63 \text{ mV}} = 1310616 \quad (3.29)$$

$$n = \text{int}[\log_2(1310616) + 0.5] = 21 \quad (3.30)$$

În cazul divizorului secvențial de tensiune propus de autor, folosind doar o parte din intervalul de tensiuni de la intrare pe canalele 2 și 3 (tabelul 3.4) numărul efectiv de biți pentru cel de-al treilea și cel de-al patrulea interval scade cu doar un bit.

Tabelul 3.4 Numărul efectiv de biți utilizați în fiecare interval al tensiunilor de intrare

Intervalul de tensiuni de intrare	0 – 0.5 kV	0.5 – 2 kV	2 – 10 kV
Cuanta de tensiune	7.63 mV	30.5 mV	152.6 mV
Numărul cuantelor folosite în domeniul de intrare dat	65536	49180	52428
Numărul efectiv a biților utilizați	16 biți	15 biți	15 biți

Folosirea a trei convertoare analog – numerice de 16 biți în locul unuia de 21 de biți este o soluție mult mai ieftină și din punct de vedere financiar.

3.3.3 Aplicații ale divizorului secvențial de tensiune în detectarea perturbațiilor tranzitorii cu amplitudini mari

Datorită particularităților pe care le prezintă divizorul secvențial de tensiune (limitarea valorilor semnalelor de la intrarea convertorului analog – numeric la ± 10 V), el poate fi utilizat în detectarea perturbațiilor tranzitorii de tensiune cu valori peste limita superioară a tensiunii preluate de pe primul canal (cel cu intervalul tensiunilor de intrare cel mai mic) [67].

Presupunem că la un moment dat, peste semnalul rețelei electrice se suprapune o perturbație tranzitorie cu amplitudinea maximă de câteva ori mai mare decât tensiunea nominală. În figura 3.30 sunt prezentate formele de undă ale semnalelor preluate pe cele trei canale ale plăcii de achiziție, folosindu-se, pentru exemplificare, o perturbație cu amplitudinea de șapte ori mai mare decât tensiunea sinusoidală aplicată la intrare.

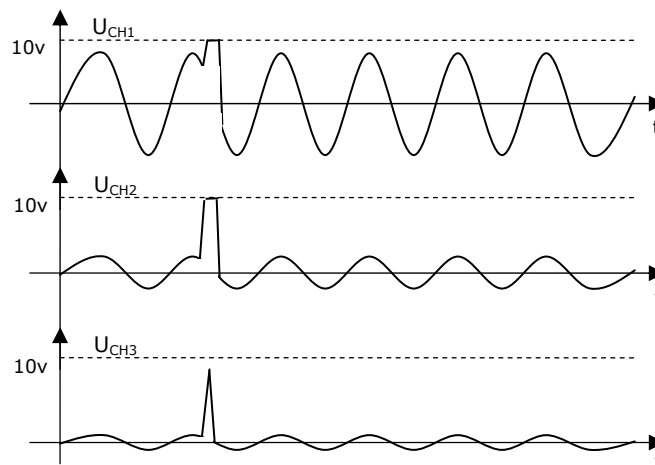


Fig. 3.30. Perturbație tranzitorie suprapusă peste semnalul din rețeaua electrică

Semnalele de pe canalele 2 și 3 ale plăcii de achiziție vor suferi o limitare la 10 V, pe când semnalul preluat de cel de-al treilea canal va avea o valoare corectă. Se poate

detecta astfel, pe baza unui simplu test asupra valorii eșantioanelor semnalelor celor trei canale, dacă există sau nu perturbații cu amplitudini mai mari decât valoarea limită acceptată la intrarea canalului cu cel mai mic raport de divizare.

Algoritmul de detecție al perturbațiilor este prezentat în figura 3.31. Semnalul de la intrarea celor trei canale ale plăcii de achiziție este preluat și înregistrat într-o memorie temporară, apoi preluat eșantion cu eșantion și analizat. Se compară eșantioanele de pe cele trei canale cu valorile limită de ± 10 V și în cazul în care există limitări ale semnalului pe unul dintre canale, blocul de decizie citește din memorie un număr de eșantioane egal cu 3 perioade ale semnalului și le înregistrează pe disc pentru a fi prelucrate/analizate ulterior.

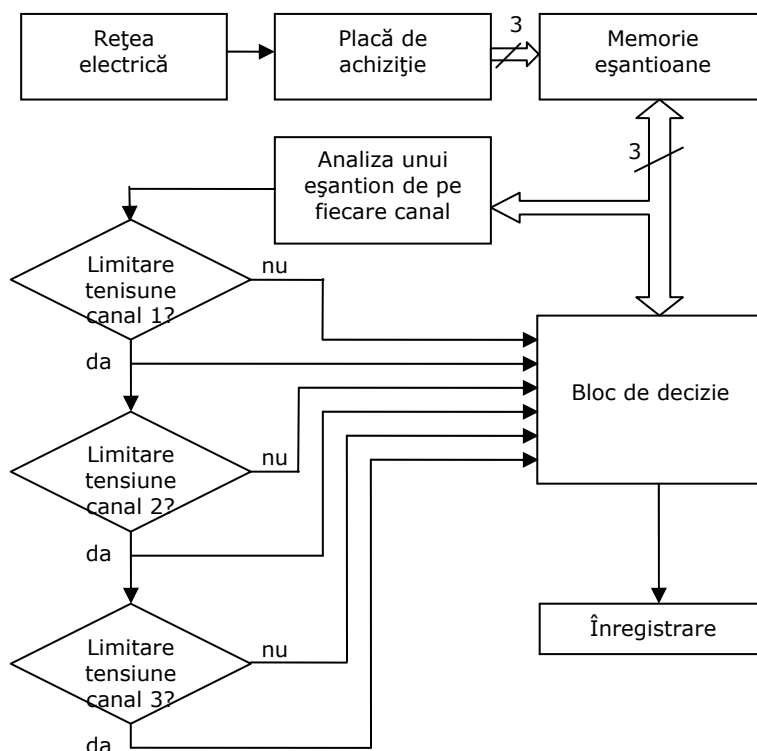


Fig. 3.31. Algoritm pentru detecția perturbațiilor tranzitorii de amplitudine mare

Testarea algoritmului a fost făcută cu ajutorul unei aplicații scrise în Microsoft Visual C++ 6, aplicație care preia semnalele de la trei dintre canalele de intrare ale plăcii de achiziție NI PCI 6254M și aplică asupra semnalelor de la intrare algoritmul prezentat mai sus.

În lipsa unui generator de semnal care să genereze perturbații cu amplitudini de 10 kV, s-a ales soluția simulării funcționării circuitului prin aplicarea de semnale cu amplitudine scăzută. S-a realizat practic un divizor secvențial de tensiune cu rapoartele de divizare 1/1, 1/4 și 1/6. La intrarea montajului a fost aplicat un semnal sinusoidal cu amplitudinea de ± 7 V, peste care s-a suprapus o perturbație tranzitorie cu amplitudinea de 50V.

Semnalele de la cele trei intrări ale plăcii de achiziție sunt reprezentate grafic în timp real în partea stângă a ferestrei aplicației (figura 3.32). De asemenea,

În momentul detectării unei perturbații, aplicația afișează pe ecran perioada perturbată (graficul marcat cu *P*), afișând totodată și evoluția în timp a indicelui de regularitate calculat pe o perioadă a semnalului (*Reg. Index*).

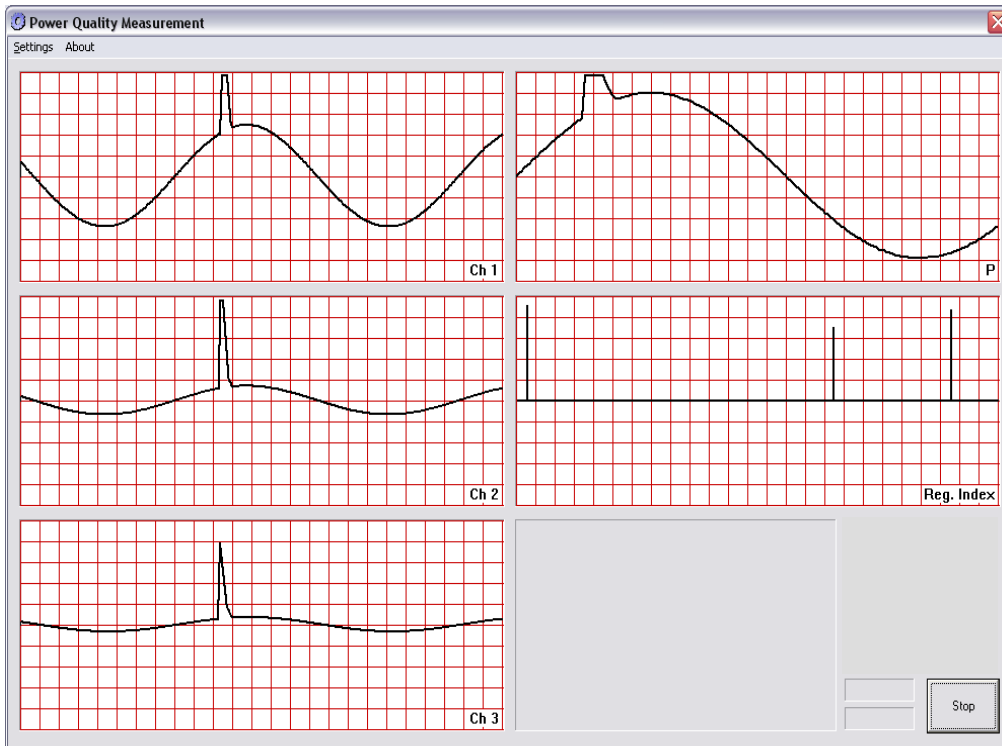


Fig. 3.32. Aplicație Visual C++ pentru testarea funcționării algoritmului de detecție a perturbațiilor tranzitorii de mare amplitudine bazat pe divizorul secvențial de tensiune

3.4 Concluzii

Circuitele de condiționare a semnalelor sunt indispensabile în realizarea unui sistem de monitorizare a calității energiei electrice, ele preluând tensiunea din rețeaua electrică și aducând-o la niveluri acceptate de către circuitele de intrare ale convertoarelor analog – numerice ale sistemelor de achiziție de date. De asemenea, aceste circuite au rolul de a realiza o compresie a semnalelor, în vederea preluării semnalelor cu o dinamică ridicată așa cum sunt semnalele din rețeaua electrică, fără a pierde detaliile fine (de exemplu variațiile de tensiune).

În acest capitol autorul prezintă, pentru început, un circuit de condiționare a semnalelor utilizând un transformator funcțional realizat cu circuite integrate, punând accentul pe problemele prezentate de acest circuit și oferind o soluție pentru rezolvarea acestora. Soluția constă în aplicarea unui semnal triunghiular la intrarea circuitului, măsurarea răspunsului circuitului și construirea unei funcții de corecție, bazată pe rezultatul obținut. Astfel, prin calibrarea circuitului, se poate îmbunătăți caracteristica de transfer a acestuia, caracteristica corectată apropiindu-se de cea proiectată.

O alternativă la acest circuit o reprezintă divizorul de tensiune secvențial prezentat în paragraful 3.2, implementat cu ajutorul a trei divizoare de tensiune rezistive cu rapoarte de divizare diferite, conectate la trei canale de intrare ale unei plăci de achiziție. Un astfel de sistem realizează o compresie a semnalului, având o caracteristică de transfer asemănătoare cu cea a transformatorului funcțional. Autorul, prin calculele efectuate, demonstrează faptul că prin conectarea acestui divizor neliniar de tensiune la trei dintre intrările unei plăci de achiziție pe 16 biți se obține, în urma prelucrărilor software, un semnal având o rezoluție echivalentă cu aceea obținută în cazul utilizării unui convertor analog – numeric pe 21 de biți.

O aplicație directă la detectarea perturbațiilor din rețelele electrice o constituie detectarea perturbațiilor cu amplitudini mai mari decât amplitudinea maximă admisă la intrarea primului canal al divizorului neliniar de tensiune. Dacă semnalul rețelei prezintă o perturbație a cărei valoare este mai mare decât valoarea limită acceptată pe primul canal (500 V în cazul prezentat), va apare o limitare a tensiunii la intrarea acestuia, care va fi sesizată prin analiza software a semnalelor provenite de pe cele trei canale de intrare, punându-se astfel în evidență prezența perturbației.

În Anexa 1 este prezentat programul PSpice cu ajutorul căruia s-a simulat caracteristica circuitului de compresie cu transformator funcțional, în Anexa 2.1 este prezentat programul MatLab elaborat pentru testarea algoritmului de calibrare a transformatorului funcțional, iar în Anexa 2.2 se prezintă programul MatLab pentru trasarea caracteristicii de transfer a divizorului secvențial de tensiune prezentată în Figura 3.29.

4. SISTEME DE ACHIZIȚIE DE DATE

4.1 Introducere

Achiziția de date este un proces de preluare a informației în formă numerică, informație care poate fi procesată de către calculatoarele personale sau sistemele cu microcontroler. Elementul cheie al unui sistem de achiziție de date este convertorul analog numeric care are rolul de a converti semnalele electrice în semnale numerice.

Componentele unui sistem de achiziție de date sunt (figura 4.1):

- *senzorii* care convertesc mărimea fizică în mărime electrică;
- *circuitele de condiționare a semnalelor* care realizează conversia semnalelor provenite de la senzori în semnale ce pot fi aplicate intrărilor convertorului analog numeric;
- *convertorul analog numeric* care transformă semnalele electrice în semnale numerice.

Aplicațiile care necesită sisteme de achiziție de date sunt de cele mai multe ori controlate de către sisteme software complexe elaborate în limbaje de programare de nivel înalt (Visual C++, Visual Basic, MatLab) sau limbaje grafice (LabView).

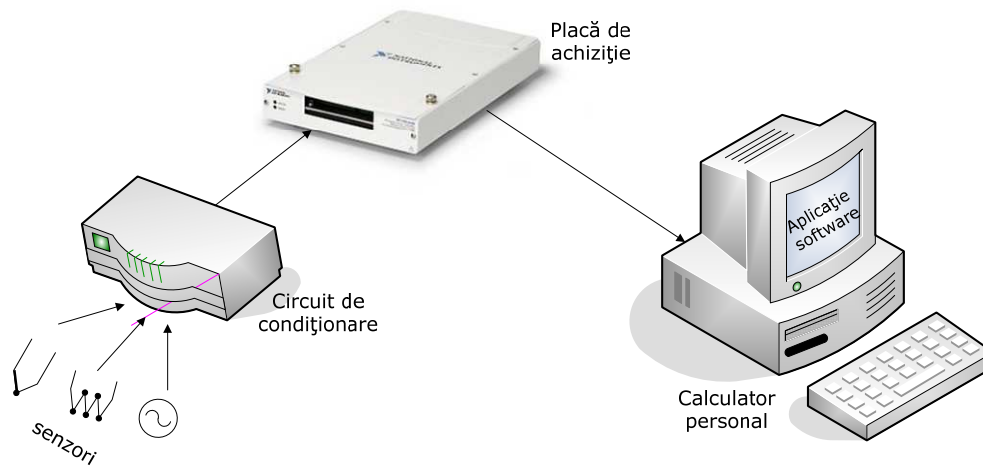


Fig. 4.1. Componentele unui sistem de achiziție de date

Frecvența de eșantionare a convertorului analog numeric trebuie aleasă de așa natură încât să fie de cel puțin 2 ori mai mare decât frecvența maximă din spectrul semnalului aplicat la intrare (Teorema lui Shannon) [68] [69]. În practică se alege frecvența de eșantionare de 5-10 ori mai mare decât frecvența maximă a semnalului de la intrare [70].

Principalii parametri ai unei plăci de achiziție sunt:

- *numărul de canale;*

- *rezoluția*: depinde de numărul de cuante ale convertorului analog numeric.
- *frecvența de eșantionare*;
- *domeniul tensiunilor aplicate la intrare*;

4.2 Placa de achiziție NI USB 6259

Pentru realizarea acestui sistem de detectare și monitorizare a perturbațiilor din rețeaua electrică s-a ales o placă de achiziție NI USB 6259 produsă de Național Instruments (figura 4.2), cu principalii parametri prezentați în tabelul 4.1 [71].

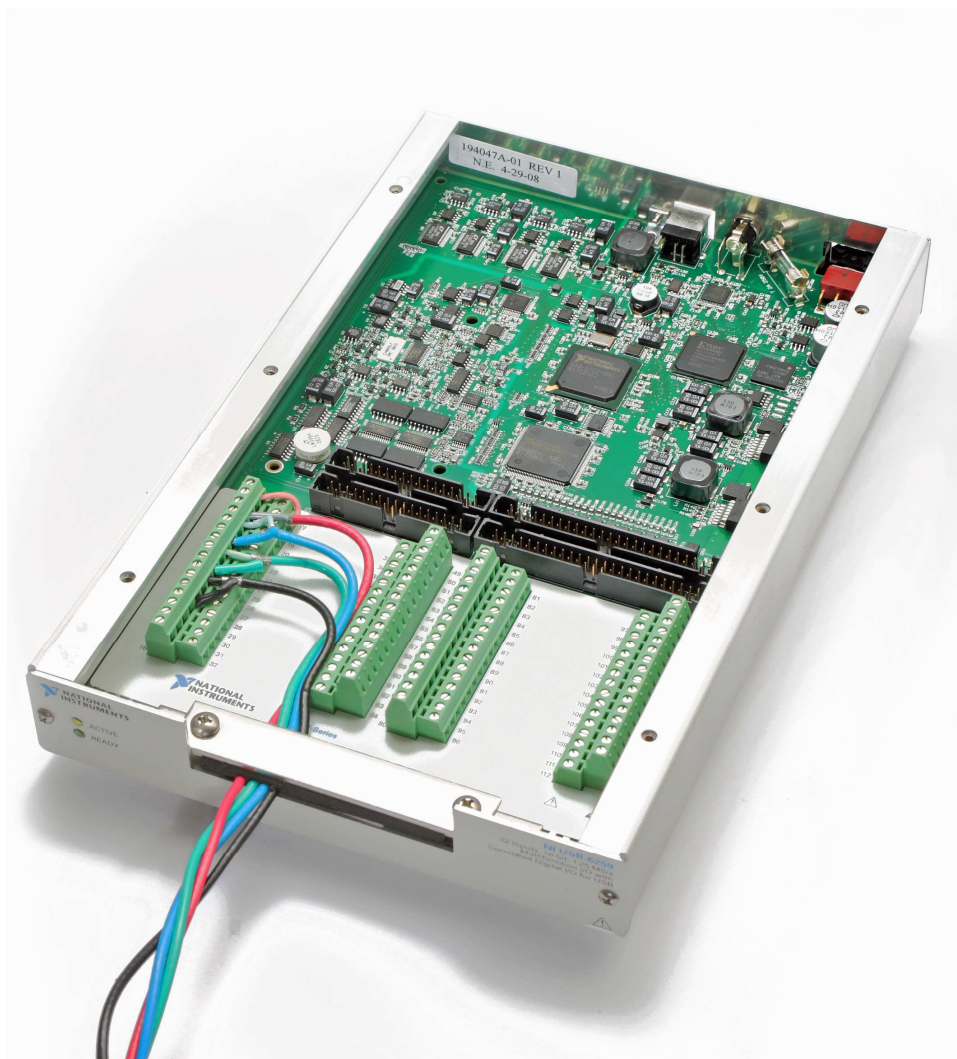


Fig. 4.2. Placa de achiziție NI USB 6259

Prezentarea succintă a performanțelor acestei plăci de achiziție de date (tabelul 4.1) este justificată pentru alegerea modului de lucru, astfel încât să corespundă cât mai performant cerințelor impuse de sistemul de monitorizare propus.

Tabelul 4.1 Principalii parametri ai plăcii de achiziție NI USB 6259

Descriere	Parametru
Intrări analogice	32 (16 diferențiale)
Tensiunea de intrare	- 10 ÷ + 10 V _{cc}
Tensiunea maximă de intrare	±11 V
Amplificarea	1,2,4,8
Rezoluția convertorului analog - numeric	16 biți
Frecvența maximă de eșantionare	1,25 Msps pe un singur canal 1 Msps pentru mai multe canale
Transferul datelor	USB 2.0
CMRR (curent continuu ÷ 60 Hz)	100 dB
Impedanța de intrare	10 GΩ 100 pF
Banda de frecvențe (-3 dB)	1,7 MHz
Memorie FIFO	4095 eșantioane

Placa de achiziție NI USB 6259 oferă utilizatorului mai multe modalități de conectare a surselor analogice de semnal:

- Modul diferențial – în acest mod de conectare nici unul dintre cele două fire de semnal nu este conectat la masa sistemului, ele fiind conectate direct la intrările neinversoare, respectiv inversoare ale amplificatorului instrumental de la intrarea plăcii de achiziție (figura 4.3). Acest mod de conectare a intrărilor de semnal contribuie la minimizarea zgomotului din sistem, în special a perturbațiilor de mod comun.

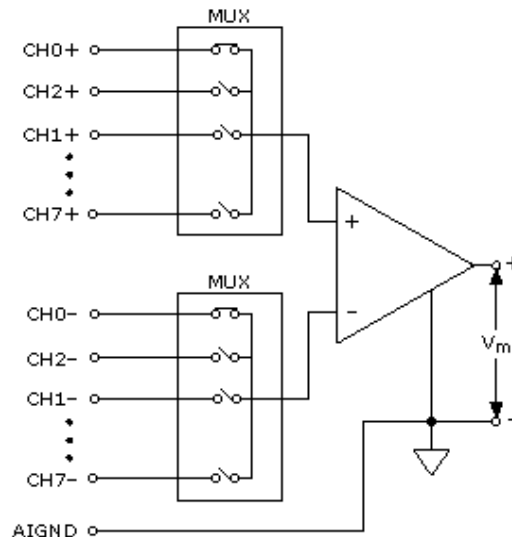


Fig. 4.3. Conectarea intrărilor în modul diferențial

- Modul unipolar (cu sau fără referință la masă) este un mod de conectare a sursei de semnal în care firul de masă al sursei de semnal este conectat la masa sistemului de achiziție (figura 4.4).

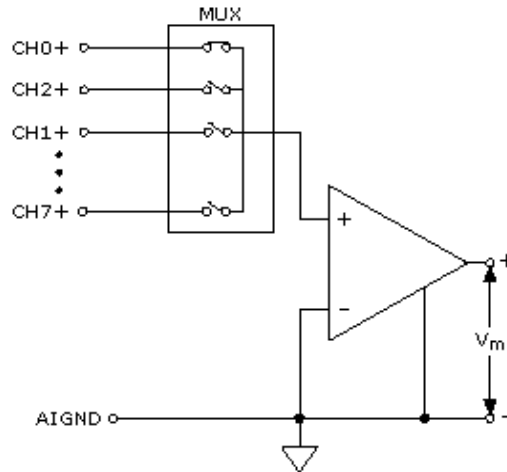


Fig. 4.4. Conectarea intrărilor în modul unipolar

Pentru a se putea conecta mai multe intrări de semnal la un singur convertor analog-numeric se folosesc, în amonte, circuite de multiplexare a semnalului analogic.

4.3 Biblioteca de funcții NI-DAQmx

Pentru controlul plăcii de achiziție, National Instruments pune la dispoziția utilizatorilor o serie întreagă de drivere și aplicații grafice, pentru a ușura munca de programare a plăcii de achiziție.

La realizarea aplicației software de monitorizare a calității energiei electrice s-a folosit biblioteca de funcții NI-DAQmx versiunea 8 și utilitarul de configurare al plăcii de achiziție *National Instruments Measurement & Automation Explorer*.

Cu ajutorul bibliotecii de funcții NI-DAQmx se pot elabora aplicații complexe în limbaje de nivel înalt, fără ca programatorul să mai fie nevoit să cunoască în profunzime structura și configurația plăcii de achiziție. Prin înlocuirea unei plăci de achiziție cu alta similară, de cele mai multe ori, grație bibliotecilor NI-DAQmx, nu trebuie făcută nici o modificare în aplicația software.

Încă de la lansarea NI-DAQmx utilizatorii sistemelor de achiziție de date produse de National Instruments au profitat pe deplin de caracteristicile sale, concepute pentru a economisi timp de dezvoltare și de a îmbunătăți performanța aplicațiilor de achiziție de date elaborate.

O caracteristică importantă a NI-DAQmx, care economisește o cantitate considerabilă de timp în dezvoltarea aplicațiilor, este NI-DAQmx Application Programming Interface (API), care oferă același set de funcții, indiferent de dispozitivul folosit sau de familia de plăci de achiziție utilizate. Toate componentele unui sistem de achiziție de date multifuncțional sunt programate cu același set de funcții (intrările analogice, ieșirile analogice, intrările și ieșirile digitale, și numărătoarele). În plus, atât o intrare/ieșire digitală cât și dispozitivele de intrare

sau ieșire analogică sunt programate folosind același set de funcții, lucru posibil datorită polimorfismului. O funcție polimorfă acceptă mai multe tipuri de date pentru unul sau mai multe dispozitive de intrare/ieșire. Biblioteca API NI-DAQmx poate fi utilizată în orice limbaj de programare Windows.

Aplicația DAQ Assistant (figura 4.5) este o unealtă software pusă la dispoziție de NI-DAQmx, care îmbunătățește viteza de dezvoltare a aplicațiilor software.

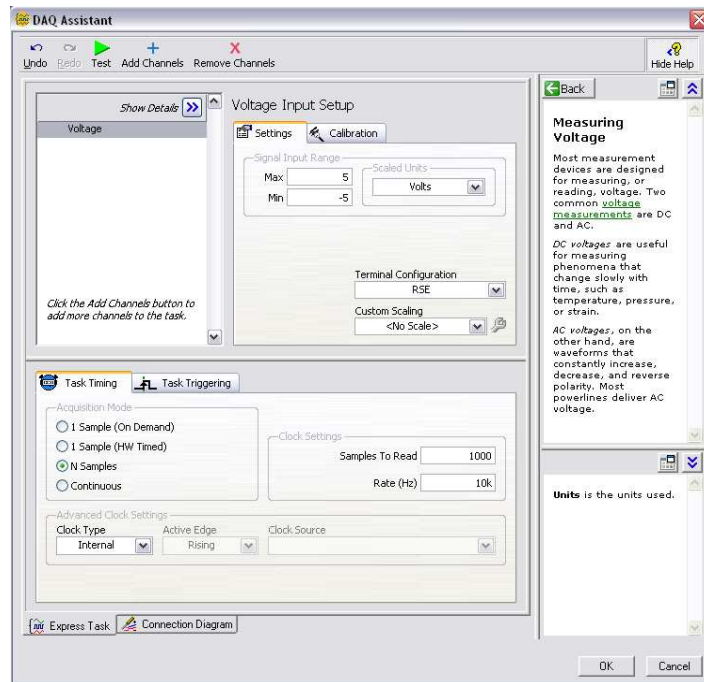


Fig. 4.5. Aplicația DAQ assistant

Acest instrument ajută utilizatorii să creeze aplicații fără a folosi limbaje de programare, printr-o interfață grafică ce permite configurarea de sarcini de achiziție de date simple sau complexe. DAQ Assistant este o interfață grafică pentru crearea, editarea și rularea interactivă de canale NI-DAQmx și sarcini virtuale.

Biblioteca de funcții NI-DAQmx definește un set de noțiuni de bază:

- un *canal virtual* NI-DAQmx constă dintr-un canal fizic asociat unui dispozitiv DAQ și informații de configurare pentru acest canal fizic, cum ar fi: gama valorilor de intrare, câștigul, modul de preluare a datelor (diferențial sau unipolar)). Canalele de intrare/ieșire pot fi specificate unul câte unul (Dev0/ai0, Dev0/ai1) sau în grup (Dev0/ai0:4 sau Dev0/ai0:Dev0/ai4).
- o *sarcină (task)* NI-DAQmx este o colecție de canale virtuale împreună cu informațiile de temporizare și declanșare a achiziției, precum și alte proprietăți privind achiziția sau generarea semnalelor.

Aplicațiile de achiziție a datelor construite folosind NI-DAQmx beneficiază de o arhitectură concepută pentru a maximiza performanța. Aceasta se bazează pe un model eficient care elimină reconfigurările inutile, fapt care ducea în trecut la încărcări inutile ale procesorului. Cu aceste reconfigurări eliminate, atât configurarea cât și achiziția sunt optimizate, nivelul de performanță atins fiind posibil datorită

utilizării registrelor de memorie mapate. Utilizând librăriile de funcții NI-DAQmx, s-a obținut o creștere a ratei de eșantionare a aplicației de monitorizare a calității energiei electrice elaborate de autor.

O altă caracteristică importantă a bibliotecii DAQmx este arhitectura sa multifilară. Deoarece NI-DAQmx este conceput ca un sistem multifilar, mai multe operațiuni de achiziție de date se pot desfășura simultan, îmbunătățind semnificativ performanța aplicațiilor care realizează simultan mai multe operații și simplificând foarte mult programarea unor astfel de aplicații.

Pentru achiziția de semnale trebuie urmați următorii pași:

1. Se creează un canal de intrare virtual cu ajutorul funcției *DAQmxCreateAIVoltageChan()*, canal virtual căruia îi este asociată o sarcină (task) creată cu funcția *DAQmxCreateTask()*.

Funcția *DAQmxCreateAIVoltageChan()* poate fi utilizată pentru crearea mai multor canale de intrare virtuale, toate aceste canale putând fi asociate aceleiași sarcini. Parametrii funcției *DAQmxCreateAIVoltageChan()* sunt:

- identificatorul activității curente;
- canalul (canalele) de intrare de pe care se face achiziția (de exemplu: „dev1/ai0”);
- un nume asociat de programator canalului virtual de intrare;
- configurația de intrare a canalului respectiv (unipolar sau diferențial);
- valoarea minimă de intrare;
- valoarea maximă de intrare;
- tipul semnalului aplicat la intrare (tensiune, curent).

Mai jos este prezentat un fragment de cod care ilustrează modul în care sunt folosite cele două funcții:

```
TaskHandle taskHandle = 0;
chan char [256] = "Dev1/ai0";
float64 min = -10, max = 10;

DAQmxCreateTask ("",& taskHandle);
DAQmxCreateAIVoltageChan( taskHandle,
                          Chan,
                          "",
                          DAQmx_Val_Cfg_Default,
                          min,
                          max,
                          DAQmx_Val_Volts,
                          NULL);
```

2. Se specifică parametrii ce controlează ceasul intern al sistemului de achiziție prin intermediul funcției *DAQmxCfgSampClkTiming()* care are ca și parametri de intrare:

- identificatorul sarcinii curente;
- tipul ceasului folosit la achiziție (intern sau extern);
- rata de eșantionare (în eșantioane pe secundă);
- frontul pe care se face sincronizarea cu semnalul de ceas (crescător sau descrescător);
- modul de achiziție (continuu, un număr finit de eșantioane sau o singură valoare);
- numărul de eșantioane conținute în memoria tampon.

În exemplul de mai jos ceasul de achiziție a fost configurat pentru o achiziție continuă de eșantioane cu rata de eșantionare de 10000 eșantioane pe secundă.

```
clockSource char [256] = "";
float64 rate = 10000;

DAQmxCfgSampClkTiming(    taskHandle,
                          clockSource,rate,
                          DAQmx_Val_Rising,
                          DAQmx_Val_ContSamps,
                          1000);
```

3. Se pune în execuție sarcina astfel creată prin apelarea funcției *DAQmxStartTask()*, al cărei unic parametru îl reprezintă sarcina pe care vrem să o lansăm în execuție. În acest moment placa de achiziție este configurată și procesul de achiziție se desfășoară așa cum a fost programat.

4. Preluarea eșantioanelor achiziționate din bufferul plăcii de achiziție se face cu funcția *DAQmxReadAnalogF64()*. Parametrii acestei funcții sunt:

- identificatorul sarcinii curente;
- numărul de eșantioane ale unui canal;
- timpul de așteptare pentru terminarea achiziției numărului prestabilit de eșantioane;
- modul în care sunt returnate eșantioanele: eșantioanele fiecărui canal putând fi returnate separat sau intercalat;
- zona de memorie în care vor fi înscrise eșantioanele achiziționate și dimensiunea ei.

Preluarea eșantioanelor achiziționate nu se face automat, ea căzând în sarcina programatorului. O secvență tipică de citire a eșantioanelor din bufferul plăcii de achiziție este următoarea:

```
TaskHandle    taskHandle=0;
int32         read;
float64       data[1000];

DAQmxErrChk (DAQmxCreateTask("",&taskHandle));

DAQmxReadAnalogF64( taskHandle,
                    1000,
                    10.0,
                    DAQmx_Val_GroupByChannel,
                    data,
                    1000, &read,
                    NULL);
```

prin care sunt citite câte 1000 de eșantioane, într-un interval maxim de așteptare de 10 secunde, eșantioanele fiind grupate după canalul de pe care provin.

Pentru a putea diagnostica și depana problemele ce pot să apară pe parcursul execuției programului, NI-DAQmx pune la dispoziția programatorilor un set de funcții cu ajutorul cărora se facilitează detectarea apariției unei erori și interpretarea corectă a acesteia. Fiecare funcție din biblioteca Ni-DAQmx returnează, în urma execuției sale, un cod de eroare. Pentru o execuție fără erori, codul returnat este zero. Pentru a putea cunoaște semnificația codului de eroare

returnat este apelată funcția *DAQmxGetErrorString()* care returnează un șir de caractere conținând o explicație succintă a codului de eroare respectiv. Pentru a afla mai multe informații despre respectiva eroare și pentru a obține sugestii de depanare a problemei se poate apela funcția *DAQmxGetExtendedErrorInfo()*.

Deoarece, unele tipuri de perturbații (impulsurile tranzitorii, armonicile) pot avea componente spectrale cu frecvențe de sute de kHz, este necesară utilizarea unei rate de achiziție ridicate, de preferat peste 1 Msps.

4.4 Concluzii

Încă de la apariția primelor echipamente electronice, s-a pus problema vizualizării și înregistrării formelor de undă ale diferitelor semnale. Primele sisteme de monitorizare a energiei electrice foloseau ca dispozitive de afișare a perturbațiilor osciloscoape cu remanență [72], sau înregistratoare cu hârtie. De atunci, tehnica a evoluat foarte mult, ajungându-se astăzi la utilizarea de sisteme de achiziție de date rapide și performante.

În zilele noastre, sistemele de achiziție de date reprezintă o componentă esențială în construcția unui sistem de monitorizare a calității energiei electrice. Fără ele nu am putea prelua și stoca informațiile sub formă numerică. De parametrii sistemului de achiziție de date depind, în bună măsură, parametrii de funcționare și de performanță ai sistemului de monitorizare și detectare a perturbațiilor.

De-a lungul timpului s-au făcut progrese semnificative în ceea ce privește performanțele convertoarelor analog-numeric, crescându-se numărul de biți și rata de eșantionare. Producerea pe scară largă a sistemelor de achiziție de date care să integreze convertoare analog-numeric pe 16 și chiar 24 de biți, cu o rată de eșantionare de peste 1 Msps, nu mai reprezintă o problemă.

Pentru a ușura munca utilizatorilor care doresc să integreze sistemele de achiziție de date în echipamentele lor, producătorii de sisteme de achiziție de date pun la dispoziția acestora medii de dezvoltare software pentru programarea și utilizarea sistemelor de achiziție de date în parametri optimi, permițând obținerea maximului de performanță. Utilizând biblioteci de funcții specializate, programatorii pot realiza aplicații de achiziție de date într-un timp foarte scurt sau pot modifica aplicațiile existente, nemaifiind necesară cunoașterea arhitecturii interne a plăcii de achiziție folosite. De asemenea, datorită driverelor puse la dispoziție de producători, aplicațiile pot rula fără probleme chiar și în cazul înlocuirii unei plăci de achiziție cu un model diferit, dar cu parametri asemănători.

5. ALGORITMI SOFTWARE PENTRU DETECTAREA PERTURBAȚIILOR DIN REȚELELE ELECTRICE DE JOASĂ TENSIUNE

5.1 Introducere

În proiectarea echipamentelor pentru monitorizarea calității sistemelor electrice de putere se folosesc diverși algoritmi pentru detectarea și caracterizarea perturbațiilor. Mulți algoritmi folosesc transformatele Fourier sau Wavelet pentru detecția și caracterizarea perturbațiilor, lucru care necesită foarte multe resurse din partea sistemului de calcul utilizat. Există la ora actuală diverse sisteme de detecție și monitorizare a perturbațiilor, unele dintre ele folosesc calculatoare personale, altele utilizează sisteme embedded [73], sau procesoare digitale de semnal (DSP) [74], [75], [76], [77], [78], [79].

Metodele cele mai uzuale citate în literatură se bazează pe transformarea Fourier [80], Wavelet [81], [82] sau pe utilizarea rețelelor neuronale [83].

Alte metode încearcă să detecteze apariția unor perturbații în domeniul timp [84], cele mai multe dintre ele bazându-se pe faptul că semnalul sinusoidal este un semnal a cărui medie pe o perioadă este zero. Astfel, dacă se adună eșantioanele corespunzătoare unei perioade a semnalului, rezultatul va fi zero sau o valoare foarte apropiată de zero. În cazul în care apare o perturbație tranzitorie, suma eșantioanelor corespunzătoare unei perioade va conduce la o valoare diferită de zero. În literatura de specialitate se indică o astfel de metodă [85] în care se presupune frecvența rețelei ca fiind egală cu frecvența nominală, de aici rezultând un inconvenient al metodei și anume că la modificarea frecvenței rețelei electrice suma eșantioanelor care se realizează pe o perioadă egală cu 20 ms variază în funcție de frecvența rețelei.

Alături de alte țări, România a început să acorde o atenție din ce în ce mai mare asigurării calității energiei electrice, specialiști din mai multe domenii ocupându-se de monitorizarea calității energiei electrice.

5.2 Determinarea apariției perturbațiilor tranzitorii

În continuare se prezintă o metodă pentru detecția perturbațiilor tranzitorii prezentată în literatură [85], metodă care va permite o comparație cu metoda propusă de autor.

Blocul de detectare a evenimentelor alarmează subsistemul dedicat caracterizării perturbației la apariția unei perturbații tranzitorii (figura 5.1).

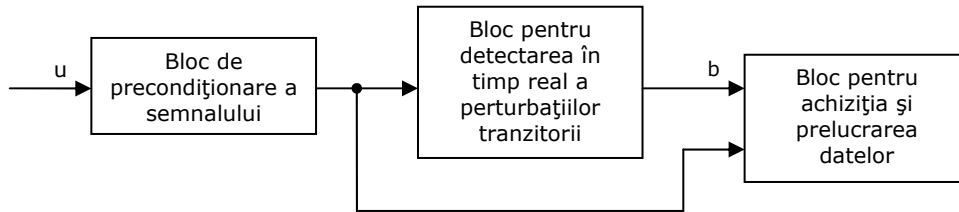


Fig. 5.1. Schema bloc a sistemului pentru detectarea perturbațiilor tranzitorii

În continuare vom nota cu $u(t)$ tensiunea liniei electrice a cărei calitate dorim să o monitorizăm. După cum se poate observa și din figura 5.1, tensiunea $u(t)$, după ce este atenuată de blocul de condiționare a semnalului, este apoi aplicată simultan blocului de detectare în timp real a perturbațiilor și blocului de achiziție și prelucrare. Blocul de detectare a perturbațiilor generează la ieșire un semnal dreptunghiular b care ne informează dacă semnalul $u(t)$ este afectat sau nu de perturbații. În cazul apariției unui eveniment, ieșirea detectorului este activată și semnalul b va comanda achiziția semnalului de către blocul de achiziție (care funcționează continuu, dar salvează datele achiziționate doar pe durata cât semnalul de la ieșirea blocului de detecție este activ pentru a economisi spațiul de stocare). Astfel soluția propusă necesită foarte puțină memorie pentru stocarea datelor și un volum foarte mic de calcul.

Algoritmul constă în folosirea unui indice de regularitate r definit în domeniul timp discret astfel:

$$r = \sum_{n=1}^N u[n] \quad (5.1)$$

unde cu n s-a notat indicele eșantionului curent, N reprezintă lungimea secvenței $u[n]$, iar $u[n]$ reprezintă semnalul de intrare $u(t)$ eșantionat pe un interval de observare egal cu perioada semnalului T sau multiplu de T .

Indicele r poate fi utilizat pentru detectarea perturbațiilor tranzitorii, deoarece pentru un semnal $u(t)$ în condiții stabile, el este constant, pe când atunci când apare o perturbație el variază.

Algoritmul se bazează pe un principiu simplu și anume acela de a împărți semnalul în mai multe segmente al căror număr îl vom nota cu η și verificarea indicelui de regularitate pe fiecare segment în parte. Întotdeauna, fiecare din cele η segmente are același număr N_t de eșantioane, $N_t = N/\eta$.

În figura 5.2 este reprezentat procesul în timp real utilizat pentru construcția segmentelor prin gruparea elementelor din secvența de intrare $u[n]$. Se observă că această tehnică este similară decimării în timp a lui $u[n]$ în timpul acestui proces. În exemplul din figura 5.2, $N = 16$, $\eta = 4$.

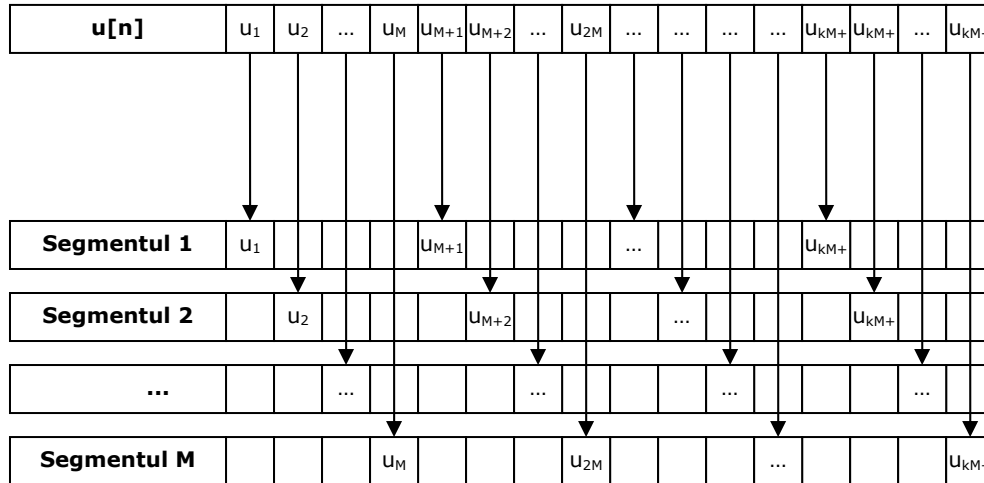


Fig. 5.2. Tehnica segmentării

Vom presupune, pentru simplificare, (dar fără a pierde din generalitate), că $u(t)$ este un semnal alternativ sinusoidal și este caracterizat de indicele de regularitate r_s evaluat pentru cel de-al s -lea segment. Pentru fiecare indice de regularitate r_s se va verifica dacă $r_s=0$, atunci nu există evenimente, iar dacă $r_s \neq 0$ atunci avem de-a face cu un eveniment tranzitoriu. Rezultatele astfel obținute pentru indicele de regularitate r_s sunt analizate și se determină dacă semnalul este sau nu perturbat.

