

CONTRIBUȚII LA ÎMBUNĂTĂȚIREA PERFORMANȚELOR ECHIPAMENTELOR INFORMATICE DE PROCES IMPLEMENTATE ÎN COMANDA SISTEMELOR HIDROENERGETICE ȘI DE NAVIGAȚIE

Teză destinată obținerii
titlului științific de doctor inginer
la
Universitatea "Politehnica" din Timișoara
în domeniul Energetică
de către

Ing. Alexandru Sorin JUDE

Conducător științific:	Prof.univ.dr.ing. Viorel NEGRU
Referenți științifici:	Prof.univ.dr.ing. Ioan FELEA Prof.univ.dr.ing. Ion MIRCEA Prof.univ.dr.ing. Mihai MOGA

Data susținerii tezei: 03.VII.2009

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, 2009

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

PREFAȚĂ

Teza de doctorat este rodul activității de mai mulți ani, desfășurată sub îndrumarea permanentă, a conducătorului științific, Prof.dr.ing. Viorel Negru, căruia îi adresez cele mai respectuoase și prețioase mulțumiri și pe această cale, atât pentru cultul rigurozității științifice pe care mi l-a imprimat, cât și pentru modul în care a știut să mă ghideze în toți acești ani. Contribuția sa a fost cu totul remarcabilă și sunt deosebit de încântat că am putut beneficia de această colaborare, începând din anii de facultate.

Lucrarea reprezintă o sinteză a rezultatelor de cercetare ale autorului, în perioada octombrie 2003 – martie 2009, în cadrul studiilor doctorale desfășurate în domeniul *Inginerie Energetică* la *Universitatea „Politehnica” din Timișoara*. În acești ani am fost permanent angrenat în activitatea unor echipe de cercetare din cadrul *Catedrei de Electroenergetică a Facultății de Electrotehnică și Electroenergetică*. Toate aceste activități de cercetare au avut o puternică influență asupra formării și orientării mele profesionale. În acest context, adresez alese mulțumiri d-lui. Prof. dr. ing. Ștefan Kilyeni, pentru ajutorul permanent acordat la sistematizarea materialului și pentru observațiile și sfaturile primite, întotdeauna constructive.

Pentru condițiile de lucru create în vederea elaborării și finalizării tezei, doresc să adresez mulțumiri călduroase Decanului Facultății de Electrotehnică și Electroenergetică, Prof.dr.ing. Petru Andea, fostului Decan al Facultății, Prof.dr.ing. Dumitru Toader, și șefului Catedrei de Electroenergetică, Prof.dr.ing. Flavius Dan Șurianu.

Mulțumesc cadrelor didactice din cadrul Universității „Politehnica” din Timișoara, care au contribuit la formarea și dezvoltarea mea profesională. În mod special doresc să-mi exprim gratitudinea față de colegii mei din cadrul Catedrei de Electroenergetică, care m-au încurajat și mi-au oferit un ajutor prețios.

Doresc să remarc, alăturând recunoștința mea, ajutorul de care am beneficiat din partea unor specialiști din cadrul S.C. HIDROTIM S.A. Timișoara, prin asigurarea suportului necesar pentru partea aplicativă a tezei.

Mulțumesc în mod deosebit, și pe această cale, membrilor comisiei de analiză a tezei de doctorat, Prof.dr.ing. Ion Mircea (Universitatea din Craiova), Prof.dr.ing. Ioan Felea (Universitatea din Oradea) și Prof.dr.ing. Mihai MOGA (Universitatea „Politehnica” din Timișoara), pentru atenția cu care au parcurs teza de doctorat, pentru criticile și aprecierile formulate, pentru sfaturile primite.

Timișoara, aprilie 2009

Alexandru Jude

Jude, Alexandru Sorin

Contribuții la îmbunătățirea performanțelor echipamentelor informatice de proces implementate în comanda sistemelor hidroenergetice și de navigație

Teze de doctorat ale UPT, Seria 3, Nr. 4, Editura Politehnica, 2009, 186 pagini, 67 figuri.

ISSN: 2066-5156

ISBN: 978-973-625-922-7

Cuvinte cheie: Echipamente informatice de proces,
Sisteme hidroenergetice și de navigație

Rezumat: Lucrarea de față abordează o tematică de mare actualitate, anume dispecerizarea și automatizarea unor aplicații industriale, în contextul rețehnologizării sistemelor de ecluzare de pe Dunăre. În prima parte a lucrării se descriu principalele mărimi măsurate în domeniul sistemului hidrotehnic, precum și noțiunile de bază legate de automatizări. O atenție deosebită se acordă Dispeceratului sistemului hidrotehnic. Se insistă asupra configurației acestui Dispecerat în varianta automatizată, precum și asupra ierarhizării elementelor constitutive. Este descrisă și stabilirea protocoalelor de comunicație între acestea. Este surprinsă structura hardware a întregului ansamblu, în ipoteza conducerii procesului printr-un sistem tip SCADA. Bazele de date utilizate în conducerea acestor procese, împreună cu algoritmi utilizați pentru filtrarea informațiilor conținute în bazele de date, sunt detaliați ulterior. Conceperea unui mod eficient de interfațare om - mașină este unul din obiectivele declarate ale întregului demers. Sunt conținute și o parte din secvențele de program utilizate pentru conceperea acestei interfețe, din punct de vedere al aplicației software care rulează în sistem. În întreaga lucrare sunt prezentate aplicații practice și originale, implementate în cazul Sistemului Hidroenergetic și de Navigație Porțile de Fier II.

CUPRINS

Prefață	3
Cuprins	5
Lista de figuri	9
1. Introducere	11
1.1. Obiectul și actualitatea temei	11
1.2. Obiectivele și structura tezei	13
1.3. Realizarea și oportunitatea tezei	15
2. Dispecerat de conducere a sistemelor hidrotehnice pe calculator.	
Aspecte tehnico – economice	17
2.1. Sisteme informatice de proces. Considerații generale	17
2.1.1. Considerații hardware	17
2.1.2. Considerații software.....	18
2.1.3. Aplicații distribuite programate în timp real	19
2.1.3.1. Elemente conceptuale ale sistemelor distribuite	19
2.1.3.2. Conceperea și programarea aplicațiilor distribuite.....	19
2.1.4. Sisteme de prelucrare distribuita a datelor. Obiective generale.....	21
2.2. Probleme legate de urmărirea și conducerea proceselor tehnologice	22
2.2.1. Conducerea proceselor tehnologice. Introducere	22
2.2.2. Sisteme dinamice	26
2.2.3. Niveluri de automatizare în conducerea proceselor industriale	29
2.2.3.1. Structuri ierarhizate de conducere	29
2.2.3.2. Nivelul de achiziție, prelucrare și reglare	30
2.2.3.3. Nivelul de identificare	31
2.2.3.4. Nivelul de optimizare	32
2.2.4. Descentralizare și coordonare	
în conducerea proceselor industriale	33
2.2.5. Integrarea în conducerea proceselor industriale	34
2.3. Probleme de optimizare în automatizările industriale	36
2.3.1. Optimizări staționare.....	36
2.3.2. Optimizări dinamice	37
2.3.3. Formularea matematica a problemelor de optimizare	39
2.3.3.1. Probleme de terminologie	39
2.3.3.2. Formularea optimizării prin intermediul	
spațiilor vectoriale.....	40
2.3.3.3. Metode de calcul pentru optimizarea fără restricții.....	42
2.4. Funcțiile unui sistem informatic de proces pentru noduri hidrotehnice	43
2.4.1. Funcțiile de bază	43
2.4.1.1. Funcția de teleurmărire	43
2.4.1.2. Funcția de telegestiune	45
2.4.1.3. Funcția de teleconducere.....	46

2.4.2. Funcțiile speciale	47
2.4.2.1. Funcția de inginer dispecer	48
2.4.2.2. Funcția de analiză și supervizare	48
2.4.2.3. Funcția de inginer de sistem și ghid operator AMC.....	49
2.5. Structura hardware a unui sistem informatic de proces (dispecerat hidrotehnic)	49
2.6. Concluzii	52
2.7. Contribuții personale	54
3. Sisteme de automatizare.....	55
3.1. Interfațarea Om – Sistem de automatizare. Analiza și conceperea interfețelor	55
3.1.1. Analiza sistemului om-mașină	55
3.1.1.1. Modalități de concepere a acestor interferențe	56
3.1.1.2 Evaluarea sistemului realizat.....	57
3.1.1.3. Modele simplificate utilizate pentru conceperea de interfețe	57
3.1.2. Cooperarea om-mașină. Principii.....	58
3.1.2.1. Cooperarea pe verticală	60
3.1.2.1.1. Supravegherea funcționării normale și detectarea greșelilor	61
3.1.2.1.2. Modul de funcționare anormală a procedurii	62
3.1.2.1.3. Justificarea raționamentului sistemului expert....	62
3.1.2.1.4. Rezolvarea conflictelor	63
3.1.2.1.5. Evoluția către cooperarea verticală interactivă	64
3.1.2.2. Cooperarea orizontală: repartiția dinamică a sarcinilor	64
3.1.2.2.1. Principii și condiții de punere în funcțiune	64
3.1.3. Sisteme expert. Definiție, modalități de construire	67
3.1.3.1. Categoriile de aplicații	67
3.1.3.2. Structura constructivă a unui sistem expert	68
3.1.3.3. Etapele realizării unui sistem expert.....	69
3.2. SCADA, o soluție pentru monitorizarea sistemelor hidrotehnice	71
3.2.1. Funcțiile de bază ale dispecerului SCADA.....	72
3.2.2. Structura software aleasă	76
3.3. Structura hardware a unui sistem informatic de proces	78
3.4. Integrarea, extinderea și upgradarea sistemului SCADA.....	80
3.4.1. Descrierea modului de implementare etapizată a sistemului.....	80
3.4.2. Cooperarea la scrierea unui nou driver în cazul cuplării cu echipamente / sisteme noi.....	81
3.4.3. Asigurarea sincronizării echipamentelor din sistem	82
3.4.4. Extinderea și upgradarea sistemului (hardware, software, comunicație).....	82
3.5. Concluzii	84
3.6. Contribuții personale	86

4. Metode și posibilități de proiectare a bazelor de date pentru sisteme informatice de proces. Filtrarea informațiilor la nivelul dispeceratului hidrotehnic	87
4.1. Configurația bazelor de date	87
4.1.1. Tipuri de baze de date.....	88
4.1.2. Sisteme de gestiune a bazelor de date.....	89
4.2. Aspecte tehnice privind structura bazelor de date pentru aplicații în timp real.....	92
4.3. Interfața om - mașină a SGBD. Tipuri de ferestre de interfațare.....	96
4.3.1. Hărți sinoptice.....	96
4.3.2. Ferestre cu grafice de evoluție în timp	97
4.3.3. Ferestre de evenimente și ferestre de raportare	99
4.3.4. Fereastra de alarmare	105
4.4. Posibilități de modernizare a bazei de date.....	107
4.5. Secvențe de program pentru realizarea interfeței om-mașină.....	110
4.6. Concluzii	135
4.7. Contribuții personale	136
5. Sistem SCADA implementat la ecluza principală română de la Sistemul Hidroenergetic și de Navigație Porțile de Fier II	137
5.1. Prezentarea generală a echipamentelor ecluzei	137
5.1.1. Comanda din turn în mod automat – ciclul complet de ecluzare.....	139
5.1.2. Comanda din turn în mod automat pentru fiecare echipament în parte	140
5.2. Considerații generale privind sistemul de monitorizare	140
5.3. Stări și semnale necesare monitorizării.....	156
5.4. Concluzii	161
5.5. Contribuții personale	162
6. Concluzii și perspective	163
6.1. Rezultatele cercetării	163
6.2. Concluziile principale	164
6.3. Contribuțiile personale și originale	165
6.4. Perspectivele continuării studiului	167
Anexe	169
Anexa 1. Schema bloc pentru monitorizarea ecluzei române PdF II.....	169
Anexa 2. Schema electrică desfășurată	170
Anexa 3. Schema tehnologică cu elemente de automatizare CAP AMONTE	171
Anexa 4. Schema bloc sistem de comandă CAP AMONTE.....	172
Anexa 5. Schema tehnologică cu elemente de automatizare CAP AMONTE – Continuare I.....	173
Anexa 6. Schema tehnologică cu elemente de automatizare CAP AMONTE – Continuare II.....	174

8 Cuprins

Anexa 7. Schema tehnologică cu elemente de automatizare CAP AMONTE – Continuare III	175
Anexa 8. Schema tehnologică cu elemente de automatizare CAP AVAL	176
Anexa 9. Schema tehnologică cu elemente de automatizare CAP AVAL- Continuare I	177
Anexa 10. Schema tehnologică cu elemente de automatizare CAP AVAL- Continuare II	178
Anexa 11. Schema tehnologică cu elemente de automatizare CAP AVAL- Continuare III	179
Anexa 12. Schema bloc comandă PPS	180
Anexa 13. Schema bloc comandă PPA	181
Anexa 14. Schema bloc a unui dulap de conexiuni.....	182
Bibliografie	183
Sinteza privind lucrările proprii	186

LISTA DE FIGURI

Nr. crt.	Numărul figurii	Titlul figurii
1.	Fig. 2.1.	Reprezentarea schematică a unui proces, cu evidențierea debitelor masice și energetice de intrare și de ieșire
2.	Fig. 2.2.	Reprezentarea unui proces, cu scoaterea în evidență a mărimilor de intrare, a mărimilor interne și a mărimilor de ieșire
3.	Fig. 2.3.	Schema bloc a unui proces tehnologic, prevăzut cu elemente de interfață cu operatorul uman sau cu mijloacele de conducere
4.	Fig. 2.4.	Evidențierea unei vehiculări, în circuit închis, a informațiilor și acțiunilor, într-o conducere manuală sau automată a procesului
5.	Fig. 2.5.	Funcție de comandă tip scară, întâlnită în sistemele de reglare discretă a unor procese continue
6.	Fig. 2.6.	Structura ierarhizată pentru conducerea numerică
7.	Fig. 2.7.	Structura unui sistem de dimensiuni mari
8.	Fig. 2.8.	Elementele unui sistem de reglare automată
9.	Fig. 2.9.	Schema simplificată a unui sistem de reglare automată
10.	Fig. 2.10.	Răspunsul sistemului de reglare automată la variația treaptă a mărimii de referință
11.	Fig. 2.11.	Sistem de reglare automată multivariabil
12.	Fig. 2.12.	Structura hardware a unui sistem informatic de proces
13.	Fig. 3.1.	Sistem de supervizare
14.	Fig. 3.2.	Tipuri de comportament ale operatorului uman
15.	Fig. 3.3.	Principiul de cooperare om - mașină pe verticală
16.	Fig. 3.4.	Principiul de cooperare om - mașină pe verticală
17.	Fig. 3.5.	Interfața dintre om și sistemul expert
18.	Fig. 3.6.	Sinteza de diferite imagini constituind interfața dintre operator și sistemul expert
19.	Fig. 3.7.	Principii de repartizare dinamică a sarcinilor de supervizare
20.	Fig. 3.8.	Cooperarea Om - Mașină
21.	Fig. 3.9.	Structura unui sistem expert
22.	Fig. 3.10.	Exemplu de arhitectură a RTU
23.	Fig. 3.11.	Comparație între diverse tipuri de sisteme SCADA
24.	Fig. 4.1.	Bază de date tabelară a aplicației de la Porțile de Fier II
25.	Fig. 4.2.	Selectarea elementelor afișabile dintr-o bază de date
26.	Fig. 4.3.	Hartă sinoptică a mecanismului de vane aferent ecluzei
27.	Fig. 4.4.	Grafic de evoluție în timp
28.	Fig. 4.5.	Fereastră de evenimente și raportare
29.	Fig. 4.6.	Fereastră de alarmare
30.	Fig. 4.7.	Exemplu de grafic al istoricului reglării
31.	Fig. 5.1.	Ecluza românească de la Porțile de Fier II
32.	Fig. 5.2.	Sistemul Hidroenergetic și de Navigație de la Porțile de Fier II
33.	Fig. 5.3.	Schema bloc a echipamentului de comandă
34.	Fig. 5.4.	Ecluzare. Etapa I
35.	Fig. 5.5.	Ecluzare. Etapa II

Nr. crt.	Numărul figurii	Titlul figurii
36.	Fig. 5.6.	Ecluzare. Etapa III
37.	Fig. 5.7.	Ecluzare. Etapa IV
38.	Fig. 5.8.	Ecluzare. Etapa V
39.	Fig. 5.9.	Ecluzare. Etapa VI
40.	Fig. 5.10.	Ecluzare. Etapa VII
41.	Fig. 5.11.	Ecluzare. Etapa VIII
42.	Fig. 5.12.	Ecluzare. Etapa IX
43.	Fig. 5.13.	Avarie în procesul de ecluzare
44.	Fig. 5.14.	Schema automatizării pentru poarta buscată
45.	Fig. 5.15.	Automatizarea pentru poarta buscată
46.	Fig. 5.16.	Automatizarea vanelor galerii
47.	Fig. 5.17.	Schema automatizare VANA 1 Amonte
48.	Fig. 5.18.	Schema automatizare VANA 2 Amonte
49.	Fig. 5.19.	Schema de automatizare VANA 1 Aval
50.	Fig. 5.20.	Schema de automatizare VANA 2 Aval
51.	Fig. 5.21.	Schema sinoptică a automatizării protecției porții buscate
52.	Fig. 5.22.	Instalația hidraulică exterioară
53.	Fig. 5.23.	Dulapul instalației de automatizare
54.	Anexa 1	Schema bloc pentru monitorizarea ecluzei române Pdf II
55.	Anexa 2	Schema electrică desfășurată
56.	Anexa 3	Schema tehnologică cu elemente de automatizare CAP AMONTE
57.	Anexa 4	Schema bloc sistem de comandă CAP AMONTE
58.	Anexa 5	Schema tehnologică cu elemente de automatizare CAP AMONTE – Continuare I
59.	Anexa 6	Schema tehnologică cu elemente de automatizare CAP AMONTE – Continuare II
60.	Anexa 7	Schema tehnologică cu elemente de automatizare CAP AMONTE – Continuare III
61.	Anexa 8	Schema tehnologică cu elemente de automatizare CAP AVAL
62.	Anexa 9	Schema tehnologică cu elemente de automatizare CAP AVAL- Continuare I
63.	Anexa 10	Schema tehnologică cu elemente de automatizare CAP AVAL- Continuare II
64.	Anexa 11	Schema tehnologică cu elemente de automatizare CAP AVAL- Continuare III
65.	Anexa 12	Schema bloc comandă PPS
66.	Anexa 13	Schema bloc comandă PPA
67.	Anexa 14	Schema bloc a unui dulap de conexiuni

1. INTRODUCERE

1.1. Obiectul și actualitatea temei

Prin așezarea sa geografică, România reprezintă o zonă de intersecție a mai multor magistrale de transport, care leagă nordul de sudul Europei și vestul de estul acesteia. Pe de altă parte, rețeaua de transport din România asigură legătura între rețeaua de transport comunitară și rețeaua de transport a statelor necomunitare vecine din Europa de est și Asia.

România are stabilite liniile directe ale căilor de comunicație de interes european și național prin Legea nr.71/1996 privind aprobarea Planului Național de Amenajarea Teritoriului secțiunea I „ Căi de comunicație”, ca suport al dezvoltării complexe și durabile a teritoriului, inclusiv al dezvoltării regionale, reprezentând totodată contribuția specifică a țării noastre la dezvoltarea spațiului european și premiza înscrierii în dinamica dezvoltării economico-sociale europene.

Rețeaua principală de căi și canale navigabile interioare este situată în partea de sud a țării și cuprinde Dunărea cu brațele secundare navigabile și canalele navigabile Dunăre-Marea Neagră și Poarta Albă-Midia Năvodari. În interior, rețeaua de căi navigabile este foarte redusă și dispersată, are un caracter local, cuprinde lacuri naturale și lacuri ale amenajărilor hidrotehnice și un este folosită pentru agrement și mic trafic local de mărfuri.

Fluviul Dunărea, cale navigabilă internațională, pe sectorul românesc, de la intrarea în țară și până la vărsarea în Marea Neagră prin Canalul Sulina, are o lungime de 1.075 km, dintre care cca. 170 km, între Brăila și Marea Neagră, asigură condiții tehnice pentru accesul navelor maritime. Datorită regimului natural de scurgere al fluviului se impun măsuri de îmbunătățire a condițiilor de navigație pe Dunăre prin implementarea de programe care să asigure exploatarea eficientă în condiții de siguranță a sectorului maritim al Dunării precum și asigurarea navigației permanente pe Canalul Sulina prin refacerea și apărarea malurilor canalului și implementarea unui sistem de măsurători topohidrografice și semnalizare pe sectorul românesc al Dunării. Totodată, pentru creșterea siguranței navigației pe sectorul românesc al Dunării se implementează sistemul de urmărire și management al traficului de nave.

În spiritul preocupărilor legate de creșterea siguranței în exploatarea a sectorului navigabil al Dunării, se înscriu și contribuțiile aduse de către această lucrare.

Sectorul fluvial al Dunării, de la intrarea în țară până la sistemul hidroenergetic și de navigație Porțile de Fier II (km 863) ce se află în regim amenajat de scurgere cu condiții optime pentru navigație, iar în aval până la Brăila este în regim natural de scurgere cu condiții dificile de navigație pe unele porțiuni în perioadele cu nivel scăzut al apelor. Canalele navigabile Dunăre-Marea Neagră (64,4 km) și Poarta Albă-Midia Năvodari (27,5 km +5,5 km) sunt realizate artificial și sunt deschise navigației internaționale și corespund clasei a VI- a de importanță a căilor navigabile și respectiv, clasei a V-a (conform clasificării CEE-ONU) și asigură legătura optimă a Dunării la portul Constanța.

Pe căile navigabile interioare, România dispune de un număr de 30 de porturi și puncte de încărcare amenajate cu o capacitate totală de trafic de 52 mil. tone/an. Dintre acestea, care fac parte din rețeaua TEN, porturile Brăila, Galați, Tulcea și Sulina sunt porturi fluvio-maritime și au o capacitate totală de trafic de cca. 34 mil. tone/an și au caracteristici tehnice ce permit accesul permanent al navelor maritime cu o capacitate de până la 25.000 tdw, 180 m lungime și cu un pescaj de 7,0 m. În condiții speciale pescajul poate fi depășit.

Datorită amplasării pe rețea trans-europeană de transport, Dunărea are potențial pentru dezvoltarea sistemului combinat de transport, dezvoltarea turistică din zona adiacentă Dunării și din Delta Dunării și îmbunătățirea exploatarea actuale a porturilor fluviale. În acest sens se promovează proiecte care au în vedere tehnologiile de operare specifice porturilor și proiecte privind protecția mediului pe Dunăre și în porturi.

În anul 2004, transportul fluvial a fost realizat în proporție de 95,6% cu nave private. Traficul de mărfuri pe căile navigabile românești a fost de 14.600,5 mii tone, respectiv de 4,3 milioane tone-km, înregistrând un nivel destul de ridicat față de noile state membre (2 milioane tone în Cehia, Ungaria și Slovacia sau 10 milioane tone în Polonia, în anul 2001), dar foarte scăzut față de alte state membre (128 milioane tone în Belgia, 329 milioane tone în Olanda sau 91 milioane tone în Finlanda). Traficul portuar de marfă în anul 2004 a fost de 71.742 mii tone, din care 43,5% reprezintă traficul fluvial, respectiv 31.211 mii tone.

Dunărea, până la vărsarea în Marea Neagră pe brațul Sulina și Canalul Dunăre - Marea Neagră sunt parte integrantă a Coridorului de Transport Pan-European nr. VII și oferă României și celorlalte state dunărene oportunități majore pentru dezvoltarea sectorului de transport pe apă. În acest sens, sunt în curs de desfășurare sau în program lucrări și proiecte pentru îmbunătățirea condițiilor de navigație pe Dunăre și pentru exploatarea în siguranță a canalelor navigabile și a nodurilor hidrotehnice specifice navigației.

Sistemul Porțile de Fier II este, după Porțile de Fier I, una dintre cele mai mari amenajări hidroenergetice din Europa ca producție de energie electrică pe de o parte și ca ansamblu arhitectural de mari proporții pe de altă parte. Sectorul amenajat prin realizarea Sistemului Porțile de Fier II se întinde în lungul Dunării de la Sistemul Porțile de Fier I (km D 942 + 950), până la barajul Sistemului Porțile de Fier II (km D 862 + 800), pe o distanță de cca. 80 km și în aval până în zona gurii râului Timoc (km D 846) până unde se simte influența sistemului.

Construcția principală a amenajării o constituie centralele hidroelectrice de bază, ambele centrale, română și iugoslavă, având o tratare arhitecturală identică. Centralele de bază au fost puse în funcțiune la capacitatea proiectată în anul 1986, centrala suplimentară română în anul 1994, iar centrala suplimentară iugoslavă în anul 2000.

Obiectul principal al Sistemului Porțile de Fier II cuprinde uvrajele pentru bararea fluviului și ridicarea retenției, precum și pe cele pentru producerea energiei electrice și asigurarea navigației, acestea fiind dispuse după cum urmează:

- pe brațul principal al Dunării la km D 862 + 800 sunt amplasate dinspre malul stâng spre cel drept două centrale, fiecare cu câte 8 agregate, un baraj cu 7 câmpuri deversoare, o ecluză pentru navigație de 34 m lățime și o centrală suplimentară cu 2 agregate; pe brațul secundar Gogoșu, la km D 875 + 000, este

- amplasat în mijlocul albiei un baraj cu 7 câmpuri deversoare și alăturat pe partea dreaptă a brațului este amplasată o centrală suplimentară cu 2 agregate.
- Pe canalul natural de pe insula Ostrovul Mare (care pornește de pe brațul principal de la km D 865 + 000 și debușează pe brațul Gogoșu la km D 863 + 000) este amplasată o ecluză pentru navigație de 34 m lățime și adiacent acesteia, o ecluză de siguranță de dimensiuni mai mici (14 m lățime).

Amplasamentul Sistemului Porțile de Fier II a fost adoptat ca urmare a multiplelor avantaje energetice, dintre care amintim:

- se asigură o simetrie de proprietate a celor două state riverane, România și fosta Iugoslavie, asupra uvrăjelor din componența obiectului principal. În acest scop, frontiera de stat care la începutul lucrărilor s-a aflat pe partea dreaptă a brațului principal a fost deplasată local spre malul stâng până la mijlocul centralelor principale. Astfel, pe teritoriul fiecărui stat există o centrală cu 8 hidroagregate, un baraj deversor cu 7 câmpuri, o ecluză pentru navigație de 34 m lățime și o centrală suplimentară cu 2 hidroagregate. Exploatarea energetică a Sistemului Porțile de Fier se face în conformitate cu prevederile *Convenției de exploatare româno-iugoslave* încheiate în anul 1998;
- prin dispunerea celor 2 centrale de bază pe brațul principal al Dunării (care este mai larg în raport cu brațul Gogoșu), se asigură căderi de apă mai mari pentru cele 16 hidroagregate din aceste centrale, ceea ce din punct de vedere energetic este favorabil;
- prin dispunerea celor două baraje deversoare în mijlocul albiilor celor două brațe, Dunărea și Gogoșu, se asigură condiții bune pentru evacuarea apelor mari și a ghețurilor, folosind optim capacitatea de scurgere a acestor brațe;
- prin dispunerea la o distanță mare, una de cealaltă, a celor două ecluze principale se asigură un grad înalt de siguranță pentru navigație, care se poate face pe două șenale complet separate, fiecare dintre ele pe câte un braț al Dunării.

Cele două centrale de bază cu 8 hidroagregate fiecare de pe brațul principal al Dunării și cele două centrale suplimentare cu 2 hidroagregate fiecare de pe brațul principal și de pe brațul Gogoșu sunt identice ca dispoziție interioară și soluții constructive și sunt echipate cu același tip de agregat (bulb orizontal de 27 MW) și aceleași tipuri de echipamente hidromecanice. Caracteristicile generale sunt identice pentru cele patru centrale. În centrale sunt prevăzute toate echipamentele auxiliare, gospodăriile și instalațiile necesare pentru funcționare și siguranță, instalații electrice pentru consum propriu și iluminat.

Modernizarea infrastructurii ecluzelor Sistemului Hidroenergetic și de Navigație de la Porțile de Fier II a constituit o preocupare constantă a autorităților navale române. În acest sens, contribuțiile aduse de către autor, prin intermediul S.C. HIDROTIM S.A. au fost esențiale în scopul definitivării soluției tehnice de automatizare a nodului hidrotehnic și de navigație.

1.2. Obiectivele și structura tezei

Obiectivul principal al acestei lucrări constă în conceperea unor sisteme moderne destinate automatizării, monitorizării și dispecerizării activităților de ecluzare, care să conducă la o mai bună funcționare a ansamblului, la optimizarea fluxurilor de transport, în condițiile unui trafic cât mai sigur.

Se are în vedere creșterea fiabilității întregului ansamblu care concurează la buna desfășurare a activității de navigație, dar fără a lua în considerare modificarea infrastructurii (porți, vane, pompe, actuatoare etc.).

Aplicarea acestor sisteme trebuie să conducă la rezultate imediate concretizate în creșterea randamentului ansamblului și identificarea rapidă și eficientă a defecțiunilor sau a problemelor în exploatare. Aspectul economic este cel esențial, chiar dacă la început o astfel de investiție înseamnă niște costuri importante, ea se amortizează în timp scurt datorită creșterii randamentului activității în ansamblu.

Întrebarea centrală la care se încearcă a se găsi un răspuns în paginile acestei teze este: "Ce trebuie făcut pentru a eficientiza activitatea de ecluzare, în contextul menținerii aceleiași infrastructuri existente?".

Întregul demers teoretic și aplicativ este subordonat acestui imperativ și este realizat din perspectiva celui care concepe sistemul de automatizare, dar în directă legătură cu beneficiarul acestuia.

S-a încercat a se evita o separare clară între conținutul teoretic și cel aplicativ, în scopul de a evidenția interdependența dintre cele două. Fiecare capitol al tezei a fost gândit ca să ofere, alături de un breviar teoretic și rezultatele cercetării aplicative în domeniu.

Problematika abordată, atât prin aspectele teoretice cât și prin cele aplicative, este subordonată următoarelor obiective majore:

- Formularea problemelor care intervin în funcționarea sistemelor hidrotehnice;
- Determinarea principalilor parametri care intervin în funcționarea acestora;
- Analiza modului în care se măsoară acești parametri;
- Sinteza bibliografică a principalelor tipuri de transductoare întâlnite în hidrotehnică;
- Alegerea metodelor și mijloacelor de măsurare;
- Exemplificarea unor soluții tehnice originale destinate măsurării sau achiziției mărimilor de interes;
- Caracterizarea performanțelor unor soluții de măsurare, atât în regim staționar cât și dinamic;
- Identificarea funcțiilor și a structurii unui dispecerat pentru activitatea hidrotehnică;
- Implementarea unui sistem integrat de automatizare, măsură și control a funcționării sistemului hidrotehnic;
- Descrierea structurii și a configurației hardware a unui dispecerat automatizat;
- Conceperea bazelor de date cu care operează software-ul implementat într-un sistem complex de automatizare măsură și control a funcționării sistemului hidrotehnic;
- Filtrarea informațiilor obținute din proces, prin utilizarea unor algoritmi performanți;
- Elaborarea unui software performant care să opereze cu acești algoritmi, în scopul creării unui dialog eficient și simplu între operatorul uman și sistemul propriu-zis;
- Optimizarea informațiilor culese din proces și a sistemului în ansamblu;
- Creșterea eficienței prelucrării informației;
- Prezentarea unor aplicații concrete, care vizează astfel de sisteme de automatizare, măsură și control, concepute de către autor și implementate la diverși beneficiari în țară;
- Propunerea unor soluții tehnice originale vizând domeniul sistemelor hidrotehnice;
- Evidențierea aspectelor economice care decurg din implementarea acestor rezultate ale cercetării aplicative;
- Expunerea unor rezultate obținute în urma implementării acestor sisteme;

- Determinarea unor posibile direcții de continuare a cercetării aplicative în domeniu, direcții justificate în primul rând de considerente economice.

În structurarea lucrării s-a urmărit expunerea clară a fiecărei probleme, înșiruirea logică, nu în ultimul rând trecerea gradată de la simplu la complex, respectiv divergența și convergența ideilor.

Lucrarea este structurată pe 6 capitole (incluzând Introducerea și Concluziile), Referințele bibliografice, precum și Anexele principale.

În prezentarea lucrării s-a optat pentru varianta precizării semnificației notațiilor și simbolurilor la prima utilizare a acestora sau ori de câte ori s-a considerat necesar, astfel încât nu mai este necesară o listă separată pentru simbolurile utilizate.

În prima parte a Capitolului 2 se descriu principalele mărimi măsurate în domeniul sistemului hidrotehnic, precum și noțiunile de bază legate de automatizări.

Capitolul 3 este dedicat Dispeceratului sistemului hidrotehnic. Se insistă asupra configurației acestui Dispecerat în varianta automatizată, precum și asupra ierarhizării elementelor constituente. Este descrisă și stabilirea protocoalelor de comunicație între acestea. Este surprinsă structura hardware a întregului ansamblu, în ipoteza conducerii procesului printr-un sistem tip SCADA.

Bazele de date utilizate în conducerea acestor procese sunt prezentate succint în Capitolul 4. În acest capitol se acordă o atenție deosebită și algoritmilor utilizați pentru filtrarea informațiilor conținute în bazele de date. Conceperea unui mod eficient de interfațare om - mașină este unul din obiectivele declarate ale întregului demers. Acest capitol conține și o parte din secvențele de program utilizate pentru conceperea acestei interfețe, din punct de vedere al aplicației software care rulează în sistem.

Capitolul 5 este rezervat în întregime descrierii unor aplicații practice și originale, implementate în cazul Sistemului Hidroenergetic și de Navigație Porțile de Fier II.

Fiecare capitol se încheie cu un subcapitol destinat concluziilor specifice precum și cu un subcapitol care evidențiază contribuțiile personale și, după caz originale, ale autorului.

Capitolul 6 sintetizează concluziile specifice fiecărui capitol anterior și evidențiază contribuțiile personale sau originale ale autorului. Sunt descrise și posibilitățile de continuare a studiului.

Indiferent de capitolul prezentat, întregul demers este subordonat obiectivului principal al lucrării, anume creșterea fiabilității și siguranței în exploatarea sistemelor hidroenergetice și de navigație, în condițiile utilizării unor sisteme performante de automatizare, monitorizare și comandă a circuitelor electrice și hidraulice implicate.

1.3. Realizarea și oportunitatea tezei

Această teză este rezultatul cercetărilor și studiilor efectuate de către autor pe parcursul întregii sale cariere. Autorul s-a implicat direct în proiectarea și realizarea a numeroase astfel de sisteme pe tot parcursul ultimilor ani, la diverși beneficiari atât din România, cât și din străinătate. Este una dintre numeroasele soluții la cheie oferite partenerilor economici, fiind una dintre cele mai reprezentative.

Ea reflectă și unele preocupări actuale ale utilizatorilor de sisteme de automatizare hidrotehnice. Beneficiarii acestor sisteme contribuie activ la materializarea soluțiilor tehnice propuse, atât prin definirea problemelor cât și prin sprijinul acordat în asimilarea acestora și nu în ultimul rând prin observațiile și sugestiile privind comportamentul în exploatare al ansamblului.

Finalitatea practică și justificarea economică este scopul fundamental al oricărei cercetări aplicative, deci și a întregului demers al acestei lucrări. Și în acest caz se respectă principiul aplicării în economie a rezultatelor cercetării, pornind de la cerințele mediului economic și industrial.

Lucrarea de față se înscrie și în rândul preocupărilor actuale ale specialiștilor de la Catedra de Electroenergetică a Facultății de Electrotehnică și Electroenergetică din cadrul Universității „Politehnica” din Timișoara, vizând îmbunătățirea siguranței în exploatare a oricăror instalații tehnologice precum și automatizarea proceselor tehnologice și energetice în ansamblul lor.

2. DISPECERAT DE CONDUCERE A SISTEMELOR HIDROTEHNICE PE CALCULATOR. ASPECTE TEHNICO – ECONOMICE

2.1. Sisteme informatice de proces. Considerații generale

2.1.1. Considerații hardware

Funcționarea în condiții optime a unui nod hidrotehnic și de navigație presupune urmărirea și corectarea în timp real a principalilor parametri de debit, presiune, temperatură, nivel, stare pompe, agregate, porți etc.

Sistemul informatic de teleurmărire și telegestiune poate fi implementat pe orice tip de sistem hidrotehnic, indiferent de complexitatea acestuia.

Acest tip de dispecer reprezintă o soluție modernă, cu posibilitatea de a fi verificat și autorizat.

Autorul a conceput, în premieră în România, pentru S.C. HIDROTIM S.A. numeroase aplicații de tip dispecer (sau automatizare) a nodurilor hidrotehnice.

Implementarea unui astfel de sistem informatic de proces s-a făcut pornind de la următoarele considerente de ordin tehnic:

- urmărirea și conducerea procesului de navigație, de complexitate medie sau mare, implică trei nivele logice de ierarhizare:

1. la nivelul inferior, numit și „nivelul 0”, procesul este văzut ca fiind format din subprocese în care se evidențiază mărimi de interacțiune. Descompunerea în subprocese se face după criteriile funcțional-tehnologice, acest tip de descompunere conducând implicit la minimizarea complexității mărimilor de interacțiune.

La acest nivel are loc automatizarea procesului, ceea ce presupune realizarea unor funcții de reglare a principalelor variabile caracteristice.

Optimizarea (calitatea reglării) este determinată de:

- performanțele traductorilor utilizați pentru obținerea valorii măsurate a mărimii ce se reglează;
- cunoașterea modelului (funcția de optimizare, funcția criteriu) a procesului condus.

Eficiența intervenției operatorului în proces (calitatea reglării), este dependența într-o foarte mare măsură de modul în care acesta are acces la procesul de care se ocupă.

Modalitatea de oferire a informațiilor despre proces, comoditatea cu care se poate acționa asupra acestuia în vederea influențării rapide a comportării, absența senzației întrepătrunderii unui mediu necunoscut între operator și proces, constituie elementele de baza ale unui sistem informatic de proces.

Din punct de vedere hardware, la acest nivel sunt plasați totalitatea traductorilor primari cât și elementele de execuție;

2. nivelul ierarhic intermediar, denumit și „nivelul 1”, este destinat cuplului de proces. Rolul esențial al unui cuplor de proces este de a prelua de la traductoare semnale analogice, de a le converti în semnale binare, de a le prelucra elementar și de a le ordona într-un format special pentru a fi ușor transmise calculatorului de la punctul central.

Între elementele componente ale „nivelului 1” nu exista comunicație directă. Soluționarea conflictelor inevitabile între elementele „nivelului 1”, se face la nivelul ierarhic coordonator (automat sau prin decizia operatorului tehnolog).

Așfel, devine esențială necesitatea unui sistem de comunicație, bine pus la punct, între cele două niveluri, calitatea acestuia condiționând în mod esențial integritatea sistemului.

Cum sistemele hidrotehnice sunt caracterizate de distribuție geografică pe arii relativ întinse, afectate de zgomote electrice de tip industrial, mediul de transmisie a informației și a procedurilor adoptate în acest scop, devin elemente cheie în soluționarea problemelor ridicate în practică.

Caracteristicile de baza a elementelor componente ale nivelului 1 sunt:

- asigură izolarea galvanică față de proces;
- are facilități de prelucrare a informației numerice (conversie de formate, calculul unor sume ponderate);
- posibilitatea memorării unui volum redus de date;
- funcționare în conformitate cu un program rezident, care se încarcă la punerea în funcțiune și care, teoretic, nu se mai modifică pe toata durata de funcționare a cuplorului;
- posibilitatea de selectare a unei anumite proceduri de reglare, din mai multe disponibile și cea de reconfigurare;
- posibilitatea de interconectare în sistemul de transmisie a informației în vederea comunicării cu nivelul ierarhic superior;

3. nivelul ierarhic superior al unui sistem informatic de proces îl constituie nivelul de supervizare sau conducere, „nivelul 2”.

Acest nivel are posibilitatea comunicării cu toate elementele componente ale „nivelului 1”, având un rol de master în acest proces.

Supervizorul poate stoca cantități mari de informație. Aceste informații se referă la istoria procesului, perioadele descrise fiind de ordinul lunilor calendaristice.

Acest nivel oferă operatorului tehnolog sau de sistem facilități extensive de introducere și extragere date, constituind o baza de date cu informații utile pentru teleurmărire și teleconducere.

2.1.2. Considerații software

Din punct de vedere software, particularitățile programării sistemului de calcul cu microprocesor, destinat supravegherii și / sau conducerii în timp real a proceselor industriale, rezidă din următoarele condiții esențiale ce trebuie îndeplinite:

- posibilitatea sistemului de calcul de a răspunde instantaneu la semnalele emise de proces, în mod obligatoriu;
- cerința ca „timpul real” în care sistemul de calcul este ocupat cu o anumită operație, să fie strict corelat cu timpul în care se produc alte evenimente din proces (exterioare calculatorului);
- capacitatea sistemului de a primi datele direct din proces sau de a le transmite direct în proces prin intermediul unor convertitoare (analog-numerice numeric-analogice etc.);
- rezolvarea contradicției între capacitatea sistemului de calcul de a executa, la un moment dat, un riguros program și cerința procesului de a fi „servit” în „n” puncte simultan. Altfel spus, sistemul trebuie să facă față desfășurării simultane a diferitelor părți specifice ale procesului;
- posibilitatea programării tuturor categoriilor de procese (continue sau discrete);
- existența elementelor pentru testarea și depanarea eficiența a programelor.

În concluzie, pentru programarea unei astfel de aplicații, limbajul de programare utilizat trebuie să conțină trei categorii de instrucțiuni:

1. instrucțiuni care să permită programarea modului în care se desfășoară evenimentele din proces;
2. instrucțiuni care să precizeze locul în care se desfășoară evenimentele din proces;
3. instrucțiuni care se referă la modul sau intervalul de timp în care au loc evenimentele din proces.

Programele implementate, pe un astfel de sistem de calcul, se împart în:

- program de sistem;
- program de utilizare sau program utilizator.

Aspectele legate de componenta hardware și software vor fi discutate mai detaliat pe parcursul acestui capitol.

2.1.3. Aplicații distribuite programate în timp real

Prelucrarea distribuită este un nou model de calcul, care poate fi materializat numai în contextul unor caracteristici arhitecturale specifice. Încercarea de a defini în mod unic conceptul de sistem cu prelucrare distribuită conduce la o prima întrebare: ce se distribuie, resursele fizice și logice, controlul sau datele? Există prin urmare, mai multe dimensiuni ale unui sistem care pot fi distribuite, fiecare dimensiune putând avea în realitate un întreg spectru de „valori”.

2.1.3.1. Elemente conceptuale ale sistemelor distribuite

Distribuirea într-un sistem de prelucrare poate fi caracterizată din trei perspective majore:

- distribuirea la nivel fizic, caracterizată doar în termeni de facilități hardware;
- distribuirea la nivel logic, caracterizată doar în termeni de facilități logice (software sau date);
- distribuire la nivelul fizic și logic.

Este dificil în aceste condiții să se distingă sistemele cu prelucrare distribuită de restul sistemelor doar prin observarea caracteristicilor lor fizice (rețelele generale de calculatoare, de exemplu rețeaua ARPA, pot fi considerate ca fizic distribuite). Distribuirea fizică este doar o condiție necesară, deoarece apare în mod natural a doua întrebare: cum interacționează și cum sunt controlate elementele fizice distribuite?

2.1.3.2. Conceperea și programarea aplicațiilor distribuite

În rezolvarea unei probleme pe un sistem de calcul se pornește de la arhitectura fizică a sistemului și se încearcă să se găsească soluția care să exploateze cât mai eficient capacitățile acesteia. În general, o problema poate avea mai multe soluții, iar acestea se pot dovedi mai mult sau mai puțin „potrivite” pentru implementare și execuție pe o arhitectura suport dată. Prin virtualizare se permite în esență ca o arhitectură de sistem să poată fi croită către o problema, în loc de a croi problema către sistem.

Un aspect important, care a dobândit din ce în ce mai mult o recunoaștere explicită, îl constituie gradul de „potrivire” care trebuie să existe între structura virtuală a unei aplicații, construită de utilizator pe baza abstractizărilor oferite de sistemul de operare și de limbajul de programare și arhitectura reală a sistemului suport pe care aplicația se implementează și se execută. Importanța problemei derivă din faptul că performanțele unei soluții sau algoritmi pot fi radical diferite pentru diferite arhitecturi. Existența unei corespondențe structurale între cele două arhitecturi, virtuală (a aplicației) și reală (a sistemului suport), constituie astfel o cerință fundamentală de eficiență. Dacă este posibilă utilizarea la maximum a capacităților arhitecturii reale, se apreciază că aceasta este „programabilă”.

În aprecierea gradului de potrivire între o arhitectură dată și o aplicație, o considerație primară de la care trebuie pornit o constituie măsura în care soluția adoptată menține un echilibru între prelucrare, comunicație și memorie. Realizarea unui mixaj corespunzător între toate acestea constituie unul din aspectele esențiale pe care trebuie să se bazeze selectarea arhitecturii „potrivite” pentru o aplicație, cerință favorizată în ultima vreme și de noua economie a informaticii, caracterizată sintetic prin „cost redus în lățime de bandă, cicluri și octeți”.

Structurile hardware pentru prelucrarea distribuită reprezintă un tip particular de hardware și nu un simplu „conglomerat” de structuri hardware distincte. Hardware-ul prelucrării distribuite are, cum s-a văzut, o organizare inerent paralelă, ceea ce la nivel software nu poate avea decât consecința unei evoluții tot paralele. La nivelul aplicațiilor, adică al mediului „logic” de prelucrare distribuită, paralelismul este prin urmare implicit și necesar.

Cea mai simplă adaptare între o arhitectură paralelă și un algoritm dat constă, evident în identificarea directă a celor două. Pornindu-se de la o astfel de idee, a crescut interesul pentru proiectarea și realizarea de noi arhitecturi de prelucrare, cum sunt arhitecturile sistolice sau arhitecturile reconfigurabile dinamice. Un masiv sistolic ar putea fi considerat ca o formă particulară de sistem distribuit, el fiind constituit dintr-o rețea de procesoare foarte simple, identice. Fiecare procesor realizează o funcție simplă și comunică doar cu vecinii apropiați prin schimburi de mesaje pe canalele de conectare. Plecându-se de la specificarea problemei, dată de exemplu sub forma unui sistem de ecuații, se poate concepe o rețea suport „potrivită” de procesoare specializate.

Arhitecturile reconfigurabile dinamice sunt cele caracterizate prin topologii de interconectare între elementele care pot fi modificate sub control software. O arhitectură poate fi astfel privită ca realizând o tranziție dintr-o stare arhitecturală în alta, fiecare stare fiind caracterizată de o topologie de interconectare specifică. Pentru o aplicație dată, dezideratul global va fi cel de stabilire a stării arhitecturale care să se „potrivească” cel mai bine cu structura algoritmului utilizat. Consecința esențială a unor asemenea capacități dinamice este legată de îmbunătățirea performanțelor pentru un spectru mai larg de aplicații, fără necesitatea creșterii în dimensiune a sistemului.

De o manieră generală, se va considera pentru cele ce urmează că, la nivelul aplicațiilor, prelucrarea distribuită va apare ori de câte ori o colecție de entități (sau unități) logice independente, spațial separate, cooperează la rezolvarea unei singure probleme. Entitățile sau unitățile logice ale aplicației, ca o alternativă neutră pentru alți termeni utilizați în literatură (de exemplu „task”, „proces”, „agent”, „actor” etc.) vor avea implicit o funcționare concurentă și vor interacționa prin comunicație explicită. În acest context, o aplicație distribuită va fi în general considerată ca reprezentând o colecție de unități de prelucrare concurente, care comunică pentru obținerea unui deziderat logic comun, făcând explicite costurile de comunicație. Se poate observa că un astfel de model general, bazat pe independența unităților logice de prelucrare și pe accentuarea explicită a aspectelor de comunicație dintre acestea, se potrivește bine cu suportul fizic distribuit.

Programul care implementează o aplicație distribuită va fi simplu denumit program distribuit, adică un program în care modulele rezidă și se execută la locații aflate în comunicație, dar geografic distincte. În contextul de față este de asemenea util de precizat că, spre deosebire de un limbaj de programare concurrentă clasic (care este orientat pe asigurarea execuției de unități de prelucrare comunicând prin variabile protejate), un limbaj de programare distribuită este destinat și orientat către asigurarea execuției concurente de unități de prelucrare care nu comunică prin variabile partajate, făcând din acest motiv explicite costurile de comunicație.

2.1.4. Sisteme de prelucrare distribuită a datelor. Obiective generale

Conceptul de sistem cu prelucrare distribuită a datelor a apărut în jurul anului 1975, ca o alternativă realistă de punere în aplicare a dezvoltărilor din microelectronică și de exprimare pe o cale mai naturală a mutațiilor produse la nivelul utilizatorului: putere de calcul crescută, flexibilitate, modularitate, integritate, fiabilitate etc.

Diversificarea domeniilor de utilizare a tehnicii de calcul a evidențiat, așa cum s-a arătat, necesitatea de specializare a sistemelor pe clase particulare de aplicații.

Obiectivele prelucrării distribuite pot avea un caracter particular, dependent de aplicație și de mediu și, prin urmare, să nu fie la fel de semnificative în toate cazurile. Trăsăturile specifice ale unui mediu de control al proceselor industriale (timp real, siguranța în funcționare etc.) nu sunt aceleași, de exemplu, cu cele ale activităților de birou. Din acest motiv, se vor considera în continuare, comparativ cu limitările prelucrării convenționale, obiectivele tot mai des recunoscute ca având caracter general pentru prelucrarea distribuită: extensibilitate, integritate, performanță.

Extensibilitate

Extensibilitatea unui sistem exprimă măsura în care funcționalitatea și performanțele sistemului pot fi modificate fără a fi necesară modificarea proiectării acestuia. Un sistem extensibil este conceput pentru a fi ușor adaptat la un context dinamic, fără întreruperea funcționării sistemului. Două atribute majore caracterizează noțiunea de extensibilitate: modificarea ușoară și evoluția incrementală.

În sensul ușurinței de modificare, extensibilitatea implică simplitatea înlocuirii unei funcții logice cu alte funcții / elemente ale sistemului. Extensibilitatea sistemului presupune creșteri în funcționalitate și performanță cu costuri incrementale reduse. Acest atribut conduce la *ușurință de instalare*, un sistem extensibil putând fi dezvoltat incremental, pornindu-se inițial de la configurații simple și ieftine. Mediul în care e instalat un sistem de prelucrare a datelor poate genera uzual două tipuri de modificări: modificări privitoare la cerințele de performanță și modificări ale cerințelor funcționale. Arhitecturile multiprocesor convenționale oferă unele îmbunătățiri ale extensibilității, comparativ cu monoprocsoarele (Sat 880). Cu toate acestea, structura lor de interconectare, de comunicație și de control impune configurații minime la un cost ridicat, limite reduse privind numărul maxim de elemente de prelucrare și o dispersare spațială foarte restrictivă.

Integritate

Integritatea unui sistem este determinată de măsura în care sistemul tolerează defectele, erorile sau anomaliile care apar în timpul funcționării sale, menținându-se într-o *stare corectă*.

Spre deosebire de noțiunea de disponibilitate, integritatea se referă la posibilitatea sistemului de a realiza prelucrări în prezența defectelor. Pentru a-și păstra integritatea în ciuda apariției de defecte, sistemul trebuie să poată să finalizeze corect orice cerere de prelucrare care apare.

Un principiu de bază în proiectarea unor astfel de sisteme distribuite din punct de vedere al integrității, îl reprezintă *suspiciunea reciprocă* între diferitele niveluri de abstractizare ale sistemului.

Arhitecturile centralizate cu elemente de prelucrare multiple pot evita resursele singulare partajate, eliminând unele dintre dezavantajele monoprocsoarelor. Cel mai adesea însă, dacă nu întotdeauna, aceste îmbunătățiri potențiale sunt compromise (uneori în mod implicit) în favoarea altor deziderate, cum ar fi: multiplexarea procesoarelor, memorie întreținută, protocoale nesigure pentru comunicații pe magistrale etc. Deși sunt posibile, detecția, diagnoza și recuperarea automată a erorilor prin cooperare între elementele de prelucrare multiple, structura centralizată a acestor sisteme reduce mult din eficiența acestei tehnici.

Performanțe

Experiența de până acum arată că, la un moment dat, indiferent de stadiul tehnologic existent, performanțele peste un anumit nivel nu pot fi atinse decât cu arhitecturi având elemente de prelucrare multiple. Trebuie arătat că există mai multe argumente, sub raport cost / eficiență, care favorizează utilizarea unei structuri cu elemente de prelucrare mai simple în locul unui singur procesor complex, chiar și atunci când performanțele sunt echivalente.

Utilizarea unui *sistem multiprocesor* corespunzător cu n elemente de prelucrare, câștigul de performanță obținut nu poate atinge de n ori performanța sistemului monoprocesor corespunzător. Diverse formule empirice au fost prezentate în literatură sau recomandate de firme pentru predicția îmbunătățirii performanțelor pe baza adăugării de noi procesoare la un sistem multiprocesor existent (creșteri semnificative ale performanțelor rezultă doar pentru primele câteva procesoare adăugate - cca. 70-80% - pentru un sistem biprocesor, numai cca.110% pentru un sistem cu 3 procesoare, de exemplu).

Această neliniaritate în creșterea performanțelor față de creșterea liniară a numărului elementelor de prelucrare dintr-un sistem multiprocesor convențional, se datorează unei diversități de factori, atât de natura hardware, cât și software, comunicarea prin memoria partajată comună și controlul global centralizat fiind cei mai importanți dintre aceștia.

2.2. Probleme legate de urmărirea și conducerea proceselor tehnologice

2.2.1. Conducerea proceselor tehnologice. Introducere

Revoluția tehnico științifică contemporană, dublată de revoluția „informațională” au implicații profunde în modul de producție al bunurilor materiale, dar și a serviciilor prestate. Ușurarea muncii omului, transferarea centrului de greutate dinspre munca fizică spre cea intelectuală, ca și reducerea diferențelor dintre acestea ocupă un loc de frunte în rândul tendințelor actuale ale tehnicii. Principalul mijloc pentru atingerea unui asemenea obiectiv îl constituie *automatizarea proceselor de producție*. Prin *automatizare* se înțelege așadar ansamblul de măsuri tehnice luate pentru reducerea sau eliminarea participării umane în desfășurarea proceselor de producție în general și al proceselor tehnologice în special.