Este evident că observând și analizând indicele de regularitate pe segmente și nu pe întregul semnal $u[n]$, probabilitatea de a obține o informație corectă despre producerea perturbației tranzitorii crește. Această probabilitate crește cu numărul de segmente η utilizate.

Algoritmul trebuie să îndeplinească următoarele cerințe:

- Semnalul care este supus testării, exceptând perturbația, trebuie să fie periodic, nefiind necesară nici o altă informație asupra formei sale de undă;
- Algoritmul poate fi folosit și în cazul în care frecvența semnalului diferă ușor față de frecvența nominală;
- În cazul în care durata perturbației tranzitorii este mai mare decât intervalul de studiu, la ieșirea blocului de detecție va rezulta un singur semnal de comandă pentru blocul de achiziție, semnal care va avea durata cel puțin egală cu durata perturbației;
- Algoritmul de detecție are mari șanse de reușită chiar și în cazurile în care perturbația are valoare medie nulă, deoarece este puțin probabil ca în toate cele η segmente ea să aibă valoarea medie nulă. Dacă se mărește numărul de eșantioane, scade probabilitatea ca un asemenea tip de eveniment să nu poată fi detectat. Numărul maxim de segmente este din păcate limitat de necesitatea evitării apariției fenomenului de aliere la calculul lui r_s (cu cât crește numărul de segmente cu atât scade frecvența de eșantionare a semnalului

corespunzător unui segment existând pericolul scăderii acesteia sub frecvența Nyquist).

- Algoritm ar trebui să poată să detecteze apariția a două sau mai multe perturbații tranzitorii care nu modifică indicele de regularitate al semnalului original.

Algoritm propus în [85] este ilustrat prin ordinograma din figura 5.3. Conținutul variabilelor este actualizat la fiecare măsurare și eșantionul curent este imediat adăugat variabilei corespunzătoare segmentului respectiv pentru a actualiza valoarea indicelui r_s . Deoarece algoritmul nu necesită memorarea nici unuia din eșantioanele semnalului $u(t)$, el memorând doar valorile curente ale indicilor r_s (vectori de lungime prestabilită), rezultă că el nu este consumator de memorie și nici de putere de calcul, calculele pe care trebuie să le efectueze procesorul fiind simple.

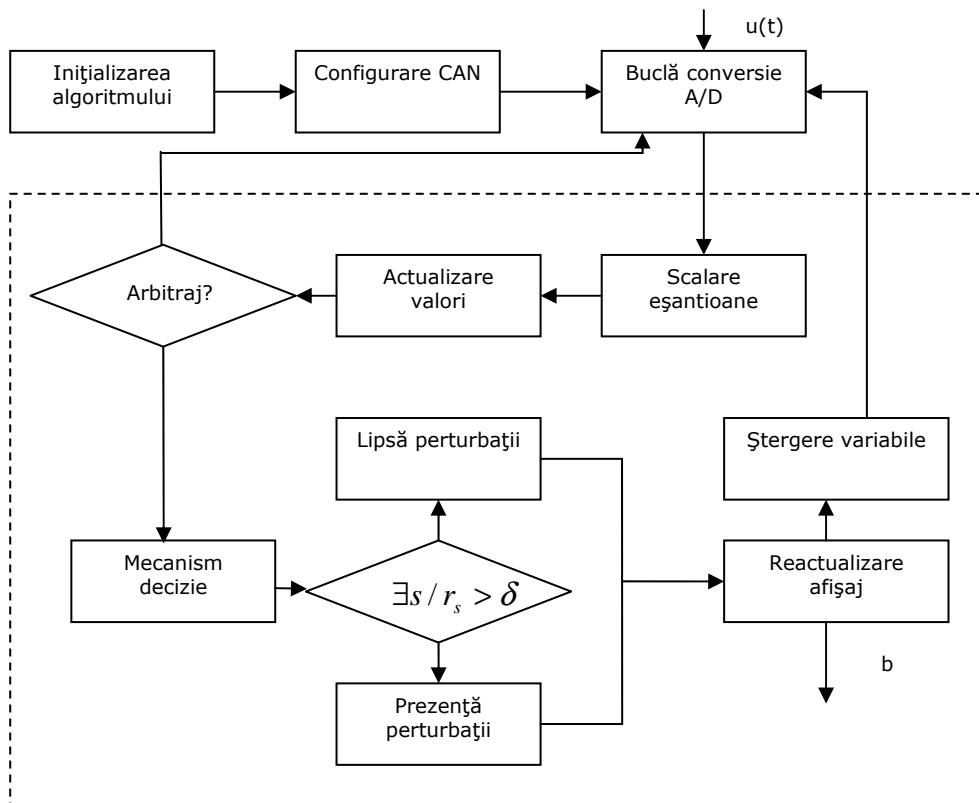


Fig. 5.3 Ordinograma algoritmului

Arbitrajul se face la sfârșitul fiecărui interval de observare care este egal cu perioada semnalului T (în cazul nostru 20 ms), pentru a decide dacă s-a produs într-adevăr un eveniment sau nu. Semnalul logic b de la ieșirea blocului de detectare a perturbațiilor este activat dacă în timpul ultimei perioade de observație a avut loc un eveniment. Trebuie amintit aici că surprinderea începutului evenimentului (care este semnalizat cu întârziere de o perioadă de observație), este sarcina blocului de achiziție și nu a blocului de decizie. Prin memorarea eșantioanelor corespunzătoare

ultimei perioade de observație de către blocul de achiziție se poate remedia și această problemă.

Privind strict din punctul de vedere al algoritmului, nu există diferențe între a considera semnalul $u(t)$ alternativ sau nu. Din punct de vedere practic însă, lucrul cu semnale alternative este mult mai simplu. De altfel, dacă semnalul nu este alternativ, se va modifica și condiția de detectare a perturbației, astfel că indicele de regularitate r_s nu mai poate fi comparat cu 0, ci cu o cantitate δ care va depinde de natura semnalului, precum și de caracteristicile etajului de intrare al plăcii de achiziție folosite. Astfel, blocul va sesiza un eveniment dacă $r_s > \delta$. Semnalul logic b de la ieșirea detectorului este activat dacă măcar unul dintre segmente are indicele de regularitate mai mare decât δ . Pentru a maximiza eficiența acestui algoritm el trebuie implementat într-un limbaj de nivel înalt care să permită optimizări de viteză ale codului executabil.

Din cauza efectului diverșilor factori (cum ar fi zgomotul, tensiunea de offset a etajelor de intrare), semnalul digital $u[n]$ poate să nu fie 100 % alternativ, spre deosebire de semnalul $u(t)$, ceea ce conduce la rezultate eronate în detectarea perturbației tranzitorii. De aceea este necesar ca indicii de regularitate să nu fie comparați cu 0, ci cu δ .

Numărul segmentelor η nu este arbitrar, el fiind dictat de o serie de factori:

- Un număr mic de segmente conduce la scăderea capacității de memorie necesară și la scăderea frecvenței de eșantionare, dar în același timp trebuie avut în vedere că sub o anumită valoare pot să apară depășiri de domeniu în calculele efectuate;
- Cu cât este mai mare numărul de segmente, cu atât este mai mare probabilitatea de a detecta perturbațiile tranzitorii care nu afectează indicele de regularitate al semnalului. Trebuie stabilită o limită superioară pentru η pentru a minimiza numărul de alarme false care crește cu creșterea lui η . În urma analizelor efectuate, a rezultat $\eta = 10$.

În figura 5.4 este prezentată fereastra principală a aplicației Visual C++ elaborată pentru testarea algoritmului propus. Este prezentat semnalul de intrare și semnalele corespunzătoare a 2 segmente alese de către utilizator. În partea dreaptă a ecranului sunt prezentați indicii de regularitate ai fiecărui segment.

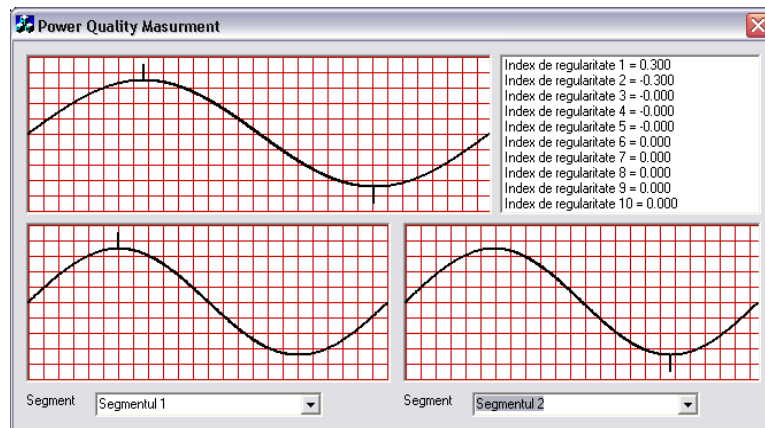


Fig. 5.4 Aplicație software pentru împărțirea pe segmente și calculul indicilor de regularitate pe fiecare segment

5.3 Îmbunătățirea metodei de detectare a perturbațiilor tranzitorii

Metoda expusă în paragraful 5.2 nu ne poate furniza informații precise asupra prezenței sau absenței perturbațiilor în condițiile în care frecvența rețelei de alimentare se modifică față de frecvența nominală.

În cele ce urmează se va prezenta o metodă nouă, originală [86], care constă în introducerea unui bloc de detectare a începutului, respectiv a sfârșitului de perioadă și calculul indicilor de regularitate pe o perioadă întreagă a semnalului. Această abordare a problemei asigură un indice de regularitate minim, indiferent de frecvența semnalului aplicat la intrare. În cadrul metodei prezentate anterior sunt folosite două canale de achiziție: unul folosit de blocul de detectare a perturbațiilor tranzitorii și celălalt fiind folosit de blocul de achiziție a semnalelor.

În figura 5.5, blocul de condiționare al semnalului din figura 5.1 a fost înlocuit aici de un bloc de compresie a semnalului, bloc care realizează o compresie neliniară a semnalului aplicat la intrarea sa, cu scopul de a aduce perturbațiile de nivel ridicat (10 kV) la o valoare care să poată fi aplicată plăcii de achiziție fără însă a atenua în aceeași măsură semnalele cu amplitudine mică (zeci de volți, sute de volți).

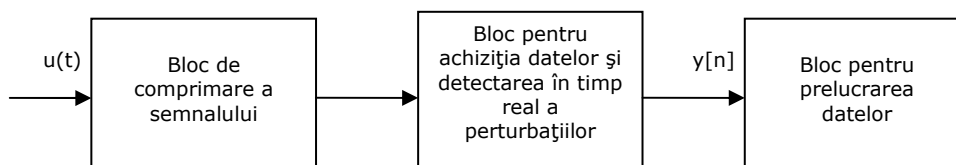


Fig. 5.5 Schema bloc a sistemului pentru detectarea perturbațiilor tranzitorii modificat

O contribuție originală constă în folosirea unui singur canal de achiziție și folosirea aceluiași eșantioane atât pentru detectarea perturbațiilor tranzitorii cât și pentru prelucrările ulterioare [87]. Prin folosirea unui singur canal se reduce mult volumul de calcul și gradul de ocupare al procesorului [88].

Decompresia semnalului se va face ulterior în cadrul blocului de prelucrare a datelor. Semnalul obținut în urma compresiei este eșantionat, cuantizat și apoi stocat într-o memorie circulară care are capacitatea de a stoca câteva perioade ale semnalului.

Memoria circulară este o zonă de memorie al cărui pointer de sfârșit coincide cu pointerul de început al zonei de memorie. Avantajul utilizării memoriei circulare este acela că în ea se poate scrie și citi: în și din zone diferite, putând fi utilizată asemenea unui circuit de întârziere controlabil.

Programul realizat în Visual C++ pentru implementarea funcției de bloc de detectare a perturbațiilor conține două fire de execuție care lucrează în paralel: unul se ocupă de achiziția semnalului citind într-o buclă un număr determinat de eșantioane și înscriindu-le în memoria circulară, iar celălalt fir de execuție se ocupă cu preluarea unui număr de eșantioane egal cu o perioadă a semnalului și calculul indecșilor de regularitate pentru fiecare segment în parte, proces urmat apoi de luarea unei decizii în ceea ce privește existența sau nu a unei perturbații. Deoarece eșantioanele corespunzătoare perioadei(lor) semnalului afectate de perturbații există deja în memoria circulară, nu mai este necesar un proces separat pentru achiziția acestora așa cum era nevoie în cazul metodei inițiale. Eșantioanele respective sunt preluate direct din memoria circulară împreună cu alte două perioade precedente. Acest lucru este necesar

datorită faptului că indicii de regularitate sunt calculați abia pe durata celei de-a doua perioade a semnalului perturbat.

Cantitatea δ cu care sunt comparați indicii r_s se determină pentru un semnal neperturbat, ea putând fi influențată de natura semnalului aplicat la intrare, de offsetul etajelor de intrare și de distorsiunile armonice introduse de acestea.

În figura 5.7 este prezentată diagrama algoritmului modificat de autor pentru detectarea perturbațiilor.

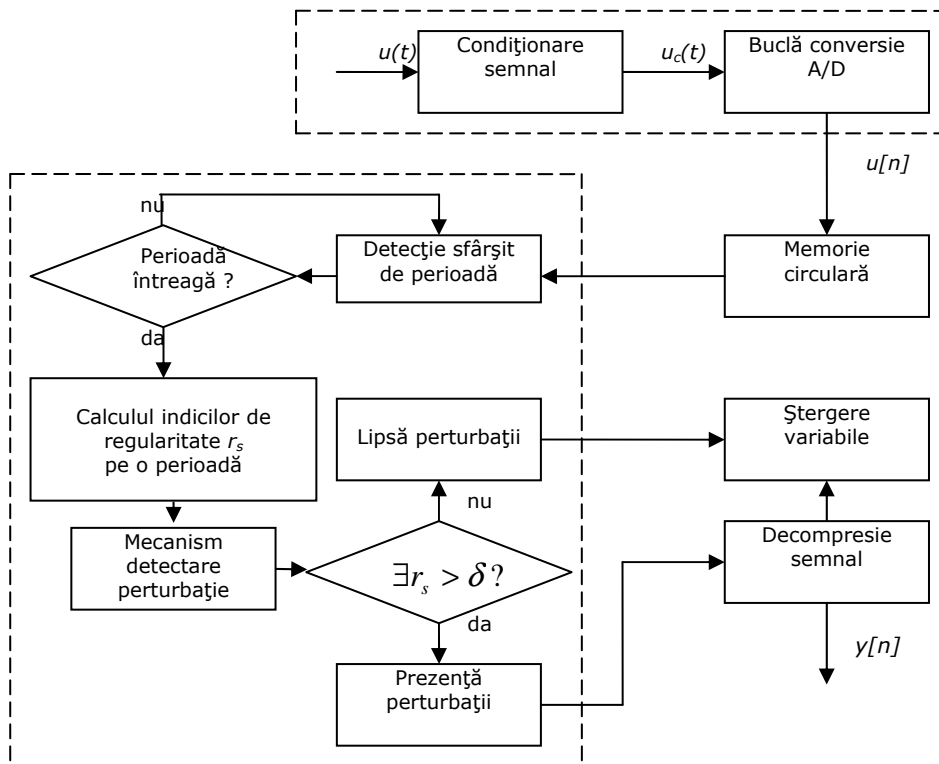


Fig. 5.6 Ordinograma algoritmului modificat

În continuare se studiază modul în care variază indicii de regularitate cu frecvența pentru metoda prezentată în paragraful 5.2 și pentru metoda propusă de autor. Deoarece în cadrul metodei inițiale calculul indicilor de regularitate nu se face întotdeauna pe o perioadă întreagă a semnalului, rezultă că semnalul asupra căruia se efectuează calculul indicilor de regularitate nu este un semnal cu medie nulă. În acest caz, indicii de regularitate pot lua diverse valori, în funcție de frecvența semnalului aplicat la intrare. După cum se poate vedea și din figura 5.7 indicele de regularitate are valori apropiate de zero doar pentru valori ale frecvenței semnalului egale cu frecvența nominală a rețelei electrice sau pentru multipli ai acesteia.

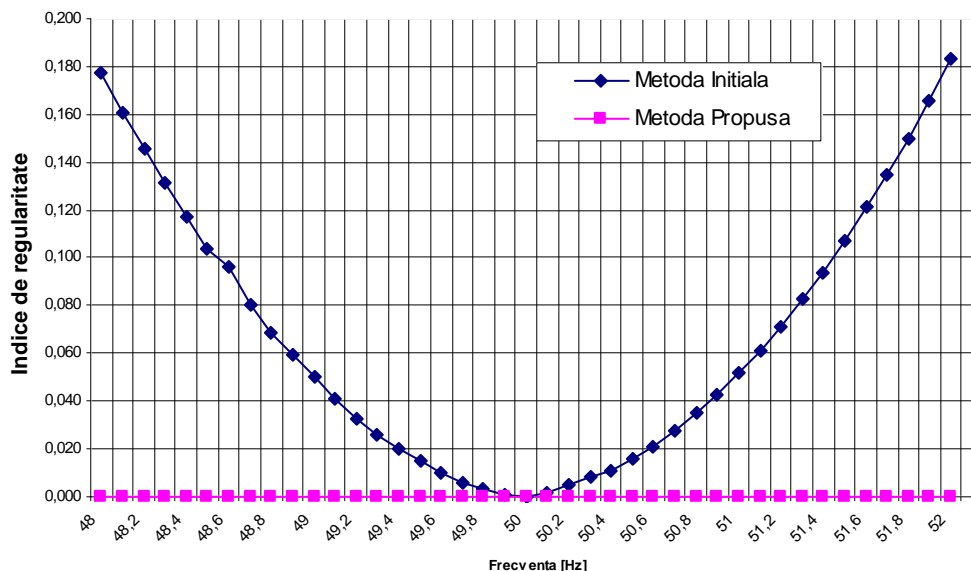


Fig. 5.7 Variația indicelui de regularitate cu frecvența

Această variație a indicelui de regularitate cu frecvența conduce la erori destul de mari în aprecierea prezenței sau absenței perturbațiilor odată cu îndepărtarea de frecvența nominală a rețelei de alimentare. Algoritmul propus de autor reușește să păstreze indicii de regularitate la o valoare apropiată de zero indiferent de frecvența semnalului, în condițiile în care nu există perturbații.

Pentru a testa algoritmul s-au aplicat diverse tipuri de perturbații: impulsuri singulare, zgomot aditiv cu distribuție uniformă și semnale sinusoidale de diferite frecvențe.

Testarea s-a realizat cu ajutorul unui program implementat în Visual C++. Motivul alegerii acestui limbaj l-a constituit viteza mare de execuție a codului compilat cu ajutorul acestui limbaj și multitudinea de funcții matematice puse la dispoziție. În anexa 1 este prezentat codul sursă al programului implementat în C++.

Programul cuprinde următoarele blocuri:

- blocul de achiziție/generare a semnalului;
- blocul de generare de zgomot cu distribuție uniformă;
- blocul de calcul al indicilor de regularitate.

Pe interfața aplicației pot fi distinse trei controale grafice, scopul acestora fiind acela de a afișa semnalul de intrare (în partea de sus a ecranului) și a oricăror două segmente specificate de utilizator prin selectarea lor din combo-box-urile corespunzătoare (în partea de jos a ferestrei). În partea dreaptă a ferestrei sunt afișați indicii de regularitate ai fiecărui segment în parte.

Vom studia în continuare cazul apariției unor perturbații de tip impuls singular. Pentru început se va studia cazul unui singur impuls pe durata unei perioade a semnalului (figura 5.8).

Menționez că segmentele reprezentate au o formă puțin diferită față de semnalul captat ca urmare a folosirii unui număr mai mic de eșantioane (v. par. 5.2).

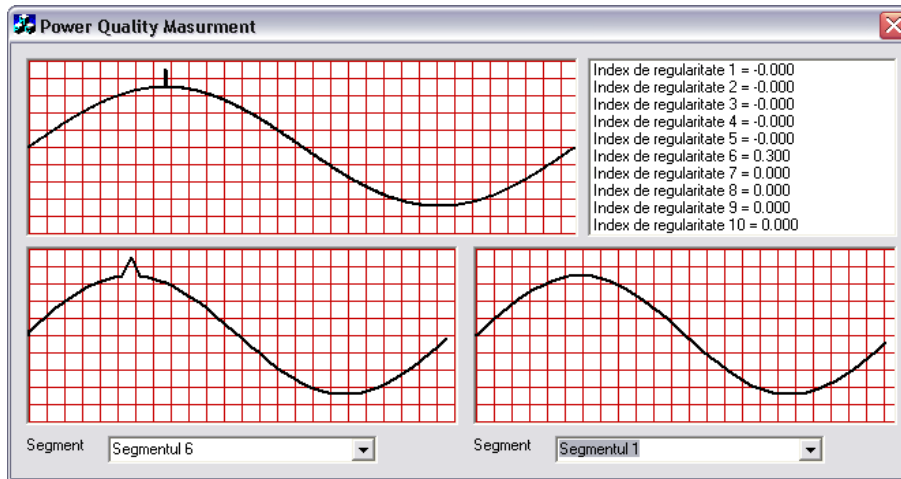


Fig. 5.8 Semnal sinusoidal cu perioada de 20 ms peste care s-a suprapus o perturbație de tip impuls

Se poate observa că perturbația a fost surprinsă în cadrul segmentului 6 al cărui indice de regularitate are valoarea 0.3, toți ceilalți indici având valori nule. Poziția perturbației în raport cu începutul, respectiv sfârșitul perioadei nu are importanță în acest caz.

Dacă însă avem două astfel de impulsuri plasate unul pe semialternanța pozitivă, celălalt pe semialternanța negativă (figura 5.9), lucrurile se complică deoarece există posibilitatea ca valorile celor două impulsuri, în cazul în care ele sunt identice și se găsesc în cadrul aceluiași segment, să se anuleze reciproc, programul neșemnalând în acest caz producerea unei perturbații.

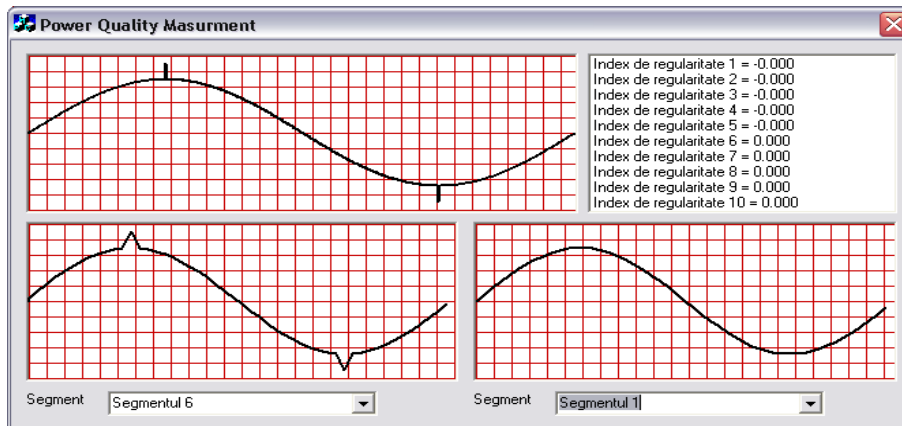


Fig. 5.9 Semnal sinusoidal cu perioada de 20 ms peste care s-a suprapus o perturbație de tip impuls dublu

Ambele impulsuri aplicate s-au presupus având durata unui singur eșantion, caz foarte puțin probabil în practică. În consecință, vom presupune impulsurile ca având o anumită durată (3-5 perioade ale frecvenței de eșantionare).

Se poate observa o valoare nenulă a indicelui de regularitate al segmentului 7, indicând prezența unei perturbații (figura 5.10).

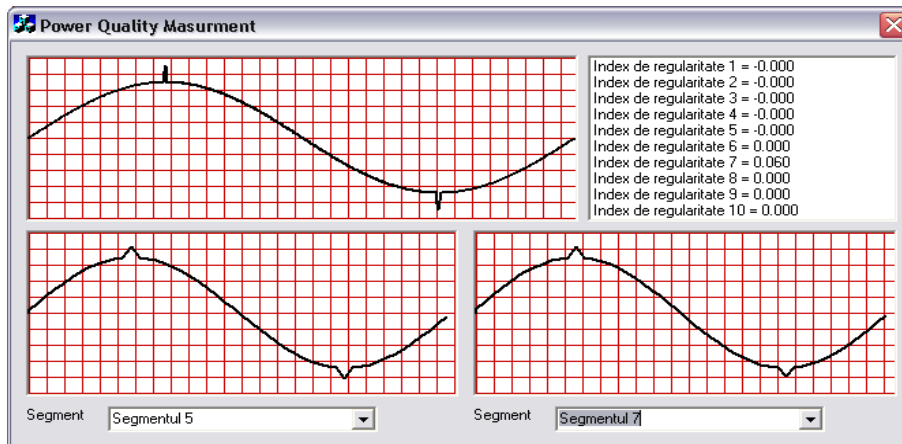


Fig. 5.10 Semnal sinusoidal cu perioada de 20 ms peste care s-a suprapus o perturbație de tip impuls dublu

Valoarea este mică și este datorată procesului de eșantionare-cuantizare care în acest caz ne este de un real ajutor deoarece putem detecta prezența acestei perturbații.

În figura 5.11 se prezintă analiza unui semnal de intrare sinusoidal peste care s-a suprapus un zgomot generat cu ajutorul unui program MatLab. Se pot observa valorile diferite pentru indicii de regularitate ale diferitelor segmente, fapt ce conduce la o detectare sigură a perturbației.

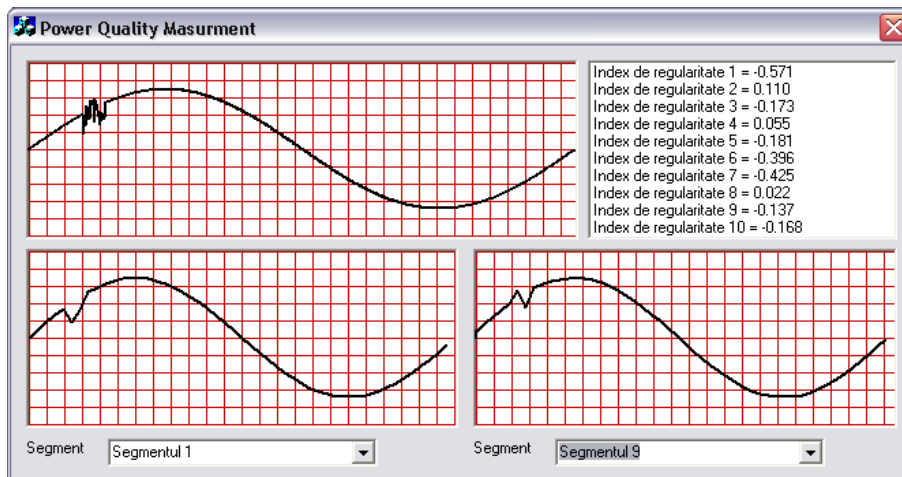


Fig. 5.11 Semnal sinusoidal cu perioada de 20 ms peste care s-a suprapus un zgomot

Pentru a putea determina mai bine apariția unui astfel de tip de perturbație este nevoie de lucrul cu un număr mai mare de segmente. Din păcate, așa cum s-a

arătat mai sus, nu se poate extinde numărul de segmente la infinit din cauza apariției fenomenului de aliere.

5.4 Concluzii

Datorită faptului că algoritmul propus operează întotdeauna asupra unei perioade a semnalului aplicat la intrare, s-a reușit menținerea indicilor de regularitate la o valoare foarte apropiată de zero indiferent de frecvența fundamentalei semnalului aplicat la intrare, lucru realizat prin detectarea cu precizie a perioadelor semnalului rețelei.

Problema utilizării unui singur canal la achiziția semnalului a fost rezolvată prin utilizarea unei memorii circulare în care sunt stocate câteva perioade ale semnalului. Spre exemplu, la o frecvență de eșantionare de 100 kHz o perioadă cuprinde un număr de 2000 de eșantioane. Astfel că, pentru memorarea unui număr de 10000 de eșantioane este suficientă o memorie de 20 KB (semnalul a fost eșantionat cu o rezoluție de 16 biți/eșantion).

De-a lungul timpului, au fost elaborate nenumărate lucrări științifice care tratează detectarea perturbațiilor din rețelele electrice.

Autorul a ales pentru detectarea perturbațiilor din rețelele electrice metode de analiză în domeniul timp, deoarece acestea necesită sisteme de calcul cu performanțe relativ modeste, putându-se astfel crește rata de eșantionare a semnalului rețelei, fapt ce conduce la creșterea frecvenței maxime din spectrul semnalului achiziționat. Pe de altă parte, memorarea și stocarea informațiilor referitoare la perturbațiile apărute, permit o prelucrare și analiză ulterioară a acestora.

6. SISTEM DE DETECTARE ȘI MONITORIZARE A PERTURBAȚIILOR DIN REȚELELE ELECTRICE

6.1 Introducere

Deoarece majoritatea standardelor prevăd măsurarea parametrilor rețelelor electrice pe o semiperioadă a tensiunii, autorul a încercat dezvoltarea aplicației de detectare și monitorizare a perturbațiilor electrice pe o semiperioadă a tensiunii rețelei. Toți parametrii rețelei electrice (valoare efectivă, valoare maximă, frecvență), sunt determinați pe o semiperioadă a semnalului și calculați conform standardelor în vigoare.

6.2 Prezentarea sistemului hardware

Sistemul de detecție și monitorizare a perturbațiilor cuprinde următoarele blocuri (figura 6.1):

- blocul de condiționare a semnalelor;
- placa de achiziție;
- calculator personal

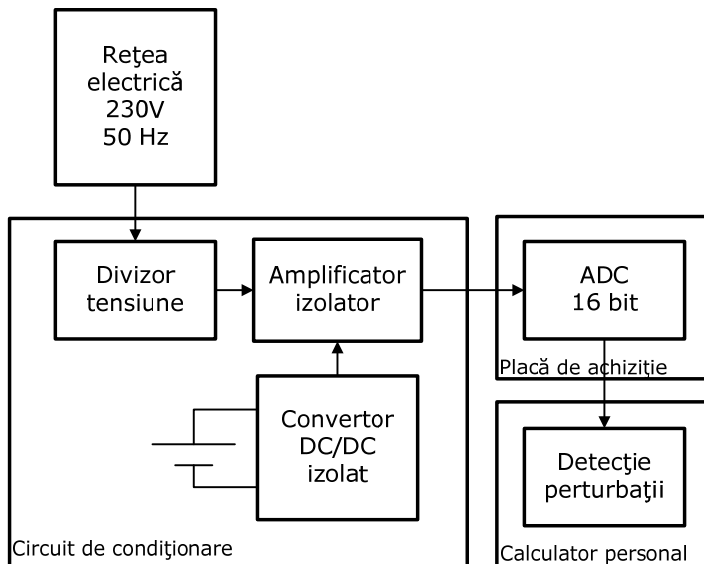


Fig. 6.1. Diagrama sistemului de monitorizare al perturbațiilor

Circuitul de condiționare este compus din patru module de intrare independente (figura 6.2), care preiau tensiunea din rețeaua electrică și o aduc la o

valoare acceptată de placa de achiziție, realizând totodată și o izolare a convertorului analog – numeric de rețeaua electrică.

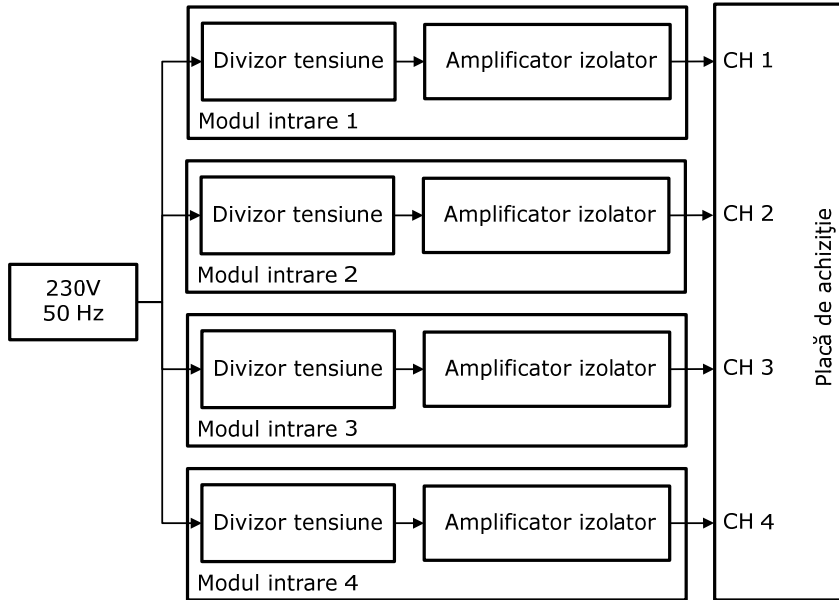


Fig. 6.2. Circuitul de condiționare a semnalelor

Circuitul de intrare a fost proiectat ca un circuit modular compus din 4 module de intrare și partea de alimentare/conectare a acestora la sistemul de achiziție de date. S-au folosit patru module de intrare, în ideea unor dezvoltări ulterioare ale sistemului pentru circuite trifazate, nefiind întotdeauna necesară folosirea tuturor celor patru module, după cum se va arăta în continuare. Fiecare divizor de tensiune poate fi configurat să aibă un anumit raport de divizare, acest lucru fiind posibil datorită rezistorului R_2 (figura 5.3) care se poate schimba rapid, fiind prevăzut cu borne de contact. În plus, cu ajutorul jumperului $JP1$, rezistența R_2 se poate înlocui cu o rezistență semireglabilă multitură, acest lucru facilitând reglajul exact al raportului de divizare. Pentru asigurarea izolării etajului de intrare față de rețeaua electrică (pentru a proteja utilizatorul de un eventual risc de electrocutare și pentru a preveni apariția buclilor de masă între rețea și masa sistemului de calcul), s-a folosit un amplificator izolator ISO 122U [89]. Principalii parametri ai acestui circuit integrat sunt prezentați în tabelul 6.1.

Tabelul 6.1. Parametrii amplificatorului izolator ISO 122U

Parametru	Valoare
Tensiunea de izolare	1500 V _{AC}
Impedanța de barieră	$10^{14} \Omega \parallel 2 \text{ pF}$
Câștigul	$1 \pm 0.05 \% \text{FSR}$
Tensiunea de intrare	$\pm 10 \text{ V}$
Tensiunea de ieșire	$\pm 10 \text{ V}$
Banda de frecvențe	50 kHz
Slew-rate	2 V/ μs
Tensiune de alimentare	$\pm 15 \text{ V}$

Deoarece amplificatorul izolator ISO 122 necesită două alimentări separate cu $\pm 15\text{ V}$, una pentru circuitele de intrare, iar cealaltă pentru circuitele de ieșire, s-a adoptat soluția utilizării unui convertor DC/DC izolat pentru alimentarea părții de intrare a ISO 122, acest lucru conducând la un bloc de intrare complet izolat. Convertorul DC/DC izolat folosit este DCH010515N7 [90], ai cărui principali parametri sunt prezentați în tabelul 6.2.

Tabelul 6.2. Parametrii circuitului integrat DCH010515N7

Parametru	Valoare
Tensiunea de izolare	3 kV
Eficiența	78 %
Tensiunea de intrare	+5 V
Tensiunea de ieșire	$\pm 15\text{ V}$
Puterea de ieșire	1 W
Frecvența de chopare	70 kHz

În figura 6.3 este prezentată schema electrică a unui modul de intrare, iar în figura 6.4 este prezentată o imagine a acestuia.

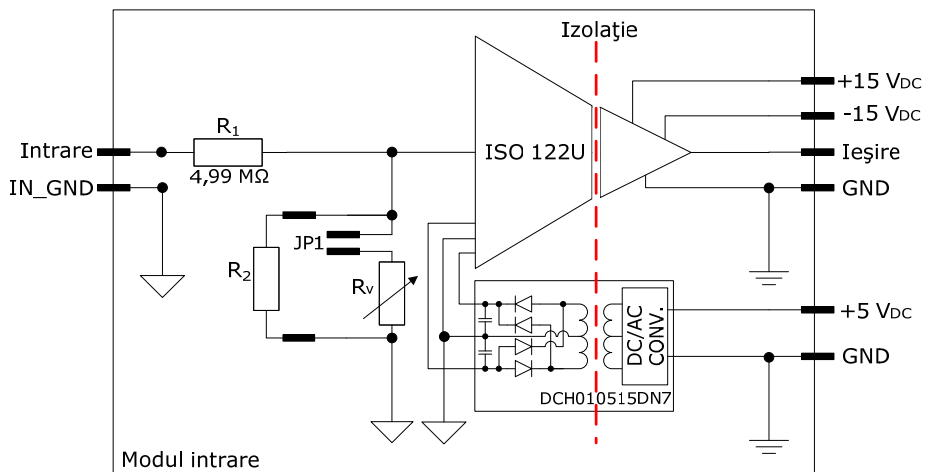


Fig. 6.3. Modul de intrare izolat

Prin plasarea pe cablajul imprimat (figura 6.4) a amplificatorului izolator și a convertorului DC/DC izolat, s-a realizat o zonă de izolare figurată punctat în figura 6.3. Per ansamblu, etajul de intrare asigură o izolare de minim 1,5 kV (pentru un raport de divizare de 1:1) și tipic 75 kV (pentru un raport de divizare de 1:50).

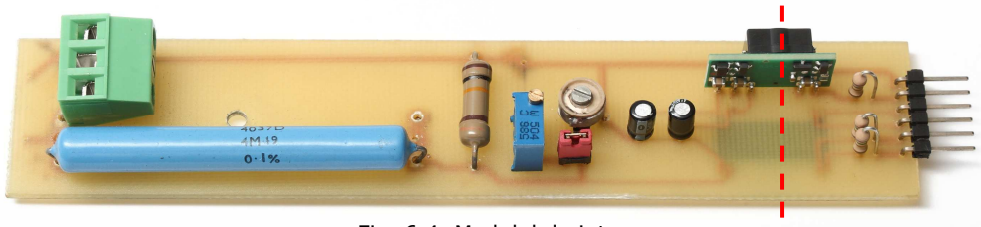


Fig. 6.4. Modulul de intrare

În figura 6.5 se prezintă modul în care sunt conectate modulele de intrare la intrările, respectiv ieșirile sistemului de condiționare. Se dorește alimentarea întregului ansamblu circuit de condiționare – placă de achiziție de la o baterie de 12 V pentru ca sistemul să funcționeze independent de tensiunea rețelei.

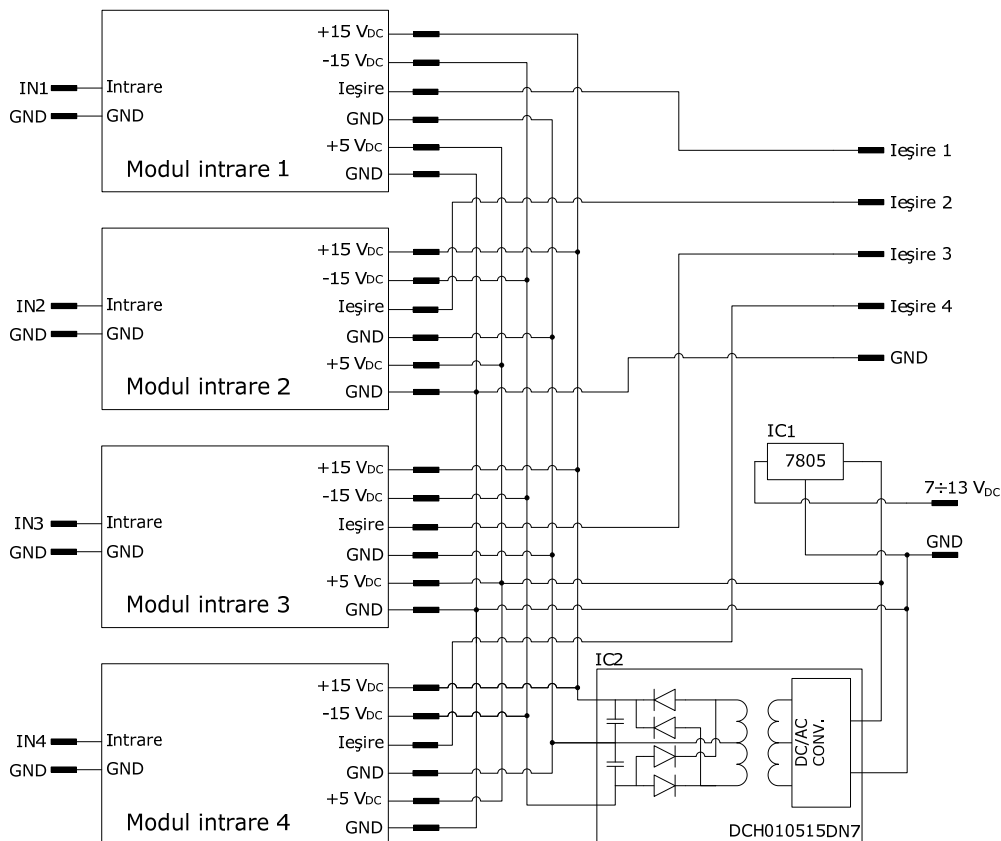


Fig. 6.5. Diagrama de conectare a modulelor de intrare

Rolul celor două circuite integrate IC1 (LM 7805) și IC2 (DCH010515DN7) din schemă este de a obține o tensiune de +5 V stabilizată, pentru alimentarea convertoarelor DC/DC și cele două tensiuni de +15 V, respectiv -15 V, necesare alimentării părții de ieșire a amplificatoarelor izolatoare. Masa semnalelor de la intrare este separată de masa semnalelor de la ieșire.

Datorită modularității sale, circuitul de intrare poate fi conectat în mai multe moduri la placa de achiziție în funcție de ceea ce se dorește să se măsoare.