O primă caracterizare a procesului tehnologic se poate face prin evidențierea unui ansamblu de fenomene fizico-chimice, care implică transferuri și transformări masice și energetice, cu o destinație funcțională preconizată. O descriere cantitativă a procesului presupune evidențierea unor mărimi caracteristice, precum și a depen-

dențelor dintre acestea, stabilindu-se legăturile cauzale ce guvernează desfășurarea sa în timp. Astfel de mărimi pot fi debitele masice și energetice de intrare - ieșire, temperatura, presiunea etc.

Notând generic ansamblul debitelor de substanță și energie W_i , iar debitele de ieșire cu W_e , între care și debitul de produs finit, putem adopta următoare reprezentare schematică a produsului, întâlnită în fig. 2.1. Neglijând pierderile de transfer ca și distribuțiile spațiale ale diverselor mărimi (altfel spus, vom presupune că procesul are parametri concentrați) putem distinge în funcționarea procesului două tipuri de regimuri:

a) **regimuri de echilibru staționar** (sau, mai scurt, regimuri staționare) în care, cu ipotezele adoptate, sunt îndeplinite condițiile de bilanț de masă și energie pe ansamblu ce pot fi exprimate prin relația:

$$W_e = W_i \quad (2.1)$$

b) **regimuri dinamice sau tranzitorii** (a doua denumire provenind de la faptul că se includ tranzițiile de la un regim staționar la altul), în care datorită fenomenelor de acumulare și dezacumulare energetică sau masică, egalitatea (2.1) nu mai este respectată, adică:

$$W_e \neq W_i \quad (2.2)$$



Fig. 2.1. Reprezentarea schematică a unui proces, cu evidențierea debitelor masice și energetice de intrare și de ieșire.

În consecință, pentru exprimarea cantitativă a echilibrului dinamic, respectiv a evoluției temporale a ansamblului de fenomene ce definesc procesul, este necesară evidențierea, printre mărimile caracteristice, a unui set de mărimi, numite *mărimi fizice de stare*, notat cu x , cu ajutorul cărora să poată fi descrise în mod univoc fenomenele tranzitorii interne.

Cu alte cuvinte, ansamblul (vectorul) mărimilor fizice de stare trebuie să fie astfel definit încât, la un moment de timp precizat, valoarea mărimilor de stare să fie suficientă pentru a putea, pe baza legilor ce guvernează desfășurarea procesului și cunoscând evoluția în timp a mărimilor din seturile W_i , W_e , începând din momentul respectiv, să determinăm univoc evoluția ulterioară a tuturor mărimilor de stare.

În acest mod, regimurile staționare pot fi definite și prin constanța în timp a tuturor mărimilor de stare, iar regimurile dinamice prin existența unor variații a cel puțin uneia dintre mărimile de stare.

În concluzie, un anumit proces tehnologic poate fi caracterizat prin ansamblul mărimilor caracteristice și al legăturilor cauzale dintre acestea, date de legile obiective ale naturii.

Conducerea unui proces tehnologic impune în primul rând stabilirea unui obiectiv al conducerii care fixează, de fapt, însăși destinația funcțională a procesului respectiv. Pentru a putea defini precis un astfel de obiectiv al conducerii este necesară introducerea unui nou set de mărimi, pe care le vom numi *mărimi de calitate ale*

procesului și le vom considera drept componente ale vectorului mărimilor de calitate notat cu z . Ele sunt nemijlocit dependente de mărimile de stare, unele dintre ele putându-se chiar identifica cu acestea. Cu ajutorul vectorului z , obiectivul conducerii poate fi exprimat simplu prin necesitatea atingerii și menținerii unui regim (în general staționar) al procesului, în care mărimile de calitate au valori impuse (prin vectorul dat z^*), stabilite de tehnolog și considerate optime.

Dacă am redefini vectorul mărimilor de calitate prin abaterea lui z față de valoarea optimă z^* , respectiv am introduce vectorul $Az = z - z^*$, atunci, evident $Az^* = 0$ și în consecință, obiectivul conducerii procesului ar fi aducerea (și menținerea) procesului într-o stare în care vectorul mărimilor de calitate este nul. O astfel de formulare o vom întâlni în formalizarea uneia dintre cele mai importante probleme de conducere și anume problema reglării.

Cu cele precizate până acum, este foarte natural a defini, neformal, conducerea unui proces tehnologic prin *ansamblul de acțiuni (intervenții) efectuate asupra procesului cu scopul aducerii și menținerii procesului într-o stare care satisface obiectivul precizat al conducerii*.

Materializarea conducerii implică deci, în plus, pe de o parte precizarea posibilităților de intervenție asupra procesului, iar pe de altă parte precizarea posibilităților de cunoaștere a stării curente a procesului, fără de care elaborarea modului de intervenție nu ar fi cu putință. Aceasta necesită fixarea a încă două categorii din cadrul mărimilor caracteristice și anume: un set de mărimi pe care le vom numi *mărimi de execuție* (notate cu m) prin modificarea cărora putem modifica starea procesului și un alt set de mărimi numite *mărimi măsurate* (notate cu y) prin intermediul cărora se poate aprecia starea curentă.

Ansamblul mărimilor măsurate este dictat pe de-o parte de posibilitățile tehnice de măsurare existente, iar pe de altă parte de necesitatea de a putea extrage din acestea starea procesului. Evident, de câte ori o mărime de stare poate fi măsurată, este oportun ca acest lucru să fie făcut.

Un aspect extrem de important al acțiunii de conducere este caracterul său permanent. Într-adevăr, odată atins obiectivul conducerii, respectiv atinsă starea în care mărimile de calitate au valorile dorite, păstrarea constantă a tuturor mărimilor de execuție nu asigură menținerea constantă și a mărimilor de calitate. Pentru a evidenția cauza unei astfel de situații, trebuie să luăm în considerare încă un set de mărimi care au proprietatea că nu depind de cele introduse până acum și, simultan au variații necontrolate și imprevizibile, influențând starea procesului și mărimile de calitate. Le vom numi *mărimi perturbatoare*, desemnând ansamblul lor printr-un vector notat cu v .

Datorită acțiunii mărimilor perturbatoare, conducerea unui proces nu poate înceta decât odată cu oprirea acestuia din funcțiune.

În majoritatea situațiilor, perturbația principală o reprezintă variațiile de sarcină. De asemenea, variația condițiilor de mediu în care se desfășoară procesul este o sursă de perturbații.

Rezumând cele de mai sus, putem detalia reprezentarea schematică din fig. 2.1, scoțând în evidență mărimile specificate (fig. 2.2), unde am considerat că sunt n mărimi de stare, I mărimi de calitate, m mărimi de execuție, q mărimi perturbatoare și p mărimi măsurate. Săgețile relevă sensul lanțului causal. Astfel, mărimile de „intrare” m și v reprezintă cauza pentru variația mărimilor „interne” x și z care, la rândul lor, reprezintă cauza pentru variația mărimilor măsurate (de „ieșire”) y .

În reprezentarea din fig. 2.2 trebuie adăugat că intervenția asupra procesului se materializează prin mijlocirea unor aparate și dispozitive numite *elemente de execuție*, care realizează adaptarea cu procesul condus, iar mărimile măsurate, după cum le arată și numele, se obțin prin mijlocirea aparatelor de măsură sau a traductoarelor.

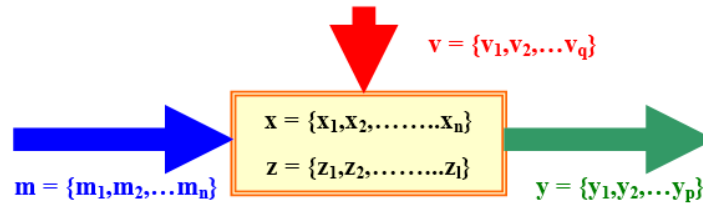


Fig. 2.2. Reprezentarea unui proces, cu scoaterea în evidență a mărimilor de intrare, a mărimilor interne și a mărimilor de ieșire.

Mărimile de intrare ale elementelor de execuție prin intermediul cărora se face, de fapt, conducerea procesului, vor fi denumite *mărimi de comandă* și vor fi notate generic cu u .

Astfel, în locul schemei din fig. 2.2, obținem reprezentarea din fig. 2.3, în care vom considera setul mărimilor fizice de stare x ca fiind extins cu eventuale mărimi de stare ce caracterizează posibile fenomene interne de acumulare - dezacumulare din elementele de execuție și aparatele de măsură sau traductoare.

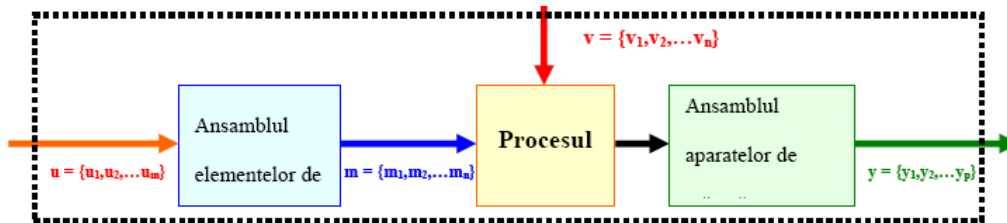


Fig. 2.3. Schema bloc a unui proces tehnologic, prevăzut cu elemente de interfață cu operatorul uman sau cu mijloacele de conducere.

Cu aceste precizări, problema conducerii procesului se formulează astfel: *fiind dat obiectivul conducerii prin setul de valori z^* considerate optime, ale mărimilor de calitate, și având acces permanent la valorile mărimilor măsurate y , să se elaboreze și să se aplice mărimile de comandă u , astfel încât să fie atins și menținut obiectivul conducerii.*

În cazul în care elaborarea și aplicarea comenzilor se efectuează de către un operator uman, vom spune că avem o *conducere manuală* (chiar dacă se face apel și la resursele intelectuale ale ființei umane), iar în cazul în care aceste operații sunt efectuate de anumite mijloace tehnice, fără intervenție umană, avem o *conducere automată*.

Trebuie remarcată, în orice operație de conducere, existența unei vehiculări în circuit închis a informațiilor și acțiunilor (fig. 2.4).

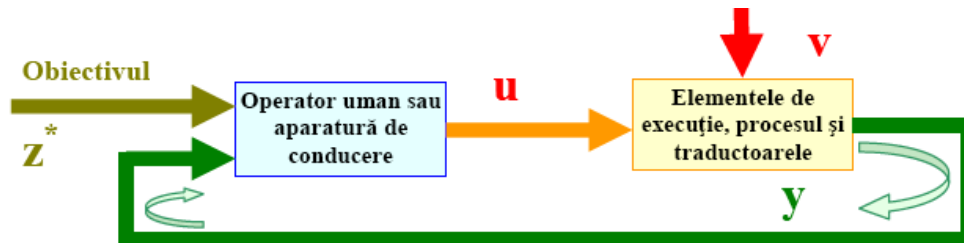


Fig. 2.4. Evidențierea unei vehiculări, în circuit închis, a informațiilor și acțiunilor, într-o conducere manuală sau automată a procesului.

Omul sau mijlocul tehnic care realizează conducerea primește în permanență informații privind evoluția procesului prin intermediul mărimilor măsurate, informații care, evident, determină, în urma unei prelucrări corespunzătoare, nivelul mărimilor de comandă. Acestea, la rândul lor, modifică starea procesului și deci valorile mărimilor de ieșire.

Apare un fenomen de autoinfluențare a mărimilor caracteristice, așa numita *reacție* (feed-back), care conferă proprietăți esențial noi ansamblului de elemente implicite, în sensul că proprietățile ansamblului diferă profund de simpla reuniune a proprietăților elementelor componente.

2.2.2. Sisteme dinamice

Plecând de la faptul că orice abordare teoretică presupune un anumit nivel de abstractizare, în baza căruia este posibilă, prin dezvoltări corespunzătoare, obținerea unor rezultate cu caracter general, vom introduce un corespondent abstract al noțiunii de proces (fizic) pe care-l vom denumi *sistem dinamic*. Într-o primă accepțiune, un sistem dinamic reprezintă o descriere matematică adecvată a unui proces fizic, descriere care face apel la mărimile introduse (mărimile de stare, calitate, comandă, măsura și perturbatoare) și exprimă legăturile cauzale dintre acestea. În mod inerent, o astfel de descriere nu poate surprinde decât aspectele esențiale ale realității procesului respectiv și, prin urmare, este un model matematic mai mult sau mai puțin ideal. Într-o viziune mai largă, noțiunea de sistem dinamic poate fi desprinsă de legăturile directe cu procesele fizice concrete și definită pe cale axiomatică. În acest fel, sistemul dinamic devine un obiect matematic, elementul central pentru o teorie a sistemelor. Având în vedere destinația acestei lucrări, vom privi sistemele dinamice în primul rând drept modele matematice ale unor posibile procese fizice. Cu titlu informativ, vom prezenta în paragraful următor o scurtă introducere axiomatică în teoria sistemelor.

Fie, așadar, variabila independentă t , care are semnificația de timp și introducem:

- vectorul mărimilor de stare $\mathbf{x} \in \mathbf{R}^n$;
- vectorul mărimilor de comandă $\mathbf{u} \in \mathbf{R}^m$;
- vectorul mărimilor perturbatoare $\mathbf{v} \in \mathbf{R}^q$;
- vectorul mărimilor măsurate $\mathbf{y} \in \mathbf{R}^p$;
- vectorul mărimilor de calitate $\mathbf{z} \in \mathbf{R}^l$.

În legătură cu mulțimea momentelor de timp, vom presupune două cazuri:

1. *cazul 1*: $\mathbf{t} \in \mathbf{R}$, situație în care vom vorbi de sisteme dinamice continue;
2. *cazul 2*: $\mathbf{t} \in \mathbf{Z}$, corespunzătoare sistemelor dinamice discrete. Ținând seama de faptul că nu vom opera decât cu sisteme dinamice, vom omite în general atributul dinamic în cele ce urmează.

Evident, sistemele simultan liniare și invariante în timp vor fi descrise de următoarele ecuații:

- în cazul *sistemelor continue*:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{E}\mathbf{v}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t); \\ \mathbf{z}(t) &= \mathbf{D}\mathbf{x}(t); \quad \mathbf{t} \in \mathbf{R} \end{aligned} \quad (2.3)$$

- în cazul *sistemelor discrete*:

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t+1) + \mathbf{B}\mathbf{u}(t+1) + \mathbf{E}\mathbf{v}(t+1) \\ \mathbf{y}(t+1) &= \mathbf{C}\mathbf{x}(t+1); \\ \mathbf{z}(t+1) &= \mathbf{D}\mathbf{x}(t+1); \quad \mathbf{t} \in \mathbf{Z} \end{aligned} \quad (2.4)$$

Atât în (2.3) cât și în (2.4), A, B, C, D, E sunt matrici constante, de dimensiuni corespunzătoare. Cu observațiile care au justificat scrierea relațiilor (2.3) și (2.4), sistemele liniare, invariante în timp, se vor scrie sub forma:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t); \quad \mathbf{t} \in \mathbf{R} \end{aligned} \quad (2.5)$$

pentru cele continue, și, respectiv:

$$\begin{aligned} x(t+1) &= Ax(t+1) + Bu(t+1) \\ y(t+1) &= Cx(t+1); \quad \mathbf{t} \in \mathbf{Z} \end{aligned} \quad (2.6)$$

pentru cele discrete.

Examinând relațiile (2.5) și (2.6), se constată că sistemele liniare constante sunt complet definite de tripletul de matrici $S = (A, B, C)$ pe care îl vom numi din acest motiv tot sistem, subînțelegându-se și atributele de constant și liniar.

Menționăm aici faptul că ne vom ocupa în primul rând de elemente de teoria sistemelor liniare constante, întrucât în acest domeniu există rezultate foarte importante, atât din punct de vedere principal cât și procedural. Se ridică însă întrebarea: „În ce măsură sistemele liniare pot fi acceptate ca modele pentru procese reale?” Răspunsul la această întrebare este la prima vedere total nesatisfăcător, în sensul că fenomenele naturii se supun rareori unor legi cu caracter liniar. Totuși, pentru aria de probleme care interesează teoria sistemelor automate, modelele liniare pot constitui aproximații satisfăcătoare. Ne referim în primul rând la situația în care este necesară menținerea stării procesului la valori în care condițiile impuse mărimilor de calitate sunt satisfăcute. În operația de conducere, abaterile de la aceste valori apar datorită, în esență, acțiunii mărimilor perturbatoare și, în consecință, în prezenta acțiunilor de conducere corect elaborate, acestea vor fi în mod necesari mici.

Pentru variații „mici” în jurul unui regim de echilibru, aproximația liniară este de cele mai multe ori suficientă pentru ca deducțiile teoretice făcute pe acest model să nu-și piardă interesul practic.

Ca să fim mai expliciti, considerăm un proces descris satisfăcător de un sistem neliniar, invariant în timp, continuu:

$$\begin{aligned} \dot{x}(t) &= f[x(t), u(t)] \\ y(t) &= g[x(t)]; \quad \mathbf{t} \in \mathbf{R} \end{aligned} \quad (2.7)$$

care admite un regim staționar de funcționare [$x(t) = 0$], caracterizat de valorile constante ale intrărilor și stării, notate cu u^* și x^* , cărora le corespunde o valoare constantă a ieșirii y^* . Evident:

$$\begin{aligned} f(x^*, u^*) &= 0 \\ y^* &= g(x^*) \end{aligned}$$

În ipoteza că f și g sunt funcții derivabile de un număr suficient de ori, se poate scrie:

$$f(x, u) = f(x^*, u^*) + \left. \frac{\partial f}{\partial x} \right|_{x=x^*} \cdot (x - x^*) + \left. \frac{\partial f}{\partial u} \right|_{u=u^*} \cdot (u - u^*) + \mu(x, u) \quad (2.8)$$

$$g(x, u) = g(x^*) + \left. \frac{\partial g}{\partial x} \right|_{x=x^*} \cdot (x - x^*) + v(x) \quad (2.9)$$

Notând cu $\Delta x = x - x^*$, $\Delta u = u - u^*$ și $\Delta y = y - y^*$, funcțiile $\mu(x, u)$ și $v(x)$ din (2.8) și (2.9), satisfac condițiile:

$$\lim_{\|\Delta x\|^2 + \|\Delta u\|^2 \rightarrow 0} \frac{\|\mu(x, u)\|}{\sqrt{\|\Delta x\|^2 + \|\Delta u\|^2}} = 0 \quad (2.10)$$

$$\lim_{\|\Delta x\| \rightarrow 0} \frac{\|v(x)\|}{\|\Delta x\|} = 0 \quad (2.11)$$

altfel spus, devin neglijabile în raport cu termenii liniari, când $\|\Delta x\|, \|\Delta u\|$ sunt suficient de mici.

Notând cu:

$$A = \left. \frac{\partial f}{\partial x} \right|_{\substack{x=x^* \\ u=u^*}}, B = \left. \frac{\partial f}{\partial u} \right|_{\substack{x=x^* \\ u=u^*}}, C = \left. \frac{\partial f}{\partial x} \right|_{x=x^*} \quad (2.12)$$

Care sunt matrici care depind de regimul staționar luat în considerație, dar constante în timp. Ținând seama de (2.7) și (2.12), ca și de faptul că:

$$\Delta x = x - x^* = x \quad (2.13)$$

Vom obține, din (2.8) și (2.9), prin neglijarea termenilor reziduali:

$$\begin{aligned} x &= Ax + Bu; \\ y &= Cx \end{aligned} \quad (2.14)$$

care este un sistem liniar capabil să descrie satisfăcător comportarea procesului în apropierea regimului staționar considerat. În acest fel, studiul sistemelor liniare răspunde efectiv unor necesități de ordin practic, legate nemijlocit de conducerea unor procese industriale concrete.

Operația prin care din (2.6) se obține (2.14) poartă numele de liniarizare și va fi utilizată curent în construcția unor modele liniare pentru procese fizice specifice.

În legătură cu necesitatea studierii sistemelor discrete, aici motivațiile decurg în principal din extrem de importantele avantaje tehnico-economice pe care le aduce utilizarea calculatoarelor în conducerea proceselor tehnologice. Or, prelucrarea informației în calculatoare sub forma numerică, precum și vitezele finite de calcul (chiar dacă sunt foarte mari) impun pe de-o parte realizarea măsurărilor la momente distincte de timp decalate între ele de intervale dependente de viteza de calcul și de complexitatea algoritmilor de conducere utilizați.

În consecință, este suficient să cunoaștem comenzile ca și mărimile măsurate numai într-o mulțime numărabilă de momente de timp, chiar dacă procesul fizic este în esență să continuu. Rezultă oportunitatea găsirii unor corespondențe discrete pentru sisteme continue date. Ne vom mărgini la a prezenta o modalitate de a face acest lucru pentru sisteme liniare constante. Pentru aceasta, fie sistemul liniar continuu și presupunem dată o constantă reală h , numită perioadă sau pas de discretizare (de eșantionare) cu dimensiune temporală.

Presupunem că mulțimea reală a momentelor de timp a fost împărțită în intervale de lungime h , prin fixarea unei diviziuni $\{kh/k \in Z\}$. Admitem de asemenea că $u(t)$ poate fi aproximată printr-o funcție scară, respectiv vom presupune că:

$$u(t) = u(kh), \text{ pentru } t \in [kh, (k+1)h], k \in Z \quad (2.15)$$

Reprezentarea unei astfel de funcții pentru cazul scalar este dată în fig. 2.5.

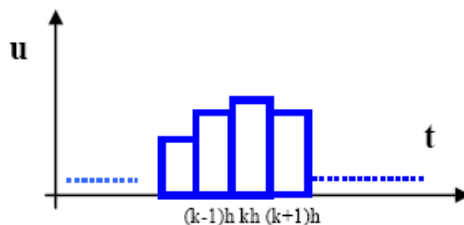


Fig. 2.5. Funcție de comandă tip scară, întâlnită în sistemele de reglare discretă a unor procese continue.

Admițând drept cunoscut $x(kh)$ și cunoscând $u(t)$, $t \in [kh, (k+1)h]$, putem calcula $x[(k+1)h]$, integrând prima ecuație. Se obține astfel:

$$\begin{aligned} x[(k+1)h] &= e^{Ah}x(kh) + \int_{kh}^{(k+1)h} e^{A[(k+1)h-n]}Bu(n)dn = \\ &= e^{Ah}x(kh) + \int_{kh}^{(k+1)h} e^{A[(k+1)h-n]}Bd(n)u(kh) = \\ &= e^{Ah}x(kh) + \left[\int_0^h e^{Ap}dp \right]B(u)(kh) \end{aligned} \quad (2.16)$$

Notând:

$$A_d = e^{Ah}, B_d = \left[\int_0^h e^{Ap}dp \right]B, C_d = C \quad (2.17)$$

vom obține:

$$\begin{aligned} x[(k+1)h] &= A_dx(kh) + B_du(kh) \\ y(kh) &= Cdx(kh) \end{aligned} \quad (2.18)$$

sau, adoptând h drept unitate de timp convențională:

$$\begin{aligned} x(k+1) &= A_dx(k) + B_du(k) \\ y(k) &= Cdx(k) \end{aligned} \quad (2.19)$$

adică exact forma unui sistem liniar discret.

Sistemele (2.5) și (2.19) au proprietatea că pentru funcții de intrare $u(t)$ în scara, respectiv $u(kh)$ și o aceeași inițializare $x(0)$ conduc la soluții (evoluții ale stării) care în momentele $t=kh$, unde $k \in \mathbb{Z}$, $k > 0$ coincid. În esență ele reprezintă modele ale aceluiași proces, dar cu destinație și utilitate diferită. Operația prin care se obține sistemul (2.19) din (2.5) poartă numele de *discretizare*.

Succinta prezentare a noțiunii de sistem dinamic servește ca bază pentru dezvoltări ulterioare. Menționăm că există și alte modalități de descriere matematică a proprietăților dinamice ale unui proces fizic, unele din ele, în anumite condiții, fiind echivalente cu descrierea prezentată mai sus, numită și *descriere de stare*.

2.2.3. Niveluri de automatizare în conducerea proceselor industriale

2.2.3.1. Structuri ierarhizate de conducere

Modalitatea ideală pentru îndeplinirea obiectivelor ce revin sistemelor de conducere este organizarea acestor sisteme în structuri ierarhizate, abordare modernă ce satisface exigențele actuale din automatizările industriale. O structură ierarhizată multinivel, care surprinde în principal aspectele funcționale ale conducerii este organizată pe niveluri de automatizare. Conceptul de conducere multinivel, în ideea implementării cu echipamente numerice, asigură în principal trei funcții de bază: *identificarea și optimizarea proceselor*, funcții realizate la nivel ierarhic superior și *funcția de achiziție, prelucrare primară de date și reglare*, la nivel ierarhic inferior (fig. 2.6). Pentru nivelul inferior sunt importante aspectele de execuție și, din acest motiv, interesează în primul rând regimul dinamic de funcționare a procesului condus.

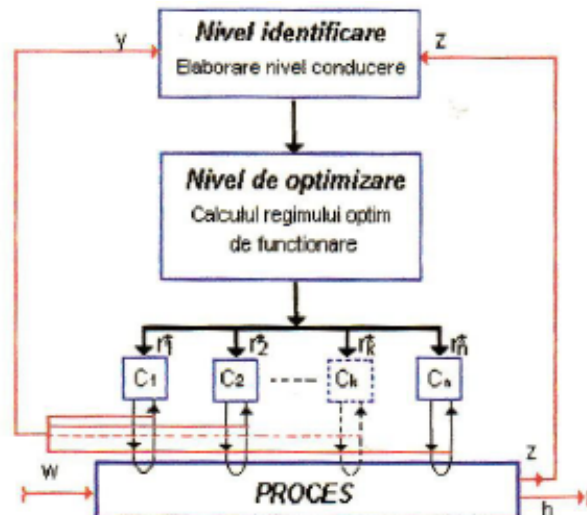


Fig. 2.6. Structura ierarhizată pentru conducerea numerică.

2.2.3.2. Nivelul de achiziție, prelucrare și reglare

La nivelul de achiziție, prelucrare și reglare se preiau mărimile din proces, se prelucrează în unitatea numerică și se elaborează comenzile necesare pentru reglarea acestor mărimi, fie prin mijloacele automatizărilor convenționale, fie prin utilizarea unor microcalculatoare de proces.

Achiziția de date este asigurată prin intermediul traductoarelor de măsurare, mărimile măsurate fiind prelucrate ulterior prin operații de filtrare, conversie, calcule de corecție etc., iar reglarea prin utilizarea unor reglatoare automate sau numerice, care permit implementarea unor algoritmi de reglare ce aduc și mențin, în cadrul unor structuri în circuit închis, parametrii y la valorile prescrise r .

Prelucrarea primară și reglarea pot fi asigurate prin metode și mijloace clasice, convenționale sau numerice. Avantajele esențiale, furnizate de soluția numerică, sunt datorate flexibilității și elasticității oferite de resursele „software”, care permit anumite modificări, ajustări sau rafinări prin prelucrare numerică.

În ceea ce privește funcția de reglare, este de menționat păstrarea algoritmilor tradiționali în trei termeni PID, verificați în practica automatizării convenționale și implementați fie în varianta continuă, fie în cea numerică.

Această opțiune pentru structurile PID este motivată prin:

- existența unei experiențe în operarea algoritmilor PID și în exploatarea lor curentă în aplicațiile industriale;
- universalitatea implementării algoritmilor PID, valabilă în reglarea unor categorii largi de procese tehnologice;
- simplitatea implementării bazate pe ajustarea unui număr restrâns de parametri și cunoașterea influenței acestora asupra performanțelor sistemului de reglare automată;
- elaborarea unor metodologii eficiente de acordare optimă, un calcul rapid al comenzilor, condiție esențială pentru gestionarea timpului de calcul;
- conservarea modelelor matematice, care reproduc procesele fizice ce urmează să fie reglate, la reprezentările:

$$H_{P1}(s) = \frac{k_1 e^{-\pi s}}{T_s + 1} \quad (2.20)$$

$$H_{P2}(s) = \frac{k_2 e^{-\pi s}}{T_1 s^2 + T_2 s + 1} \quad (2.21)$$

care pretind algoritmi PID. Aceștia asigură fiabilitatea și robustețea necesară sistemelor de reglare automată.

2.2.3.3. Nivelul de identificare

Pentru nivelul ierarhic superior, interesează în special aspectele decizionale asociate regimului staționar în cadrul acțiunii de conducere. La nivelul de identificare se elaborează un model matematic de conducere $z = f(Q, y)$ unde, (conform notațiilor din fig. 2.6), z reprezintă mărimea de calitate (ieșirea modelului), y - vectorul mărimilor de intrare în model, θ - vectorul coeficienților.

Un model este determinat univoc, dacă este precizată structura sa, adică dacă este explicitată dependența dintre mărimile z și y exprimate prin funcția f și dacă sunt estimați cantitativ coeficienții modelului θ . Având în vedere gradul de generalitate pe care trebuie să-l ofere un model (aceleși model poate reproduce procese de natură fizică diferită, dar cu aceeași evoluție), se poate preciza că un model matematic propus pentru o aplicație concretă este un reprezentant ales dintr-o clasă de modele echivalente.

Evaluarea modelului de conducere pentru un proces dat poate fi dezvoltată pe două căi:

- pe baza cunoașterii legilor fizice care guvernează evoluția procesului, obținându-se un model analitic determinist. Limitările procedurii analitice sunt evidențiate de imposibilitatea descrierii proceselor cu structură complexă datorită unei insuficiente cunoașteri a legităților ce guvernează evoluția acestor procese;
- prin prelucrarea datelor achiziționate din proces, obținându-se un model experimental. Avantajul procedurii statistice experimentale este evident, dacă se dispune de un echipament numeric pentru prelucrarea datelor.

Cele două modalități posibile nu se exclud; dimpotrivă, procedura analitică asigură o informație apriorică pentru procedura experimentală, cel puțin în faza de precizare a structurii modelului. Etapele de construcție ale unui model de conducere sunt eșalonate astfel:

- **modelarea procesului**, operațiune prin care se asigură determinarea calitativă, adică structura modelului, în sensul precizării clasei de echivalență din care face parte structura sa;
- **identificarea modelului**, operațiune prin care se determină cantitativ modelul, în sensul că se estimează valoric coeficienții modelului, prin localizarea acestuia în clasa de echivalență precizată;
- **validarea modelului**, operațiune prin care se stabilește gradul de adecvanță al modelului la procesul condus, sau în ce măsură se regăsește funcționarea procesului în modelul care a fost determinat. Din punct de vedere matematic, un model este validat formal, dacă o distanță definită între model și proces este sub o limită impusă. Dacă se notează prin $D(M, P) \leq \varepsilon_p$. În mod uzual, D se definește ca:
 - distanța de structură, prin care se exprimă diferența între parametrii modelului și cei ai procesului:

$$D_S(\theta_m, \theta_p) = (\theta_m - \theta_p)^T A (\theta_m - \theta_p) \quad (2.22)$$

unde A este o matrice metrică pozitiv definită. Această distanță nu este măsurabilă, deoarece θ_p nu este cunoscut, dar în măsura în care θ_p este evaluat, devine operațională;

- distanța de predilecție, prin care se exprimă diferența dintre ieșirea modelului, care prezice procesul și ieșirea procesului la momente discrete de timp k :

$$D_p = \sum_{k=1}^N \left[\overset{\bullet}{z(k)} - z(k) \right]^2 \quad (2.23)$$

unde N este numărul de observații efectuate în proces;

- **aducerea la zi a modelului** este operația prin care se încearcă eventuale ajustări mai puțin în structură și mai frecvent în estimarea coeficienților, atunci când se constată diferențe între model și funcționarea procesului. Aceste ajustări sunt realizate prin instruirea modelului pe baza unor date noi, achiziționate din proces.

2.2.3.4. Nivelul de optimizare

La nivelul de optimizare se asigură alegerea din mulțimea de regimuri de funcționare posibile a regimului optim, exprimat prin condiția $y = y^*$, unde y^* este soluția unei probleme de optimizare parametrică:

$$\min F(y)_{y \in D_{adm}} \quad (2.24)$$

Funcția F este indicele de calitate sau performanță, asociat procesului condus, iar D_{adm} este domeniul admisibil în care vectorul mărimilor y poate lua valori. Indicele de calitate $F(y)$ se construiește din modelul $z = f(\theta, y)$, sau uneori se confundă cu acest model și are o semnificație tehnologică (temperatură, concentrație etc.), sau economică (randament, cost, producție orară etc.).

Domeniul admisibil $D_{adm} \in R^n$ se stabilește din restricțiile care se impun asupra mărimilor y și poate fi definit astfel:

$$D_{adm} = \{y \in R^n \mid g_j(y) \leq 0, j = 1, 2, \dots, m\}, \quad (2.25)$$

$g_j(y)$ semnificând funcțiile de restricții, asociate problemei de optimizare (2.24).

Datorită disponibilităților oferite de echipamentul numeric de conducere, soluția y^* se obține prin metode recursive de calcul, cunoscute în optimizarea parametrică drept metode de programare matematică.

În esență, se generează șirul $y(k)$, $k \in N$, după relația de recurență:

$$y(k+1) = y(k) + a_k \cdot d_k \quad (2.26)$$

cu inițializarea $y(0) = y_0$ precizată și unde $a_k \in R$ este pasul de deplasare, iar $d_k \in R$ este direcția de deplasare din punctul curent $y(k)$ în punctul următor $y(k+1)$.

Șirul $y(k)$, $k \in N$ este convergent către valoarea y^* . Soluția problemei de optimizare reprezintă vectorul referințelor optimale $r^* = y^*$ și este transferată nivelului inferior, mai precis reglatoarelor C_i , $i = 1, 2, \dots, n$, din fig. 2.6.

Practica implementării structurilor ierarhizate de conducere dovedește două aspecte importante în legătură cu modalitatea de finalizare a sarcinilor de nivel superior și anume: dacă dimensiunea problemelor de identificare și optimizare este specifică unor sisteme „mari”, problemele de identificare se pot trata în mod unitar, în schimb problemele de optimizare necesită o descentralizare a calculului de optimizare pe subprobleme accesibile algoritmilor tradiționali și echipamentelor uzuale; invers, dacă sarcinile nivelului ierarhic superior sunt de o amploare relativ redusă, pot fi tratate prin asamblare, ca proceduri de agregare și integrare a eforturilor de identificare și optimizare, așa cum se poate constata în continuare.

2.2.4. Descentralizare și coordonare în conducerea proceselor industriale

Sistemele mari, ca obiective ale conducerii, sunt reprezentate prin colecții de subsisteme, care respectă un anumit aranjament și un anumit mod de interconexiune. Aceste sisteme sunt caracterizate prin următoarele trăsături esențiale:

- complexitate, în raport cu funcționalitatea și dimensionalitatea spațiilor variabile de descriere;
- staționaritate, în raport cu regimul de funcționare;
- neliniaritate, în raport cu caracteristicile intrare-ieșire ale unor subsisteme componente.

O reprezentare generală a unui astfel de sistem S este dată în fig. 2.7, în care semnificația notațiilor este:

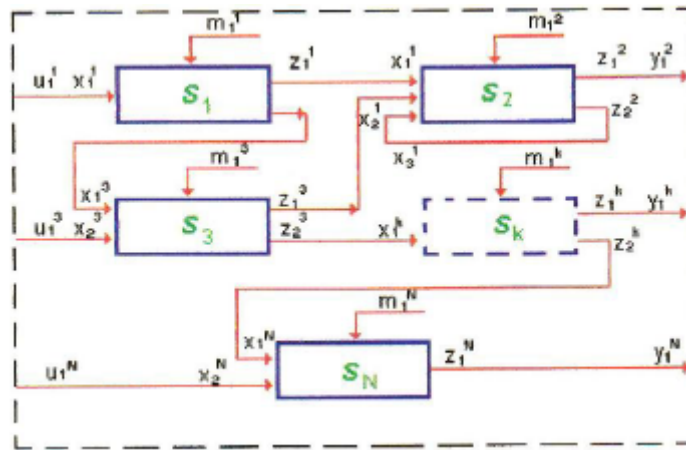


Fig. 2.7. Structura unui sistem de dimensiuni mari.

S_i – subsisteme din S ($i = 1, 2, \dots, N$); u^i – vector de intrare al lui S și al lui S_i ; m^i – vector de comandă al subsistemului S_i ; x^i – vector de intrare de cuplaj al subsistemului S_i ; z^i – vector de ieșire de cuplaj al subsistemului S_i ; y^i – vector de ieșire de apreciere al subsistemului S_i .

El poate fi decupat dintr-un ansamblu de alte sisteme, prin secționarea vectorilor de intrare u și ieșire h . Prin aranjamente particulare ale subsistemelor S_i se pot obține structuri tehnologice tip serie, paralel, cu recirculare etc.

Un subsistem S_i este univoc determinat de relațiile intrare-ieșire:

$$\begin{aligned} y^i &= y^i(m^i, x^i) \\ z^i &= T^i(m^i, x^i); \quad i = 1, 2, \dots, N \end{aligned} \quad (2.27)$$

precum și de restricțiile de interconexiune:

$$x^i = H^i(z^1, z^2, \dots, z^N) \quad (2.28)$$

Funcția criteriu - globală $F(x, m)$ se presupune aditiv - decompozabilă:

$$F(x, m) = \sum f^i(m^i, x^i) \quad (2.29)$$

În relația (2.29), funcțiile $f^i(m^i, x^i)$ sunt construite din ieșirile de apreciere y^i din (2.29), pentru vectorii u^i fixați, iar vectorii x și m sunt definiți prin:

$$\begin{aligned} x &= (x^1, x^2, \dots, x^N) \\ m &= (m^1, m^2, \dots, m^N) \end{aligned} \quad (2.30)$$

Problema globală de optimizare pentru sistemul S se formulează astfel:

$$\begin{aligned} \max_{(x,m)} F(x,m) \\ z^i = T^i(m^i, x^i) \\ x^i = H^i(z^1, z^2, \dots, z^i, \dots, z^N), \text{ unde } i = 1, 2, \dots, N \end{aligned} \quad (2.31)$$

Lagrangeanul problemei descrise de (2.31) este:

$$L = \sum_{i=1}^N f^i(m^i, x^i) + \sum_{i=1}^N (\mu^i)^T (T^i - z^i) + \sum_{i=1}^N (\rho^i)^T (x^i - H^i) \quad (2.32)$$

în care μ^i și ρ^i sunt vectorii multiplicatorilor Lagrange.

Relația (2.32) se poate scrie ca:

$$L = \sum_{i=1}^N f^i(m^i, x^i) + \sum_{i=1}^N (\mu^i)^T (T^i - z^i) + \sum_{i=1}^N (\rho^i)^T (x^i - H^i) = \sum_{i=1}^N L^i \quad (2.33)$$

de unde se observă că lagrangeanul L este aditiv - decompozabil și că problema globală poate fi soluționată prin rezolvarea a N subprobleme de forma:

$$\max_{(m^i, x^i)} f^i(m^i, x^i)$$

cu restricțiile:

$$\begin{aligned} z^i &= T^i(m^i, x^i); \\ x^i &= H^i(z^1, z^2, \dots, z^i, \dots, z^N); \end{aligned} \quad (2.34)$$

integrate în sublagrangeeni L^i .

Avantajele esențiale care rezultă din utilizarea tehnicii de descompunere și coordonare ierarhizată în problemele de conducere optimală sunt materializate prin:

- repartizarea rațională a efortului de calcul, pe baza unei descentralizări complexe la nivele ierarhice de conducere;
- soluționarea problemelor mari de conducere pe configurații de calcul tip multiprocesor, printr-o distribuție pe orizontală, cu repartizarea unui echipament de calcul pentru fiecare subsistem și o distribuție pe verticală după nivele ierarhice de conducere.

Pentru procedeele de coordonare propuse, a fost analizată convergența algoritmilor și au fost precizate criteriile de stabilitate a acestor algoritmi pe baza unui studiu asupra calității unor matrice, construite cu derivatele de ordinul unu și doi ale funcției L .

2.2.5. Integrarea în conducerea proceselor industriale

În cazul unor probleme de identificare și optimizare cu un grad de dificultate al calculului nu prea ridicat, se poate considera o tratare implicită, cu rezolvarea simultană a celor două probleme.

Problemele de identificare și optimizare pentru un proces industrial, cu funcționare în regim staționar, pot fi formulate astfel:

Problema de identificare:

$$\min[I_1 = S(a)] \quad (2.35)$$

cu:

$$S(a) = \sum_{i=1}^N [y_i - H_i(u_i, \alpha)]^T W_i [y_i - H_i(u_i, \alpha)] \quad (2.36)$$

în care u și y sunt mărimi de intrare și respectiv de ieșire ale procesului, iar a , parametrii modelului.

Problema de identificare:

$$\min_u [I_2 = f(u, a)] \quad (2.37)$$

cu restricțiile:

$$g(u, a) \leq 0 \quad (2.38)$$

O viziune unitară a problemelor formulate prin (2.36) și (2.38) este asigurată de următoarea reprezentare:

$$\begin{cases} \min_a S(a) \\ \min_{u,a} f(u, a) \end{cases} \quad (2.39)$$

cu restricțiile $g(u, a) \leq 0$.

Această reprezentare este o problemă bicriterială de optimizare, care se poate exprima în mod echivalent în următoarele formulări implicite:

a) **Formularea parametrică.** Problema (2.38) se transformă în varianta parametrizată:

$$\begin{aligned} \min_{u,a} \{ \lambda S(a) + (1-\lambda)f(u, a) \}; \lambda \in (0,1) \\ g(u, a) \leq 0 \end{aligned} \quad (2.40)$$

Soluția limită pentru (2.37), în ipoteza în care parametrul λ tinde la 1, converge la soluția problemei implicite.

b) **Formularea recursivă, prin partiționare** Procedura partiționării se aplică succesiv pentru mărimile de intrare u și pentru parametrii necunoscuți ai modelului a , după următorul algoritm:

- se consideră o estimatie inițială a intrării u^0 și se rezolvă problema:

$$\begin{aligned} \min_{u,a} f(u^0, a), \\ g(u^0, a) \leq 0, \end{aligned} \quad (2.41)$$

care are soluția u^1 ;

- se repetă succesiv partiționarea între u și a , până se ajunge la perechea (u^*, a^*) , cu care se aproximează soluția problemei implicite (2.37);

c) **Formularea prin restricții adiționale.** Această tehnică presupune introducerea unei restricții suplimentare, construită cu una din funcțiile-criteriu.

Pentru problema (2.41), cel mai util este să se restricționeze funcția pătratică de identificare S (2.39), întrucât se cunoaște că valoarea sa minimă este 0.

Se obține problema echivalentă cu restricții adiționale: $g(u, \alpha) \leq 0$;

$$\begin{aligned} \min_{u,\alpha} f(u, \alpha); \\ \varepsilon < \delta; \delta > 0; S(\alpha) \leq \varepsilon \end{aligned} \quad (2.42)$$

în care: ε este toleranța maximă a erorii, iar δ un număr arbitrar pozitiv.

d) **Formularea pe criteriul min-max.** O alternativă posibilă pentru (2.38) este cea bazată pe strategia min-max din teoria jocurilor. Versiunea min-max pentru problema (2.38) este următoarea:

$$\min_u \{ \max_{\alpha \in A} [f(u, \alpha)] \} \quad (2.43)$$

unde:

$$A = \{ \alpha \in \mathbf{R} / S(\alpha) \leq \varepsilon; \varepsilon < \delta; \delta > 0; g(u, \alpha) \leq 0 \} \quad (2.44)$$

Și presupune că strategiile posibile, cu $S(\alpha) \leq \varepsilon$, joacă împotriva strategiilor adoptate pentru comenzile u , după principiul min - max. Soluția pentru (2.38) este dată printr-o metodă de rezolvare a jocurilor matriciale.

e) **Formularea prin analiza sensibilității funcțiilor de optimizare.** Această abordare se bazează pe examinarea sensibilității funcțiilor criteriu f și S , în raport cu vectorul parametrilor α , comun celor două funcții.

Dacă se acceptă valorile cele mai defavorabile f_w și S_w , corespunzătoare valorilor limită α_w , se construiește criteriul de optimizare:

$$\max \left\{ \left[\sum_k (df)^2 a \frac{1}{f_w^2} + (dS)^2 \frac{1}{S_w^2} \right] \Delta \alpha_i \right\} \quad (2.45)$$

cu restricțiile:

$$g(u^x, \alpha) \leq 0; \Delta \alpha_i = \alpha - \alpha_{iw} > 0; (i = 1, 2, \dots, k) \quad (2.46)$$

unde: $df = \left(\frac{\partial f}{\partial \alpha} \right)_{\alpha_w}$ - sensibilitatea funcției criteriu f , în raport cu α ; $dS = \left(\frac{\partial S}{\partial \alpha} \right)_{\alpha_w}$ - sensibilitatea funcției de identificare S , în raport cu α .

Toate reprezentările problemei (2.46) pot fi rezolvate prin algoritmi de optimizare.

2.3. Probleme de optimizare în automatizările industriale

2.3.1. Optimizări staționare

În procesul de comandă al unor sisteme hidrotehnice se pune problema determinării anumitor valori optime (nivel, presiuni, debite, ...) și a menținerii acestor valori pe perioade de timp relativ îndelungate, fără ca în cadrul acestei probleme să fie oportună și studierea caracteristicilor unor regimuri tranzitorii care apar la eventuale modificări ale valorilor mărimilor menționate. Asemenea optimizări, care au ca scop determinarea valorilor staționare optime pentru anumite mărimi sunt denumite optimizări ale regimurilor staționare sau optimizări staționare.

În unele cazuri, valorile staționare optime sunt determinate prin intermediul unor calculatoare care nu sunt conectate la procesele tehnologice și la echipamentele de reglare automată aferente (calculatoare „offline”), operatorul uman luând măsuri pentru aducerea mărimilor caracteristice la valorile optime, de regulă prin fixarea acestor valori pentru semnalele de referință ale unor bucle de reglare. În alte cazuri, calculatorul care determină valorile optime realizează o conducere în regim de supraveghere a procesului tehnologic, primind direct de la proces o serie de mărimi măsurate și asigurând stabilirea valorilor optime pentru semnalele de referință ale buclelor de reglare menționate.

Evident, în fiecare buclă de reglare vor exista regimuri tranzitorii, provocate de modificarea valorii semnalelor de referință sau de acțiunea unor perturbări, dar optimizarea acestor regimuri constituie o problemă locală a fiecărei bucle, separată de problema de ansamblu a optimizării regimurilor staționare.

Optimizarea staționară are ca obiectiv să determine pentru instalația tehnologică acel regim de funcționare de lungă durată - respectiv acele valori staționare optime ale mărimilor caracterizând funcționarea instalației care să asigure maximizarea ratei anuale de recuperare a investițiilor, în ipoteza unui anumit număr de ore de funcționare pe an (de exemplu, 8,400 ore/an).

Optimizarea staționară își propune ca obiectiv determinarea valorilor staționare optime ale unor mărimi, care vor fi menținute timp îndelungat pentru obținerea unui maxim al criteriului, reprezentat de rata anuală de recuperare a investițiilor, însuși criteriul fiind formulat prin intermediul unui interval mare de timp.

Menținerea anumitor mărimi la valorile optime stabilite se va face prin intermediul unor bucle locale de reglare automată în care vor interveni regimuri tranzitorii, dar eventuala optimizare a acestor regimuri constituie obiectul unei alte clase de probleme.

2.3.2. Optimizări dinamice

După cum se știe, în cazul sistemelor de reglare automată (pentru nivel, sarcină, încărcare), regimurile tranzitorii (care pot fi denumite și regimuri dinamice) sunt provocate de variațiile mărimii de intrare a sistemului, (numită și mărime de referință) sau de variațiile unor mărimi perturbatoare.

În cazul cel mai simplu al unui sistem de reglare automată în funcție de ieșire, monovariabil (cu o singură mărime de referință w și cu o singură mărime reglată), liniar și continuu, schema de elemente are aspectul din fig. 5.8, unde RA este regulatorul automat, EE- elementul de execuție, IT- instalația tehnologica, Tr- traductorul, p_1, p_2, \dots, p_k - perturbările care acționează asupra instalației tehnologice.

Sistemele de reglare în funcție de ieșire au fost primele folosite în practica industrială și se utilizează pe scară largă și în prezent, deși în ultimul deceniu s-a dezvoltat cu succes teoria reglării în funcție de stare.

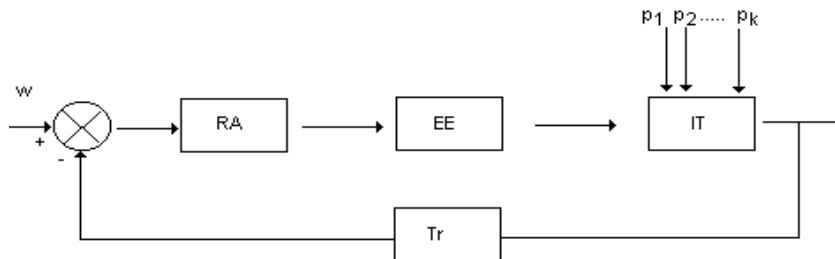


Fig. 2.8. Elementele unui sistem de reglare automată.

Grupând elementele EE, IT și Tr într-un singur bloc echivalent, notat cu F , rezultă pentru schema de elemente aspectul din fig. 5.9., unde F este blocul echivalent al părții fixate a sistemului („fixate” în sensul că, pentru a influența performanțele sistemului, proiectantul automatizării are foarte puține posibilități de a modifica parametrii din blocul F , în raport cu posibilitățile de a alege structura și parametrii regulatorului automat), ε - eroarea, u - mărimea de comandă, y - mărimea reglată, respectiv mărimea de ieșire a sistemului, iar RA, w , p_1, p_2, \dots, p_k au aceleași semnificații ca și în fig. 2.8.

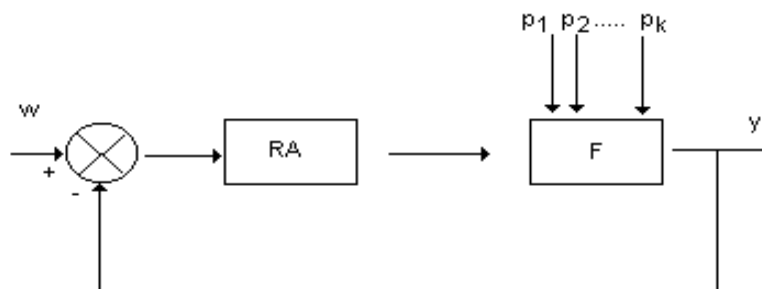


Fig. 2.9. Schema simplificată a unui sistem de reglare automată.

În ipoteza unei valori nule a erorii staționare ε_{st} , regimul tranzitoriu provocat de o variație treaptă a unei perturbări are aspectul din fig. 2.10, în aceeași ipoteză.

Considerând schema din fig. 2.9, din ea se obține:

$$\varepsilon = w - y \quad (2.47)$$

În cazul răspunsului la variația treaptă unitară pozitivă a mărimii de referință w , regimul tranzitoriu ideal al mărimii reglate y ar fi cel în care mărimea reglată ar urmări fidel - și deci instantaneu - variația mărimii de referință, deci pentru mărimea y ar rezulta tot o treaptă unitară pozitivă. O asemenea comportare ideală nu este posibilă. Notațiile σ_1 și σ_2 reprezintă dispersiile mărimii monitorizate.

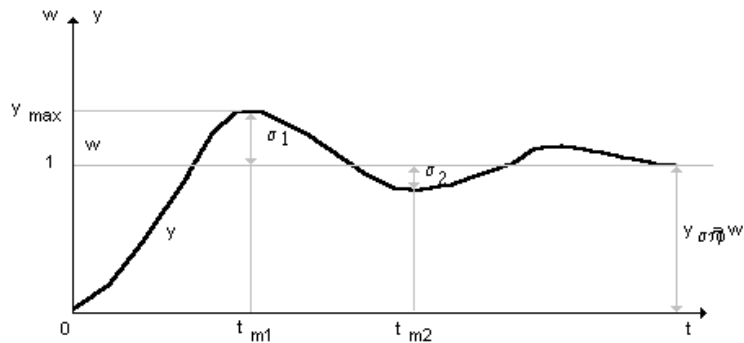


Fig. 2.10. Răspunsul sistemului de reglare automată la variația treaptă a mărimii de referință.

Cu cât însă aria mărginită de curba y și dreapta w va fi mai mică, cu atât regimul tranzitoriu se va apropia de cel ideal. Acest considerent a condus la ideea alegerii unui criteriu de optimizare dinamică legat de minimalizarea ariei respective.

Valoarea ariei este definită de integrala:

$$I = \int_0^{\infty} \varepsilon dt \quad (2.48)$$

dar aceasta expresie nu poate fi folosită pentru elaborarea de criterii, deoarece ordonatele mărimii ε pot interveni cu semne diferite și deci au o valoare redusă pentru a nu atesta prezența unui regim tranzitoriu de calitate bună.

Acest inconvenient poate fi eliminat pe mai multe căi. Cea mai simplă constă în adoptarea unui criteriu de optimizare de forma:

$$J_1 = \int_0^{\infty} \varepsilon^2 dt \quad (2.49)$$

minimizarea acestui criteriu integral pătratic asigurând un optim pentru regimul tranzitoriu în sensul obținerii unei anumite apropieri dorite de regimul ideal.

În mai multe cazuri, criteriile integrale pătratice au un obiectiv combinat, urmărind nu numai optimizarea regimului tranzitoriu, ci și o optimizare care reprezintă un compromis între atingerea primului obiectiv (calitate bună a regimului tranzitoriu) și a celui de al doilea obiectiv care vizează reducerea consumului de energie pentru realizarea acțiunii de reglare automată, respectiv pentru comanda deplasării elementului de execuție.

Reducerea acestui consum poate fi obținută prin adoptarea unui criteriu integral pătratic de forma:

$$J_2 = \int_0^{\infty} u^2 dt \quad (2.50)$$

unde u este mărimea de comandă de la ieșirea regulatorului automat RA (fig. 2.8) - întrucât mărimea u este proporțională cu puterea solicitată la ieșirea regulatorului, proporționalitate care devine evidentă când mărimea u este un curent electric sau o tensiune, cum este cazul reguletoarelor electronice.