În figura 6.6 se prezintă o imagine de ansamblu a circuitului de condiționare a semnalelor, putându-se observa modul în care sunt conectate modulele la circuitul de măsurat și la circuitul de alimentare. Circuitul este montat într-o carcasă din PVC cu grosimea de 5 mm, pentru a realiza o bună izolare electrică. Pentru a elimina pe cât este posibil perturbațiile ce pot să apară în circuit acesta este ecranat, fiind introdus într-o cutie metalică realizată din tablă de 1 mm.

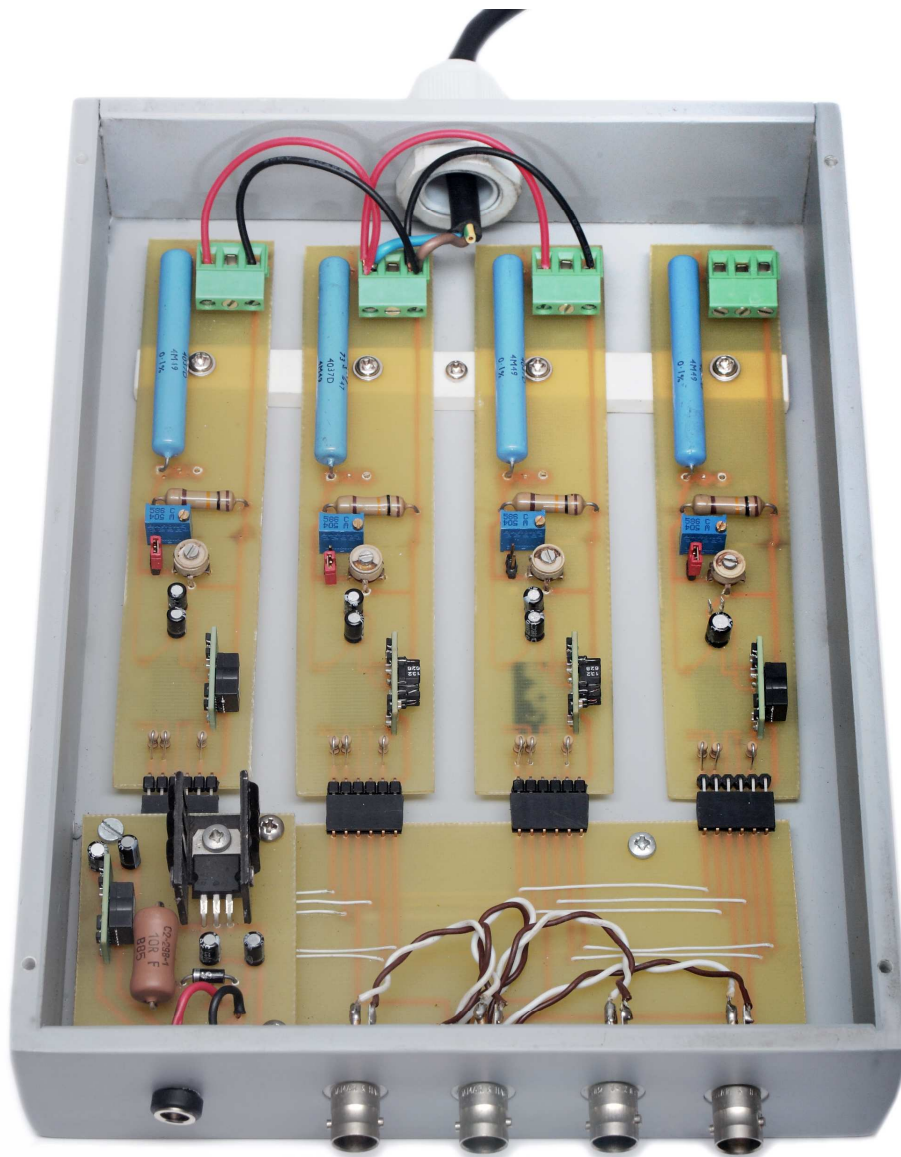


Fig. 6.6. Circuitul de condiționare a semnalelor

Pentru monitorizarea unei rețele de joasă tensiune de 230V și 50 Hz, intrările celor patru module de intrare se pot conecta în paralel, fiecare modul având un alt raport de divizare (pentru a se realiza un divizor de tensiune secvențial similar cu cel prezentat în capitolul 3), iar ieșirile celor patru module (în cazul folosirii unui divizor de tensiune secvențial cu patru segmente) sunt legate la placa de achiziție după cum se arată în figura 6.7.

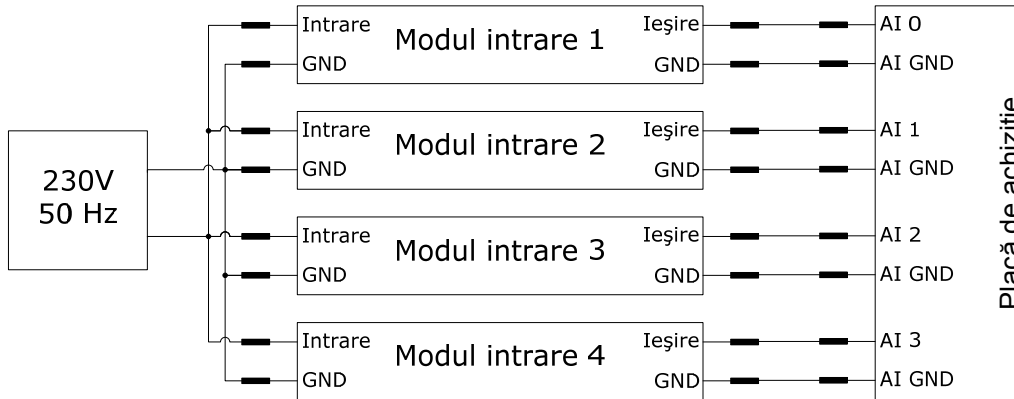


Fig. 6.7. Conectarea circuitului de intrare în modul masă comună

Dacă se dorește eliminarea perturbațiilor ce pot să apară pe traseul circuit de condiționare – placă de achiziție, pe lângă ecranarea cablurilor care leagă cele două subansambluri, se recomandă folosirea intrărilor plăcii de achiziție în mod diferențial (figura 6.8).

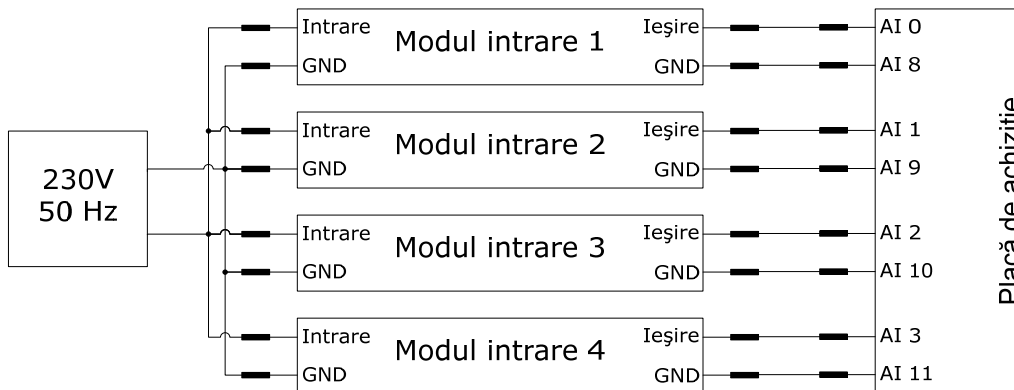


Fig. 6.8. Conectarea circuitului de intrare în modul diferențial

Pentru utilizarea modului de intrare diferențial, placa de achiziție trebuie programată în consecință, lucru care este realizat de către aplicația software prin intermediul driverelor NI-DAQmx.

Sistemul poate să monitorizeze două rețele de alimentare diferite (figura 6.9), spre exemplu, una dintre ele poate fi rețeaua națională, iar alta poate fi o rețea locală alimentată de la panouri solare, energie eoliană sau o minihidrocentrală. Se poate urmări astfel modul în care invertorul, respectiv generatorul este capabil să se sincronizeze cu rețeaua națională, și se pot studia diferențele care apar în ceea ce privește forma tensiunii, respectiv diferența de frecvență și defazajul dintre tensiunile celor două rețele.

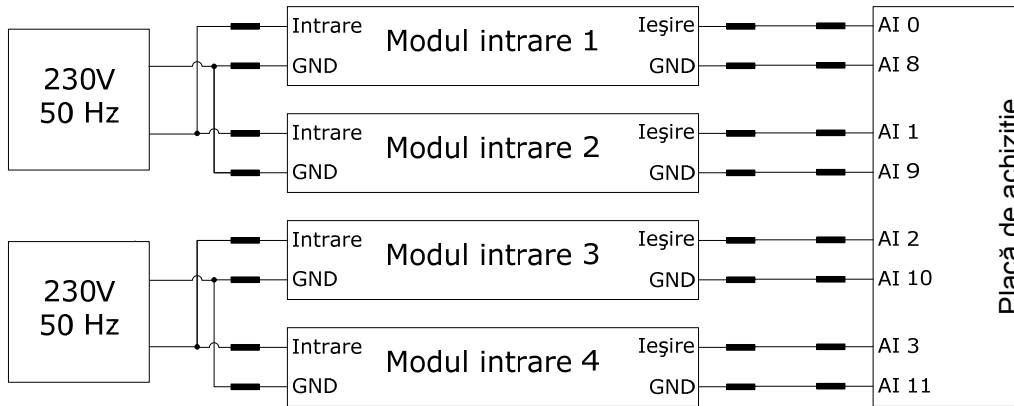


Fig. 6.9. Conectarea a două rețele electrice diferite la circuitul de intrare

Pentru cazul în care se dorește monitorizarea unei rețele trifazate, există două posibilități de conectare; prima posibilitate este de a lega toate intrările de masă ale modulelor de intrare la nulul sistemului trifazat, în acest caz unul dintre module rămânând nefolosit. Cea de-a doua posibilitate este să se lege toate circuitele de masă ale modulelor de intrare împreună și nulul sistemului trifazat stea se leagă la una dintre intrările de semnal ale unuia dintre modulele de intrare (figura 6.10).

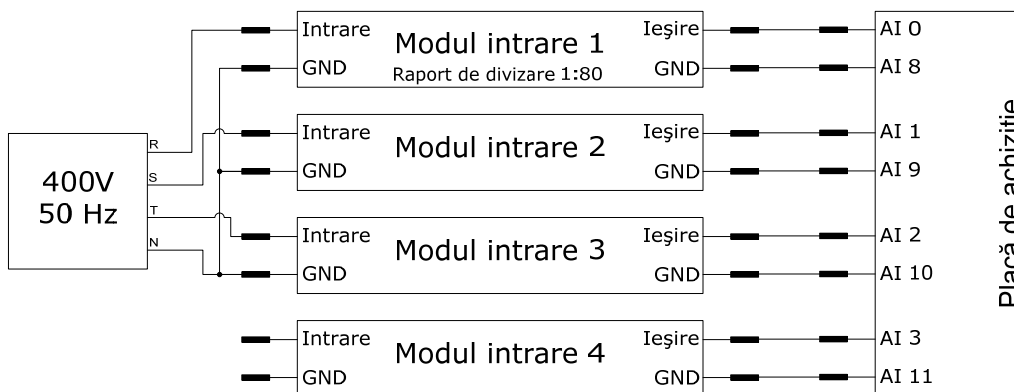


Fig. 6.10. Conectarea unei rețele electrice trifazice la circuitul de intrare

În acest caz, toate circuitele de masă de la ieșirea modulelor de intrare trebuie legate împreună la masa virtuală a plăcii de achiziție configurate în mod diferențial. În cazul configurației trifazate stea, energia electrică este livrată prin intermediul a trei conductoare active (notate cu R, S și T) și a unui fir de masă (nulul, notat cu N). Sistemul trifazat în triunghi are doar cele trei fire active (R, S și T), lipsindu-i firul de nul, acesta fiind reconstituit ulterior prin algoritmi hardware sau software. Circuitul de intrare poate fi configurat să funcționeze cu oricare dintre cele 2 configurații, stea sau triunghi.

În figura 6.11 se prezintă modul de conectare diferențial al circuitului de intrare la o rețea electrică trifazată stea. Pentru intrările în cele patru module s-a realizat un nul artificial.

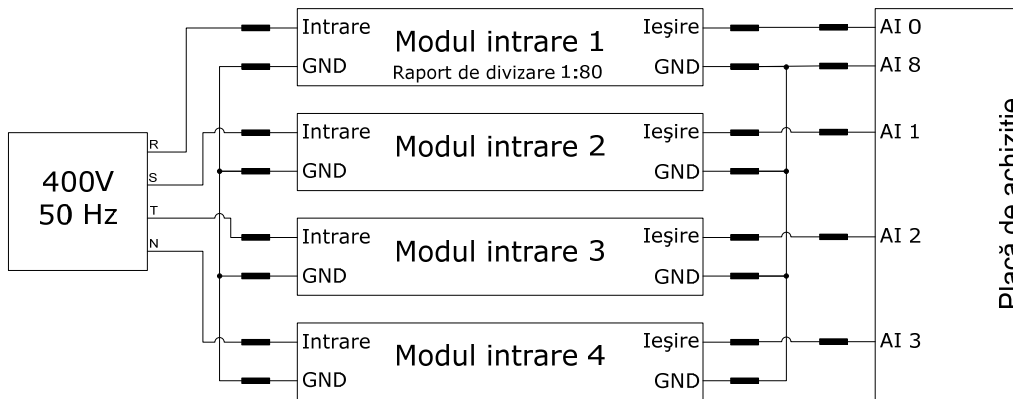


Fig. 6.11. Conectarea unei rețele electrice trifazice la circuitul de intrare în mod diferențial

Aplicația software va prelua semnalele de la placa de achiziție și le va procesa astfel încât la ieșire să se obțină trei semnale corespunzătoare celor trei tensiuni aplicate la intrare.

Pentru achiziția semnalelor din rețeaua trifazată, este necesar ca toate modulele folosite să aibă aceeași amplificare. Pentru a facilita calibrarea rapoartelor de divizare ale celor patru module de intrare, a fost proiectat un modul software special, care atunci când la intrarea celor patru module se aplică același semnal, calculează diferențele dintre module și stabilește niște coeficienți de compensare, astfel încât pentru aceeași tensiune de intrare, toate cele patru module să dea aceeași valoare numerică la ieșirea blocului de procesare a datelor.

Circuitul de condiționare a semnalelor a fost elaborat cu ajutorul aplicației CAD Eagle, iar simulările, efectuate pentru studierea funcționării circuitului (înainte de implementarea practică), au fost realizate cu ajutorul utilitarului PSpice.

6.3 Aplicație software pentru detectarea perturbațiilor

6.3.1 Descrierea aplicației

Aplicația software ce controlează funcționarea întregului sistem hardware, monitorizează rețeaua electrică și detectează perturbațiile ce apar în aceasta, a fost elaborată utilizând mediul de programare Microsoft Visual C++ 7.1 (.NET) [91], [92]. Acest mediu de programare oferă accesul la funcții low-level Windows și permite lucrul facil cu ferestre și controale grafice prin intermediul bibliotecii de funcții MFC (Microsoft Foundations Class) [93], [94]. Spre deosebire de alte limbaje de programare, limbajul C++ este unul dintre limbajele cele mai versatile și oferă posibilități multiple de compilare și optimizare a codului executabil, astfel încât programele elaborate și compilate în C++ să poată rula foarte rapid și fără un consum foarte mare de resurse, lucru important pentru aplicația realizată de autor.

Înainte de a se trece la prezentarea modului în care a fost elaborată aplicația software se vor prezenta câteva noțiuni de bază privind limbajul de programare C++ utilizat în această aplicație.

Un limbaj de programare este o modalitate de descriere a acțiunilor (cu ajutorul unor instrucțiuni specifice), pe care un calculator trebuie să le facă pentru a îndeplini o anumită sarcină.

Pentru a păstra un caracter modular și structurat, aplicația software pentru monitorizarea perturbațiilor din rețelele electrice este structurată pe clase.

Clasele reprezintă tipuri de date abstracte, asemănătoare structurilor, care încapsulează comportamentul și datele asociate unei entități. Comportamentul este descris cu ajutorul *metodelor* (funcții incluse), iar datele, cu ajutorul *atributelor*. O instanță (realizare concretă), a unei clase se numește *obiect*.

Definirea unei clase presupune declararea clasei, la care se specifică:

- numele clasei;
- lista claselor de bază din care este derivată clasa, dacă există;
- membrii clasei, atât datele cât și funcțiile.

Metodele unei clase sunt similare funcțiilor obișnuite, cu două excepții:

- primul parametru (adăugat automat de către compilator), este un pointer către obiectul curent;
- metodele pot accesa direct atributele.

Accesul din exterior la atributele și metodele unei clase se face folosind operatorul „.” (sau „->” în cazul pointerilor).

Atributele și metodele unei clase pot fi grupate din punctul de vedere al drepturilor de acces în trei categorii:

- *private*: pot fi accesate doar din interiorul clasei;
- *protejate*: pot fi accesate din interiorul clasei și din clase derivate;
- *publice*: pot fi accesate atât din interiorul, cât și din exteriorul clasei.

O clasă poate oferi acces nerestricționat la membrii proprii prin utilizarea cuvântului cheie *friend*.

Controlul accesului permite dezvoltatorului clasei limitarea modului de utilizare al clasei și ascunderea implementării interne a operațiilor. Prin aceasta cel care implementează clasa are posibilitatea de a modifica modul în care sunt stocate intern datele sau sunt implementate operațiile fără a afecta aplicațiile deja construite pe baza clasei respective.

Orice obiect derivat dintr-o clasă trebuie construit cu ajutorul unui *constructor* și atunci când nu mai este folosit, trebuie distrus cu ajutorul unei funcții speciale numită *destructor*. Constructorul este o funcție specială apelată automat de compilator la crearea unui nou obiect. În cazul în care nu există nici un constructor definit de către programator, compilatorul va genera un constructor implicit fără nici un parametru. Rolul constructorului este de a inițializa atributele obiectului. O clasă poate avea mai mulți constructori care pot să difere prin numărul și/sau tipul parametrilor, în funcție de necesități.

Destructorii sunt metode speciale apelate de către compilator la ștergerea din memorie a obiectului. Destructorii nu au tip returnat și nu pot primi parametri, o clasă putând avea doar un singur destructor, fiind folosiți pentru a elibera resursele alocate (dezalocarea zonelor de memorie alocate dinamic), la distrugerea obiectului.

Pentru dezvoltarea rapidă a aplicațiilor Windows, Microsoft a pus la dispoziția programatorilor o serie de clase de funcții grupate într-o bibliotecă de clase numită MFC (Microsoft Foundation Class Library). Numele originar al acestei biblioteci de funcții a fost AFX (Application Framework eXtension) și conține funcții care încapsulează porțiuni din biblioteca Windows API în clase de C++, formând o bază pentru aplicațiile care rulează sub sistemul de operare Microsoft Windows [95].

Biblioteca MFC a fost introdusă în anul 1992, împreună cu compilatorul Microsoft C/C++ 4.0 folosit pentru versiunile de Windows pe 16 biți. A fost parte a unui efort al Microsoft de a mai câștiga din piața de dezvoltare a produselor software, cât și pentru a arăta câteva avantaje ale programării în C++. Limbajul C++ era pe cale să înlocuiască limbajul C în cadrul instrumentelor de dezvoltare

pentru aplicații comerciale, iar C/C++ 4.0 era primul compilator al Microsoft, cu suport pentru MFC. În figura 6.12 este prezentată ierarhia de clase MFC.

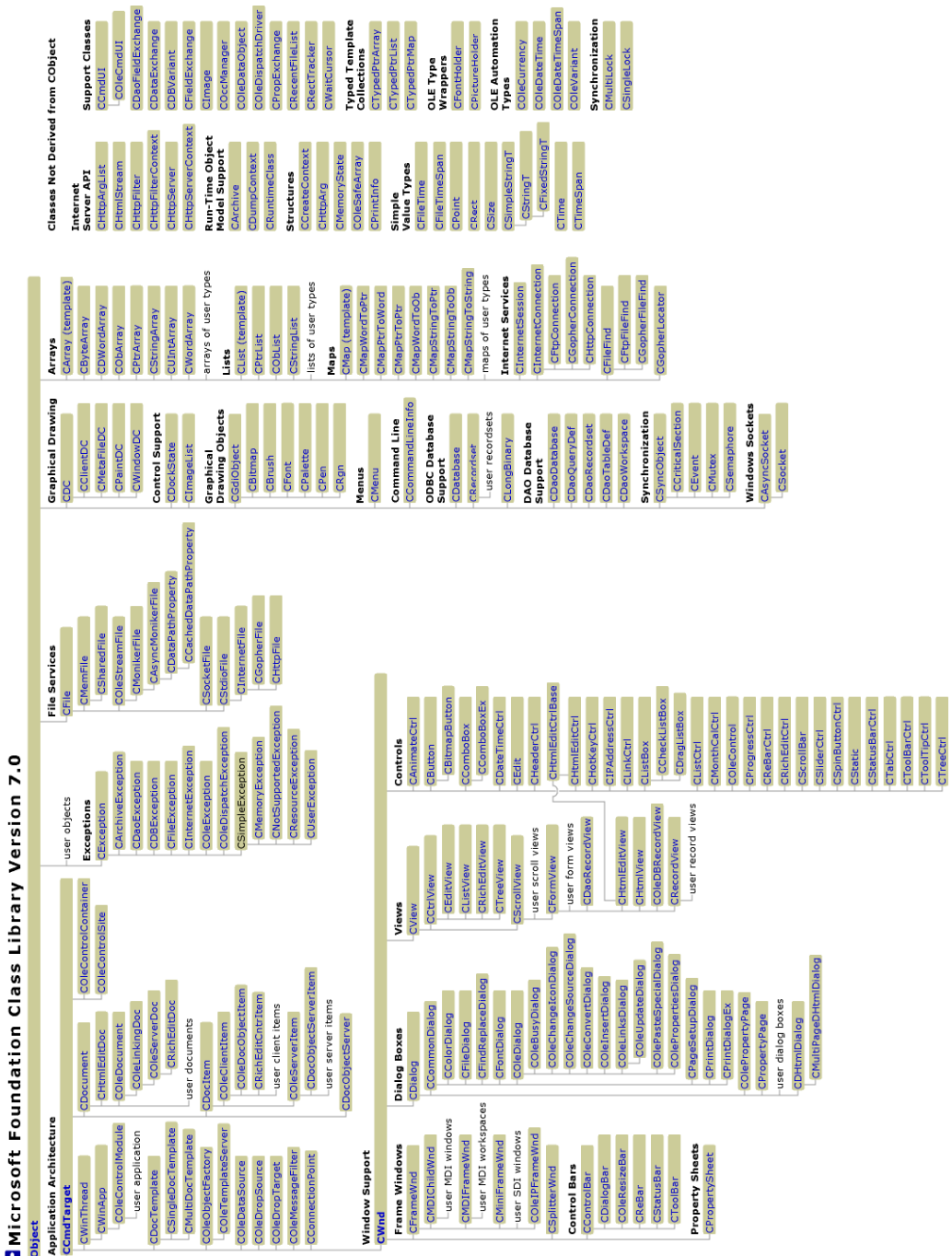


Fig. 6.12. Ierarhia de clase MFC [96]

Versiunea 8 a MFC a fost lansată împreună cu produsul Microsoft Visual Studio 2005 și nu este inclusă în varianta gratuită denumită Express, a produsului. Microsoft recomandă folosirea alternativei .NET pentru viitorul dezvoltării software.

Object Windows Library (OWL), creată de către Borland și inclusă în compilatorul Borland Turbo C compiler, era un competitor al acelor vremuri. Din moment ce urmărea principiile programării orientată pe obiecte, OWL a fost mai populară decât MFC pentru un timp, popularitate pierdută odată cu lansarea MFC de către Microsoft.

Atunci când MFC a fost lansat, Microsoft a extins sintaxa C++ cu o serie de macro-uri pentru a ușura lucrul cu mesajele Windows, excepțiile, identificări ale tipurilor la rulare, instanțierea dinamică a claselor (multe dintre aceste caracteristici – excepții sau identificarea tipurilor la rulare, spre exemplu – nu au fost construite în compilatorul Microsoft, la acea vreme, dar au apărut în versiuni ulterioare). Clasele bibliotecii MFC pot fi clasificate după cum urmează [97]:

- Clasa de bază: Cobject;
- Clase MFC aplicație
 - aplicație și suport fire de execuție;
 - rutare a comenzilor ;
 - document;
 - reprezentare vizuală;
 - ferestre tip ramă (frame);
 - șablon documente;
- Ferestre, dialog-uri, și controale
 - Fereastră (Windows);
 - Reprezentare vizuală (Windows);
 - Casetă de dialog;
 - Clase de control;
 - Bară derulantă;
- Desen și afișare grafică
 - ieșire (ecran, imprimantă);
 - Instrumente de desen;
- Tipuri simple de date
- Tablouri, liste, și hărți de mesaje
 - Clase șablon pentru tablouri, liste, și hărți de mesaje;
 - Șabloane de tip tablou;
 - Șabloane de tip listă de obiecte;
- Fișiere și baze de date
 - Fișiere (intrare/ieșire);
 - DAO;
 - ODBC;
 - OLE DB;
- Internet și rețele de calculatoare
 - ISAPI;
 - Windows Sockets;
 - Win32 Internet;
- Clase OLE (object linked and embedded)
 - Container OLE;
 - Server OLE;
 - OLE drag-and-drop și transfer de date;
 - Dialog-uri OLE;
 - Clase de automatizare OLE;

- Control OLE;
- Arhive de documente;
- Depanare și excepții
 - Suport și depanare;
 - Excepții.

Aplicația pentru monitorizarea calității energiei electrice a fost gândită să funcționeze pe sisteme de operare multi-tasking Microsoft Windows, sisteme ce permit execuția multifilară.

Un program simplu este o secvență de instrucțiuni, în orice moment, pe durata rulării lui existând un singur punct de execuție.

Un fir de execuție (thread) este similar unui program secvențial: instrucțiunile se execută secvențial, iar în orice moment al rulării lui există un singur punct de execuție [98].

Un fir de execuție nu este el însuși un program, deoarece nu poate fi executat de sine stătător. În schimb, firul este executat (rulează) într-un program. Posibilitatea utilizării mai multor fire de execuție într-un singur program este numită execuție multifilară (multithreading) [99].

Aplicația dezvoltată de autor cuprinde trei fire de execuție care rulează în paralel, fiecare dintre ele având o sarcină foarte bine definită:

- firul de achiziție de date;
- firul de detecție a perturbațiilor;
- firul de afișare.

Este necesar ca afișarea datelor să se facă într-un fir de execuție separat, deoarece firele de execuție de interfață au de obicei o prioritate scăzută, iar în cadrul aplicației de monitorizare a perturbațiilor se dorește afișarea instantanee a semnalului achiziționat, precum și a principalilor parametri ai semnalului monitorizat.

Clasele de funcții folosite la realizarea aplicației Power Quality Monitoring System, elaborată de autor, sunt următoarele:

- Clasa MFC aplicație
 - CPowerQualityMonitoringSystemApp – este clasa principală a aplicației care încapsulează toate celelalte clase;
 - CProgramParameters – încapsulează informații despre setările aplicației;
 - CIniHelper – conține toate funcțiile de acces la fișiere .ini;
 - CSignalParameters – cuprinde toți parametrii semnalului procesat;
 - CSignalProcessing – este clasa care conține algoritmul de monitorizare a calității energiei electrice;
 - CCircularBuffer – implementează memoria circulară folosită la stocarea temporară a eșantioanelor semnalului;
 - Lock – implementează semaforizarea firelor de execuție și accesul la memoria circulară.
- Ferestre, dialog-uri, și controale
 - CPowerQualityMonitoringSystemDlg – conține funcțiile ce descriu fereastra principală a aplicației;
 - CAboutDlg – afișează fereastra de informație a aplicației.
- Depanare și excepții
 - SysEvent – implementează funcții sistem.
- Desen și afișare grafică
 - CTextDisplay – afișează textual diverse informații;
 - CWaveScope – afișează grafic semnalele;

- CMyToolBar – implementează un toolbar personalizat;
- CEventsListCtrl – afișează o listă cu ultimele perturbații detectate în sistem;
- CCircularBufferCtrl – control grafic pentru afișarea stării memoriei circulare.

6.3.2 Inițializarea aplicației

În figura 6.13 este prezentată fereastra principală a aplicației. La pornirea aplicației este instanțiată clasa CPowerQualityMonitoringSystemApp care are rolul de a inițializa aplicația și de a genera o nouă instanță a clasei CPowerQualityMonitoringSystemDlg. Codul sursă al acestei clase este prezentat în Anexa 3.1.

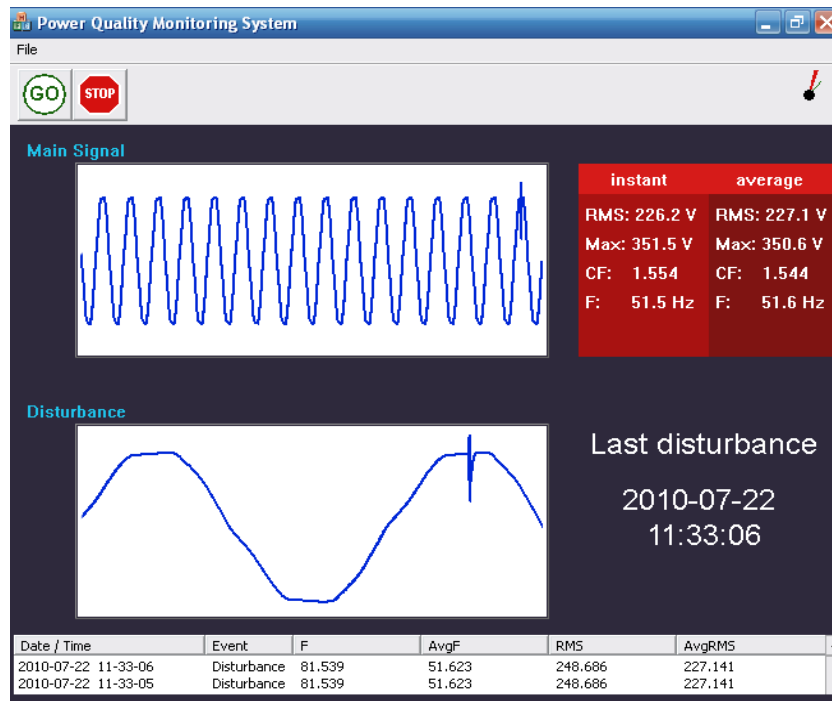


Fig. 6.13. Fereastra principală a aplicației

Clasa CPowerQualityMonitoringSystemDlg inițializează fereastra aplicației și toate controalele grafice conținute de aceasta, îndeplinind în același timp o serie de alte funcții în cadrul aplicației (figura 6.14):

- inițializarea bibliotecilor de funcții utilizate în cadrul aplicației;
- citirea parametrilor de configurare ai aplicației și ai plăcii de achiziție prin intermediul clasei CprogramParameters;
- inițializarea plăcii de achiziție;
- achiziția semnalului și procesarea acestuia în vederea detectării perturbațiilor;
- extragerea parametrilor semnalului (frecvență, valoare de vârf, valoare efectivă);
- afișarea semnalului și a parametrilor acestuia;

- afișarea perturbațiilor detectate.

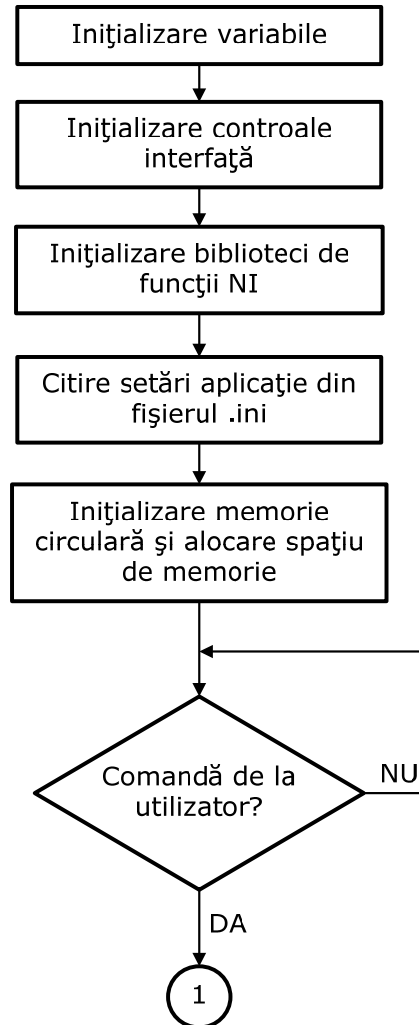


Fig. 6.14. Secvența de inițializare a aplicației

Prin intermediul clasei `CProgramParameters` aplicația citește configurația modulelor de intrare dintr-un fișier `.ini`. În acest fișier sunt stocate toate informațiile despre numărul de canale, modul de configurare al acestora, rata de achiziție, precum și alți parametri necesari bunei funcționări a aplicației. Parametrii sunt citați cu ajutorul funcțiilor API ai clasei `CIniHelper`.

După inițializare, aplicația așteaptă comenzi de la utilizator. Utilizatorul are la dispoziție un toolbar din care poate selecta acțiunea dorită. Butonul *GO* de pe toolbar permite utilizatorului să pornească monitorizarea rețelei electrice, iar butonul *Stop* oprește monitorizarea.

În meniul *File* există un submeniu *Settings* prin accesarea căruia i se permite utilizatorului să modifice parametrii de funcționare a aplicației.

În cazul în care utilizatorul dorește începerea monitorizării, prin apăsarea butonului *GO* este creat firul de execuție achiziție de date care citește continuu date de la placa de achiziție și le stochează în memoria circulară (figura 6.15).

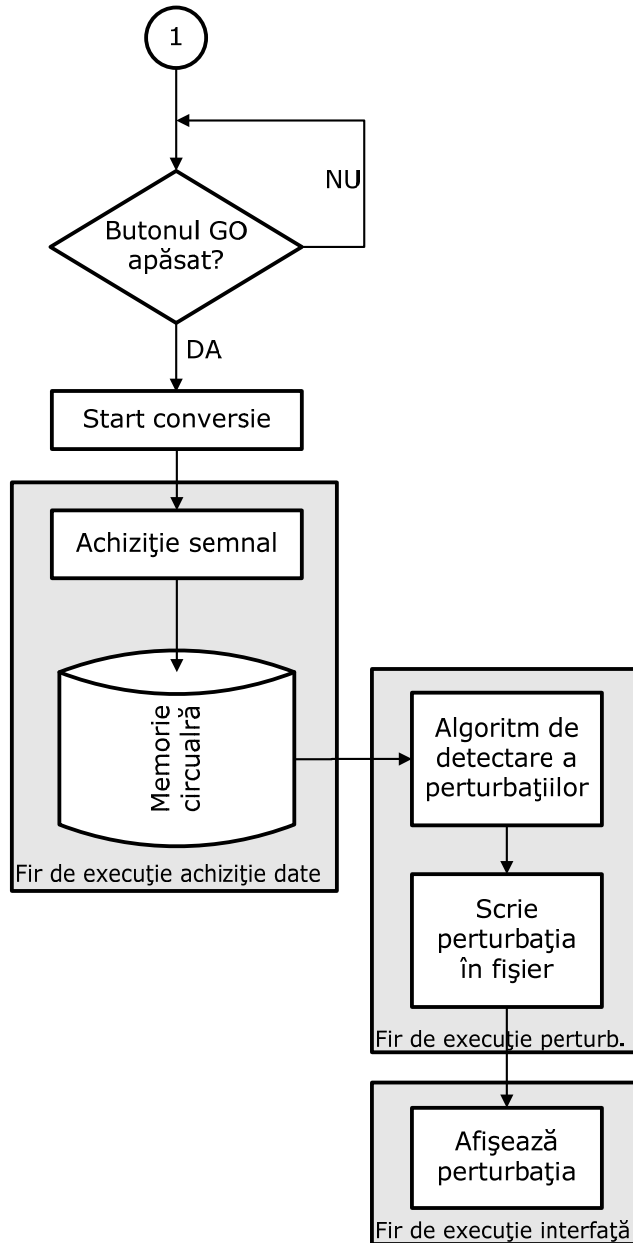


Fig. 6.15. Diagrama rutinei soft de detectare a perturbațiilor

Codul sursă al programului care implementează algoritmul de detecție al perturbațiilor este prezentat în Anexa 3.2.

6.3.3 Memoria circulară

Memoria circulară (figura 6.16), implementată în clasa `CCircularBuffer` este utilizată aici datorită flexibilității pe care o conferă aplicației: prin intermediul ei se poate realiza scrierea și citirea eșantioanelor simultan, fără ca cele două procese care fac acest lucru să se deranjeze reciproc.

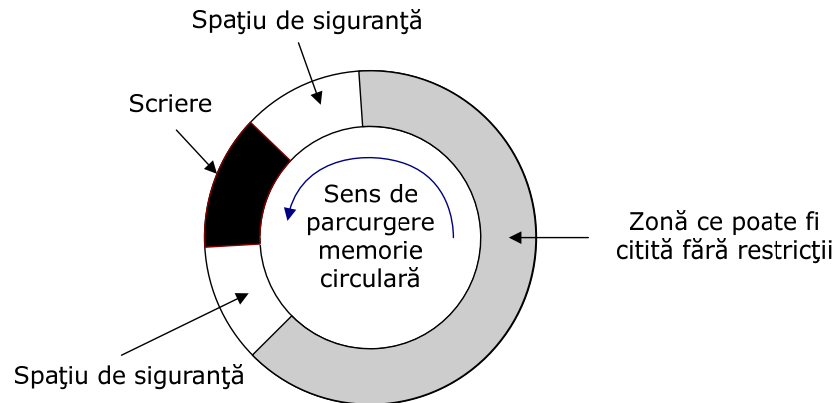


Fig. 6.16. Organizarea memoriei circulare

Astfel, firul de achiziție scrie datele în această memorie circulară, iar firul de detecție a perturbațiilor citește datele din altă zonă a memoriei circulare. Între cele două zone de scriere, respectiv de citire, există un interval de siguranță, pentru a nu se suprapune zona de scriere cu cea de citire, fapt care ar conduce la date citite eronate. Mărimea zonei de siguranță depinde de mărimea bufferului de date care este scris în memoria circulară. Ca regulă generală, ca să nu apară probleme, zona de siguranță are minim mărimea bufferului de scriere. Datorită folosirii memoriei circulare, apare o întârziere între momentul apariției perturbației și momentul semnalizării ei, întârziere ce poate fi determinată cu acuratețe.

Pentru a putea vizualiza corectă funcționare a firelor de execuție ce folosesc memoria circulară, a fost creat un control grafic (implementat în clasa `CCircularBufferCtrl`) care afișează continuu starea memoriei circulare (a zonei de scriere și a zonei de citire) (figura 6.17).

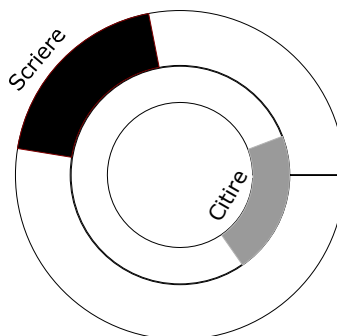


Fig. 6.17. Afișarea grafică a stării memoriei circulare

Cele două buffere de date (cel de scriere și cel de citire), sunt reprezentate sub forma unor sectoare de cerc. Unghiul de deschidere al acestor sectoare de cerc este dat de mărimea bufferului pe care îl descrie, raportată la mărimea memoriei

circulară. Astfel se poate observa foarte ușor modul în care funcționează memoria circulară și se pot face reglaje fine asupra funcționării acesteia.

Programul care implementează memoria circulară este prezentat în Anexa 3.3, iar programul care implementează controlul grafic pentru afișarea stării memoriei circulare este prezentat în Anexa 3.4.

6.3.4 Detectarea și monitorizarea perturbațiilor

Funcțiile care procesează datele provenite de la placa de achiziție și detectează perturbațiile conform algoritmului prezentat în figura 6.18 sunt implementate în clasa CSignalProcessing.

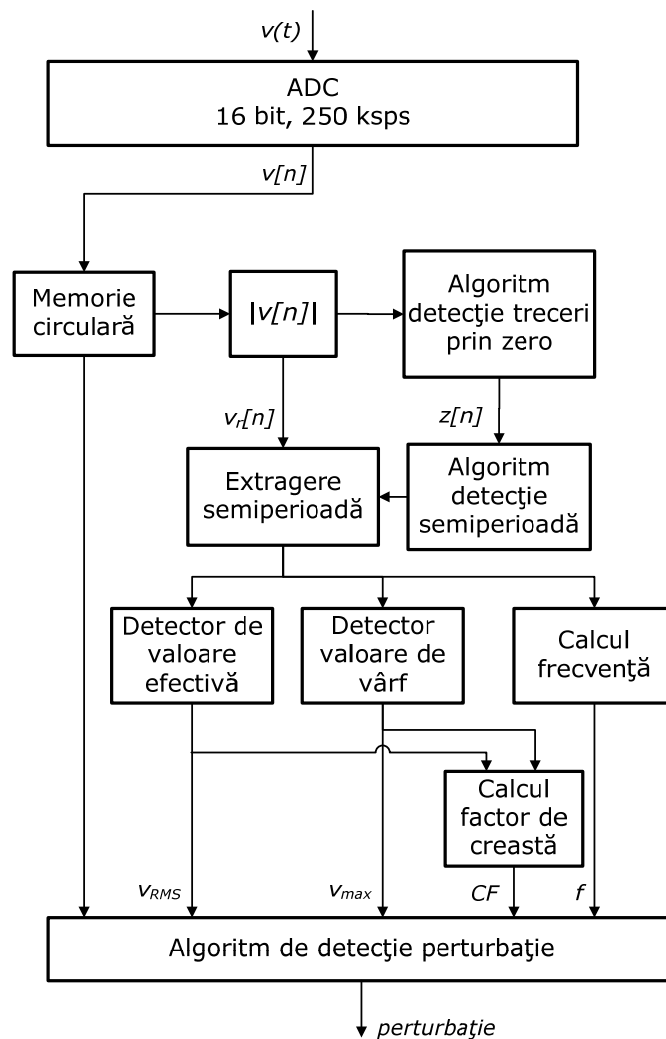


Fig. 6.18. Algoritmul de detectare a perturbațiilor

Clasa CSignalParameters încapsulează într-o formă mai simplă principalii parametri ai semnalului: frecvența, valoarea de vârf, valoarea efectivă și factorul de

creastă. Astfel, pentru fiecare semiperioadă a semnalului sunt memorate aceste valori sub forma unei liste având 1000 de poziții. Prin înregistrarea pe disc a acestei liste la apariția unei perturbații, utilizatorul poate vedea modul în care au variat acești parametri înainte, în timpul și după producerea perturbației, lucru util în analiza ulterioară a perturbației.