Compromisul între obiectivele referitoare la regimul tranzitoriu și la consumul de energie se realizează ușor, adoptând un criteriu combinat, de forma:

$$J_3 = J_1 + \rho \cdot J_2 = \int_0^{\infty} (\varepsilon^2 + \rho \cdot u^2) dt \quad (2.51)$$

unde ρ este un factor de ponderare.

Expresia (2.51) ilustrează unul dintre cele mai simple cazuri de optimizare după mai multe criterii, denumita „polioptimizare”.

În stadiul actual de dezvoltare al instalațiilor tehnologice, care au atins un grad de complexitate foarte ridicat, necesitatea polioptimizării intervine într-un număr din ce în ce mai mare de probleme de optimizare.

Un exemplu imediat poate fi oferit de un sistem de reglare automată multivariabil, cu două mărimi de referință w_1 și w_2 și cu două mărimi reglate y_1 și y_2 , având schema din fig. 2.11. În cadrul regulatorului automat intervin blocurile R_1 și R_2 , care primesc erorile ε_1 și ε_2 ale celor două bucle, iar în cadrul părții fixate F intervin blocurile F_{11} , F_{22} , F_{21} și F_{12} , ultimele două reprezentând căile de transmitere a acțiunilor de interfluență.

Se constată că în calitate de criteriu integral pătratic, pentru optimizare poate fi folosită expresia:

$$J_4 = \int_0^{\infty} \varepsilon_1^2 dt + \int_0^{\infty} \varepsilon_2^2 dt = \int_0^{\infty} (\varepsilon_1^2 + \varepsilon_2^2) dt \quad (2.52)$$

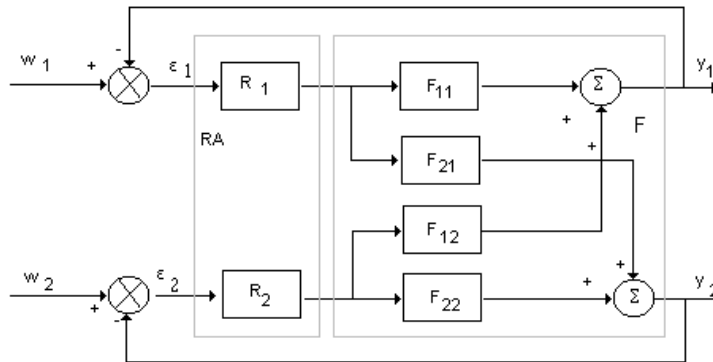


Fig. 2.11. Sistem de reglare automată multivariabil.

2.3.3. Formularea matematică a problemelor de optimizare

2.3.3.1. Probleme de terminologie

Funcții criteriu și restricții

După cum a rezultat din subcapitolele anterioare, optimizarea include:

- alegerea unui criteriu de optimizare;
- stabilirea restricțiilor de tip egalitate și inegalitate;

c) determinarea valorilor optime ale variabilelor (care intervin în expresia criteriului și în expresiile restricțiilor) asigurând un extrem - maxim sau minim, după caz - al criteriului de optimizare.

Notând variabilele care intervin în expresiile criteriului și restricțiilor prin:

$$x_1, x_2, \dots, x_n \quad (2.53)$$

criteriul de optimizare poate fi notat prin funcția:

$$f=f(x_1, x_2, \dots, x_n) \quad (2.54)$$

Această funcție a căpătat mai multe denumiri în literatura tehnică: funcție criteriu, funcție obiectiv, indice de performanță, funcție de cost, funcție de optimizare, funcție scop. Oricare dintre denumiri poate fi utilizată cu egala justificare. În prezenta lucrare va fi utilizată denumirea de funcție criteriu.

Restricțiile de tip egalitate pot fi exprimate prin relații de forma:

$$h_i(x_1, x_2, \dots, x_n) = 0 \quad (2.55)$$

cu $i=1,2, \dots, m$, presupunând că sunt impuse m restricții de acest tip. În unele lucrări restricțiile de tip egalitate sunt denumite și restricții active.

Restricțiile de tip inegalitate pot fi exprimate prin relații de forma generală:

$$l_j \leq g_j(x_1, x_2, \dots, x_n) \leq L_j \quad (2.56)$$

(cu $j=1,2, \dots, r$, presupunând ca intervin r expresii de acest tip), unde l_j și L_j sunt limita inferioară admisă și limita superioară admisă.

În unele cazuri se impune o singură limită, iar uneori aceasta poate fi nulă, rezultând restricții de tip inegalitate de forma:

$$g_j(x_1, x_2, \dots, x_n) \geq 0 \quad (2.57)$$

sau de forma:

$$g_j(x_1, x_2, \dots, x_n) \leq 0 \quad (2.58)$$

Numărul m al restricțiilor de egalitate trebuie să fie mai mic decât numărul n al variabilelor, deci este necesară o relație de forma:

$$m < n \quad (2.59)$$

În situația în care ar avea loc egalitatea $m = n$, atunci nu mai are rost căutarea unui optim, întrucât pentru respectarea restricțiilor instalația tehnologică poate funcționa numai cu anumite valori fixe ale variabilelor.

Dacă numărul restricțiilor de tip egalitate ar fi mai mare decât numărul variabilelor, problema devine în general incompatibilă.

În cazul restricțiilor de tip inegalitate nu intervin condiții de tipul (2.59) întrucât aceste restricții delimitează anumite domenii și deci numărul lor poate fi mai mare decât numărul variabilelor.

2.3.3.2. Formularea optimizării prin intermediul spațiilor vectoriale

Un spațiu S poate fi considerat ca o colecție de elemente. De exemplu, spațiile R^1 și R^2 considerate în paragraful anterior, au ca elemente punctele de pe o dreaptă, respectiv punctele dintr-un plan; în R^2 fiecare punct este definit de două coordonate, care formează un vector. Astfel, pentru punctul x se obține:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (2.60)$$

pentru punctul y se obține:

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \quad (2.61)$$

Într-un spațiu real cu n dimensiuni R^n , fiecare punct este definit de n coordonate, respectiv de vectori n -dimensionali:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (2.62)$$

Un spațiu S este liniar dacă x și y fiind elemente ale spațiului S , respectiv $x \in S, y \in S$, combinația liniară $ax + by$ este de asemenea un element al aceluiași spațiu, respectiv $(ax + by) \in S$, a și b fiind constante scalare.

Un spațiu este complet dacă orice șir Cauchy format cu elemente ale spațiului are limita șirului în spațiul respectiv.

Un spațiu are un produs scalar dacă în spațiul respectiv este definită o operație care atribuie un număr real fiecărei perechi de elemente ale spațiului. Considerând vectorii din (2.62) și notând prin x^T vectorul x transpus, vom avea: $x^T = [x_1, x_2, \dots, x_n]$.

Se constată că produsul:

$$x^T y = [x_1, x_2, \dots, x_n] \cdot \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = x_1 y_1 + x_2 y_2 + \dots + x_n y_n = \sum_{i=1}^n x_i y_i \quad (2.63)$$

atribuie un număr real unei perechi de elemente din R^n (întrucât $\sum_{i=1}^n x_i y_i$ este un număr real), deci reprezintă un produs scalar. După cum se cunoaște, produsul scalar a doi vectori, denumit și produs intern, se notează prin $\langle x, y \rangle$.

Produsul:

$$x^T y = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \cdot [y_1, y_2, \dots, y_n] = \begin{bmatrix} x_1 y_1 & x_1 y_2 & \dots & x_1 y_n \\ x_2 y_1 & x_2 y_2 & \dots & x_2 y_n \\ \vdots & \vdots & \vdots & \vdots \\ x_n y_1 & x_n y_2 & \dots & x_n y_n \end{bmatrix} \quad (2.64)$$

se numește produs extern (uneori este denumit și produs diadic) și se notează prin $\langle x, y \rangle$.

În problemele de optimizare sunt utilizate frecvent spațiile Hilbert. Un spațiu Hilbert este un spațiu liniar, complet și având un produs scalar; exemple de spații Hilbert sunt spațiile reale R^1, R^n și spațiul funcțiilor de pătrat integrabil.

Considerând că variabilele x_1, x_2, \dots, x_n din funcția criteriu sunt coordonatele unui vector x în R^n , optimizarea constă în a găsi acel vector x^* pentru care funcția are un minim, deci pentru care rezultă:

$$f(x^*) \leq f(x) \quad (2.65)$$

pentru orice x din vecinătatea lui x^* .

După cum va rezulta și din capitolele ulterioare, metodele numerice de optimizare – care permit utilizarea calculatoarelor numerice - prevăd construirea unei secvențe (șir) de vectori x^0, x^1, x^2, \dots astfel încât să se obțină:

$$f(x^{i+1}) < f(x^i) \quad (2.66)$$

Construirea șirului continuă fie până când nu mai poate fi găsit un vector x^{i+1} care să satisfacă relația (2.66) - și atunci $x^i = x^*$, fie până când șirul de vectori se apropie de o limită, această limită fiind chiar x^* .

În practică șirul de vectori este obținut printr-o relație iterativă de forma:

$$x^{i+1} = x^i + \alpha_i p^i \quad (2.67)$$

unde α_i este un scalar, iar p^i este un vector din același spațiu cu x^i (vectorul p^i determină direcția căutării optimului, iar scalarul α_i asigură pasul căutării).

Avantajele utilizării spațiilor Hilbert pentru optimizare apar evidente. Astfel, spațiul Hilbert fiind liniar, din (2.67) rezultă ca x^{i+1} va fi un element al spațiului din care fac parte x^i și p^i , întrucât x^{i+1} reprezintă o combinație liniară a vectorilor x^i și p^i , deci întregul șir de vectori va fi conținut în spațiul respectiv.

Pe de altă parte, spațiul Hilbert fiind complet, limita șirului de vectori se va găsi în spațiul respectiv, ceea ce este evident de dorit.

În sfârșit, faptul că în spațiul Hilbert este definit un produs scalar are o importanță deosebită pentru determinarea vectorului p^i din (2.66), care asigură direcția căutării optimului. Produsul scalar permite definirea conceptului de ortogonalitate, doi vectori x și y fiind ortogonali dacă produsul lor scalar este nul, deci dacă este îndeplinită condiția $\langle x, y \rangle = 0$.

2.3.3.3. Metode de calcul pentru optimizarea fără restricții

Problemele de optimizare întâlnite în practică sunt probleme cu restricții, dar metodele de calcul pentru optimizarea fără restricții sunt importante prin faptul că pot fi folosite cu succes după transformarea problemelor de optimizare cu restricții în probleme de optimizare fără restricții (transformare din categoria TR).

Principalele categorii de metode de calcul:

După cum se știe, în cazul funcțiilor $f(x)$ de o singură variabilă - x fiind deci scalar - condițiile necesare și suficiente de extrem se stabilesc prin intermediul primei derivate $f'(x) = df(x)/dx$ și al celei de a doua derivate $f''(x) = d^2f(x)/dx^2$.

Condiția necesară pentru un extrem (maxim sau minim) este anularea primei derivate:

$$f'(x) = 0 \quad (2.68)$$

reprezentând condiția de staționaritate.

Pentru un maxim, condițiile necesare și suficiente sunt (2.68) împreună cu condiția de convexitate:

$$f''(x) < 0 \quad (2.69)$$

Pentru un minim condițiile necesare și suficiente sunt (2.68) împreună cu condiția de convexitate:

$$f''(x) > 0 \quad (2.70)$$

Condiția de staționaritate este îndeplinită în punctul optim x^* , iar condiția de convexitate este îndeplinită în jurul acestui punct.

Relațiile menționate sunt valabile pentru funcții $f(x)$ cel puțin de gradul doi, respectiv nu sunt valabile pentru funcții de gradul întâi, liniare, de forma $f(x) = ax + b$, întrucât la acestea rezultă $f'(x) = a$ și deci prima derivată nu depinde de variabila x .

Datorită acestui fapt, metodele de programare neliniară - folosite pentru optimizarea funcțiilor criteriu neliniare $f(x)$, unde x este un vector - nu pot fi folosite pentru rezolvarea problemelor de programare liniară, cu funcții criteriu și restricții liniare.

În cazul optimizării funcțiilor criteriu $f(x)$, de variabilă vectorială, principalele categorii de metode de calcul se disting după faptul că unele fac apel la primele derivate, altele necesită derivatele prime și secunde, iar o a treia categorie nu necesită nici determinarea primelor derivate, nici a derivatelor secunde; cele trei categorii de metode aproximează deci dezvoltarea în serie Taylor a funcției $f(x)$ prin reținerea unui număr diferit de termeni.

Metodele din prima categorie pot fi denumite metode de gradient sau metode bazate pe prima variație, metodele din a doua categorie pot fi denumite metode Newton sau metode bazate pe a doua variație, iar metodele din a treia categorie sunt denumite metode de căutare directă sau metode directe, întrucât asigură optimizarea apelând numai la valorile funcției criteriu.

2.4. Funcțiile unui sistem informatic de proces pentru noduri hidrotehnice

2.4.1. Funcțiile de bază

Un sistem informatic de proces pentru un sistem hidroenergetic și de navigație realizează măsurarea, gestiunea și comanda sistemului pornind de la măsurători executate la nivel de ansamblu și continuă cu măsurători în detaliu, Operațiile de reglare (ajustare parametri) se execută de la nivel de detaliu spre ansamblu.

Aceasta presupune pentru partea de măsură, efectuarea unor măsurători la nivelul mărimilor de intrare în sistem (la sursă), apoi măsurători în punctele (nodurile de rețea) importante, la traductorii de poziție, de nivel, continuând cu măsurători în instalațiile electrice și hidraulice și terminând cu măsurarea parametrilor de interes pe partea de înregistrare de date.

În privința reglajului acesta se face în sens invers, se reglează parametrii la nivelul ecluzei și al porților batante, se așteaptă stabilizarea procesului, se continuă cu reglaje în punctele sau în instalațiile monitorizate.

Toate referirile din acest capitol sunt în directă legătură cu aplicațiile dezvoltate de către autor pentru S.C. HIDROTIM S.A., atât la Porțile de Fier II, cât și în alte sisteme de ecluzare.

Funcțiile de bază ale unui sistem informatic pentru rețele de transport energie termică sunt:

- a) Funcția de teleurmărire;
- b) Funcția de telegestiune;
- c) Funcția de teleconducere.

2.4.1.1. Funcția de teleurmărire

Această funcție realizează în ansamblu măsurarea valorilor instantanee a tuturor parametrilor de interes și le transmite punctului dispecer sau altui post de lucru racordat la sistem.

Prin telemărire se înțelege practic măsurarea de la distanță, în timp real a anumitor mărimi.

Pentru un sistem informatic complicat, implementat pe un nod hidrotehnic, unde numărul parametrilor măsurați este mare, funcția de telemărire are mai multe componente:

- *telemărire parametri la intrarea în sistem*
Sunt telemăriți parametri de: temperatură, presiune, debit, poziție, nivel, la intrarea apei în ecluză, pentru stabilirea procedurii de ecluzare.
- *telemărire parametri în instalațiile sistemului*
Parametrii telemăriți sunt: mărimi de debite și presiuni precum și poziția organului de reglare, temperatura exterioară. Acești parametri sunt de regulă folosiți la echilibrarea hidraulică. Se mai măsoară și mărimi electrice (curent, tensiune, putere, frecvență), în scopul evitării supraîncărcării motoarelor pompelor.
- *telemărire parametrilor în zona acționărilor porților*
Această funcție presupune telemărirea tuturor parametrilor de acționare a porților ecluzei. Parametrii măsurați sunt:
 - debit, presiune, temperaturi pentru apa din ecluză;
 - stare echipamente, pompe;
 - poziție organe de reglare;
 - poziție poartă (unghiuri, distanțe);
 - solicitări mecanice în instalație (eforturi, deformări);

După modul cum sunt prezentate aceste informații în ideea de a oferi operatorului de serviciu (dispecer) detalii care pot fi comparate și analizate comod, aceste funcții pot fi împărțite în:

- 1) telemărire cu afișare tablou;
- 2) telemărire cu afișarea parametrilor pe scheme simplificate;
- 3) telemărire cu afișare grafică;
- 4) telemărire în care parametrii sunt afișați sub formă de bargrafuri.

1. Telemărirea cu afișare tablou are trei subcomponente de bază

1.a. Telemărirea parametrilor apei în ecluză

- temperatura;
- debit;
- cotă (nivel);
- diferențe nivel;

1.b. Telemărirea parametrilor instalațiilor electrice:

Sunt afișate într-un tabel următoarele mărimi instantanee (de la toate punctele esențiale):

- tensiuni;
- curenți;
- puteri;
- starea și timpul de funcționare a căilor de comunicație către dispecer.

1.c. Starea pompe și starea circuitelor hidraulice.

Sunt afișate într-un tabel următoarele mărimi instantanee (de la toate punctele esențiale):

- încărcare;
- debite prin diverse secțiuni esențiale;
- turație;

- putere;
- consum electricitate;
- stare electrovalve;
- stare actuatoare;
- stare metrologică și operațiunile de mentenanță care trebuie efectuate.
- presiuni în diverse puncte esențiale.

2. Telemăsurarea și teleurmărirea cu afișarea parametrilor pe scheme simplificate.

Aceste subfuncții au o formă grafică evoluată care pornește de la afișarea schemei tehnologice simplificată, în primul rând a circuitelor de echilibrare, de unde cu ajutorul unei proceduri de căutare se poate selecta orice punct unde se dorește vizualizarea unei anumite mărimi.

Un circuit de interes este reprezentat printr-o schemă tehnologică simplă, pe care sunt figurate traseele conductelor, poziția pompelor, electrovalvelor, actuatorilor, locul de amplasare a punctelor de măsură, precum și sensurile de circulație pentru mărimile măsurate.

Pe această schemă sunt afișate valorile parametrilor, primite la ultima scanare, acest fapt este dovedit de timpul afișat (oră, minut, secundă).

Fiecare mărime analogică afișată este prevăzută cu un câmp în care sunt semnalizate stările de depășire a limitelor tehnologice prescrise (protecție, avarie, traductor defect), precum și cu unitățile de măsură aferente.

3. Telemăsurarea cu afișare grafică Fiecare mărime telemăsurată are asociat un grafic care are pe axa „y” domeniul de măsură, iar pe axa „x” timpul (ziua și ora). Folosind o procedură specială, tehnica ferestrelor, se poate urmări evoluția în timp a oricărui parametru pe durata unei zile, unei ore sau a unui minut.

Această funcție are și posibilitatea teelurmării evoluției în timp a oricărui parametru. Fiecare parametru are asociată o arhivă unde sunt trecute:

- graficele de evoluție pe durata unei zile, luni sau chiar ani;
- valori instantanee la anumite ore pe durate de timp practic nelimitate;
- valori medii precum și limite extreme (maxime și minime).

Toate mărimile afișate pot fi listate la o imprimantă.

4. Telemăsurarea cu afișare sub formă de bargrafuri

Fiecărei mărimi telemăsurate i se poate asocia un bargraf. Afișarea cu ajutorul bargrafurilor permite o urmărire comodă prin compararea nivelului de umplere a bargrafelor respective, fără a mai citi valoarea mărimii.

Afișarea cu bargrafuri este în general utilizată atunci când se urmărește evoluția mai multor parametri și când valorile acestora trebuie să fie menținute apropiate.

2.4.1.2. Funcția de telegestiune

Sistemele informatice de proces implementate realizează funcția de telegestiune în două moduri:

- a) Mărimile telemăsurate sunt utilizate în mod direct sau indirect în calculul mărimilor de gestiune (cantitate, energii). Această modalitate este utilizată atunci când traductoarele primare nu au posibilitatea transmiterii automate a indecșilor de debit și energie;

b) Calculatorul de la punctul control primește pe lângă mărimile instantanee, la intervale de timp prestabilite, și informații cumulate pe acea perioadă de timp (cantități, energii, timp). În acest caz calculatorul face calculul energiei și volumului utilizând datele primite ca valori instantanee, calcule care apoi le compară la sfârșitul zilei sau lunii cu mărimile cumulate în contoarele locale și le teletransmite la punctul central.

Fiecare valoare cumulată (cantitate, energie), indiferent de modul în care a fost obținută (calculată sau teletransmisă) are obligatoriu asociat și timpul pentru care s-a făcut această integrare.

Această dublă posibilitate de obținere a cantităților, energiei și timpului este valabilă numai pentru circuitul de la sursă până la inclusiv punctul de măsurare.

În general, mărimile preluate de pe circuitele secundare sunt mărimi cumulate (mărimi de debite apă, energie consumată, volum, masă, etc.).

Funcția de telegestiune este găsită, pe un calculator de dispecer sau supervisor, în evidența de „Rapoarte”. Aceste rapoarte cuprind toate valorile de funcționare și facturare, valori care sunt grupate pe categorii specifice.

La calculatorul „Dispecer” comanda „Rapoarte” are următoarea configurație:

- rapoarte de zi;
- rapoarte la decadă;
- rapoarte la lună;
- rapoarte anuale;
- arhivă.

Rapoartele de zi, decadă și lună cuprind toate valorile de facturare pentru aceste perioade. Ziua energetică este marcată de ora „0”. Valorile arhivate la nivel de zi, decadă, lună formează o bază de date specializate și pot fi modificate numai de persoane autorizate, după un protocol bine protejat. Acestea pot fi considerate ca bază de facturare pentru procedura de ecluzare.

Valorile din aceasta bază de date pot fi exportate în alte programe, cu ajutorul cărora se pot genera orice tip de document, cumulativ sau pe obiect, facturi etc.

2.4.1.3. Funcția de teleconducere

Această funcție este foarte dezvoltată în aplicațiile hidrotehnice. Există și o componentă off-line care a devenit preponderentă în ultimul timp. Datorită dimensiunilor și modului de interdependență a parametrilor dintr-un circuit hidrotehnic, trebuiesc remarcate două aspecte importante ale funcției de teleconducere:

1. Acționarea la distanță a elementelor de reglaj și din nodurile rețelelor hidraulice. Acest tip de reglaj este de tip off-line și are ca scop schimbarea unor regimuri de funcționare (regim de vară, iarnă, regim de avarie etc.) prin acționarea elementelor de reglaj de la sursă, precum și modificări ale regimului hidraulic datorită schimbărilor care apar în momentul izolării sau cuplării unor zone din rețea - aceste comenzi fiind făcute asupra elementelor de reglaj din nodurile respective ale rețelei. Acest tip de teleconducere poate fi interpretată ca o funcție de ajustare globală a parametrilor rețelei.
2. Acționarea de la distanță a elementelor de reglaj din puncte / scheme hidraulice prin transmitere de la punctul dispecer a unor valori de consemn (referințe).

Într-un circuit hidraulic, buclele de reglare sunt:

- reglare debite;
- închidere – deschidere electrovalve;
- reglare presiuni.

Parametrul reglat în fiecare caz cazuri este nestaționar. Pentru a nu perturba sistemul, fiecare punct de reglaj este prevăzut cu un element de menținere la o valoare constantă a căderii de presiune pe punct.

Buclele de măsură menționate sunt proiectate să funcționeze individual. Întrucât buclele de măsură se pot condiționa reciproc (îndeosebi atunci când sursa nu poate satisface consumul), operatorul de la punctul dispecer poate transmite pentru fiecare regulator semnale de referință care să aibă ca efect funcționarea optimă a unui regulator prin diminuarea parametrilor celuilalt.

Acest tip de reglaj este impus de obicei și de funcționarea intermitentă a circuitului de echilibrare hidraulică.

2.4.2. Funcțiile speciale

Pentru a putea utiliza volumul foarte mare de informații pe care un sistem informatic le poate oferi, a apărut necesitatea creării unor funcții speciale, care să grupeze informații pe diferite criterii (tehnologice, date operative, date de sinteza etc.) și apoi să li se dea o destinație precisă, dispecer, director coordonator, inginer de sistem etc.

Pornind de la aceasta observație se pot distinge trei grupe de informații:

1. informații operative - care cuprind totalitatea datelor instantanee citite (în timp real), informații privind sensul de evoluție al unor parametri, precum și starea acestor parametri la anumite perioade de timp (tabel ... de date sau foaie de date). Aceasta funcție se identifică, în baza datelor de intrare, cu funcțiile de teleurmărire și teleconducere în totalitate și parțial cu funcția de telegestiune. Utilizarea acestor informații face obiectul funcției de inginer dispecer.
2. informații de analiză și sinteză, calcule de cantități și energii, randamente, calcule de bilanț etc. Aceste informații fac în mare obiectul funcțiilor de telegestiune și teleconducere.
3. informații cu privire la starea sistemului; parametrii de funcționare ai punctului central, integritatea căilor de comunicație, starea de funcționare a traductoarelor primare. De asemenea în această categorie se pot înscrie și informațiile obținute după prelucrări speciale ale semnalelor primite direct de la traductoare. Aceste informații se referă la: nivel și clasă de precizie, rata defectelor și cronologia apariției lor la diferite tipuri de traductori, calcule de mărimi prin metode indirecte etc.

Aceste trei clase mari de informații au condus la crearea a trei utilizatori diferiți având următoarele funcții:

- Funcția de inginer dispecer;
- Funcția de analiză și supervizare;
- Funcția de inginer sistem și AMC.

2.4.2.1. Funcția de inginer dispecer

Inginerul dispecer conduce operativ întreg sistemul de telemăsura și telegestiune. Totalitatea informațiilor formează în fapt baza de date pentru inginerul dispecer.

Un sistem de telegestiune și teleconducere grupează aceste informații pe categorii de instalații (electrice, hidraulice), pe categorii de mărimi, etc., creând o interfață comodă și prietenoasă între sistem și operatorul uman, astfel încât luarea unor decizii privitoare la funcționarea sistemului să se facă imediat după analiza unei singure imagini tehnologice sau a unui tabel de date de sinteză.

Principalele atribute ale acestei funcții sunt:

- urmărește încadrarea în cotele de energie alocate fiecărui consumator;
- urmărește încadrarea în limitele prescrise a tuturor parametrilor tehnologici măsurați;
- optimizează randamentele utilajelor din exploatare (pompe, electrovalve, actuatori) prin intervenții off line date de la dispecer sau din punctele specifice;
- urmărește menținerea unui regim hidraulic calculat (optim);
- urmărește respectarea programului de ecluzare;
- urmărește evoluția parametrilor de interes și poate dispune efectuarea de manevre în caz de avarie sau modificări în regimul de funcționare;
- memorează evoluția în timp a tuturor parametrilor de interes și poate oferi grafice cu abaterile valorilor acestor parametri față de valorile prescrise;
- urmărește menținerea valorilor parametrilor din sistem după diagrama de reglaj;
- permite reglarea prin acțiune manuală sau automată a parametrilor de poziție, nivel, debit sau presiune.

2.4.2.2. Funcția de analiză și supervizare

Această funcție nu are un regim operativ, ea este destinată compartimentelor de conducere (director exploatare, director tehnic). Funcția a fost creată pentru a oferi o informație globală asupra stării întregului sistem la un moment dat, pentru a oferi informații despre istoricul anumitor parametri precum și prognoza evoluției acestora într-un timp foarte scurt și fără ca operatorul uman să aibă cunoștințe deosebite de informatică sau sisteme de măsurare.

Această funcție este deosebit de utilă și compartimentelor care se ocupă cu facturarea energiei, a serviciilor prestate și cu activitatea de „mers economic”.

Funcția de analiză are ca date de intrare informațiile pe care sistemul de telegestiune le păstrează în evidența de „Rapoarte”. Această funcție poate fi exportată pe orice alt calculator, în configurația: rapoarte de consum, rapoarte de producție, randamente, chei de control, grafice comparative și de reglaj etc., dar oricare ar fi suportul hardware, utilizarea acestei funcții poate fi făcută numai prin preluarea datelor (automat - legătură serială sau prin rețea, manuală - prin diskete, stick usb, etc.) de la calculatorul „dispecer”.

Principalele atribute ale acestei funcții sunt:

- efectuează calcule de bilanț și mers economic;
- efectuează telemăsurarea tuturor parametrilor sau numai a anumitor parametri prin selecție, la comanda operatorului uman la orice moment de timp;
- prezintă selectiv sau în totalitate rapoarte de producție sau rapoarte de consum.

2.4.2.3. Funcția de inginer de sistem și ghid operator AMC

Această funcție este destinată inginerului de sistem, precum și inginerului AMC (Aparatură de Măsură și Control). Este o funcție nouă, pentru un sistem informatic de proces, oferind informații deosebit de utile, care privesc mai multe aspecte.

1. Starea de funcționare a tuturor traductoarelor, la un moment dat

Pentru un sistem de măsură clasic, distribuit pe o arie geografică medie, așa cum sunt cele implementate pe rețeaua hidraulică a unui sistem hidrotehnic, supravegherea funcționării traductoarelor de măsură reprezintă o problemă tehnică deosebit de grea.

Sistemul informatic de proces rezolvă aceste probleme astfel:

- fiecare mărime analogică telemăsurată are atribuită, în formatul de afișare, un câmp special în care sunt trecute informații cu privire la starea traductorului de măsură, astfel:
 - "T" - traductor ieșit din domeniul de măsură-limită superioară ($I > 20 \text{ mA}$, $U > 10 \text{ V}$);
 - "t" - traductor ieșit din domeniul de măsură-limită inferioară ($I < 4 \text{ mA}$, $U < 2 \text{ V}$);
 - "?" - traductor defect electric (traductor decuplat de la sistemul de măsură);
 - "m" - traductor de la care nu s-a prelevat semnal o perioadă de timp mai mare de 30 minute;
 - "!" - traductor defect - mărime măsurată necorelată.

2. Rata defectelor și cronologia apariției lor, la diferite tipuri de traductoare

Orice tip de defect semnalat, la un traductor, este memorat după o procedură preliminară de validare. Validarea este făcută automat de calculator sau de către operatorul AMC. Orice defect semnalat la punctul dispecer poate fi analizat la fața locului (prin efectuarea de măsurători directe), și apoi informația poate fi validată.

Toate stările de defect validate și memorate de calculator formează o bază de date specializată, care poate fi transferată pe un alt calculator sau gestionată de calculatorul „dispecer”.

Un program special ordonează aceste date și oferă informații cu privire la:

- rata defectelor, la un anumit traductor;
- cronologia apariției unui defect la un anumit traductor;
- cronologia apariției aceluiași defect la mai multe traductoare;
- care sunt cele mai frecvente defecte și cauza apariției lor.

3. Ieșirea din clasa de precizie a unor traductoare, detectarea și corecția erorilor de măsură.

Depistarea acestui defect este un procedeu practic imposibil fără un sistem informatic bine proiectat.

2.5. Structura hardware a unui sistem informatic de proces (dispecerat hidrotehnic)

Așa cum a fost descris în prealabil, din punct de vedere hardware, un sistem informatic dedicat pentru rețele hidrotehnice este organizat pe trei nivele ierarhice. (fig. 2.12).

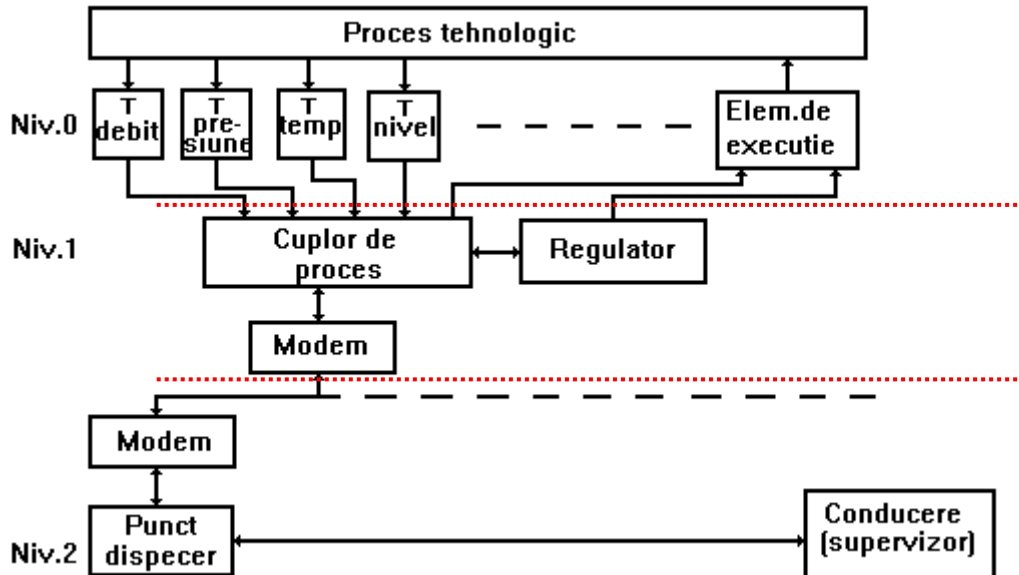


Fig. 2.12. Structura hardware a unui sistem informatic de proces.

Nivelul „0” sau nivelul de bază este format din totalitatea traductorilor și elementelor de execuție conectate la procesul urmărit și condus. Acest nivel este localizat la baza piramidei schemei de reglaj.

Nivelul „1” este destinat cuplorului de proces, care este de fapt un calculator numeric care preia, prin fir fizic, informațiile de la traductoare și le transmite într-un format special și după un cod anume, la calculatorul central „Dispecer”, după eventuale prelucrări, prin rețeaua telefonică sau prin undă radio. Tot la acest nivel pot fi plasate și căile de comunicație, de la traductori la cuplor și de la cuplor la punctul dispecer.

Nivelul „2” are ca suport fizic echipamentele de calcul (calculator, imprimantă, MODEM, UPS) de la punctul dispecer și de la nivelul de conducere sau supervizare.

Acest tip de sistem reprezintă o soluție modernă, cu posibilitatea de a fi verificat și autorizat metrologic. Implementarea unui astfel de sistem informatic de proces s-a făcut pornind de la necesitatea de a urmări și de a conduce de la distanță procesul de producere, transport și furnizare a apei și energiei termice, procesul având o complexitate medie sau mare. Scopul este acela de a ajunge la un management eficient al resurselor, al consumurilor energetice și al producției. O astfel de implementare impune utilizarea unui sistem ierarhic organizat pe cele trei nivele logice definite anterior.

Vom detalia în continuare elementele fizice care intră în configurația fiecărui nivel:

- **Primul nivel**, numit și „**nivelul 0**”, este constituit din elementele primare care interacționează cu procesul. La acest nivel are loc atât automatizarea procesului, ceea ce presupune realizarea unor funcții de reglare a principalelor variabile caracteristice, cât și preluarea și măsurarea unor mărimi de interes. Nivelul „0” este constituit din:
 - traductoare: de debit, de temperatură, de presiune, de nivel, alte elemente de măsurare / detectare / analiză;

- dispozitive electronice de prelucrare, automatizare și afișare locală în teren : integratoare de debit și volum, unități de comandă și reglare automată etc.;
 - elemente primare și de execuție: robinete, vane de reglare, motoare de comandă, pompe, regulatoare diferențiale de presiune, etc.
- **Al doilea nivel** ierarhic numit și „**nivelul 1**” este destinat cuplului de proces. La acest nivel se realizează preluarea de la elementele primare (traductoare, integratoare, pompe, etc.) a semnalelor analogice, în impulsuri sau în frecvență și convertirea acestor semnale în date numerice care pot fi prelucrate elementar și care pot fi ordonate într-un format special, pentru vizualizarea locală într-un centru local operativ (uzual în dulapul de automatizare) și facilitarea transmisiei informației utile la calculatorul central al Dispecerului (care poate să se afle la orice distanță de toate centrele locale operative).

Tot la acest nivel se pot include:

- cartelele de comunicație: cartele MBus pentru traductoarele specifice, transmițătoare Cyble Pulsar sau Mbus pentru debitmetre de apă, convertoare MBus/RS232, adaptoare 4-20 mA/mbus, dispozitive de achiziție și retransmitere etc.;
 - modem-uri radio sau telefonice;
 - mediul fizic de transmisie: cabluri adecvate de legătură între punctele de măsură și centrele operative (PT, CT).
- Având în vedere complexitatea acestui sistem precum și răspândirea elementelor de măsură și control pe o arie geografică destul de extinsă, transmisia datelor de la elementele de măsură înspre calculatoarele de proces (sau controller-ele specifice) precum și a ordinelor de la elementele de comandă la elementele de execuție necesită existența unor interfețe performante.
- Cele mai des întâlnite interfețe de comunicare cu procesul sunt:
- Interfața în curent unificat 4 - 20 mA (transfer analogic);
 - Interfața RS232;
 - Interfața RS422 și interfața (bucla de curent multipunct) RS 485;
 - Interfața M-BUS;
 - Interfețele MOD-BUS, PROFI-BUS și LON.
- **Al treilea nivel** ierarhic al unui sistem informatic de proces îl constituie echipamentul de supervizare sau conducere. El are posibilitatea comunicării cu toate elementele componente ale nivelului „1” și având rolul unui „master” în acest sistem. Nivelul „3” este reprezentat de:
- calculatorul central „Dispecer” pentru urmărirea și controlul global al datelor din sistem;
 - rețea neoperativă de calculatoare PC pentru informare.

Calculatorul supervisor poate stoca cantități mari de informație. Informațiile se referă la istoria procesului și la toate mărimile esențiale ce trebuie urmărite. Acest nivel oferă operatorului de sistem facilități extinse de introducere și extragere date, formându-se astfel o bază de date cu informații utile pentru teleurmărire și teleconducere.

Funcțiile principale ale acestui sistem informatic sunt:

- achiziția, transferul datelor și emiterea comenzilor;
- întreținerea și actualizarea permanente a unei baze de date, care conține atât valori momentane cât și medii, contoare, date de evoluție și valori calculate;
- afișarea grafică a schemelor sinoptice, pe care sunt figurate valorile momentane ale mărimilor analogice și starea agregatelor. Depășirile de limite trebuie figurate prin culori vii, clipitoare;
- crearea și vizualizarea diagramelor de evoluție pentru mărimile analogice sau calculate;

- înregistrarea cronologică a evenimentelor (depășiri / reveniri în limite ale mărimilor analogice, schimbări de stare ale mărimilor binare);
- realizarea automată de calcule pentru determinarea energiei electrice, a apei, debitelor de apă, a presiunilor, echilibrărilor;
- realizarea de medieri și contorizări;
- editarea rapoartelor periodice la oră, zi și luna, care conțin date sintetice despre regimul de funcționare a instalațiilor tehnologice dispecerizate;
- arhivarea datelor timp de un an. Este o facilitate utilă pentru realizarea analizelor de regimuri și a diverselor statistici legate de funcționarea pe termen lung a instalațiilor ;
- tipărirea la imprimantă color a rapoartelor și graficelor de evoluție;
- mecanism de alarmare vizuală și prin voce, în situațiile anormale de funcționare;
- introducerea manuală a unor parametri, necesari pentru calcule;
- posibilitatea definirii unor calcule specifice, export către Excel;
- acces din rețea, cu mai multe posturi active simultan;
- autodiagnoza stării de funcționare a echipamentelor sistemului și a liniilor de comunicație.

2.6. Concluzii

Domeniul sistemelor hidrotehnice este un domeniu care poate beneficia de avantajele tehnicilor moderne de reglare și de automatizare, mai ales având în vedere lipsa de multe ori a acestor sisteme și gradul de uzură ridicată a celor existente.

În cazul sistemelor automate conducerea proceselor efectuându-se fără intervenția directă a omului, mijloacele prin care aceasta se realizează - inclusiv cele care se referă la funcția de informare - se modifică în concordanță cu noile condiții.

În consecință, operațiile de măsurare în sistemele automate sunt efectuate cu traductoare, dispozitive care stabilesc o corespondență între mărimea de măsurat și o mărime cu un domeniu de variație calibrat, aptă de a fi recepționată și prelucrată de echipamentele de conducere (regulatoare, calculatoare de proces).

Chiar dacă noua generație de aparatură și instalații de automatizare utilizate în sistemele hidrotehnice reprezintă un progres incontestabil, obținerea informațiilor despre procesul comandat se mai face încă, în multe situații, în mod tradițional.

Elementele de măsurare din prezent nu mai corespund nici ca gabarit, nici parametri de conectare (interfață pentru transmiterea informațiilor) și nici ca performanțe. Mai „domină” încă semnalul analogic de curent continuu, iar tehnologia de fabricație a elementelor de măsurare provine în cea mai mare parte din domeniul mecanicii fine.

Toate considerațiile și relațiile stabilite pentru analiza și sinteza sistemelor de reglare pornesc de la premisa ca erorile pe care le-ar putea introduce traductorul sunt neglijabile. În consecință, devine evident importantă precizia acestuia pentru problema reglării și pentru conducerea automată în general.

În concluzie, se poate afirma că traductoarele sunt elemente componente tipice ale sistemelor automate, prin intermediul cărora se realizează funcția informațională și că ele trebuie să întrunească o serie de calități care să le apropie de caracteristicile ideale de liniaritate, dinamică și precizie pentru a asigura valabilitatea ipotezelor și relațiilor matematice pe baza cărora sunt formalizate problemele de conducere automată a proceselor. În realitate traductoarele, ca și elementele de execuție, sunt

unități constructive distincte dispuse pe cele două căi de interconectare între procesul propriu-zis și elementele sistemului de conducere cu calculator de proces.

Chiar dacă traductoarele nu au făcut obiectul în sine al acestei lucrări, alegerea acestora trebuie să fie făcută luând în considerare aspecte legate de fiabilitate, integrabilitate, flexibilitate, performanțe, costuri, etc.

Indiferent de localizarea automatizării, în ansamblul întregului sistem hidrotehnic și de navigație, putem spune că NU există un algoritm unic, predestinat a rezolva orice problemă care ar putea apărea în întregul ansamblu. Mai mult chiar, nu putem afirma nici existența unor soluții tehnice unice și general valabile, bazate pe aplicarea unor principii, metode sau raționamente din domeniul ingineriei de sistem.

Domeniul automatizării este, la ora actuală, într-o dinamică fără precedent, fiind impulsionat pe de-o parte de evoluția echipamentelor „hardware”, dar mai ales de apariția unor noi concepte privind componenta „software”, adică în primul rând de apariția unor noi structuri ale sistemelor de automatizare industrială, de apariția unor noi modele matematice, de optimizarea permanentă a algoritmilor și structurilor deja implementate.

Optimizarea rezultatului final al muncii inginerului de sistem este o necesitate, știut fiind că, pe lângă avantajele economice evidente pe care le aduce orice „inovație” inclusă în ansamblul automatizării, se disponibilizează în acest sens și noi resurse hardware, iar cele existente sunt exploatate la un nivel optimal. Mai mult chiar, prin clarificarea aspectelor legate de procesul controlat în sine și de metoda de control care va fi aplicată, se deschid noi posibilități de proiectare a automatizării respective sau a altora asemănătoare.

Conceperea unei scheme de automatizare performante nu poate fi făcută decât luând în calcul și modelul matematic al procesului controlat precum și cel al metodei de control. Formularea matematică a problemei de optimizare trebuie făcută astfel încât să permită conceperea soluției de optimizare, fără a genera un număr important de restricții.

Realizarea unui sistem modern de automatizare, măsurare și control a rețelelor de termoficare presupune implementarea unor sisteme informatice de proces performante, atât sub aspect hardware cât și software. Teoria sistemelor distribuite permite o foarte bună ierarhizare a elementelor constituente ale ansamblului.

O variantă performantă a acestui sistem presupune gruparea tuturor elementelor pe trei nivele ierarhice, anume **Nivelul „0”** sau nivelul de bază (format din totalitatea traductorilor și elementelor de execuție conectate la procesul urmărit și condus), **Nivelul „1”** (destinat cuplurului de proces) și **Nivelul „2”** care ca suport fizic echipamentele de calcul (calculator, imprimantă, MODEM, UPS) de la punctul dispecer și de la nivelul de conducere sau supervizare. Această soluție tehnică stă la baza tuturor aplicațiilor concepute de către autor, îndeplinind toate funcțiile și exigențele impuse unui dispecerat hidrotehnic. Configurarea hardware a rețelei trebuie să țină seama de realitatea aplicației, fiind de dorit utilizarea unor calculatoare de proces sau a unor controllere care au o fiabilitate și o robustețe sporită și mai puțin a calculatoarelor personale, care sunt chiar mai scumpe sau necesită numeroase alte elemente de interfațare.

Comunicația între elementele sistemului trebuie să fie fiabilă și rentabilă din punct de vedere economic, soluțiile cele mai utilizate fiind protocoalele RS232, RS422, RS485, M-BUS pentru comunicarea cu procesul și comunicarea prin fibră optică sau linie telefonică pentru comunicarea cu dispecerul central.

2.7. Contribuții personale

Capitolul 2 este rezultatul, în primul rând, al unei ample documentări din literatura de specialitate, dar și al unor contribuții esențiale în automatizarea sistemelor hidraulice.

Dintre principalele contribuții ale autorului amintim:

- Sinteza critică a unui număr mare de referințe bibliografice din domeniu;
- Descrierea celor mai noi metode, tendințe și principii de automatizare, utilizate în domeniul sistemelor hidrotehnice;
- Formularea și studiul unor modele matematice capabile să contribuie la definitivarea soluției de automatizare aleasă;
- Evaluarea soluțiilor de automatizare aplicabile în cazul sistemelor hidrotehnice.

3. SISTEME DE AUTOMATIZARE PENTRU HIDROTEHNICĂ

3.1. Interfațarea Om – Sistem de automatizare. Analiza și conceperea interfețelor

3.1.1 Analiza sistemului om-mașină

Toate sistemele informatice moderne, fac apel la un dialog Om - Mașină, care în mare măsură, reprezintă tocmai valoarea și importanța aplicației respective.

Dincolo de considerentele de ordin filozofic, legate de modul de conlucrare între om și rezultatul artefactului uman, relația om - mașină trebuie întotdeauna să conducă la efectul scontat, fără a fi deficitară, incompletă, ineficientă, inutilă sau chiar periculoasă.

Analiza comportamentului uman în fața unor echipamente sofisticate se face prin două metode:

- **AMDE** (Analiza Modalităților de Greșeală și Efectele lor);
- **AD** (Analiza prin Arborele de Greșeli).

În fig. 3.1 este prezentată structura unui sistem de supervizare.

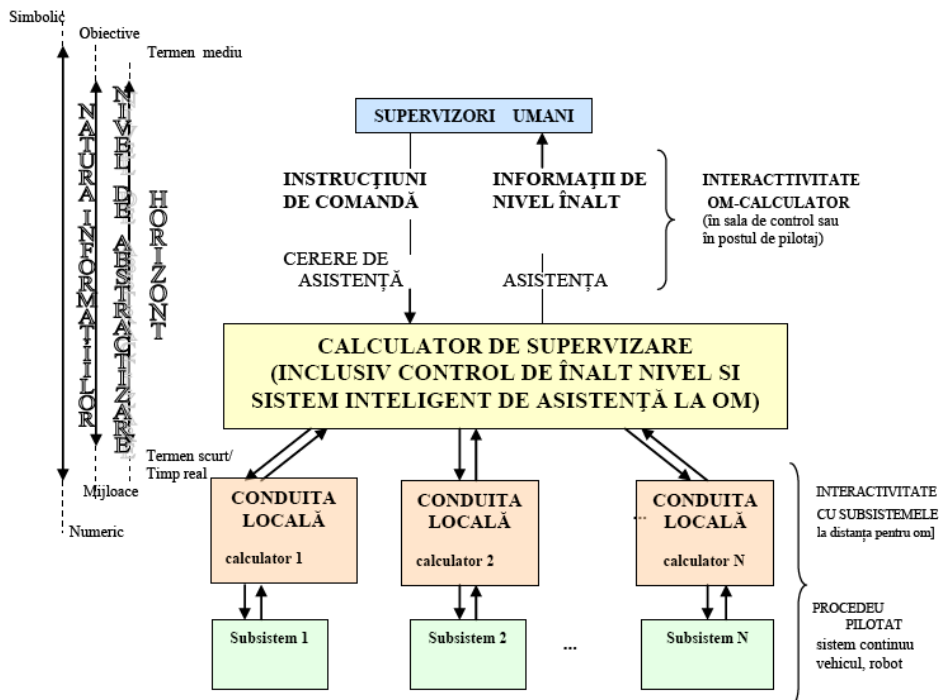


Fig. 3.1. Sistem de supervizare.

Prima metodă (AMDE) constă în abordarea ascendentă care, plecând de la cauzele greșelii, caută să cunoască efectele. Operatorul uman este luat în considerare atât ca un mod de compensare, cât și ca o cauză a greșelii, fără a discrimina totuși, diferitele moduri de greșeală.

A doua metodă (AD) este descendentă și pleacă de la fiecare efect, adică de la fiecare element de nedorit pentru a deduce cauzele, adică evenimentele de bază și combinația lor. Metoda tratează relațiile cauzale prezentate sub formă de arbori, fiecare vârf reprezentând fiecare eveniment nedorit, în timp ce baza lor, reprezintă evenimentele ce au cauzat greșelile.

Aceste două metode, AMDE și AD, sunt deci utile pentru a identifica metodele de funcționare anormale pentru a le compensa, aceste posibilități putând fi încredințate operatorului uman, ceea ce duce la definirea sarcinilor umane în sistemul Om - Mașină.

Adecvarea capacităților umane la constrângerile născute din îndeplinirea acestor sarcini, obligația este deci verificată plecând de la modelul de operator, cu scopul de a defini interferențele Om - Sistem.

3.1.1.1. Modalități de concepere a acestor interferențe

Modelul de operator (sau de echipă de operatori) definește resursele umane disponibile, și de asemenea limitele intrinseci ale operatorilor pentru achiziționarea și tratarea informațiilor pentru executarea acțiunilor rezultante. Conceptele realizate și formalismele utilizate vor depinde de modul de concepere adoptat, știindu-se că există deja două școli în zilele noastre.

Școala „fiabilă”, intervenind în domenii de mare risc, cum ar fi aeronautica sau domeniul nuclear. Posibilitățile de eroare sunt evaluate pornind de la analizele de teren sau de la experiențele în laborator și sunt regrupate într-o bancă de date ce joacă rolul unui model de limite umane.

A doua școală, cea „cognitivă”, s-a dezvoltat în jurul, RASMUSEN [86] și vizează nu atât să contabilizeze erorile umane, ci, mai mult de atât, să înțeleagă geneza plecând de la conceptele de psihologie cognitivă.

După abordarea fiabilă obligațiile umane cuantificate în raport cu posibilitatea lor de eșec sunt înapoiate în ansamblul sistemului pentru a estima ponderea lor în slăbiciunea / punctul slab al sistemului. Obligațiile care se dezvoltă conduc la un studiu ergonomic al interfețelor corespondente și, la nevoie, la o ameliorare, fie ergonomică, fie fiabilă, prin introducerea redundanței.

În calitate de exemplu metoda SHERPA/EMBREY [86] prezintă avantajul de a alătura la evaluarea (q) cantitativă a erorilor umane și o evaluare calitativă, apropiindu-se astfel de școala cognitivă. Abordarea „cognitivă” ne apare mai bogată, cel puțin în obiectivele sale, căci, sprijinindu-se pe o descriere a mecanismelor de decizie umane puse în joc pentru a îndeplini obligațiile, ea, abordarea, permite o specificare mai fină a interfețelor ce se realizează.

Ea se sprijină pe o „ analiză cognitivă a obligațiilor” (HOLLNAGEL, [89]) de realizat pentru a deduce „nevoile informaționale” și, în același timp „nevoile de ajutor în a decide”. Analiza nevoilor informaționale ne permite să definim care sunt informațiile ce trebuie afișate pe ecranele de supervizare și să structurăm imaginile corespondente după conținuturile operaționale ale procedurii.

În completare, analiza cognitivă a obligațiilor se sprijină pe cunoștințele ergonomice relative la limite atât în senzorio-motorii cât și cognitive, pentru a defini nevoile de ajutor (la decizie și / sau la acțiune) al clasei de operatori modelați mai înainte. Aceasta constituie într-o oarecare măsură caietul de sarcini pentru sisteme IA. De ajutor în decizia de integrare al sistemului de supervizare al instalației. Bineînțeles, acest caiet de sarcini va fi onorat după posibilitățile tehnice ale Instrumentelor de Inteligență Artificială existente.

3.1.1.2. Evaluarea sistemului realizat

Această etapă este indispensabilă pentru evaluarea sistemului conceput și realizat în etapa precedentă. Criteriile de evaluare țin cont de performanțele sistemului Om - Mașină, global exprimate, de exemplu, în relație cu despărțirea dintre producția reală și obiective în același timp în raport cu criteriile ergonomice ce permit evaluarea dificultăților întâlnite de operatori în momentul realizării obligațiilor / sarcinilor.

Non-observabilitatea directă a parametrilor semnificativi de dificultăți umane complică considerabil această evaluare economică și impune faptul de a construi observatori, plecând de la parametrii direct observabili, cum ar fi secvențele de acțiuni și mișcările efectuate de operatori, sau de la parametrii estimați de metodele subiective cum ar fi sarcina de lucru, oboseală sau, în sfârșit, parametrii estimați plecând de la un model cum ar fi anumite activități cognitive.

În realitate multiplicitatea parametrilor necesar să fie observați sau estimați este legată, pe de o parte, de complexitatea comportamentului uman, și pe de altă parte, de faptul că fiecare din acești parametri nu este destul de reprezentativ, în mod individual și nu este destul de rafinat pentru a caracteriza, el singur, statutul de sistem „operator uman executând o sarcină”. Acesta este de altfel un argument pe care îl putem pune în opoziție cu metodele pur viabile evocate mai înainte.

Intrările și acțiunile operatorilor, ca și comportamentul ca gest și vizual sunt analizate și la nevoie „rejucate” pentru a obține din partea lor explicații complementare asupra comportamentului lor și astfel de a deduce raționamentul care l-a condus la executarea acțiunilor ca răspuns la informațiile care le-au fost prezentat. Bineînțeles, punerea în evidență a acestui raționament uman trebuie să fie întărită prin cunoașterea resurselor cognitive particulare puse în joc de operatorul testat. Acesta trebuie să permită afișarea sau descrierea modelului decizional al operatorului propus chiar la începutul etapei descendente și în același timp modelului de sarcină ce decurge din aceasta. În acest sens, analiza activității reprezintă pentru munca reală ceea ce analiza sarcinii reprezintă pentru munca prescrisă.

Abordarea trebuie să poată să se sprijine pe metodele de extragere a cunoștințelor dezvoltate de științele cognitive și utilizate pentru a umple băncile de date ale sistemelor de Inteligență Artificială (HOURIEZ [94]).