Pentru a ilustra funcționarea blocului de detectare al perturbațiilor prezentat în figura 6.18, se presupune că la intrarea sistemului se aplică un semnal de forma celui din figura 6.19, unde $v(t)$ este semnalul preluat din rețeaua electrică, $v_r(t)$ este semnalul rețelei redresat, $z(t)$ reprezintă trecerile prin zero ale tensiunii rețelei, V_{RMS} și V_{MAX} fiind valorile efectivă și respectiv maximă ale semnalului rețelei, CF este factorul de creastă (crest factor), iar D reprezintă prezența sau absența perturbațiilor.

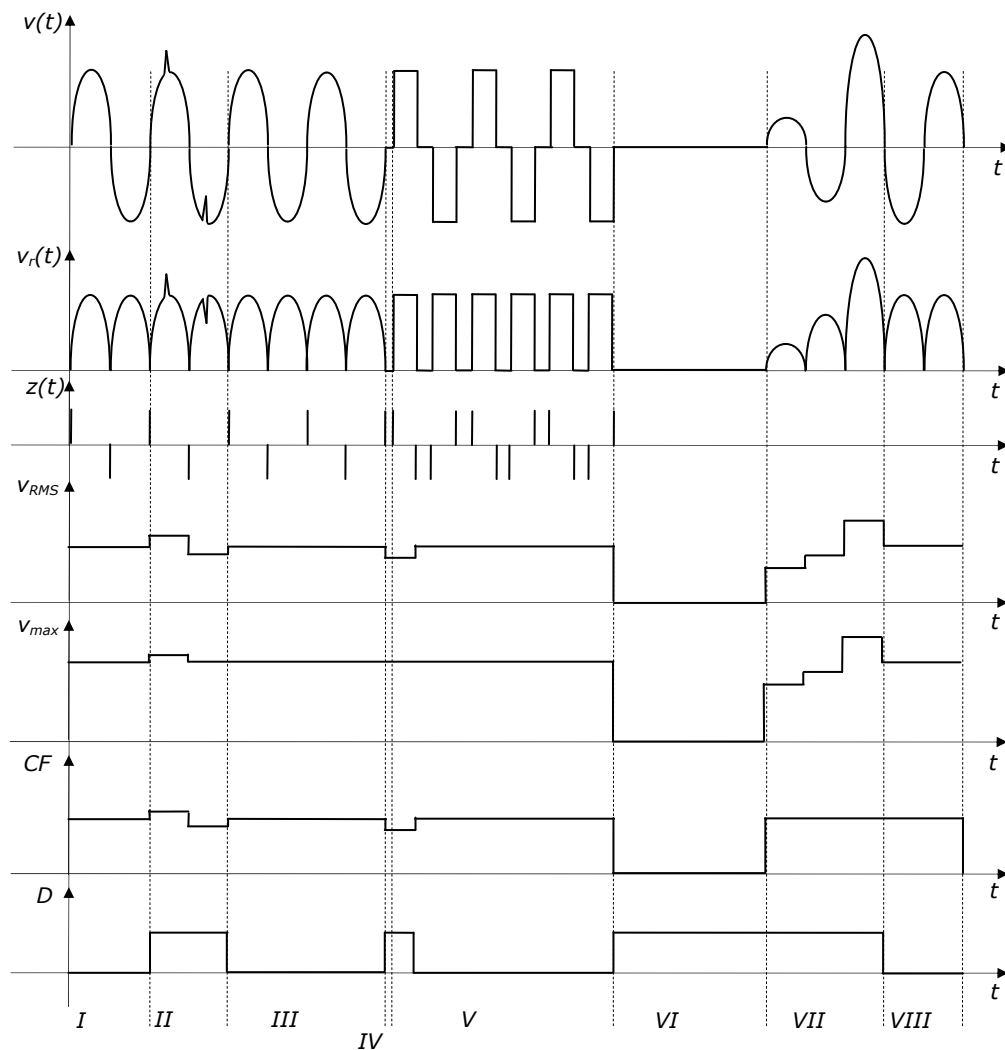


Fig. 6.19. Diagrame temporale pentru explicarea funcționării sistemului de detectare a perturbațiilor

Sistemul de monitorizare este plasat într-o zonă rezidențială, conectat în aceeași priză cu un calculator personal legat la rețea prin intermediul unui UPS. Semnal din figura 6.19 prezintă opt zone distincte. După o perioadă de funcționare normală (zona I), în rețeaua electrică apar unele perturbații (zona II), urmate de o altă perioadă normală de funcționare (zona III). La un moment dat are loc o întrerupere bruscă de tensiune (zona IV), iar UPS-ul intră în funcțiune. UPS-ul funcționează o perioadă de timp (zona V), până când bateria lui se descarcă (zona VI). După un timp, defecțiunea apărută în rețeaua electrică este remediată și se reia alimentarea cu energie electrică (zona VII), apoi totul reintră în normal (zona VIII).

În cazul rețelelor monofazate, detectarea perturbațiilor se face pe canalul cu cea mai mare amplitudine, celelalte canale fiind folosite pentru a reconstrui forma corectă a perturbațiilor cu amplitudini de peste 500 V. Pentru sistemele trifazate, analiza perturbațiilor se face pe fiecare dintre cele trei canale de intrare cu un program software mai complex.

Semnalele provenite de la circuitul de condiționare sunt preluate de către placa de achiziție ($v(t)$) și convertite cu o rată de eșantionare de 250 kps (rezultând un număr de 12500 eșantioane pe fiecare perioadă a semnalului pentru rețelele de 50 Hz), în eșantioane numerice ($v[n]$).

Semnalul $v[n]$ este apoi redresat, obținându-se semnalul $v_r[n]$. Pentru a separa semiperioadele semnalului $v_r[n]$, semnalul $v_r[n]$ este aplicat blocului de detecție a minimelor semnalului redresat. Între două minime avem o semiperioadă a semnalului de intrare (în lipsa perturbațiilor), (figura 6.20).

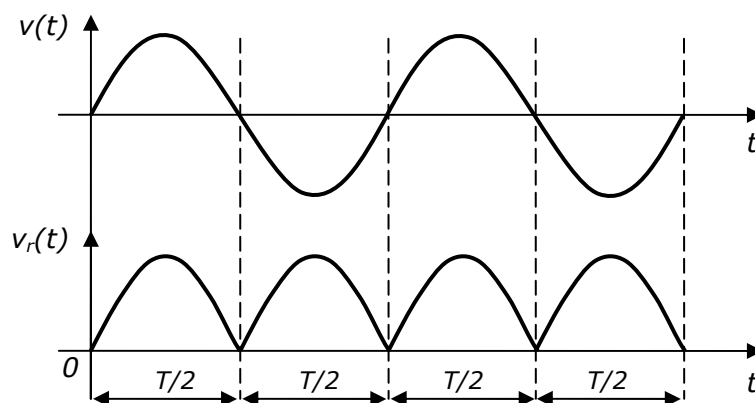


Fig. 6.20. Detectarea duratei unei semiperioade a semnalului rețelei

Pentru fiecare semiperioadă a semnalului se calculează valoarea efectivă, valoarea de vârf, factorul de creastă și frecvența tensiunii aplicate la intrare. Valoarea efectivă a tensiunii se calculează conform formulei 2.6, iar frecvența se calculează pe baza numărului eșantioanelor dintr-o semiperioadă cu ajutorul formulei de mai jos:

$$f = \frac{f_{\text{eșantionare}}}{2 \times \text{număr eșantioane semiperioadă}} \quad (6.1)$$

În figura 6.21 se prezintă algoritmul de calcul pentru determinarea frecvenței semnalului bazată pe detecția pozițiilor minimelor semnalului de intrare redresat.

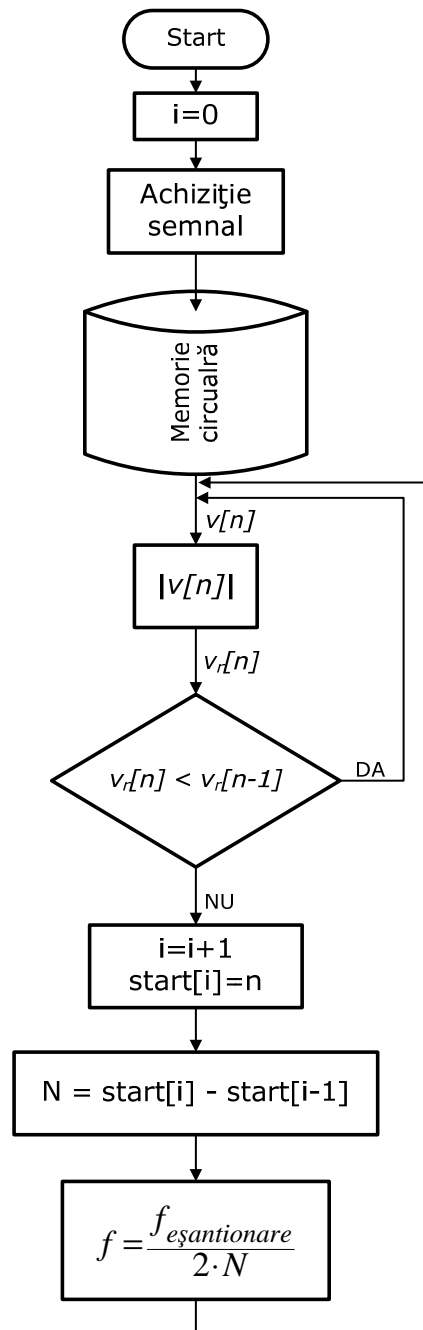


Fig. 6.21. Algoritm pentru calculul frecvenței rețelei

Algoritmul de detectare al perturbațiilor este prezentat în figura 6.22 și constă în compararea celor trei valori cu mediile lor pe 10 perioade ale semnalului. Dacă există abateri de la medie, mai mari decât o valoare de prag δ , atunci se semnalează prezența unei perturbații.

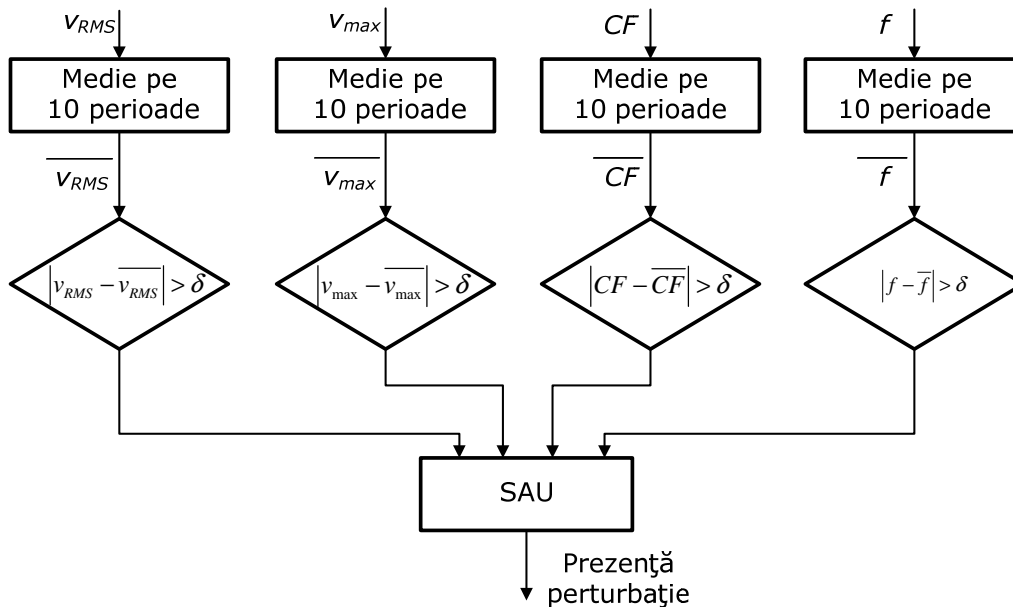


Fig. 6.22. Blocul de detectare a perturbațiilor

În cazul în care există o perturbație care se suprapune peste semnalul de intrare și cauzează detectarea unui minim fals al semnalului, aceasta este imediat detectată de către sistem prin simpla comparare a frecvenței instantanee cu frecvența medie pe 10 perioade. Pentru a se reduce numărul detectării de minime false datorate zgomotului prezent în sistem, odată detectat un minim, poziția acestuia este memorată și următorul minim care se ia în considerare este la o distanță de minim un sfert de perioadă față de minimul detectat anterior.

În conformitate cu standardele în vigoare (prezentate în capitolul 2) sunt calculate valorile medii ale frecvenței și tensiunii după cum urmează:

- media valorii efective a tensiunii rețelei pe 200 ms;
- media valorii efective a tensiunii rețelei pe 3 secunde;
- media valorii efective a tensiunii rețelei pe 10 minute;
- media valorii efective a tensiunii rețelei pe 2 ore;
- media frecvenței pe 10 secunde;

După ce au fost detectate, perturbațiile sunt salvate în 2 fișiere:

- un fișier cu extensia *.txt* în care sunt înregistrate eșantioanele semnalului perturbat;
- un fișier cu extensia *.ext* în care sunt salvate valorile parametrilor monitorizați surprinși în momentul apariției perturbației. În acest fișier poate fi specificată fie automat de către aplicația software, fie ulterior, de către utilizator, locația unde a fost detectată perturbația, lucru util în cazul monitorizării mai multor situri.

Alături de aceste două fișiere, care conțin strict date despre perturbațiile detectate, mai sunt înregistrate și alte 2 fișiere care conțin valorile tensiunii efective

și ale frecvenței (măsurate pe o semiperioadă a semnalului), împreună cu momentul de timp la care au fost măsurate. Valorile corespunzătoare unei zile sunt înregistrate în același fișier. Momentul de timp la care apare perturbația este înregistrat atât în numele fișierului care conține perturbația, cât și în fișierul cu extensia .ext.

6.3.5 Interfața cu utilizatorul

Interfața cu utilizatorul este una dintre cele mai importante componente ale unei aplicații deoarece prin intermediul ei utilizatorul poate controla modul de funcționare al aplicației, poate introduce comenzi și poate vizualiza rezultatele rulării aplicației.

Există mai multe modalități de implementare a unei interfețe cu utilizatorul, cea mai utilizată fiind *Interfața Grafică Utilizator* (GUI), interfață care, așa cum îi spune și numele, este realizată în mod grafic, ușurând modul în care sunt introduse și extrase datele din aplicație.

Interfața utilizator cuprinde funcții de intrare, vizualizare și ieșire.

Deoarece biblioteca MFC pune la dispoziția programatorilor doar funcții de bază pentru afișarea de text și primitive grafice, a fost nevoie de elaborarea unor controale grafice cu ajutorul cărora să fie afișat semnalul de intrare, perturbația apărută precum și parametrii semnalului de intrare, fiecare dintre aceștia fiind implementați într-o clasă separată.

Bara de instrumente

Bara de instrumente este implementată în clasa CMyToolBar derivată din clasa MFC CToolBar și căreia i s-au adăugat funcții specifice aplicației prezentate:

- două butoane personalizate GO și STOP cu ajutorul cărora este pornită/oprită achiziția datelor;
- un control grafic pentru indicarea stării memoriei circulare (amplasat în partea stângă a barei de instrumente).

Controlul grafic pentru afișarea stării memoriei circulare este implementat în clasa CCircularMemoryCtrl fiind utilizat ca și variabilă membru a clasei CmyToolBar, motiv pentru care s-a acordat o atenție deosebită redesenării și poziționării acestui control pe suprafața barei de instrumente (figura 6.23).



Fig. 6.23. Bara de instrumente a aplicației

Afișarea semnalelor

Pentru afișarea grafică a semnalelor au fost elaborate funcții speciale grupate în clasa CWaveScope. Fereastra aplicației are două controale identice ca și funcționalitate, unul dintre ele afișând semnalul preluat de la intrarea plăcii de achiziție, iar celălalt afișează perturbația detectată (figura 6.24).

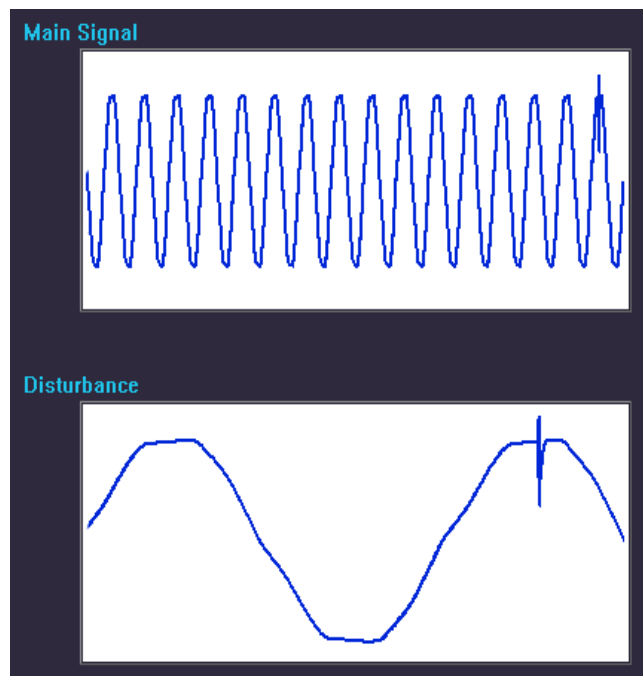


Fig. 6.24. Controale pentru afișarea grafică a semnalelor

6.3.6 Configurarea aplicației

Așa cum s-a arătat în paragraful 6.2, sistemul de achiziție poate fi configurat în funcție de natura semnalului/semnalelor achiziționate. Deoarece se dorește o configurare ușoară a aplicației s-a ales configurarea acestora bazată pe fișiere de inițializare (.ini). Acestea sunt fișiere cu o structură specială și permit configurarea parametrilor aplicației într-o manieră simplă și intuitivă, spre deosebire de modalitatea de configurare a aplicației folosind regiștrii sistemului de operare Microsoft Windows. Utilizatorul poate pre-configura mai multe fișiere în funcție de scopul urmărit, fișiere ce pot fi folosite apoi în configurarea aplicației. Spre exemplu, se poate configura un fișier pentru monitorizarea tensiunii de pe o singură fază (într-o clădire, laborator etc.), folosind trei divizoare de tensiune (configurat ca în figura 6.8) și un alt fișier pentru monitorizarea unui sistem trifazat (configurat ca în figura 6.11). Prin simpla înlocuire a fișierului de configurare cu unul sau altul din aceste două fișiere, aplicația este configurată în modul de funcționare dorit.

Un fișier de inițializare conține următoarele entități: secțiuni, chei și valori. O secțiune este asemănătoare unui capitol de carte, conținând mai multe chei (subcapitole). Secțiunile sunt notate între paranteze drepte. O cheie codifică în mod unic o proprietate a secțiunii din care face parte. Fiecare cheie are o valoare și numai una la un moment dat. Pot exista și chei cu același nume dar obligatoriu în secțiuni diferite.

Secțiunile, cheile și valorile corespunzătoare ale acestora sunt prezentate sub formă de diagramă în figura 6.25. Secțiunile sunt reprezentate prin dreptunghiuri cu chenar dublu, cheile cu chenar simplu iar valorile cu linie punctată.

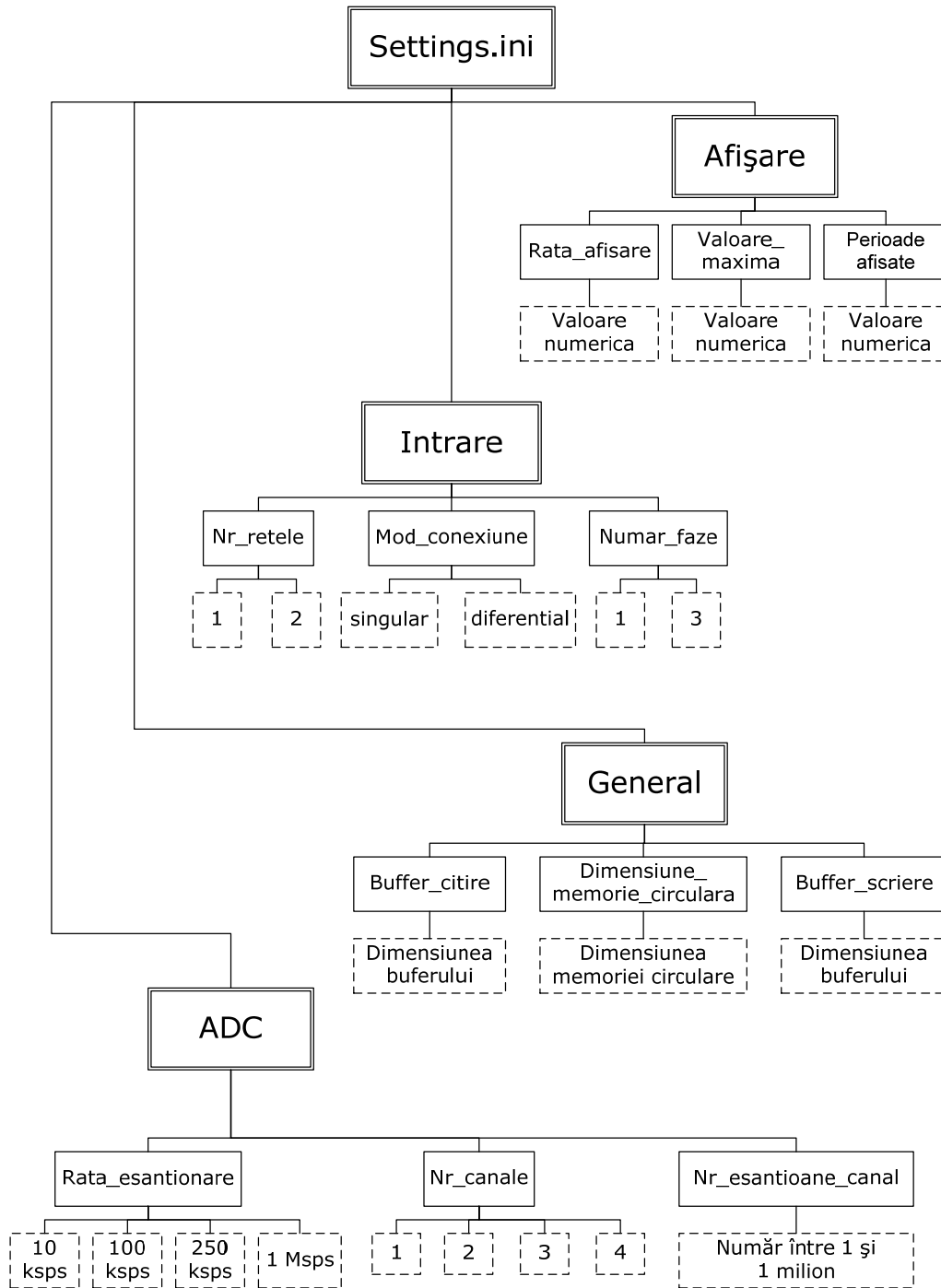


Fig. 6.25. Structura fișierului settings.ini

În exemplul de mai jos se prezintă o secțiune "General" care are două chei: "Cheie_1" cu valoarea 1 și "Cheie_2" cu valoarea "val". Tot ce este scris după marcajul ";" reprezintă comentarii. Marcajul se întinde până la sfârșitul liniei curente.

```
[General]           ; numele secțiunii
Cheie_1 = 1        ; prima cheie
Cheie_2 = val      ; a doua cheie
```

6.3.7 Testarea aplicației software

O etapă importantă în realizarea oricărei aplicații hardware și/sau software o reprezintă etapa de testare.

De-a lungul timpului, diverși autori au realizat sisteme hardware pentru generarea de perturbații electromagnetice, instrumente folosite apoi la testarea sistemelor de detectare și monitorizare a perturbațiilor [100], [101].

Testarea aplicațiilor software se face, în general, luând fiecare funcție în parte și studiindu-i comportamentul la aplicarea diferitelor combinații de parametri de intrare. Aceasta este o metodă scumpă, laborioasă și consumatoare de timp, motiv pentru care autorul a optat pentru utilizarea unei metode mult mai simple pentru testarea aplicației software.

În cazul sistemului pentru detectarea și monitorizarea perturbațiilor descris în paragraful 6.2, datorită particularităților acestuia, constând în folosirea unui sistem de achiziție de date ce preia semnale din mediul extern, se poate realiza mult mai simplu testarea aplicației software prin generarea de semnale sinusoidale cu frecvența de 50 Hz peste care se suprapun perturbații asemănătoare cu cele din rețeaua electrică. Modificând amplitudinea perturbației în raport cu semnalul rețelei și poziția perturbației în raport cu trecerea prin zero a semnalului rețelei, se poate face o acordare fină a parametrilor sistemului de detectare a perturbațiilor, astfel încât acesta să funcționeze în parametrii optimi.

Autorul a elaborat de-a lungul timpului diverse aplicații software pentru testarea sistemelor, utilizând generarea de semnale cu ajutorul plăcii de sunet a calculatorului, programate fie în MatLab [102], [103], [104], fie în Visual C++ [105].

În continuare se va prezenta o aplicație Visual C++ pentru generarea de semnale utilizată la testarea sistemului de monitorizare a perturbațiilor (figura 6.26).

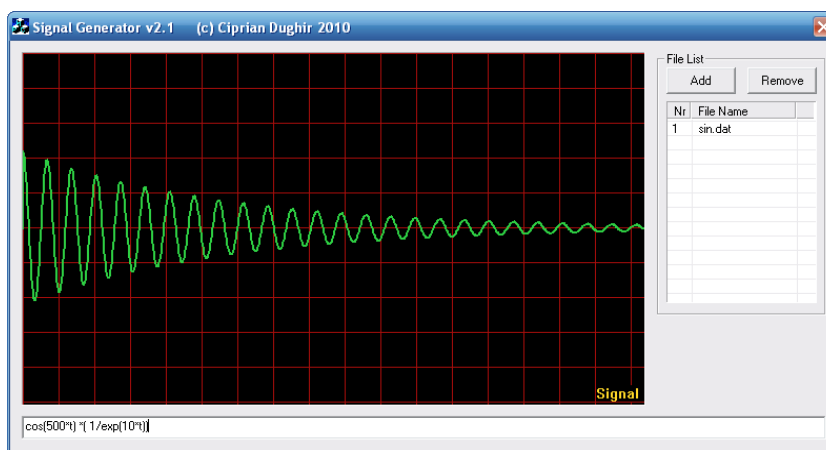


Fig. 6.26. Aplicația de generare a semnalelor

Aplicația este scrisă în Visual C++ și folosește placa de sunet a calculatoarelor personale pentru generarea de semnale cu perturbații. Se pot genera semnale utilizându-se descrierea matematică a acestora sau semnale preînregistrate în fișiere.

Fereastra aplicației cuprinde un afișaj grafic cu ajutorul căruia este afișat semnalul care va fi generat, o căsuță de introducere a formulei matematice ce descrie semnalul generat și o listă în care se pot introduce unul sau mai multe fișiere conținând forme de undă ale semnalelor care se doresc a fi generate. Semnalele sunt generate continuu la ieșirea plăcii de sunet, repetarea secvenței de semnal dorită făcându-se automat. Pentru încărcarea unui fișier conținând eșantioanele care vor fi generate, se apasă butonul *Add* și se selectează fișierul dorit, acesta apărând în lista de fișiere. Referirea datelor din fișier se face prin utilizarea numelui fișierului ca și variabilă de intrare pentru interpretorul de funcții.

În cazul folosirii descrierii matematice a semnalului generat, o componentă a aplicației numită *interpretor de funcții*, analizează funcția complexă introdusă de către utilizator și o desparte în funcții componente pe care le generează independent una de alta, iar apoi aplică operatorii specificați.

Toți operatorii matematici de bază sunt admiși: adunare, scădere, înmulțire, împărțire și ridicare la putere. Pe lângă aceștia au fost implementate și câteva funcții matematice de bază, funcții cu ajutorul cărora se pot construi forme de undă complexe. În tabelul 6.3 sunt prezentate funcțiile matematice acceptate de aplicație (menționez că în aplicație frecvența se înmulțește cu 2π).

Tabelul 6.3. Funcțiile matematice admise de aplicație și sintaxa acestora

Funcția matematică	Sintaxa utilizată în aplicație
$\sin(x)$	$\sin(\text{frecvență} * t + \text{offset})$
$\cos(x)$	$\cos(\text{frecvență} * t + \text{offset})$
$\tan(x)$	$\tan(\text{frecvență} * t + \text{offset})$
$\text{ctn}(x)$	$\text{ctn}(\text{frecvență} * t + \text{offset})$
$\text{sqrt}(x)$	$\text{sqrt}(t + \text{offset})$
$\text{pow}(x,y)$	$\text{pow}(x, y, t, \text{offset})$
$\text{rand}(x)$	$\text{rand}(\text{valoare maximă})$
$\sinh(x)$	$\text{sh}(\text{frecvență} * t + \text{offset})$
$\cosh(x)$	$\text{ch}(\text{frecvență} * t + \text{offset})$
$\tanh(x)$	$\text{th}(\text{frecvență} * t + \text{offset})$
$\text{ctanh}(x)$	$\text{cth}(\text{frecvență} * t + \text{offset})$
$\text{asin}(x)$	$\text{asin}(\text{frecvență} * t + \text{offset})$
$\text{acos}(x)$	$\text{acos}(\text{frecvență} * t + \text{offset})$
$\text{atan}(x)$	$\text{atan}(\text{frecvență} * t + \text{offset})$
$\text{actn}(x)$	$\text{actn}(\text{frecvență} * t + \text{offset})$
$\text{sign}(x)$	$\text{sign}(t + \text{offset})$
$\text{abs}(x)$	$\text{abs}(t + \text{offset})$
$\log(x)$	$\log(t + \text{offset})$
$\ln(x)$	$\ln(t + \text{offset})$
$\text{exp}(x)$	$\text{exp}(t + \text{offset})$
t (timp)	t
eșantioane din fișiere	nume_fișier

În cazul funcțiilor trigonometrice se specifică ca și parametru frecvența în Hz semnalului care va fi generat, parametrul t utilizat în argumente de forma

„frecvența * t” având rolul de a informa interpretorul de funcții asupra faptului că se dorește desfășurarea în timp a funcției respective și nu doar generarea unei valori punctuale a funcției, lucru posibil prin omiterea parametrului t din argumentul funcției.

În figura 6.27 este prezentată diagrama de funcționare a aplicației.

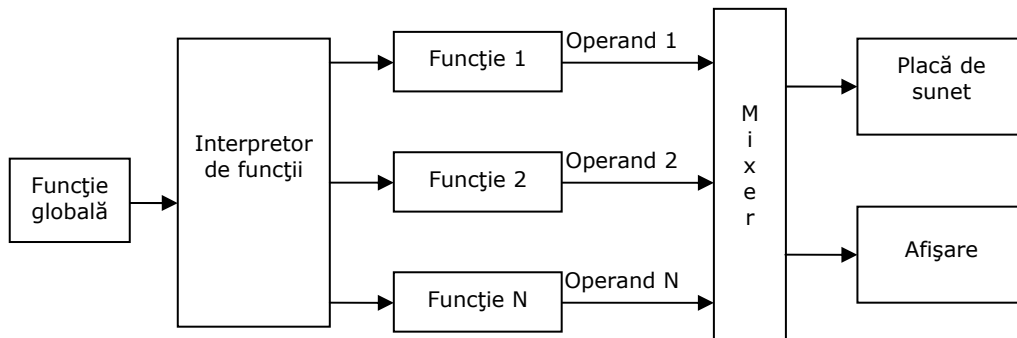


Fig. 6.27. Diagrama aplicației de generare a semnalelor de test

Accesul aplicației la driverele plăcii de sunet este asigurat de un set de funcții API dedicate.

În figura 6.28 este prezentată fereastra principală a aplicației în cazul generării unui semnal de tipul impuls biexponențial.

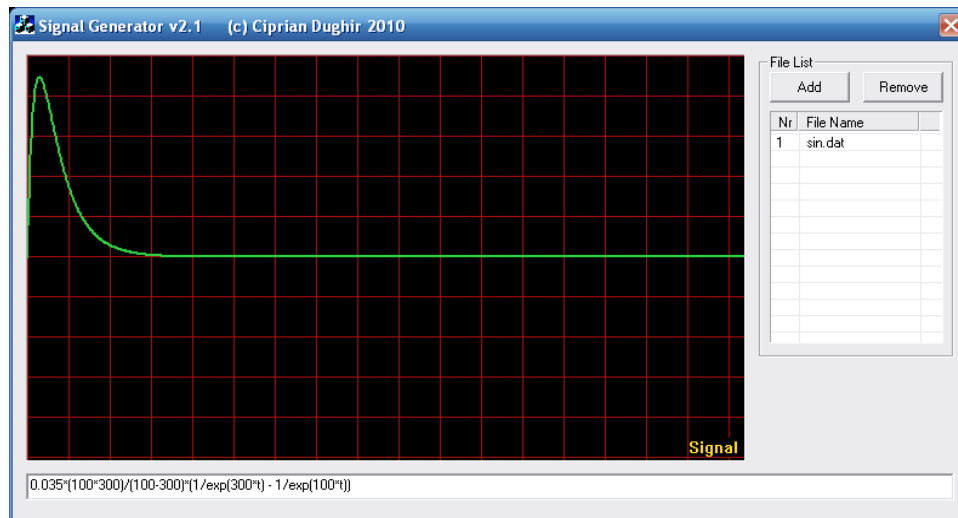


Fig. 6.28. Aplicația de generare a semnalelor. Generarea unui semnal biexponențial

Astfel de impulsuri apar în rețelele electrice în cazul trăsnetului sau în cazul conectării surselor de alimentare în comutație [106]. Deoarece aplicația nu pune la dispoziția utilizatorului o funcție dedicată pentru generarea unui astfel de semnal, implementarea semnalului biexponențial a fost făcută pornindu-se de la formula sa matematică [107]:

$$f(t) = \frac{a \cdot b}{a - b} (e^{-bt} - e^{-at}) \quad (6.2)$$

Aplicația nu poate calcula funcții exponențiale cu argument negativ, motiv pentru care s-a utilizat următoarea formulă echivalentă:

$$f(t) = \frac{a \cdot b}{a - b} \left(\frac{1}{e^{bt}} - \frac{1}{e^{at}} \right) \quad (6.3)$$

Prin modificarea coeficienților a și b se pot obține diverse forme ale funcției biexponențiale (figura 6.29) [107].

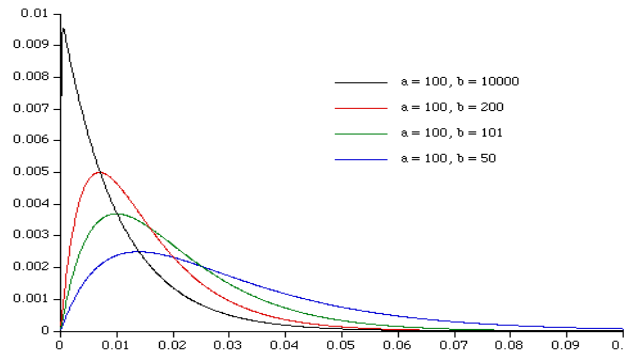


Fig. 6.29. Impuls biexponențial pentru diferite valori ale parametrilor a și b

Această perturbație biexponențială poate fi suprapusă peste un semnal sinusoidal cu frecvența de 50 Hz, constituind un punct de plecare în analiza sistemului de detectare a perturbațiilor (figura 6.30). Poziția impulsului poate fi modificată cu ușurință de către utilizator prin folosirea unei funcții de translație $t + x$, unde x reprezintă numărul de milisecunde cu care este întârziat impulsul biexponențial.

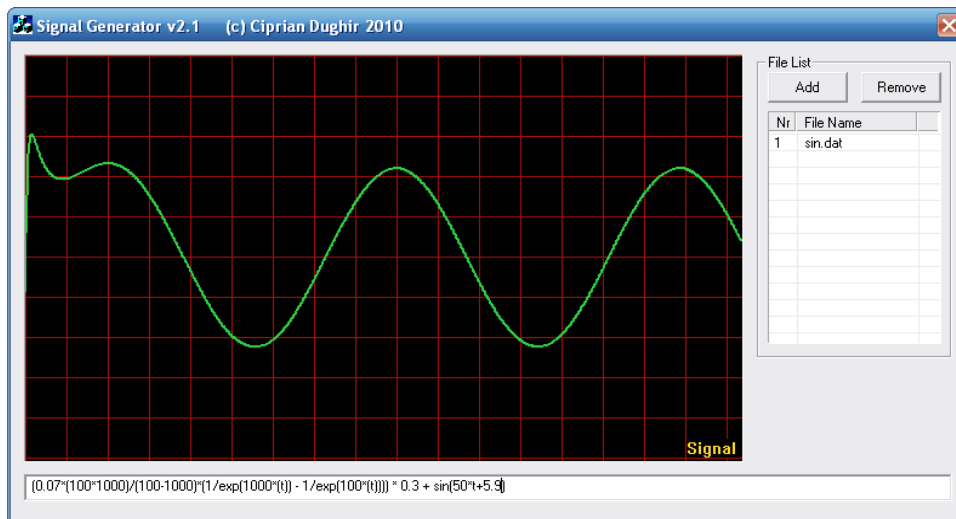


Fig. 6.30. Aplicația de generare a semnalelor. Generarea unui semnal sinusoidal cu frecvența de 50 Hz peste care se suprapune un impuls biexponențial

Pentru ca semnalele generate să se apropie cât mai mult de semnalele din practică, aplicația oferă posibilitatea suprapunerii de zgomot care poate fi generat cu ajutorul unor funcții C++ peste semnalul generat. În figura 6.31 este prezentat un semnal sinusoidal peste care a fost aplicat un zgomot având o amplitudine modulată cu ajutorul unei funcții biexponențiale, astfel încât doar două perioade ale semnalului să fie perturbate, celelalte rămânând intacte.

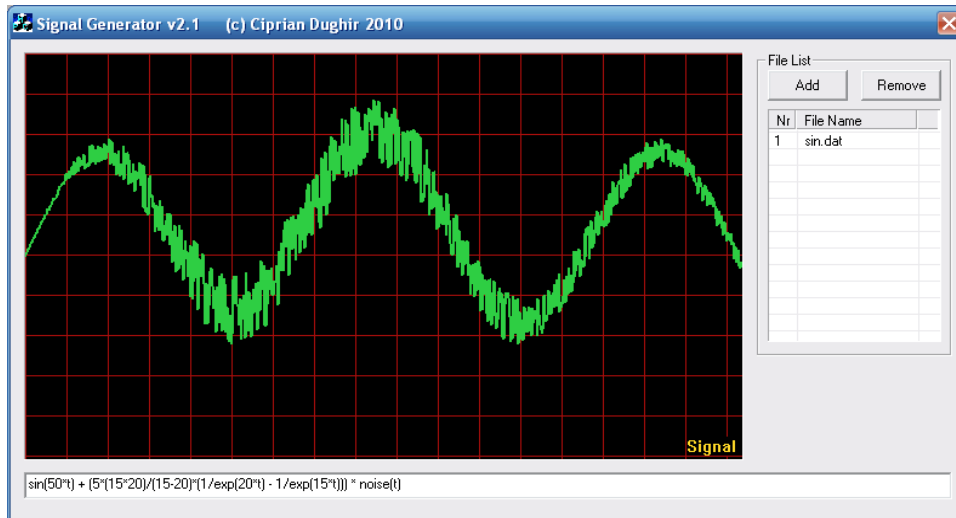


Fig. 6.31. Aplicația de generare a semnalelor. Generarea unui semnal sinusoidal însoțit de zgomot

6.4 Server web pentru accesarea on-line a datelor înregistrate

Sistemul de monitorizare a calității energiei electrice poate fi amplasat în orice locație, fie ea și greu accesibilă, motiv pentru care a fost implementat un server web ce permite consultarea valorilor curente ai parametrilor mășurați, vizualizarea perturbațiilor apărute în sistem și descărcarea datelor pe un alt calculator în vederea analizării acestora, fără a implica prezența utilizatorului la site-ul monitorizat. Accesul aplicației la Internet poate fi asigurat de un modem GSM acolo unde există semnal GSM, sau de un modem radio, acolo unde nu există semnal GSM sau alt tip de acces Internet.

Pagina web implementată cu ajutorul serverului web poate fi accesată de o mare varietate de clienți web: calculatoare personale sau portabile, dispozitive PDA, telefoane mobile etc. (figura 6.32).

Pentru realizarea serverului web s-a folosit aplicația *Xampp* [108], care permite implementarea unui server web bazat pe limbajele HTML [109] și PHP [110]. HTML este prescurtarea de la Hyper Text Mark-up Language și este codul care stă la baza paginilor web [111]. HTML este un limbaj de programare orientat către prezentarea documentelor text pe o singură pagină, utilizând un software de redare specializat, numit în general, browser web. Limbajul HTML pune la dispoziția programatorului mijloacele prin care conținutul unui document poate fi adnotat cu diverse tipuri de metadata și indicații de redare. Indicațiile de redare pot varia de la

decorații minore ale textului, cum ar fi specificarea faptului că un anumit cuvânt

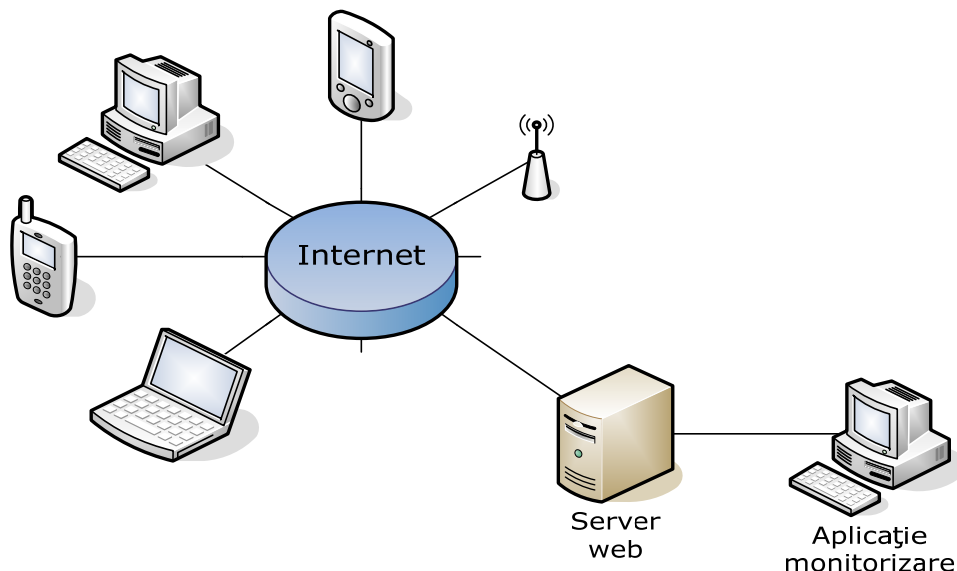


Fig. 6.32. Conectarea serverului web la Internet

trebuie subliniat, sau că o imagine trebuie introdusă în text, până la scripturi sofisticate, hărți de imagini și formulare. Metadatele pot include informații despre titlul și autorul documentului, informații structurale despre cum este împărțit documentul în diferite segmente, paragrafe, liste, titluri etc. și informații esențiale care permit ca documentul să poată fi legat de alte documente pentru a forma astfel hiperlink-uri.