Un al doilea obiectiv al analizei de activitate îl reprezintă adecvarea informațiilor prezentate de interfața realizată cu informațiile așteptate de operatori și necesare pentru efectuarea raționamentului efectuat mai sus.

Acesta conduce la anularea interfețelor, dacă este necesar, de exemplu dacă anumite informații lipsesc sau sunt prost prezentate.

3.1.1.3. Modele simplificate utilizate pentru conceperea de interfețe

YOHANNSEN, BORYS și SUNSTROM propun în proiectul ESPRIT GRANDIENT al cărui obiect este conceperea interferențelor de supervizare, un modul de sistem Om - Mașină incluzând un model de sistem și un model de utilizator. Acesta cuprinde o bază de cunoștințe de suprafață a utilizatorului care simulează resursele cognitive ale operatorului.

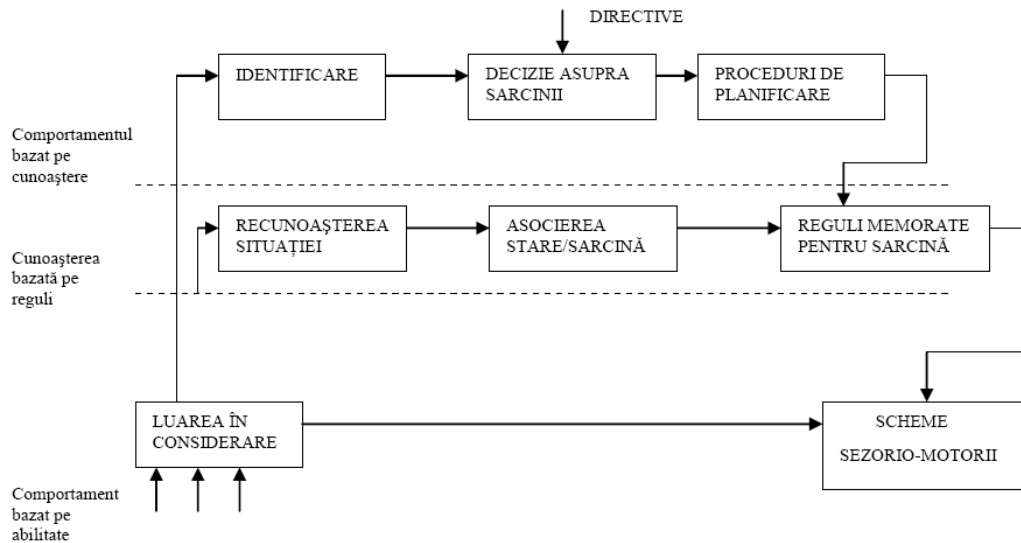


Fig. 3.2. Tipuri de comportament ale operatorului uman.

Cele două modele sunt folosite pentru a deduce în diferite contexte de funcționare a instalației de supervizare, sarcinile pe care „operatorul” simulat este considerat că le execută, după modalitatea conceptuală descendentă prezentată în paragraful de mai sus. În acest exemplu, modelul cognitiv al utilizatorului este în mod hotărât baza de cunoștințe ale viitorilor utilizatori. Problematika modelării cunoștințelor operatorului confirmă problemele de extragere de cunoștințe întâlnite în punerea în evidență a sistemelor experte.

Proiectul SINOP este complementar precedentului deoarece el modelează limitele și resursele perceptuale ale unui operator de supervizare sub forma unor cunoștințe ergonomice relative și cu imagini sinoptice afișate pe un ecran grafic în culori. SINOP este un sistem expert de reguli de producție (q) care este pus în interfața cu un generator grafic de imagini.

El interpretează imaginea grafică realizată sub formă de rețea semantică a cărei calitate ergonomică o evaluează apoi plecând de la bazele sale de cunoștințe. El modifică, dacă este nevoie, imaginea și o reconstituie pe ecran KOLSKI [89]. Acest proiect a fost validat și folosit în mai multe proiecte industriale. Muncile referitoare la software ar trebui să se concentreze asupra instrumentelor generatorului grafic ce facilitează conceperea imaginilor și ar putea să fie folosite în modalitatea de concepere ergonomică a sistemelor om-mașină.

Se pare că sunt două condiții necesare pentru a duce la bun sfârșit un astfel de proiect de modelare a comportamentului decizional uman și acesta, bineînțeles în limitele realiste stabilite mai sus, sunt pe de o parte definirea unei norme ce permite descrierea comportamentului decizional uman, iar pe de altă parte emitența instrumentelor tehnice ce permit modelarea și simularea mașinii în scopul verificării. Definirea „normei” revine în mod particular cercetătorilor din sfera științelor umane, cum ar fi psihologii de muncă, în timp ce modelarea și punerea în evidență relevă în principal competența informaticienilor și automaticienilor umani.

3.1.2. Cooperarea om-mașină. Principii

Conceptul de cooperare Om - Mașină s-a născut ca urmare a apariției instrumentelor Inteligenței Artificiale, considerate a fi asistenții persoanei abilitate să ia

măsurii, dar și din necesitatea de a prevedea și mai ales de a preveni conflictele susceptibile ce apar între aceste două tipuri de responsabili cu decizia: umani și artificiali.

Cercetarea s-a concentrat într-o primă etapă asupra sistemelor non-operatorii pentru a defini două moduri de cooperare numite respectiv vertical și orizontal.

1. În cooperarea verticală, operatorul este responsabil de toate variabilele procedurii și el poate face apel, dacă este necesar, la instrumentul de ajutorare în decizie care îi va furniza sfaturi, (așa cum reiese din fig. 3.3). În acest cadru, se pot distinge două principii:

- un principiu vizează ghidare operatorului în modalitatea de rezolvare a problemei pentru a-l conduce la descoperirea unei soluții, el însuși făcând acest lucru. Acest principiu este foarte atractiv căci îi permite operatorului de a-și păstra și îmbogăți cunoștințele operative ale procedurii. Totuși acest principiu are nevoie de o mare adaptivitate din partea sistemului de inteligență artificială pentru a asigura coerența între raționamentul său / și cel al operatorului, ținând cont de nivelul cognitiv al acestuia. Timpul de decizie poate fi destul de lung, ceea ce exclude situațiile urgente;
- Al doilea principiu vizează situațiile urgente în care operatorul poate „pierde” capacitățile de raționament obiectiv ca urmare a stress-ului. Calculatorul îi propune soluții, căci presiunea temporală nu-l poate permite să descopere el însuși. În acest caz, pot apărea conflicte când operatorul refuză soluțiile propuse de instrumentul de ajutorare și persoana însărcinată cu concepția trebuie deci să fie prevăzută posibilitatea dialogului și justificarea raționamentului ce permite rezolvarea rapidă a acestor conflicte.

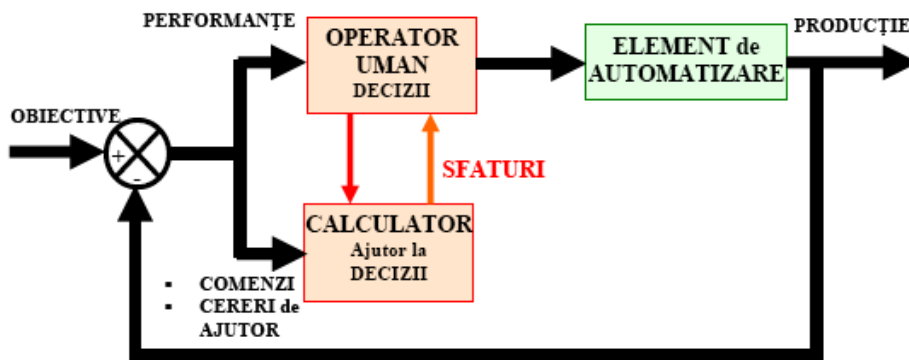


Fig. 3.3. Principii de cooperare om - mașină pe verticală.

2. În cooperarea orizontală, ieșirile instrumentului IA sunt conectate la dispozitivele sistemului de comandă al procedurii, ceea ce presupune că acest instrument IA posedă capacitățile de raționament în timp real. Astfel de instrument se va numi *agent*. Cei doi care decid, operatorul și agentul sunt deci la același nivel ierarhic și sarcinile de supervizare ca și acțiunile care rezultă din ele pot fi repartizate dinamic între ele, într-un mod în care să poată să-l ajute pe operator în situațiile sarcinilor prea multe de lucru, (așa cum reiese din fig. 3.3). Această cooperare poate fi pusă în evidență după două principii:

- primul principiu este repartiția dinamică „explicită” comandată de operator cu ajutorul unei interfețe de dialog. El evaluează el însuși sarcina de lucru și performanțele procedurii și alocă anumite sarcini agentului când este supraîncărcat. Punerea în evidență este relativ ușoară dar inconvenientul major constă

în sarcina de lucru umană complementară indusă / rezultată din gestiunea repartii sarcinilor și din comenzile trimise agentului.

- GREENSTEIN și LAM [85] propun optimizarea calității ergonomice a interfeței de dialog pentru a minimiza această sarcină de lucru complementară și într-un studiu experimental asupra simulatorului de control aerian și definesc și evaluează criteriile ce trebuie luate în considerare pentru a caracteriza această interfață;
- al doilea principiu este o repartii „implicită” girată / administrată de calculator. Dificultatea esențială constă deci în necesitatea de a defini un criteriu de repartii și mai constă în punerea în evidență automatic.
- GREENSTEIN și REVESMAN [86] propun să se integreze în sistemul de comandă al repartii un model de acțiuni umane și să se aloce agentului sarcinile pe care operatorul nu le-a putut executa, după predicțiile acestui model.

Cu toate că ideea pare atractivă, ea este dificil de realizat astăzi, ținându-se cont de dificultățile întâlnite în încercările de realizare a dezvoltărilor decizionale umane amintite mai înainte. O altă metodă în căutarea unei repartizări optime a sarcinilor între operator și agent după o modalitate iterativă.

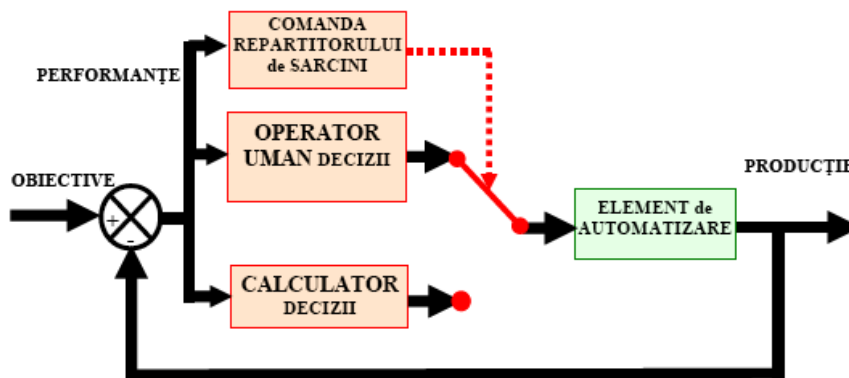


Fig. 3.4. Principiul de cooperare om - mașină pe verticală.

3.1.2.1. Cooperarea pe verticală

Studiul descris mai sus se plasează în contextul unei situații urgente în care sistemul de ajutorare în decizie propune sfaturi de acțiune sau de prevenire a operatorului.

Studiul a fost condus în cadrul proiectului ALLIANCE, al cărui scop este punerea în evidență a unui sistem expert în timp real de filtrare, de alarmă, de diagnostic și de ajutor în decizie destinat operatorilor de supervizare a unui procedeu industrial. Procedeul suport este o stație de încercare de generatori de abur.

Structura sa este apropiată de cea a unei centrale nucleare: un circuit primar produce și transmite energie generatorului, care o transformă în abur. Producerea energiei primare este asigurată de rezistențe electrice. Stația de încercare - analiză este pilotată de un calculator și supervizată de experți operatori, aflați într-o sală de control.

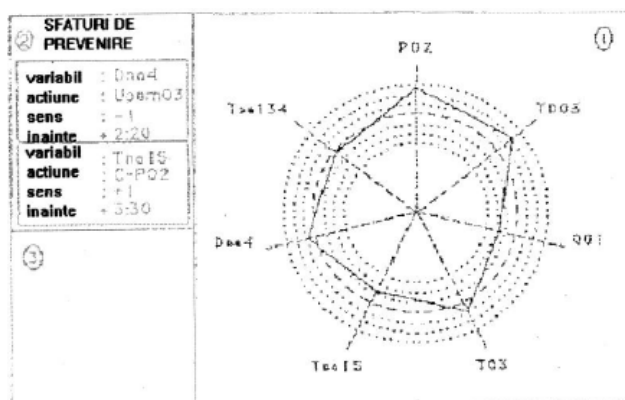


Fig. 3.5. Interfața dintre om și sistemul expert.

Sistemul expert realizat prezice continuu evoluția viitoare a procedului după un orizont de predicție de mai multe minute (până la 15 minute), cu o perioadă de predicție de 1 minut. El compară evoluția prezisă cu pragurile de alarmă și, dacă este necesar, propune operatorului sfaturi de prevenire.

În continuare vom detalia modalitățile de concepere a interfeței dintre operator și sistemul expert, pe de-o parte și de concepere a integrării pe de altă parte.

Interfața a fost concepută pentru a răspunde nevoilor informaționale ale operatorului în diferite moduri de funcționare a procedului:

- supravegherea funcționării normale și detectarea greșelilor,
- rezolvarea problemei și elaborarea unei acțiuni compensatorii sau preventive în timpul funcționării anormale detectate mai înainte.

3.1.2.1.1. Supravegherea funcționării normale și detectarea greșelilor

În acest tip de procedee interconectate, operatorul supraveghează numai un număr redus de variabile semnificative în timpul funcționării normale. În acest caz, aceste variabile au fost regrupate într-o „vezi pe stea” după o idee de COEKIN [68] reluată de WOODS [69]. „Vederea în formă de stea” cuprinde spate raze ale aceluiași cerc, așa cum reiese din fig. 5.16.

Caracteristicile fiecărei variabile sunt afișate pe fiecare rază. Sunt stabilite praguri pentru fiecare variabile: prag înalt, foarte înalt și prag de alarmă gravă, prag jos, prag foarte jos și alarmă redusă). Aceste caracteristici sunt normate în așa manieră încât ele pot fi reglate de cercuri concentrice. Amplitudinea fiecărei variabile este reprezentată de un punct de pe raza sa. Cele șapte puncte sunt legate între ele de segmente de dreaptă în așa fel încât să formeze un poligon regulat atunci când procedeul funcționează normal.

Când apare o greșeală, poligonul se deformează, ceea ce îl pune în alertă pe operator și evoluțiile viitoare ale stării procedului sunt succesiv afișate după prezicerile sistemului expert.

Interfața este un ecran grafic în culori cu trei ferestre. Vederea de stea este afișată în fereastră.

1. Sfaturile de corectare și / sau de prevenire furnizate de sistemul expert sunt afișate în fereastră.
2. Procedeul este deci în funcționare anormală.

3.1.2.1.2. Modul de funcționare anormală a procedurii

Sfaturile de prevenire (sau de corectare) regroupează în fereastra 2 etichetele variabilei în greșeli și în dispozitivele pentru corectare, amplitudinea acțiunii și întârzierea maximă a intervenției.

Când există mai multe sfaturi ele sunt ierarhizate după întârzierile lor de intervenție. În acest context operatorul uman poate avea trei tipuri de reacții diferite față de sfaturile sistemului expert:

- nu este nici o soluție pentru corectarea greșelii și de aceea el este tentat să aplice soluția propusă de sistemul expert;
- propria sa soluție coincide cu cea a sistemului expert și în acest caz el rămâne la propria sa soluție;
- propria sa soluție diferă de cea a sistemului expert și în acest caz apare un conflict între cei doi care decid.

Operatorul trebuie să verifice cele două modalități decizionale – a sa și cea a sistemului expert – și să ajungă la soluțiile care asigură un consens, adică să determine cine are dreptate și de ce. Această etapă de verificare poate fi lungă și complexă și sistemul expert trebuie să-l ajute pe operator explicându-i modalitatea pe care a urmat-o.

Abordarea din punctul nostru de vedere a concepției de interfață a constat în a căuta care sunt punctele de consens posibile între cei doi care decid. Această metodă conceptuală a permis definirea informațiilor pe care sistemul expert trebuie să le furnizeze operatorului. Prezentăm în continuare imaginile realizate pentru explicarea raționamentului sistemului expert și modalitatea conceptuală care a permis generarea acestor informații.

3.1.2.1.3. Justificarea raționamentului sistemului expert

Cele două nivele de justificare posibile ale sistemului expert sunt, pe de o parte diagnosticul care servește drept bază pentru elaborarea sfaturilor de acțiune și, pe de altă parte variabilele luate în considerare pentru a ajunge la diagnostic.

Diagnosticul este prezentat sub forma unei perspective numită „propagare” pe care operatorul o poate numi, în calitate de claviatură specializată și care se afișează în fereastra 1. Variabilele ce privesc greșeala sunt plasate într-un sinoptic simplificat al instalației și sunt legate între ele de arcuri orientate ce simbolizează semnalul propagării greșelii de la origine și până la variabilele ce privesc acțiunea (așa cum reiese din fig. 5.16).

Dacă operatorul dorește explicații suplimentare el poate apela la al doilea nivel de justificare sub forma de „curbe” care se afișează în fereastra 1. Această perspectivă prezintă mai multe curbe de variabile, cuprinzând fiecare (în legătură cu fig. 3.5):

- istoricul variației;
- momentul ultimei estimări;
- curba evoluției viitoare a variabilei prezisă de sistemul expert;
- și când perioada de estimare este mai lungă decât perioada de eșantionaj a variabilei, o a treia curbă afișează evoluția reală a acesteia de la ultimul moment de precizie: aceasta permite operatorului să compare evoluțiile reale prezise. Aceste curbe sunt repuse în joc la fiecare moment de precizie.

Fereastra 3 este folosită pentru a afișa un ajutor în utilizarea interfeței pe care operatorul o poate chema cu ajutorul comenzii „ Help” (ajutor).

3.1.2.1.4. Rezolvarea conflictelor

Când apare un conflict, operatorul trebuie să găsească cel mai rapid posibil consens. S-a analizat în paralel fiecare din modalitățile de rezolvare a problemei dintre cei doi care decid astfel încât să se determine punctele de consens posibile. Metoda constă în punerea în paralel a modalităților decizionale ale fiecăruia dintre cei care decid. Aceea a operatorului este realizată de modelul RASMUSSEN [67] cu patru etaje: detectarea evenimentului, evaluarea situației, luarea deciziei, acțiunea.

Raționamentul sistemului expert este presupus a fi compus din 3 stadii:

- un etaj de supervizare ce efectuează precizarea evoluției variabilelor semnificative;
- un etaj de detectare a greșelilor pentru a identifica evoluțiile anormale prezise;
- un etaj de corectare care trage concluzia asupra unui ansamblu de acțiuni preventive.

Procedurile de acțiuni diferite pot proveni fie din diagnostice diferite, fie dintr-un diagnostic ce conduce la mai multe soluții posibile. Diagnosticul este deci un punct de consens și propagarea trebuie să-l ajute pe operator să-l găsească. Mai mult, diagnostice diferite pot proveni din omiterea anumitor variabile în timpul evaluării situației. Ansamblul de variabile luate în considerare reprezintă al doilea punct de consens pe care curba ar trebui să permită să-l atingă.

Fig. 3.6 rezumă diferitele imagini ce constituie interfața, în paralel cu modalitatea de rezolvare a problemei de către operator realizată sub formă „lineară” a modelului RASNUSSENN. Această interfață a fost implantată pe stația de încercări la Cadarache dar, din nefericire, nu a constituit obiectul unui studiu de validare detaliată după modalitatea descendentă amintită mai înainte.

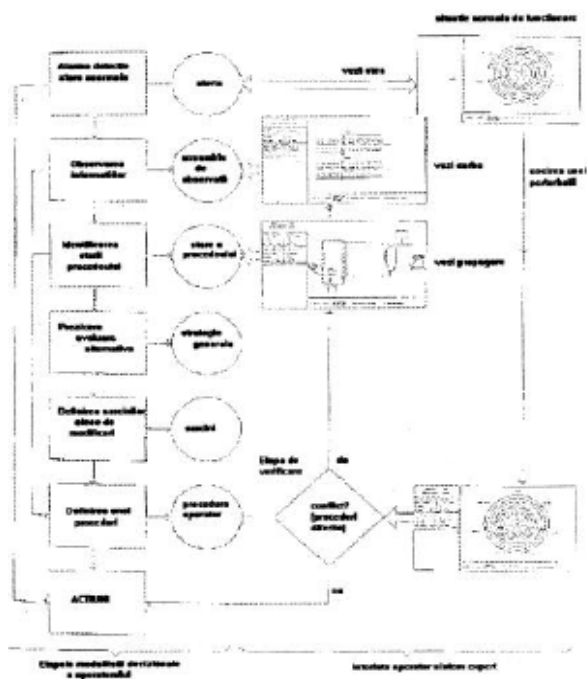


Fig. 3.6. Sinteza de diferite imagini constituind interfața dintre operator și sistemul expert.

3.1.2.1.5. Evoluția către cooperarea verticală interactivă

Pe plan mondial, cercetarea se orientează anual către o cooperare verticală interactivă în care schimburile de informații (rezultate parțiale sau cerere de ajutor) se pot efectua în diferite etape ale modalității decizionale a unuia sau a celuilalt (dintre cei care decid) și în cele două sensuri, așa cum observa GANDIBLEUX [95].

Miza ergonomică este tentantă, în sensul că flexibilitatea tehnică poate contribui la acceptarea instrumentului de către operator. Miza tehnică nu este mai prejosă căci permite operatorului de a ghida instrumentul în cercetarea soluțiilor, poate contribui la ameliorarea eficacității sale și la simplificarea structurii. Dar, pentru a ajunge la aceste mize, rămâne de făcut un anumit număr de pași; se pare că problematica explicării reprezintă un punct bun de pornire.

Sistemul trebuie să fie capabil să înțeleagă tipul de cerere de explicație emisă operatorului. Aceasta aduce cercetătorilor în psihologie cognitivă mai ales, posibilitatea găsirii modalității de rezolvare a problemelor și de planificare a operatorilor (mecanisme cognitive, tipurile de comportament cognitiv, recunoașterea scopurilor, intențiile) dar în același timp și posibilitatea identificării mecanismelor de erori umane: cererea de explicație se înscrie în raționamentul corect sau rezultă dintr-o eroare a operatorului.

3.1.2.2. Cooperarea orizontală: repartizarea dinamică a sarcinilor

3.1.2.2.1. Principii și condiții de punere în funcțiune

Structura de comandă și de supervizare generală a unui procedeu poate fi descompusă în trei nivele:

- procedeu;
- nivelul de comandă și de reglare automată;
- nivelul de supervizare ce revine în mod obișnuit operatorului și care cuprinde sarcini de decizie cum ar fi poziționarea consemnării, gestiunea greșelilor și sarcinile de planificare și de organizare generală, așa cum rezultă din fig. 3.7. „Agenții” vor fi integrați în nivelul 3. Este ușor de definit, o organizare generală a acestui nivel de supervizare ce permite repartizarea dinamică a responsabilităților între operator și agent. Principiul constă în includerea în nivelul 3 a unui repartizator de sarcini care distribuie acestui procedeu între cei doi care decid.

Așa cum am arătat-o mai înainte, comanda celui care repartizează sarcini poate fi asigurată fie de calculator (repartiție implicită) fie de operator (repartiție explicită). Obiectivul este deci căutarea performanței optime a procedurii modificând repartizarea sarcinilor după performanțele respective ale celor doi care decid și ținându-se cont de limitele fiecăruia dintre ei: algoritmul de calcul prea lent pentru calculator sau stare de supraîncălzire a operatorului.

Punerea în funcțiune a unui astfel de principiu este deci subordonată existenței unei repartii optime a sarcinilor între cei care decid și care optimizează performanțele procedurii pe care-l pilotează. Un studiu experimental preliminar a permis să se poată asigura acest lucru. MILLOT, TABORIN, [89]. Acest studiu a fost realizat sub forma unei platforme experimentale cuprinzând o stație de supraveghere a procedurii simulat pe un calculator de timp real.

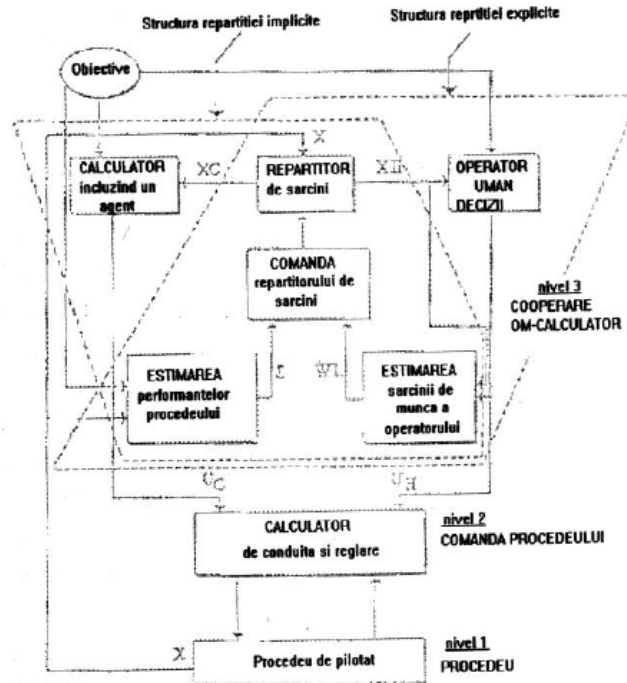


Fig. 3.7. Principii de repartizare dinamică a sarcinilor de supervizare.

Procedeu simulat cuprinde 96 variabile independente având timp de a răspunde diferiți (între 6 mm și 16 mm) și putând fi supuse perturbațiilor adiționale aleatorii. Operatorul dispune de un post de supraveghere ce cuprinde două ecrane grafice colorate și claviatură funcțională.

Primul ecran afișează o vedere globală a celor 96 variabile sub formă de grafice – bară și le repartizează în 12 grupe de câte f variabile. Sub fiecare grafic-bară sunt suprapuse 4 „linii” ce simbolizează respectiv: referința, pragul de alarmă înaltă și un prag de alarmă slabă și o valoare de comandă pentru a compensa perturbarea.

Cel de-al doilea ecran afișează vederea de conduită a unei singure variabile regrupând o grupă istorică a variabilei, ca și valorile numerice ale consemnării, praguri de alarmă, comanda și mărimea variabilei. O claviatură funcțională permite operatorului să selecționeze variabila ce trebuie afișată la vederea de conduită și să modifice comanda compensatoare.

Rolul operatorului sau a agentului este de a detecta variabilele perturbate care îi sunt alocate și de a compensa fiecare perturbare. Agentul este integrat sistemului de supraveghere și este simulat prin regulatori PID. Performanța E a procedurii este calculată ca o sumă de valori absolute de devieri între măsura și consemnarea fiecăreia dintre variabile.

În contextul experimental studiat, 90 de variabile erau perturbatoare simultan. Experiențele realizate vizau evaluarea performanțelor E ale procedurii simulat, după repartitia variabilelor între operator și agent, adică, după numărul de variabile revenite fiecăruia dintre ei.

Zece serii de experiențe au fost realizate, în prima serie agentul superviza ansamblul de 90 de variabile, în a doua, operatorul superviza 10, și agentul trata cele 80 de variabile rămase, în cea de a treia operatorul ...20 și agentul 70 și tot așa până la al zecelea experiment unde operatorul superviza ansamblul de 90 de variabile. Aceste experimente au fost efectuate cu zece operatori diferiți.

Compararea performanțelor Σ ale procedurii, obținute în diferitele experimente a arătat că, atunci când performanțele intrinseci ale celor doi care decid sunt apropiate, există o repartitie a variabilelor între ei ce conduce la o performanță optimă a procedurii globale.

Acest experimente au fost apoi repetate modificându-se performanțele intrinseci ale agentului (modificându-se parametrii regulatorilor PID). Rezultatele obținute arată că, atunci când performanțele intrinseci ale operatorului sunt mai bune decât cele ale agentului, performanța optimă a procedurii este obținută ținând cont de toate variabilele operatorului. Acest lucru poate conduce la o supraîncărcare în munca de operator și pentru a evita acest lucru trebuie introdus un prag de sarcină de lucru maxim admisă pentru operator, în comanda repartitorului de variabile.

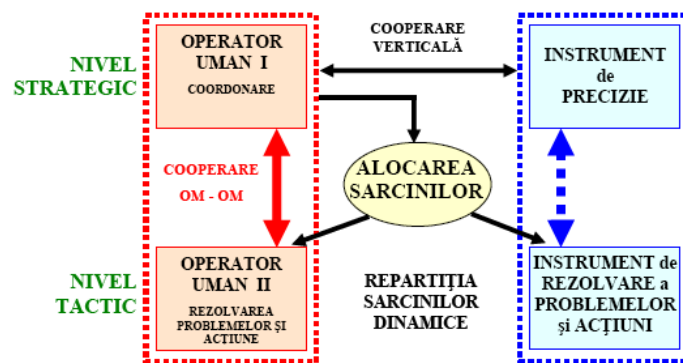


Fig. 3.8. Cooperarea Om - Mașină.

Invers, când agentul posedă performanțe intrinseci superioare operatorului, alocarea optimă vine să încredințeze toate variabilele primului, în detrimentul unei supraîncărcări a muncii operatorului. Pentru a evita acest lucru, trebuie introdus un prag minim admis operatorului, în comanda repertoriului de variabile.

Problemele puse cuprind pe de o parte cooperarea Om - Mașină la fiecare nivel și cooperarea OM - OM, naturală între cele două nivele și cooperările Om - Mașină dintre nivele. NANDERHAEGEN [94]; întâlnim și structura pe mai multe nivele (GRAVEZSP [95]).

A doua platformă experimentală este la capătul realizării pentru a studia acest sistem cooperativ pe mai multe nivele într-un control aerian în curs. Protocoalele experimentale au fost definite pentru a răspunde întrebărilor amintite mai sus și bineînțeles pentru a testa noua organizare pe mai multe nivele propusă. Acest studiu a fost făcut în colaborare strânsă cu o echipă de psihologie cognitivă ergonomică.

În modalitatea concepere / evaluare a sistemelor Om - Mașină se pot observa două probleme fundamentale cu definirea unei structuri de cooperare Om - Mașină:

Prima problemă privește împărțirea sarcinilor și necesitatea de a defini aceste metode, astfel încât să apară trei clase, după criterii atât tehnice cât și ergonomice:

- o clasă de sarcini care relevă capacități exclusive ale automatismelor (pe care operatorul nu poate sau nu trebuie să le facă);
- o clasă de sarcini care arată competențele exclusive ale operatorilor umani (acelea pe care tehnica nu știe să le rezolve);
- și, ceea ce este nou la automatizări, o a treia clasă care regroupează sarcinile pe care automatismul le poate executa, preferă să le încredințeze operatorilor.

NU există, după câte știm, o metodologie dovedită, riguroasă ce duce la această clasificare. Acest lucru este și mai adevărat când sarcinile sunt deja repartizate în mod natural între mai mulți operatori. De notat că, acest tip de dificultate metodologică pentru a descompune o problemă pentru un sistem multiagent în vederea rezolvării de probleme este de asemenea flagrantă în IA și nu are, după cum știm, un răspuns

satisfăcător. Răspunsurile la această problemă au interese ergonomice evidente dar ele condiționează punerea în funcțiune a unui sistem de sarcini Om - Mașină. A treia clasă indicată mai sus constituie ansamblul de sarcini „partajabile” între operator și sistemul de ajutor, acest partaj putând fi dinamic.

A doua problemă este legată de precedenta și privește gradul de automatizare pe care automaticianul trebuie să-l aleagă. Acest punct este mai larg decât precedentul în sensul că nu implică în mod necesar un partaj dinamic ale sarcinilor între Oameni și Automatisme dar poate fi un partaj static al sarcinilor între ei.

3.1.3. Sisteme expert. Definiere, modalități de construire

Utilizarea calculatoarelor pentru rezolvarea problemelor complexe de analiză a stării unui sistem are o istorie îndelungată. Atunci când vorbim despre identificarea stării unui sistem complex cum este sistemul hidrotehnic și de navigație, punerea în ecuație, și, deci, rezolvarea prin metode algoritmice a problemei, începe să pună semne de întrebare, cu atât mai mult cu cât luăm în considerare mai multe aspecte, mai multe informații. Chiar la nivelul unei porțiuni restrânse a acestui sistem hidrotehnic (un dulap de automatizare a unei porți, de exemplu), este foarte dificil a găsi un algoritm unic, universal valabil, care să satisfacă un obiectiv precizat, spre exemplu alarmarea în cazurile de funcționare necorespunzătoare a instalațiilor din acest punct de lucru al instalației.

Astfel, apare necesitatea utilizării unor metode care să fie capabile să înglobeze experiența acumulată, referitoare la o gamă de situații și chiar să învețe din experiențe noi. Aceste tehnici sunt grupate generic sub denumirea de tehnici de inteligență artificială (IA). Cuvintele cheie ale acestui domeniu sunt: Sisteme Expert (SE), Rețele Neuronale Artificiale (RNA) și Logică Fuzzy.

3.1.3.1. Categoriile de aplicații

Controlul instalațiilor din domeniul producției, transportului și distribuției energiei electrice sau al sistemelor hidrotehnice presupune, pe lângă instrumente de analiză și identificare a stării procesului condus, și existența mecanismelor, a tehnicilor de acțiune pentru schimbarea stării sistemului. Operatorul însărcinat cu controlul acestui proces va trebui să aibă o viziune clară a stării sistemului, să-și construiască un plan de acțiune pentru situații particulare, cât și stările intermediare ale procesului condus. Trebuie să luăm în considerare și perturbațiile care pot apărea în acest timp, ale căror implicații trebuie să fie corect identificate, care vor impune ajustarea planului de acțiune. Toate cele enumerate mai înainte sunt factori de stres pentru operator, stres care poate duce la erori, unele cu consecințe extrem de grave.

O constatare importantă este aceea că în foarte multe cazuri (în special în situațiile care presupun corelări ale multiplelor informații despre procesul condus), experiența operatorului este hotărâtoare în ceea ce privește alegerea celui mai potrivit plan de acțiune. Tehnicile procedurale, bazate pe procesarea algoritmică a informațiilor nu sunt capabile să ia în considerare acest ultim fapt. Definiția concretă a unui sistem expert este, de regulă, nuanțată după domeniul de utilizare. O definiție generală a fost formulată de către Edward Feigenbau, de la Universitatea Standford S.U.A. [68]: „...un program inteligent care utilizează cunoștințe, fapte și tehnici de raționare pentru a rezolva probleme care în mod normal necesită experiența experților umani”.

SE reprezintă o ramură a inteligenței artificiale care folosește cunoștințe specializate pentru a rezolva o problemă la nivelul unui expert uman. Toate definițiile și referirile la SE scot în evidență această trăsătură definitorie, anume faptul că un SE utilizează informații extrase din experiența umană, putând deci furniza decizii la nivelul

de competență corespunzător informațiilor primite și a metodelor de raționare implementate.

Asemănător expertului uman, care bazându-se pe cunoștințele pe care le posedă, prin raționament propriu ajunge la anumite concluzii, un sistem expert se bazează pe cunoștințele înglobate și pe mecanismul de inferență, care are funcția de a lua decizii în urma unor procese de raționament logic.

Potrivit aceluiași studiu CIGRE [33], motivația de a implementa tehnici IA rezultă în principal pentru probleme caracterizate de următoarele proprietăți:

- Date inconsistente, incomplete și conflictuale: cazul aplicațiilor de procesare a alarmelor, unde rolul SE este de a extrage informația utilă dintr-o avalanșă de informații incomplete și adesea conflictuale;
- Complexitatea structurii sistemului condus: SE poate modela cunoștințele expertului uman pentru o rapidă identificare a regiunilor slabe dintr-un sistem energetic;
- Natura combinatorie a soluțiilor: un exemplu elocvent este problema restaurării stării de funcționare după o avarie, care necesită aplicarea unei strategii de căutare într-o problemă multidimensională.

Spre deosebire de aplicațiile tradiționale, care necesită cunoașterea dinainte a algoritmului aplicației, sistemele expert lasă la latitudinea utilizatorului introducerea modului de raționament, prin editarea bazei de reguli. În cadrul programelor convenționale, cunoștințele legate de o problemă și metodele de rezolvare sunt „amestecate”, ceea ce face orice modificare ulterioară a programului să fie dificilă. Datorită unei delimitări clare între componente (reguli, bază de date, motor de inferență), un SE poate fi modificat prin simpla adăugare sau extragere de reguli, fără a fi necesară rescrierea programului.

Pentru un SE sunt importante 4 caracteristici:

- Lucrează la un nivel expert de competență;
- Folosește un mecanism de inferență pentru a realiza deducția;
- Expertiza efectuată de un SE se bazează pe cunoștințele real dobândite;
- Programarea unui SE implică descrierea și prezentarea cunoștințelor unor experți în domeniu.

Sistemele expert nu sunt potrivite oricărei probleme de rezolvat. A existat o perioadă în care „moda” sistemelor expert a forțat nota, încercând să rezolve multe dintre aplicațiile care ar fi fost mai potrivit de abordat prin alte metode. Unele probleme pot fi descrise utilizând algoritmi existenți, ori metode statistice de evaluare. Altele însă, nu sunt la fel de bine definite, solicitând implicarea unui expert uman în rezolvare. Aceste aplicații par cele mai potrivite pentru abordarea cu SE. În special în cazul datelor incomplete și inconsistente, ele constituie alternativa preferată pentru soluționarea cu SE.

Există un test imaginar, „testul telefonului”, care poate fi aplicat pentru a testa rezolvabilitatea unei probleme utilizând tehnicile SE. Se consideră că dacă expertul uman poate rezolva problema comunicând telefonic cu utilizatorul, atunci un SE poate fi dezvoltat în acest scop. Pe de altă parte, dacă utilizatorul nu este capabil să descrie verbal datele problemei, sau expertul uman de la celălalt capăt al firului nu este capabil, bazându-se pe o chestionare telefonică, să rezolve satisfăcător problema, atunci abordarea SE este nesatisfăcătoare.

3.1.3.2. Structura constructivă a unui sistem expert

Structura unui SE este prezentată în fig. 3.9.

În conformitate cu această figură, un sistem expert este alcătuit din următoarele module:

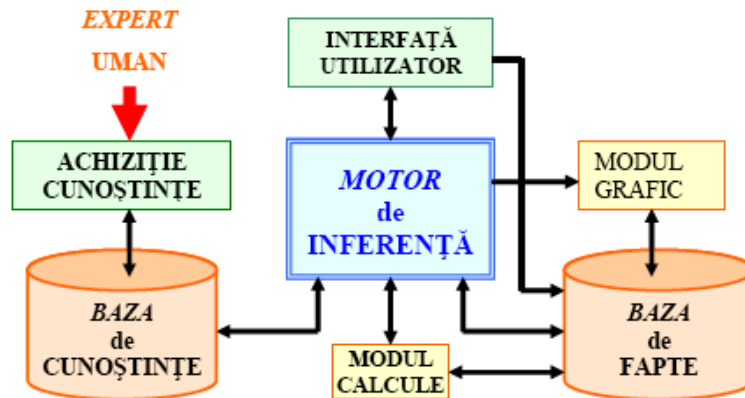


Fig. 3.9. Structura unui sistem expert.

- ❖ **Baza de cunoștințe.** Este o colecție de cunoștințe relevante despre un anumit domeniu. Baza de cunoștințe cuprinde *regulile*, care se referă la operațiile care pot fi efectuate asupra obiectelor conținute în baza de date. În esență, regulile constituie un ansamblu complet și necontradictoriu de cunoștințe, necesare rezolvării unei probleme.
- ❖ **Baza de fapte (date),** conține informațiile relative la domeniul de aplicație studiat. Faptele reprezintă partea dinamică a bazei de cunoștințe și au rolul de a reprezenta starea obiectelor la un moment dat.
- ❖ **Motorul de inferență** este un program general, care implementează mecanismul prin care se construiesc raționamentele. Motorul de inferență prelucrează cunoștințele și datele pe baza acestor raționamente.
- ❖ **Interfața utilizator** asigură dialogul între SE și utilizator, permițând acestuia din urmă să pună întrebări sistemului expert, să introducă noi informații sau să vizualizeze rezultatele deducțiilor. Această componentă are un impact esențial asupra percepției pe care o dă utilizatorul problemei rezolvate de către SE, fiind importantă corelarea modelului mental pe care utilizatorul îl folosește, cu reprezentarea pe care o oferă SE.
- ❖ **Modulul de achiziție cunoștințe** permite completarea și modificarea bazei de cunoștințe a SE. Acțiunea propriu – zisă de achiziționare a cunoștințelor se realizează prin chestionarea experților umani.
- ❖ **Modulul explicativ** are rolul de a furniza explicații operatorului în legătură cu raționamentul folosit pentru a ajunge la concluzia prezentată.

3.1.3.3. Etapele realizării unui sistem expert

Există mai multe abordări ale dezvoltării unui SE. Una dintre acestea a fost propusă de Waterman [165] și constă din următoarele etape:

- Identificarea;
- Concepția;
- Formalizarea;
- Implementarea;
- Testarea.

Etapele de mai sus sunt strâns legate și interdependente. Are loc un proces iterativ, cu frecvente reveniri la etape anterioare, până când programele ating un nivel acceptabil de performanță.

- **Identificarea** este pasul de analiză necesar în oricare dintre metodele de dezvoltare software tradițională. Implică o analiză formală a cerințelor externe ale aplicației, stabilește intrările și ieșirile, aria de aplicare a programului și stabilește utilizatorul. Tot în această etapă sunt identificați participanții, problemele, obiectivele, resursele și costurile.

Participanții sunt un grup care colaborează la efortul de dezvoltare a aplicației, care include în mod necesar și inginerul specializat în problematica cunoștințelor („knowledge engineer”) precum și expertul în domeniu. De subliniat că de expertul uman depinde, în ultimă instanță, calitatea informațiilor furnizate în final de SE. Sunt însă cazuri de aplicații SE, care pot avea ca sursă de expertiză nu un expert uman, ci un material de referință din literatura de specialitate, în special atunci când se cere orientarea rapidă a utilizatorului într-o bază de cunoștințe voluminoasă și mai puțin contradictorie. În mod uzual, aceste din urmă cazuri sunt prezentate ca sisteme bazate pe cunoștințe sau sisteme bazate pe reguli și constau mai degrabă în găsirea rapidă a informației relevante într-un material de referință vast.

- **Concepția** înseamnă proiectarea programelor astfel încât să asigure interacțiunile și relațiile în domeniul specific al aplicației. În această etapă sunt determinate conceptele de bază, relațiile între obiecte, procese și mecanismele de control.

Inginerul de cunoștințe poate utiliza întrebări, de genul celor de mai jos, pentru a înțelege exact ce are de făcut expertul:

- Care sunt deciziile pe care trebuie să le ia expertul?
- Cum se iau aceste decizii?
- Care dintre decizii cer o reflecție îndelungată, o explorare suplimentară sau interacțiuni?
- Ce resurse sunt necesare pentru a lua o decizie?
- Ce condiții sunt necesare pentru a lua o decizie particulară?
- Fiind dat un caz specific, predicțiile oferite de SE vor fi conforme cu recomandările expertului uman?

Sarcina inginerului de cunoștințe este de a identifica sursele de cunoștințe utilizate de expertul uman, atunci când acesta face o recomandare specifică, spre exemplu să determine materialul de referință care trebuie consultat, calculele care trebuie efectuate, regulile euristice pe care le aplică, informațiile care sunt cerute utilizatorului pentru aplicarea acestor reguli euristice. Structurile create de inginerul de cunoștințe rezultă de regulă în urma unor frecvente discuții cu expertul uman.

Unele unelte de dezvoltare ale SE au facilități de inducție, care permit crearea de reguli bazate pe exemple date de expert. Asemenea abordări sunt în general utilizate în cazul problemelor de clasificare.

- **Formalizarea** constă în organizarea conceptelor cheie, a subproblemelor, în reprezentarea formală a fluxului de informație. În fapt, în această etapă se proiectează logica programelor. Munca inginerului de cunoștințe constă în construirea unui set de structuri interdependente, de tip arbore, pentru reprezentarea bazei de cunoștințe. Reprezentarea grafică a acestor structuri este importantă pentru claritatea și credibilitatea acestora, în ultimă instanță pentru acceptarea lor de către utilizator.

Este dificil de separat etapa de formalizare de cea de concepție. În realitate, proiectarea bazei de cunoștințe se face aproape în paralel cu achiziția de cunoștințe. Două elemente sunt importante în etapa formalizării:

- rafinarea, organizarea cunoștințelor în structuri și relații;
- determinarea cât mai precisă a interacțiunii între utilizator și SE.

- **Implementarea** constă în transpunerea cunoștințelor formale în prototipurile de lucru ale aplicației. Conținutul structurilor de cunoștințe, regulilor de inferență și al strategiilor de control stabilite în etapele anterioare sunt organizate într-un format corespunzător și transpuse în cod.

Aici este punctul în care se stabilește modul în care aplicația SE va interacționa cu celelalte programe și cu bazele de date.

- **Testarea** implică mai mult decât simpla identificare și corectare a erorilor de sintaxă în programe. Se referă la verificarea relațiilor și interdependențelor individuale între componente, la validarea performanțelor programului și la evaluarea utilității pachetului software.

De validarea corespunzătoare a programului depinde esențial succesul și acceptarea aplicației de către utilizatori.

În procesul de testare - validare se au în vedere următoarele aspecte:

- Regulile trebuie să fie corecte, complete și consistente;
- Abilitatea strategiei de control de a lua în considerare informația corespunzătoare problemei de rezolvat;
- Justificarea informațiilor cerute de la utilizator pentru fiecare concluzie furnizată;
- Verificare de către expertul uman a răspunsurilor date de către SE.

3.2. SCADA, o soluție pentru monitorizarea sistemelor hidrotehnice

SCADA este un acronim pentru **Supervisory Control And Data Acquisition** (Control Supervizat și Achiziție de Date).

Este un sistem ce folosește calculatorul pentru a oferi facilități de operare centralizată a utilităților (de exemplu distribuția energiei electrice, a apei, a energiei termice) sau a capacităților de producție. Mai poate fi întâlnită și denumirea de Sistem de Telemărire (**Telemetry System**) sau Sistem de Automatizare și Control de la Distanță (**Automation and Remote Control**).

Un sistem SCADA oferă posibilitatea unui control eficient și sigur în cazul în care rețeaua deservită este una cu întindere geografică foarte mare și este structurată pe mai multe nivele și puncte de transformare. Sistemul oferă posibilitatea monitorizării și controlului centralizat al tuturor nodurilor și concentratoarelor de rețea.

Un sistem SCADA are în alcătuire un centru de control (numit dispecer sau master terminal unit - MTU); unul sau mai multe dispozitive pentru culegerea și emiterea comenzilor către elementele de câmp (numite de obicei RTU - remote terminal units); și o colecție de pachete software pentru culegerea datelor de la distanță și emiterea comenzilor către RTU. Sistemele SCADA actuale oferă de regulă control în buclă deschisă și comunicație pe distanțe foarte mari dar pot în același timp include și elemente de control în buclă închisă. Sisteme similare cu cele SCADA pot fi văzute de obicei în întreprinderi, stații de tratare, etc.

Aceste sisteme sunt denumite DCS (Distributed Control Systems) și au funcții similare cu sistemele SCADA dar culegerea datelor și emiterea comenzilor se referă în acest caz la o arie geografică mult mai restrânsă. Comunicarea între echipamente se face în acest caz într-un LAN (Local Area Network) și este mult mai sigură și la viteze mult mai mari decât în cazul sistemelor SCADA. Un sistem DCS implică aproape întotdeauna control în buclă închisă.

Pe de altă parte, sistemele SCADA acoperă o arie geografică cu mult mai mare și pot folosi o multitudine de medii de comunicare care pot fi mult mai puțin fiabile decât o rețea LAN. De aceea în acest caz se evită controlul în buclă închisă. Achiziția de date este realizată în primă instanță de RTU care scanează dispozitivele conectate (inclusiv intrările analogice și binare) cu o viteză suficient de mare. Sistemul central va scana la rândul său RTU-urile dar cu o viteză de scanare mai mică. Datele sunt procesate și dacă se detectează o alarmă aceasta va fi afișată în fereastra specială de alarmare. Datele achiziționate pot fi în principal de 3 tipuri: date analogice (cu

evoluție în timp și pentru care de obicei se păstrează istoricul evoluției în timp), date digitale (on/off) care pot avea atașate alarma unei anumite stări și date tip contor de impulsuri (sau indecși dacă sunt preluate de la dispozitive inteligente). Interfața operator este de obicei un ecran cu o reprezentare grafică a datelor citite de la echipamente. Datele binare se reprezintă de obicei ca simboluri dinamice pe un fundal static care își schimbă culoarea, forma, dimensiunea, etc. în funcție de starea mărimii, iar datele analogice se pot reprezenta ca numere sau în format grafic. Sistemul dispune de multe asemenea ecrane din care utilizatorul poate alege una, la un anumit moment.

Sisteme SCADA există implementate pe aproape toate arhitecturile hardware și pe toate sistemele de operare. Istoric vorbind aceste sisteme au folosit de-a lungul timpului arhitecturi sau extensii propriietate ale sistemelor de operare sau hard-ului pe care lucrau. Aceasta s-a datorat faptului că cerințele pentru sistemele SCADA au împins la limită tehnologia care există din punct de vedere al performanțelor cerute. În acest sens au fost dezvoltate module hardware sofisticate și s-au folosit baze de date în timp real (care stocau o mare parte din baza de date direct în memorie). De-a lungul timpului Unix a fost sistemul de operare care a primit o acceptare largă din partea utilizatorilor. După 1990 explozia calculatoarelor bazate pe procesoare Intel precum și performanțele pe care le au acestea coroborat cu dezvoltarea sistemelor de operare Windows au înclinat balanța în favoarea acestora. Astăzi aproape toate sistemele SCADA folosesc ca stații pentru vizualizare sisteme de operare Windows și ca servere platforme bazate pe tehnologia WindowsNT.

3.2.1. Funcțiile de bază ale dispecerului SCADA

Cea mai vizibilă parte a unui sistem SCADA este "dispecerul". Acesta este centrul nervos al sistemului și este alcătuit dintr-o componentă de achiziție și comandă a datelor (server de comunicație) și o componentă de afișare a datelor, de obicei numită HMI (Human - Machine - Interface). (În trecut era numit MMI - Man Machine Interface).

Principalele funcții ale dispecerului, așa cum au fost înfățișate anterior, sunt:

- achiziția de date (citirea datelor de la echipamentele RTU) ;
- stocarea datelor istorice de evoluție (trending);
- procesarea alarmelor (analizează datele citite de la RTU pentru a depista situațiile de funcționare anormale și alertează operatorul la depistarea unor astfel de situații);
- control (în buclă închisă sau supervizat – la cererea operatorului);
- afișarea grafică a datelor citite;
- generarea rapoartelor;
- hot standby (creșterea fiabilității sistemului se face prin folosirea unui echipament de rezervă care să preia funcțiile echipamentului principal în caz de defecțiune);
- interfațarea cu sisteme (exportul datelor către alte sisteme folosite de către beneficiar);
- securitate (controlul accesului la diferite componente din sistem);
- monitorizarea rețelei (monitorizează starea de funcționare a liniilor de comunicație);
- managementul bazei de date (posibilitatea adăugării de noi puncte în sistem, modificarea interfeței grafice, editarea graficelor și a rapoartelor);
- aplicații specializate (aproape orice pachet SCADA dispune de funcții speciale specifice domeniului concret în care activează – un exemplu ar fi detecția pierderilor în conductele de transport);
- sisteme expert (pachetul SCADA poate conține un sistem expert care să ajute în procesul decizional și permite modelarea dinamică a sistemului).

Caracteristica distinctivă a unui sistem SCADA este capacitatea sa de comunicare. Comparată cu sistemele DCS (Distributed Control Systems), care de obicei

lucrează într-o arie restrânsă geografic cum ar fi într-o întreprindere, un sistem SCADA acoperă de obicei arii geografice extinse și folosesc tipuri diferite (și de multe ori destul de nefiabile) de medii de comunicare. Un aspect important al tehnologiei SCADA îl reprezintă abilitatea sa de a garanta siguranța datelor comunicate prin aceste medii. Inițial sistemele SCADA foloseau linii de comunicație lente și, deci, eficiența era extrem de importantă. Trebuiau efectuate calcule complicate pentru determinarea volumului de date ce putea fi transportat și care nu satura mediul de comunicație. Pentru ca performanțele sistemului să nu fie degradabile se considera cazul cel mai defavorabil și se impunea funcționarea normală în acest caz.

Sistemele SCADA pot utiliza cele mai diverse medii pentru a realiza comunicația cu dispozitivele RTU. Dintre cele mai uzuale amintim:

- linii dedicate. Sunt linii de comunicație pe 2 fire disponibile între dispecer și RTU. Pe acestea se comunică de obicei multidrop în configurație RS485 sau point-to-point dacă se folosesc la capete modem-uri de linie închiriată. Al doilea mod permite linii de comunicație mai lungi, în primul caz distanța este limitată, așa cum am precizat anterior, la 1200m;
- linii telefonice închiriate; sunt la fel ca cele dedicate numai că sunt închiriate de la un deținător (de regulă operatorul de telefonie ROMTELECOM are asemenea linii);
- linii telefonice comutate (dial-up) au avantajul că sunt ușor de instalat și nu necesită un sistem dispecer foarte complicat. Implică însă costuri telefonice adiționale;
- Comunicație radio. În unele țări este probabil cea mai răspândită modalitate de comunicare în sistemele SCADA. Soluțiile și tehnologiile folosite în acest caz sunt mature și bine stabilite. Principalul dezavantaj este viteza de comunicare mică (până la 9600 bauds);
- Comunicație Ethernet. În ultimul timp a căpătat acceptare largă comunicarea de mare viteză în sisteme Ethernet. Deși nedeterministă, comunicarea Ethernet asigură practic performanțe care o recomandă ca fiind o tehnologie de viitor în sistemele SCADA. Folosită cu succes în industria calculatoarelor de ani de zile, tehnologia Ethernet poate fi implementată de diverse arhitecturi fizice tip LAN/WAN care oferă performanțe ridicate:
 - Rețea locală LAN;
 - Rețea de fibră optică pentru distanțe mari (zeci de km);
 - Rețea CATV (la nivelul unui oraș);
 - Rețea wireless (în banda liberă 2.4 GHz) pentru orașe nu foarte mari.