Codul HTML poate fi generat utilizând tehnologii de codare pe partea de server cum ar fi: PHP, JSP sau ASP. Multe aplicații cum ar fi sistemele de gestionare a conținutului (CMS), ERP-uri sau forumuri web generează pagini HTML în funcție de context și de acțiunile utilizatorului.

Numele limbajului de programare PHP provine din limba engleză și este un acronim recursiv: Php Hypertext Preprocessor. Folosit inițial pentru a produce pagini web dinamice, este folosit pe scară largă în dezvoltarea paginilor și aplicațiilor web. Se folosește în principal înglobat în codul HTML, dar începând de la versiunea 4.3.0 se poate folosi și în mod „linie de comandă”, permițând crearea de aplicații independente. Este unul din cele mai importante limbaje de programare web server-side open-source, existând versiuni disponibile pentru majoritatea serverelor web și pentru toate sistemele de operare. Conform statisticilor este instalat pe 20 de milioane de situri web și pe 1 milion de servere web. Este disponibil sub Licența PHP iar Free Software Foundation îl consideră a fi un software liber.

Inițial, limbajul a fost dezvoltat de inventatorul său, Rasmus Lerdorf, însă odată cu creșterea numărului de utilizatori, dezvoltarea a fost preluată de o nouă entitate, numită The PHP Group (Grupul PHP).

Aplicația *xampp* este în esență un server web open-source ce conține un server web Apache, un server de baze de date MySQL, și interpretoare de scripturi PHP și Pearl.

Utilizând toate aceste resurse, autorul a realizat un server web care permite accesul on-line la datele înregistrate de către sistemul de monitorizare a calității energiei electrice.

În figura 6.33 este prezentat panoul de configurare al serverului web xampp. Dintre toate funcționalitățile puse la dispoziție de această aplicație este folosită doar aceea de server web, celelalte fiind inactivate pentru a nu ocupa resurse.

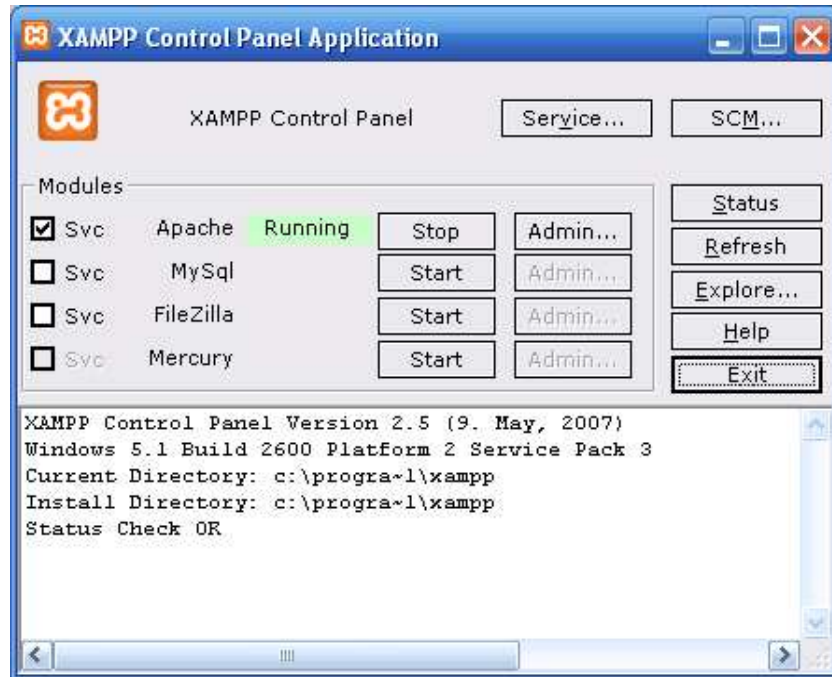


Fig. 6.33. Panoul de configurare a serverului web xampp

Serverul web poate rula pe același calculator pe care rulează sistemul de monitorizare a calității energiei electrice, sau pe un alt calculator, caz în care trebuie asigurat accesul prin rețea la datele înregistrate de sistemul de monitorizare.

Aplicația web permite realizarea următoarelor funcții:

- afișează frecvența și valoarea efectivă a tensiunii (tensiunilor) monitorizate;
- afișează un istoric al parametrilor monitorizați și trasează grafic variația acestora în timp;
- afișează grafic ultimele 10 perturbații apărute în sistem;
- permite descărcarea fișierelor înregistrate de sistemul de monitorizare;
- afișează date despre aplicație și despre modul de funcționare a acesteia;

În figura 6.34 este prezentată una dintre paginile aplicației web care afișează ultimele patru perturbații apărute în sistemul monitorizat. Este prezentată doar porțiunea afectată de perturbații, pe axa y fiind afișate valorile tensiunii în volți. Sub fiecare grafic este trecută data și ora la care au fost detectate perturbațiile respective.

În Anexa 4 sunt prezentate programele elaborate în HTML și PHP care implementează funcționalitatea serverului web.

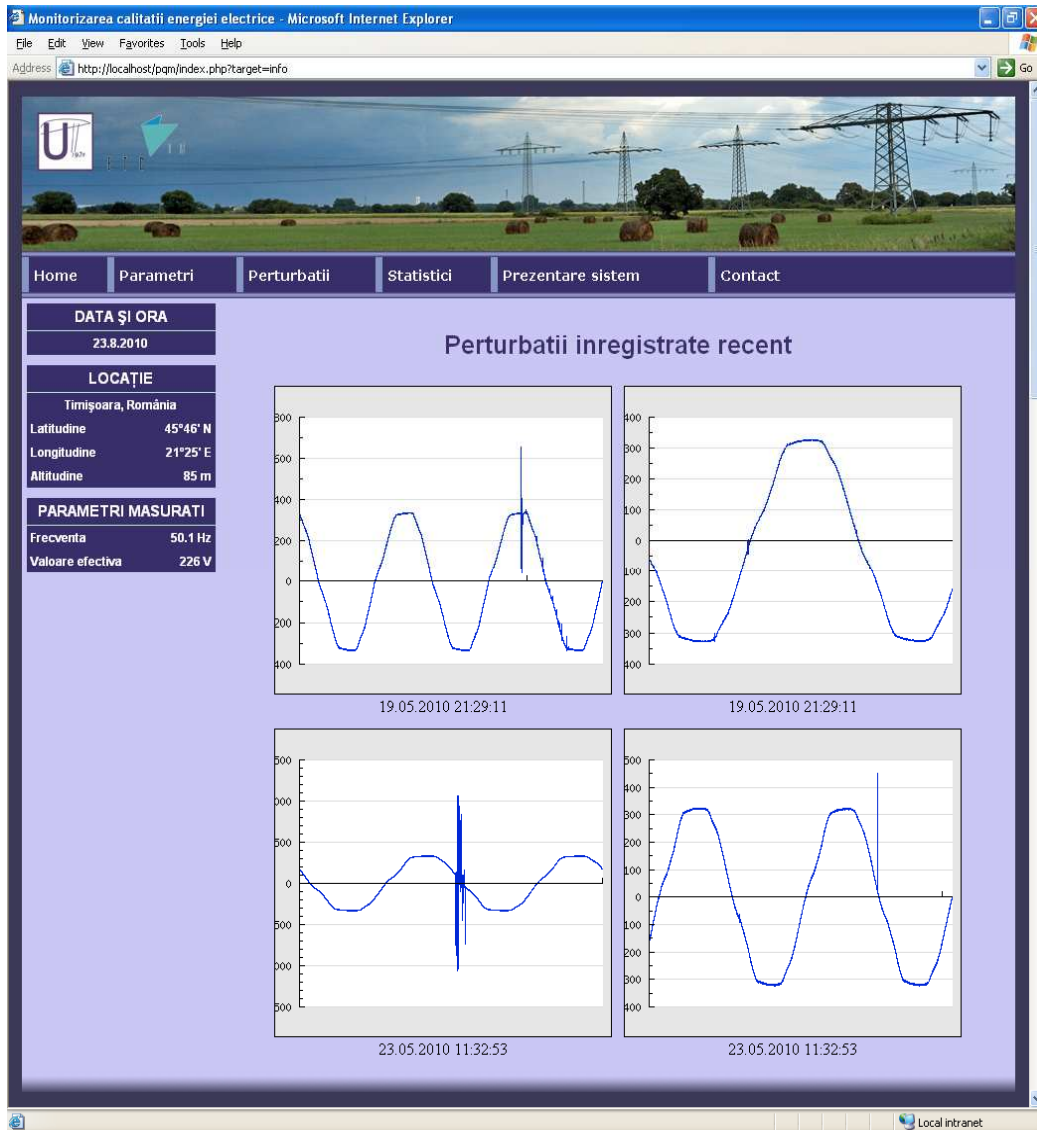


Fig. 6.34. Fereastra Perturbații a aplicației web

6.5 Concluzii

Pe baza considerațiilor prezentate s-a realizat proiectarea și execuția unui sistem funcțional pentru monitorizarea calității energiei electrice bazat pe analiza semnalelor în domeniul timp, sistem ce include atât partea hardware cât și partea software. Partea hardware a sistemului a fost proiectată cu ajutorul aplicației CAD Eagle (versiunea 5.4). Simularea comportamentului circuitului a fost făcută cu ajutorul utilitarului PSpice versiunea 9.2, iar prelucrarea datelor înregistrate cu ajutorul aplicației a fost făcută cu programe soft elaborate în MatLab 7.

A fost elaborat și implementat un algoritm de detectarea a perturbațiilor utilizând analiza semnalelor în domeniul timp, reușindu-se astfel reducerea numărului de calcule ce trebuiesc efectuate pentru detectarea perturbațiilor.

În figura 6.35 este prezentat sistemul de monitorizare a calității energiei electrice conectat la rețeaua electrică dintr-o zonă rezidențială.

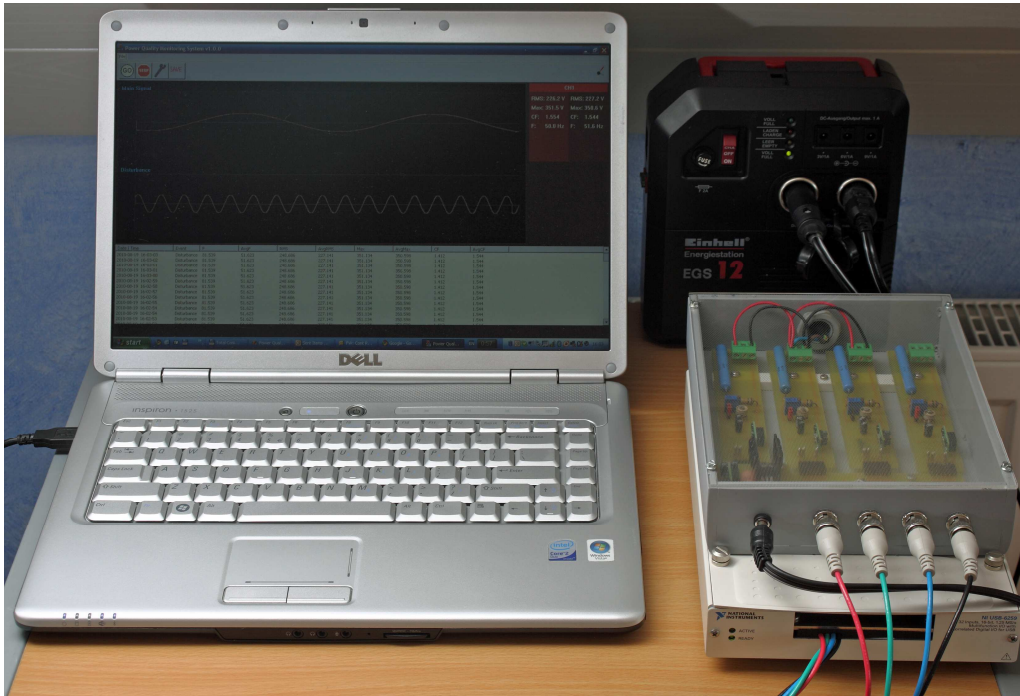


Fig. 6.35. Sistem de monitorizare a calității energiei electrice

Sistemul constă dintr-un calculator portabil pe care rulează ultima versiune a aplicației "Power Quality Monitoring System". Semnalul din rețeaua electrică este preluat cu ajutorul unui sistem de achiziție de date compus din circuitul de condiționare al semnalului prezentat în paragraful 6.2 și placa de achiziție NI USB-6259. Sistemul de achiziție de date este alimentat de la o sursă de alimentare portabilă Einhell EGS 12 care poate livra la ieșire un curent de 11 A la o tensiune nominală de 12 V. Sistemul poate funcționa astfel în orice condiții, alimentarea acestuia nefiind afectată de apariția unei eventuale întreruperi în livrarea de energie electrică.

Conform standardului SR EN 61000-4-30, sistemul elaborat de autor se încadrează în clasa A de funcționare, atât în ceea ce privește măsurarea valorii tensiunii rețelei, cât și în ce privește măsurarea frecvenței acesteia.

Sistemul propus, proiectat și realizat, afișează în timp real semnalul provenit din rețeaua electrică, parametrii semnalului (tensiune, frecvență) și ultima perturbație apărută. Aplicația afișează, de asemenea, și o listă cronologică cu ultimele perturbații apărute în situl monitorizat.

Pentru accesul facil la datele înregistrate de sistem, acolo unde nu este posibil accesul rapid la site-ul de măsurare, aplicația dispune de un server web cu ajutorul căruia pot fi vizualizați online parametrii rețelei electrice, evoluția în timp a acestora și ultimele perturbații apărute.

Pentru testarea algoritmului de detectare a perturbațiilor a fost elaborată o aplicație software care generează perturbații asemănătoare celor prezente în rețelele electrice. Perturbațiile sunt generate prin intermediul plăcii de sunet a calculatorului, semnalul generat fiind amplificat și apoi aplicat sistemului de monitorizare a perturbațiilor. Utilizând această aplicație pentru generarea de semnale perturbate a fost posibilă reglarea fină a parametrilor aplicației de monitorizare a calității energiei electrice, precum și îmbunătățirea algoritmului de detecție a perturbațiilor.

Un prim pas în portarea sistemului de monitorizare a calității energiei electrice pe microcontrolere a fost deja făcut, autorul reușind implementarea aplicației pe un sistem cu microcontroler MSP430 [112] dar, datorită limitărilor sistemului utilizat, rata maximă de achiziție a semnalelor a fost de doar 10 ksps, mult prea mică față de cerințele unui sistem de monitorizare a calității energiei electrice modern din punctul de vedere al frecvenței maxime a semnalelor preluate. Se impune astfel, mărirea ratei de achiziție prin utilizarea unui sistem cu microcontroler mai performant sau prin utilizarea unui procesor de semnal.

Autorul dorește să abordeze pe viitor oportunitatea realizării unui sistem de monitorizare a valorilor efective ale tensiunii rețelei utilizând voltmetrul numeric de valoare efectivă Fluke în modul event-logging, similar sistemului de monitorizare a câmpului electric prezentat în [113] și [114].

7. REZULTATE EXPERIMENTALE

Sistemul de monitorizare a calității energiei electrice prezentat în capitolul 6 a fost pus în funcțiune pentru prima oară în cadrul Universității "Politehnica" din Timișoara, sistemul fiind instalat ulterior și în alte locații.

Sistemul a fost instalat în următoarele locații:

- Facultatea de Electronică și Telecomunicații din Timișoara;
- Facultatea de Mecanică din Timișoara;
- o reședință privată din orașul Recaș, județul Timiș, locație în care apar frecvent perturbații în rețeaua electrică soldate cu defectarea echipamentelor conectate la aceasta;
- o altă zonă rezidențială rurală din Germania;
- un sistem de producere a energiei electrice bazat pe panouri fotovoltaice.

În toate cele cinci locații s-au obținut rezultate interesante, măsurările contribuind la înțelegerea fenomenelor de producere a perturbațiilor, identificarea surselor perturbatoare respectiv, motivului defectării unor echipamente conectate la rețelele de alimentare respective.

S-au monitorizat doar rețele monofazate, configurația sistemului fiind cea din figura 6.8, achiziția făcându-se pe trei canale. Rata de achiziție a semnalelor a fost de 250 ksp/s pentru marea majoritate a măsurărilor, cu excepția unei singure situații, în care s-a utilizat o rată de achiziție de 1MSPS pentru a testa la maxim performanțele sistemului. Acolo unde nu se specifică rata de eșantionare se presupune ca fiind de 250 ksp/s.

Calculatorul pe care a rulat aplicația a fost un notebook Dell Inspiron 1525 echipat cu un procesor Intel Core 2 Duo T5550 la 1,83GHz, cu 4 GB de memorie RAM. În cazul utilizării aplicației la o rată de eșantionare de 250 ksp/s pe canal, la achiziția a trei canale (echivalentul eșantionării unui singur canal cu o rată de 750 ksp/s), gradul mediu de ocupare al procesorului a fost de 55 %, iar în cazul eșantionării unui singur canal cu o rată de 1 Msp/s, gradul de ocupare al procesorului a fost cuprins între 99 % și 100 %, sistemul funcționând în parametri proiectați.

Pentru a testa funcționarea sistemului, au fost conectați la rețeaua electrică diverși consumatori, cunoscuți fiind ca surse de perturbații, rezultatele fiind prezentate în cele ce urmează.

În figura 7.1 se prezintă un semnal perturbat cu un impuls tranzitoriu, impuls generat de conectarea unui laptop la rețeaua electrică. Astfel de impulsuri sunt frecvente în rețelele electrice rezidențiale și metropolitane, amplitudinile acestor impulsuri fiind, în cazul măsurărilor efectuate, în limitele prevăzute de standarde.

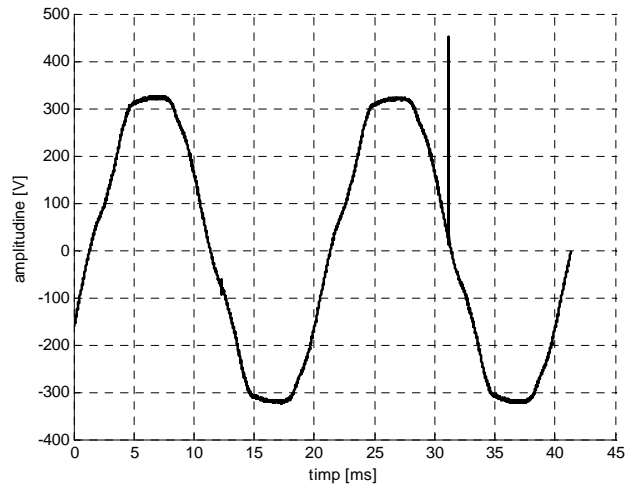


Fig. 7.1. Perturbație de tip impuls tranzitoriu

În figura 7.2 este prezentat un semnal perturbat cu o serie de impulsuri generate de către un motor cu perii. Un detaliu al acestei perturbații este prezentat în figura 7.2 b).

Rata de eșantionare folosită a fost de 1 Msp/s, pentru a avea o bandă de frecvențe cât mai mare.

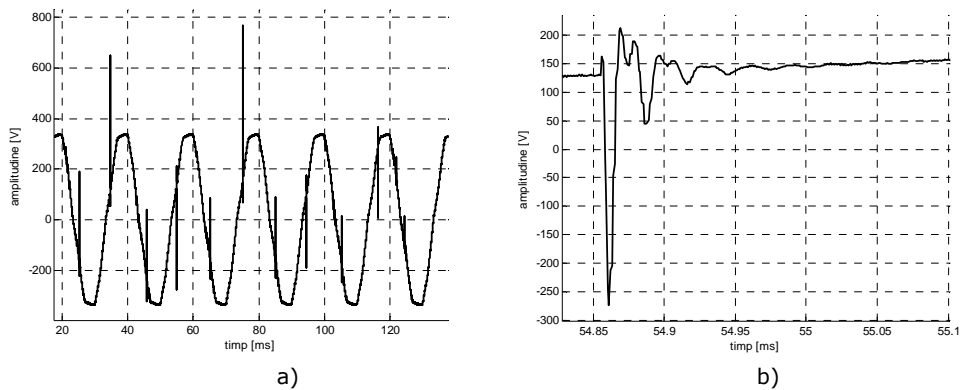


Fig. 7.2. Perturbație de tip impuls tranzitoriu. a) semnal perturbat, b) detaliu

În figura 7.3 se prezintă o altă perturbație de tip impuls tranzitoriu generată la conectarea la rețea a unei surse de calculator de 400 W. Se poate observa prezența unor perturbații mai mici ca amplitudine decât perturbația inițială.

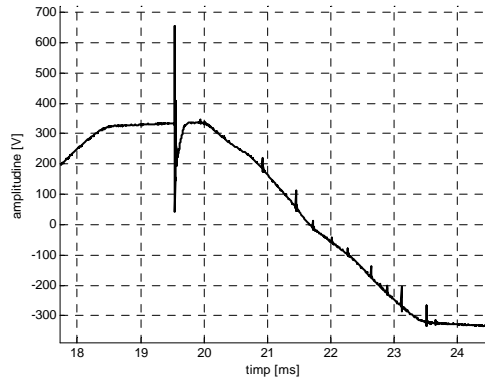


Fig. 7.3. Perturbație de tip impuls tranzitoriu

În site-ul de măsurare de la Recaș au fost identificate perturbațiile prezentate în figura 7.4, perturbații datorate intrării în funcție a motorului unui frigider conectat la rețea pe aceeași fază cu sistemul de monitorizare.

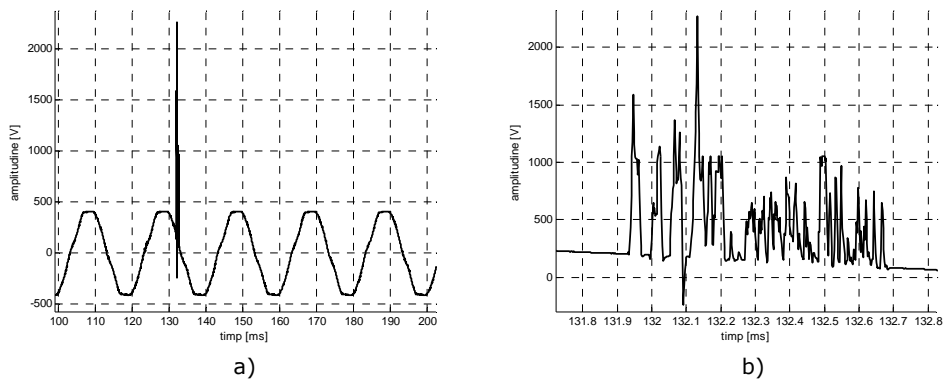


Fig. 7.4. Perturbație de tip impuls tranzitoriu. a) semnal perturbat, b) detaliu

Originea perturbației din figura 7.5 nu este cunoscută, ea fiind surprinsă în timpul monitorizării.

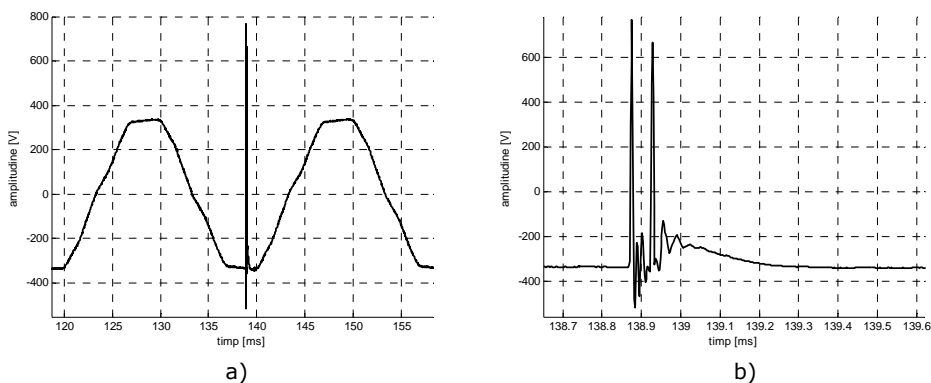


Fig. 7.5. Perturbație de tip impuls tranzitoriu. a) semnal perturbat, b) detaliu

În figura 7.6 sunt prezentate două detalii ale unor perturbații înregistrate în site-ul de măsurare de la Recaș.

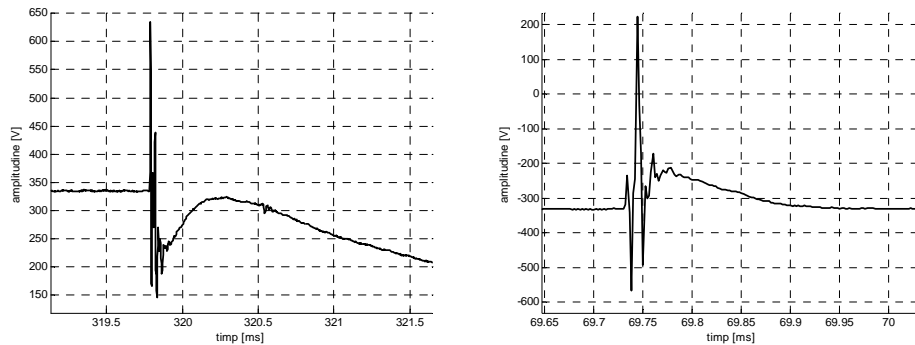


Fig. 7.6. Perturbații de tip impuls

Sistemul de monitorizare a fost amplasat și în cadrul Facultății de Mecanică, deoarece alimentarea cu energie electrică a mașinilor unelte este realizată din altă sursă de energie decât cea a Facultății de Electronică. S-a dorit monitorizarea calității energiei electrice de aici și compararea acesteia cu cea din Facultatea de Electronică și Telecomunicații.

Au fost detectate câteva perturbații pe parcursul celor două zile cât a fost monitorizată calitatea energiei electrice la Facultatea de Mecanică, cele mai multe dintre ele fiind perturbații de tipul căderilor de tensiune, generate de echipamentele de putere conectate la rețea. În figura 7.7 sunt prezentate două perturbații produse de o mașină de testare a rezistenței la îndoire și rupere a barelor de metal.

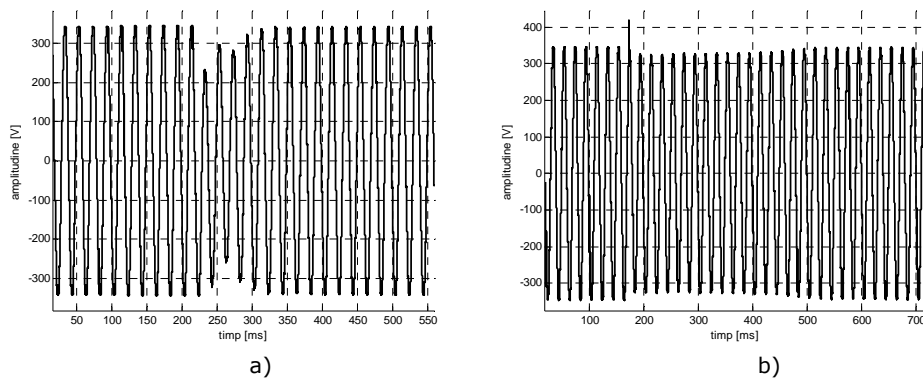


Fig. 7.7. Perturbații a) cădere de tensiune, b) cădere de tensiune și impuls tranzitoriu

Această mașină este echipată cu o pompă hidraulică acționată de un motor electric asincron trifazat cu puterea de 18 kVA.

Pe durata pornirii instalației, timp în care este pornită pompa hidraulică și alte echipamente mecanice, au fost înregistrate căderi de tensiune și perturbații de tip impuls singular, evoluția în timp a valorii efective a semnalului rețelei fiind prezentată în figura 7.8. În acest caz s-a achiziționat tensiunea rețelei de pe o singură fază.

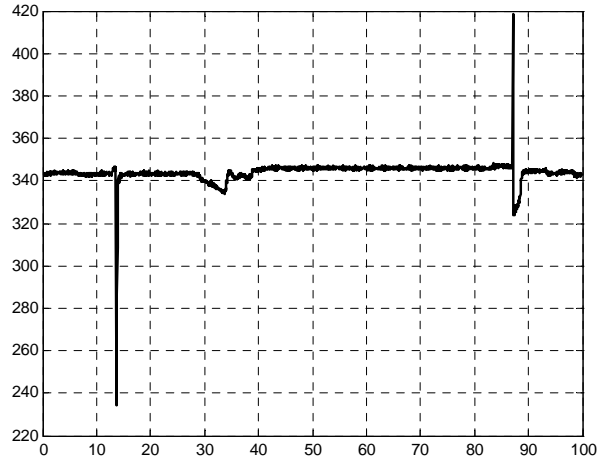


Fig. 7.8. Variația valorii maxime a tensiunii rețelei la pornirea unui motor asincron de 18 kVA

În continuare se vor prezenta, spre comparație, formele de undă ale tensiunii din site-urile măsurate și o exemplificare a posibilităților de prelucrare ulterioară, de exemplu spectrul de frecvențe.

În figura 7.9 este prezentată o perioadă a tensiunii rețelei electrice a Facultății de Electronică și Telecomunicații din Timișoara împreună cu spectrul acesteia.

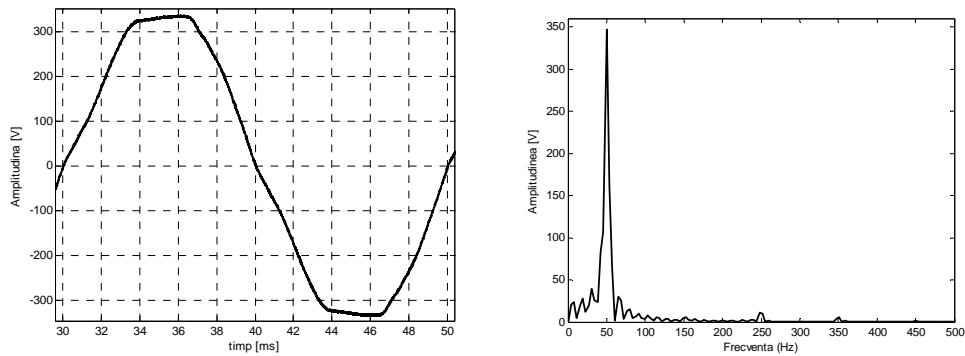


Fig. 7.9. Semnalul din rețeaua electrică a Facultății de Electronică și Telecomunicații și spectrul acestuia

Se observă forma distorsionată a semnalului sinusoidal, prezența unor subarmonici și respectiv componenta armonică de ordinul 5.

În figura 7.10 este prezentat semnalul preluat din rețeaua electrică a Facultății de Mecanică și spectrul acestuia care conține componentele armonice de ordinul 5,7 și 9.

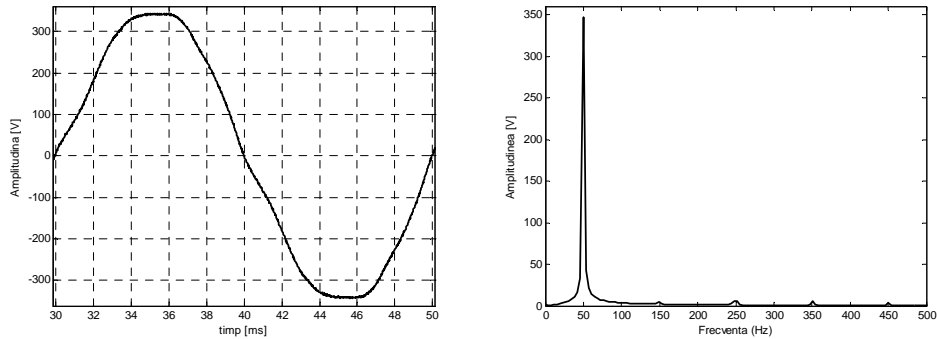


Fig. 7.10. Semnalul din rețeaua electrică a Facultății de Mecanică și spectrul acestuia

În figura 7.11 este prezentat semnalul preluat din rețeaua electrică a unei zone rezidențiale puțin încărcate. Se poate aprecia vizual forma semnalului, acesta fiind cel mai apropiat de forma sinusoidală dintre toate cele trei semnale comparate. Analizând spectrul semnalului se observă valori foarte mici ale armonicilor de ordin superior (practic doar componenta armonică de ordinul 5).

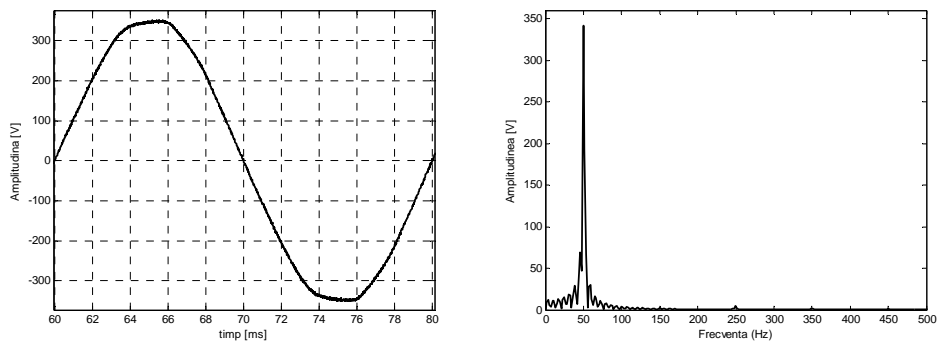


Fig. 7.11. Semnalul rețelei electrice rurale din Germania și spectrul acestuia

Perturbații de tipul căderi de tensiune au fost înregistrate și în site-ul de măsurare din zona rezidențială în timpul unei furtuni (figura 7.12).

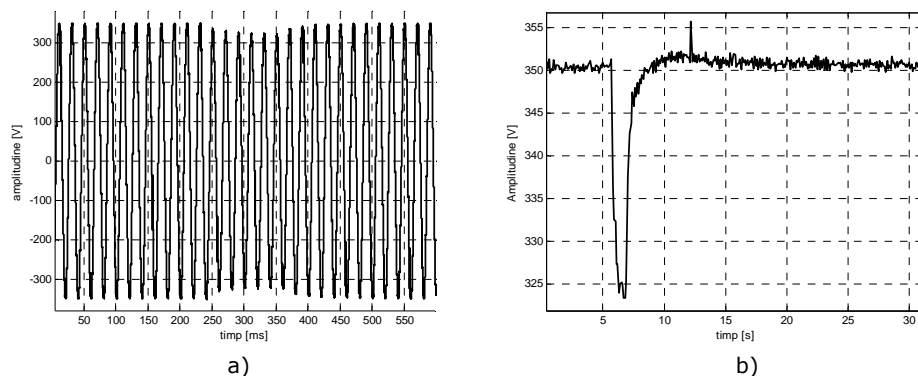


Fig. 7.12. Cădere de tensiune a) tensiune, b) evoluția în timp a valorii maxime a tensiunii rețelei în timpul golului de tensiune

În figura 7.13 se prezintă două perturbații de tipul impuls tranzitoriu, suprapuse peste tensiunea rețelei, înregistrate în site-ul de măsurare de la Recaș. Amplitudinea relativ mică a perturbațiilor confirmă sensibilitatea și performanțele sistemului realizat.

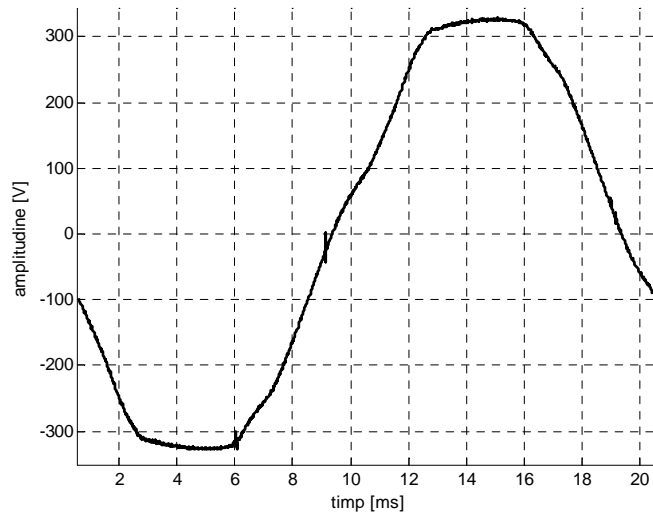


Fig. 7.13. Perturbații de scurtă durată

De asemenea, sistemul de monitorizare a perturbațiilor a fost instalat în cadrul unui sistem fotovoltaic de producere a energiei electrice (figura 7.14), pentru determinarea comportamentului acestuia în cazul în care energia regenerabilă captată de sistem devine indisponibilă, moment în care se trece automat la alimentarea consumatorilor cu energie electrică preluată din sistemul energetic național.

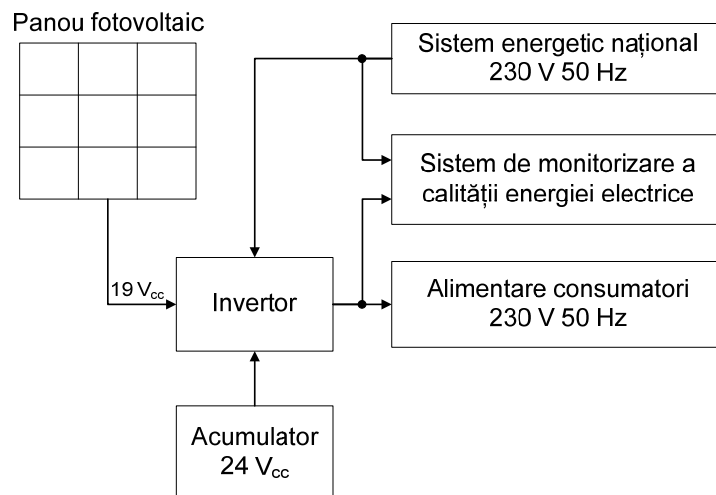


Fig. 7.14 Conectarea sistemului de monitorizare a calității energiei electrice la un sistem fotovoltaic

Sistemul fotovoltaic captează energia soarelui (energie regenerabilă) prin intermediul panoului fotovoltaic și o convertește, cu ajutorul unui inverter, în energie electrică cu tensiunea de 230 V și frecvența de 50 Hz, alimentând o serie de consumatori din rețeaua electrică locală a unui laborator. Energia în surplus este stocată într-un acumulator de 24 V, urmând a fi utilizată noaptea, sau atunci când energia captată de panoul fotovoltaic este redusă. Cât timp panoul solar generează suficientă energie sau cât timp există suficientă energie în baterie, inverterul preia energia de la acestea și o convertește în tensiune electrică cu valoarea efectivă de 230 V și frecvența de 50 Hz. Noaptea, după ce energia stocată în acumulator se epuizează, inverterul preia în mod automat energia electrică din sistemul național și o livrează consumatorilor. Atunci când energia provenită de la soare atinge un prag prestabilit, inverterul folosește din nou energia soarelui pentru producerea tensiunii de ieșire. În momentul comutării tensiunii de ieșire între rețeaua națională și sursa de energie regenerabilă, în sistemul de alimentare cu energie electrică al laboratorului pot să apară perturbații care să deranjeze echipamentele conectate la această rețea. Astfel, se impune monitorizarea rețelei electrice a laboratorului pentru a detecta eventualele perturbații ce pot să apară.

Sistemul de monitorizare a calității energiei electrice propus de autor în cadrul tezei de doctorat a fost conectat atât la rețeaua electrică a laboratorului cât și la rețeaua electrică națională, utilizând configurația de intrare prezentată în figura 6.9. Sistemul a fost conectat la ambele rețele electrice pentru a urmări dacă tensiunea generată de inverter este sincronă cu tensiunea rețelei naționale și pentru a se putea monitoriza perturbațiile introduse în ambele rețele la comutarea de pe sursa de energie regenerabilă pe rețeaua electrică națională.

În urma monitorizării celor două sisteme electrice s-au depistat probleme la comutarea de pe sursa de energie regenerabilă pe sistemul național, cauzate de lipsa sincronizării tensiunii generate de inverter cu tensiunea rețelei naționale și de arcul electric ce se produce între lamelele releului care asigură comutarea între rețele.

În figura 7.15 este prezentat momentul în care are loc comutarea alimentării cu tensiune de la sistemul energetic național la generatorul de tensiune bazat pe energia regenerabilă, moment în care, datorită elementelor de comutare, apare o perturbație de tip impuls tranzitoriu de scurtă durată cu o valoare de 857 V.

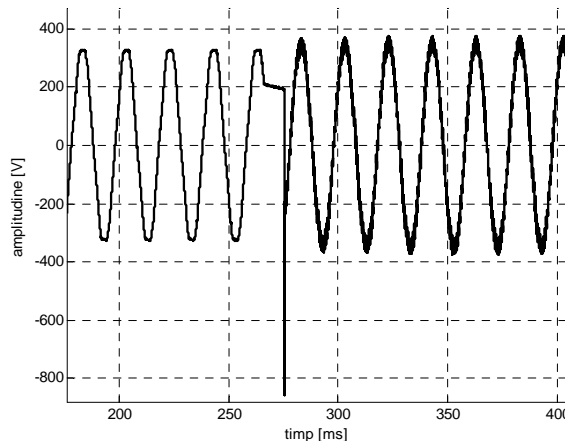


Fig. 7.15 Trecerea alimentării consumatorilor de la sistemul energetic național la sursa de energie regenerabilă

În figura 7.16 se prezintă comparativ semnalul rețelei energetice naționale și semnalul generat la ieșirea inverterului alimentat din sursa de energie regenerabilă.