Tipurile de comunicare de bază în cazul sistemelor SCADA sunt:

Polling

La început RTU-urile nu erau înzestrate cu prea multă inteligență, iar sistemul central era forțat să efectueze toată munca. El interoga RTU-urile pentru a colecta datele și de asemenea putea emite comenzi către acesta. Acest mod de comunicare are câteva avantaje:

- este deterministic. Se poate calcula cu exactitate lățimea de bandă folosită și de asemenea se poate garanta un timp de răspuns determinist;
- Nu necesită dispozitive RTU inteligente;
- Nu există problema interferenței între RTU de vreme ce ele vorbesc doar când sunt întrebate.
- Este eficient în cazul în care parametrii monitorizați sunt puternic schimbători deoarece la orice interogare a masterului RTU-ul îi va furniza date noi.

Report by Exception

Pe măsură ce tehnologia informației a evoluat RTU-urile au devenit din ce în ce mai inteligente. Aceasta a deschis noi perspective privind funcțiile pe care le puteau

îndeplini acestea. A fost posibil ca RTU-urile să scaneze încontinuu intrările și să detecteze schimbările de stare sau valoare. RTU-ul înregistra aceste schimbări (împreună cu timpul) și, la cererea master-ului răspundea doar cu schimbările de date. Acesta este evident mult mai eficient în cazul proceselor cu o stabilitate în timp mare de vreme ce nu este nevoie ca să se transmită toate datele de la RTU la fiecare interogare.

De asemenea, schimbările pot avea o etichetă de timp foarte precisă lucru foarte benefic mai ales în electricitate (unde se cere acuratete de 1 ms). Această tehnologie a generat la rândul ei o serie de probleme. De exemplu pentru datele analogice se considera schimbare dacă data se modifica cu 2% (deci nu vor fi transmise datele în cazul unor modificări mai mici). Cea mai mare problema a fost însă cu datele care au o dinamică mare în timp. Acestea generează un volum foarte mare de date, uneori inacceptabil și forțează folosirea primei tehnologii.

Conceptul de RTU – Remote Terminal Unit

RTU-ul este un mic calculator în format industrial și de obicei include următoarele module:

- CPU și memorie;
- Memorie permanentă (nevolatilă) pentru stocarea programului și datelor;
- Posibilități de comunicare fie printr-un port serial fie prin modem on-board;
- Sursa de alimentare (de obicei cu UPS);
- Watchdog timer (pentru restart în condiții de eroare);
- Protecții electrice împotriva interferențelor electromagnetice;
- Interfețe I/O - DI/DO/AI/AO's.
- Alte interfețe specifice de citire a datelor;
- Ceas de timp real.

În fig. 3.10 este exemplificată o arhitectură RTU.

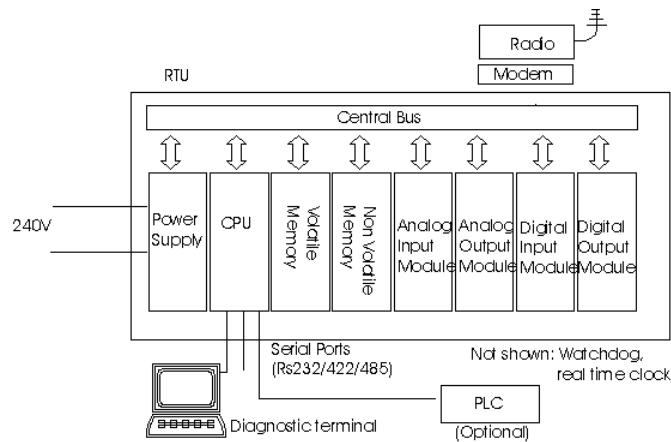


Fig. 3.10. Exemplu de arhitectură a RTU.

Funcții și specificații software ale RTU:

- sistem de operare în timp real. Poate fi un sistem de operare de timp real sau un program care are o buclă în care citește intrările, monitorizează interfețele de comunicație și emite comenzile;
- are drivere de comunicație cu dispozitivul master;
- are drivere de comunicație cu elementele de câmp și cu interfețele I/O;
- are posibilitatea de configurare a anumitor parametri cum ar fi rata de scanare;
- posibilitate de diagnostic;
- unele RTU pot include un sistem de fișiere care suportă download. Această facilitate este folosită pentru modificarea programelor și a configurațiilor.

Un RTU este de obicei un dispozitiv care asigură „prezența inteligentă” în teren. El achiziționează date de la elementele de câmp și poate avea funcții de conducere locală. De asemenea comunică sistemului central datele achiziționate și acționează ca un intermediar în emiterea comenzilor de la nivel central.

Există două tipuri de bază de RTU: „single board RTU”, care este compact și conține toate modulele I/O pe placă și „modular RTU”, care conține module separate pentru CPU, module I/O, module de comunicație, etc.

RTU-ul compact are un număr fix de pini pentru I/O de exemplu 16 intrări digitale, 8 intrări analogice și 4 ieșiri digitale. În mod uzual numărul de I/O nu poate fi extins.

RTU-ul modular este proiectat pentru a suporta extinderea numărului de intrări și/sau ieșiri. Extinderea se face cu I/O ce au capacități puteri ale lui 2 (2,4,8,16,...). Alte tipuri de module pot fi adăugate (de ex. modul de sincronizare timp GPS).

RTU-ul va scana intrările în mod normal cu o viteză de scanare destul de mare. De asemenea poate efectua procesări primare asupra datelor, asignarea informației temporale la date, etc. Anumite RTU pot iniția raportări către master deși în mod obișnuit masterul este cel care inițiază comunicația. De asemenea RTU-ul poate să proceseze alarme. La cererea masterului poate răspunde fie cu toate datele fie numai cu o parte din ele (care au suferit modificări etc).

RTU-urile sunt dispozitive speciale care sunt fabricate de obicei în serii mici, la comandă. În dorința de a reduce costurile RTU-urile se produc cu un număr limitat de funcționalități dar care este suficient aplicației în care sunt folosite. Există RTU-uri care pot avea sute de intrări și pot funcționa ca submastere pentru alte RTU-uri. Evident acestea vor fi mult mai scumpe decât un RTU de dimensiuni foarte mici care are 4 intrări analogice și 8 intrări digitale. Puterea de procesare merge de la procesoare de 4 biți, cu posibilități reduse, la sisteme complexe de tip PC cu posibilități de procesare foarte mari.

Câteva dimensiuni fizice ale RTU:

- Foarte mic – este de obicei un RTU compact care se alimentează din baterie și are doar facilități de pooling. De multe ori scanează măsurătorile de intrare doar când primește telegrama de citire;
- Mic – este un sistem capabil să efectueze măsurători ale senzorilor din când în când și să înregistreze aceste date în memoria nevolatilă. Poate fi echipat cu baterii și panouri solare pentru economisirea energiei;
- Mediu. Acesta este construit de obicei pe o arhitectura tip PC industrial și are facilități avansate de procesare și chiar afișare locală a datelor. Se construiește în jurul unei arhitecturi industriale cum ar fi ISA, VME, PC104 etc.;
- Mare. Acest tip de RTU acționează de obicei ca un submaster și are facilități avansate de control și interpretare a datelor.

Un aspect deosebit de important este conlucrarea permanentă între echipamentele și dispozitivele numerice care compun sistemul, astfel că putem vorbi mai degrabă de o prelucrare a datelor la nivelul rețelei de date ce leagă echipamentele numerice, care sunt în ultimă instanță microcalculatoare. Fiecare componentă a sistemului efectuează prelucrări ale informației disponibile local și livrează servicii (informații în formă prelucrată sau brută) celorlalte componente ale sistemului.

Costul cablurilor și al pozării lor în stații este destul de ridicat, deci într-o arhitectură distribuită, unde echipamentele de achiziție și comandă (PLC sau EAD) sunt amplasate cât mai aproape de echipamentele primare ce le comandă, acestea sunt evident mai mici decât într-o soluție centralizată. Pe de altă parte, progresul tehnologic face ca astăzi componentele cele mai importante ale unui calculator – procesorul și memoria - să aibă prețuri incomparabil mai mici decât în urmă cu numai câțiva ani.

Deși este discutabil, dacă la momentul implementării soluția cu un singur EAD centralizat este mai ieftină decât soluția cu mai multe EAD distribuite, semnificativ este că aceste costuri inițiale devin puțin importante atunci când este vorba de modificarea unei instalații existente sau de încorporarea unor facilități suplimentare în cazul EAD convenționale.

O comparație semnificativă se poate face în fig. 3.11. Între costurile totale ale unor implementări SCADA pe toată durata de viață a sistemului. În analiză s-a luat în considerare o structură a costurilor după cum urmează:

- costuri primare: costuri materiale, costuri de proiectare, costuri de punere în funcțiune;
- costuri secundare: întreținere, specializare personal, costuri de extindere și reconfigurare, costuri de documentare și factori de cost comerciali.

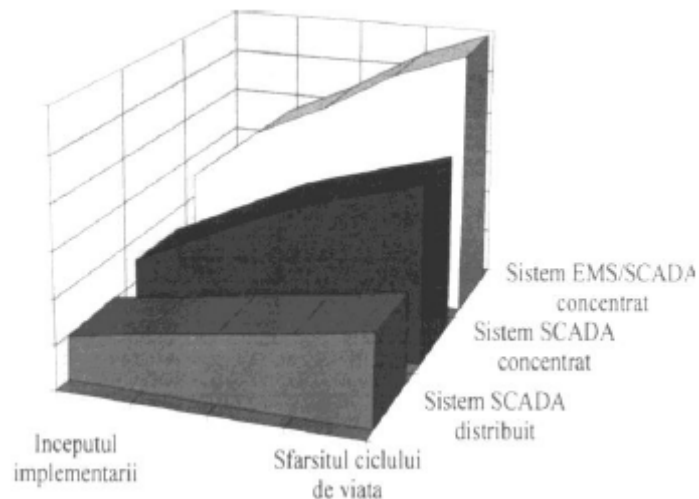


Fig. 3.11. Comparație între diverse tipuri de sisteme SCADA.

3.2.2. Structura software aleasă

Principala sarcină pentru realizarea sistemului de dispecerizare al Sistemului Hidrotehnic și de Navigație Porțile de Fier II o constituie armonizarea sistemelor deja existente și integrarea lor într-un sistem unitar care să permită conducerea eficientă a întregului ansamblu și să constituie în același timp sursa de informații pentru analize și rapoarte de funcționare. În acest context am considerat implementarea ultimelor tehnologii disponibile în ceea ce privește sistemele SCADA, protocoalele de comunicație și sistemele de tip EAD. Pachetul software ales pentru implementarea SCADA este Iconics Genesis32 versiunea 9. Acesta este un pachet modular, puternic și flexibil destinat utilizării în aplicații SCADA de mari dimensiuni. Iconics Genesis 32 se bazează în exclusivitate pe tehnologia OPC fiind dezvoltat de la bun început pentru a funcționa cu această tehnologie beneficiind astfel din plin de toate facilitățile oferite de tehnologia OPC. Această facilitate, denumită și OPC-to-the-core asigură maximum de interoperabilitate cu orice sistem de tip OPC existent pe piață. De altfel Iconics are implementat unul din cele mai mari sisteme SCADA din lume la Transneft-Russian Pipeline, sistem ce include 500 calculatoare, 2100 PLCuri și are aproximativ 800.000 taguri.

În fapt, pachetul Iconics este construit dintr-o suită de aplicații cu funcționalități specifice, dintre care cele mai importante sunt:

- **GraphWorX32:** Componenta care implementează interfața HMI de vizualizare pentru servere OPC și care deține o colecție impresionantă de simboluri deja definite pentru diverse industrii. Permite crearea interfeței operator prin adăugarea de simboluri, valori, animații, obiecte ActiveX, etc. pentru crearea schemelor care să corespundă cât mai fidel procesului modelat.
- **TrendWorX32:** Este componenta care asigură afișarea trendurilor (graficelor) istorice sau în timp real. Permite afișarea de tipuri diferite de trenduri (liniare, logaritmice, circulare, etc); oricare din parametrii de afișare (dimensiuni, fonturi, culori, etc) pot fi cu ușurință configurați/schimbați
- **AlarmWorX32:** Este un sistem complet de management al alarmelor și evenimentelor, permițând detecția, sortarea, filtrarea, vizualizarea, raportarea și acceptarea acestora.
- **ReportWorX:** Este componenta care permite rularea raportărilor și exportul lor în web. Cu acest instrument se pot crea rapoarte profesionale și poate folosi pentru schimbul de date cu nivelele superioare.

Componente adiționale din pachetul Genesis32:

- ProjectWorX32: Pentru managementul și dezvoltarea proiectelor Genesis32;
- DataWorX32: Pentru asigurarea redundanței și legăturii de date cu alte nivele (prin tunelare/agregare);
- ScriptWorX: Permite rularea de scripturi VBA multithreading;
- TraceWorX32: Utilitare pentru diagnosticare și depanare;
- VCRWorX32: Pentru vizualizarea prin tehnologie playback a datelor istorice;
- WebHMI: Interfața de vizualizare de tip web a datelor în timp real;
- AlarmWorX32 Multimedia: Pachet de alarmare multimedia (e-mail, SMS, telefon etc);
- Unified Data Manager: Asigură managementul centralizat al variabilelor de stare (cum ar fi filtre, expresii, etc).

La proiectarea soluției SCADA de la nivelul unui dispecerat s-a luat în calcul posibilitatea utilizării tehnologiei thin-client (adică afișarea în interfața web). Deoarece componenta WebHMI asigură operarea identică cu aplicația GenesisBrowser (client de vizualizare) în ceea ce privește performanțele grafice, modul de afișare și posibilitățile de operare (inclusiv comandă) s-a optat pentru utilizarea acestei tehnologii. Astfel, baza de date cu imagini de proces (scheme tehnologice) este menținută la nivel centralizat pe server, iar clienții (stațiile de operare/de lucru etc.) accesează aceste scheme printr-un simplu browser web. În acest fel modificările se pot face la nivel centralizat (și modificarea se reflectă imediat la toți clienții serverului) și, în același timp, politica de securitate se poate aplica, în mod unitar, tot de la acest nivel.

Pe serverul central (sau pe serverul central și pe cel redundant, în cazul redundanței) va rula serverul de baze de date **Microsoft SQL Server 2005** precum și componentele server ale pachetului **Iconics Genesis 32 v9**. Tot pe servere vor rula și componentele software pentru legătura cu EAD-urile de tip OPC Server. Stația de engineering va dispune de instrumente pentru configurarea componentelor server. Legătura între dispecerate se va realiza prin componenta **Iconics DataWorX** (care permite accesul de la distanță la datele din serverele OPC locale) pentru transferul bidirecțional de parametri, iar pentru transferul de fișiere se va folosi mecanismul de raportare și trimitere automată către stația de nivel superior a rapoartelor necesare.

La nivelul echipamentelor de câmp și a sistemelor de tip EAD s-au respectat total cerințele din caietul de sarcini, alegându-se tehnologii modulare de tip PLC care asigură mentenanța ușoară a sistemului și facilități sporite de autodiagnosticare.

Avantajele oferite de soluția propusă:

- Sistemul se bazează pe standardul deschis OPC, având implementată această tehnologie în modulul de bază (OPC-to-the-core). Acesta este un avantaj pentru cazul de față, unde există EADuri de tipuri și generații diferite, care vor trebui integrate în sistem;
- Managementul centralizat al configurațiilor asigură propagarea unei modificări imediat la toate stațiile client (un alt avantaj este că, dacă o stație de operare se strică, o altă stație poate să-i ia locul foarte ușor);
- Securitatea la nivel centralizat asigură controlul precis și ușor al drepturilor de acces;
- Nu este nevoie de dezvoltarea unei interfețe paralele pentru accesul din internet (bineînțeles că acest acces este securizat și permis doar în citire);
- Există și o oarecare experiență în dezvoltarea de servere OPC. Se dispune, de asemenea, de utilitarul OPC ToolWorX, care permite dezvoltarea rapidă de astfel de servere.
- Componenta ReportWorX permite editarea de rapoarte complexe, direct publicabile și accesibile, prin intermediul browserului web;
- Se pot realiza și sisteme răspândite pe arii geografice mari;
- Sistemul conceput este modular, ceea ce permite dezvoltarea ușoară și etapizată;
- Sistemul beneficiază de simulatoare de date OPC, cu care interfața poate fi dezvoltată sau validată, fără să fie nevoie de conexiunea reală cu echipamentele. În acest fel, interfața grafică poate fi validată chiar în timp ce modelul software de comunicație se implementează.

Pachetul de dezvoltare pentru produsul Iconics este gratuit, făcând posibilă implementarea interfețelor înainte de achiziționarea propriu-zisă a licenței.

3.3. Structura hardware a unui sistem informatic de proces

Așa cum a fost descris în prealabil, din punct de vedere hardware, un sistem informatic pentru aplicațiile din domeniul hidrotehnic este organizat pe trei nivele ierarhice.

Acestea sunt:

- **Nivelul „0”** sau nivelul de bază este format din totalitatea traductorilor și elementelor de execuție conectate la procesul urmărit și condus;
- **Nivelul „1”** este destinat cuplorului de proces, care este de fapt un calculator numeric care preia, prin fir fizic, informațiile de la traductoare și le transmite într-un format special și după un cod anume, la calculatorul central „Dispecer”, după eventuale prelucrări, prin rețeaua telefonică sau prin unde radio. Tot la acest nivel pot fi plasate și căile de comunicație, de la traductori la cuplor și de la cuplor la punctul dispecer;
- **Nivelul „2”** are ca suport fizic echipamentele de calcul (calculator, imprimantă, MODEM, UPS) de la punctul dispecer și de la nivelul de conducere sau supervizare.

Acest tip de sistem reprezintă o soluție modernă, cu posibilitatea de a fi verificat și autorizat metrologic. Implementarea unui astfel de sistem informatic de proces s-a făcut pornind de la necesitatea de a urmări și de a conduce de la distanță succesiunea operațiilor de ecluzare, procesul având o complexitate medie sau mare. Scopul este acela de a ajunge la un management eficient al resurselor, al consumurilor energetice și al fluxului de nave. O astfel de implementare impune utilizarea unui sistem ierarhic organizat pe cele trei nivele logice definite anterior.

Vom detalia în continuare elementele fizice care intră în configurația fiecărui nivel:

- **Primul nivel**, numit și „nivelul 0”, este constituit din elementele primare care interacționează cu procesul. La acest nivel are loc atât automatizarea procesului, ceea ce presupune realizarea unor funcții de reglare a principalelor variabile caracteristice, cât și preluarea și măsurarea unor mărimi de interes.

Nivelul „0” este constituit din :

- ❑ traductoare: de debit, de temperatură, de presiune, de nivel, de poziție, alte elemente de măsurare / detectare / analiză;
- ❑ dispozitive electronice de prelucrare, automatizare și afișare locală în teren: unități de comandă și reglare automată etc.;
- ❑ elemente primare și de execuție: porți cu servomecanisme, actuatori, robinete, vane de reglare, motoare de comandă, pompe, regulatoare diferențiale de presiune etc.

- **Al doilea nivel** ierarhic numit și „nivel1” este destinat cuplului de proces. La acest nivel se realizează preluarea de la elementele primare (traductoare, integroare, pompe etc.) a semnalelor analogice, în impulsuri sau în frecvență și convertirea acestor semnale în date numerice care pot fi prelucrate elementar și care pot fi ordonate într-un format special, pentru vizualizarea locală într-un centru local operativ (uzual în camera de comandă), precum și facilitarea transmisiei informației utile la calculatorul central al Dispecerului (care poate să se afle la orice distanță de toate centrele locale operative).

Tot la acest nivel se pot include:

- ❑ cartelele de comunicație: cartele Mbus, convertoare Mbus/RS232, adaptoare 4-20 mA/Mbus, dispozitive de achiziție și retransmitere etc.;
- ❑ modem-uri radio sau telefonice (mai puțin întâlnite în aplicația de la Porțile de Fier II);
- ❑ mediul fizic de transmisie: cabluri adecvate de legătură între punctele de măsură și centrul operativ.

- **Al treilea nivel** ierarhic al unui sistem informatic de proces îl constituie echipamentul de supervizare sau conducere. El are posibilitatea comunicării cu toate elementele componente ale nivelului „1” și având rolul unui „master” în acest sistem. Nivelul „3” este reprezentat de

- ❑ calculatorul central „Dispecer” pentru urmărirea și controlul global al datelor din sistem;
- ❑ rețea neoperativă de calculatoare PC pentru informare;

Calculatorul supervisor poate stoca cantități mari de informație. Informațiile se referă la istoria procesului și la toate mărimile esențiale ce trebuie urmărite. Acest nivel oferă operatorului de sistem facilități extinse de introducere și extragere date, formându-se astfel o bază de date cu informații utile pentru teleurmărire și teleconducere.

Funcțiile principale ale acestui sistem informatic sunt:

- achiziția, transferul datelor și emiterea comenzilor,
- întreținerea și actualizarea permanente a unei baze de date, care conține atât valori momentane cât și medii, contoare, date de evoluție și valori calculate;
- afișarea grafică a schemelor sinoptice, pe care sunt figurate valorile momentane ale mărimilor analogice și starea agregatelor. Depășirile de limite trebuie figurate prin culori vii, clipitoare;
- crearea și vizualizarea diagramelor de evoluție pentru mărimile analogice sau calculate;
- înregistrarea cronologică a evenimentelor (depășiri / reveniri în limite ale mărimilor analogice, schimbări de stare ale mărimilor binare);

- realizarea automată de calcule pentru determinarea unor mărimi specifice;
- editarea rapoartelor periodice la oră, zi și luna, care conțin date sintetice despre regimul de funcționare a instalațiilor tehnologice dispecerizate;
- arhivarea datelor timp de un an. Este o facilitate utilă pentru realizarea analizelor de regimuri și a diverselor statistici legate de funcționarea pe termen lung a instalațiilor;
- tipărirea la imprimantă color a rapoartelor și graficelor de evoluție;
- mecanism de alarmare vizuală și prin voce, în situațiile anormale de funcționare;
- introducerea manuală a unor parametri, necesari pentru calcule;
- posibilitatea definirii unor calcule specifice, export către Excel;
- acces din rețea, cu mai multe posturi active simultan;
- autodiagnoza stării de funcționare a echipamentelor sistemului și a liniilor de comunicație.

Din punct de vedere software, particularitățile programării sistemului de calcul cu microprocesor, destinat supravegherii și / sau conducerii în timp real a proceselor industriale, rezidă din următoarele condiții esențiale ce trebuie îndeplinite:

- posibilitatea sistemului de calcul de a răspunde instantaneu la semnalele emise de proces, în mod obligatoriu;
- cerința ca „timpul real” în care sistemul de calcul este ocupat cu o anumită operație, să fie strict corelat cu timpul în care se produc alte evenimente din proces (exterioare calculatorului);
- capacitatea sistemului de a primi datele direct din proces sau de a le transmite direct în proces prin intermediul unor convertitoare (analog-numeric numeric-analogice etc.);
- rezolvarea contradicției între capacitatea sistemului de calcul de a executa, la un moment dat, un riguros program și cerința procesului de a fi „servit” în n puncte simultan. Altfel spus, sistemul trebuie să facă față desfășurării simultane a diferitelor părți specifice ale procesului;
- posibilitatea programării tuturor categoriilor de procese (continue sau discrete);
- existența elementelor pentru testarea și depanarea eficiența a programelor.

În concluzie, pentru programarea unei astfel de aplicații, limbajul de programare utilizat trebuie să conțină trei categorii de instrucțiuni:

1. instrucțiuni care să permită programarea modului în care se desfășoară evenimentele din proces;
2. instrucțiuni care să precizeze locul în care se desfășoară evenimentele din proces;
3. instrucțiuni care se referă la modul sau intervalul de timp în care au loc evenimentele din proces.

Programele implementate, pe un astfel de sistem de calcul, se împart în:

- program de sistem;
- program de utilizare sau program utilizator.

Aspectele legate de componenta hardware și software vor fi discutate mai detaliat pe parcursul acestui capitol.

3.4. Integrarea, extinderea și upgradarea sistemului SCADA

3.4.1. Descrierea modului de implementare etapizată a sistemului

Având în vedere că, în cadrul dispeceratului Sistemului Hidrotehnic și de Navigație Porțile de Fier II sunt, de fapt, mai multe dispecerate grupate pe tipuri de

funcționalitate, din punct de vedere al sistemului SCADA, implementarea etapizată impune dezvoltarea unui model validat de către beneficiar, care să poată fi apoi aplicat și la celelalte obiective similare.

Interfețele grafice cu operatorul vor fi armonizate la toate punctele de lucru, astfel încât validarea unui tip de dispecer va însemna validarea mult mai rapidă a celorlalte.

Dezvoltarea modelului (pilotului) nu depinde de tipul echipamentelor subordonate (EAD-uri), astfel încât pentru implementarea proiectului pilot poate fi ales unul din obiectivele pentru care există toate driverele de comunicație. În etapele inițiale de validare se pot folosi date simulate pentru exemplificarea funcționării sistemului. În paralel cu dezvoltarea proiectului pilot, care să permită beneficiarului evaluarea și validarea tipului de sistem SCADA, se va proceda la dezvoltarea de drivere noi (unde este cazul) astfel încât, după validare, implementarea la celelalte obiective similare să poată fi făcută într-un timp foarte scurt.

Etape:

- inventarierea tipurilor de EAD-uri din sistem;
- analiza driverelor existente și inventarierea driverelor noi ce trebuie dezvoltate;
- stabilirea obiectivului pilot pentru Dispecerul Principal; la stabilirea obiectivului pilot se va avea în vedere tipul de EAD-uri (în sensul existenței driverelor) precum și posibilitatea de a asigura legătura cu obiectivele subordonate în timp util;
- implementare și validare pilot Dispecer Principal;
- implementare și validare legătură cu punctele de măsură;
- armonizarea tuturor operatorilor din sistem.

3.4.2. Cooperarea la scrierea unui nou driver în cazul cuplării cu echipamente / sisteme noi

O cerință majoră a Sistemului SCADA o reprezintă posibilitatea integrării unor sisteme tehnologice, instalații sau echipamente livrate de către alți furnizori, respectiv, furnizarea la cerere de specificații de programare pentru acel driver și cooperarea la dezvoltarea de drivere noi. Există experiență în dezvoltarea de astfel de drivere și utilizarea unui toolkit care permite dezvoltarea foarte rapidă a acestor drivere. În esență, se pot furniza la cerere specificații de programare pentru un anumit driver.

Pachetul software propus, Iconics Genesis32 v9 permite schimbul de informații între elementele sistemului pe baza standardului deschis OPC. Tehnologia OPC este frecvent utilizată în sistemele SCADA și permite eliminarea dezvoltării de interfețe dedicate și drivere tradiționale necesare în transmiterea datelor în cadrul unui sistem informațional de proces.

Se rezolvă în acest mod interoperabilitatea între diverse sisteme și platforme de dezvoltare ceea ce creează cadrul unei extinderi ulterioare relativ ușoară atât pe orizontală cât și pe verticală. În fapt, tehnologia OPC cuprinde o serie de reguli și standarde menite să unifice funcționalitatea diverselor sisteme SCADA ceea ce permite atât o implementare mai rapidă a unui sistem nou cât și posibilitatea interconectării acestuia cu alte sisteme provenite de la alți furnizori. Este un sistem deschis bazat pe standarde publice utilizabile de orice dezvoltator sau integrator de astfel de sisteme.

Toți producătorii importanți de echipamente au dezvoltate deja servere OPC care permit altor aplicații să comunice cu hardware-ul specific fiecăruia.

Există șanse foarte mari ca pentru fiecare tip de echipament existent în punctele/modulele/nodurile termice să existe deja un server OPC care să poată fi utilizat direct.

Pentru cazurile în care un astfel de driver nu este disponibil se poate dezvolta unul. Folosind un instrument pentru conceperea rapidă de servere OPC (Iconics ToolWorX) se poate dezvolta un driver într-un timp foarte scurt. Experiența în dezvoltarea de astfel de drivere (pentru produse hardware proprii sau pentru terți, de ex. Danfoss) ne îndreptățește să credem că nu există nici un fel de probleme pentru integrarea cu succes în sistem a oricărui dispozitiv hardware de la terți.

În ceea ce privește furnizarea de specificații, deoarece protocolul OPC este unul public (disponibil la <http://www.opcfoundation.org>) singurele specificații de programare în acest caz vor fi specificațiile de protocol pentru comunicația cu dispozitivul hardware terțiar, specificații pe care le deține producătorul respectivului echipament. (Un driver în cazul de față poate fi identificat cu un convertor software de protocol de la specificația publică OPC la protocolul privat specific fiecărui tip de hardware).

3.4.3. Asigurarea sincronizării echipamentelor din sistem

Principala componentă de sincronizat pentru sistemul de față este timpul. În vederea raportării cu exactitate a evenimentelor și schimbărilor de stare este esențial ca timpul de lucru al echipamentelor implicate să fie același sau să se încadreze într-o anumită abatere permisă. De asemenea trecerea la ora de vară/iarnă trebuie să se facă în același moment pe toate sistemele.

Deoarece în sistem sunt elemente de generații diferite și cu performanțe diferite, sincronizarea timpului se va face diferențiat, în funcție de posibilitățile tehnice ale fiecărui echipament. Astfel, pentru echipamentele de tip PC (pe care rulează Windows) se va utiliza serviciul de sincronizare a timpului time-server. Sincronizarea se va face cu serverul de domeniu sau cu un internet time server. Sincronizarea celorlalte echipamente din sistem (EAD-uri) se va face diferențiat, în funcție de tipul fiecărui echipament și, se vor utiliza, în principal, două tehnologii:

- trimiterea timpului prin mecanisme de broadcast către echipamentele conectate la sistem;
- actualizarea timpului echipamentelor de tip EAD în cadrul procedurii de scanare a acestuia (sincronizarea se face de către echipamentul de nivel superior și nu are loc la fiecare scanare ci la un interval/număr de scanări, programabil).

Sincronizarea configurației sistemului se face prin managementul centralizat al acestuia, la nivelul serverului/serverelor aferente fiecăruia din dispecere; fiecare stație de afișare sau de lucru va prelua configurația de la serverul corespunzător.

3.4.4. Extinderea și upgradarea sistemului (hardware, software, comunicație)

Extinderea sistemului

Extinderea sistemului poate viza:

- Adăugarea unei noi stații în sistem (de tipul stațiilor de lucru).
 - Hardware: Pentru aceasta trebuie achiziționat un calculator cu performanțe similare sau superioare celor descrise în caietul de sarcini;
 - Software: Pe o astfel de stație software-ul rulează într-o interfață de tip browser, dar își ia licența de funcționare de pe server pe sistemul conexiunilor concurente. Dacă este necesar, se va achiziționa o licență suplimentară pentru acest PC.
 - Comunicație: Calculatorul se va cupla la rețeaua aferentă locației în care este montat.

- Adăugarea unui nou echipament de tip EAD
 - Hardware: Acest echipament trebuie sa poată fi accesat prin TCP/IP (trebuie să dețină port de comunicare Ethernet RJ45);
 - Software: Pentru integrarea fără probleme a acestui echipament este necesară existența unui server OPC pentru el (condiție care poate fi pusă înainte de achiziție);
 - Comunicație: El va comunica cu nivelul superior prin infrastructura de tip FO/CATV/radio existentă, prin care comunică și celelalte echipamente.

Upgradarea sistemului

- Upgradarea hardware
 - Echipamentele de tip PC acestea pot fi înlocuite cu echipamente mai performante;
 - Echipamentele de tip EAD pot fi upgrdate cu versiuni mai noi, asigurând compatibilitatea protocolului de comunicație cu cel existent, sau oferind un OPC Server pentru echipamentul de tip nou.
 - Comunicația: orice upgrade la infrastructura sau echipamentele de comunicație nu influențează sistemul, subsistemul de comunicație fiind transparent pentru sistemul SCADA.
- Upgrade software
 - Sisteme de operare: dacă este necesar sistemele de operare pot fi upgrdate dar nu vor aduce beneficii semnificative aplicației
 - Software de bază (pachet Iconics Genesis32 v9). În pachetul de bază este inclus upgrade gratuit și suport tehnic pentru 2 ani.

Comunicația în Iconics Genesis32 se bazează pe standardul OPC. În acest sens noțiunea de driver este echivalentă cu cea de server OPC. Lista de servere OPC disponibile se regăsește în documentația aferentă produsului. Pachetul software nu se livrează cu toate driverele (serverele OPC). Deoarece nu este cunoscut tipul de echipamente de tip EAD care poate fi adăugat ulterior, a fost inclus un server OPC pentru comunicația IEC 60870-5 care este cerut în mod explicit. Pentru celelalte tipuri de EAD-uri driverele (serverele OPC) se vor achiziționa sau dezvolta ulterior, în momentul cunoașterii cu exactitate a tipului de comunicație cu fiecare EAD. Nu există o limitare în ceea ce privește numărul de drivere suportate simultan de către sistem. Adăugarea unui driver nou (în fapt un server OPC nou) nu necesită recompilare, link-editare sau resetare. Considerăm numărul de drivere cu care se livrează sistemul relevant pentru cazul de față doar în măsura în care acestea includ drivere necesare pentru comunicația cu echipamentele existente în cazul aplicației de la Porțile de Fier II. Există posibilitatea ca un produs să aibă foarte multe drivere, dar nici unul să nu se potrivească cu echipamentele din teren, fiind astfel necesară dezvoltarea de drivere noi.

Pentru sistemul SCADA Iconics Genesis32 mediul de dezvoltare al aplicației este gratuit (configuration mode), licențele sunt necesare numai pentru operarea „runtime.” În cazul de față, în care sistemul se implementează unitar, licențele de software SCADA au fost incluse în listele de cantități. Pentru upgrade gratuit pe o perioadă de 2 ani a fost inclus un plan de suport tehnic SupportWorX pentru întregul sistem. Licențele depind de numărul de puncte de măsură, dar pentru sistemul de față s-au considerat doar licențe cu număr nelimitat de puncte. Licențele nu depind de numărul de utilizatori ci de numărul de stații pe care rulează aplicația. Se poate utiliza pentru licență cheie hardware sau software. În cazul de față este recomandat și se va folosi o cheie hardware. Licențele includ toate utilitățile de care este nevoie pentru sistemul de față. În ceea ce privește driverele, este inclus un driver pentru IEC 870-5.

Întreg mediul de configurare și dezvoltare pentru pachetul Iconics este gratuit. Pentru dezvoltarea de servere OPC există utilitarul Genesis OPC ToolWorX, dar nu considerăm că trebuie achiziționat de beneficiar, deoarece se presupune că nu va dezvolta el însuși servere OPC. De asemenea, utilitarele pentru programare/ setare/ diagnosticare EAD-uri sunt oferite gratuit.

Un defect hardware sau software se reflecta în funcționarea sistemului după cum urmează:

- defectarea hardware/software a unei stații de lucru/stație operator sau stația de engineering nu influențează funcționarea sistemului, rolul ei putând fi preluat de către o alta stație din sistem;
- pentru cazul serverelor redundante, dacă unul din servere se defectează hardware/software rolul lui este preluat imediat de celalalt server din sistem (rezerva caldă);
- defectarea hardware/software a serverului, în cazul neredundanței duce la nefuncționarea sistemului;
- nefuncționarea unei porțiuni din rețeaua de comunicație se reflectă în sistem prin lipsa datelor de la echipamentele subordonate afectate de lipsa comunicației. Acest lucru este semnalizat prin alarme la dispecer;
- În general, defectarea unui element de la nivelul de jos (senzor, plăci de achiziție, EAD) este depistată imediat de sistem prin mecanisme de detecție simple, cum ar fi valori în afara domeniului de măsură fizic al aparatului (detecție fizică) fie prin mecanisme mai complexe care iau în calcul mai mulți parametri de intrare pentru stabilirea defectului (detecție logică). În oricare din cazuri, sistemul de alarmare va informa operatorul despre defectul apărut.

Sistemul SCADA comunică cu echipamentele de nivel inferior prin TCP/IP și rețea Ethernet. Utilizarea unui mediu public de comunicație (cum este sistemul CATV) poate ridica probleme de securitate mai ales în ceea ce privește comunicația între EAD-uri și dispecerul aferent fiecărui EAD (având în vedere mai ales emiterea comenzilor). Comunicarea pe protocoale standardizate, disponibile publicului larg, combinată cu diversitatea de echipamente de tip EAD din sistem crește insecuritatea sistemului în ansamblu și-l face vulnerabil unui atac din exterior. Comunicația între punctele de lucru poate fi securizată, deoarece echipamentele folosite sunt de tip PC și dispun de instrumente software pentru realizarea acestei securizări (în plus fiecare dispecer dispune de un router care poate fi configurat în acest sens). Dată fiind diversitatea echipamentelor de tip EAD, singura metodă eficientă de securizare este crearea unei rețele de tip VPN între dispecer și EAD-uri.

Accesul în sistem, pe bază de utilizatori și parole este controlat de componenta Iconics Security Server. Nu există o limitare în ceea ce privește numărul de utilizatori care pot accesa simultan aplicația ci în numărul de calculatoare (stații de lucru) care pot accesa concurrent aplicația. Set pot defini liber grupe de utilizatori și se pot stabili drepturile pentru fiecare grupă în parte (incluzând: doar vizualizare, acces la arhivă și rapoarte, permisia de a da comenzi, permisia de a configura sistemul, permisia de a administra sistemul). Drepturile se pot stabili de asemenea și pentru fiecare utilizator în parte, drepturile efective ale acestuia fiind cele ale grupului/grupelor din care face parte plus drepturile sale individuale. Lista utilizatorilor poate fi preluată și de la un server de domeniu.

3.5. Concluzii

Domeniul aplicațiilor hidrotehnice și de navigație fluvială este un domeniu care poate beneficia de avantajele tehnicilor moderne de reglare și de automatizare, mai ales având în vedere lipsa de multe ori a acestor sisteme și gradul de uzură ridicată a celor existente.

Indiferent de localizarea automatizării, în ansamblul întregului sistem de aplicații hidrotehnice, putem spune, din nou, că nu există un algoritm unic, predestinat a rezolva orice problemă care ar putea apărea în întregul ansamblu.

Mai mult chiar, nu putem afirma nici existența unor soluții tehnice unice și general valabile, bazate pe aplicarea unor principii, metode sau raționamente din domeniul ingineriei de sistem.

Domeniul automatizărilor este, la ora actuală, într-o dinamică fără precedent, fiind impulsivat pe de-o parte de evoluția echipamentelor „hardware”, dar mai ales de apariția unor noi concepte privind componenta „software”, adică în primul rând de apariția unor noi structuri ale sistemelor de automatizare industrială, de apariția unor noi modele matematice, de optimizarea permanentă a algoritmilor și structurilor deja implementate. Optimizarea rezultatului final al muncii inginerului de sistem este o necesitate, știut fiind că, pe lângă avantajele economice evidente pe care le aduce orice „inovație” inclusă în ansamblul automatizării, se disponibilizează în acest sens și noi resurse hardware, iar cele existente sunt exploatate la un nivel optimal. Mai mult chiar, prin clarificarea aspectelor legate de procesul controlat în sine și de metoda de control care va fi aplicată, se deschid noi posibilități de proiectare a automatizării respective sau a altora asemănătoare.

Conceperea unei scheme de automatizare performante nu poate fi făcută decât luând în calcul și modelul matematic al procesului controlat precum și cel al metodei de control. Formularea matematică a problemei de optimizare trebuie făcută astfel încât să permită conceperea soluției de optimizare, fără a genera un număr important de restricții. Poate cel mai important aspect legat de conceperea unei automatizări îl constituie modul de conlucrare Om – Sistem de automatizare. Dincolo de aspectele pur filozofice sau psihologice, artefactul uman este conceput de către oameni, pentru a servi unor necesități ale oamenilor, fiind utilizat de către oameni. Factorul uman, supus legilor hazardului și erorilor, reprezintă totuși entitatea care supraveghează întregul proces, până la apariția sistemelor de inteligență artificială performante. Colaborarea între om și sistemul de automatizare se recomandă a se face combinând cele două modalități descrise în acest capitol, anume pe verticală, respectiv pe orizontală.

Inteligența artificială, indiferent de gradul de complexitate al sistemului de automatizare, reprezintă alternativa cea mai performantă în proiectarea unei automatizări fiabile. Chiar dacă domeniul este într-o dinamică permanentă, apariția sistemelor cu adevărat „inteligente”, capabile de autoînvățare, memorare a istoricului și mai ales de elaborare a unor decizii pe baza celor învățate.

Sistemele expert sunt, pentru moment, o alternativă aplicabilă, la nivelul resurselor software sau hardware existente la ora actuală. Singura dificultate reală care apare în conceperea unor astfel de sisteme constă în stabilirea setului de reguli, respectiv al setului de date. Nu există nici în acest caz o metodă general valabilă și utilizabilă. Implementarea unor sisteme expert la nivelul dispecerului sistemului hidrotehnic este o necesitate, cerută în primul rând de optimizarea automatizării, fără a mai fi considerată o „tendență” sau o „modă trecătoare”.

Sisteme SCADA există implementate pe aproape toate arhitecturile hardware și pe toate sistemele de operare. Istoric vorbind aceste sisteme au folosit de-a lungul timpului arhitecturi sau extensii proprietare ale sistemelor de operare sau hardware pe care lucrau. Aceasta s-a datorat faptului că cerințele pentru sistemele SCADA au împins la limită tehnologia care există din punct de vedere al performanțelor cerute. În acest sens au fost dezvoltate module hardware sofisticate și s-au folosit baze de date în timp real (care stocau o mare parte din baza de date direct în memorie).

De-a lungul timpului Unix a fost sistemul de operare care a primit o acceptare largă din partea utilizatorilor. După 1990 explozia calculatoarelor bazate pe procesoare

Intel precum și performanțele pe care le au acestea coroborat cu dezvoltarea sistemelor de operare Windows au înclinat balanța în favoarea acestora. Astăzi aproape toate sistemele SCADA folosesc ca stații pentru vizualizare sisteme de operare Windows și ca servere platforme bazate pe tehnologia WindowsNT, XP sau chiar Vista.

Interfața Om – Sistem de automatizare, care este localizată la nivelul calculatoarelor cu funcție de dispecer sub forma unor aplicații software, în situația automatizărilor din domeniul sistemelor hidrotehnice, trebuie concepută într-un mod cât mai simplu.

Trebuie încurajat conceptul „user – friendly”, dar fără a diminua din eficiența procesului de automatizare în ansamblu. Acest aspect va fi tratat mai în detaliu în capitolele următoare.

3.6. Contribuții personale

Capitolul 3 este rezultatul, în primul rând, al unei ample documentări din literatura de specialitate.

Dintre principalele contribuții ale autorului amintim:

- Sinteza critică a unui număr mare de referințe bibliografice din domeniu;
- Descrierea celor mai noi metode, tendințe și principii de automatizare, utilizate în domeniul sistemelor hidrotehnice și de navigație;
- Formularea și studiul unor modele matematice capabile să contribuie la definitivarea soluției de automatizare aleasă;
- Analiza obiectivă a modului de conlucrare Om – Sistem de Automatizare;
- Evaluarea soluțiilor de automatizare aplicabile în cazul sistemelor hidrotehnice și de navigație;
- Propunerea unor soluții de automatizare bazate pe conceptul de sistem expert;
- Justificarea aplicabilității sistemelor expert în cazul concret al conducerii proceselor din domeniul sistemelor hidrotehnice și de navigație.

4. METODE ȘI POSIBILITĂȚI DE PROIECTARE A BAZELOR DE DATE PENTRU SISTEME INFORMATICE DE PROCES. FILTRAREA INFORMAȚIILOR LA NIVELUL DISPECERATULUI HIDROTEHNIC

4.1. Configurația bazelor de date

Un sistem de teleurmărire nu este complet dacă nu oferă soluții pentru înmagazinarea datelor și nu dispune de funcții eficiente de accesare a acestor informații în orice moment. Bazele de date constituie instrumentul de stocare a informației, care ulterior va fi valorificată.

O bază de date reprezintă un ansamblu de date integrat, anume structurat și dotat cu o descriere a acestei structuri. Descrierea poartă numele de dicționar de date sau meta – date (informații despre date) și creează o independență între datele propriu – zise și programe.

Conform acestei definiții consacrate, o bază de date este mai mult decât o colecție de fișiere. Ea include, pe lângă aceasta, dicționarul de date și o descriere a relațiilor dintre înregistrări, descriere chemată și utilizată pe întreaga durată a prelucrării informațiilor.

Structura evidențiază nu numai legăturile virtuale între obiecte și caracteristicile lor, dar și corespondența între adresa virtuală și adresa fizică a datelor pe suportul fizic (hard – disc, dischete, CD-uri etc.). Adresa fizică este, evident, funcție de spațiul real de stocare disponibil al bazei de date și corespunde algoritmului de transformare propriu sistemului de gestiune și tipului de calculator avut în vedere.

Structura este legată nemijlocit de dicționarul datelor unei baze de date și acționează prin intermediul acestuia, atât în înscrierea cât și în regăsirea informațiilor, selecția după anumite criterii, etc. Se poate spune, fără riscul de a greși, că orice operație în baza de date se efectuează prin intermediul structurii și al informațiilor conținute în baza de date.

Această interdependență se referă la două niveluri de înțelegere:

- În primul rând există o interdependență absolută a programelor sursă, scrise într-un limbaj apropiat de limbajul natural, numit limbajul de manipulare a datelor (*Data Manipulation Language*) față de ordinea și mărimea caracteristicilor obiectelor, acestea din urmă fiind descrise cu ajutorul limbajului de descriere a datelor (*Data Description Language*);
- Dacă ne referim însă la programul – obiect (executabil), rezultat în urma interpretării (compilării) șirului de instrucțiuni, orice modificare a geografiei și structurii datelor este semnificativă. Atâta timp cât structura bazei de date și spațiul real pentru care a fost definită această structură rămân neschimbate, programele rămân valabile. Dacă una dintre ele se modifică, de exemplu spațiul real devine la un moment dat insuficient pentru stocarea informației, întregul sistem trebuie reconvertit la noii parametri: datele trebuie salvate și reintroduse în noua bază de date, iar programele trebuie recompilate.

Sistemele de gestiune a bazelor de date (SGBD) pun de obicei la dispoziția utilizatorului programe speciale prin care recalculază și reconvertesc datele și programele din bază la schimbarea spațiului fizic, fără a mai fi necesară recompilarea tuturor programelor existente. Acesta constituie de fapt cel de-al doilea nivel de înțelegere a independenței datelor și programelor în cadrul unei baze de date.

4.1.1. Tipuri de baze de date

În aplicațiile curente, care vizează atât sistemele de telegestiune și monitorizare, cât și accesarea, prelucrarea și ordonarea rapidă a informației, întâlnim trei tipuri de baze de date:

- *Baze de date structurate relațional*

Acest tip de baze de date este cel mai simplu, atât din punct de vedere al definirii, cât și ca posibilitate de utilizare și sferă de aplicații. În mod paradoxal, dată fiind simplitatea lui, el s-a răspândit cel mai târziu, doar după apariția calculatoarelor personale profesionale.

Sistemul de baze de date relațional are aceeași structură fizică cu datele care trebuie prelucrate. De multe ori, acestea din urmă se prezintă sub forma unor tablouri (relații), cu linii și coloane. Liniile constituie obiectele, iar coloanele constituie atributele care caracterizează aceste obiecte.

Baza de date relațională se prezintă sub forma unui tablou. Fiecare înregistrare are o lungime constantă, fiecare rubrică (atribut) are o anumită caracteristică declarată la generarea bazei (șir de caractere, valoare numerică, valoare logică) și o lungime fixă prestabilită.

Căutarea se face parcurgând secvențial toate articolele și comparând criteriile. Rezultatul este ADEVĂRAT sau FALS pentru fiecare comparație. Articolele (obiectele) care au răspuns ADEVĂRAT la toate criteriile impuse sunt selectate.

Pentru ușurința calculului se pot memora anumite subansambluri create prin căutări anterioare sub formă de subansambluri indexate în raport cu unul sau altul din criterii, în așa fel încât regăsirea să se facă ulterior prin interconectarea acestor subansambluri.

Aplicațiile dezvoltate de către autor pentru operează preponderent cu baze de date tabelare. În fig. 4.1 este prezentat un exemplu de bază de date tabelară, vizualizată cu ajutorul software-ului, dezvoltat de către autor în cadrul aplicațiilor implementate la Sistemul Hidrotehnic și de Navigație Porțile de Fier II.

- *Baze de date structurate în mod ierarhic*

Bazele de date ierarhice sunt constituite pentru a facilita informația organizată ierarhic. Diagramele asociate unor astfel de baze de date sunt de tip arborescent, fiecare element fiind subordonat unui singur element și numai unuia, aparținând bazei.

Spre deosebire de structura sub formă de tablou, în care se prezintă înregistrările în bazele de date relaționale, într-o bază de date ierarhice, dependența unui segment de alte segmente de date de nivel superior se exprimă printr-un punctuator (adresă). În acest fel se face economie considerabilă de spațiu și se simplifică regăsirea informațiilor din bază. Timpul de prelucrare, în general, se scurtează. Pentru volume mari de date și prelucrări intense, acest lucru constituie un avantaj. Principalul dezavantaj îl constituie lipsa de suplețe la schimbarea procedurii de prelucrare.

NIVELE - SAGEATA - EFORTURI - TEMPERATURI

NIVELE [cm], SAGEATA [mm], EFORT UNITAR [daN/cm²] (28 linii)

mom	ora	nivam	nivav	sag	TMS1	TM1	TMD1	TMS2	TMD2	TMS3	TM3	TMD3	TT1	TT2
0	164649	5200	5200	0	112	61	74	-13	-9	133	-95	129	5	
49	164747	5200	5200	0	112	61	74	-13	-9	133	-95	129	5	
53	164752	5200	5200	0	112	61	74	-13	-9	133	-95	129	5	
61	164801	5200	5200	0	112	61	74	-13	-9	133	-95	129	5	
63	164803	5350	5200	0	158	112	130	48	57	205	-18	211	5	
65	164805	5410	5200	0	189	146	168	89	102	253	33	266	5	

COTE - PRESIUNI

COTE [mm], PRESIUNI [bar] (56 linii)

mom	ora	cotas	cotad	press	presd	TT1
0	164649	0	0	0	0	5
3	164653	30	30	140	0	0
6	164656	150	150	140	0	0
10	164701	270	270	140	0	0
12	164703	600	600	140	0	0
13	164814	600	600	140	0	0

DEPLASARI

DEPLASARI sectii [mm] (6 linii)

mom	ora	dir_X	dir_Y1	dir_Y2	dir_Z	cotas	ni
0	164649	0	0	0	0	0	0
49	164747	0	0	0	0	0	0
53	164752	0	0	0	0	0	0
96	164838	0	0	0	0	0	0
101	164844	0	0	0	0	0	0
103	164905	0	0	0	0	0	0

AVARII

5 avarii

data	ora	avarie	mom	cotas	cotad	press	presd	nivam	nivav	sag	dir_X	dir_Y
01.04.03	164747	Desincr. dr. sus 1 (50mm)	49	13200	13250	144	0	5200	5200			
01.04.03	164752	Desincr. dr. sus 2 (70mm)	53	14700	14775	14	0	5200	5200			
01.04.03	164838	TMD3 lim.1 (1106daN/cm2)	96	16650	16650	0	0	6340	5200			
01.04.03	164844	TMS3 lim.1 (1106daN/cm2)	101	16650	16650	0	0	6520	5200			
01.04.03	164849	TM3 lim.1 (1107daN/cm2)	106	16650	16650	0	0	6670	5200			

Generate grafice

Fig. 4.1. Bază de date tabelară a aplicației de la Porțile de Fier II.

- **Baze de date structurate în rețea**

Bazele de date „în rețea” seamănă cu cele ierarhice, diferența constând în faptul că un „copil” poate avea mai mult decât un singur „părinte”. Această structură se bazează pe conceptul de ansamblu (set).

Într-un sistem de bază de date în rețea, baza de date este constituită dintr-o colecție de seturi. Fiecare set se compune dintr-o sumă de înregistrări, acestea din urmă fiind asemănătoare cu cele din sistemul relațional (tabelar), numai că lungimea lor este variabilă.

O înregistrare poate fi rezultatul compunerii mai multor seturi.

Aceste două ultime structuri de baze de date sunt mai dificil de implementat, fiind mai rar întâlnite în aplicațiile pentru monitorizarea sistemelor hidrotehnice.

4.1.2. Sisteme de gestiune a bazelor de date

Sistemul de gestiune a bazelor de date (SGBD) este acel sistem de programare care facilitează și supervizează introducerea de informații în baza de date, actualizarea și extragerea datelor din bază, controlul și autorizarea accesului la date, precum și asigurarea unei independențe între structura bazei de date și programele de aplicație.

Un sistem de gestiune a bazelor de date trebuie să fie capabil să îndeplinească următoarele funcții:

- **Funcția de descriere**

Permite definirea structurii datelor și a relațiilor dintre acestea. De asemenea, tot prin intermediul acestei funcții se definesc și condițiile de acces la

informațiile conținute în baza de date. Descrierea structurii se realizează cu ajutorul unui limbaj de descriere a datelor (LDD) propriu fiecărui sistem de gestiune. Există totuși și unele elemente comune de descriere aproape unanim acceptate, cum ar fi (cu exemplificare în situația aplicației de la Pdf II):

- **Articolul (câmpul)** reprezintă cel mai mic element al unei structuri logice sau virtuale care poate fi identificat. El este asociat, de regulă, unei valori. Fiecare câmp conține valorile mărimilor indicate, exprimate în unitățile de măsură corespunzătoare, la un anumit moment de timp specificat de utilizator.
- **Subgrupul** reprezintă o primă grupare de câmpuri sub același identificator. De exemplu, pentru aplicația amintită mai sus, un subgrup este alcătuit din toate câmpurile corespunzătoare unui motor anume.
- **Grupul (sau segmentul)** reprezintă un ansamblu de articole și subgrupuri corelate logic, care dispun de un nume și o caracteristică comună. În cazul bazei de date de la Pdf II, un grup corespunde, de exemplu, unei automatizări aferente unei porți, adică este suma subgrupurilor care includ câmpurile cu mărimile preluate pe fiecare senzor aferent automatizării respective.
- **Înregistrarea** este o reuniune de articole, subgrupe și grupuri de date, la un anumit moment (de exemplu data de 1 aprilie 2004, ora 11:52:12);
- **Fișierul** reprezintă o reuniune de mai multe înregistrări (de exemplu pentru o zi).