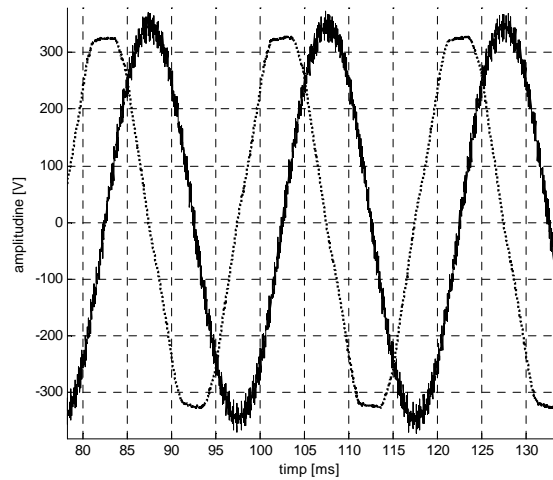


Fig. 7.16 Semnalul rețelei energetice naționale (punctat) și semnalul generat de inverter (linie continuă)

Deși semnalul generat de inverter este însoțit de zgomot, componentele armonice se încadrează în limitele impuse de standard, ele fiind mai mici de 8 % din valoarea tensiunii nominale (figura 7.17). Analizând figura 7.17 b) se observă prezența în spectrul semnalului generat a unor componente spectrale cu frecvențe ridicate, spectrul semnalului întinzându-se peste 50 kHz, putând afecta buna funcționare a unor echipamente mai sensibile conectate la acesta.

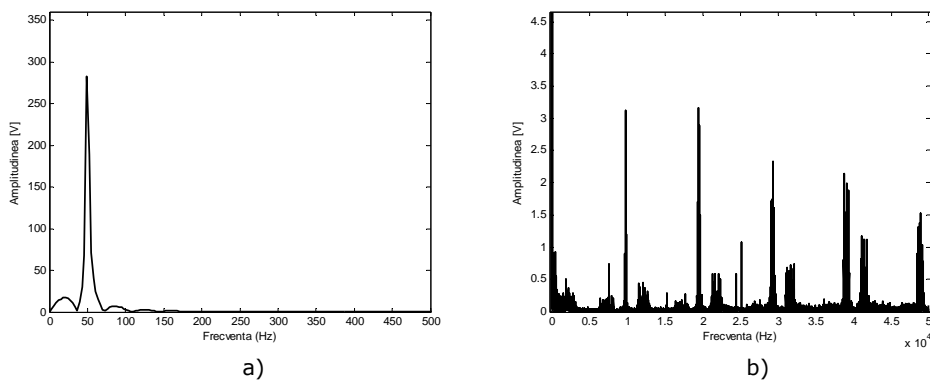


Fig. 7.17 Spectrul semnalului generat de inverter
a) spectrul semnalului la frecvențe joase
b) spectrul semnalului la frecvențe ridicate (detaliu)

8. CONCLUZII ȘI CONTRIBUȚII

Pe parcursul ultimelor patru decenii, în domeniul monitorizării calității energiei electrice s-a desfășurat o activitate de cercetare intensă concretizată prin elaborarea unui număr foarte mare de articole și publicații, precum și prin apariția unor noi standarde în domeniu sau îmbunătățirea celor existente. La baza detectării perturbațiilor din rețelele electrice stau metode matematice complexe care utilizează transformatele Fourier sau Wavelet, rețelele neuronale, logica fuzzy sau analiza în domeniul timp. Preocuparea continuă a multor cercetători din întreaga lume în elaborarea de noi metode de detectare a perturbațiilor din rețelele electrice pe de o parte și construcția de echipamente de monitorizare pe de altă parte, confirmă actualitatea temei abordate de autor.

Autorul a ales analiza în domeniul timp a tensiunii rețelei, deoarece această metodă de analiză necesită un număr mult mai mic de calcule pentru detectarea perturbațiilor, putându-se realiza prelucrarea unui număr mai mare de eșantioane ale semnalului în cazul utilizării aceluiași sistem de calcul comparativ cu celelalte metode și prin aceasta, extinderea benzii de frecvențe a aplicației.

Printre preocupările actuale în ceea ce privește monitorizarea calității energiei electrice se numără și creșterea ratei de eșantionare a tensiunii rețelei, fapt care conduce la creșterea spectrului de frecvențe a semnalului achiziționat, putându-se evidenția mai bine desfășurarea în timp a perturbațiilor de foarte scurtă durată.

Autorul, în urma cercetărilor și experimentelor efectuate, și-a adus contribuția la dezvoltarea acestui domeniu după cum urmează:

- A realizat un studiu bibliografic amplu asupra stadiului actual al monitorizării calității energiei electrice, evidențiind actualitatea temei, studiu bazat pe analiza a peste o sută de titluri bibliografice incluzând articole, cărți, standarde și site-uri web.
- Pe baza cercetărilor și experimentelor efectuate, autorul a elaborat 19 lucrări științifice (coautor la 3 cărți și autor sau coautor la 16 lucrări științifice, dintre care una cotate ISI).
- Pentru compresia gamei dinamice a tensiunii monitorizate, s-a analizat oportunitatea folosirii unui transformator funcțional ca și circuit de condiționare a semnalelor. Autorul a simulat circuitul utilizând utilitarul PSpice și a identificat principalele probleme pe care le prezintă acesta - neliniarități și comportarea la variațiile de temperatură - elaborând în urma studiilor efectuate - o soluție practică pentru rezolvarea acestor probleme. Soluția constă în utilizarea unui circuit de calibrare a funcției de transfer a transformatorului funcțional. Calibrarea circuitului se realizează prin aplicarea unui semnal triunghiular la intrare, analizându-se răspunsul circuitului la acest semnal și calculându-se apoi o funcție de corecție a caracteristicii transformatorului funcțional. Autorul a studiat (cu ajutorul simulărilor realizate în MatLab), comportamentul acestui tip de circuit, cu și fără circuitul de calibrare și a arătat

utilitatea folosirii circuitului de calibrare propus, evidențiind faptul că și acest circuit simplu, cunoscut încă de la apariția circuitelor integrate analogice, poate fi folosit ca și circuit de condiționare a semnalelor.

- A studiat o metodă de detecție a perturbațiilor prezentată în literatura de specialitate bazată pe media nulă pe o perioadă a unui semnal sinusoidal și a îmbunătățit-o prin adăugarea unui sistem de detecție a perioadelor semnalului rețelei, măbind astfel probabilitatea detectării perturbațiilor. Autorul a inclus în sistem o memorie circulară cu scopul de a memora temporar eşantioanele semnalului în vederea înregistrării ulterioare a acestora, atunci când sunt detectate perturbații. Prin utilizarea memoriei circulare, autorul a eliminat necesitatea folosirii unui al doilea canal de achiziție a semnalelor și a circuitului de întârziere în vederea înregistrării semnalului perturbat.
- Ținând seama de gama dinamică mare a perturbațiilor din rețelele de alimentare, dar fără afectarea sensibilității din jurul tensiunii nominale, a elaborat un sistem original de compresie a semnalelor, bazat pe utilizarea a trei divizoare de tensiune rezistive conectate la trei intrări ale unui convertor analog – numeric (divizor secvențial de tensiune). Fiecare divizor de tensiune are un alt raport de divizare, iar prin compunerea software a caracteristicilor de transfer ale celor trei divizoare de tensiune se obține o caracteristică de transfer similară cu cea a transformatorului funcțional. A rezultat astfel un circuit de compresie simplu și eficient, care poate fi folosit ca și circuit de condiționare a semnalelor în sistemele de monitorizare a calității energiei electrice într-o gamă dinamică mare.
- A elaborat un algoritm software cu ajutorul căruia semnalele preluate de la divizorul secvențial de tensiune sunt prelucrate și procesate exploatând la maxim dinamica semnalelor prezente în rețeaua electrică. Pe baza acestui sistem, autorul propune un algoritm extrem de simplu de detectare a perturbațiilor cu amplitudini de peste 500 V, bazat pe intrarea în limitare a canalului cu cel mai mic raport de divizare. Prin compararea semnalelor provenite de la cele trei intrări ale plăcii de achiziție se determină simplu prezența unei perturbații cu amplitudine mai mare de 500 V. Deoarece se folosesc doar două funcții de test *if* pentru detectarea unei perturbații, metoda este una dintre cele mai simple și rapide metode de detectare a perturbațiilor din rețelele electrice și poate fi folosită acolo unde se dorește detectarea perturbațiilor cu amplitudini mari (peste 500 V). Această metodă se pretează la a fi implementată în cadrul sistemelor de monitorizare a calității energiei electrice cu microcontroler, conducând (prin reducerea numărului de calcule efectuate pentru detecția unei perturbații), la creșterea ratei de eşantionare, lucru benefic analizei ulterioare a formei perturbației înregistrate de acestea într-o gamă mai largă de frecvențe.
- A realizat un sistem de decompresie software a semnalelor provenite de la divizorul secvențial de tensiune, bazat pe aplicarea unei funcții matematice corespunzătoare asupra semnalului comprimat, simplificând astfel partea electrică a circuitului de condiționare a semnalelor. Pe baza acestui sistem, prin prelucrare numerică, pot fi

analizate atât detalii de interes fine, cum ar fi variațiile din jurul valorii nominale ale tensiunii rețelei, cât și perturbațiile de amplitudine ridicată de ordinul kilovolților.

- Autorul a demonstrat, prin calcule matematice, creșterea rezoluției sistemului de achiziție de date prin folosirea circuitului de condiționare a semnalelor cu divizor secvențial de tensiune propus de acesta. Deși sistemul folosește trei canale de intrare ale unui sistem de achiziție de date pe 16 biți, din punct de vedere al rezoluției, acesta este echivalent cu folosirea unui sistem de achiziție de date cu un singur canal pe 21 de biți (utilizând un convertor analog - numeric cu o caracteristică neliniară, similară unei cuantizări neuniforme). Se obține astfel o creștere a rezoluției sistemului de monitorizare a calității energiei electrice.
- Pe baza cercetărilor efectuate, autorul a realizat practic un sistem original și autonom de monitorizare a calității energiei electrice (funcționând în Clasa A, conform standardului SR EN 61000-4-30), sistem care include atât o parte hardware, cât și dezvoltarea de programe software care să asigure desfășurarea întregii aplicații. Cu un astfel de sistem se poate asigura o monitorizare continuă de circa zece zile.
- De asemenea, s-a realizat un circuit de condiționare a semnalelor modular, care poate fi reconfigurat cu ușurință în funcție de necesitățile utilizatorului. Astfel, sistemul poate fi utilizat fie la monitorizarea unei rețele electrice monofazate, a unei rețele trifazate (stea sau triunghi), sau a două sau mai multe rețele monofazate simultan, soluție ce ar putea folosită la supravegherea sistemelor neconvenționale de producere de energie electrică care sunt cuplate la rețeaua de alimentare. Sistemul de condiționare asigură și izolarea semnalelor preluate din rețeaua electrică de restul sistemului de achiziție de date, asigurând astfel, reducerea cuplajelor parazite, creșterea nivelului de securitate și reducerea riscului de electrocutare.
- A elaborat o metodă nouă de detecție a perturbațiilor din rețelele electrice bazată pe analiza semnalelor în domeniul timp. Autorul calculează principalii parametri ai semnalului rețelei electrice (tensiune efectivă, tensiune maximă, factor de vârf, frecvență), pe o semiperioadă a semnalului rețelei și compară aceste valori cu media valorilor pe 10 perioade ale semnalului. Dacă există diferențe între valorile instantanee și mediile acestora, mai mari decât o valoare de referință δ (stabilită la calibrarea sistemului), atunci se semnalează prezența unei perturbații în sistem. Pentru detecția semiperioadelor semnalului rețelei, autorul folosește, în locul metodei de detecție a trecerilor prin zero, metoda detecției minimelor semnalului redresat al rețelei, metodă care reduce mult numărul de calcule.
- Autorul a realizat o aplicație software pentru monitorizarea calității energiei electrice, utilizând sistemul original de detectare a perturbațiilor prezentat în teza de doctorat. Aplicația a fost elaborată în Visual C++ utilizând sistemul de clase MFC de la Microsoft. Aplicația poate fi reconfigurată de către utilizator pe baza unui fișier de configurare. Astfel se pot modifica parametri precum: rata de eșantionare, numărul de canale, modul de conectare al canalelor de

intrare, tipul rețelei electrice conectată la intrare (monofazată sau trifazată). Aplicația permite afișarea în timp real a semnalului rețelei, a perturbației detectate și a parametrilor semnalului rețelei: tensiune efectivă, tensiune maximă și frecvență, afișând de asemenea și un istoric al perturbațiilor apărute în rețeaua electrică. În vederea studierii ulterioare, perturbațiile detectate sunt înregistrate în fișiere separate, marcându-se totodată și momentul apariției fiecărei perturbații. De asemenea, sistemul de monitorizare înregistrează valorile medii pe o semiperioadă ale următorilor parametri: tensiune efectivă, tensiune maximă, durată semiperioadă, pentru a putea fi analizate ulterior. Înregistrarea acestor parametri se face continuu, fără a fi perturbată de apariția unei eventuale perturbații.

- Prin elaborarea de programe software adecvate pentru a avea o interfață cât mai prietenoasă cu operatorul, a realizat controale grafice pentru afișarea semnalului de intrare, a perturbațiilor, și un control grafic pentru vizualizarea stării memoriei circulare, precum și funcții de configurare automată a aplicației.
- În vederea verificării sistemului, a elaborat o aplicație Visual C++ care permite testarea funcționalității sistemului propus, aplicație care asigură generarea de semnale cu frecvența de 50 Hz peste care se pot suprapune diverse tipuri de perturbații. Aplicația poate rula pe orice calculator personal pe care este instalat sistemul de operare Microsoft Windows și care este dotat cu o placă de sunet. Datorită interpretorului original de funcții matematice elaborat de autor, aplicația poate genera inclusiv semnale matematice complexe, pornind de la funcții matematice simple și/sau de la semnale preînregistrate în fișiere.
- De asemenea, a realizat un server web care permite accesul de la distanță la parametrii înregistrați de aplicație, vizualizarea evoluției lor în timp și vizualizarea ultimelor perturbații apărute. Pentru implementarea serverului web, autorul a folosit utilitarul *xampp*, pe care l-a configurat astfel încât să corespundă scopului urmărit. Autorul a elaborat programe în limbajele *HTML* și *PHP* pentru afișarea în timp real a datelor preluate de la aplicația de monitorizare a calității energiei electrice, precum și pentru afișarea grafică a perturbațiilor detectate în rețeaua electrică. Prin intermediul serverului web pot fi descărcate de la distanță fișierele cu perturbații și valorile parametrilor rețelei electrice, mărind astfel autonomia sistemului de monitorizare, fără a fi nevoie de prezența unui operator uman în site-ul monitorizat.
- Cu sistemul de monitorizare realizat, a efectuat experimente de monitorizare în cinci rețele electrice diferite, din două țări europene, în perioade de timp cuprinse între 15 martie 2010 - 31 august 2010 și a analizat rezultatele obținute; astfel s-a constatat eficiența sistemului și corectitudinea soluțiilor propuse de autor.
- Folosind sistemul de monitorizare a calității energiei electrice prezentat în teza de doctorat, autorul a reușit să surprindă o serie de perturbații din rețeaua electrică: impulsuri tranzitorii cu amplitudini de circa 2 kV, căderi de tensiune, supratensiuni, goluri de tensiune.

- Referitor la viteza de lucru a sistemului, s-a reușit achiziția tensiunii rețelei electrice cu o rată de un milion de eșantioane pe secundă și prelucrarea acestora în timp real, lucru extrem de important pentru înregistrarea fidelă a formei de undă a perturbațiilor de scurtă durată și deci, asigurarea unei benzi de frecvențe a achiziției de ordinul sutelor de kiloherți, suficientă pentru majoritatea aplicațiilor de monitorizare.

Ca și concluzii, se poate afirma că sistemul propus și realizat de autor poate fi îmbunătățit prin optimizarea algoritmilor de detectare a perturbațiilor și prin mărirea ratei de achiziție a semnalului. Autorul preconizează că în urma acestor optimizări (și prin înlocuirea plăcii de achiziție utilizate cu alta cu o rată de eșantionare mai mare), aplicația de monitorizare a calității energiei electrice să poată procesa în timp real semnale achiziționate cu o rată de până la 10 Msps.

Pe viitor, autorul își propune să studieze implicațiile asupra rezoluției semnalului de ieșire, în urma folosirii mai multor canale de achiziție în circuitul de compresie cu divizor secvențial de tensiune.

Autorul va implementa pe viitor aplicația de monitorizare a energiei electrice pe sisteme de dezvoltare dotate cu procesoare digitale de semnal – sisteme dedicate (Embedded Systems) - obținându-se astfel miniaturizarea sistemului de monitorizare, dar și un preț de cost mai redus. Deoarece unele fenomene pot fi surprinse mult mai bine prin analiza curentului absorbit de consumatori [115], [116], [117], autorul își propune să dezvolte sistemul de monitorizare cu circuite de condiționare care să accepte la intrare și curenți. Se poate realiza astfel un sistem de monitorizare care să permită aplicarea la intrare atât a unor tensiuni, cât și a unor curenți, lărgindu-se astfel aria de aplicabilitate și oportunitățile de utilizare ale acestuia.

BIBLIOGRAFIE

- [1] F. D. Martzloff, T. M. Gruz, "Monitoring Power Quality", in *Powertechnics Magazine*, 1990.
- [2] M. Roșca, "Supravegherea centralizată a calității energiei electrice", in *A XVI-a conferință confort, eficiență, conservarea energiei și protecția mediului* București.
- [3] R. P. K. Lee, L. L. Lai, N. Tse, "A Web-based multi-channel power quality monitoring system for a large network", in *Fifth International Conference on Power System Management and Control*. vol. 488, pp. 112 - 117.
- [4] B. E. Kushare, A. A. Ghatol, S. Kala, "Development of web based power quality monitoring system for handling user custom power quality query and auto power quality monitoring report notification via email", in *IET-UK International Conference on Information and Communication Technology in Electrical Sciences (ICTES 2007)* Chennai, Tamil Nadu, India, pp. 1-7.
- [5] J. Nanjing, "Power Quality Monitoring System over the Internet", in *2009 First International Conference on Information Science and Engineering*.
- [6] S. Nikolovski, Z. Klaic, Z. Kraus, G. Slipac, "Online Power Quality Measurements and Voltage Sags Analysis", in *43rd International Engineering Conference (UPEC 2008)* Padova, Italy, pp. 1-5.
- [7] K. M. Chan, *Power Quality - REFERENCE GUIDE, 6th edition*. Ontario, Canada: In-House Energy Efficiency, 1996.
- [8] S. Santoso, H. W. Beaty, R. C. Dugan, M. F. McGranaghan, *Electrical Power Systems Quality; 2 edition* McGraw-Hill Professional, 2002.
- [9] H. Zhijian, C. Chengxue, C. Ruifan, "Harmonic Real Time Estimation by Artificial Neural Network Based on Network and GPS Technology", in *The 7th International Power Engineering Conference*. vol. 2: IPEC 2005, pp. 1182 - 1187
- [10] B. Yunus, H. Li, "Analysis of Power Quality Waveform for Data Transmission Efficiency over IEC 61850 Communication Standard", in *IEEE International Power and Energy Conference*, Putra Jaya pp. 161 - 166.
- [11] I. Dobson, B. A. Carreras, V. E. Lynch, O. Ridge, D. E. Newman, "Complex Systems Analysis of Series of Blackouts: Cascading Failure, Criticality, and Self-organization", in *Bulk Power System Dynamics and Control - VI Cortina d'Ampezzo*, Italy, 2004.
- [12] J. D. Aspnos, B. W. Evans, R. P. Merrit, "Rural Alaska Electric Power Quality", *IEEE Transactions PAS-104*, March 1995.
- [13] L. Shusen, C. Xiaoyun, Z. Zhengjun, L. Liangjun, "The study 500 kV system harmonics in Central China Power Network", in *Proceedings of the 8th International Conference on Harmonics And Quality of Power*, Athens, Greece, pp. 814 - 819.
- [14] R. Belmans, H. D. Keulenaer, J. Manson, E. Schellekens, "Towards improved electrical installations in European homes", 2004.
- [15] M. F. Akram, S. M. Bajwa, "A Sample Power Quality Survey for Emerging Competitive Electricity Market in Pakistan", in *IEEE International Multi Topic*

- Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century.*, 2001, pp. 38 - 44
- [16] H. M. S. C. Herath, S. McHardy, "Power Quality Trends in EnergyAustralia Distribution Network", in *13th International Conference on Harmonics and Quality of Power, 2008. ICHQP 2008.*, Wollongong, NSW, 2008, pp. 1 - 6.
- [17] D. M. Divan, W. E. Brumsickle, G. A. Luckjiff, J. W. Freeborg, R. L. Hayes, "Realizing a Nation-wide Power Quality and Reliability Monitoring System", in *Conference Record of the Industry Applications Conference, 2003. 38th IAS Annual Meeting.*, pp. 1063 - 1067.
- [18] ***, "EMC Directive 89/336/EEC".
- [19] "Ordin privind aprobarea listei cuprinzand standardele romane care adopta standardele europene armonizate din domeniul compatibilitatii electromagnetice": Ministerul Economiei si Ministerul Comunicatiilor si Comertului Tehnologiei Informatiei, 2004.
- [20] F. Smadu, "Sistem pentru analiza calitatii energiei electrice in sistemele de electroalimentare trifazate", in *Conferinta nationala de instrumentatie virtuala, editia a II-a Bucuresti*, pp. 29 - 32.
- [21] C. Stănescu, F. Vatră, A. Poida, P. Postolache, "Power Quality in Romanian Electricity Market", in *9th International Conference. Electrical Power Quality and Utilisation* Barcelona.
- [22] A. Hermina, G. Carmen, N. Golovanov, E. Luminița, "Probleme actuale privind monitorizarea calitatii energiei electrice in Romania".
- [23] H. Albert, N. Golovanov, C. Golovanov, V. Rășcanu, L. Elefterescu, "Monitorizarea calitatii energiei electrice".
- [24] A. Ignea, M. Lascu, **C. Dughir**, C. Iftode, *Aspecte ale compatibilitatii electromagnetice in medicina*. Timisoara: Waldpress, 2005.
- [25] J. F. G. Cobbe, J. F. G. Cobbe, "Classification Methodologies for Power Quality", February 2006 2006.
- [26] M. Azam, F. Tu, K. Pattipati, R. Karanam, "A Dependency Model Based Approach for Identifying and Evaluating Power Quality Problems", *IEEE Transactions on Power Delivery*, vol. 19, July 2004.
- [27] G. Bucci, E. Fiorucci, C. Landi, "Digital Measurement Station for Power Quality Analysis in Distributed Environments", *IEEE Transactions on Instrumentation and Measurement* vol. 52, February 2003.
- [28] M. H. J. Bollen, E. Styvaktakis, I. Y.-H. Gu, "Categorization and Analysis of Power System Transients", *IEEE Transactions on Power Delivery*, vol. 20, July 2005.
- [29] J.-J. G. de-la-Rosa, A. M. Munoz, "Categorization of Power Quality Transients using Higher-Order Statistics and Competitive Layers-based Neural Networks", in *CIMSA 2008 - IEEE International Conference on Computational Intelligence for Measurement Systems and Applications* Istanbul, Turkey.
- [30] M. H. H. M. Erhan Balci, "Effects of Source Voltage Harmonics on Power Factor Compensation in AC Chopper Circuits", *Electrical Power Quality and Utilisation, Journal*, vol. XIV, p. 7, 2008.
- [31] C. Dinu, I. I. Odor, "Aspecte privind asigurarea trasabilitatii la si a instrumentatiei utilizata pentru masurarea parametrilor de calitate a tensiunii in retelele publice de distributie a energiei electrice", *Metrologie*, vol. 1, 2010.
- [32] L. Zhang, M. H. J. Bollen, "A method for characterisation of three-phase unbalanced dips (sags) from recorded voltage waveshapes", in *The 21st*

- International Telecommunication Energy Conference INTELEC '99*, Copenhagen.
- [33] D. Chapman, *Power Quality Application Guide - Harmonics Causes and Effects*, 2001.
- [34] Z. J. Paracha, A. Kalam, R. Ali, "A Novel Approach of Harmonic Analysis in Power Distribution Networks using Artificial Intelligence", in *International Conference on Information and Communication Technologies, 2009, ICICT '09*, Karachi pp. 157 - 160
- [35] A. Ignea, *Compatibilitate Electromagnetică: Editura de Vest*, 2007.
- [36] M. H. J. Bollen, "Characterisation of voltage sags experienced by three-phase adjustable-speed drives", *IEEE Transactions on Power Delivery*, pp. 1666 - 1671 Oct 1997
- [37] M. H. J. Bollen, G. Yalcinkaya, J. Pellis, M. R. Qader, "A voltage sag study in a large industrial distribution system", in *IEEE Industry Applications Conference*, San Diego, CA, USA pp. 2372 - 2377.
- [38] A. Sundaram, H. Mehta, V. Tahiliani, "Custom Power-EPRI's Response to Power Quality Issues", Electric Power Research Institute.
- [39] P. Teikari, "Flicker Metrology", in *Power Systems Engineering, Special Assignment* Espoo, Finland.
- [40] X. X. Yang, M. Kratz, "Power System Flicker Analysis by RMS Voltage Values and Numeric Flicker Meter Emulation", *IEEE Transactions on Power Delivery*, vol. 24, pp. 1301-1318, July 2009.
- [41] P. G. V. Axelberg, M. H. J. Bollen, I. Y.-H. Gu, "Trace of Flicker Sources by Using the Quantity of Flicker Power", *IEEE Transactions on Power Delivery*, vol. 23, pp. 465-471, January 2008.
- [42] A. Ignea, "Compatibilitate Electromagnetica".
- [43] "IEEE Standard 1159 -1995, Recommended Practice for Monitoring Electric Power Quality".
- [44] "IEC 77A/301/CD - Electromagnetic Compatibility (EMC) -Part 4-30: Testing and measurement techniques - Power Quality Measurement Methods".
- [45] M. Radulescu, D. C. Rucinski, "Some contributions for developing dedicated EMC specifications for components of telecommunication networks".
- [46] IEC, "IEC Guide 107 Electromagnetic compatibility - Guide to the drafting of electromagnetic compatibility publications", 2009.
- [47] F. Martzloff, "A New IEC Standard on the Measurement of Power Quality Parameters", in *Proceedings, EMC Europe 2000 Symposium*, Bruges, 2000.
- [48] T. S. Key, "Diagnosing Power-Quality Related Computer Problems", *IEEE Transactions*, vol. LA-15, July 1979.
- [49] J. M. Clemmensen, R. J. Ferraro, "The Emerging Problem of Electric Power Quality", in *Public Utilities Fortnightly*, Arlington VA.
- [50] G. O. Sullivan, "Lead article", in *PCIM Magazine*.
- [51] IEEE, "IEEE Recommended Practice for Powering and Grounding Electronic Equipment", in *Std 1 100-1992*
- [52] IEEE, "IEEE Recommended Practice for Monitoring Electric Power Quality", in *Std 1 159- 1995*
- [53] IEEE, "IEEE Guide for Service to Equipment Sensitive to Momentary Voltage Disturbances", in *Std 1250-1995*
- [54] ***, *Eskom Handbook on Quality of Supply*: Eskom QOS Group.
- [55] IEEE, "IEEE Recommended Practice for Powering and Grounding Electronic Equipment", in *Std 1100-1999*.

- [56] T. Hoevenaars, K. LeDoux, Matt Colosino, "Interpreting IEEE Std 519 and Meeting its Harmonic Limits in VFD Applications".
- [57] IEEE, "IEEE Standard 519: Recommended Practices and Requirements for Harmonic Control in Electrical Power Systems", 1992.
- [58] IEEE, "IEEE Std 1159: IEEE Recommended Practice for Monitoring Electric Power Quality", 1995.
- [59] IEC, "IEC 61000-4-30 Standard", 2007.
- [60] H. Markiewicz, A. Klajn, *Standard EN 50160 - Voltage Characteristics in Public Distribution Systems*. Wrocław: European Copper Institute, 2004.
- [61] G. Asch, e. collaborateurs, *Les capteurs en instrumentation industrielle*. France: Imprimeie Gauthier-Villards, 1991.
- [62] M. Ciugudean, *Circuite integrate liniare-Aplicații*. Timișoara: Editura Facla, 1986.
- [63] D. Belega, **C. Dughir**, R. Pazsitka, *Circuite Integrate Analogice - Experimentare și Simulare*: Editura Politehnica, 2008.
- [64] **C. Dughir**, G. Gășpăresc, "Preconditioning Circuit for Electrical Power System Disturbances Measurement", *ETC 2006*.
- [65] A. Ignea, **C. Dughir**, G. Gășpăresc, "Calibration Method For Nonlinear Voltage Divider Used as a Preconditioning Circuit in the Electrical Power Network Disturbances Monitoring System", *ACERO 2006*.
- [66] **C. Dughir**, A. Ignea, G. Gășpăresc, A. Vârtosu, "Nonlinear voltage divider", *Proceedings of the IMEKO Conference, Iași, 2007*.
- [67] **C. Dughir**, G. Gășpăresc, "Three Channels Analysis System for Electrical Power System Disturbances Measurement", *Buletinul Institutului Politehnic Iași*, 2006.
- [68] I. Naforniță, A. Câmpeanu, A. Isar, *Semnale Circuite și Sisteme*: Editura "Politehnica" Timișoara, 1995.
- [69] E. Creangă, I. Munteanu, A. Bratcu, M. Culea, *Semnale Circuite și Sisteme*: Editura Academica, Galați, 2001.
- [70] T. Jurca, D. Stoiciu, *Aparate electronice de masurat*: Universitatea Tehnica, Timișoara, 1993.
- [71] N. Instruments, "NI 625x Specifications".
- [72] F. Martzloff, "High-Speed Transient Recording System", *General Electric TIS Report 68-C-019*, January 1968.
- [73] P. Lu, S. Su, G. Liu, H. Rong, L. Zheng, "A new power quality detection device based on embedded technique", in *Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, DRPT 2008*, Nanjing pp. 1635 - 1640.
- [74] G. H. Yang, B. Y. Wen, "A Device for Power Quality Monitoring Based on ARM and DSP", in *IEEE Conference on Industrial Electronics and Applications*, Singapore
- [75] J. Mindykowski, T. Tarasiuk, "DSP-Based Instrument for Power Quality Monitoring on Ships", *XIX IMEKO World Congress Fundamental and Applied Metrology*, September 6–11, 2009, Lisbon, Portugal.
- [76] C. Gherasim, J. V. d. Keybus, J. Driesen, R. Belmans, "DSP Implementation of Power Measurements According to the IEEE Trial-Use Standard 1459", *IEEE Transactions on Instrumentation and Measurement*, vol. 53, p. 7, August 2004.
- [77] J. Gou, B. Xie, P. Enjeti, "A DSP based real time power quality measurement system", in *Seventh Annual Applied Power Electronics Conference and Exposition, APEC '92*, Boston, MA, pp. 299 - 302

- [78] A. Lakshmikanth, M. M. Morcos, "A Power Quality Monitoring System: A Case Study in DSP-Based Solutions for Power Electronics", *IEEE Transactions on Instrumentation and Measurement*, vol. 50, pp. 724 - 731, Jun 2001.
- [79] T. S. Sidhu, "Accurate Measurement of Power System Frequency Using a Digital Signal Processing Tehnique", *IEEE Transactions on Instrumentation and Measurement*, vol. 48, p. 7, February 1999.
- [80] L. Stankovic, "A Method for Time-Frequency Analysis", *IEEE Transactions on Signal Processing*, vol. 42, pp. 225 - 229 Jan 1994.
- [81] W. G. Morsi, M. E. El-Hawary, "A New Fuzzy-Wavelet Based Representative Quality Power Factor for Stationary and Nonstationary Power Quality Disturbances", in *IEEE Power & Energy Society General Meeting, PES '09*, Calgary.
- [82] C. N. Bhende, S. Mishra, "An Integrated Approach of Wavelet-Rough Set Technique for Classification of Power Quality Disturbances", in *13th International Conference on Harmonics and Quality of Power*, Wollongong.
- [83] B. Perunicic, M. Mallini, Z. Wang, Y. Liu, "Power quality disturbance detection and classification using wavelets and artificial neural networks", in *Proceedings of 8th International Conference on Harmonics And Quality of Power*, Athens, Greece, pp. 77 - 82.
- [84] W. G. Morsi, M. E. El-Hawary, "A new fuzzy-based representative quality power factor for unbalanced three-phase systems with nonsinusoidal situations", in *IEEE Power & Energy Society General Meeting, PES '09*, Calgary, AB
- [85] M. Artioli, G. Pasini, L. Peretto, R. Sasdelli, F. Fillipetti, "Low-Cost DSP-Based Equipment for the Real-Time Detection of Transients in Power Systems", *IEEE Transactions on Instrumentation and Measurement*, vol. 53, p. 7, August 2004.
- [86] **C. Dughir**, T. Jurca, "Detectarea apariției perturbațiilor în rețeaua de distribuție a energiei electrice", in *Energetica*, 2004.
- [87] **C. Dughir**, "Detecting Symmetrical Disturbances in the Electrical Power Systems", *Doctor ETc 2004*.
- [88] A. Ignea, T. Jurca, A. D. Sabata, **C. Dughir**, A. Mihaiuti, *Aspecte privind monitorizarea perturbațiilor dintr-un site*. Timisoara: Waldpress, 2006.
- [89] "<http://focus.ti.com/lit/ds/symlink/iso122.pdf>".
- [90] "<http://focus.ti.com/lit/ds/symlink/dch010515s.pdf>".
- [91] Microsoft, *Microsoft Visual C++ .NET Standard*: Microsoft Press, 2001.
- [92] D. Gosselin, *Microsoft Visual C++ .NET Course Technology*, 2002.
- [93] "[http://msdn.microsoft.com/en-us/library/d06h2x6e\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/d06h2x6e(VS.71).aspx)".
- [94] J. He, "Microsoft Foundation Class (MFC) Quick Reference".
- [95] Wikipedia, "http://ro.wikipedia.org/wiki/Microsoft_Foundation_Classes".
- [96] Microsoft, "[http://msdn.microsoft.com/en-us/library/ws8s10w4\(v=VS.71\).aspx](http://msdn.microsoft.com/en-us/library/ws8s10w4(v=VS.71).aspx)".
- [97] Microsoft, "[http://msdn.microsoft.com/en-us/library/fe1cf721\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/fe1cf721(VS.80).aspx)".
- [98] C. Hughes, T. Hughes, *Object-Oriented Multithreading Using C++*: Wiley, 1997.
- [99] A. Williams, *C++ Concurrency in Action: Practical Multithreading*: Manning Publications, 2010.
- [100] F. Martzloff, "Transient Control Level Test Generators", 1977.
- [101] F. Martzloff, G. Pellegrini, "Real, Realistic Ring Waves for Surge Testing", in *Proceedings, 9th International Symposium on Electromagnetic Compatibility*, Zürich, 1991.

- [102] G. Găspăresc, **C. Dughir**, "Electromagnetic Disturbances Generator with Graphical User Interface Bild in MATLAB", *Buletinul Institutului Politehnic Iași*, 2006.
- [103] G. Găspăresc, **C. Dughir**, A. Ignea, "Studierea Perturbațiilor Electromagnetice Folosind un Generator de Perturbații cu GUI", *ACERO*, 2006.
- [104] G. Găspăresc, **C. Dughir**, L. Stoica, "Signal Generator for Usual Waveforms and Electromagnetic Disturbances with Graphical User Interface", *Buletinul Institutului Politehnic Petroșani*, 2006.
- [105] **C. Dughir**, "Visual C++ Complex Mathematical Signal Generator", *Proceedings of the IMEKO Conference, Florence, Italy, 2008*.
- [106] *The Electric Power Engineering Handbook*: Boca Raton: CRC Press LLC, 2001.
- [107] "<http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/functions/>".
- [108] "<http://en.wikipedia.org/wiki/Xampp>".
- [109] "<http://en.wikipedia.org/wiki/Html>".
- [110] "<http://en.wikipedia.org/wiki/Php>".
- [111] E. L. Stamate, *Crearea și publicarea unui site web. Limbajul HTML și elemente de web design*: Paralela 45, 2005.
- [112] **C. Dughir**, V. Groza, A. Vârtosu, G. Proștean, "Electrical Power Distributions Network Quality Monitoring using MSP430", *IEEE Explore*, 2009.
- [113] D. Belega, **C. Dughir**, "Acquisition signals from electromagnetic field-meters using digital multimeters with event logging mode", in *XIX IMEKO World Congress, Fundamental and Applied Metrology*, Lisbon, Portugal.
- [114] D. Belega, **C. Dughir**, A. Ignea, "A system for electric field measurement", in *2nd IMEKO TC19 Conference on Environmental Measurements*, Budapest, Hungary.
- [115] E. Nica, D. Stoiciu, P. Brad, **C. Dughir**, "Sistem de achiziție și prelucrare de date pentru studiul unei acționări cu motor asincron", in *„Energetica” Oradea 1999*.
- [116] E. Nica, D. Stoiciu, P. Brad, **C. Dughir**, "Modeling and simulation study of induction motors in unstable states", in *IASTED Pittsburgh, USA, 2000*, pp. p. 129-135.
- [117] E. Nica, D. Stoiciu, P. Brad, **C. Dughir**, "Computational procedure to analyze the parameters change on rotor speed estimation of the induction motor", in *Proceedings of the International Conference on Electrical Machines*, Espoo, Finland, pp. 208-212.