Pentru aplicațiile care vizează numai sesizarea unor evenimente, structura bazei de date este simplificată, conținând următoarele articole, așa cum rezultă și din fig. 4.2:

The screenshot shows a software interface titled 'MDIForm1' with a 'Raport Detaliat' window. The main window displays a table of sensor data with columns for 'mom', 'ora', 'nivam', 'nivav', 'sag', and various 'TMS' and 'TT' sensors. A 'GRAFICE' dialog box is open, allowing the user to select which elements to display in a graph. The dialog has checkboxes for 'TMD3', 'TM3', 'TMS3', 'TMD2', 'TMS2', 'TMD1', 'TM1', and 'TMS1'. The 'TMS2' and 'TM1' checkboxes are checked. Below the dialog, there is a 'DEPLASARI' table with columns for 'mom', 'ora', 'dir_X', 'dir_Y1', 'dir_Y2', 'dir_Z', 'cotas', and 'ni'. At the bottom, there is an 'AVARII' table with columns for 'ad', 'press', 'presd', 'nivam', 'nivav', 'sag', 'dir_X', and 'dir_Y'. A 'Generare grafice' button is located at the bottom of the main window.

Fig. 4.2. Selectarea elementelor afișabile dintr-o bază de date.

De exemplu, în situația bazei de date de mai sus se pot selecta, în cadrul grupului, numai anumite timbre tensiometrice, pentru stabilirea eforturilor și a deplasărilor.

Acestea sunt doar două exemple de baze de date, relative la automatizările din sectorul hidrotehnic. În fiecare dintre aplicațiile descrise, se pot adăuga și câmpurile corespunzătoare alarmelor specifice.

Relațiile între diferitele elemente ale structurii sunt denumite în mod diferit în cadrul unor SGBD - uri: lanț, legătură sau set. Setul este format dintr-o înregistrare principală (owners) și mai multe înregistrări subordonate (members).

În general, cu ajutorul LDD se realizează:

- atribuirea unui identificator pentru fiecare informație sau grup de informații;
- descrierea ierarhică a informațiilor;
- specificarea grupurilor repetitive de date;
- definirea lungimii și naturii fiecărui câmp (numeric, alfa-numeric etc.);
- definirea intervalului de variație posibilă a câmpului;
- definirea parolelor și a autorizațiilor speciale pentru anumite câmpuri.

• **Funcția de manipulare**

Permite efectuarea următoarelor operații: crearea, inserarea, suprimarea sau actualizarea unor înregistrări definite de către utilizator. De asemenea, funcția de manipulare facilitează căutarea, sortarea și editarea totală sau parțială a unor înregistrări virtuale corespunzătoare rezultatului unei întrebări formulate prin intermediul limbajului de manipulare.

Limbajele de manipulare pot fi grupate în două mari categorii:

- *limbaje autonome* – în cadrul cărora comenzile de manipulare reprezintă chiar funcții referitoare la utilizarea datelor (de exemplu utilitarul EXCEL[®], din cadrul pachetului standard Microsoft OFFICE[®], utilizat mai ales pentru crearea bazelor de date);
- *limbaje gazdă* – în care operațiile de manipulare se realizează cu ajutorul limbajelor de nivel înalt. Această variantă de manipulare a bazelor de date este preferată în cadrul aplicațiilor complexe dezvoltate de către autor, ca limbaj avansat folosindu-se mediul Visual C.

• **Funcția de utilizare**

Permite comunicarea între utilizator și baza de date, sub aspectul asigurării acelor mijloace de comunicare care îl avantajează cel mai mult pe utilizator.

Din punctul de vedere al funcției de utilizare, utilizatorii pot fi de mai multe categorii:

- *Utilizatori liberi sau convenționali* – care au la dispoziție limbaje de interogare într-o formă apropiată de vorbirea curentă și formează grupa utilizatorilor așa-zisi "nespecialiști". Aceștia folosesc, de regulă, display-ul, întrebările sunt prestabilite, nu cunosc nici structura, nici modul de lucru cu bazele de date și se rezumă la aplicarea unor funcții sau proceduri puse la dispoziție de către producătorul pachetului software. Este cazul dispecerilor umani care urmăresc funcționarea întregului ansamblu, utilizând facilitățile programului, fără a avea cunoștințe de programare în limbaje avansate, dar având suficiente cunoștințe generale de utilizare a calculatorului.
- *Utilizatorii parametrici* – fac uz, de regulă, de limbaje de manipulare, în special pentru interogare, utilizând proceduri prestabilite. Aceștia sunt inginerii de sistem, fie ai producătorului de software, fie ai beneficiarului, care cunosc atât structura bazelor de date cât și problemele sistemului de

operare, ceea ce le permite să obțină avantaje maxime în comparație cu prelucrarea prestabilită a fișierelor clasice. Aceștia asigură exploatarea eficientă a întregului sistem și chiar optimizarea funcționării lui.

- *Administratorul bazei de date* – acesta este un utilizator special, care răspunde de toate activitățile și operațiile referitoare la baza de date, pe care o gestionează, urmărind inclusiv performanțele acesteia. El definește obiectivele sistemului, ajută la definirea cerințelor utilizatorilor, definește structura virtuală și împarte drepturile de acces ale utilizatorilor, stabilește procedurile de validare a datelor, elaborează concepția de protecție a datelor și evaluează performanțele sistemului. Administratorul răspunde de alegerea și implementarea SGBD - ului, asigură încărcarea bazei de date, definește strategia de lucru și distribuie documentația utilizatorilor. De regulă, administratorul este unul dintre inginerii de sistem ai beneficiarului, (sau chiar un grup), care lucrează în strânsă colaborare cu producătorul aplicației, stabilind toate cerințele, obiectivele, facilitățile și prioritățile aplicației.

4.2. Aspecte tehnice privind structura bazelor de date pentru aplicații în timp real

Bazele de date de tip relațional (tabelar) sunt cele mai utilizate pentru aplicațiile prezentate în această lucrare. Nu vom insista asupra modelului matematic relațional, bazat pe teoria ansamblurilor și relațiilor dintre acestea, ci mai mult asupra modalității de creare și de exploatare eficientă a unor asemenea baze de date, care conferă și caracterul de originalitate al lucrării.

Facilitățile de exploatare și întreținere a relațiilor din cadrul bazei de date se datorează în principal limbajelor specifice acestor tipuri de baze de date. Un limbaj relațional trebuie să asigure definirea datelor, crearea bazei de date, interogarea, actualizarea (întreținerea), protecția și securitatea datelor conținute în bază.

Deși soluția stocării sub forma unor baze de date consacrate (oracle, informix, paradox, format dbf, etc) pare a fi într-o primă instanță soluția ideală, trebuie avută în vedere natura datelor ce urmează a fi stocate. Bazele de date consacrate oferă avantajul portabilității, de cele mai multe ori în detrimentul vitezei (dezavantaj ce poate fi compensat de un hardware puternic dar scump).

Soluția aleasă depinde, desigur, și de existența sau nu a unui server de baze de date în rețeaua beneficiarului. Dacă acest server există (și marea majoritate a aplicațiilor ce rulează în rețea îl folosesc pentru stocarea datelor), este probabil util ca și programul de teleurmărire să stocheze datele pe acest server pentru a fi ușor disponibile unei terțe aplicații care are nevoie de ele (este vorba de regulă de aplicații complexe de analiză a datelor). Existența unui server de baze de date oferă de asemenea soluții complete de întreținere și back-up, eliberând programul de teleurmărire de aceste sarcini. În plus, înregistrarea datelor și regăsirea acestora cade în sarcina SGBD-ului (Sistemul de Gestiune a Bazelor de Date) și nu a programului de teleurmărire, care nu cunoaște structura internă a bazei de date. Un alt avantaj demn de menționat este posibilitatea înregistrării / regăsirii datelor prin comenzi SQL cu efect în independență față de motorul de baze de date folosit și platforma pe care acesta rulează – avantaj major în cazul în care se dorește ca programul să ruleze pe platforme eterogene. Unul din dezavantajele acestei abordări este că datele sunt văzute uniform și nu sunt stocate în formate specifice care să ușureze anumite operații pe care programul de teleurmărire le efectuează. Spre exemplu, dacă programul de teleurmărire

trebuie să reprezinte grafic un set de date, acestea pot fi în prealabil înregistrate sub o formă care să permită execuția acestei funcții foarte rapid și eficient. În cazul unui server de baze de date, datele trebuie mai întâi cerute serverului, operație care poate dura destul de mult, apoi convertite într-o formă ușor de afișat grafic și abia apoi transmise rutinei de afișare.

Dacă viteza este un factor critic, de multe ori soluția utilizării unui server de baze de date este inaplicabilă. În acest caz, datele vor fi stocate în format propriu. Este necesar însă ca programul de teleurmărire să fie însoțit de alte programe sau de funcții care să permită exportul acestor date în formate de baze de date consacrate, cel mai adesea pentru analiza ulterioară a datelor de către alte programe. Un exemplu clasic este exportul datelor în format Excel.

În cazul alegerii soluției stocării datelor în format propriu, toate funcțiile de stocare / regăsire a informațiilor cad în sarcina programului de teleurmărire. Acesta va stoca datele sub forma unui sau a mai multor fișiere. Decizia stocării sub forma unui singur fișier sau a mai multora este dată de dimensiunile fișierelor ce urmează a fi stocate, limitări ale sistemului de operare folosit în lucrul cu fișiere, precum și de algoritmi de regăsire a datelor folosiți (de multe ori o parte din logica de regăsire a informațiilor este implementată în structura arborescentă a sistemului de fișiere).

O particularitate a sistemelor de teleurmărire este că, de regulă, nu avem operațiuni de ștergere la nivel unitar a datelor, ci doar operațiuni de ștergere la nivel global (pe perioade lungi de timp). Acest aspect face ca implementarea sub forma de fișiere, în format propriu, să fie ușoară, programul nefiind silit să implementeze algoritmi complecși de compactare a datelor. O analiză a tipurilor de date care intervin în procesele de teleurmărire ne va da o imagine a modului de înregistrare a datelor în aceste fișiere. Trebuie ținut seama de faptul că aproape toate datele vehiculate sunt furnizate de convertoare cu rezoluții uzuale de 10-16 biți.

Orice conversie a acestei mărimi în alt format (de ex. în număr real) este inevitabil însoțită de o pierdere a preciziei.

De aceea este de dorit ca datele să fie transferate de la sursa la destinație (în acest caz de la traductori), la programul de teleurmărire, fără conversii suplimentare pentru păstrarea preciziei de măsurare. De asemenea, reprezentarea datelor în format **int**,¹ care este format de bază, asigură cea mai mare viteză de procesare, sporind eficiența programului în general.

Vom detalia principalele tipuri de date ce intervin în procesele de măsurare și forma uzuală de reprezentare a lor:

- a) **datele provenite de la convertoare A/D** (numite și date analogice) se reprezintă de regulă în limbajele de programare prin tipul **int**. Deoarece convertoarele uzuale au 10-12 biți (sunt rare cazurile în care se folosesc convertoare pe 16 biți) mai rămân 4 biți (sau mai puțini) în care pot fi stocate diferite informații de stare cum ar fi defect / corect, depășirea anumitor limite etc. Deși aceasta este o practică frecventă, generată de nevoia de economisire a memoriei sau spațiului pe disc în sistemele mai vechi, în cazul sistemelor actuale ce dispun de suficientă memorie și spațiu de stocare este de preferat folosirea unui octet suplimentar care conține informații de stare. Dacă programul

¹ **int**, sau integer este formatul de bază în care se reprezintă datele. Toate operațiunile elementare (cum ar fi comparări, adunări, scăderi, etc.) se execută la nivelul procesorului cu operanzi de acest tip. De aceea, reprezentarea datelor în acest format asigură cea mai mare viteză de prelucrare a datelor. În cazul sistemelor de operare pe 16 biți (cum este de ex. MS-DOS), așa cum o spune și numele, tipul int are o reprezentare internă pe 16 bit. În cazul sistemelor de operare pe 32 biți (cum este de ex. Windows NT), tipul int este reprezentat prin 4 octeți, adică 32 biți.

lucrează pe un SO (sistem de operare) pe 16 biți și se urmărește trecerea ulterioară la SO pe 32 biți, se va avea în vedere utilizarea a 32 biți, adică a tipului **long** pentru reprezentarea datelor, care are aceeași reprezentare internă în cazul ambelor SO, dar care aduce avantajul portabilității și dezavantajul folosirii ineficiente a resurselor în SO pe 16 biți. O altă problemă care trebuie luată în seamă este cea a **alinierii**² structurilor de date, deoarece multe compilatoare folosesc această tehnică pentru optimizări de viteză. Dacă spațiul ocupat pe disc este o resursa critică, atunci este util ca alinierea să fie forțată la nivel de octet (alinieră 1).

- b) **Datele binare** (provenite de la intrări ce pot avea doar 2 stări logice). Deși de multe ori este suficient un singur bit pentru a reprezenta o mărime binară, în practică se folosește reprezentarea pe 2 sau mai mulți biți, pentru a introduce pe lângă valoare (care poate fi reprezentată întotdeauna pe 1 bit) și informații de stare pentru respectiva mărime. Mărimile binare sunt citite de regulă prin operații **in** (input from port) de la porturi de 8 sau 16 biți. Deoarece în multe cazuri există o dualitate în reprezentarea logică a unei mărimi (cel mai adesea valoarea logică „1” se reprezintă fie ca valoare la intrare fie prin semnal lipsă) se va urmări ca semnalele ce se leagă la intrare să respecte regula conform căreia, la un moment dat cel puțin unul dintre ele are valoare opusă față de celelalte. În acest mod se poate detecta absența semnalului de intrare. În fișiere aceste date se vor reprezenta în grupuri de câte 8 biți, urmate de un octet ce codifică starea întregului grup de semnale. O altă soluție, care permite aflarea stării fiecărui semnal în parte și nu a întregului grup presupune folosirea a câte 2 biți pentru fiecare mărime, unul fiind furnizat de valoarea logică a mărimii iar celălalt de negata acesteia. În acest mod se pot detecta situațiile de nefuncționare la nivel de canal cu dezavantajul înjumătățirii nr. de semnale ce pot fi culese (în condițiile folosirii aceluiași echipament).

Valoare logică	Semnificație
(00)	Canal defect (probabil scurt circuit)
(01)	Intrare nivel „0”
(10)	Intrare nivel „1”
(11)	Canal defect (probabil fir întrerupt)

Tabel cu corespondența între (**semnal, semnal negat**) și semnificația acestor valori.

- c) **Datele de altă natură**. Uneori datele nu sunt furnizate direct de către convertoare ci sunt furnizate de dispozitive cu un grade de inteligență mai avansate, care transmit datele de la traductori sau date derivate din acestea prin calcule convertite în valori reale (de regulă float sau double). Decizia

² prin **alinieră** se înțelege procesul de completare a unei structuri de date cu octeți suplimentari, nefolosiți, până ce dimensiunea structurii de date ajunge la multiplu de 2, cu scopul creșterii vitezei de procesare. De remarcat că viteza de procesare crește, deoarece procesorul poate efectua operații elementare cu date de dimensiune 2, 4 sau 8 și este eliminat over-head - ul dat de operațiile de izolare a unei părți dintr-un operand sau rezultat multiplu de 2. Compilatoarele pe 16 biți aliniază implicit structurile la multiplu de 2 iar cele pe 32 biți la multiplu de 4. Să luăm de ex. următoarea structură: { short int x; unsigned char y;} care ar trebui să aibă dimensiunea de 2+1=3 octeți. În cazul unui compilator pe 16 biți această structură va ocupa implicit 4 octeți (deși informația utilă este stocată pe 3) iar în cazul compilatoarelor pe 32 biți, va ocupa 8 octeți. Scrierea acestor structuri pe disc va prezerva dimensiunea generată de compilator, rezultând o pierdere de spațiu de 25%, respectiv 62.5% (pierderea este totuși compensată de o creștere a vitezei de execuție).

convertirii acestor mărimi în alte formate trebuie luată în considerare precizia finală ce trebuie obținută și de metodele de rezolvare a incompatibilităților în reprezentarea datelor. O atenție deosebită trebuie acordată datelor care vin codate în format BCD deoarece acestea pot fi folosite direct în calcule financiare de mare precizie (majoritatea limbajelor de programare oferă suport pentru acest format) și deci pot fi folosite cu succes în operații de facturare / decontare.

- d) **Timpul în sistemele de teleurmărire.** O mărime care intervine oricând în cazul acestor sisteme este timpul, orice mărime având atașată pe lângă valoare și stare și o informație de timp ce descrie momentul la care mărimea a avut respectiva valoare / stare. Operațiunile cu date calendaristice se efectuează prin apelarea unor funcții din bibliotecile de timp, regula generală fiind lucrul cu timp absolut (timp GMT) și afișarea datelor temporale în funcție de informațiile zonale (time-zone). Aceasta asigură programului o independență față de poziția geografică unde va rula și în plus va putea ține seama de toate setările specifice zonei respective. Stocarea timpului în fișiere se face folosind 4 octeți în cazul în care se folosește rezoluția în secunde și respectiv 6 octeți dacă se măsoară în milisecunde. Unele sisteme de operare pot garanta o bază de timp (cum ar fi MS-DOS-ul prin apelarea funcției de ceas la fiecare 55 ms, sau QNX-ul) pe când, în cazul altor sisteme (mai ales în cele multitasking cum ar fi Windows 95/NT) nu mai este garantată o bază de timp constantă. În aceste situații trebuie luate măsuri suplimentare, operațiunile cu timpul făcându-se folosind funcții ce determină timpul mai degrabă decât prin calculul „manual” al timpului din cuante.

În aplicațiile întâlnite în practică există patru familii de limbaje relaționale:

- limbaje bazate pe algebra relațională – care utilizează o serie de operatori algebrici relaționali (permutare, proiecție, restricție, selecție, împărțire, reuniune, intersecție, diferență, concatenare etc.). Limbajul algebric relațional este un limbaj procedural complet, dar dificil pentru necunoscători. El se bazează pe utilizarea unui ansamblu de operatori cu ajutorul cărora se acționează asupra uneia sau mai multor relații din cadrul unei baze de date relaționale. Ca rezultat se obține o nouă relație.
- limbaje care utilizează o serie de simboluri relaționale cu ajutorul cărora se definesc o serie de calcule relaționale. Limbajul bazat pe calculul relațional este un limbaj neprocedural care utilizează printre altele cuantificatori matematici universali, de tipul (\forall) și de existență (\exists), permițând utilizatorilor să obțină ansambluri de elemente de date și diferite relații între aceste elemente. Este totuși destul de greu de utilizat în mod direct în aplicații.
- limbaje orientate pe transformări – constituie o clasă de limbaje neprocedurale care, cu ajutorul relațiilor transformă datele de intrare în ieșiri dorite de către utilizator. Aceste limbaje produc structuri ușor de înțeles și manipulat în termeni practici: ce trebuie obținut, pornind de la ceea ce este cunoscut.
- Limbajele relaționale de tip grafic – ca mod de lucru, utilizatorul completează o serie de răspunsuri, pe un exemplu, prin care sistemul „ghicește” ce trebuie făcut și generează instrucțiuni corespunzătoare ale limbajului.

Cerințele utilizatorului uman al bazei de date sunt cele care dictează modul în care un limbaj oarecare stabilește relații între articolele bazei respective. Prin urmare, funcțiile care operează asupra bazei de date (indiferent de limbajul în care sunt generate) trebuie să dovedească putere de calcul, exactitate, suplețe și adaptabilitate la cerințele utilizatorului.

Sistemele de Gestiune a Bazelor de Date (SGBD) moderne, aplicate în situația monitorizării sistemelor hidrotehnice se adresează unui public larg, de neprogramatori, propunându-le mijloace de lucru simple, eficiente și atractive, în general aplicații „la cheie”. De aceea interfața om – mașină a acestor SGBD trebuie concepută astfel încât să conducă la eficientizarea muncii operatorului uman și la buna funcționare a sistemului, fără situații conflictuale și ambiguități în exploatare.

4.3. Interfața om - mașină a SGBD. Tipuri de ferestre de interfațare

Operatorul uman interacționează cu procesul urmărit prin intermediul unei interfețe grafice și prin intermediul dispozitivelor de intrare (mouse, tastatură). Interfața grafică este puternic dependentă de sistemul de operare folosit și de pachetul software utilizat.

În ultimul timp este unanim acceptată ideea că interfața grafică să ruleze sub sistemul de operare Windows. Aceasta oferă multe avantaje dintre care cel mai important este familiaritatea utilizatorilor cu acest mediu precum și multitudinea de programe existente pentru prelucrarea ulterioară a datelor (cel mai cunoscut exemplu este Microsoft Excel). Acest mediu structurează informațiile cu ajutorul unor ferestre (window) pe ecran, care se comportă standard indiferent de conținutul lor.

Orice fereastră are butoane pentru minimizare / maximizare / restaurare, bară de titlu, margini dimensionabile sau nu etc. În acest context putem defini un număr de tipuri de ferestre care se așteaptă a fi disponibile într-o interfață om-operator modernă precum și funcțiile minime pe care acestea trebuie să le implementeze.

4.3.1. Hărți sinoptice

O schemă sinoptică este o reprezentare schematică a unei instalații tehnologice. Pe această schemă sunt afișate sub formă de text sau grafică valorile mărimilor citite din proces. Reprezentarea în acest mod oferă avantajul unei priviri de ansamblu foarte bune asupra unei instalații. Afișarea grafică a valorilor este dependentă de valoarea mărimii, depășirea limitelor de alarmare duce la afișarea cu alte culori (ex: galben = alarmare preventivă, roșu = alarmare generală) permițând operatorului uman să detecteze dintr-o singură privire situațiile de funcționare anormale.

Un caz particular de schemă sinoptică îl reprezintă afișarea schemei generale a sistemului pe care este figurată starea de funcționare a tuturor dispozitivelor și interfețelor de comunicație implicate în proces.

Orice reprezentare sub formă de schemă sinoptică este posibilă și numărul acestor scheme nu trebuie să fie limitativ. De menționat că după procesul de configurare inițială sistemul va afișa doar schemele definite, adăugarea sau ștergerea unor scheme necesitând reconfigurare. Sistemul trebuie de asemenea să implementeze un mecanism de navigare facilă prin aceste scheme. Dintre modelele de navigare, cele mai întâlnite sunt alegerea schemei dintr-o listă sau definirea unor zone „active” din schemă pe care dacă se face clic se ajunge în altă schemă sinoptică.

Ultimul mod mulează modul natural de navigare de la general la particular (întâi am o privire de ansamblu și apoi merg în amănunt). În fig. 4.3 este prezentat un exemplu de fereastră tip hartă sinoptică.

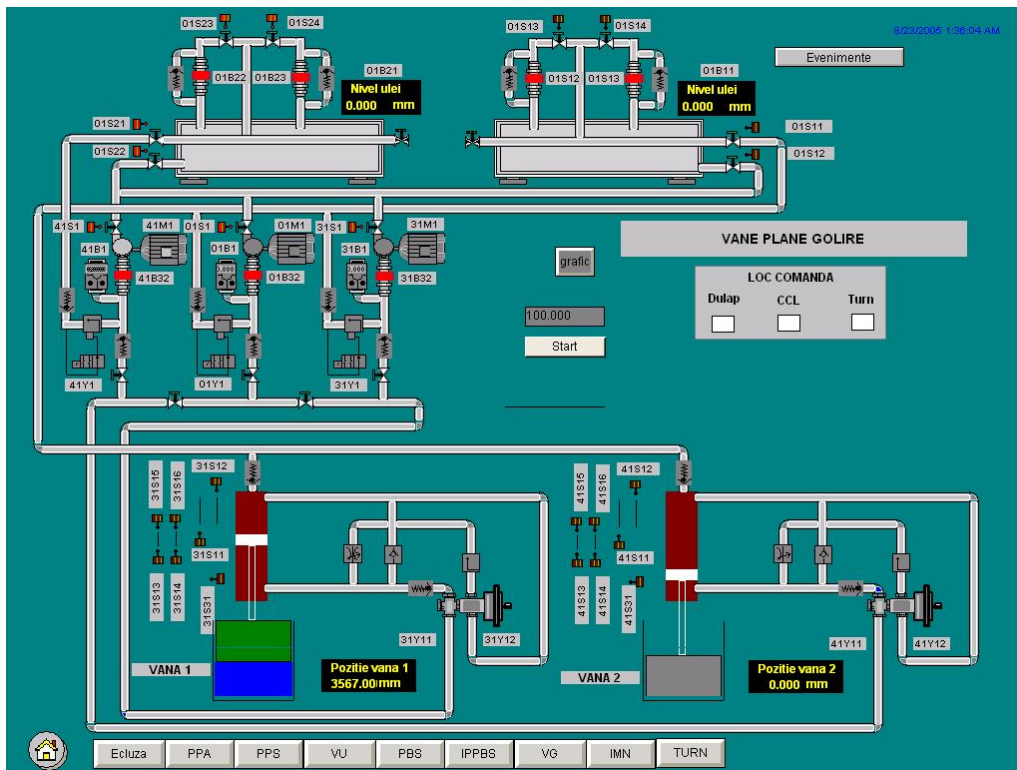


Fig. 4.3. Hartă sinoptică a mecanismului de vane aferent ecluzei.

Deși afișarea valorilor instantanee ale unei mărimi pe scheme sinoptice ne oferă toate informațiile despre acea mărime, uneori este necesar ca un grup de mărimi să fie urmărite simultan. În acest caz, dacă mărimile trebuie corelate între ele, de exemplu menținute la aceeași valoare, urmărirea lor sub formă numerică pe scheme este un proces dificil. De aceea, ele pot fi grupate și reprezentate în aceeași fereastră sub formă de *bargrafuri*. Și cum o reprezentare grafică valorează uneori mai mult decât 1000 de numere, și în acest caz reprezentarea în această formă face procesul de urmărire foarte ușor.

4.3.2. Ferestre cu grafice de evoluție în timp

Valorile instantanee, afișate pe schemele sinoptice dau informații despre valoarea mărimii în momentul ultimei citiri a acesteia. De multe ori este util ca acestei mărimi să i se traseze un grafic de evoluție, care să dea informații despre evoluția sa într-un anumit interval de timp.

Facilitățile ferestrelor de afișare grafică pot varia de la ferestre de afișare simple, neinteractive, la reprezentări complicate capabile de autoscalare și interacțiune dinamică cu utilizatorul, dotate cu funcții de localizare pentru minim / maxim etc.

Graficele sunt de 2 tipuri: real time, care permite reprezentarea grafică a evoluției mărimii pe o perioadă de timp în jurul momentului actual (de obicei ziua sau ora curentă) și care au o singură mărime afișată și grafice istorice care permit

reprezentarea unui număr mai mare de mărimi (pentru comparare, etc.) și pe orice perioadă. Graficele real time folosesc de regulă la urmărirea tendințelor pe termen scurt și pot include un număr mare de puncte precum și detalierea până la nivel de secundă a evoluției datelor.

Datele istorice pe de altă parte pot reprezenta perioade mai mari de timp și uneori folosesc valori medii pentru aceste reprezentări. Atenție deosebită trebuie acordată în cazul în care pe același grafic se reprezintă mărimi care nu au aceeași scală (sau chiar unități de măsură diferite) caz în care se va folosi o reprezentare procentuală pe grafic.

De asemenea, o atenție deosebită trebuie acordată valorii lipsă (sau stare defect), când o mărime nu trebuie afișată (valoarea ei lipsește pe respectiva perioadă sau senzorul a fost defect). Baza de date trebuie să fie înzestrată cu posibilitatea înregistrării valorii lipsă (valoare NULL în terminologia de specialitate) sau să dețină un câmp suplimentar cu starea mărimii.

Vor fi prezentate două categorii de grafice privind evoluția în timp a principalilor parametri de funcționare a echipamentelor hidromecanice aferente ecluzei.

O primă categorie se referă la variația unor parametri funcție de timp sau funcție de alți parametri, în cadrul unei manevre de ecluzare.

A doua categorie de grafice se referă la tendința pe care o are evoluția anumitor parametri de-a lungul mai multor ecluzări pentru o anumită perioadă de timp.

Programul de monitorizare va oferi spre vizualizare și tipărire la imprimantă următoarele grafice:

Poartă plană

- $P=f(s)$ – presiune circuit hidraulic funcție de poziție pentru ambele servomotoare în două culori.
- $\Delta S=f(s)$ – desincronizare funcție de poziție considerându-se valoarea pozitivă a lui ΔS când malul drept este mai sus ca cel stâng ($\Delta S=S_{dr}-S_{stg}$).
Graficul va avea două drepte de culoare roșie paralele cu abscisa, corespunzătoare valorilor limită ale desincronizării $\pm\Delta S_{max}$.
- $V=f(t)$ – viteza de deplasare funcție de timp.

Vane galerii

- $P=f(s)$ – presiune circuit hidraulic funcție de poziție.

Poartă buscată

- $P=f(s)$ – presiune circuit hidraulic funcție de poziție pentru ambele servomotoare în două culori.
- $V=f(t)$ – viteza de deplasare funcție de timp pentru ambele canate în două culori.

Graficul oferă legea de mișcare a porții cu posibilitatea de urmărire a funcționării porții la trecerea dintr-o treaptă de viteză în alta precum și în punctele de așteptare pentru sincronizarea închiderii.

Nivel apă SAS ecluză

- $H=f(t)$ – evoluția în timp a nivelului de apă în ecluză.
Acest grafic va avea două drepte orizontale reprezentând nivele amonte și aval de ecluză între care va evolua curba nivelului din SAS. Acest grafic interpretat împreună cu graficele $P=f(s)$ ale VG și PB pot furniza date cu privire la acționarea echipamentului funcție de încărcare. Din categoria graficelor privind tendințele de evoluție a unor parametri amintim:
 - $H_{am}=f(t)$ – evoluție nivel amonte;
 - $H_{av}=f(t)$ – evoluție nivel aval;
 - $P_{max}=f(t)$ – tendință presiune max. pe parcursul mai multor manevre la fiecare echipament în parte.
 - $\Delta S_{max}=f(t)$ – tendință desincronizare maximă la poarta plană.
 - $A=f(t)$ – tendința numărului de avarii înregistrate într-o perioadă de timp.

În fig. 4.4 prezentăm o astfel de fereastră grafică.

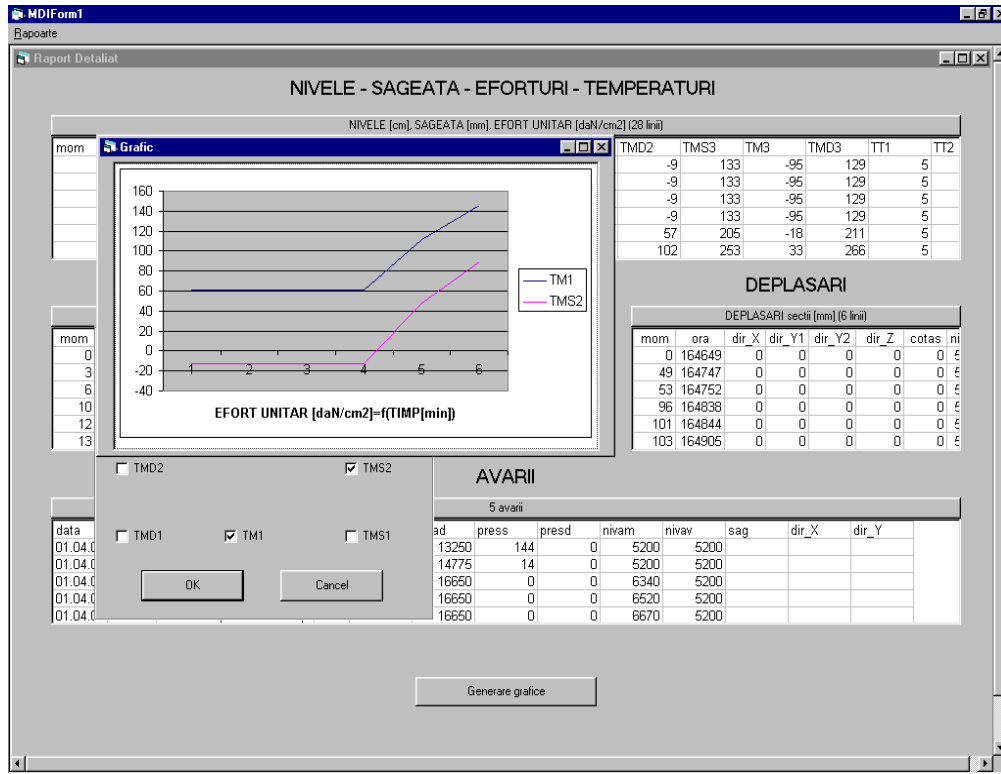


Fig. 4.4. Grafic de evoluție în timp.

4.3.3. Ferestre de evenimente și ferestre de raportare

De o mare importanță în urmărirea unui proces tehnologic se dovedesc a fi evenimentele. Deși mai puțin severe decât alarmele, evenimentele înregistrează orice modificare survenită în procesul normal de funcționare a unei instalații. După afișarea listei de evenimente navigarea se va face cu ajutorul săgeților și a tastelor funcționale.

Informațiile stocate în acest loc sunt de tip echipament *pornit / oprit / defect* precum și schimbările de stare. Fiecare astfel de eveniment este înregistrat împreună cu momentul de timp la care s-a produs.

Pentru urmărirea ușoară a evenimentelor se utilizează următorul cod al culorilor.

Un posibil model de cod al culorilor este următorul:

- **GALBEN** eveniment de tip informativ, precizând că un echipament funcționează OK;
- **NEGRU** eveniment ce precizează starea de nefuncționare a unui echipament (oprit / defect);
- **VERDE** un element binar se găsește în starea normală (de ex. un bec normal stins);
- **ROSU** un element binar se găsește în starea de funcționare opusă stării normale.

Una dintre cele mai folositoare funcții a aplicațiilor SCADA este cea de generare a rapoartelor. În cadrul acestor rapoarte se obțin date sintetice, în formă tabelară. În general, dar nu obligatoriu, rapoartele sunt zilnice (care conțin date orare) și lunare (care conțin date zilnice).

Valorile conținute de către aceste rapoarte sunt furnizate de calcule specifice, definite în cadrul fișierelor de configurare. Cele mai uzuale sunt minimul / maximumul valorii unei mărimi, media orară / zilnică, indexul la o anumită dată sau diferența de valoare pe o perioadă de timp.

Pe baza datelor stocate în baza de date să se ofere periodic (zilnic, săptămânal, lunar etc.) o serie de rapoarte informative privind desfășurarea activității de exploatare a ecluzei.

Aceste rapoarte sunt oferite de către analizele ulterioare ale datelor stocate, analize efectuate pe baza algoritmilor cuprinși în pachetul de programe off – line ale programului de monitorizare. Soft-ul of – line oferă posibilitatea vizualizării și/sau tipăririi la imprimantă a acestor rapoarte. Ele oferă o serie de avantaje, care conduc la îmbunătățirea activității de exploatare, dintre care enumerăm:

- Depistarea rapidă a cauzelor care au dus la apariția stărilor de defect ale echipamentelor precum și a sistemelor de conducere a lor.
- Intervenție rapidă pentru înlăturarea stărilor de defect și repunerea în funcțiune a instalațiilor.
- Oferă informații utile pentru întocmirea programelor de revizii și reparații.
- Asigură o creștere a disciplinei de manevră din partea operatorilor ecluzei.

Programul de monitorizare va realiza vizualizarea și/sau tipărirea la imprimantă a unor situații cu privire atât la succesiunea desfășurării operațiilor cât și la evenimentele apărute pe parcursul procesului de ecluzare. Aceste situații vor fi editate sub forma unor formulare pe care le prezentăm în continuare:

a) MANEVRE

Pentru fiecare echipament se va completa un formular care cuprinde o listă cu toate manevrele efectuate.

Pe formular se va specifica:

- denumire echipament;
- anul;
- perioada în care s-au efectuat manevrele.

Pentru fiecare manevră se vor nota următoarele:

- ziua, luna, an;
- start oră, minut, sec;
- stop oră, minut, sec;
- perioada de citire (m sec);
- nr. citire;
- nr. de avarii;
- rezoluție poziție;
- rezoluție presiune;
- rezoluție nivele;
- rezoluție temperatură.

În cazul funcționării ecluzei în regim automat pe formatul MANEVRE vor fi înregistrate succesiv manevre efectuate de către fiecare echipament în parte în cadrul unui ciclu de ecluzare (ridicare, respectiv coborâre). Se va nota suplimentar în acest caz start (oră, minut, sec.) pentru fiecare inițializare a ciclului de ecluzare.

b) AVARII

Pentru ecluză în ansamblu sau pentru fiecare echipament în parte se va completa un formular care va conține următoarele date:

- anul;
- perioada în care s-au produs avariile;
- data (zi, lună,an);
- ora, minut, sec;
- denumire avarie;
- poziție servomotor stâng;
- poziție servomotor drept;
- presiune stânga;
- presiune dreapta;
- nivele amonte, SAS, aval;
- temperatură;

Așa după cum s-a mai arătat parametrii care însoțesc avariile în acest tabel vor fi notate cu valorile atinse în momentul apariției avariei, neținându-se cont de rezoluție impuse în vederea reducerii volumului de date stocate.

c) COTE, PRESIUNI, NIVELE

Pentru fiecare echipament în parte se va realiza un formular cu denumirea de mai sus, în care se vor completa valorile parametrilor la fiecare citire, dacă aceste valori diferă de la o citire la alta cu mai mult decât rezoluția impusă.

Tabelul va avea notate următoarele:

- denumire echipament;
- data (zi, lună, an);
- ora, minut, sec;
- cota stângă;
- cota dreaptă;
- presiune stânga;
- presiune dreapta;
- nivele amonte (de echipament);
- nivel aval (de echipament);
- încărcarea (diferența de nivel).

De asemenea vor fi înregistrate și rezoluțiile cu care au fost notați parametrii.

d) STĂRI ECLUZĂ

Pentru semnalele discrete culese din instalație și care definesc o anumită stare a ecluzei se vor nota principalele stări ale ecluzei pe care le definim mai jos:

- | | |
|----------|---|
| Starea 1 | - POARTA PLANĂ deschisă; |
| | ▪ Vana amonte închisă, are condiții; |
| | ▪ IPPBS coborâtă, are condiții; |
| | ▪ POARTA BUSCATĂ, închisă, fără condiții; |
| | ▪ Vana aval, închisă, fără condiții; |
| | ▪ Nivel amonte, egalizat; |
| | ▪ Nivel aval, neegalizat; |
| Starea 2 | - POARTA PLANĂ se închide; |
| | ▪ Vana amonte închisă, are condiții; |
| | ▪ IPPBS coborâtă, are condiții; |
| | ▪ POARTA BUSCATĂ, închisă, fără condiții; |
| | ▪ Vana aval închisă, fără condiții; |
| | ▪ Nivel amonte, egalizat; |
| | ▪ Nivel aval, neegalizat; |

Starea 3	<ul style="list-style-type: none"> - POARTA PLANĂ închisă; <ul style="list-style-type: none"> ▪ Vana amonte închisă, are condiții; ▪ IPPBS coborâtă, are condiții; ▪ POARTA BUSCATĂ, închisă, fără condiții; ▪ Vana aval închisă, are condiții; ▪ Nivel amonte, egalizat; ▪ Nivel aval, neegalizat;
Starea 4	<ul style="list-style-type: none"> - POARTA PLANĂ închisă, fără condiții; <ul style="list-style-type: none"> ▪ Vana amonte închisă, fără condiții; ▪ IPPBS se ridică; ▪ POARTA BUSCATĂ, închisă, fără condiții; ▪ Vana aval închisă, are condiții; ▪ Nivel amonte egalizat; ▪ Nivel aval neegalizat;
Starea 5	<ul style="list-style-type: none"> - POARTA PLANĂ închisă, fără condiții; <ul style="list-style-type: none"> ▪ Vana amonte închisă, fără condiții; ▪ IPPBS ridicată, are condiții; ▪ POARTA BUSCATĂ, închisă, fără condiții; ▪ Vana aval se deschide; ▪ Nivel amonte neegalizat; ▪ Nivel aval neegalizat;
Starea 6	<ul style="list-style-type: none"> - POARTA PLANĂ închisă, fără condiții; <ul style="list-style-type: none"> ▪ Vană amonte închisă, fără condiții; ▪ IPPBS ridicată, are condiții; ▪ POARTA BUSCATĂ închisă, fără condiții; ▪ Vana aval deschisă, are condiții; ▪ Nivel amonte neegalizat; ▪ Nivel aval neegalizat;
Starea 7	<ul style="list-style-type: none"> - POARTA PLANĂ închisă, fără condiții; <ul style="list-style-type: none"> ▪ Vană amonte închisă, fără condiții; ▪ IPPBS ridicată, are condiții; ▪ POARTA BUSCATĂ, închisă, are condiții; ▪ Vana aval deschisă, are condiții; ▪ Nivel amonte neegalizat; ▪ Nivel aval egalizat;
Starea 8	<ul style="list-style-type: none"> - POARTA PLANĂ închisă, fără condiții; <ul style="list-style-type: none"> ▪ Vana amonte închisă, fără condiții; ▪ IPPBS ridicată, are condiții; ▪ POARTA BUSCATĂ închisă, are condiții; ▪ Vana aval se închide; ▪ Nivel amonte neegalizat; ▪ Nivel aval egalizat;
Starea 9	<ul style="list-style-type: none"> - POARTA PLANĂ închisă, fără condiții; <ul style="list-style-type: none"> ▪ Vana amonte închisă, fără condiții; ▪ IPPBS ridicată, are condiții; ▪ POARTA BUSCATĂ închisă, are condiții; ▪ Vana aval închisă, are condiții; ▪ Nivel amonte neegalizat; ▪ Nivel aval egalizat;

Starea 10	<ul style="list-style-type: none">- POARTA PLANĂ închisă, fără condiții;▪ Vana amonte închisă, fără condiții;▪ IPPBS ridicată, are condiții;▪ POARTA BUSCATĂ se deschide;▪ Vana aval închisă, are condiții;▪ Nivel amonte neegalizat;▪ Nivel aval egalizat;
Starea 11	<ul style="list-style-type: none">- POARTA PLANĂ închisă, fără condiții;▪ Vana amonte închisă, fără condiții;▪ IPPBS ridicată, are condiții;▪ POARTA BUSCATĂ deschisă, are condiții;▪ Vana aval închisă, are condiții;▪ Nivel amonte neegalizat;▪ Nivel aval egalizat;
Starea 12	<ul style="list-style-type: none">- POARTA PLANĂ închisă, fără condiții;▪ Vana amonte închisă fără condiții;▪ IPPBS ridicată, are condiții;▪ POARTA BUSCATĂ se închide;▪ Vana aval închisă, are condiții;▪ Nivel amonte neegalizat;▪ Nivel aval egalizat;
Starea 13	<ul style="list-style-type: none">- POARTA PLANĂ închisă, fără condiții;▪ Vana amonte închisă, are condiții;▪ IPPBS ridicată, fără condiții;▪ POARTA BUSCATĂ închisă, are condiții;▪ Vana aval închisă, are condiții;▪ Nivel amonte neegalizat;▪ Nivel aval egalizat;
Starea 14	<ul style="list-style-type: none">- POARTA PLANĂ închisă, fără condiții;▪ Vana amonte se deschide;▪ IPPBS ridicată, fără condiții;▪ POARTA BUSCATĂ închisă, fără condiții;▪ Vana aval închisă, fără condiții;▪ Nivel amonte neegalizat;▪ Nivel aval neegalizat;
Starea 15	<ul style="list-style-type: none">- POARTA PLANĂ închisă, fără condiții;▪ Vana amonte deschisă, are condiții;▪ IPPBS ridicată, fără condiții;▪ POARTA BUSCATĂ închisă, fără condiții;▪ Vana aval închisă, fără condiții;▪ Nivel amonte neegalizat;▪ Nivel aval neegalizat;
Starea 16	<ul style="list-style-type: none">- POARTA PLANĂ închisă, are condiții;▪ Vana amonte deschisă, are condiții;▪ IPPBS ridicată, fără condiții;▪ POARTA BUSCATĂ închisă, fără condiții;▪ Vana aval închisă, fără condiții;▪ Nivel amonte egalizat;▪ Nivel aval neegalizat;

- Starea 17
 - POARTA PLANĂ închisă, are condiții;
 - Vana amonte se închide;
 - IPPBS ridicată, fără condiții;
 - POARTA BUSCATĂ închisă, fără condiții;
 - Vana aval închisă, fără condiții;
 - Nivel amonte egalizat;
 - Nivel aval neegalizat;
- Starea 18
 - POARTA PLANĂ închisă, are condiții;
 - Vana amonte închisă, are condiții;
 - IPPBS ridicată, are condiții;
 - POARTA BUSCATĂ închisă, fără condiții;
 - Vana aval închisă, fără condiții;
 - Nivel amonte egalizat;
 - Nivel aval neegalizat;
- Starea 19
 - POARTA PLANĂ închisă, are condiții;
 - Vana amonte închisă, are condiții;
 - IPPBS se coboară;
 - POARTA BUSCATĂ închisă, fără condiții;
 - Vana aval închisă, fără condiții;
 - Nivel amonte egalizat;
 - Nivel aval neegalizat;
- Starea 20
 - POARTA PLANĂ se deschide;
 - Vana amonte închisă, are condiții;
 - IPPBS coborâtă, are condiții;
 - POARTA BUSCATĂ închisă, fără condiții;
 - Vana aval închisă, fără condiții;
 - Nivel amonte egalizat;
 - Nivel aval neegalizat;

Starea 21=Starea 1

În fig. 4.5. este prezentată o fereastră de raportare a manevrelor și evenimentelor aferente unor zile de lucru. Se observă că toate evenimentele sunt colorate în galben, ceea ce corespunde unor situații normale de funcționare.

The screenshot shows a software window with two data tables. The top table, titled 'Manevre', has columns: start_zi, start_ora, stop_zi, stop_ora, per_ci, nr_ci, nr_avarii, senr1_ca_in, senr2_ca_in. The bottom table, titled 'Avars', has columns: data, ora, avans, mom, cubet, colid, press, presd, nivam, nivav, sag, dir_X.

start_zi	start_ora	stop_zi	stop_ora	per_ci	nr_ci	nr_avarii	senr1_ca_in	senr2_ca_in
09.03.03	13:29:24	09.03.03	13:28:36	1000	14	0	ca.in.4,7,8,9,10,11	ca.in.nu
01.04.03	16:46:49	01.04.03	16:46:50	1000	107	5	ca.in.4,7,8,9,10,11	ca.in.nu

data	ora	avans	mom	cubet	colid	press	presd	nivam	nivav	sag	dir_X
01.04.03	16:47:47	Desinc. dr. sus 1 (50mm)	49	13200	13250	144	0	5200	5200		
01.04.03	16:47:52	Desinc. dr. sus 2 (70mm)	53	14700	14775	14	0	5200	5200		
01.04.03	16:48:36	TMS3 lim.1 (1109daN/cm2)	96	16650	16650	0	0	6340	5200		
01.04.03	16:48:44	TMS3 lim.1 (1109daN/cm2)	101	16650	16650	0	0	6520	5200		
01.04.03	16:48:49	TMS3 lim.1 (1107daN/cm2)	106	16650	16650	0	0	6670	5200		

Fig. 4.5. Fereastră de evenimente și raportare.

4.3.4. Fereastra de alarmare

În cazul urmării unor procese complexe și cu grad înalt de periculozitate, apariția unei funcționări anormale în sistem trebuie semnalată prompt și prin toate mijloacele posibile. Trebuie spus că nu toate mărimile furnizează semnale de alarmă ci numai cele configurate în acest sens.

Recepționarea unor informații despre o stare anormală de funcționare a unei mărimi generatoare de alarmă este semnalată prin apariția în prim plan a unei ferestre în care este specificat tipul erorii apărute.

ceastă fereastră nu poate fi închisă și nu dispăre decât în cazul revenirii la normal a mărimii sau a acceptării sale de către operator. Dacă sistemul dispune de o placă de sunet, pe lângă fereastra de atenționare este posibilă și emiterea unui semnal sonor de alarmă.

Deși alarmele sunt foarte utile în situații critice, uneori simpla afișare a valorii mărimii cu altă culoare (roșu) este un semnal de alarmă suficient. În aceste cazuri alarmele se vor dezactiva deoarece pot stânjeni procesul de urmărire a parametrilor prin modul lor ireversibil de apariție / dispariție.

De regulă, alarmele pot fi clasificate pe grupe și pot fi asociate priorități anumitor grupe sau alarme individuale iar utilizatorul poate alege să inhibe automat alarmele care au prioritate mică.

O altă funcție a sistemului de monitorizare o constituie declanșarea alarmelor la depășirea valorilor prescrise ale parametrilor urmăriți. În această categorie intră și semnalele preventive care nu se constituie în avarii și care trebuie să atragă atenția operatorului asupra apropierii parametrilor de valorile limită.

Aceste mesaje vor fi însoțite de un semnal pâlpâitor de culoare roșie pe corpul echipamentului invalid și de un semnal sonor intermitent.

De asemenea se vor afișa unele informații care să ajute operatorul să determine cauza care a dus la apariția avariei cum ar fi:

- desincronizare maximă poartă plană (valoarea desincronizării);
- timp manevră depășit (timpul scurs);
- presiune maximă în circuitul hidraulic (valoarea presiune atinsă);
- tasare de avarie vană;
- starea ecluzei la apariția avariei.

Mesajele preventive se vor afișa pe fond de culoare galbenă însoțite de semnal luminos pâlpâitor și de către un semnal sonor de altă natură ca cel de avarie. Mesajele vor fi memorate într-un istoric al avariilor împreună cu momentul apariției lor.

Informațiile din proces care pot fi trimise PC-urile Industriale vor fi înmagazinate de către acesta și vor fi eficient protejate împotriva unor acțiuni perturbatoare din partea unor operatori rău intenționați.

Valorile preluate de către PC-urile Industriale vor fi prelucrate după diverși algoritmi urmând ca rezultatele acestor prelucrări să fie înmagazinate în baza de date în vederea utilizării ulterioare pentru analiza funcționării instalației (grafice, rapoarte etc.).

Algoritmii de prelucrare a datelor achiziționate în vederea memorării lor diferă de la un tip de mărime la altul după cum urmează:

a) mărimi cu variație continuă cum ar fi:

- poziție echipament
- nivele apă
- presiune circuit hidraulic
- nivelul uleiului în rezervor
- temperatură
- direcție, viteză vânt.

Pentru acest tip de mărimi, setul de date ce sunt citite la un moment dat vor fi memorate numai dacă cel puțin unul din ele diferă față de valoarea anterioară cu cel puțin o valoare ce reprezintă rezoluția cu care se dorește stocarea evoluției în timp a unui parametru. Programul va oferi posibilitatea modificării acestor rezoluții în vederea optimizării raportului între acuratețea analizării procesului de manevrare a echipamentelor hidromecanice și volumul necesar de date stocate.

b) Semnalele discrete culese din instalațiile de automatizare cum ar fi pentru fiecare echipament în parte:

- echipament valid
- poziție închis
- poziție deschis
- condiții de închidere
- condiții de deschidere
- se închide
- se deschide.

Pentru acest tip de mărimi rezultă un vector de stare al ecluzei alcătuit din n biți, vector ce va fi memorat împreună cu momentul citirii numai dacă diferă de vectorul citit anterior prin cel puțin un bit.

La depășirea limitelor de funcționare a parametrilor, programul va declanșa semnalele de alarmă.

Acest tip de semnale vor fi memorate într-un istoric al alarmelor împreună cu momentul apariției lor și cu toți ceilalți parametri citiți.

În fig. 4.6. prezentăm o astfel de fereastră de alarmare.

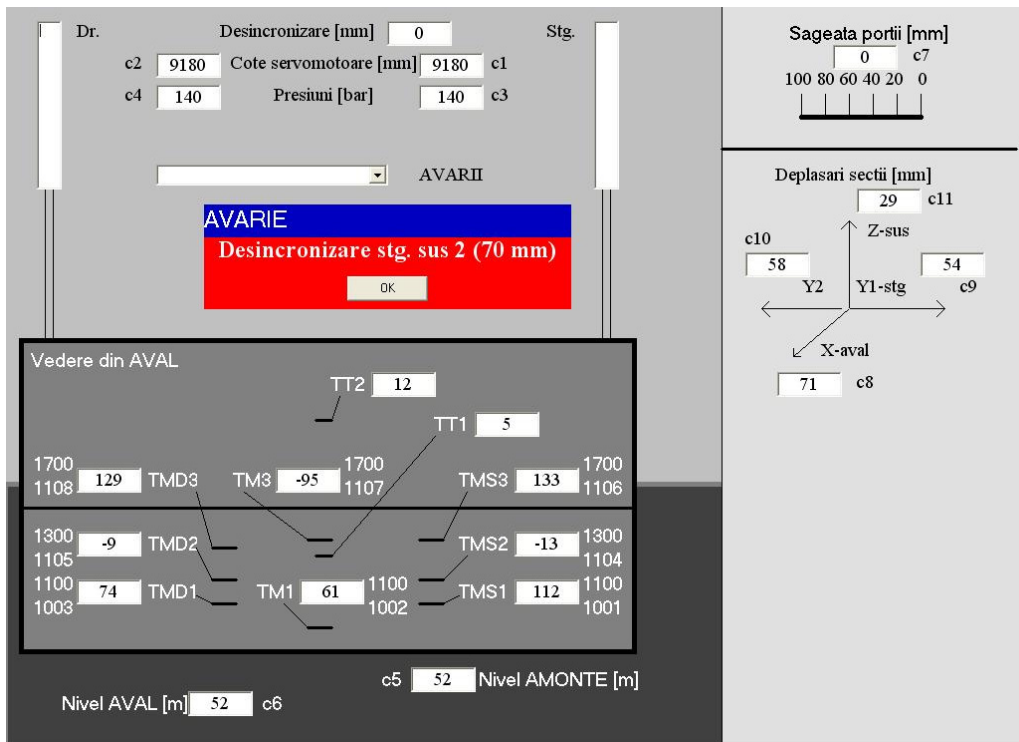


Fig. 4.6. Fereastră de alarmare.

4.4. Posibilități de modernizare a bazei de date

Baza de date care poate fi folosită este Microsoft SQL Server 2005.

Aplicațiile din pachetul Iconics Genesis32 v9, descris în capitolele anterioare, folosesc servere OPC. Pachetul Iconics Genesis32 include tehnologia „OPC to the core” adică este construit de la început pe baza tehnologiei OPC și include/implementează nativ toate specificațiile acestui standard de comunicație.

Mecanismul de reactualizare a bazei de date

Reactualizarea bazei de date este făcută în principal de următoarele componente:

- Servere OPC – reactualizează baza de date cu valori momentane la fiecare interval de eșantionare;
- AlarmWorX – reactualizează baza de date cu alarme curente ori de câte ori apare o alarmă nouă/eveniment în sistem;
- AlarmWorX Datalogger - reactualizează baza de date cu alarme istorice ori de câte ori o alarmă/eveniment dispăre din sistem, ea fiind înregistrată în istoricul de alarmare;
- TrendWorX Datalogger – colectează datele de la serverele OPC și arhivează măsurile în sistem cu frecvența precizată în etapa de configurare.

În vederea optimizării accesului la date, sistemul Iconics Genesis32 păstrează o parte din date în memorie (bufferare) iar altă parte este stocată persistent pe disc; în etapa de configurare se aleg parametrii care asigură utilizarea optimă a acestor tehnologii.

Mecanismul de interogare periodică și la cerere a EAD-urilor.

Comunicația cu EAD-urile de pe nivelele inferioare se realizează prin intermediul serverelor OPC. Atunci când este nevoie de date (de exemplu pentru a le afișa sau pentru a le înregistra în baza de date) clienții de tip OPC (cum este interfața de afișare grafică Iconics) adresează cererea serverelor de tip OPC. Acestea din urmă interoghează echipamentele de tip EAD și livrează datele citite clienților OPC.

Există două tipuri de cereri de la un client OPC către un server OPC: citirea de date din cache și citirea de date direct de la dispozitivul fizic, în cazul de față EAD. Citirea din cache este mult mai rapidă și permite eficientizarea comunicației prin achiziția datelor la anumite intervale de timp (programabile de către utilizator). Citirea direct de la dispozitiv durează mai mult dar aduce ultimele date disponibile. Pachetul software Iconics Genesis32 v9 permite utilizarea ambelor metode de culegere a datelor, folosirea uneia sau alteia dintre ele se face în procesul de configurare.

Interogarea efectivă a dispozitivelor de tip EAD depinde de modul în care a fost implementat intern serverul OPC. În timp ce multe din serverele OPC permit citirea în paralel de la mai multe dispozitive de tip EAD altele permit doar citirea secvențială a lor. Serverele OPC permit citirea în paralel de grupuri de EAD-uri (cea mai eficientă metodă, care asigură echilibrul între viteza de citire și încărcarea procesorului de pe serverul de comunicație).

Mecanismul de salvare și de refacere a bazei de date

Funcțiile pentru salvarea și refacerea bazei de date (backup/restore) sunt incluse în pachetul de administrare a bazei de date Microsoft SQL Server 2005. Schema de salvare, stabilită de comun acord cu beneficiarul, este de obicei formată

dintr-o salvare totală, o dată pe săptămână a bazei de date pe suport extern (DVD) și salvări incrementale zilnice.

Baza de date de tip Microsoft SQL Server permite accesul la date prin toate tehnologiile existente, inclusiv ODBC și OLEDB și DDE. Datele pot fi exportate în fișiere text (ASCII, CSV, Tab separated) precum și în alte baze de date relaționale prin ODBC, inclusiv în Access și Excel.

În pachetul Iconics Genesis32 structura grafică a unui ecran (echivalent noțiunii de imagine) este memorată sub forma unui fișier pe disc. Astfel, modul de organizare în ceea ce privește memorarea acestor imagini este modul de organizare pe disc sub formă de foldere și fișiere (este un model ierarhic, similar modului obișnuit de organizare a fișierelor). În ceea ce privește afișarea pe ecran o astfel de imagine se afișează într-o fereastră de tip windows, poziția și numărul acestor ferestre fiind stabilit în etapa de configurare a sistemului.

Nu există practic nici o limitare privind numărul de imagini ce se pot defini în sistem, alta decât spațiul fizic de pe hard disc.

Nu există nici o limitare software privind numărul de mărimi și simboluri dintr-o imagine.

Adăugarea unei mărimi pe imagine se face online. Timpul de reactualizare a unei imagini depinde de complexitatea imaginii care trebuie să fie afișată și de performanța hardware a calculatorului pe care rulează și este, în general, sub 1.5 secunde.

Sistemul Iconics Genesis32 permite rularea în orice rezoluție grafică (scalarea se face automat). În acest fel nu vor fi necesare imagini diferite pentru rezoluții diferite, reducând timpul de dezvoltare. Modificarea unei imagini se poate face on-line și nu necesită conversie, recompilare și/sau restartare. Se utilizează simboluri color, 3D, pâlpâitoare și animate.

Sistemul permite înglobarea de obiecte de tip ActiveX.

Editorul de imagini este o aplicație tipică Windows, similară cu Microsoft Office și poate fi ușor de folosit. Pentru crearea imaginilor de fundal, editorul dispune de o bibliotecă bogată de simboluri grafice, organizate logic în grupe cu proprietăți comune (de ex. simboluri pentru apă, termoficare, electricitate, etc). Simbolurile pot fi create și de către utilizator folosind elemente de bază și apoi salvate în bibliotecă pentru a fi folosite ulterior sau pot fi importate din alte programe (sunt suportate toate formatele grafice răspândite).

Fiecare simbol de pe ecran poate fi animat (deplasare/ redimensionare/ rotire/ ascundere/colorare/flash) prin definirea unei funcții de animație care are în componență tag-uri din sistem. În acest fel pot fi de ex. animate pompele, pus flash pentru simboluri de alarmă, colorate mărimile funcție de limitele de alarmare, etc. Mărimile simple (valori analogice, formule) sunt tratate la fel ca simbolurile acceptând același tip de comportament dinamic. În scheme pot fi incluse de asemenea obiecte de tip ActiveX din alte aplicații sau specifice Iconics. Obiectele pot fi organizate pe layere, salvarea unei imagini se face în format propriu sub formă de fișier gdf.

Curbele de evoluție sunt gestionate de componenta TrendWorX din pachetul Iconics Genesis32. Curbele de evoluție sunt de doua tipuri, în timp real și curbe istorice. Pe curbele în timp real se poate reprezenta orice semnal din sistem (sau o funcție bazată pe semnalele din sistem). Pe curbele istorice se poate reprezenta orice semnal memorat în baza de date cu istorice (sau o funcție de astfel de semnale). Se pot crea grafice de evoluție în timp real pentru orice mărime, iar graficele istorice se pot trasa pentru toate mărimile care au fost configurate pentru arhivare. Arhiva se poate vizualiza folosind graficele. Parametrii unei curbe (incluzând rezoluție, domeniu, axe, numărul și parametrii mărimilor de reprezentat) se definesc în procesul de configurare. Nu există limitări pentru numărul de grafice reprezentate pe ecran și numărul de curbe de pe un grafic, altele decât limitele ecranului (altfel reprezentarea ar fi prea „înghesuită”).

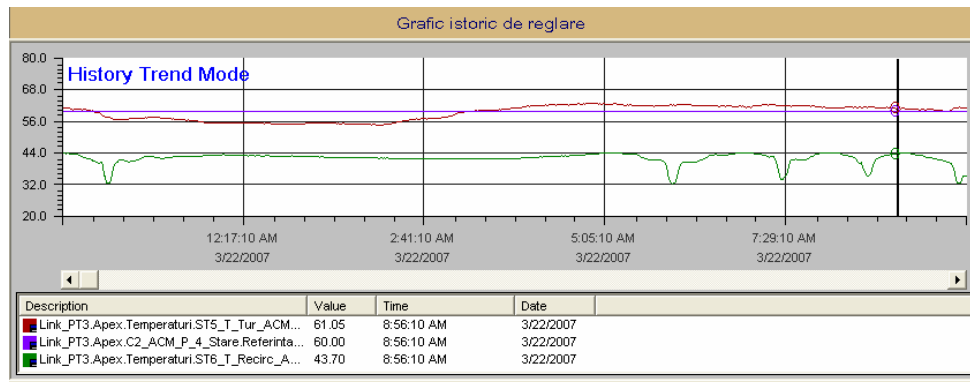


Fig. 4.7. Exemplu de grafic al istoricului reglării.

Configurarea aplicației se realizează cu instrumente grafice specializate fiecărui tip de activitate, după cum urmează:

- Servere OPC. Configurarea se realizează cu instrumente grafice prin care se definesc punctele din sistem, tipul lor, limite tehnologice și ingineresti, parametri fizici de acces precum și orice alte date necesare accesării respectivei informații și care sunt dependente de fiecare tip de server OPC în parte. Tot aici se construiește structura arborescentă de acces la parametri și se dă un nume fiecăruia dintre ei pentru a putea ulterior fi referit de către aplicațiile grafice client;
- Imaginile grafice se construiesc cu editorul specializat Iconics GraphWorX32 care permite crearea imaginilor cu ajutorul simbolurilor, plasarea valorilor analogice și a simbolurilor grafice pe imaginea creată, adăugarea de funcții de culoare, deplasare, pâlpâire etc. în funcție de cerințe;

Toți parametrii specifici unui trend sunt modificabili, inclusiv în regim runtime:

- intervalul de timp și rezoluția pe axa timp;
- domeniul pe axa valorilor (se poate opta pentru un singur domeniu sau pentru domenii diferite pentru fiecare mărime în parte);
- tipul diagramelor, culori pentru fundal și text, grosimea și tipul carioajului, etc.;
- caracteristicile diagramelor (grosime linie, culoare, tip linie, etc.);
- legenda (tip fonturi, culori);
- se poate trage prin drag & drop o mărime pentru a i se face graficul.

Configurarea aplicației se realizează cu instrumente grafice specializate fiecărui tip de activitate după cum urmează:

- Curbele sunt gestionate cu utilitarul TrendWorX. Se pot specifica toți parametrii necesari (lista de semnale, culori, domenii, grosimea liniei, etc.);
- Alarmerile se configurează cu utilitarul grafic AlarmServerConfigurator. Se pot defini tipul alarmei, limitele, nivelul alarmei, dacă necesită acceptare sau nu, textul mesajului de alarmă, etc. Cu aplicația AlarmLoggerConfigurator se configurează modul de memorare a alarmelor în arhivă;
- Istoricul se definește cu utilitarele specializate TrendWorX32 Configurator și permit definirea de grupe de mărimi de memorat pentru care se precizează condițiile de memorare (frecvență, condiția de start, condiția de stop etc.);
- Rapoartele se configurează grafic cu ReportWorX.

Alarmerile sunt gestionate de componenta AlarmWorX. La apariția unei alarme noi aceasta este plasată în lista de alarme active. Ea poate fi acceptată de către utilizator (și se poate adăuga un comentariu liber) dar nu va dispărea din sistem până la momentul dispariției cauzei alarmei, moment în care va fi mutată în arhiva de alarme istorice.

În procesul de configurare se stabilește dacă o alarmă trebuie neapărat acceptată sau nu, se introduc informații ajutătoare pentru rezolvarea alarmei respective și se poate seta o listă de mărimi care au relevanță cu alarma de față pentru a putea fi rapid vizualizate de operator.

În fereastra de alarmare se prezintă toate alarmele active din sistem. În funcție de dimensiunea ferestrei de alarmare și de numărul de alarme sunt prezente bare de defilare pentru vizualizarea întregii liste. Alarmele pot fi filtrate după diferite criterii sau ordonate (de ex. după prioritate).

Nu există limitare în ceea ce privește numărul de alarme ce pot fi memorate. Există alarme de tip analogic (depășiri de limite), binare (schimbare de stare), rata de schimbare a unui semnal, defect de comunicație.

Ștampila de timp se asociază unui eveniment, de regulă de către echipamentul EAD, deci depinde de tipul de EAD. Sistemul de alarmare Iconics Genesis32 poate gestiona alarme și evenimente cu precizia de 1 ms. Vizualizarea alarmelor se face în ferestre dedicate de alarmare (parte din interfața grafică) care pot avea orice dimensiuni.

4.5. Secvențe de program pentru realizarea interfeței om-mașină

În acest capitol vom descrie secvențele de program aferente realizării interfeței grafice om-mașină. Această secvență de program este generată utilizând mediul de programare Visual C.

```
//-----
// Procedura de navigare cu mouse-ul pe scheme sinoptice
// Aplicatie PDFe2
//
void CSchWnd::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    CString sNumeFerIni;
    CRect rd;
    char wname[50];
    char bf[15],cx[40];
    char nume[50];
    int rez;
    genbd_type *ap,*apm;
    wname[0]=0;
    for(int ai=0;ai<nrAna;ai++)
    {
        if(rc[ai].PtInRect(point))
        {
            if(NULL==(ap=pWBDate->atm_find_off(ida[ai]/*,val,stare,modif*/))) {
                break; // poate este necesar sa intre in adancime
            }
            CWnd *pDsk=GetDesktopWindow();
            if(pDsk!=NULL)
                pDsk->GetClientRect(&rd);
            if (nFlags & MK_CONTROL) {
                char mes[100];
                id_type idm;
                CRect r(point.x+10,point.y,point.x,point.y);
```

```

CMainFrame* pM=(CMainFrame*)pMainWnd;
if(pM->fii)
    pWndLim->DestroyWindow();
pWndLim=new CLim;
switch (ida[ai].lista) {
case aA:
case AA:
    r.right+=330;
    r.bottom+=220;
    pWndLim->lim_adm_inf=ap->wa.mi;
    pWndLim->lim_adm_sup=ap->wa.ms;
    pWndLim->lim_avarie_inf=ap->wa.lia;
    pWndLim->lim_avarie_sup=ap->wa.lsa;
    pWndLim->lim_prev_inf=ap->wa.lip;
    pWndLim->lim_prev_sup=ap->wa.lsp;
    pWndLim->lim_conv_inf=ap->wa.litr;
    pWndLim->lim_conv_sup=ap->wa.lstr;
    pWndLim->canal=ap->wa.canal;
    pWndLim->id=ap->wa.id;
    pWndLim->tscan=ap->wa.tscan;
    pWndLim->atrib=ap->wa.atrib;
    pWndLim->prel_inst=ap->wa.prel_inst;
    idm.lista=ap->wa.id_placa.lista;
    idm.echidx=ap->wa.id_placa.echidx&0xffc0;
    if(idm.lista==MA){
        if (NULL!=(apm=pWBDate->atm_find_off(idm)))
            strcpy(pWndLim->ech_name,apm->wm.name);
        else
            strcpy(pWndLim->ech_name,"Neconfigurat");
        idm=ap->wa.id_placa;
        if (NULL!=(apm=pWBDate->atm_find_off(idm)))
            strcpy(pWndLim->module_name,apm->wm.name);
        else
            strcpy(pWndLim->module_name,"Neconfigurat");
    }
    else{
        strcpy(pWndLim->ech_name,"Neconfigurat");
        strcpy(pWndLim->module_name,"Neconfigurat");
    }
    strcpy(mes,ap->wa.name);
    strcat(mes," (analogica)");
    break;
case cA:
case CA:
    r.right+=330;
    r.bottom+=140;
    pWndLim->lim_adm_inf=ap->wc.di;
    pWndLim->lim_adm_sup=ap->wc.ds;
    pWndLim->lim_avarie_inf=ap->wc.lia;
    pWndLim->lim_avarie_sup=ap->wc.lsa;
    pWndLim->lim_prev_inf=ap->wc.lip;
    pWndLim->lim_prev_sup=ap->wc.lsp;
    pWndLim->id=ap->wc.id;
    pWndLim->atrib=ap->wc.atrib;
    pWndLim->prel_inst=ap->wc.prel_inst;
    strcpy(mes,ap->wc.name);
    strcat(mes," (calcul)");
    break;

```

```

        default:
            CMDIChildWnd::OnLButtonDown(nFlags, point);
            return;
    }
    ClientToScreen(&r);
    if(r.right>rd.right){
        r.left-=r.right-rd.right+5;
        r.right=rd.right-5;
    }
    if(r.bottom>rd.bottom-20){
        r.top-=r.bottom-rd.bottom+20;
        r.bottom=rd.bottom-20;
    }
    pMainWnd->ScreenToClient(&r);
    pWndLim->Create(NULL,mes,WS_CHILD | WS_VISIBLE |
                    WS_BORDER | WS_CAPTION | WS_SYSMENU,r,NULL);
    CMDIChildWnd::OnLButtonDown(nFlags, point);
    return;
}

switch(ida[ai].lista){
    case AA:
    case aA:
        strcpy(wname,ap->wa.name);
        break;
    case CA:
    case cA:
        strcpy(wname,ap->wc.name);
        break;
    case TA:
        strcpy(wname,ap->wt.name);
        break;
    default:
        CMDIChildWnd::OnLButtonDown(nFlags, point);
        return;
}

    CTime t = CTime::GetCurrentTime();
    strcat(strcpy(ume,DirArhiva),"\");
    strcat(ume,Luni[t.GetMonth()-1]);
    sprintf(bf,"\\%d",t.GetDay());
    strcat(strcat(ume,bf),"\");
    strcat(strcat(ume,Id2Str(ida[ai],cx)),".his");
    struct _stat st;
    rez=_stat(ume,&st);
    if(rez!=0){ // Nu exista arhiva cu date
        MessageBox("Nu exista date pentru grafic in arhiva!", "Scuze dar...",
MB_ICONEXCLAMATION);
        CMDIChildWnd::OnLButtonDown(nFlags, point);
        return;
    }
    else {
        CRect r(point.x+10,point.y,point.x+560,point.y+350);
        CWnd *pDsk;
        CRect rd;
        pDsk=GetDesktopWindow();
        if(pDsk!=NULL)
            pDsk->GetClientRect(&rd);
        ClientToScreen(&r);
        if(r.right>rd.right){

```

```

    r.left=r.right-rd.right+5;
    r.right=rd.right-5;
}
if(r.bottom>rd.bottom-20){
    r.top=r.bottom-rd.bottom+20;
    r.bottom=rd.bottom-20;
}
pMainWnd->ScreenToClient(&r);
CGraf2 *pGLWnd = new CGraf2;
    strcat(strcpy(ume,wname)," azi");
pGLWnd->zua=t.GetDay();
pGLWnd->luna=t.GetMonth();
pGLWnd->anul=t.GetYear();
    pGLWnd->actualizare=1;
    pGLWnd->m_indg=ida[ai];
    strcpy(pGLWnd->numew,wname);

    if (!pGLWnd->Create(ume,WS_CHILD | WS_VISIBLE | WS_OVERLAPPED |
WS_CAPTION |
        WS_SYSMENU | WS_THICKFRAME | WS_MINIMIZEBOX |
WS_MAXIMIZEBOX,
                                r,pMainWnd)){
        CMDIChildWnd::OnLButtonDown(nFlags, point);
        return;
    }
    return;
}
}
}

for(int b=0;b<nrBin;b++)
{
    if(rcbin[b].PtInRect(point) {
        genbd_type *ap,*apm;
        if(NULL==(ap=pWBDate->atm_find_off(idbin[b]))) {
            break; // pentru a permite intrarea in adancime
        }
        if (nFlags & MK_CONTROL) {
            char mes[100];
            id_type idm;
            CRect rd,r(point.x+10,point.y,point.x,point.y);
            CWnd *pDsk;
            pDsk=GetDesktopWindow();
            if(pDsk!=NULL)
                pDsk->GetClientRect(&rd);
            CMainFrame* pM=(CMainFrame*)pMainWnd;
            if(pM->fii)
                pWndLim->DestroyWindow();
            pWndLim=new CLim;
            switch (idbin[b].lista) {
                case DA:
                case BA:
                case bA:
                    r.right+=340;
                    r.bottom+=170;
                    pWndLim->canal=ap->wb.canal;
                    pWndLim->id=ap->wb.id;
                    pWndLim->tscan=ap->wb.filt;

```

```

    pWndLim->atrib=ap->wb.atrib;
    idm.lista=ap->wb.id_placa.lista;
    idm.echidx=ap->wb.id_placa.echidx&0xffc0;
    if(idm.lista==MA){
        if (NULL!=(apm=pWBDate->atm_find_off(idm)))
            strcpy(pWndLim->ech_name,apm->wm.name);
        else
            strcpy(pWndLim->ech_name,"Neconfigurat");
        idm=ap->wb.id_placa;
        if (NULL!=(apm=pWBDate->atm_find_off(idm)))
            strcpy(pWndLim->module_name,apm->wm.name);
        else
            strcpy(pWndLim->module_name,"Neconfigurat");
    }
    else {
        strcpy(pWndLim->ech_name,"Neconfigurat");
        strcpy(pWndLim->module_name,"Neconfigurat");
    }
    strcpy(mes,ap->wb.name);
    strcpy(pWndLim->diag_norm,ap->wb.diag_norm);
    strcpy(pWndLim->diag_anorm,ap->wb.diag_anorm);
    strcat(mes," (binara)");
    break;
default:
    CMDIChildWnd::OnLButtonDown(nFlags, point);
    return;
}
ClientToScreen(&r);
if(r.right>rd.right){
    r.left=r.right-rd.right+5;
    r.right=rd.right-5;
}
if(r.bottom>rd.bottom-20){
    r.top=r.bottom-rd.bottom+20;
    r.bottom=rd.bottom-20;
}
pMainWnd->ScreenToClient(&r);
pWndLim->Create(NULL,mes,WS_CHILD | WS_VISIBLE |
                WS_BORDER | WS_CAPTION | WS_SYSMENU,r,NULL);
CMDIChildWnd::OnLButtonDown(nFlags, point);
return;
}
//grafic marime binara
switch(idbin[b].lista) {
    case DA:
    case BA:
    case bA:
        strcpy(wname,ap->wb.name);
        break;
default:
    CMDIChildWnd::OnLButtonDown(nFlags, point);
    return;
}

    CTime t = CTime::GetCurrentTime();
    strcat(strcpy(ume,DirArhiva),"\");
    strcat(ume,Luni[t.GetMonth()-1]);
    sprintf(bf,"\\%d",t.GetDay());

```

```

        strcat(strcat(nume,bf),"\\");
        strcat(strcat(nume,Id2Str(idbin[b],cx)),".his");
        struct _stat st;
        rez=_stat(nume,&st);
        if(rez!=0){
            MessageBox("Nu exista date pentru grafic in arhiva!","Scuze dar...",
MB_ICONEXCLAMATION);
            CMDIChildWnd::OnLButtonDown(nFlags, point);
            return;
        }
        else {
            CRect r(point.x+10,point.y,point.x+560,point.y+350);
            CWnd *pDsk;
            CRect rd;
            pDsk=GetDesktopWindow();
            if(pDsk!=NULL)
                pDsk->GetClientRect(&rd);
            ClientToScreen(&r);
            if(r.right>rd.right){
                r.left-=r.right-rd.right+5;
                r.right=rd.right-5;
            }
            if(r.bottom>rd.bottom-20){
                r.top-=r.bottom-rd.bottom+20;
                r.bottom=rd.bottom-20;
            }
            pMainWnd->ScreenToClient(&r);
            CGraf2 *pGLWnd = new CGraf2;
                strcat(strcpy(nume,wname)," azi");
            pGLWnd->zium=t.GetDay();
            pGLWnd->luna=t.GetMonth();
            pGLWnd->anul=t.GetYear();
            pGLWnd->actualizare=1;
            pGLWnd->m_indg=idbin[b];
            strcpy(pGLWnd->numew,wname);
            if (!pGLWnd->Create(nume,WS_CHILD | WS_VISIBLE | WS_OVERLAPPED | WS_CAPTION |
WS_SYSMENU | WS_THICKFRAME | WS_MINIMIZEBOX | WS_MAXIMIZEBOX,r,pMainWnd)){
                return;
            }
            return;
        }
    }

    for(int ri=0;ri<nrRect;ri++) {
        if(rct[ri].PtInRect(point)) {
            CSchWnd *pSchemaW;
            CRect rect,rec1;
            int fh,result,xbmp,ybmp;
            line_type lin;
            char fname[64];
            char numebmp[32];

            strcat(strcat(strcpy(fname,DirBdate),"\\"),"grafic.ini");
            if (-1 != (fh=fopen(fname,O_RDONLY | O_TEXT))) {

                up_case(szNumeF[ri],numebmp);
                if (find_line(fh,numebmp,"BITMAP",lin)) {

```

```

        copy_par(lin,MAXPAR,TabelPar,result);
        if (result || (strlen(TabelPar[1])==0)) {
            MessageBox("Linie incorecta in .ini","Eroare!",MB_ICONSTOP);
            tclose(fh);
            return;
        }
        xbmp=ini_integer(TabelPar[2],result);
        ybmp=ini_integer(TabelPar[3],result);
            tclose(fh);
    }
    else {
        MessageBox("Nu este specificat numele bitmap-ului!\n","Eroare!",MB_ICONSTOP);
        tclose(fh);
        return;
    }
}
else {
    MessageBox("Nu exista fisierul grafic.ini!\n","Eroare!",MB_ICONSTOP);
    tclose(fh);
    return;
}
}

    GetWindowRect(&rect1);
    pMainWnd->GetClientRect(&rec1);
    rect.left=xbmp;
    rect.top =ybmp;
    rect.right=rect1.right;
    rect.bottom=rect1.bottom;
    Mod=1; //S-a modificat pozitia ferestrei parinte, deci nu o distrug.
    if((pSchemaW=ListaFer.IsInList(szNumeF[ri]))==NULL)
    {
        pSchemaW=new CSchWnd;
        if (!pSchemaW->Create(szNumeF[ri],WS_CHILD | WS_VISIBLE | WS_BORDER |
WS_SYSMENU | WS_CAPTION | WS_MINIMIZEBOX ,rect,NULL))
        {
            CMDIChildWnd::OnLButtonDown(nFlags, point);
            return;
        }
    }
    else
        pSchemaW->BringWindowToTop();
    CMDIChildWnd::OnLButtonDown(nFlags, point);
    if(Mod!=1)
    {
        if(strcmp(NumeFerIni,szNumeFer)!=0)
            DestroyWindow();
        }
        return;
    }
}
}
    CMDIChildWnd::OnLButtonDown(nFlags, point);
}
// Handler pentru click mouse
// Aplicatie PDFe2
//
void CBargraf::OnLButtonDown(UINT nFlags, CPoint point)
{

```



```

// TODO: Add your message handler code here and/or call default
CRect r;
char wname[50];
wname[0]=0;
for(int i=0;i<data->nr_marimi;i++)
{
    r.left=m_coord[0]+1; r.top=(i+1)*sp-sp/20-m_coord[2]+1;
    r.right=m_coord[1]-1; r.bottom=(i+1)*sp-sp/20-1;
    if(r.PtInRect(point))
    {
        char bf[15],cx[40];
        char nume[50];
        int rez;
        genbd_type *ap,*apm;
        if(NULL==(ap=pWBDate->atm_find_off(data->marime_bargraf[i].id))) {
            CMDIChildWnd::OnLButtonDown(nFlags, point);
            return;
        }
        CRect rd;
        CWnd *pDsk=GetDesktopWindow();
        if(pDsk!=NULL)
            pDsk->GetClientRect(&rd);
        if (nFlags & MK_CONTROL) {
            char mes[100];
            id_type idm;
            CRect r(point.x+10,point.y,point.x,point.y);
            CMainFrame* pM=(CMainFrame*)pMainWnd;
            if(pM->fii)
                pWndLim->DestroyWindow();
            pWndLim=new CLim;
            switch (data->marime_bargraf[i].id.lista) {
                case aA:
                case AA:
                    r.right+=330;
                    r.bottom+=220;
                    pWndLim->lim_adm_inf=ap->wa.mi;
                    pWndLim->lim_adm_sup=ap->wa.ms;
                    pWndLim->lim_avarie_inf=ap->wa.lia;
                    pWndLim->lim_avarie_sup=ap->wa.lsa;
                    pWndLim->lim_prev_inf=ap->wa.lip;
                    pWndLim->lim_prev_sup=ap->wa.lsp;
                    pWndLim->lim_conv_inf=ap->wa.litr;
                    pWndLim->lim_conv_sup=ap->wa.lstr;
                    pWndLim->canal=ap->wa.canal;
                    pWndLim->id=ap->wa.id;
                    pWndLim->tscan=ap->wa.tscan;
                    pWndLim->atrib=ap->wa.atrib;
                    pWndLim->prel_inst=ap->wa.prel_inst;
                    idm.lista=ap->wa.id_placa.lista;
                    idm.echidx=ap->wa.id_placa.echidx&0xffc0;

                    if(idm.lista==MA){
                        if (NULL!=(apm=pWBDate->atm_find_off(idm)))
                            strcpy(pWndLim->ech_name,apm->wm.name);
                        else
                            strcpy(pWndLim->ech_name,"Neconfigurat");
                        idm=ap->wa.id_placa;
                        if (NULL!=(apm=pWBDate->atm_find_off(idm)))
                            strcpy(pWndLim->module_name,apm->wm.name);
                    }
            }
        }
    }
}

```

```

        else
            strcpy(pWndLim->module_name,"Neconfigurat");
        }
        else{
            strcpy(pWndLim->ech_name,"Neconfigurat");
            strcpy(pWndLim->module_name,"Neconfigurat");
        }
        strcpy(mes,ap->wa.name);
        strcat(mes," (analogica)");
        break;
    case cA:
    case CA:
        r.right+=330;
        r.bottom+=140;
        pWndLim->lim_adm_inf=ap->wc.di;
        pWndLim->lim_adm_sup=ap->wc.ds;
        pWndLim->lim_avarie_inf=ap->wc.lia;
        pWndLim->lim_avarie_sup=ap->wc.lsa;
        pWndLim->lim_prev_inf=ap->wc.lip;
        pWndLim->lim_prev_sup=ap->wc.lsp;
        pWndLim->id=ap->wc.id;
        pWndLim->atrib=ap->wc.atrib;
        pWndLim->prel_inst=ap->wc.prel_inst;
        strcpy(mes,ap->wc.name);
        strcat(mes," (calcul)");
        break;
    default:
        CMDIChildWnd::OnLButtonDown(nFlags, point);
        return;
    }
    ClientToScreen(&r);
    if(r.right>rd.right){
        r.left=r.right-rd.right+5;
        r.right=rd.right-5;
    }
    if(r.bottom>rd.bottom-20){
        r.top=r.bottom-rd.bottom+20;
        r.bottom=rd.bottom-20;
    }
    pMainWnd->ScreenToClient(&r);
    pWndLim->Create(NULL,mes,WS_CHILD | WS_VISIBLE |
        WS_BORDER | WS_CAPTION | WS_SYSMENU,r,NULL);
    CMDIChildWnd::OnLButtonDown(nFlags, point);
    return;
}
switch(data->marime_bargraf[i].id.lista){
    case AA:
    case aA:
        strcpy(wname,ap->wa.name);
        break;
    case CA:
    case cA:
        strcpy(wname,ap->wc.name);
        break;
    default:
        CMDIChildWnd::OnLButtonDown(nFlags, point);
        return;
    }
}

```

```

    CTime t = CTime::GetCurrentTime();
    strcat(strcpy(ume,DirArhiva),"\");
    strcat(ume,Luni[t.GetMonth()-1]);
    sprintf(bf,"\\%d",t.GetDay());
    strcat(strcat(ume,bf),"\");
    strcat(strcat(ume,Id2Str(data->marime_bargraf[i].id,cx)),".his");
    struct _stat st;
    rez=_stat(ume,&st);
    if(rez!=0){
        MessageBox("Nu exista date pentru grafic in arhiva!","Scuze dar...",
MB_ICONEXCLAMATION);
        CMDIChildWnd::OnLButtonDown(nFlags, point);
        return;
    }
    else {
        CRect r(point.x+10,point.y,point.x+560,point.y+350);
        ClientToScreen(&r);
        if(r.right>rd.right){
            r.left=r.right-rd.right+5;
            r.right=rd.right-5;
        }
        if(r.bottom>rd.bottom-20){
            r.top=r.bottom-rd.bottom+20;
            r.bottom=rd.bottom-20;
        }
        pMainWnd->ScreenToClient(&r);
        CGraf2 *pGLWnd = new CGraf2;
        strcat(strcpy(ume,wname)," , azi");
        pGLWnd->ziua=t.GetDay();
        pGLWnd->luna=t.GetMonth();
        pGLWnd->anul=t.GetYear();
        pGLWnd->actualizare=1;
        pGLWnd->m_indg=data->marime_bargraf[i].id;
        strcpy(pGLWnd->numew,wname);
        if (!pGLWnd->Create(ume,WS_CHILD | WS_VISIBLE | WS_OVERLAPPED |
WS_CAPTION |
        WS_SYSMENU | WS_THICKFRAME | WS_MINIMIZEBOX |
WS_MAXIMIZEBOX,
        r,pMainWnd)){
            return;
        }
    }
}
}
}
}
}
//-----
// Afisare grafice
//

void CGrafX::OnPaint()
{
    CPaintDC dc(this); // device context for painting
    CRect r;
    GetClientRect(r);
    if((stfocus==1)&&(actlin==1)){
        marclinie(antpoint);
        actlin=0;
    }
}

```

```

    DesenPaint(&dc,&r);
    // Do not call CMDIChildWnd::OnPaint() for painting messages
}
//-----
extern HPEN hpen[13];
void CGrafX::DesenPaint(CDC *pdc,RECT *r) {
    pMyApp->DoWaitCursor(1);
    height=(r->bottom-r->top)*heighti/Hi; //noua inaltime a fontului
    width=(r->right-r->left)*widthi/Wi; //noua latime a fontului
    H=r->bottom-r->top; //noua inaltime a ferestrei
    W=r->right-r->left; //noua latime a ferestrei
    //coordonate zona utila grafic
    if(flana) {
        int lung;
        char sir[10];
        int lg,lg1;
        sprintf(sir,"%0f",m_limval[0]);
        lg=strlen(sir);
        sprintf(sir,"%0f",m_limval[1]);
        lg1=strlen(sir);
        if(lg>lg1) lung=lg;
        else lung=lg1;
        if (lung<5) lung=5;
        m_coord[0]=r->left+(lung+5)*width;
    }
    else
        m_coord[0]=r->left+W/30;
    m_coord[1]=r->bottom-H/20-(m_nrgraf+1)*height-(m_nrgraf+1)*H/100;
    m_coord[2]=r->right-W/30;
    m_coord[3]=r->top+height+height/4;

    TrasareAxe(pdc,nr_ore/24,10); //24 gradatii pe x, 10 gradatii pe y
    ValGrafice(pdc,r,nr_ore/24,10);
    if(flana)
        Curbe(pdc,r);
    else
        Diagrame(pdc,r);
    pMyApp->DoWaitCursor(-1);
}
//-----
// Trasare curbe
//
void CGrafX::Curbe(CDC *pdc, RECT *r) {
    int i,j,x,y,litr,lstr,tipc;
    short v,tmax,pas=1;
    float vf,vf1,cic,px,py,di,ds;
    BOOL prim;
    genbd_type *ap;
    CPoint pnt[2];
    time_t ltime;
    time (&ltime);
    struct tm *timpcrt;
    timpcrt=localtime(&ltime);
    if (timer)
        tmax=timpcrt->tm_hour*60+timpcrt->tm_min;
    else
        tmax=24*60;
        tmax=nr_ore;

```

```

for(j=0; j<m_nrgraf; j++) {
    HPEN pOldPen = (HPEN)SelectObject(pdc->m_hDC,hpen[indcul[j]]);
    ap=pWBDate->atm_find_off(ids[j]);
    switch (ids[j].lista) {
        case AA: case aA:
            ds=ap->wa.ms;
            di=ap->wa.mi;
            litr=ap->wa.litr;
            lstr=ap->wa.lstr;
            if (ap->wa.prel_inst==PI_A_LIN) {
                tipc=0;
            }
            cic=(ds-di)/(float)(lstr-litr);
        }
        else {
            tipc=1;
            cic=(ds-di)/(float)sqrt((double)(lstr-litr));
        }
        break;
        case CA: case cA:
            ds=ap->wc.ds;
            di=ap->wc.di;
            litr=0;
            lstr=4096;
            tipc=0;
            cic=(ds-di)/(float)(lstr-litr);
            break;
        case TA:
            ds=ap->wt.ds;
            di=ap->wt.di;
            litr=0;
            lstr=4096;
            tipc=0;
            cic=(ds-di)/(float)(lstr-litr);
            break;
    }
    px=pas * (m_coord[2]-m_coord[0]) / (float)nr_ore; //(float)(24*60);
    py=(float)(m_coord[1]-m_coord[3])/(m_limval[1]-m_limval[0]);
    prim=TRUE;
    for (i=0; i<tmax; i++) {
// extrage urmatoarea valoare din buffer
        vf=buff[j][i];
        if (vf==V_GOL)
            v=-2;
        else
            v=1;
        if (v>=0) { // valoare existenta (v=-2 daca e inexistentă)
            if (tipgraf) vf1=vf;
            else vf1=100*(vf-di)/(ds-di); //conversie in procente
            if (vf1>m_limval[1]) vf1=m_limval[1];
            if (vf1<m_limval[0]) vf1=m_limval[0];
            x=(int)(m_coord[0]+i*px);
            y=(int)(m_coord[1]-(vf1-m_limval[0])*py);
        }
        if (prim) {
            pdc->MoveTo(x,y); //pentru primul punct se face numai pozitionare
            prim=FALSE;
            pdc->SetPixel(x,y,indcul[j]);
            oldx=x;
            oldy=y;
        }
    }
}

```

```

    }
    else
        if((oldx!=x)||(oldy!=y)){
            pnt[0].x=oldx;
            pnt[0].y=oldy;
            pnt[1].x=x;
            pnt[1].y=y;
            pdc->Polyline(pnt,2);

            oldx=x;
            oldy=y;
        }

        if (i==tmax-1) //sint la sfirsit
        {
            pnt[0].x=oldx;
            pnt[0].y=oldy;
            pnt[1].x=oldx+px;
            pnt[1].y=oldy;
            pdc->Polyline(pnt,2);
        }
    }
    else {
        vf=V_GOL;
        prim=TRUE;
    }
}

SelectObject(pdc->m_hDC,pOldPen);
if (i>0) SetValCurenta(pdc,j,vf,"",r); // daca in buffer (fisier) era macar o valoare
}
}
//-----
// Trasare Diagrame
//
void CGrafX::Diagrame(CDC *pdc, RECT *r) {

    int i,j,x,y; //litr,lstr;
    short v,tmax,pas=1;
    float vf,px,py; //vf1,cic,di,ds;
    char diag[32]="";
    BOOL prim;
    genbd_type *ap;
    CTime t=CTime::GetCurrentTime();
    CPoint pnt[2];
    struct tm *timp=t.GetLocalTm(NULL);
    if (timer)
        tmax=timp->tm_hour*60+timp->tm_min;
    else
        tmax=24*60;
    py=(float)(m_coord[1]-m_coord[3])/m_nrgraf;
    for(j=0; j<m_nrgraf; j++) {
        HPEN pOldPen = (HPEN)SelectObject(pdc->m_hDC,hpen[indcul[j]]);
        ap=pWBDate->atm_find_off(ids[j]);
        px=pas * (m_coord[2]-m_coord[0]) / (float)(24*60);
        SetGetVal(GV_MINUT,buff[j],bufflen[j]); //pregateste variabilele pentru GetVal
        prim=TRUE;
        for (i=0; i<tmax; i++) {

            if(v>=0){
                x=(int)(m_coord[0]+i*px);
            }

```

```

        vf=1-v;
        if(vf==1)
            y=(int)(m_coord[1]-(j*py+py*0.8));
        else
            y=(int)(m_coord[1]-(j*py+py*0.2));
    if (prim) {
        pdc->MoveTo(x,y); //pentru primul punct se face numai pozitionare
        prim=FALSE;
        pdc->SetPixel(x,y,indcul[j]);
        oldx=x;
        oldy=y;
    }
    else
        if((oldx!=x)||(oldy!=y)){
            pdc->LineTo(x,oldy);
            pdc->LineTo(x,y);
            oldx=x;
            oldy=y;
        }
    }
    else {
        vf=V_GOL;
        prim=TRUE;
    }
}
if (v==VL_B_NORMAL)
    strcpy(diag,ap->wb.diag_norm);
else
    strcpy(diag,ap->wb.diag_anorm);
    SelectObject(pdc->m_hDC,pOldPen);
if (i>0) SetValCurenta(pdc,j,vf,diag,r); // daca in buffer (fisier) era macar o valoare
    }
}
//-----
// Desenare fereastra evenimente
//
void CEvFis::OnPaint() {
    int i;
    eventl str;
    CPaintDC dc(this); // device context for painting

    // TODO: Add your message handler code here
    CRect rct;
    CBrush br(RGB(255,255,0));
    GetClientRect(&rct);
    ninceput=max(0,nVscrollpos+rct.top/nychar);
    nsfirsit=min(nnrinii,nVscrollpos+rct.bottom/nychar+1);

    HGDIOBJ hfont=GetStockObject(SYSTEM_FIXED_FONT);
    SelectObject(dc.m_hDC,hfont);
    for(i=ninceput;i<nsfirsit;i++) {
        str=buff[i];
        if(str.atrib==AT_ACTIV) {
            if (str.stare==VL_B_DEF) dc.SetTextColor(RGB(183,193,192));
                else if(str.stare==VL_B_NORMAL) dc.SetTextColor(RGB(33,130,0));
            else if(str.stare==VL_B_ANORMAL) dc.SetTextColor(RGB(255,0,0));
            else if(str.stare==4) dc.SetTextColor(RGB(255,255,0));
            else if(str.stare==5) dc.SetTextColor(RGB(0,0,0));
        }
    }
}

```

```

        if(str.atrib==AT_PASIV)          dc.SetTextColor(RGB(0,0,0));
        dc.SetBkColor(RGB(192,192,192)); //fondul
            dc.TextOut(0,(i-nVscrollpos)*nychar,szbuffer,sprintf(szbuffer," %s",str.stri));
        }
        // Do not call CMDIChildWnd::OnPaint() for painting messages
    }
void CEvFis::OnSize(UINT nType, int cx, int cy) {
    CMDIChildWnd::OnSize(nType, cx, cy);

    // TODO: Add your message handler code here
    nyclient=cy;
    nVscrollmax=max(0,nrlnii-nyclient/nychar);
    nVscrollpos=min(nVscrollpos,nVscrollmax);
    SetScrollRange(SB_VERT,0,nVscrollmax,FALSE);
    SetScrollPos(SB_VERT,nVscrollpos,TRUE);
}
void CEvFis::OnVScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar) {
    // TODO: Add your message handler code here and/or call default
    switch(nSBCode){
        case SB_LINEUP:
            ninc=-1;
            break;
        case SB_LINEDOWN:
            ninc=1;
            break;
        case SB_PAGEUP:
            ninc=min(-1,-nyclient/nychar);
            break;
        case SB_PAGEDOWN:
            ninc=max(1,nyclient/nychar);
            break;
        case SB_TOP:
            ninc=-nVscrollpos;
            break;
        case SB_BOTTOM:
            ninc=nVscrollmax-nVscrollpos;
            break;
        case SB_THUMBTRACK:
            ninc=nPos-nVscrollpos;
            break;
        default:
            return;
    }
    ninc=max(-nVscrollpos,min(ninc,nVscrollmax-nVscrollpos));
    nVscrollpos+=ninc;
    ScrollWindow(0,-nychar*ninc);
    SetScrollPos(SB_VERT,nVscrollpos,TRUE);
    UpdateWindow();
    CMDIChildWnd::OnVScroll(nSBCode, nPos, pScrollBar);
}
//-----
// Operatii initiale de creare rapoarte
//
int CReport::OnCreate(LPCREATESTRUCT lpCreateStruct) {
    if (CMDIChildWnd::OnCreate(lpCreateStruct) == -1)
        return -1;
}

```



```

// TODO: Add your specialized creation code here
CRect rcWin,rcCIWin;
HFILE hf;
LONG colors,sizeimage,c,counti;
TEXTMETRIC tm;
LOGFONT f;
int fh,rez,i,j,m,n,tipbm,nrd;
char fname[80],nume_sch[80],idsi[20][100],vls[50][10],oldcwd[80],stare;
int xi[10],yi[10],xinc[10],yinc[10],nrlin[10],nrcol[10];
float value;
BOOL fl_paint;
genbd_type *ap, *app;
char numedir[64];
fis_type fis;

Expresie expr; //pentru calcule pe coloane
CString formula;

m_bHasBits=FALSE;
//Initializare Cels
for (i=0; i<600; i++) {
    Cels[i].id.lista=0;
    Cels[i].tip='x';
    strcpy(Cels[i].buffer,"");
}
strcat(strcpy(fname,DirBdate),"\arcview.ini");
//strcpy(fname,"..\offdb\arcview.ini");
fh=topen(fname,O_RDONLY | O_TEXT);
up_case(numew,numew);

//suport pt. raport automat
autobitmap=false;

//pregateste nume fisier .bmp
if (find_line(fh,numew,"BITMAP",Lin))
{
    copy_par(Lin,MAXPAR,TabelPar,Result);
    if (Result) {
        /*
        MessageBox("Nume bitmap in arcview.ini","Eroare!",MB_ICONSTOP);
        tclose(fh);
        return -1;
        */
        autobitmap=true;
    }
    else
    {
        strcpy(nume_sch,DirScheme);
        strcat(nume_sch,"\\");
        strcat(nume_sch,TabelPar[1]);
        strcat(nume_sch,".BMP");
    }
}
else
    autobitmap=true;

//tip raport
find_line(fh,numew,"TIP",Lin);
copy_par(Lin,MAXPAR,TabelPar,Result);

```

```

//nr. tabele in raport
nrTab=ini_integer(TabelPar[1],Result);
if (Result) {
    MessageBox("Numar tabele in arcview.ini","Eroare!",MB_ICONSTOP);
    tclose(fh);
    return -1;
}
    if (autobitmap && (nrTab!=1))
    {
        MessageBox("Pentru desenare automata nr. de tabele trebuie sa fie 1",
"Eroare",MB_ICONSTOP);
        tclose(fh);
        return -1;
    }

    if (nrTab>10) nrTab=10; //maxim 10 tabele
//nume fisier curent
strcpy(CrtFileName,"");
CrtFileHnd=-1;
//tip date (zi sau luna)
up_case(TabelPar[2],TabelPar[2]);
if (0==strcmp(TabelPar[2],"LUNA")) TipRap='L';
else
    if (0==strcmp(TabelPar[2],"ZI")) TipRap='Z';
    else {
        MessageBox("Tip raport in arcview.ini","Eroare!",MB_ICONSTOP);
        tclose(fh);
        return -1;
    }
//preia datele din ini, pentru fiecare tabel
nrId=0;
for(i=0; i<nrTab; i++) {
    itoa(i+1,cx,10);
    strcpy(cod,"T");
    strcat(cod,cx);
    find_line(fh,numew,cod,Lin);
    copy_par(Lin,MAXPAR,TabelPar,Result);
    if (Result) {
        MessageBox("Coordonate tabel in arcview.ini","Eroare!",MB_ICONSTOP);
        tclose(fh);
        return -1;
    }
    //x,y colt stanga sus tabel
    xi[i]=atoi(TabelPar[1]);
    yi[i]=atoi(TabelPar[2]);
    //latime,inaltime (in pixeli) ale unei celule
    xinc[i]=atoi(TabelPar[3]);
    yinc[i]=atoi(TabelPar[4]);
//coloane
strcpy(cod,"COL");
strcat(cod,cx);
find_line(fh,numew,cod,Lin);
copy_par(Lin,MAXPAR,TabelPar,Result);
if (Result) {
    MessageBox("Coloane tabel in arcview.ini","Eroare!",MB_ICONSTOP);
    tclose(fh);
    return -1;
}
}

```

```

//numar coloane
nrcol[i]=ini_integer(TabelPar[1],Result);
if (nrcol[i]>20) nrcol[i]=20; //maxim 20 coloane
//id-uri marimi de pe coloane
for (j=0; j<nrcol[i]; j++)
strcpy(idsi[j],TabelPar[2+j]);
if (autobitmap) //denumirile capetelor de coloana
{
strcpy(cod,"CAPTABEL");
strcat(cod,cx);
if (find_line(fh,numew,cod,Lin))
{
copy_par(Lin,MAXPAR,TabelPar,Result);
if (Result) {
MessageBox("Nu sint specificate capetele de coloana!!!",
"Eroare!",MB_ICONSTOP);
tclose(fh);
return -1;
}
}
//numele capetelor de coloane
for (j=0; j<nrcol[i]; j++)
strcpy(capcoloana[j],change_(TabelPar[2+j]));
}
else
{
MessageBox("Nu sint specificate denumirile coloanelor!", "Eroare ...",MB_ICONSTOP);
tclose(fh);
return -1;
}
}
//linii
strcpy(cod,"LIN");
strcat(cod,cx);
find_line(fh,numew,cod,Lin);
copy_par(Lin,MAXPAR,TabelPar,Result);
if (Result) {
MessageBox("Linii tabel in arcview.ini","Eroare!",MB_ICONSTOP);
tclose(fh);
return -1;
}
//numar linii
nrlin[i]=ini_integer(TabelPar[1],Result);
if (nrlin[i]>50) nrlin[i]=50; //maxim 50 linii
//tipuri valori de pe linii
int inceput,sfarsit;
char *p;
for (n=j=0; j<nrlin[i]; n++) {
if (NULL!=(p=strchr(TabelPar[2+n],'-')) {
*p='\0';
if (0!=(inceput=atoi(TabelPar[2+n]+1)))
if (0!=(sfarsit=atoi(p+1)))
if (sfarsit>=inceput) {
for (m=0; m<sfarsit-inceput+1; m++) {
vls[j][0]=TabelPar[2+n][0];
vls[j][1]='\0';
strcat(vls[j],itoa(inceput+m,cx,10));
j++;
}
continue;
}
}
}
}
}
}

```

```

strcpy(vls[j],TabelPar[2+n]);
j++;
}
for(m=0;m<nrcol[i];m++)
for(n=0;n<nrlin[i];n++) {
Cels[nrId].rct.left=xi[i]+m*xinc[i];
Cels[nrId].rct.right=xi[i]+(m+1)*xinc[i];
Cels[nrId].rct.top=yi[i]+n*yinc[i];
Cels[nrId].rct.bottom=yi[i]+(n+1)*yinc[i];
if (idsi[m][0]!='<') {
Cels[nrId].tip='s'; //string
strcpy(Cels[nrId].buffer,idsi[m]+1); //fara '<'
Cels[nrId].buffer[strlen(Cels[nrId].buffer)-1]='\0'; //fara '>'
}
else //adaugata pt. calcule pe coloana
if ((idsi[m][0]!='c') || (idsi[m][0]!='('))
{
Cels[nrId].id.echidx=i; //pentru determinare start
Cels[nrId].tip='='; //formula
Cels[nrId].req=n;
strcpy(Cels[nrId].buffer,idsi[m]);
nrId++;
//valorile sint de forma c1+c2 unde c1 si c2 reprezinta coloana 1 respectiv 2
}
else {
Cels[nrId].id=ini_id(idsi[m],Result);
if (Result) {
MessageBox("Id raport in arcview.ini","Eroare!",MB_ICONSTOP);
fclose(fh);
return -1;
}
switch(vls[n][0]) {
case 'n': // nume marime
Cels[nrId].tip='n';
break;
case 'u': // unitate masura
Cels[nrId].tip='u';
break;
case 'v':
Cels[nrId].req=atoi(vls[n]+1);
if ( ((TipRap=='Z') && ((Cels[nrId].req<0) || (Cels[nrId].req>24))) ||
((TipRap=='L') && ((Cels[nrId].req<0) || (Cels[nrId].req>31))) )
Cels[nrId].tip='x';
else
Cels[nrId].tip='v';
break;
case 'm':
Cels[nrId].req=atoi(vls[n]+1);
if ( ((TipRap=='Z') && ((Cels[nrId].req<0) || (Cels[nrId].req>24))) ||
((TipRap=='L') && ((Cels[nrId].req<0) || (Cels[nrId].req>31))) )
Cels[nrId].tip='x';
else
Cels[nrId].tip='m';
break;
case 'M':
Cels[nrId].req=atoi(vls[n]+1);
if ( ((TipRap=='Z') && ((Cels[nrId].req<0) || (Cels[nrId].req>24))) ||
((TipRap=='L') && ((Cels[nrId].req<0) || (Cels[nrId].req>31))) )

```



```

        case 'M':
            strcpy(text, "M");
            break;
        case 'v':
        case 'f':
            if (Cels[i].req==0)
                strcpy(text, "T");
            else
                if (tiprap0=='Z') //pe zi
                    sprintf(text, "%02d-%02d", Cels[i].req-
1, Cels[i].req);
                else
                    sprintf(text, "%02d", Cels[i].req);
        }
    dc->DrawText(text, -1, &r, DT_CENTER | DT_VCENTER | DT_SINGLELINE);
}

dc->SetTextColor(RGB(255,0,0));
//desenez titlurile coloanelor
r.top=0; r.bottom=yi0;
for (i=0; i<nrcol0; i++)
{
    r.left=xi0+i*xinc0;
    r.right=r.left+xinc0;
    dc->DrawText(capcoloana[i], -1, &r, DT_CENTER | DT_VCENTER |
DT_SINGLELINE);
}
dc->SelectObject(oldFont);
//partea de desenare efectiva a liniilor
capete[0].x=capete[0].y=capete[1].y=0;
capete[1].x=nWidth;
dc->Polyline(&capete[0], 2); //linia din cap
//deseneaza liniile
for (i=0; i<nrlin0+1; i++)
{
    capete[0].x=0;
    capete[0].y=capete[1].y=yi0+i*yinc0;
    capete[1].x=nWidth;
    dc->Polyline(&capete[0], 2);
}
capete[0].x=capete[1].x=capete[0].y=0;
capete[1].y=nHeight;
dc->Polyline(&capete[0], 2); //linia din stanga
//deseneaza coloanele
for (i=0; i<nrcol0+1; i++)
{
    capete[0].y=0;
    capete[0].x=capete[1].x=xi0+i*xinc0;
    capete[1].y=nHeight;
    dc->Polyline(&capete[0], 2);
}
dc->SelectObject(oldPen);
dc->SetTextColor(color);
}
// -----
// functii ajutatoare pentru implementarea unei liste cu alarme
//
CListAlm::CListAlm()

```

```

{
lstm=NULL;
act=0;
}
CListAlm::~~CListAlm()
{
    lalarm *al;
    al=lstm;
    while(al!=NULL){
        lstm=al;
        al=al->next;
        delete lstm;
    }
}
lalarm *CListAlm::first_msg()
{
    return(lstm);
}
lalarm *CListAlm::last_msg(int *n)
{
    lalarm *tt;
    int i=0;
    tt=lstm;
    if(tt==NULL){
        *n=i;
        return NULL;
    }
    while(tt->next!=NULL){
        i++;
        tt=tt->next;
    }
    *n=i+1;
    return tt;
}
lalarm *CListAlm::next_msg(lalarm *poz)
{
    if(poz==NULL)
        return NULL;
    return poz->next;
}
lalarm *CListAlm::prev_msg(lalarm *poz,int k)
{
    int i;
    if(poz==NULL)
        return NULL;
    for(i=0;i<k;i++)
        if(poz->ant==NULL)
            return lstm;
    if(poz->ant==NULL)
        return lstm;
    return poz->ant;
}
BOOL CListAlm::StergeId(id_type idalm)
{
    lalarm *pntr,*delalm;
    pntr=lstm;
    if((lstm->aid.echidx==idalm.echidx)&&(lstm->aid.lista==idalm.lista)){
        delalm=lstm;
    }
}

```

```

    if(delalm->blk<2){
        delalm->blk=2;
        repaint=1;
        delalm->cul=IDCUL_VERDE;
    }
    if(delalm->blk==3){
        repaint=1;
        lst=lst->next;
        if(lst!=NULL)
            lst->ant=NULL;
        repaint=1;
        delete delalm;
    }
    return TRUE;
}
while(pntr->next!=NULL){
    if((pntr->next->aid.echidx==idal.echidx)&&(pntr->next->aid.lista==idal.lista)){
        delalm=pntr->next;
        if(delalm->blk<2){
            delalm->blk=2;
            repaint=1;
            delalm->cul=IDCUL_VERDE;
        }
        if(delalm->blk==3){
            pntr->next=delalm->next;
            if(delalm->next!=NULL)
                delalm->next->ant=pntr;
            repaint=1;
            delete delalm;
        }
        return TRUE;
    }
    pntr=pntr->next;
}
return FALSE;
}
lalarm * CListAlm::InLista(id_type idalm)
{
    lalarm *pntr;
    pntr=lst;
    while(pntr!=NULL){
        if((pntr->aid.echidx==idal.echidx)&&(pntr->aid.lista==idal.lista)){
            pntr->act=act;
            return pntr;;
        }
        pntr=pntr->next;
    }
    return NULL;
}
BOOL CListAlm::AdaugaId(id_type idalm,st_val_type st,time_t tp)
{
    lalarm *pntr,*ant,*nal;
    pntr=lst;
    ant=lst;
    repaint=1;
    if(lst==NULL){
        lst=new lalarm;
        if(lst==NULL)

```



```

    return FALSE;
    lst->aid=idalm;
    lst->timp=tp;
    lst->act=act;
    lst->blk=0;
    mesaj_alm(lst,st);
    lst->next=NULL;
    lst->ant=NULL;
    return TRUE;
}
while(pntr!=NULL){
    if(tp<pntr->timp){
        nalm=new lalarm;
        if(nalm==NULL)
            return FALSE;
        nalm->aid=idalm;
        nalm->timp=tp;
        nalm->act=act;
        nalm->blk=0;
        mesaj_alm(nalm,st);
        nalm->next=pntr;
        nalm->ant=pntr->ant;
        pntr->ant=nalm;
        if(pntr==lst)
            lst=nalm;
        else
            nalm->ant->next=nalm;
        return TRUE;
    }
    ant=pntr;
    pntr=pntr->next;
}
pntr=new lalarm;
if(pntr==NULL)
    return FALSE;
pntr->aid=idalm;
pntr->timp=tp;
pntr->act=act;
pntr->blk=0;
mesaj_alm(pntr,st);
pntr->next=NULL;
pntr->ant=ant;
ant->next=pntr;
return TRUE;
}
BOOL CListAlm::mesaj_alm(lalarm *alm,st_val_type st)
{
    char name[50];
    char nmc[30];
    genbd_type *ap;
    id_type idalm;
    time_t t;
    struct tm cev;
    float val;

    idalm=alm->aid;
    t=alm->timp;
    if((ap=pWBDat->atm_find_off(idalm))==NULL){

```

```

    strcpy(alm->amsg,"ALARMA PENTRU O MARIME NECUNOSCUTA!");
    return FALSE;
}
cev=*localtime(&t);
sprintf(nmc,"%s %02d %s, %02d:%02d  ",Zi[cev.tm_wday],
    cev.tm_mday,Luni[cev.tm_mon],cev.tm_hour,cev.tm_min);
strcpy(alm->amsg,nmc);
switch(idalm.lista){
case AA:
case aA:
    nrmodul(ap->wa.id,nmc);
    adaugmsg(alm->amsg,ap->wa.name,20);
    switch (ap->wa.prel_inst) {
        case PI_A_LIN:
            val=(st.val-ap->wa.litr)*(ap->wa.ms-ap->wa.mi)/(ap->wa.lstr-ap->wa.litr)+ap-
>wa.mi;
            break;
        case PI_A_SQR:
            val=(float)(sqrt(st.val-ap->wa.litr)/sqrt(ap->wa.lstr-ap->wa.litr))*(ap->wa.ms-ap-
>wa.mi)+ap->wa.mi;
            break;
        default:
            val=0;
    }
    alm->cul=IDCUL_ROSU;
    sprintf(nmc," %.2f",val);
    if(st.stare!=ST_VAL_OK){
        strcpy(nmc," ??????");
        alm->cul=IDCUL_GRI;
    }
    strcpy(alm->stare,nmc);
    adaugmsg(alm->stare,ap->wa.um,12);
    break;
case DA:
case BA:
case bA:
    nrmodul(ap->wb.id,nmc);
    adaugmsg(alm->amsg,ap->wb.name,20);
    strcpy(alm->stare," ");
    switch(st.stare){
        case ST_ALR_AN:
            strcat(alm->stare,ap->wb.diag_anorm);
            alm->cul=IDCUL_ROSU;
            break;
            case ST_ALR_DM:
            strcat(alm->stare,"Defect masura");
            alm->cul=IDCUL_GRI;
            break;
    }
    break;
    if(st.stare!=ST_VAL_OK){
        strcpy(nmc," ??????");
        alm->cul=IDCUL_GRI;
    }
    strcpy(alm->stare,nmc);
    adaugmsg(alm->stare,ap->wc.um,12);
    break;
case TA:

```

```

        adaugmsg(name,ap->wt.name,20);
        sprintf(nmc," %.2f",ap->wt.value);
        alm->cul=IDCUL_ROSU;

        if(st.stare!=ST_VAL_OK){
            strcpy(nmc," ??????");
            alm->cul=IDCUL_GRI;
        }
        strcpy(alm->stare,nmc);
        adaugmsg(alm->stare,ap->wt.um,12);
        break;
    default:
        alm->ams[0]=0;
        alm->cul=IDCUL_ROSU;
        return FALSE;
    }
    return TRUE;
}
}
int CListAlm::Actualizare()
{
    act=(act+1)%500;
    return act;
}
void adaugmsg(char *dest,char *sursa,unsigned int n)
{
    unsigned int i;
    if(strlen(dest)<n)
        for(i=strlen(dest);i<n;i++)
            dest[i]=' ';
    dest[n]='\0';
    strcat(dest,sursa);
}
}

```

4.6. Concluzii

Utilizarea bazelor de date, ca parte componentă a sistemelor de monitorizare (teleurmărire) din domeniul producție și transportului de energie termică este absolut necesară, atât pentru determinarea valorilor instantanee, cât și pentru urmărirea evoluției în timp a acestora, a istoricului evenimentelor care au loc în sistem și pentru alarmare în situația producerii unor evenimente deosebite.