ANEXA 1. PROGRAME REPREZENTATIVE ELABORATE ÎN PSPICE

Program PSpice pentru simularea caracteristicii de transfer și a comportamentului la aplicarea unui semnal biexponențial al transformatorului funcțional prezentat în figura 3.2.

```

*version 9.2 2183373009
u 566
U? 12
V? 13
R? 16
D? 11
? 8
@libraries
@analysis
.AC 0 1 0
+0 101
+1 10
+2 1.00K
.DC 0 0 0 1 1
+ 0 0 Vinput
+ 0 4 -10000
+ 0 5 10000
+ 0 6 10
.TRAN 1 0 1 0
+0 1us
+1 3ms
+3 10us
+4 1kHz
+6 100
.MC 0 1 2 1 0 1 0 1 1 0
+1 V(out)
.STEP 0 0 1
+ 4 10
+ 5 50
+ 6 10
.TEMP 0 20
.TF 0 V(Out) V6
.OP 0
.SENS 0 v(out)
@targets
@attributes
@translators
a 0 u 13 0 0 0 hln 100
PCBOARDS=PCB
a 0 u 13 0 0 0 hln 100
PSPICE=PSPICE
a 0 u 13 0 0 0 hln 100
XILINX=XILINX
@setup
unconnectedPins 0
connectViaLabel 0
connectViaLocalLabels 0
NoStim4ExtIFPortsWarnings 1
AutoGenStim4ExtIFPorts 1
@index
pageloc 1 0 9573
@status
n 0 107:01:08:00:04:28;1170885868
e
s
2832
107:01:08:00:04:30;1170885870 e
*page 1 0 970 720 iA
@ports
port 18 GND_ANALOG 520 680 h
@parts
part 16 D1N4148 520 370 u
a 0 sp 11 0 15 25 hln 100
PART=D1N4148
a 0 s 0:13 0 0 0 hln 100
PKGTYPE=DO-35
a 0 s 0:13 0 0 0 hln 100 GATE=
a 0 a 0:13 0 0 0 hln 100 PKGREF=D2
a 0 ap 9 0 15 0 hln 100 REFDES=D2
part 15 D1N4148 520 410 u
a 0 sp 11 0 15 25 hln 100
PART=D1N4148
a 0 s 0:13 0 0 0 hln 100
PKGTYPE=DO-35
a 0 s 0:13 0 0 0 hln 100 GATE=
a 0 a 0:13 0 0 0 hln 100 PKGREF=D1

```

```
a 0 ap 9 0 15 0 hln 100 REFDES=D1
part 4 VDC 540 640 h
a 0 sp 0 0 22 37 hln 100 PART=VDC
a 0 a 0:13 0 0 0 hln 100 PKGREF=V2
a 1 ap 9 0 24 7 hcn 100 REFDES=V2
a 1 u 13 0 -11 18 hcn 100 DC=18V
part 3 VDC 490 640 h
a 0 sp 0 0 22 37 hln 100 PART=VDC
a 0 a 0:13 0 0 0 hln 100 PKGREF=V1
a 1 ap 9 0 24 7 hcn 100 REFDES=V1
a 1 u 13 0 -11 18 hcn 100 DC=-18V
part 193 D1N4148 490 260 h
a 0 sp 11 0 15 25 hln 100
PART=D1N4148
a 0 s 0:13 0 0 0 hln 100
PKGTYPE=DO-35
a 0 s 0:13 0 0 0 hln 100 GATE=
a 0 a 0:13 0 0 0 hln 100 PKGREF=D5
a 0 ap 9 0 15 0 hln 100 REFDES=D5
part 195 D1N4148 490 220 h
a 0 sp 11 0 15 25 hln 100
PART=D1N4148
a 0 s 0:13 0 0 0 hln 100
PKGTYPE=DO-35
a 0 s 0:13 0 0 0 hln 100 GATE=
a 0 a 0:13 0 0 0 hln 100 PKGREF=D6
a 0 ap 9 0 15 0 hln 100 REFDES=D6
part 399 r 600 430 h
a 0 sp 0 0 10 10 hlb 100 PART=r
a 0 s 0:13 0 0 0 hln 100
PKGTYPE=RC05
a 0 s 0:13 0 0 0 hln 100 GATE=
a 0 a 0:13 0 0 0 hln 100
PKGREF=R15
a 0 ap 9 0 15 0 hln 100 REFDES=R15
a 0 u 13 0 15 25 hln 100 VALUE=10k
part 9 r 690 410 h
a 0 sp 0 0 10 10 hlb 100 PART=r
a 0 s 0:13 0 0 0 hln 100
PKGTYPE=RC05
a 0 s 0:13 0 0 0 hln 100 GATE=
a 0 a 0:13 0 0 0 hln 100 PKGREF=R3
a 0 ap 9 0 15 0 hln 100 REFDES=R3
a 0 u 13 0 15 25 hln 100 VALUE=2.5k
part 199 r 690 260 h
a 0 sp 0 0 10 10 hlb 100 PART=r
a 0 s 0:13 0 0 0 hln 100
PKGTYPE=RC05
a 0 s 0:13 0 0 0 hln 100 GATE=
a 0 a 0:13 0 0 0 hln 100
PKGREF=R12

a 0 ap 9 0 15 0 hln 100 REFDES=R12
a 0 u 13 0 15 25 hln 100 VALUE=2.5k
part 10 r 690 370 h
a 0 sp 0 0 10 10 hlb 100 PART=r
a 0 s 0:13 0 0 0 hln 100
PKGTYPE=RC05
a 0 s 0:13 0 0 0 hln 100 GATE=
a 0 a 0:13 0 0 0 hln 100 PKGREF=R4
a 0 ap 9 0 15 0 hln 100 REFDES=R4
a 0 u 13 0 15 25 hln 100 VALUE=222
part 198 r 690 220 h
a 0 sp 0 0 10 10 hlb 100 PART=r
a 0 s 0:13 0 0 0 hln 100
PKGTYPE=RC05
a 0 s 0:13 0 0 0 hln 100 GATE=
a 0 a 0:13 0 0 0 hln 100
PKGREF=R11
a 0 ap 9 0 15 0 hln 100 REFDES=R11
a 0 u 13 0 15 25 hln 100 VALUE=222
part 13 r 650 170 v
a 0 sp 0 0 10 10 hlb 100 PART=r
a 0 s 0:13 0 0 0 hln 100
PKGTYPE=RC05
a 0 s 0:13 0 0 0 hln 100 GATE=
a 0 a 0:13 0 0 0 hln 100 PKGREF=R7
a 0 ap 9 0 15 0 hln 100 REFDES=R7
a 0 u 13 0 -11 39 hln 100 VALUE=450
part 196 r 600 170 v
a 0 sp 0 0 10 10 hlb 100 PART=r
a 0 s 0:13 0 0 0 hln 100
PKGTYPE=RC05
a 0 s 0:13 0 0 0 hln 100 GATE=
a 0 a 0:13 0 0 0 hln 100 PKGREF=R9
a 0 ap 9 0 15 0 hln 100 REFDES=R9
a 0 u 13 0 -13 39 hln 100 VALUE=450
part 355 VDC 780 100 u
a 0 sp 0 0 22 37 hln 100 PART=VDC
a 0 a 0:13 0 0 0 hln 100 PKGREF=V7
a 1 ap 9 0 24 7 hcn 100 REFDES=V7
a 1 u 13 0 -11 18 hcn 100 DC=15V
part 6 VDC 400 200 h
a 0 sp 0 0 22 37 hln 100 PART=VDC
a 0 a 0:13 0 0 0 hln 100 PKGREF=V4
a 1 ap 9 0 24 7 hcn 100 REFDES=V4
a 1 u 13 0 -11 18 hcn 100 DC=15V
part 8 r 420 560 h
a 0 sp 0 0 10 10 hlb 100 PART=r
a 0 s 0:13 0 0 0 hln 100
PKGTYPE=RC05
a 0 s 0:13 0 0 0 hln 100 GATE=
a 0 a 0:13 0 0 0 hln 100 PKGREF=R2
```



```
s 520 220 600 220 459
a 0 up 33 0 645 219 hct 100 V=
s 600 220 690 220 470
s 600 220 600 170 330
w 311
a 0 up 0:33 0 0 0 hln 100 V=
s 520 260 580 260 464
a 0 up 33 0 635 259 hct 100 V=
s 580 260 690 260 476
s 580 90 580 260 339
w 357
a 0 up 0:33 0 0 0 hln 100 V=
s 650 20 620 20 319
s 740 20 650 20 223
s 650 130 650 20 317
a 0 up 33 0 652 75 hlt 100 V=
s 620 50 620 20 314
s 740 60 740 20 410
s 780 60 740 60 416
w 411
a 0 up 0:33 0 0 0 hln 100 V=
s 780 510 600 510 366
s 600 510 600 590 368
s 600 590 570 590 370
s 570 590 570 680 372
s 570 680 540 680 374
s 540 680 520 680 19
s 520 680 490 680 21
s 420 680 490 680 136
s 420 560 420 680 77
s 780 100 780 510 364
s 290 680 420 680 79
s 290 240 290 510 86
a 0 up 33 0 292 375 hlt 100 V=
s 290 240 400 240 519
s 290 510 290 680 556
s 290 510 390 510 554
w 558
a 0 up 0:33 0 0 0 hln 100 V=
s 390 470 410 470 557
a 0 up 33 0 420 469 hct 100 V=
s 450 470 460 470 559
s 410 470 450 470 563
w 29
a 0 up 0:33 0 0 0 hln 100 V=
s 540 640 540 460 28
a 0 up 33 0 542 550 hlt 100 V=
s 540 460 560 460 30
w 23
a 0 up 0:33 0 0 0 hln 100 V=
s 490 630 490 640 26

s 560 630 490 630 24
s 560 520 560 630 22
a 0 up 33 0 562 575 hlt 100 V=
w 70
a 0 up 0:33 0 0 0 hln 100 V=
s 490 260 490 220 194
s 490 470 490 430 386
a 0 up 33 0 492 440 hlt 100 V=
s 490 410 490 370 395
s 490 430 490 410 402
s 490 430 600 430 428
s 520 510 490 510 442
s 490 510 490 470 444
s 490 370 490 260 485
w 158
a 0 sr 0:3 0 670 488 hcn 100
LABEL=Out
a 0 up 0:33 0 0 0 hln 100 V=
s 600 490 700 490 499
a 0 sr 3 0 670 488 hcn 100
LABEL=Out
s 740 490 740 430 200
a 0 up 33 0 742 450 hlt 100 V=
s 740 260 740 220 273
s 740 260 730 260 239
s 740 220 730 220 240
s 740 370 740 260 276
s 740 370 730 370 249
s 740 410 740 370 282
s 740 410 730 410 254
s 740 430 740 410 405
s 640 430 740 430 403
s 700 490 740 490 504
w 437
a 0 up 0:33 0 0 0 hln 100 V=
s 460 560 510 560 436
s 510 560 510 470 438
a 0 up 33 0 512 515 hlt 100 V=
s 510 470 520 470 440
@junction
j 650 370
+ w 397
+ w 397
j 650 170
+ p 13 1
+ w 397
j 730 260
+ p 199 2
+ w 158
j 740 260
+ w 158
```

+ w 158	+ w 311
j 740 370	j 580 50
+ w 158	+ p 197 2
+ w 158	+ w 214
j 740 410	j 580 20
+ w 158	+ w 214
+ w 158	+ w 214
j 740 430	j 690 220
+ w 158	+ p 198 1
+ w 158	+ w 307
j 540 640	j 600 220
+ p 4 +	+ w 307
+ w 29	+ w 307
j 690 410	j 400 200
+ p 9 1	+ p 6 +
+ w 398	+ w 214
j 730 410	j 700 490
+ p 9 2	+ p 498 pin1
+ w 158	+ w 158
j 620 90	j 640 430
+ p 14 1	+ p 399 2
+ w 398	+ w 158
j 620 410	j 520 410
+ w 398	+ p 15 1
+ w 398	+ w 398
j 650 20	j 520 370
+ w 357	+ p 16 1
+ w 357	+ w 397
j 650 130	j 520 260
+ p 13 2	+ p 193 2
+ w 357	+ w 311
j 620 50	j 520 220
+ p 14 2	+ p 195 2
+ w 357	+ w 307
j 690 370	j 490 220
+ p 10 1	+ p 195 1
+ w 397	+ w 70
j 730 370	j 490 260
+ p 10 2	+ p 193 1
+ w 158	+ w 70
j 730 220	j 490 370
+ p 198 2	+ p 16 2
+ w 158	+ w 70
j 690 260	j 490 410
+ p 199 1	+ p 15 2
+ w 311	+ w 70
j 580 90	j 490 430
+ p 197 1	+ w 70
+ w 311	+ w 70
j 580 260	j 600 430
+ w 311	+ p 399 1

```

+ w 70
j 490 470
+ p 7 2
+ w 70
j 460 560
+ p 8 2
+ w 437
j 780 60
+ p 355 -
+ w 357
j 490 640
+ p 3 +
+ w 23
j 520 680
+ s 18
+ w 411
j 540 680
+ p 4 -
+ w 411
j 490 680
+ p 3 -
+ w 411
j 420 560
+ p 8 1
+ w 411
j 780 100
+ p 355 +
+ w 411
j 420 680
+ w 411
+ w 411
j 400 240
+ p 6 -
+ w 411
j 290 510
+ w 411
+ w 411
j 450 470
+ p 7 1
+ w 558
j 600 170
+ p 196 1
+ w 307
j 600 130
+ p 196 2
+ w 214
j 390 470
+ p 561 -
+ w 558
j 390 510
+ p 561 +
+ w 411
j 410 470
+ p 562 pin1
+ w 558
j 560 460
+ p 565 V+
+ w 29
j 560 520
+ p 565 V-
+ w 23
j 520 510
+ p 565 -
+ w 70
j 600 490
+ p 565 OUT
+ w 158
j 520 470
+ p 565 +
+ w 437
@attributes
a 0 s 0:13 0 0 0 hln 100 PAGETITLE=
a 0 s 0:13 0 0 0 hln 100 PAGENO=1
a 0 s 0:13 0 0 0 hln 100 PAGESIZE=A
a 0 s 0:13 0 0 0 hln 100
PAGECOUNT=1
@graphics

```


ANEXA 2. PROGRAME REPREZENTATIVE ELABORATE ÎN MATLAB

Anexa 2.1 Program Matlab pentru verificarea algoritmului de calibrare a sistemului de condiționare a semnalelor cu transformator funcțional

```
% Program pentru analiza semnalelor provenite de la transformatorul funcțional
% Author: Ciprian Dughir
clear;
% Citirea datelor din fișierul .XLS (datele au fost obținute cu ajutorul programului
PSpice prezentat anterior în anexe)
bIsCalibEnable = 0;
if (bIsCalibEnable == 1)
    [pArr, pTxt] = xlsread('Triangle10kV_2.xls',1, 'A1:D2000');
else
    [pArr, pTxt] = xlsread('10kVBiexp_1kSamples.xls',1, 'A1:D2000');
end
[pCalibArr, pTxt2] = xlsread('calib3.xls',1, 'B2:E500');
pCalibOutput = pCalibArr(:,1);
pCalibInput = pCalibArr(:,2);
pCalibTimeLine = pCalibArr(:,3) * 1000;
pCalibration = pCalibArr(:,4);
pTimeLine=pArr(:,1) * 1000;
pOut=pArr(:,2);
pIn=pArr(:,4);
nSignalLength = length(pTimeLine);
nSamplesToShow = nSignalLength
% Decompresia valorilor semnalului de ieșire
nLastCalibArrPos = 0;
fCalibValue = 0;
pRecoveredOutputSignal = 1:nSignalLength;
for i=1:nSignalLength
    if (pOut(i)>=0)
        if (pOut(i)<=5)
            pRecoveredOutputSignal(i)=100*pOut(i);
        elseif pOut(i)>5
            if pOut(i)<8.1
                pRecoveredOutputSignal(i)=516*pOut(i)-2080;
            else
                pRecoveredOutputSignal(i)=4157*pOut(i)-31579;
            end
        end
    end
end
```

```

else
    if (pOut(i)>-5)
        pRecoveredOutputSignal(i)=100*pOut(i);
    else
        if pOut(i)<=-5
            if pOut(i)>=-8.1
                pRecoveredOutputSignal(i)=516*pOut(i)+2080;
            else
                pRecoveredOutputSignal(i)=4157*pOut(i)+31579;
            end
        end
    end
end
end
for j=1:271
    if (pRecoveredOutputSignal(i) >= pCalibOutput(j))
        fCalibVal = pCalibration(j);
    end
end
if (pRecoveredOutputSignal(i) < 490)
    fCalibVal = 0;
end
pRecoveredOutputSignalCalib(i) = pRecoveredOutputSignal(i) - fCalibVal;
end
% Generarea fişierului de calibrare
if (bIsCalibEnable == 1)
    xlswrite('calib3.xls',pRecoveredOutputSignal,'Sheet1','B2');
    xlswrite('calib3.xls',pTimeLine,'Sheet1','D2');
    xlswrite('calib3.xls',pIn,'Sheet1','C2');
end
% Calcularea parametrilor semnalului de ieşire decomprimat
nIndexInput10 = 0;
nIndexInput50 = 0;
nIndexInput90 = 0;
nIndexOutput10 = 0;
nIndexOutput50 = 0;
nIndexOutput90 = 0;
[maxInputValue,maxInputValueIndex]=max(pIn);
[maxOutputValue,maxOutputValueIndex]=max(pRecoveredOutputSignal);
maxInputValue10=maxInputValue/10;
maxInputValue50=maxInputValue/2;
maxInputValue90=0.9*maxInputValue;
maxOutputValue10=maxOutputValue/10;
maxOutputValue50=maxOutputValue/2;
maxOutputValue90=0.9*maxOutputValue;
for i=1:maxInputValueIndex
    if (pIn(i) < maxInputValue10)
        nIndexInput10 = i;
    end
    if (pIn(i) < maxInputValue50)
        nIndexInput50Start = i;
    end
end

```

```

    if (pIn(i) < maxInputValue90)
        nIndexInput90 = i;
    end
end
for i=maxInputValueIndex:nSignalLength
    if (pIn(i) > maxInputValue50)
        nIndexInput50End = i;
    end
end
for i=1:maxOutputValueIndex
    if (pRecoveredOutputSignal(i) < maxOutputValue10)
        nIndexOutput10 = i;
    end
    if (pRecoveredOutputSignal(i) < maxOutputValue50)
        nIndexOutput50Start = i;
    end
    if (pRecoveredOutputSignal(i) < maxOutputValue90)
        nIndexOutput90 = i;
    end
end
for i=maxOutputValueIndex:nSignalLength
    if (pRecoveredOutputSignal(i) > maxOutputValue50)
        nIndexOutput50End = i;
    end
end
fInputSignalRiseTime = pTimeLine(nIndexInput90)-pTimeLine(nIndexInput10);
fOutputSignalRiseTime = pTimeLine(nIndexOutput90)-pTimeLine(nIndexOutput10);
fInputSignalTime50 = pTimeLine(nIndexInput50End)-
    pTimeLine(nIndexInput50Start);
fOutputSignalTime50 = pTimeLine(nIndexOutput50End)-
    pTimeLine(nIndexOutput50Start);

fRelativeErrorRecoveredSignalRiseTime = ((fOutputSignalRiseTime-
    fInputSignalRiseTime)/fInputSignalRiseTime)*100;
fRelativeErrorRecoveredSignal50Ampl = ((fOutputSignalTime50-
    fInputSignalTime50)/fInputSignalTime50)*100;
% Calculul timpului de creștere și a timpului la jumătatea amplitudinii a semnalului
% decomprimat calibrat
nIndexInput10 = 0;
nIndexInput50 = 0;
nIndexInput90 = 0;
nIndexOutput10 = 0;
nIndexOutput50 = 0;
nIndexOutput90 = 0;
[maxInputValue,maxInputValueIndex]=max(pIn);
[maxOutputCalibValue,maxOutputValueIndex]=max(pRecoveredOutputSignalCalib);
maxInputValue10=maxInputValue/10;
maxInputValue50=maxInputValue/2;
maxInputValue90=0.9*maxInputValue;
maxOutputValue10=maxOutputCalibValue/10;
maxOutputValue50=maxOutputCalibValue/2;

```

```

maxOutputValue90=0.9*maxOutputCalibValue;
for i=1:maxInputValueIndex
    if (pIn(i) < maxInputValue10)
        nIndexInput10 = i;
    end
    if (pIn(i) < maxInputValue50)
        nIndexInput50Start = i;
    end
    if (pIn(i) < maxInputValue90)
        nIndexInput90 = i;
    end
end
for i=maxInputValueIndex:nSignalLength
    if (pIn(i) > maxInputValue50)
        nIndexInput50End = i;
    end
end
for i=1:maxOutputValueIndex
    if (pRecoveredOutputSignalCalib(i) < maxOutputValue10)
        nIndexOutput10 = i;
    end
    if (pRecoveredOutputSignalCalib(i) < maxOutputValue50)
        nIndexOutput50Start = i;
    end
    if (pRecoveredOutputSignalCalib(i) < maxOutputValue90)
        nIndexOutput90 = i;
    end
end
for i=maxOutputValueIndex:nSignalLength
    if (pRecoveredOutputSignalCalib(i) > maxOutputValue50)
        nIndexOutput50End = i;
    end
end
fInputSignalRiseTime = pTimeLine(nIndexInput90)-pTimeLine(nIndexInput10);
fOutputSignalRiseTime = pTimeLine(nIndexOutput90)-pTimeLine(nIndexOutput10);
fInputSignalTime50 = pTimeLine(nIndexInput50End)-
    pTimeLine(nIndexInput50Start);
fOutputSignalTime50 = pTimeLine(nIndexOutput50End)-
    pTimeLine(nIndexOutput50Start);
fRelativeErrorRecoveredSignalCalibRiseTime = ((fOutputSignalRiseTime-
    fInputSignalRiseTime)/fInputSignalRiseTime)*100;
fRelativeErrorRecoveredSignalCalib50Ampl = ((fOutputSignalTime50-
    fInputSignalTime50)/fInputSignalTime50)*100;
% Calculul transformatei Fourier a semnalului de la ieşirea sistemului
nFFTLen=2^7;
pInputSignalPeriodic = repmat(pIn,5,1);
pRecoveredOutputSignalPeriodic = repmat(pRecoveredOutputSignal,1,5);
z=fft(pInputSignalPeriodic);
z1=abs(z);
z2=max(z1);
pInputSignalFFT=(z1/z2);

```

```

pInputSignalFFT=pInputSignalFFT(1:nFFTlen);
y=fft(pRecoveredOutputSignalPeriodic);
y1=abs(y);
y2=max(y1);
pRecoveredOutputSignalFFT=(y1/y2);
pRecoveredOutputSignalFFT=pRecoveredOutputSignalFFT(1:nFFTlen);
nFFFTimeLine=1:nFFTlen;
figure(2);
subplot(2,2,2);
plot(nFFFTimeLine, 1.5+pInputSignalFFT,'r', nFFFTimeLine,
      pRecoveredOutputSignalFFT,'b');
title('FFT semnal intrare (r) și FFT semnal ieșire (b)');
grid on;
subplot(2,2,4);
h=bar(fRiseTiemes,'Tag','bar1');
title('Rise time and time at 50% of ampl');
set(get(h(1),'BaseLine'),'LineWidth',2,'LineStyle',':');
colormap summer % Change the color scheme
pRecoveredOutputSignal=pRecoveredOutputSignal(1:nSamplesToShow);
pIn=pIn(1:nSamplesToShow);
pTimeLine=pTimeLine(1:nSamplesToShow);
maxOutputValue10Array = maxOutputValue10* ones(nSamplesToShow);
maxOutputValue50Array = maxOutputValue50* ones(nSamplesToShow);
maxOutputValue90Array = maxOutputValue90* ones(nSamplesToShow);
maxInputValue10Array = maxInputValue10* ones(nSamplesToShow);
maxInputValue50Array = maxInputValue50* ones(nSamplesToShow);
maxInputValue90Array = maxInputValue90* ones(nSamplesToShow);
subplot(2,2,1);
plot( pTimeLine, pIn,'k','LineWidth',1);
hold on;
plot( pTimeLine, pRecoveredOutputSignal,'b--','LineWidth',2);
title('Semanlul de intrare si semnalul reconstruit (punctat)');
axis([0 1.5 0 max(maxOutputValue+10, maxInputValue + 10)]);
xlabel('timp [ms]');
ylabel('amplitudinea [V]');
grid on;
subplot(2,2,2);
plot( pTimeLine, pIn,'k','LineWidth',1);
hold on;
plot( pTimeLine, pRecoveredOutputSignalCalib,'b--','LineWidth',2);
title('Semanlul de intrare si semnalul de iesire (punctat)');
axis([0 1.5 0 max(maxOutputCalibValue+10, maxInputValue + 10)]);
xlabel('timp [ms]');
ylabel('amplitudinea [V]');
grid on;

% Calculul erorilor relative ale semnalului calibrat și decomprimat
for i=1:nSamplesToShow
    if (abs(pRecoveredOutputSignal(i)-pIn(i)) < 0.001)
        fRelativeError(i)=0;
    else

```

```

        fRelativeError(i)=((pRecoveredOutputSignal(i)-pIn(i))/maxInputValue)*100;
    end
end
for i=1:nSamplesToShow
    if (abs(pRecoveredOutputSignal(i)-pIn(i)) < 0.001)
        fRelativeErrorCalib(i)=0;
    else
        fRelativeErrorCalib(i)=((pRecoveredOutputSignalCalib(i)-
pIn(i))/maxInputValue)*100;
    end
end

pTimeLine=pTimeLine(2:nSamplesToShow) ;
fRelativeError=fRelativeError(2:nSamplesToShow);
fRelativeErrorCalib=fRelativeErrorCalib(2:nSamplesToShow);
subplot(2,2,3);
plot(pTimeLine,fRelativeError, 'k');
axis([0 1.5 0 maxOutputValue]);
axis 'auto y';
%title('');
xlabel('timp [ms]');
ylabel('eroarea relativa (%)');
grid on
subplot(2,2,4);
plot(pTimeLine,fRelativeErrorCalib, 'k');
axis([0 1.5 0 maxOutputValue]);
axis 'auto y';
%title('Eroarea relativa (%)');
xlabel('timp [ms]');
ylabel('eroarea relativa (%)');
grid on
% Calculul energiei semnalelor și a erorilor relative ale energiilor
fRecoveredSignalEnergy = 0;
fRecoveredCalibSignalEnergy = 0;
fInputSignalEnergy = 0;
for i=1:nSignalLength
    fRecoveredSignalEnergy = fRecoveredSignalEnergy + pRecoveredOutputSignal(i);
    fRecoveredCalibSignalEnergy = fRecoveredCalibSignalEnergy +
pRecoveredOutputSignalCalib(i);
    fInputSignalEnergy = fInputSignalEnergy + pIn(i);
end
fRecoveredSignalEnergy = fRecoveredSignalEnergy * 0.0015
fRecoveredCalibSignalEnergy = fRecoveredCalibSignalEnergy * 0.0015
fInputSignalEnergy = fInputSignalEnergy * 0.0015
fRelativeErrorOfRecoveredSignalEnergy = ((fRecoveredSignalEnergy -
fInputSignalEnergy) / fInputSignalEnergy) * 100
fRelativeErrorOfRecoveredCalibSignalEnergy = ((fRecoveredCalibSignalEnergy -
fInputSignalEnergy) / fInputSignalEnergy) * 100

```

Anexa 2.2 Program MatLab pentru trasarea caracteristicii de transfer a divizorului secvențial de tensiune

```
% Program MatLab pentru afisarea caracteristicii de transfer
% a divizorului de tensiune secvential
```

```
clear;
```

```
% initializare variabile
```

```
a1 = 0.02;
```

```
a2 = 0.005;
```

```
a3 = 0.001;
```

```
C = 10;
```

```
nSamples = 10000;
```

```
% genereaza esantioanele caracteristicilor de transfer
```

```
% ale celor trei divizoare de tensiune rezistive
```

```
for i=1:nSamples
```

```
    x(i) = i;
```

```
    x1(i) = i*a1;
```

```
    if (x1(i) > 10)
```

```
        x1(i) = 10;
```

```
    end
```

```
    x2(i) = i*a2;
```

```
    if (x2(i) > 10)
```

```
        x2(i) = 10;
```

```
    end
```

```
    x3(i) = i*a3;
```

```
end
```

```
rez2 = x1+x2+x3;
```

```
% Afiseaza caracteristicile de transfer ale
```

```
% fiecarui divizor rezistiv in parte
```

```
figure(1);
```

```
plot(x, x1, x, x2, x, x3);
```

```
% genereaza esantioanele caracteristicii de transfer
```

```
% a divizorului de tensiune secvential
```

```
y1 = C*(a2/a1)
```

```
y2 = C*(a3/a2)
```

```
delta1 = C-y1
```

```
delta2 = C-y1 + C-y2
```

```
for i=1:nSamples
```

```
    if (x2(i) > y1)
```

```
        if (x3(i) > y2)
```

```
            rez(i) = x3(i) + delta2;
```

```
        else
```

```
            rez(i) = x2(i) + delta1;
```

```
        end
```

```
end
```

```
    else
        rez(i) = x1(i);
    end
end

% Afiseaza caracteristica de transfer a divizorului de tensiune secvential
figure(2);
plot(x, rez);
grid on;

% reconstruieste semnalul de intrare
for i=1:nSamples
    if (rez(i) > 10)
        if (rez(i) > 17.5)
            y(i) = (rez(i) - delta2)/a3;
        else
            y(i) = (rez(i) - delta1)/a2;
        end
    else
        y(i) = rez(i)/a1;
    end
end

% Afiseaza semnalul reconstruit
figure(3);
plot(x, y);
grid on;
```


ANEXA 3. PROGRAM VISUAL C++ PENTRU MONITORIZAREA CALITĂȚII ENERGIEI ELECTRICE

Anexa 3.1 Program pentru inițializarea aplicației și achiziția de date

Clasa CPowerQualityMonitoringSystemDlg înglobează funcțiile de interfață cu utilizatorul și funcțiile de achiziție de date

```
////////////////////////////////////  
// Power Quality Monitoring SystemDlg.cpp : implementare  
////////////////////////////////////  
#include "stdafx.h"  
#include "Power Quality Monitoring System.h"  
#include "Power Quality Monitoring SystemDlg.h"  
#include <math.h>  
#include ".\power quality monitoring systemdlg.h"  
  
#ifdef _DEBUG  
#define new DEBUG_NEW  
#endif  
  
CPowerQualityMonitoringSystemDlg::CPowerQualityMonitoringSystemDlg(CWnd*  
pParent /*=NULL*/)  
: CDialog(CPowerQualityMonitoringSystemDlg::IDD, pParent) {  
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);  
  
    m_bStopThread          = FALSE;  
    m_pReadDataBuffer      = NULL;  
    m_nChannels            = 3;  
    m_dblSampleRate        = 250000.0;  
    m_nReadChanSamples     = (int)floor(m_dblSampleRate / 40.);  
}  
  
void CPowerQualityMonitoringSystemDlg::DoDataExchange(CDataExchange* pDX)  
{  
    CDialog::DoDataExchange(pDX);  
    DDX_Control(pDX, IDC_STATIC_SIGNAL_1, m_ctrlSignalScope1);  
    DDX_Control(pDX, IDC_STATIC_SIGNAL_2, m_ctrlSignalScope2);  
    DDX_Control(pDX, IDC_STATIC_TEXT_DISPLAY, m_ctrlTextDisplay);  
    DDX_Control(pDX, IDC_LIST_EVENTS, m_ctrlEventsList);  
}
```

```
BEGIN_MESSAGE_MAP(CPowerQualityMonitoringSystemDlg, CDialog)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    //}}AFX_MSG_MAP
    ON_WM_DESTROY()
    ON_COMMAND(ID_BUTTON_START, OnToolButtonStart)
    ON_COMMAND(ID_BUTTON_STOP, OnToolButtonStop)
    ON_COMMAND(ID_BUTTON_SAVE, OnToolButtonSave)
    //}}AFX_MSG_MAP
    ON_WM_SIZE()
END_MESSAGE_MAP()

BOOL CPowerQualityMonitoringSystemDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {
        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty()) {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }
    SetIcon(m_hIcon, TRUE);
    SetIcon(m_hIcon, FALSE);
    ShowWindow(SW_MAXIMIZE);

    // inițializare listă evenimente
    m_ctrlEventsList.Init();

    // inițializare controale grafice
    CreateToolBar();
    InitControls();

    // inițializare variabile
    InitVariables();

    return TRUE;
}

void CPowerQualityMonitoringSystemDlg::OnSysCommand(UINT nID, LPARAM
lParam)
{
```

```

        if ((nID & 0xFFF0) == IDM_ABOUTBOX)
        {
            CAboutDlg dlgAbout;
            dlgAbout.DoModal();
        }
        else
        {
            CDialog::OnSysCommand(nID, lParam);
        }
    }

void CPowerQualityMonitoringSystemDlg::OnPaint()
{
    if (IsIconic()) {
        CPaintDC dc(this);

        SendMessage(WM_ICONERASEBKGND,
            reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;
        dc.DrawIcon(x, y, m_hIcon);
    }
    else {
        CPaintDC dc(this);
        CDC dcMemory;
        CBitmap bitmap;
        CBitmap* pOldBitmap;
        if (this->IsWindowVisible()) {
            CRect rcClient;
            CSize textExtent;
            GetClientRect(&rcClient);
            CBrush brBkBrush;
            brBkBrush.CreateSolidBrush(RGB_BACK_COLOR);
            dcMemory.CreateCompatibleDC(&dc);
            dcMemory.SetBkColor(RGB(0,0,0));
            bitmap.CreateCompatibleBitmap(&dc, rcClient.Width(),
rcClient.Height());
            pOldBitmap = dcMemory.SelectObject(&bitmap);
            dcMemory.FillRect(rcClient, &brBkBrush);
            dc.BitBlt(0, 0, rcClient.Width(), rcClient.Height(),
&dcMemory, 0, 0, SRCCOPY);
            dcMemory.SelectObject(pOldBitmap);
        }
        CDialog::OnPaint();
    }
}

```

```

HCURSOR CPowerQualityMonitoringSystemDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

void CPowerQualityMonitoringSystemDlg::InitVariables()
{
    m_wndToolBar.m_ctrlCircularBuffer.SetBufferSize(m_ProgramParameters.GetCircularBufferSize());
    m_CircularBuffer.Create(m_ProgramParameters.GetCircularBufferSize());
    m_nChannels = m_ProgramParameters.GetChannels();
    m_ctrlSignalScope1.SetChannels(m_nChannels);
    m_SignalProcessing.SetChannels(m_nChannels);
}

BOOL CPowerQualityMonitoringSystemDlg::InitControls(void)
{
    CRect      rcClient;
    int        nGraphHeight,
              nGraphY,
              nGraphWidth;

    // Repoziționare automată a controalelor pe interfață
    GetClientRect(&rcClient);
    nGraphHeight = ((rcClient.Height() - GRAPH_MARGIN_Y) / NO_OF_GRAPHS
- (NO_OF_GRAPHS - 1) * GRAPH_SPACING_Y );
    nGraphWidth = ((rcClient.Width() - GRAPH_MARGIN_X) - 3 *
GRAPH_SPACING_X - TEXT_CONTROL_WIDTH);
    nGraphY = GRAPH_MARGIN_Y;
    m_ctrlSignalScope1.MoveWindow(GRAPH_MARGIN_X, nGraphY,
nGraphWidth - GRAPH_SPACING_X, nGraphHeight, TRUE);
    m_ctrlTextDisplay.MoveWindow(nGraphWidth, nGraphY, rcClient.Width() -
nGraphWidth - GRAPH_MARGIN_X, nGraphHeight, TRUE);
    nGraphY = GRAPH_MARGIN_Y + nGraphHeight * 1 + 1 *
GRAPH_SPACING_Y;
    m_ctrlSignalScope2.MoveWindow(GRAPH_MARGIN_X, nGraphY,
nGraphWidth - GRAPH_SPACING_X, nGraphHeight, TRUE);
    nGraphY = GRAPH_MARGIN_Y + nGraphHeight * 2 + 2 *
GRAPH_SPACING_Y;
    //m_ctrlSignalScope3.MoveWindow(GRAPH_MARGIN_X, nGraphY, 600,
nGraphHeight, TRUE);
    m_ctrlEventsList.MoveWindow(GRAPH_MARGIN_X, nGraphY, rcClient.right,
nGraphHeight, TRUE);

    // forțează redesenarea interfeței
    Invalidate();
    return TRUE;
}

void CPowerQualityMonitoringSystemDlg::CreateToolBar()
{

```

```

    if(!m_wndToolBar.Create(this)
        || !m_wndToolBar.LoadToolBar(IDR_TOOLBAR1))
    {
        TRACE0("Failed to Create Dialog Toolbar\n");
        EndDialog(IDCANCEL);
    }

    CRect rcClientOld;
    CRect rcClientNew;
    GetClientRect(rcClientOld);
    RepositionBars(AFX_IDW_CONTROLBAR_FIRST,
        AFX_IDW_CONTROLBAR_LAST,0,reposQuery,rcClientNew);
    CPoint ptOffset(rcClientNew.left-rcClientOld.left,
        rcClientNew.top-rcClientOld.top);
    CRect rcChild;
    CWnd* pwndChild = GetWindow(GW_CHILD);
    while(pwndChild)
    {
        pwndChild->GetWindowRect(rcChild);
        ScreenToClient(rcChild);
        rcChild.OffsetRect(ptOffset);
        pwndChild->MoveWindow(rcChild,FALSE);
        pwndChild = pwndChild->GetNextWindow();
    }

    CRect rcWindow;
    GetWindowRect(rcWindow);
    rcWindow.right += rcClientOld.Width() - rcClientNew.Width();
    rcWindow.bottom += rcClientOld.Height() - rcClientNew.Height();
    MoveWindow(rcWindow,FALSE); // Redesenează fereastra
    RepositionBars(AFX_IDW_CONTROLBAR_FIRST,
        AFX_IDW_CONTROLBAR_LAST,0);
}

BOOL CPowerQualityMonitoringSystemDlg::InitADC(void)
{
#define DAQmxErrChk(functionCall)
{
    if(DAQmxFailed(error=(functionCall))) {
        DisplayError(error); }
}

    int32    error = 0;
    float64  min   = -10.0;
    float64  max   = 10.0;
    char     clockSource[] = "OnboardClock";
    CString  strBoardName;
    CString  strBoardChannels;

    m_hTaskHandle = 0;

```

```

#ifdef _PQM_TEST
    m_pReadDataBuffer = new double[200000 * sizeof(double)];
    return TRUE;
#else
    m_pReadDataBuffer = new double[m_nChannels * m_nReadChanSamples *
sizeof(double)];
#endif

    strBoardName = m_ProgramParameters.GetBoardName();
    switch (m_nChannels) {
        case 1: strBoardChannels.Format("%s/ai0", strBoardName);
            break;
        case 2: strBoardChannels.Format("%s/ai0:1", strBoardName);
            break;
        case 3: strBoardChannels.Format("%s/ai0:2", strBoardName);
            break;
        case 4: strBoardChannels.Format("%s/ai0:3", strBoardName);
            break;
        default: strBoardChannels.Format("%s/ai0", strBoardName);
            break;
    }

    DAQmxErrChk (DAQmxCreateTask("", &m_hTaskHandle));
    DAQmxErrChk (DAQmxCreateAIVoltageChan(m_hTaskHandle,
        strBoardChannels.GetBuffer(0),
        "",
        DAQmx_Val_RSE,
        min,
        max,
        DAQmx_Val_Volts,
        NULL));
    DAQmxErrChk (DAQmxCfgSampClkTiming(m_hTaskHandle, clockSource,
        m_dblSampleRate,
        DAQmx_Val_Rising,
        DAQmx_Val_ContSamps,
        m_nReadChanSamples));
    DAQmxErrChk (DAQmxCfgInputBuffer(m_hTaskHandle, 100000));
    DAQmxErrChk (DAQmxStartTask(m_hTaskHandle));
    return TRUE;
}

DWORD WINAPI CPowerQualityMonitoringSystemDlg::AquisitionThread(LPVOID
lpParam)
{
    CPowerQualityMonitoringSystemDlg* pDlg =
        (CPowerQualityMonitoringSystemDlg*)lpParam;

    int32  pointsToRead = pDlg->m_nReadChanSamples;
    int32  nPointsRead;
    float64 timeout = 10.0;
    int32  totalRead = 0;

```

```

char    errBuff[2048];
int     nError = 0;

while( !pDlg->m_bStopThread ) {
    nError = DAQmxReadAnalogF64(pDlg->m_hTaskHandle,
        pointsToRead,
        timeout,
        DAQmx_Val_GroupByScanNumber,
        pDlg->m_pReadDataBuffer,
        pDlg->m_nReadChanSamples * pDlg->m_nChannels,
        &nPointsRead,
        NULL);

    // scrie eşantioanele citite în bufferul circular
    while(!pDlg->m_CircularBuffer.Write(pDlg->m_pReadDataBuffer,
nPointsRead / pDlg->m_nChannels));
        pDlg->m_wndToolBar.SetWriteData(pDlg->
            m_CircularBuffer.GetWritePos(), nPointsRead);
    // permite rularea altor procese în paralel
    Sleep(0);
}

pDlg->DAQmxStopTask(pDlg->m_hTaskHandle);
pDlg->DAQmxClearTask(pDlg->m_hTaskHandle);

// semnalizare sfârșit fir de execuție
::SetEvent( pDlg->m_hThreadEvents[ADCThreadEvent] );

// așteaptă finalizarea execuției funcțiilor ce gestionează sfârșitul firului de
// execuție
Sleep(2000);
ExitThread(0);
return TRUE;
}

DWORD WINAPI CPowerQualityMonitoringSystemDlg::
ReadFromCircularBufferThread(LPVOID lpParam)
{
    CPowerQualityMonitoringSystemDlg* pDlg =
        (CPowerQualityMonitoringSystemDlg*)lpParam;

typedef CTypedPtrList<COBList, CSignalParameters*> CSgnParamList;

    long    lRead = 0;
    CString strRead;
    long    lFirstPos = 0;
    long    lOldReadPos = 0;
    long    lSignalAnalyzeLastPos = 0;
    double  dblFreq = 0.00;
    long    lRefreshRate = 0;
    int     nLastPert = 0;

```

```

int    nCurrPert = 0;
int    nOneSemiPeriodSamples = 1;
long   lSamplesToReadFromBuffer;
long   lSamplesToRecord = (int)(pDlg->m_dbfSampleRate) / 3;
nOneSemiPeriodSamples = (int)(pDlg->m_dbfSampleRate / 100.);
lSamplesToReadFromBuffer = nOneSemiPeriodSamples * pDlg->m_nChannels * 4;
SYSTEMTIME sysTime;
CString strDateTime;
double *pData;
double *pDisturbanceData;
pData = (double*)malloc(5 * pDlg->m_nReadChanSamples * pDlg->m_nChannels
* sizeof(double));
ZeroMemory(pData, 5 * pDlg->m_nReadChanSamples * pDlg->m_nChannels *
sizeof(double));
pDisturbanceData = (double*)malloc(5 * (int)(pDlg->m_dbfSampleRate) * pDlg-
>m_nChannels * sizeof(double));
ZeroMemory(pDisturbanceData, 5 * (int)(pDlg->m_dbfSampleRate) * pDlg-
>m_nChannels * sizeof(double));

pDlg->m_SignalProcessing.SetSampleRate(pDlg->m_dbfSampleRate);
pDlg->m_SignalProcessing.SetDelta(0.6);
pDlg->m_ctrlSignalScope1.SetYAxisRange(-6, 6);
pDlg->m_ctrlSignalScope1.SetTitle("Main Signal");
pDlg->m_ctrlSignalScope2.SetYAxisRange(-6, 6);
pDlg->m_ctrlSignalScope2.SetTitle("Disturbance");

while(!pDlg->m_bStopThread) {
    if (pDlg->m_CircularBuffer.Read(pData, lSamplesToReadFromBuffer)) {
        lSignalAnalyzeLastPos =
            pDlg->m_SignalProcessing.ProcessSignalSeparateOneSemiPeriod(
                pData, 0, lSamplesToReadFromBuffer);
        pDlg->m_ctrlTextDisplay.SetTextValues(
            pDlg->m_SignalProcessing.GetRMS(),
            pDlg->m_SignalProcessing.GetMax(),
            pDlg->m_SignalProcessing.GetCrestFactor(),
            pDlg->m_SignalProcessing.GetFrequency(),
            pDlg->m_SignalProcessing.GetAvgRMS(),
            pDlg->m_SignalProcessing.GetAvgMax(),
            pDlg->m_SignalProcessing.GetAvgCrestFactor(),
            pDlg->m_SignalProcessing.GetAvgFrequency());

        if (pDlg->m_SignalProcessing.DetectDisturbance()) {
            pDlg->m_ctrlTextDisplay.SetDisturbance(TRUE);
            nCurrPert++;
            nLastPert = nCurrPert;
            GetLocalTime(&sysTime);
            strDateTime.Format(" %04d-%02d-%02d
                %02d-%02d-%02d",
                sysTime.wYear,
                sysTime.wMonth,
                sysTime.wDay,

```



```

        sysTime.wHour,
        sysTime.wMinute,
        sysTime.wSecond);

        pDlg->m_ctrlEventsList.AddData(strDateTime,
        "Disturbance",
        pDlg->m_SignalProcessing.GetFrequency(),
        pDlg->m_SignalProcessing.GetAvgFrequency(),
        pDlg->m_SignalProcessing.GetRMS(),
        pDlg->m_SignalProcessing.GetAvgRMS(),
        pDlg->m_SignalProcessing.GetMax(),
        pDlg->m_SignalProcessing.GetAvgMax(),
        pDlg->m_SignalProcessing.GetCrestFactor(),
        pDlg->m_SignalProcessing.GetAvgCrestFactor());
    }
    else {
        pDlg->m_ctrlTextDisplay.SetDisturbance(FALSE);
        nLastPert = nCurrPert;
        nCurrPert = 0;
    }
    if ((nLastPert > 0) && (nCurrPert == 0)) {
        pDlg->m_CircularBuffer.GetBufferFrom(pDisturbanceData,
        IOldReadPos - ISamplesToRecord / 3 , ISamplesToRecord);
        pDlg->m_ctrlSignalScope2.SetSignal(pDisturbanceData,
        ISamplesToRecord );
        pDlg->m_CircularBuffer.SaveBufferFrom(
        IOldReadPos,
        ISamplesToRecord,
        pDlg->m_SignalProcessing.GetFrequency(),
        pDlg->m_SignalProcessing.GetAvgFrequency(),
        pDlg->m_SignalProcessing.GetRMS(),
        pDlg->m_SignalProcessing.GetAvgRMS(),
        pDlg->m_SignalProcessing.GetMax(),
        pDlg->m_SignalProcessing.GetAvgMax(),
        pDlg->m_SignalProcessing.GetCrestFactor(),
        pDlg->m_SignalProcessing.GetAvgCrestFactor(),
        pDlg->m_dbfSampleRate);
    }

    // actualizare indicație control grafic memorie circulară
    pDlg->m_wndToolBar.SetReadData(IOldReadPos,
    ISignalAnalyzeLastPos);
    pDlg->m_CircularBuffer.IncrementPos(
    ISignalAnalyzeLastPos , 0 );
    IOldReadPos = pDlg->m_CircularBuffer.GetRreadPos();

    // Afișare semnal rețea
    if ((IRefreshRate % 1000) == 0) {
        IRefreshRate = 0;
        pDlg->m_ctrlSignalScope1.SetSignal(pData, ISamplesToReadFromBuffer);
    }

```

```

}

    lRefreshRate++;

    Sleep(0);
}

// Eliberare zone de memorie
free(pData);
free(pDisturbanceData);

// semnalizare terminare fir de execuție
::SetEvent( pDlg->m_hThreadEvents[ReadCircBufferThreadEvent] );
ExitThread(0);
return TRUE;
}

void CPowerQualityMonitoringSystemDlg::DisplayError(int nError)
{
    char    errBuff[2048];
    if( DAQmxFailed(nError) )
        DAQmxGetExtendedErrorInfo(errBuff,2048);
    if(m_hTaskHandle != 0) {
        DAQmxStopTask (m_hTaskHandle);
        DAQmxClearTask (m_hTaskHandle);
    }
    if( DAQmxFailed(nError) )
        AfxMessageBox( CString(errBuff));
    return;
}

void CPowerQualityMonitoringSystemDlg::OnToolButtonStart()
{
    InitADC();
    m_hThreadEvents[ADCThreadEvent] = ::CreateEvent(
        NULL, TRUE, FALSE, NULL );
    m_hThreadEvents[ReadCircBufferThreadEvent] = ::CreateEvent(
        NULL, TRUE, FALSE, NULL );
    m_hThreadEvents[InterfaceThreadEvent] = ::CreateEvent(
        NULL, TRUE, FALSE, NULL );
    m_hADCThread = CreateThread(
        NULL,
        0,
        AquisitionThread,
        this,
        CREATE_SUSPENDED,
        &m_ulADCThreadID );
    m_hReadCircBufferThread = CreateThread( NULL,
        0,
        ReadFromCircularBufferThread,
        this,
        CREATE_SUSPENDED,

```

```

        &m_ulReadCircBufferThreadID );

    ResumeThread(m_hADCThread);
    ResumeThread(m_hReadCircBufferThread);
}

void CPowerQualityMonitoringSystemDlg::OnToolButtonStop()
{
    m_bStopThread = TRUE;

    // așteaptă oprirea firelor de execuție
    DWORD dwWait = WaitForMultipleObjects(    2,
                                             m_hThreadEvents,
                                             FALSE,
                                             3000 );

    switch ( dwWait )
    {
        case WAIT_OBJECT_0:
        {
            ::ResetEvent( m_hThreadEvents[ADCThreadEvent] );
        }
    }
    Sleep(200);
    OnCancel();
}

void CPowerQualityMonitoringSystemDlg::OnToolButtonSave()
{
    m_CircularBuffer.SaveBuffer();
}

void CPowerQualityMonitoringSystemDlg::OnDestroy()
{
    CDialog::OnDestroy();

    int i = 0;
    if (m_pReadDataBuffer) {
        delete m_pReadDataBuffer;
    }
    m_CircularBuffer.Release();
}

void CPowerQualityMonitoringSystemDlg::OnSize(UINT nType, int cx, int cy)
{
    CDialog::OnSize(nType, cx, cy);
    if(this->IsWindowVisible()) {
        m_wndToolBar.ReSizeControls();
        InitControls();
    }
}

```

}

Anexa 3.2 Program pentru detecția perturbațiilor din rețeaua electrică

```

////////////////////////////////////
// SignalProcessing.cpp: implementation of the CSignalProcessing class
////////////////////////////////////

#include "stdafx.h"
#include "../Power Quality Monitoring System.h"
#include "SignalProcessing.h"
#include <Math.h>

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

////////////////////////////////////
// Construction/Destruction
////////////////////////////////////

CSignalProcessing::CSignalProcessing()
{
    // Inițializarea buffer-elor de memorie
    for (int i = 0; i < NO_OF_SIGNALS; i++) {
        m_pSignalBuffer[i] = NULL;
    }

    for (int i = 0; i < NO_OF_PARAMETERS; i++) {
        m_pSignalParameters[i] = NULL;
    }

    // Valorile implicite ale parametrilor
    SetSampleRate(1000.0);
    m_dblFreq = 50.0;
    m_dblFreqDelta = 5.0;
    m_lSamplesPerPeriod = 0;
    m_nCurrentParamItem = 0;
    m_nChannels = 1;

    InitParamBuffers();
    m_dblDelta = 0.5;
    m_dblAvgMax = 6.0;
    m_dblAvgRMS = 5.0;
    m_dblAvgCrestFactor = 1.6;
    m_dblAvgFreq = 50.0;
}

```

```

}

CSignalProcessing::~CSignalProcessing()
{
    // Funcție apelată la distrugerea obiectului
    // Sunt eliberate toate zonele de memorie alocate
    ReleaseBuffers();

    //
    for (int i = 0; i < NO_OF_PARAMETERS; i++) {
        if (m_pSignalParameters[i] != NULL) {
            free (m_pSignalParameters[i]);
        }
    }
}

long CSignalProcessing::GetSamplesPerPeriod()
{
    return m_lSamplesPerPeriod;
}

long CSignalProcessing::ProcessSignal(double *dwInputSignal, long lStart, long lEnd)
{
    double dblAbsSignalVal = 0.0;
    long lLastZeroCross = 0;

    for (long lIndex = lStart; lIndex < lEnd; lIndex++) {
        m_pSignalBuffer[OriginalSignal][lIndex] = dwInputSignal[lIndex];
        dblAbsSignalVal = abs(m_pSignalBuffer[OriginalSignal][lIndex]);
        if (floor(m_pSignalBuffer[OriginalSignal][lIndex] * 10000) !=
            floor(dblAbsSignalVal * 10000)) {
            m_pSignalBuffer[ZeroCrossSignal][lIndex] = -1;
        }
        else {
            m_pSignalBuffer[ZeroCrossSignal][lIndex] = +1;
        }
        m_pSignalBuffer[RectifiedSignal][lIndex] = dblAbsSignalVal;
        if (lIndex > lStart) {
            if (m_pSignalBuffer[ZeroCrossSignal][lIndex] !=
                m_pSignalBuffer[ZeroCrossSignal][lIndex - 1]) {
                CalculateSignalParams(lLastZeroCross, lIndex);
                lLastZeroCross = lIndex;
            }
        }
    }

    return lLastZeroCross;
}

```

```

void CSignalProcessing::CalculateSignalParameters(long IStart, long IEnd)
{
    CSignalParameters *sgnParam = new CSignalParameters;

    double dblMax      = -1000.0;
    double dblRMS      = 0.0;
    double dblCrestFactor = 0.0;

    if (IEnd - IStart <= 0) {
        return;
    }

    for (int i = IStart; i < IEnd; i++) {
        if (m_pSignalBuffer[RectifiedSignal][i] > dblMax) {
            dblMax = m_pSignalBuffer[RectifiedSignal][i];
        }
        dblRMS += m_pSignalBuffer[RectifiedSignal][i];
    }

    dblRMS /= (IEnd - IStart);

    sgnParam->SetMax(dblMax);
    sgnParam->SetCrestFactor(dblMax / dblRMS);
    sgnParam->SetRMS(dblRMS);
    sgnParam->SetFirstSample(IStart);

    m_obParamList.AddTail(sgnParam);
}

BOOL CSignalProcessing::ReleaseBuffers(void)
{
    // eliberează zonele de memorie alocate semnalelor
    for (int i = 0; i < NO_OF_SIGNALS; i++) {
        if (m_pSignalBuffer[i] != NULL) {
            free (m_pSignalBuffer[i]);
        }
    }

    // eliberează toate zonele de memorie alocate, altfel acestea rămânând
    // alocate la ieșirea din aplicație
    for (POSITION pos = m_obParamList.GetHeadPosition(); pos != NULL; )
    {
        CSignalParameters *sgnParam = m_obParamList.GetNext(pos);
        delete sgnParam;
    }

    // șterge elementele listei de obiecte
    while (m_obParamList.GetCount() > 0) {
        m_obParamList.RemoveAll();
    }
}

```

```

        return TRUE;
    }

// Inițializarea zonelor de memorie corespunzătoare parametrilor sistemului de
// detecție a perturbațiilor
BOOL CSignalProcessing::InitParamBuffers()
{
    for (int i = 0; i < NO_OF_PARAMETERS; i++) {
        m_pSignalParameters[i] = (double*) malloc(MAX_PARAM_ITEMS *
sizeof(double));

        if (m_pSignalParameters[i] == NULL) {
            return FALSE;
        }
        else {
            for (int j = 0; j < MAX_PARAM_ITEMS; j++) {
                m_pSignalParameters[i][j] = NULL;
            }
        }
    }
    return TRUE;
}

BOOL CSignalProcessing::InitBuffers(long lLength)
{
    // Eliberează zonele de memorie alocate anterior
    ReleaseBuffers();

    // Alocare zonă de memorie
    for (int i = 0; i < NO_OF_SIGNALS; i++) {
        m_pSignalBuffer[i] = (double*) malloc(lLength * sizeof(double) );

        if (m_pSignalBuffer[i] == NULL) {
            return FALSE;
        }
    }
    return TRUE;
}

double* CSignalProcessing::GetZeroCrossBuffer(void)
{
    return m_pSignalBuffer[ZeroCrossSignal];
}

long CSignalProcessing::ProcessSignal (double *dwInputSignal,
                                       long lFirstSample,
                                       long lLastSample)
{
    double dblAbsSignalVal    = 0.0;
    int     nOldSign          = 0;

```

```

int     nCurrentSign          = 0;
long    lLastZeroCross        = 0;
BOOL    bFirstZeroCrossDetected = FALSE;
int     nDeltaZeroCrossSamples = 0;

    nDeltaZeroCrossSamples = (int)(m_ulSamplesPerSemiperiod * 0.02);
m_lSamplesPerPeriod = lLastSample;
m_dblMax           = -100000.0;
m_dblRMS           = 0.0;
m_dblCrestFactor   = 0.0;

if (!InitBuffers(lLastSample)) {
    AfxMessageBox("Eroare alocare buffere!", MB_OK);
    return 0;
}

for(long lIndex=lFirstSample; lIndex< lLastSample; lIndex+=m_nChannels)
{
    m_pSignalBuffer[OriginalSignal][lIndex] = dwInputSignal[lIndex];
    dblAbsSignalVal = abs(m_pSignalBuffer[OriginalSignal][lIndex]);
    if (floor(m_pSignalBuffer[OriginalSignal][lIndex] * 10000) !=
        floor(dblAbsSignalVal * 10000))
    {
        m_pSignalBuffer[ZeroCrossSignal][lIndex] = -1;
    }
    else {
        m_pSignalBuffer[ZeroCrossSignal][lIndex] = +1;
    }
    m_pSignalBuffer[RectifiedSignal][lIndex] = dblAbsSignalVal;
    if (lIndex > (lFirstSample + nDeltaZeroCrossSamples * m_nChannels))
    {
        if((m_pSignalBuffer[ZeroCrossSignal][lIndex] !=
            m_pSignalBuffer[ZeroCrossSignal][lIndex - m_nChannels]) &&
            !bFirstZeroCrossDetected)
        {
            bFirstZeroCrossDetected = TRUE;
            // a fost detectată o trecere prin zero
            for (int i = lFirstSample; i < lIndex; i += m_nChannels) {
                if (m_pSignalBuffer[RectifiedSignal][i] > m_dblMax) {
                    m_dblMax = m_pSignalBuffer[RectifiedSignal][i];
                }
                m_dblRMS += m_pSignalBuffer[RectifiedSignal][i];
            }
            m_lSamplesPerPeriod = lIndex;
            m_dblRMS /= (lIndex - lFirstSample);
            if (m_dblRMS != 0) {
                m_dblCrestFactor = m_dblMax / m_dblRMS;
            }
            else {
                m_dblCrestFactor = 1.0;
            }
        }
    }
}

```



```

        if (m_nSamplesPerPeriod != 0) {
            m_dblFreq = m_dblSampleRate / (m_nSamplesPerPeriod * 2);
        }
        else {
            m_dblFreq = 0.0;
        }

m_pSignalParameters[ParamRMSVal][m_nCurrentParamItem] = m_dblRMS;
m_pSignalParameters[ParamMaxVal][m_nCurrentParamItem] = m_dblMax;
m_pSignalParameters[ParamCrestFact][m_nCurrentParamItem]
=m_dblCrestFactor;
m_pSignalParameters[ParamFreq][m_nCurrentParamItem] = m_dblFreq;
CalculateAverageParamValues();
IncrementParamItem();
    }
}

if (m_nSamplesPerPeriod == (lLastSample - lFirstSample)) {
    // Nu s-a gasit nici o trecere prin zero
    // Cel mai probabil nu avem nici un semnal aplicat la intrare.
    m_dblMax = 5;
    m_dblRMS = 1;
    m_dblCrestFactor = 5;
    m_dblFreq = 0;
    m_pSignalParameters[ParamRMSVal][m_nCurrentParamItem] = 1;
    m_pSignalParameters[ParamMaxVal][m_nCurrentParamItem] = 5;
    m_pSignalParameters[ParamCrestFact][m_nCurrentParamItem] = 5;
    m_pSignalParameters[ParamFreq][m_nCurrentParamItem] = 0;
    CalculateAverageParamValues();
    IncrementParamItem();
}
return m_nSamplesPerPeriod;
}

BOOL CSignalProcessing::DetectDisturbance()
{
    if (abs(m_dblRMS - m_dblAvgRMS) > m_dblDelta ) {
        return TRUE;
    }

    if (abs(m_dblMax - m_dblAvgMax) > m_dblDelta ) {
        return TRUE;
    }

    if (abs(m_dblCrestFactor - m_dblAvgCrestFactor) > m_dblDelta ) {
        return TRUE;
    }

    if (abs(GetFrequency() - m_dblAvgFreq) > m_dblFreqDelta ) {
        return TRUE;
    }
}

```

```
    }  
    return FALSE;  
}  
  
void CSignalProcessing::CalculateAverageParamValues()  
{  
    m_dblAvgMax      = 0.0;  
    m_dblAvgRMS      = 0.0;  
    m_dblAvgCrestFactor = 0.0;  
    m_dblAvgFreq     = 0.0;  
  
    for (int i = 0; i < MAX_PARAM_ITEMS; i++) {  
        m_dblAvgMax += m_pSignalParameters[ParamMaxVal][i];  
        m_dblAvgRMS += m_pSignalParameters[ParamRMSVal][i];  
        m_dblAvgFreq += m_pSignalParameters[ParamFreq][i];  
    }  
  
    m_dblAvgMax /= (double)MAX_PARAM_ITEMS;  
    m_dblAvgRMS /= (double)MAX_PARAM_ITEMS;  
    m_dblAvgFreq /= (double)MAX_PARAM_ITEMS;  
    m_dblAvgCrestFactor = m_dblAvgMax / m_dblAvgRMS;  
}  
  
void CSignalProcessing::IncrementParamItem()  
{  
    if (m_nCurrentParamItem < (MAX_PARAM_ITEMS - 1)) {  
        m_nCurrentParamItem++;  
    }  
    else {  
        m_nCurrentParamItem = 0;  
    }  
}  
  
double* CSignalProcessing::GetRectifiedSignalBuffer(void)  
{  
    return m_pSignalBuffer[RectifiedSignal];  
}  
  
void CSignalProcessing::SetDelta(double dblDelta)  
{  
    m_dblDelta = dblDelta;  
}  
  
double CSignalProcessing::GetDelta()  
{  
    return m_dblDelta;  
}  
  
void CSignalProcessing::SetFreqDelta(double dblDelta)
```

```
{
    m_dbIFreqDelta = dbIDelta;
}

double CSignalProcessing::GetFrequency()
{
    return m_dbIFreq;
}

void CSignalProcessing::SetSampleRate(double dValue)
{
    m_dbISampleRate = dValue;
    m_ulSamplesPerSemiperiod = m_dbISampleRate / 100.0;
}

double CSignalProcessing::GetCrestFactor()
{
    return m_dbICrestFactor;
}

double CSignalProcessing::GetMax()
{
    return m_dbIMax * 100.;
}

double CSignalProcessing::GetRMS()
{
    return m_dbIRMS * 100.;
}

double CSignalProcessing::GetAvgCrestFactor()
{
    return m_dbIAvgCrestFactor;
}

double CSignalProcessing::GetAvgFrequency()
{
    return m_dbIAvgFreq;
}

double CSignalProcessing::GetAvgMax()
{
    return m_dbIAvgMax * 100.;
}

double CSignalProcessing::GetAvgRMS()
{
    return m_dbIAvgRMS * 100.;
}
```

```

long CSignalProcessing::GetListItems()
{
    return (long)m_obParamList.GetCount();
}

LPARAM CSignalProcessing::GetParamList(long IItem)
{
    return (LPARAM) &m_obParamList;
}

void CSignalProcessing::SetChannels(int nChannels)
{
    m_nChannels = nChannels;
}

```

Anexa 3.3 Program pentru implementarea memoriei circulare

```

////////////////////////////////////
// CCircularBuffer.cpp
////////////////////////////////////
#include "stdafx.h"
#include "../Power Quality Monitoring System.h"
#include "CircularBuffer.h"

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

CCircularBuffer::CCircularBuffer()
{
    m_pData = NULL;
    m_lQueueSize = 0;
    m_lReadPos = 0;
    m_lWritePos = 0;
    m_lock.createLock();
    m_rxEvent.create();
    m_lGlobalWritePos = 0;
    m_lGlobalReadPos = 0;
}

CCircularBuffer::~CCircularBuffer()
{
    Release();
}

```

```

BOOL CCircularBuffer::Create(long lQueueSize)
{
    if ( lQueueSize <= 0 )
        return FALSE;

    // alocare zonă de memorie
    m_lQueueSize = lQueueSize;
    m_ppData = new double[m_lQueueSize * sizeof(double)];

    if ( m_ppData )
    {
        memset( m_ppData, 0, m_lQueueSize * sizeof(double));
        m_rxEvent.create();
        return TRUE;
    }
    else
    {
        m_lQueueSize = 0;
        return FALSE;
    }
}

BOOL CCircularBuffer::Read(double *ppData, long lDataLen, BOOL bWaitForWrite)
{
    long lWriteToEnd = 0;
    long lReadPos = m_lReadPos;
    long lWritePos = m_lWritePos;

    // testează lungimea blocului de date ce se dorește a fi citit
    if ( lDataLen < 0 || lDataLen > m_lQueueSize ) {
        return FALSE;
    }

    // verifică dacă există o operație de scriere în derulare
    if (bWaitForWrite) {
        if (m_lGlobalReadPos + 3 * lDataLen > m_lGlobalWritePos) {
            return FALSE;
        }
    }

    // Dacă nu există nici o operație de scriere, blochează accesul
    // la memoria circulară pe parcursul citirii zonei de memorie
    if ( !m_lock.lock() )
        return FALSE;

    long lLenToEnd = GetLengthToEnd(lReadPos);
    if (lLenToEnd > lDataLen) {
        lLenToEnd = lDataLen;
    }

    if ( lLenToEnd > 0 )

```

```

        memcpy( pData, &m_pData[m_IReadPos], ILenToEnd * sizeof(double));
        long lenRemaining = IDataLen - ILenToEnd;
        if ( lenRemaining > 0 )
            memcpy( &pData[ILenToEnd], &m_pData[0], lenRemaining *
sizeof(double));

        // Deblochează accesul la memoria circulară
        m_lock.unlock();

        return TRUE;
    }

long CCircularBuffer::GetReadPos()
{
    return m_IReadPos;
}

long CCircularBuffer::GetWritePos()
{
    return m_IWritePos;
}

BOOL CCircularBuffer::SetReadPos(long IPos)
{
    if ( IPos > m_IQueueSize )
        return FALSE;

    if (IPos < 0) {
        m_IGlobalReadPos += (IPos - m_IReadPos);
        m_IReadPos = m_IQueueSize + IPos;
        return TRUE;
    }

    m_IGlobalReadPos += (IPos - m_IReadPos);
    m_IReadPos = IPos;
    return TRUE;
}

BOOL CCircularBuffer::SetWritePos(long IPos)
{
    if ( IPos < 0 || IPos > m_IQueueSize )
        return FALSE;
    m_IWritePos = IPos;
    return TRUE;
}

BOOL CCircularBuffer::Write(double *pData, long IDataLen)
{
    // verifică lungimea blocului de memorie care se dorește a fi scris
    if ( IDataLen < 0 || IDataLen > m_IQueueSize )
        return FALSE;

```

```

// blochează accesul la memoria circulară
if ( !m_lock.lock() )
    return FALSE;

long ILenToEnd = GetLengthToEnd(idWrite);
if ( ILenToEnd > IDataLen ) {
    ILenToEnd = IDataLen;
}
if ( ILenToEnd > 0 )
    memcpy( &m_pData[m_IWritePos ], pData, ILenToEnd*sizeof(double));
long ILenRemaining = IDataLen - ILenToEnd;
if ( ILenRemaining > 0 )
    memcpy( &m_pData[0], &pData[ILenToEnd], ILenRemaining *
sizeof(double));
IncrementPos( IDataLen, idWrite );

m_lock.unlock();
return true;
}

BOOL CCircularBuffer::IncrementPos(long IIncrement, long IOperation)
{
    if ( IIncrement < 0 || IIncrement > m_IQueSize )
        return FALSE;
    long * IPos = &m_IWritePos;
    if ( IOperation == idRead )
        IPos = &m_IReadPos;
    long ILenToEnd = GetLengthToEnd(IOperation);
    if ( IIncrement <= ILenToEnd )
        *IPos += IIncrement;
    else
    {
        *IPos = IIncrement - ILenToEnd;
    }
    if (IOperation == idRead) {
        m_GlobalReadPos +=IIncrement;
    }
    else {
        m_GlobalWritePos +=IIncrement;
    }
    if (m_GlobalWritePos > m_IQueSize * 5) {
        m_GlobalWritePos=m_GlobalWritePos - m_GlobalReadPos+m_IQueSize;
        m_GlobalReadPos = m_IQueSize;
    }
    return TRUE;
}

long CCircularBuffer::GetLengthToEnd(long IOperation)
{

```

```
        // returnează lungimea bufferului de memorie de la poziția curentă
        // până la sfârșitul zonei de memorie alocate
        if ( IOperation == idRead )
            return m_IQueSize - m_IReadPos;
        else
            return m_IQueSize - m_IWritePos;
    }

    BOOL CCircularBuffer::Release()
    {
        if ( !m_pData )
            return FALSE;
        Reset();
        delete[] m_pData;
        m_pData = NULL;
        m_IQueSize = 0;
        return TRUE;
    }

    long CCircularBuffer::GetSize()
    {
        return m_IQueSize;
    }

    void CCircularBuffer::Reset()
    {
        m_IReadPos = 0;
        m_IWritePos = 0;
        memset( m_pData, 0, m_IQueSize );
    }

    void CCircularBuffer::WriteOneValue(double dblVal)
    {
        while(!m_lock.lock() );
        m_pData[m_IWritePos] = dblVal;
        IncrementPos( 1, idWrite );
        m_lock.unlock();
    }

    BOOL CCircularBuffer::GetBufferFrom(double *pData, long IStart, long ISamples)
    {
        long IWriteToEnd = 0;
        long IReadPos = m_IReadPos;
        long IWritePos = m_IWritePos;

        if (ISamples > m_IQueSize ) {
            return FALSE;
        }

        if ( !m_lock.lock() )
            if ( IReadPos < ISamples ) {
                long ISamplesLeft = ISamples - IReadPos;
```



```

        memcpy( pData, &m_pData[m_lQueSize - ISamplesLeft],
(ISamplesLeft) * sizeof(double));
        memcpy( &pData[ISamplesLeft], &m_pData[0], (ISamples -
ISamplesLeft) * sizeof(double));
    }
    else {
        memcpy( pData, &m_pData[lReadPos - ISamples], (ISamples) *
sizeof(double));
    }
    m_lock.unlock();
    return TRUE;
}

```

```

BOOL CCircularBuffer::GetAllBuffer(double *pData)

```

```

{
    if (!m_lock.lock()) {
        return FALSE;
    }
    memcpy( pData, m_pData, m_lQueSize * sizeof(double));
    m_lock.unlock();
    return TRUE;
}

```

```

void CCircularBuffer::SaveBufferFrom( long lStart, long lSamples, float fFreq, float
fAvgFreq, float fRMS, float fAvgRMS, float fMax, float fAvgMax, float fCF, float
fAvgCF, float fSampleRate)

```

```

{
    double        *pBuffer;
    FILE          *pFile;
    CString       strWriteLine, strDateTime, strNewFile;
    SYSTEMTIME    sysTime;

    GetLocalTime(&sysTime);

    pBuffer = (double*)malloc(lSamples * sizeof(double));
    ZeroMemory(pBuffer, lSamples * sizeof(double));

    GetBufferFrom(pBuffer, lStart, lSamples);
    strNewFile.Format("E:\\\\Doctorat\\\\%04d-%02d-%02d
        %02d-%02d-%02d-%04d.txt",
        sysTime.wYear,
        sysTime.wMonth,
        sysTime.wDay,
        sysTime.wHour,
        sysTime.wMinute,
        sysTime.wSecond,
        sysTime.wMilliseconds
    );

    if((pFile = fopen(strNewFile, "wt")) != NULL) {
        // scriere date în fișier
    }
}

```

```

        for (long IIndex = 0; IIndex < ISamples; IIndex++ ) {
            strWriteLine.Format("%.12f\n", pBuffer[IIndex]);
            fputs(strWriteLine, pFile);
        }
        fclose(pFile);
    }

    strNewFile.Format("E:\\Doctorat\\%04d-%02d-%02d
                    %02d-%02d-%02d-%04d.ext",
                    sysTime.wYear,
                    sysTime.wMonth,
                    sysTime.wDay,
                    sysTime.wHour,
                    sysTime.wMinute,
                    sysTime.wSecond,
                    sysTime.wMilliseconds
                    );

    if((pFile = fopen(strNewFile, "wt")) != NULL) {
        CString strHelp;
        strHelp.Format("Data si timp: %04d-%02d-%02d
                    %02d:%02d:%02d.%03d \n\r",
                    sysTime.wYear,
                    sysTime.wMonth,
                    sysTime.wDay,
                    sysTime.wHour,
                    sysTime.wMinute,
                    sysTime.wSecond,
                    sysTime.wMilliseconds
                    );
        fputs(strHelp, pFile);

        strHelp.Format("Locatie: \n" );
        fputs(strHelp, pFile);

        strHelp.Format("Comentarii: \n" );
        fputs(strHelp, pFile);
        strHelp.Format("rata esantionare: %.0f kSPS\n\r",
fSampleRate / 1000.);
        fputs(strHelp, pFile);

        strHelp.Format("frecventa: %.2f \tAverage Frequency: %.3f
\n", fFreq, fAvgFreq);
        fputs(strHelp, pFile);

        strHelp.Format("valoare efectiva: %.2f \tAverage RMS
Voltage: %.3f \n", fRMS, fAvgRMS);
        fputs(strHelp, pFile);

        strHelp.Format("valoare maxima: %.2f \tAverage Max
Voltage: %.3f \n", fMax, fAvgMax);

```

```

        fputs(strHelp, pFile);

        strHelp.Format("factor de creasta: %.2f \tAverage Crest
Factor: %.3f \n\n\r", fCF, fAvgCF);
        fputs(strHelp, pFile);
        fclose(pFile);
    }

    if (pBuffer) {
        free (pBuffer);
    }
}

void CCircularBuffer::SaveBuffer()
{
    double *pBuffer;
    FILE *pFile;
    CString strWriteLine, strDateTime, strNewFile;
    SYSTEMTIME sysTime;
    GetLocalTime(&sysTime);
    pBuffer = (double*)malloc(m_lQueSize * sizeof(double));
    ZeroMemory(pBuffer, m_lQueSize * sizeof(double));
    if(GetAllBuffer(pBuffer)) {
        strNewFile.Format("E:\\\\Doctorat\\\\%04d-%02d-%02d
                        %02d-%02d-%02d-%04d.txt",

            sysTime.wYear,
            sysTime.wMonth,
            sysTime.wDay,
            sysTime.wHour,
            sysTime.wMinute,
            sysTime.wSecond,
            sysTime.wMilliseconds
        );

        if((pFile = fopen(strNewFile, "at")) != NULL) {
            fseek(pFile, 0, SEEK_END);
            for (long lIndex = 0; lIndex < m_lQueSize; lIndex++) {
                strWriteLine.Format("%.12f\n", pBuffer[lIndex]);
                fputs(strWriteLine, pFile);
            }
            fclose(pFile);
        }
        if (pBuffer) {
            free (pBuffer);
        }
    }
    else {
        AfxMessageBox("Eroare salvare Buffer!");
    }
}

```

Anexa 3.4 Program pentru implementarea controlului grafic care afișează starea memoriei circulare

```

////////////////////////////////////
// CircularBufferCtrl.cpp
////////////////////////////////////
#include "stdafx.h"
#include "../Power Quality Monitoring System.h"
#include "CircularBufferCtrl.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

CCircularBufferCtrl::CCircularBufferCtrl()
{
    m_IReadPos = 0;
    m_IWritePos = 0;
    m_IReadLen = 0;
    m_IWriteLen = 0;
    m_BufferSize = 0;
    m_fReadStartAngle = 0;
    m_fReadEndAngle = 0;
    m_fWriteStartAngle = 0;
    m_fWriteEndAngle = 0;
}

CCircularBufferCtrl::~CCircularBufferCtrl()
{
}

BEGIN_MESSAGE_MAP(CCircularBufferCtrl, CStatic)
   //{{AFX_MSG_MAP(CCircularBufferCtrl)
    ON_WM_PAINT()
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()

void CCircularBufferCtrl::OnPaint()
{
    CPaintDC dc(this);
    CDC dcMemory;
    CBitmap bitmap;
    CBitmap* pOldBitmap = NULL;
    CRect rcDraw;

    if (this->IsWindowVisible()) {
        CRect rcClient;
        GetClientRect(&rcClient);
    }
}

```

```

CBrush brBkBrush,
        brReadBrush,
        brWriteBrush,
        brCircleBrush,
        brSecondCircle,
        *pOldBrush = NULL;
CPen *pOldPen = NULL,
        readPen,
        writePen,
        circlePen,
        intCirclePen,
        secCirclePen;
COLORREF crReadColor,
        crWriteColor,
        crCircleExtColor,
        crCircleIntColor,
        crBkColor,
        crSecCircle;

crReadColor = RGB(0,150,0);
crWriteColor = RGB(255,0,0);
crCircleExtColor = RGB(0,100,200);
crCircleIntColor = crCircleExtColor;
crSecCircle = RGB(0,0,0);
crBkColor = GetSysColor(COLOR_BTNFACE);

brBkBrush.CreateSolidBrush(crBkColor);
brReadBrush.CreateSolidBrush(crReadColor);
brWriteBrush.CreateSolidBrush(crWriteColor);
brCircleBrush.CreateSolidBrush(crCircleIntColor);
brSecondCircle.CreateSolidBrush(crSecCircle);
readPen.CreatePen(PS_SOLID, 1, crReadColor);
writePen.CreatePen(PS_SOLID, 1, crWriteColor);
circlePen.CreatePen(PS_SOLID, 3, crCircleExtColor);
intCirclePen.CreatePen(PS_SOLID, 1, crCircleIntColor);
secCirclePen.CreatePen(PS_SOLID, 1, crSecCircle);

dcMemory.CreateCompatibleDC(&dc);
dcMemory.SetBkColor(RGB(0,0,0));
bitmap.CreateCompatibleBitmap(&dc,
                               rcClient.Width(),
                               rcClient.Height());
pOldBitmap = dcMemory.SelectObject(&bitmap);
dcMemory.FillRect(rcClient, &brBkBrush);
CopyRect(&rcDraw, rcClient);
if (rcClient.Width() < rcClient.Height()) {
    rcDraw.DeflateRect(5,
                      (rcClient.Height() - rcClient.Width()) / 2,
                      5,
                      (rcClient.Height() - rcClient.Width()) / 2);
}

```

```
else {
    rcDraw.DeflateRect(
        (rcClient.Width() - rcClient.Height()) / 2,
        5,
        (rcClient.Width() - rcClient.Height()) / 2,
        5);
}

pOldPen = dcMemory.SelectObject(&writePen);
pOldBrush = dcMemory.SelectObject(&brWriteBrush);
dcMemory.BeginPath();
dcMemory.MoveTo(    rcDraw.CenterPoint().x,
                   rcDraw.CenterPoint().y);
dcMemory.AngleArc(  rcDraw.CenterPoint().x,
                   rcDraw.CenterPoint().y,
                   rcDraw.Height()/2,
                   m_fWriteStartAngle,
                   m_fWriteEndAngle);
dcMemory.LineTo(    rcDraw.CenterPoint().x,
                   rcDraw.CenterPoint().y);
dcMemory.EndPath();
dcMemory.StrokeAndFillPath();
dcMemory.SelectObject(pOldBrush);
dcMemory.SelectObject(pOldPen);

pOldPen = dcMemory.SelectObject(&readPen);
pOldBrush = dcMemory.SelectObject(&brReadBrush);
dcMemory.BeginPath();
dcMemory.MoveTo(    rcDraw.CenterPoint().x,
                   rcDraw.CenterPoint().y);
dcMemory.AngleArc(  rcDraw.CenterPoint().x,
                   rcDraw.CenterPoint().y,
                   rcDraw.Height()/3 ,
                   m_fReadStartAngle,
                   m_fReadEndAngle);
dcMemory.LineTo(    rcDraw.CenterPoint().x,
                   rcDraw.CenterPoint().y);
dcMemory.EndPath();
dcMemory.StrokeAndFillPath();
dcMemory.SelectObject(pOldBrush);
dcMemory.SelectObject(pOldPen);

pOldPen = dcMemory.SelectObject(&secCirclePen);
pOldBrush = dcMemory.SelectObject(&brSecondCircle);
dcMemory.BeginPath();
dcMemory.MoveTo(    rcDraw.CenterPoint().x,
                   rcDraw.CenterPoint().y);
dcMemory.AngleArc(  rcDraw.CenterPoint().x,
                   rcDraw.CenterPoint().y,
                   rcDraw.Height()/10,
                   0,
```

```

        360);
        dcMemory.LineTo(    rcDraw.CenterPoint().x,
                           rcDraw.CenterPoint().y);
        dcMemory.EndPath();
        dcMemory.StrokeAndFillPath();
        dcMemory.SelectObject(pOldBrush);
        dcMemory.SelectObject(pOldPen);
        dc.BitBlt(0, 0, rcClient.Width(), rcClient.Height(), &dcMemory, 0, 0,
SRCCOPY);
        dcMemory.SelectObject(pOldBitmap);
    }
}

void CCircularBufferCtrl::SetReadData(long IReadPos, long IReadLen)
{
    m_IReadPos = IReadPos;
    m_IReadLen = IReadLen;

    if (m_IBufferSize > 0) {
        m_fReadStartAngle =
            (float)((m_IReadPos * 360.00) / m_IBufferSize);
        m_fReadEndAngle =
            (float)(( m_IReadLen * 360.00) / m_IBufferSize);
    }
    Invalidate();
}

void CCircularBufferCtrl::SetWriteData(long IWritePos, long IWriteLen)
{
    m_IWritePos = IWritePos;
    m_IWriteLen = IWriteLen;

    if (m_IBufferSize > 0) {
        m_fWriteStartAngle =
            (float)((m_IWritePos * 360.00) / m_IBufferSize);
        m_fWriteEndAngle =
            (float)(( m_IWriteLen * 360.00) / m_IBufferSize);
    }
    Invalidate();
}

void CCircularBufferCtrl::SetBufferSize(long IBufferSize)
{
    m_IBufferSize = IBufferSize;
}

```

ANEXA 4. PROGRAME REPREZENTATIVE ELABORATE ÎN HTML ȘI PHP

Index.php – pagina de start a aplicației web

```
<?
    session_start();
    $disp="";
    if(isset($_GET['target']))
    {
        switch($_GET['target'])
        {
            case "home":
                $areaToLoad = "mainPageTemplate.php";
                include("mainpage.php");
                break;
            case "download":
                $areaToLoad = "download.php";
                include("mainpage.php");
                break;
            case "algoritmi":
                $areaToLoad = "algoritmi.php";
                include("mainpage.php");
                break;
            case "graph":
                $areaToLoad = "onlineData.php";
                include("mainpage.php");
                break;
            case "info":
                $areaToLoad = "info.php";
                include("mainpage.php");
                break;
            case "contact":
                $areaToLoad = "contact.php";
                include("mainpage.php");
                break;
            default:
                include("mainpage.php");
                break;
        }
    }
    else// if issset target
        include("mainpage.php");
?>
```


Mainpage.html – pagina principală a site-ului web

```

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
  <title>
    Monitorizarea calitatii energiei electrice
  </title>
  <link rel="stylesheet" href="css/style.css" type="text/css" media="screen" />
</head>
<body class = "body">
  <TABLE cellspacing=0 cellpadding=0 border=0 width="1000px" align="center"
bgcolor = "#cdbafe">
  <tr>
  <td>
    <table width="1000px" cellpadding="0" cellspacing="0" align="center"
border="0">
  <tr>
    <td valign="top" background = "./images/top.jpg" height = "150">
    </td>
  </tr>
  <tr>
    <td width = "1000px">
      <table border="0" cellpadding="0" cellspacing="0" width = "100%">
        <tr>
          <td valign="top">
            <?php include("include/topMenu.php");?>
          </td>
        </tr>
      </table>
    </td>
  </tr>
  <tr width="1000px">
    <td valign="top" width="1000px">
      <table width="1000px" border="0" cellpadding="0" cellspacing="0">
        <tr width="1000px">
          <td valign = "top">
            <table border="0" cellpadding="0" cellspacing="0">
              <tr>
                <td valign="top" width = "190px">
                  <? include("include/leftData.php"); ?>
                </td>
              </tr>
            </table>
          </td>
          <td align="center" width="800px" valign = "top">
            <table border = "0" cellspacing = "5px" cellpadding = "0px" width =
"790px">
              <tr>
                <td align = "center" valign = "top">
                  <?php

```

```
        if(isset($areaToLoad))
            include("include/".$areaToLoad);
        else
            include("include/mainPageTemplate.php");
    ?>
</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
<tr>
    <td colspan = "3">
        <? include("pageFooter.php");?>
    </td>
</tr>
</table>
</td>
</TR>
</table>
</body>
</html>
```

Disturbances.php – afișează perturbațiile detectate de către sistem în rețeaua electrică

```
<?php

global $FileName;
$lines = file($_GET["file"], FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES );
require_once ('jpgraph/jpgraph.php');
require_once ('jpgraph/jpgraph_line.php');

// Create the graph. These two calls are always required
$graph = new Graph(340,300);
$graph->SetScale("textlin");
$graph->SetMargin(25,10,30,30);
$graph->xaxis->SetTextTickInterval(10000,0);
$graph->xaxis->SetTextLabelInterval(2);
$graph->xaxis->HideLabels(true);
$graph->xaxis->SetLabelFormatString('My',true);

// Create the linear plot
$lineplot=new LinePlot($lines);
$lineplot->SetColor("blue");
```

```
// Add the plot to the graph
$graph->Add($lineplot);

// Display the graph
$graph->Stroke();
?>
```

Topmenu.php – afișează meniul orizontal al paginii web

```
<table border=0 width="1000px" cellpadding="0" cellspacing="0">
  <tr>
    <td width="100%" background = "./images/hr_green.jpg" height = "6px">
    </td>
  </tr>
  <tr>
    <td>
      <table border = "0px" cellspacing = "0px" cellpadding = "0px" width =
"1000px" height = "*" bgcolor = "3f1f75">
        <tr>
          <td>
            <div class="TopMenu">
              <a href=?php echo "index.php?target=home"?>>Home</a>
            </div>
          </td>
          <td>
            <div class="TopMenu">
              <a href=?php echo "index.php?target=graph"?>>Parametri</a>
            </div>
          </td>
          <td>
            <div class="TopMenu">
              <a href=?php echo "index.php?target=info"?>>Perturbatii</a>
            </div>
          </td>
          <td>
            <div class="TopMenu">
              <a href=?php echo "index.php?target=download"?>>
Statistici</a>
            </div>
          </td>
          <td>
            <div class="TopMenu">
              <a href=?php echo "index.php?target=algoritmi"?>>Prezentare
sistem</a>
            </div>
          </td>
          <td>
            <div class="TopMenu">
              <a href=?php echo "index.php?target=contact"?>>Contact</a>
```

```

        </div>
      </td>
    <td width = "200px">
      <div class="TopMenu">
        &nbsp;   
      </div>
    </td>
  </tr>
</table>
</td>
</tr>
<tr>
  <td width="100%" background = "./images/hr_green.jpg" height = "6px">
  </td>
</tr>
</table>

```

Mainpagetemplate.php – afișează conținutul paginii principale

```

<table border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td width = "*" valign="top" align = "center">
      <table border="0" cellpadding="0" cellspacing="0" height = "436" width =
"600px">
        <tr>
          <td width = "*" valign="top" >
            <center><h1> Sistem de monitorizare a calitatii energiei
electrice</h1></center>
            <p align="justify" style = "font-family: Verdana, Arial, Helvetica, sans-
serif; font-size: 13px; line-height:'150%';">
              Sistemul de monitorizare al calitatii energiei electrice a fost dezvoltat in
cadrul tezei de doctorat cu titlul: Contributii la detectarea si monitorizarea
perturbatiilor din rețelele de distributie a energiei electrice, elaborata de Ing. Ciprian
Dughir.
            </p>
          </td>
        </tr>
      </table>
    </td>
  </tr>
</table>

```

Leftdata.php – afișează un meniu vertical în partea stângă a paginii

```

<table border="0" cellpadding="0" cellspacing="0" width="*">
  <tr>
    <td valign="top">

```

```

    <? include("location.php");?>
  </td>
</tr>
<tr>
  <td valign="top">
    <? include("parameters.php");?>
  </td>
</tr>
</table>

```

Parameters.php - afișează parametrii curenți: tensiune efectivă și frecvență

```

<table border = "0" cellspacing = "5px" cellpadding = "0px" width = "200px">
  <tr>
    <td>
      <table border = "0" cellspacing = "0px" cellpadding = "0px" width = "190px">
        <tr>
          <td >
            <table border = "0px" cellspacing = "0px" cellpadding = "4px"
width="190" class = "radiatie_header" align="center">
              <tr>
                <td class="radiatie_header" width="100%" align="center"
colspan = "2"><b>PARAMETRI MASURATI</b>
              </tr>
            </table>
          </td>
        </tr>
        <tr>
          <td>
            <table border = "0px" cellspacing = "0px" cellpadding = "0px" width =
"190px">
              <tr>
                <td>
                  <table border = "0px" cellspacing = "0px" cellpadding = "4px"
width="190" class="radiatie" align="right">
                    <tr><td class="radiatie" width="130"
align="left"><b>Frecventa</b></td>
                      <td class="radiatie" width="60" align="right"><b>50.1
Hz</b></td></tr>
                    <tr><td class="radiatie" width="130" align="left"><b>Valoare
efectiva</b></td>
                      <td class="radiatie" width="60" align="right"><b>226
V</b></td></tr>
                  </table>
                </td>
              </tr>
            </table>
          </td>
        </tr>
      </table>
    </td>
  </tr>
</table>

```

```
</td>  
</tr>  
</table>  
</td>  
</tr>  
</table>
```