Bazele de date relaționale (tabelare) reprezintă soluția cea mai adecvată pentru stocarea, afișarea și manipularea valorilor diverselor mărimi întâlnite în cadrul ansamblului. Utilitarul Excel, din cadrul pachetului standard Microsoft Office constituie, cel puțin în etapa de creare, un instrument software performant pentru implementarea funcției de descriere, în cadrul. Sistemului de Gestionare a Bazei de Date (SGBD)

Pentru a asigura funcția de manipulare a bazei de date este necesară existența unui SGBD care să facă apel la funcții și proceduri generate prin intermediul unor limbaje de nivel mai înalt, ca de exemplu limbajele din familia C (Visual C, C++, Borland C etc.).

Funcția de utilizare a bazei de date trebuie implementată pornind de la ideea că dispecerul uman care supervizează sistemul nu dispune de cunoștințe vaste de programare sau utilizare baze de date, ci doar de cunoștințe generale de utilizare a calculatorului. Prin urmare aplicarea acestui concept de mediu software prietenos cu utilizatorul, („User friendly”), ne conduce spre alegerea filozofiei sistemului de operare Microsoft Windows (NT), anume dialogarea prin intermediul ferestrelor și butoanelor virtuale.

Astfel, interfața cu utilizatorul (interfața om – mașină) devine mai performantă, conducând la obținerea rezultatelor scontate în exploatare. Folosirea unor ferestre dedicate pentru dialogul om – mașină (om – bază de date) este extrem de necesară pentru prelucrarea și valorificarea informației, informație care este vitală atât pentru buna funcționare a automatizării în ansamblu dar și pentru repartizarea consumurilor, evaluarea pierderilor și facturarea corectă a energiei termice vehiculate.

4.7. Contribuții personale

Acest capitol se remarcă printr-un grad de originalitate sporit, dintre contribuțiile personale cele mai importante ale autorului trebuie evidențiate următoarele:

- Analiza critică a problemelor legate de structura bazelor de date;
- Selectarea variantelor optime pentru configurarea bazelor de date pentru diferite aplicații (realizate de către autor pentru S.C. HIDROTIM S.A.);
- Selectarea mediilor de programare adecvate aplicațiilor;
- Stabilirea, în baza cerințelor utilizatorilor, a parametrilor măsurați și introduși în baza de date, a modalității de apelare, vizualizare și manipulare, în funcție de destinația informației (automatizare, monitorizare sau contorizare);
- Conceperea unui software specializat, flexibil, care rulează în numeroase dispecerate hidrotehnice automatizate în România, una din componentele acestui pachet software fiind și un Sistem de Gestionare a Bazelor de Date;
- Elaborarea, în cadrul acestui pachet software, a unei interfețe performante om-mașină (om-bază de date), axată pe conceptul de „user friendly”, utilizând ferestre de dialog;
- Conceperea algoritmilor și a programelor în mediul Visual C, programe destinate manipulării și prelucrării datelor.

5. SISTEM SCADA IMPLEMENTAT LA ECLUZA PRINCIPALĂ ROMÂNĂ DE LA SISTEMUL HIDROENERGETIC ȘI DE NAVIGAȚIE PORȚILE DE FIER II

Creșterea siguranței în exploatare a ecluzelor prin realizarea unui sistem modern și eficient de comunicare om – mașină în vederea informării prompte și exacte a operatorului despre starea echipamentelor pe care le deservește, este o cerință tot mai frecvent întâlnită.

Lucrarea are drept scop descrierea implementării unui sistem modern și eficient de urmărire centralizată a funcționării echipamentelor și întocmirea unor rapoarte informative cu privire la activitatea de exploatare.

5.1. Prezentarea generală a echipamentelor ecluzei

Ecluza principală PdFII are o singură cameră de echilibrare nivele (SAS), echipată cu următoarele echipamente hidromecanice principale:

- Poarta plană de serviciu (PPS);
- Poarta plană de avarie (PPA);
- Vane galerii umplere (VU);
- Vane galerii golire (VG);
- Instalația de protecție a porții buscate (IPPBS);
- Poarta buscată de serviciu (PBS);

Aceste echipamente constituie elementele de execuție spre care pleacă comenzile sistemului de ecluzare.

Ecluza românească este prezentată în fig. 5.1.



Fig. 5.1. Ecluza românească de la Porțile de Fier II.

Ansamblul sistemului hidroenergetic și de navigație de la Porțile de Fier II este reprezentat în fig. 5.2.



Fig. 5.2. Sistemul Hidroenergetic și de Navigație de la Porțile de Fier II.

În afară de echipamentele hidromecanice, în componența unei ecluze mai intră o serie de instalații care prelucrează parametrii, semnale, informații fără de care procesul automat general nu poate lua decizii cu privire la desfășurarea procesului de ecluzare. Aceste instalații sunt:

- Instalația de sincronizare porți plane
- Instalația de măsură nivele
- Instalația de semaforizare
- Instalația de măsură direcție și viteză vânt
- Instalație sesizare prezență nave
- Instalația de măsură temperatură exterioară

Pentru fiecare echipament există 3 locuri de comandă:

- Din turnul de comandă
 - Din camerele electrice de la dulapurile de comandă
 - Local de la cutiile de comandă pentru fiecare echipament în parte
- Modul de comandă în cele trei cazuri poate fi după cum urmează:
Din turn:

- Automat – ciclu de ecluzare complet:
AMONTE – AVAL, AVAL – AMONTE

Din camera electrică:

- Automat – fiecare echipament în parte în ciclu complet.
- De la cutiile locale :

- Manual – fiecare secvență în parte în cadrul unei manevre a echipamentului.

Schema bloc a echipamentului de comandă este prezentată în fig. 5.3.

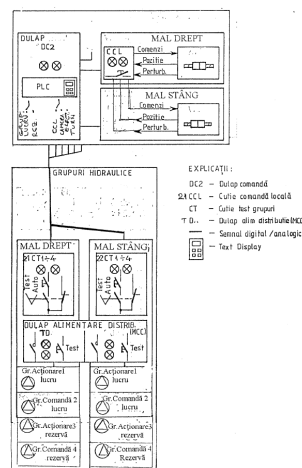


Fig. 5.3. Schema bloc a echipamentului de comandă.

5.1.1. Comanda din turn în mod automat – ciclul complet de ecluzare

În acest mod de comandă se pot desfășura două cicluri automate de ecluzare:

- Amonte – Aval – COBORÂRE
- Aval – Amonte – RIDICARE

a) Ciclul de COBORÂRE

Starea ecluzei înainte de inițierea ciclului este:

- Navă ancorată
- Luminile de intrare – ieșire amonte și aval: roșu
- IPPBS coborită
- PBS închisă
- PPS deschisă (Coborâtă)
- Vane aval închise
- Vane amonte închise
- Nu sunt nave în zona porții amonte
- Vânt în limite admise.

Pentru inițializarea oricărei comenzi este semnalizată prezența sau absența condițiilor de manevră. Ciclul de COBORÂRE se declanșează și se automenține până la dispariția condițiilor de manevră (încheierea lui). Defecțiunile care pot apărea și care întrerup ciclul automat sunt semnalizate în mod general, ele putând fi identificate la nivelul echipamentului semnalizat.

Ordinea de desfășurare a secvențelor este:

- Închidere poartă plană (ridicare), blocare și așezare pe furcă.
Poarta plană de serviciu închisă și vanele umplere închise înseamnă cap amonte închis ce creează condiții pentru secvența următoare.
- Ridicare IPPBS
- Deschiderea vanelor aval
- Golirea SAS – ului
Nivel egalizat aval creează condiții pentru deschiderea porții buscate dacă nu sunt nave în zona porții.
- Deschiderea porții buscate.
- Închiderea vanelor de golire
Vane golire închise înseamnă finalizarea ciclului de COBORÂRE.
Operatorul poate controla semaforul de ieșire aval pe lumina verde, navele fiind dirijate prin semnale de navigație spre portul aval.

b) Ciclul de RIDICARE

Starea ecluzei înainte de inițierea ciclului este:

- Navă ancorată
- Luminile de intrare – ieșire, amonte și aval: roșu
- Poarta plană închisă (ridicată)
- IPPBS ridicată
- Vanele amonte închise
- Vanele aval închise
- Poarta buscată deschisă
- Nu sunt nave în zona porții buscate
- Vânt în limite admise

Ordinea de desfășurare a secvențelor este:

- Închidere poartă buscată
Poarta buscată închisă și vanele golire închise înseamnă cap aval închis ce creează condiții pentru:

- Deschidere vane umplere
- Umplerea SAS – ului
Nivelul egalizat în amonte creează condiții pentru manevra porții plane amonte:
- Deschiderea porții plane adică ridicare de pe furcă, deblocare și coborâre poartă.
Poarta plană de serviciu coborâtă dă condiții vanelor de umplere
- Închidere vane umplere
Vane umplere închise înseamnă finalizarea ciclului de RIDICARE.
Semaforul de ieșire amonte poate fi comutat pe lumina verde, navele fiind dirijate spre portul amonte.

5.1.2. Comanda din turn în mod automat pentru fiecare echipament în parte

În acest regim de funcționare fiecare echipament va primi comenzi individuale de manevrare. Pentru aceasta, pe pupitrul de comandă din turn fiecărui echipament s-au afectat 3 butoane:

- închidere;
- deschidere;
- stop;

Trecerea de la o secvență de ecluzare la alta va fi sugerată de existența sau inexistența condițiilor de manevră care va fi semnalizată pentru fiecare echipament în parte.

5.2. Considerații generale privind sistemul de monitorizare

Structura sistemului de monitorizare este următoarea:

- Traductoarele implantate în instalație pentru culegerea parametrilor procesului;
- Sistem de achiziție care culege semnalele de la traductoare sau de la instalația de acționare și automatizare și le transmite la PC-urilor Industriale printr-o placă de achiziție (CP 1613) compatibilă cu echipamentele de automatizare existente în cadrul instalației prin intermediul PLC
- Soft achiziție și comunicație de date între sistem achiziție și PC-urile Industriale (S7 1613)
- Program monitorizare.

Sistemul de monitorizare trebuie să îndeplinească următoarele funcții:

- a) Culegerea datelor de la sistemul de achiziție la o unitate centrală de calcul (PC Industrial);
- b) Afișarea pe scheme sinoptice sau imagini dinamice a informațiilor culese;
- c) Declanșarea alarmelor la depășirea limitelor de funcționare ale parametrilor;
- d) Memorarea într-un istoric al alarmelor;
- e) Realizarea funcției de calcul sau centralizării;
- f) Realizarea funcției Trend Display, respectiv înregistrarea evoluției unor parametri;
- g) Editarea de trend-uri și rapoarte care se întocmesc zilnic, lunar etc.

Acste funcții au fost detaliate pe larg în capitolele anterioare.

În acest capitol vom descrie pe scurt numai elementele specifice aplicației de la ecluza principală română de la Porțile de Fier II.

Una din cerințele fundamentale ale sistemului de ecluzare este necesitatea afișării datelor tip imagine dinamică a procesului de ecluzare.

Datele care sunt aduse și apoi stocate în baza de date de pe PC-urile industriale amplasate în structura de comandă, iar prin intermediul sistemului de achiziție de date vor fi prelucrate cu ajutorul programului de monitorizare.

Prezentarea procesului de ecluzare afișat pe monitor în forma grafică și deservit prin mouse și tastatură reprezintă o soluție modernă și eficientă de comunicare om – mașină în vederea informării prompte și exacte a operatorului despre starea echipamentelor pe care le deservește.

Această soluție tehnică este în spiritul observațiilor efectuate în capitolele anterioare.

Ansamblul ecluză

În timpul procesului de ecluzare se va afișa pe monitor în același timp o vedere în plan a ecluzei și o secțiune longitudinală prin ecluză.

În cadrul acestor imagini se vor prezenta următoarele:

a) În formă grafică:

- Echipamentele principale în mișcare pe parcursul manevrării lor (echipamentul în mișcare va avea o imagine pâlpâitoare).
- Nivelele de apă amonte, SAS și aval în mișcare pe parcursului procesului de ecluzare.
- Existența condițiilor de manevre pentru fiecare echipament în parte – semnalele verzi pe corpul echipamentului.
- Avarierea unui echipament – colorarea în roșu a echipamentului respectiv.
- Starea luminilor de trafic – roșu sau verde

b) În stare alfanumerică

- Poziția momentană pentru fiecare echipament în parte în imediata apropiere a imaginii echipamentului.
- Nivelele de apă în amonte, SAS și aval.
- Orice alți parametri solicitați cum ar fi: presiune ulei circuit hidraulic, debit viteză vânt, temperatură etc.
- Locul de comandă:
 - Turn
 - Camera electrică
 - Local
- Mod de comandă
 - Automat – un ciclu complet de ecluzare
 - Automat – fiecare echipament în parte.
 - Manual – fiecare secvență a echipamentului în parte.
- Timpul de la inițierea unui ciclu de ecluzare.
- Date generale cum ar fi:
 - Denumire imagine: ANSAMBLU ECLUZĂ
 - Oră, minut, secundă
 - Data: zi, lună, an.

De asemenea această imagine generală a ecluzei va oferi posibilitatea ca la efectuarea cu mouse-ul a unui clic pe unul din echipamente să ofere o imagine dinamică a celui echipament.

În figurile de mai jos sunt prezentate capturi video ale imaginilor dinamice de funcționare a ecluzei.

Toate imaginile sunt preluate de la ecluzări amonte-aval.

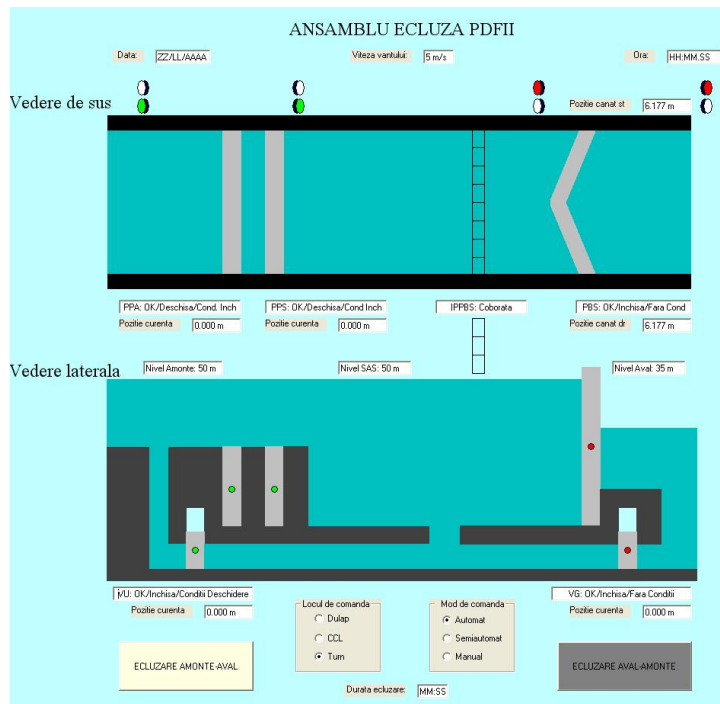


Fig. 5.4. Ecluzare. Etapa I.

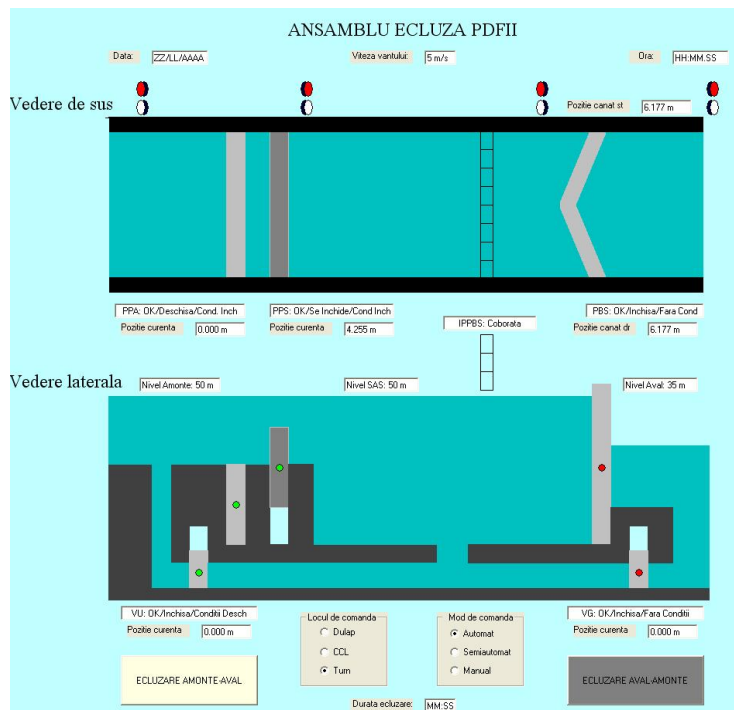


Fig. 5.5. Ecluzare. Etapa II.

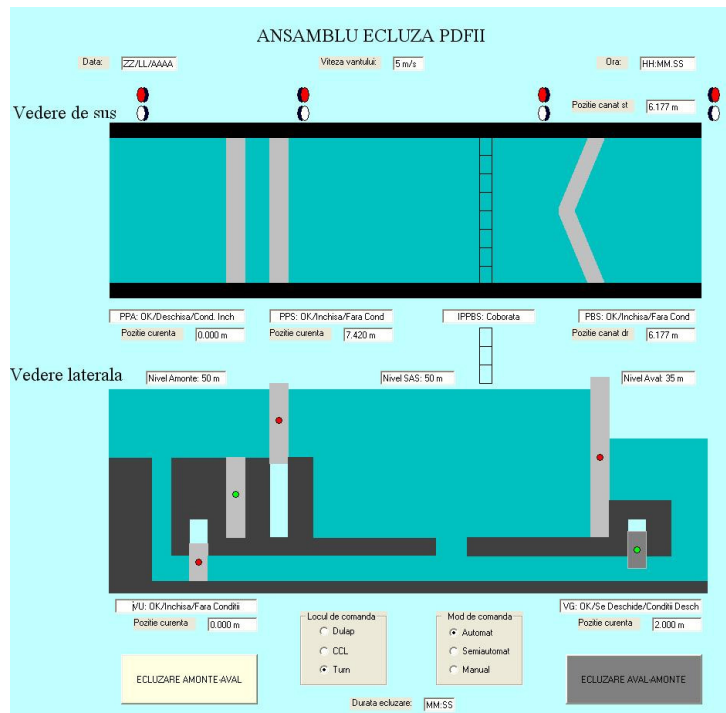


Fig. 5.6. Ecluzare. Etapa III.

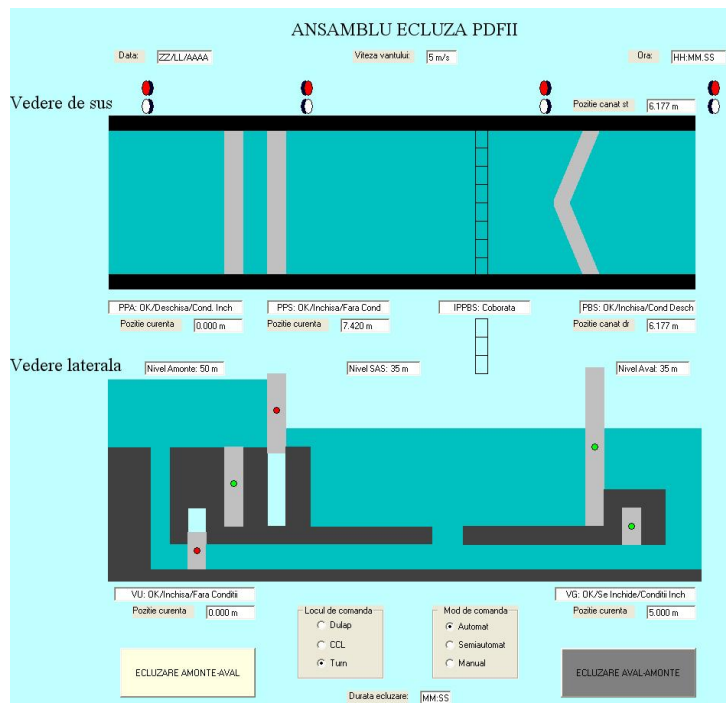


Fig. 5.7. Ecluzare. Etapa IV.

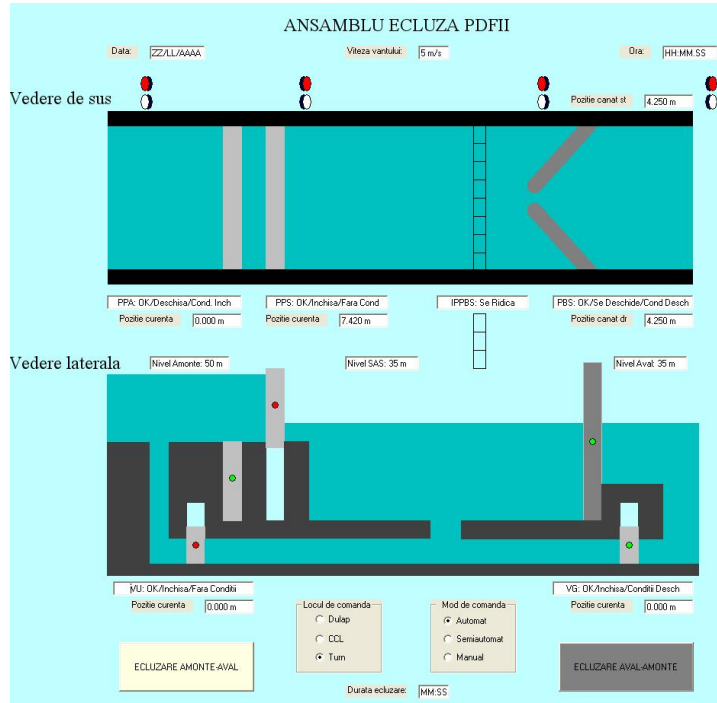


Fig. 5.8. Ecluzare. Etapa V.

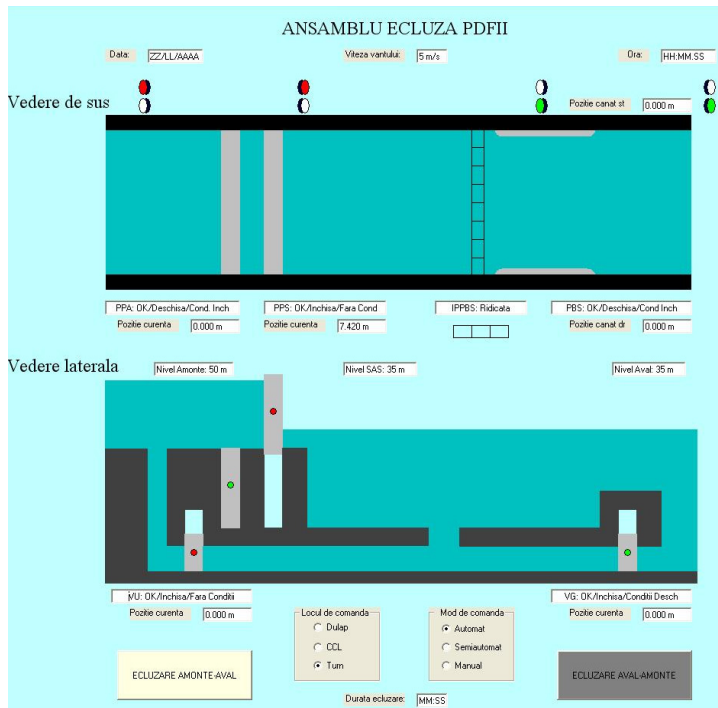


Fig. 5.9. Ecluzare. Etapa VI.

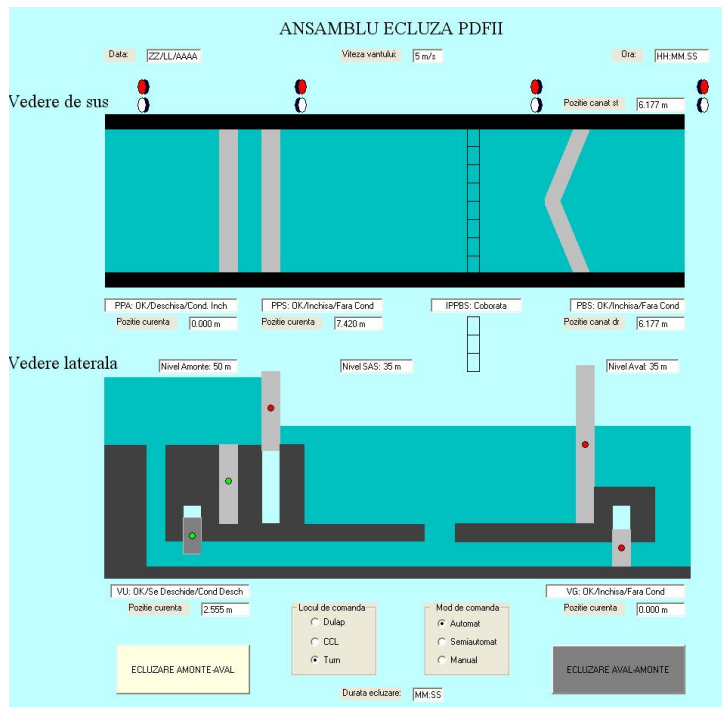


Fig. 5.10. Ecluzare. Etapa VII.

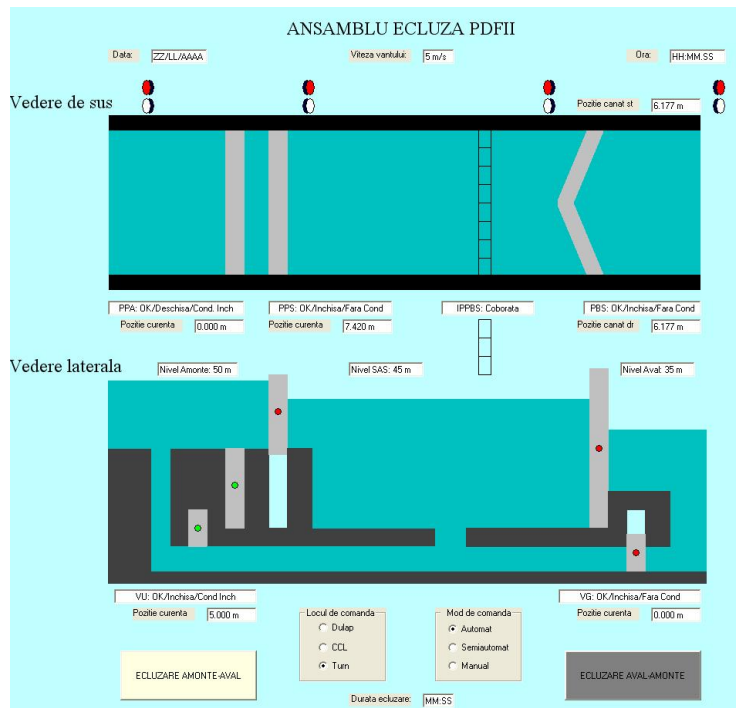


Fig. 5.11. Ecluzare. Etapa VIII.

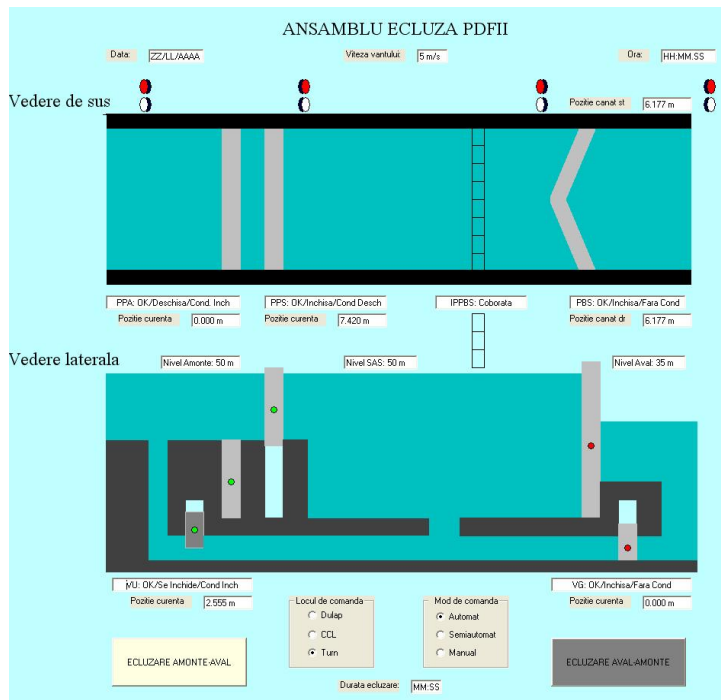


Fig. 5.12. Ecluzare. Etapa IX.

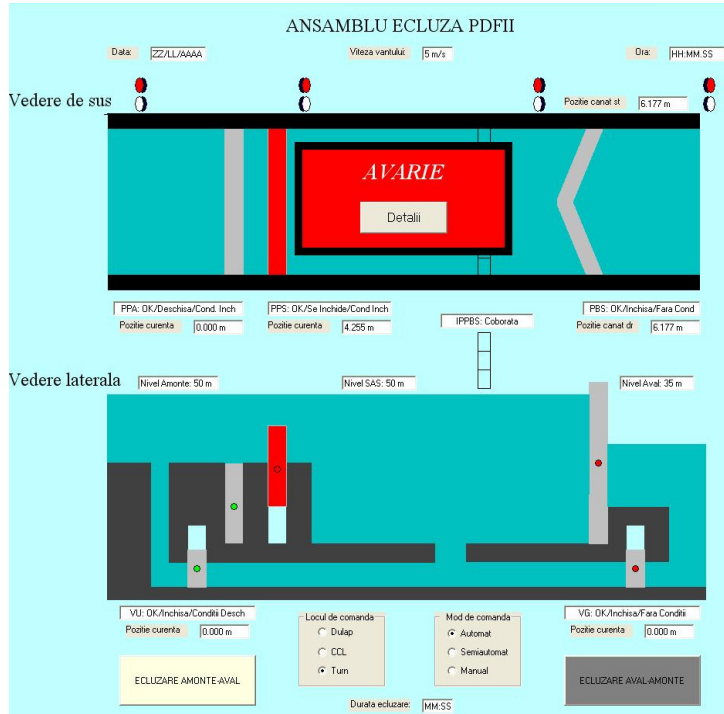


Fig. 5.13. Avarie în procesul de ecluzare.

Schema completă a sistemului de automatizare a ecluzei este prezentată în Anexa 1. În Anexa 2 este prezentată, de asemenea, interconectarea diverselor subgrupuri ale sistemului de automatizare a ecluzei.

Poarta plană

Imaginea dinamică poarta plană cuprinde o secțiune transversală a ecluzei în zona porții plane precum și o secțiune prin poartă. Imaginea prezintă următoarele:

- a) în formă grafică (dinamică):
- Condiția de manevră a porții prin luminița de culoare verde
 - Poarta plană în mișcare pe parcursul manevrării ei.
 - Cele două servomotoare de acționare mal stâng și mal drept.
 - O linie între cele două servomotoare care să arate abaterea de la orizontalitate.
 - Pe corpul porții vor apărea săgeți pâlپاitoare care vor indica sensul de deplasare (pentru fiecare mal în parte).
- b) În formă alfanumerică:
- Poziția momentană pentru fiecare mal în parte în apropierea servomotoarelor.
 - Diferența dintre cele două poziții (desincronizarea)
 - O baretă a cărei extremitate reprezintă desincronizarea maximă admisă și care prin colorare să sugereze cât reprezintă desincronizarea momentană
 - Nivelul apei amonte și aval de poartă.
 - Orice alți parametri solicitați cum ar fi: presiunea uleiului din circuitul hidraulic, nivel ulei în rezervor, etc.
 - Locul de comandă:
 - Turn
 - Camera electrică
 - Local
 - Modul de comandă:
 - Automat
 - Manual
 - Date generale:
 - Denumire imagine: POARTA PLANĂ
 - Data, ora

Schema completă de automatizare a porții plane va fi detaliată în fig. 5.15.

Poarta buscată

Imaginea dinamică poarta buscată va cuprinde o vedere în plan a porții care să fie alcătuită din următoarele componente:

- a) în formă grafică (dinamică):
- Condiția de manevră a porții prin luminile de culoare verde.
 - Cele două canate ale porții în mișcare de rotație
 - Cele două servomotoare de acționare ale canatelor (pe corpul canatelor în apropierea busculei vor apărea săgeți pâlپاitoare care să indice sensul mișcării dat fiind că viteza de rotație este mică; de asemenea în cazul în care unul dintre canate se oprește, în punctele de așteptare, această săgeată va dispărea).
- b) În formă alfanumerică
- Poziția momentană pentru fiecare canat în parte în apropierea fiecărui servomotor.
 - Diferența dintre cele două poziții.
 - Locul de comandă:
 - Turn
 - Camera electrică
 - Local de la CCL
 - Modul de comandă:
 - Automat
 - Manual

- Date generale:
 - Denumire imagine: POARTĂ BUSCATĂ
 - Data, ora.

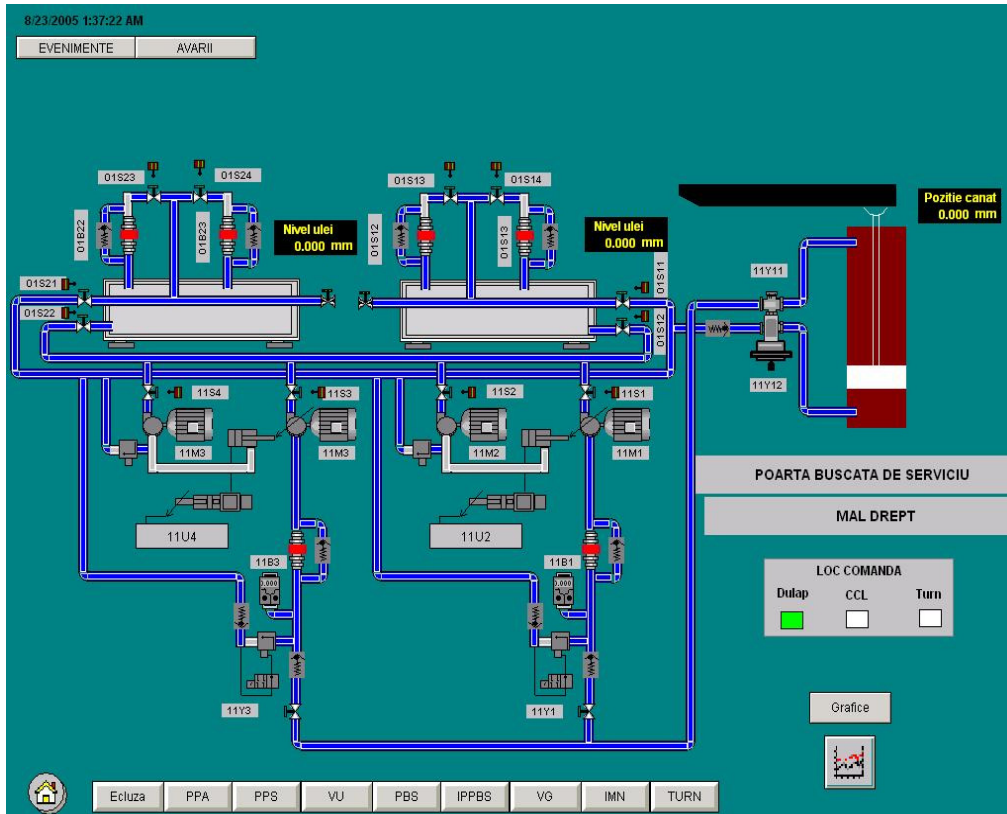


Fig. 5.14. Schema automatizării pentru poarta buscată.

Vane Galerii

Pentru fiecare vană se va crea o imagine dinamică care va cuprinde o secțiune transversală prin vană care va cuprinde următoarele:

- a) în formă grafică (dinamică):
 - condiția de manevră a porții prin luminițe de culoare verde.
 - Vana plană în mișcare în carcasa sa inclusiv servomotorul de acționare.
 - O bareță care să sugereze domeniul de tasare și care să indice prin diferență de culoare cât din domeniu a tasat vana.
- b) În formă alfanumerică:
 - poziția momentană
 - timpul scurs de la ultima atingere a poziției ridicată complet
 - media intervalului de timp dintre două repompări
 - locul de comandă:
 - turn
 - camera electrică
 - local de la CCL.

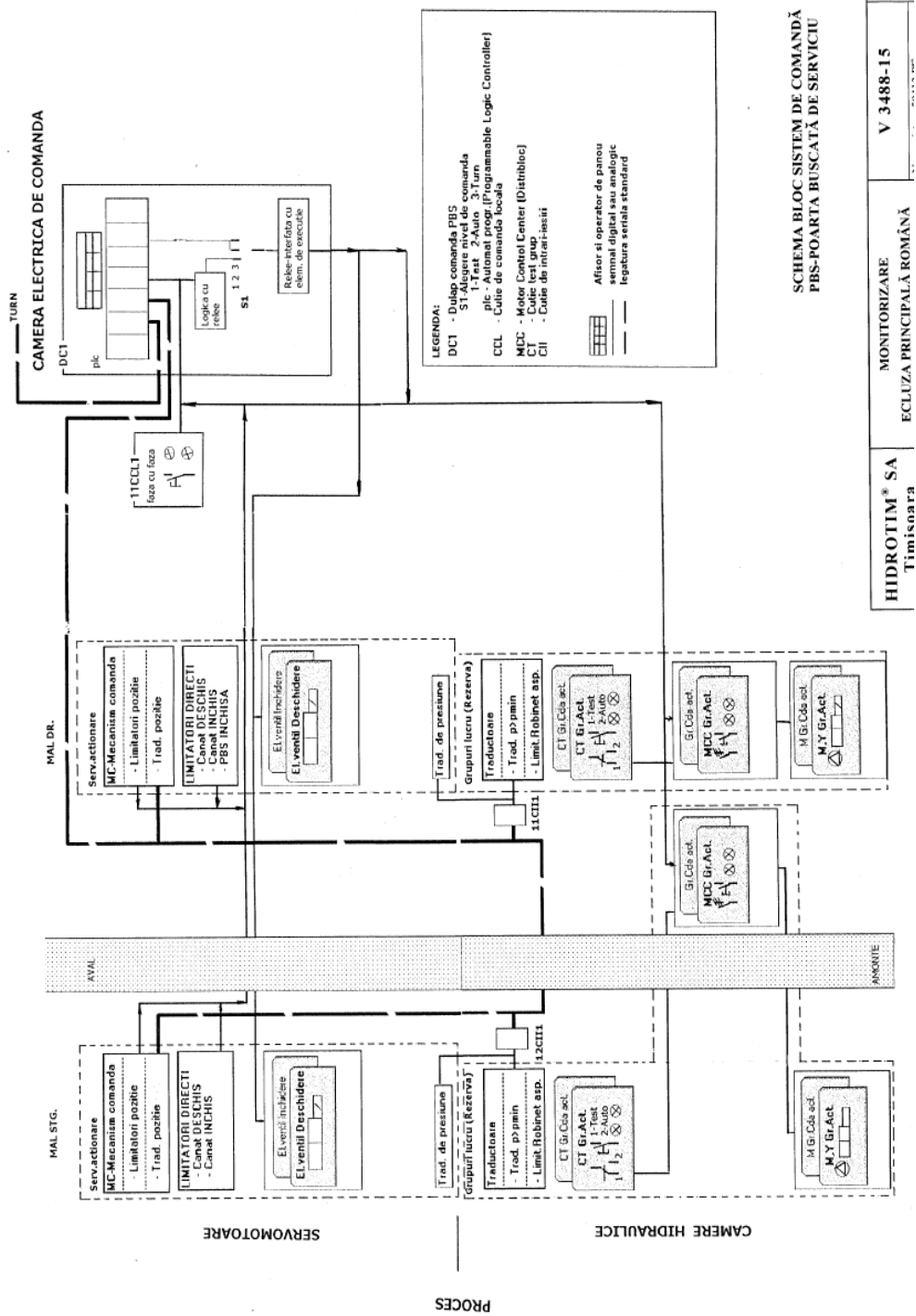


Fig. 5.15. Automatizarea pentru poarta buscăță.

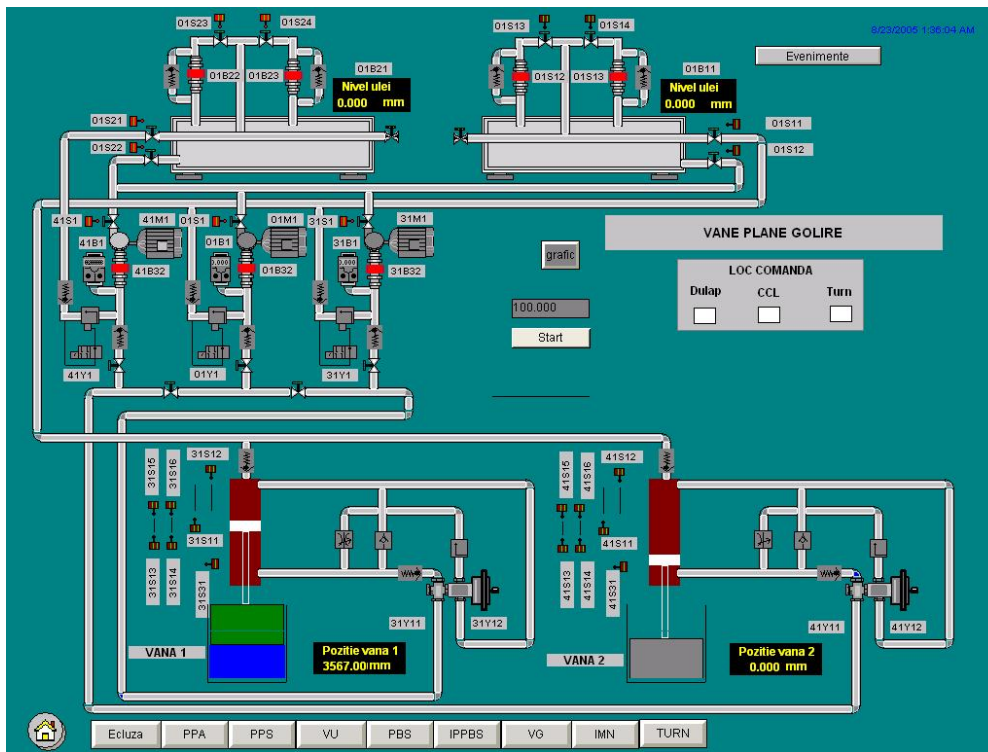


Fig. 5.16. Automatizarea vanelor galerii.

- modul de comandă:
 - automat
 - manual
- date generale:
 - denumire imagine: VANA Nr. 1 – 4
 - data, ora.

În fig. 5.14 este descrisă schema sinoptică pentru automatizarea vanelor galerii.

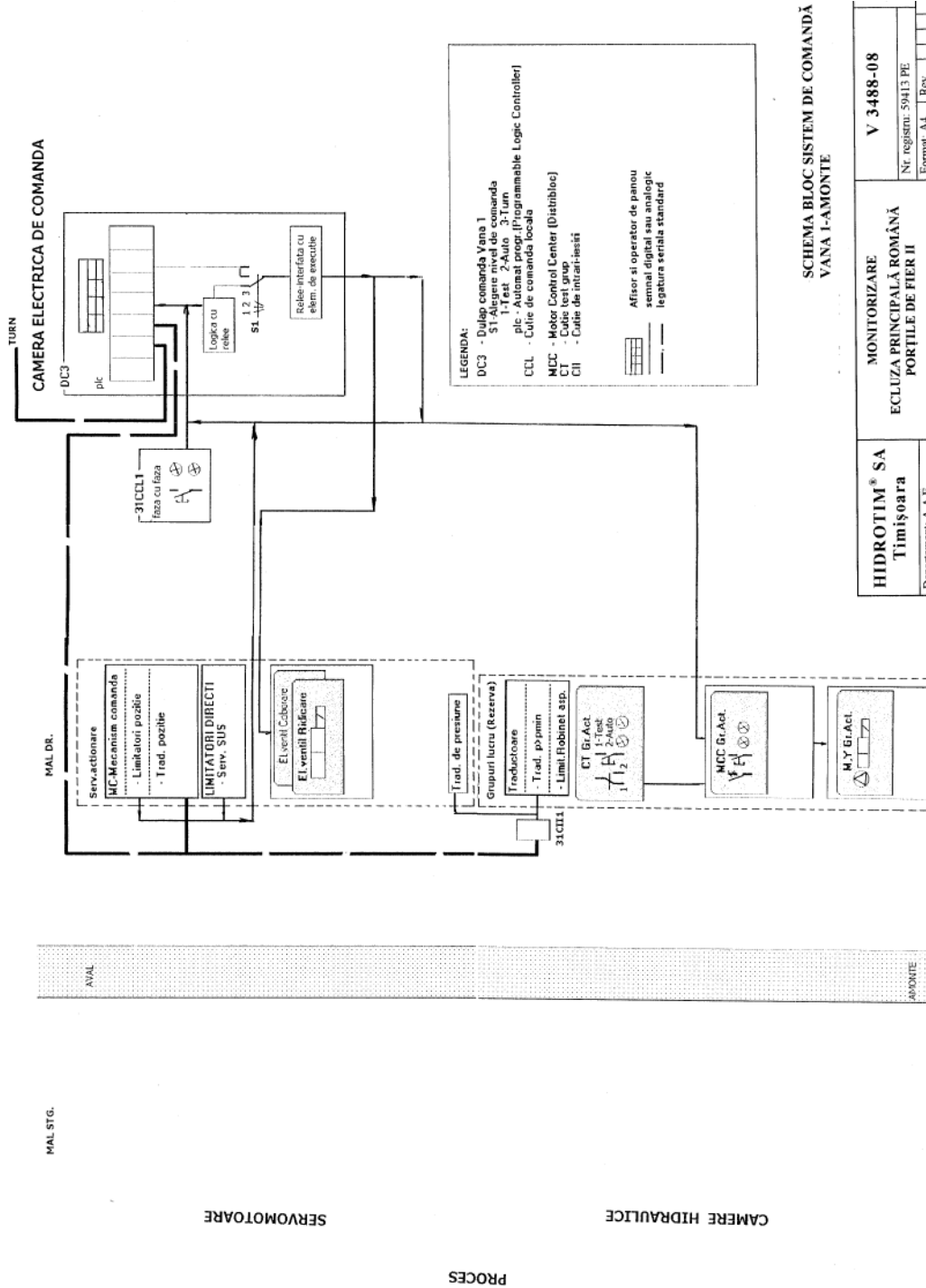


Fig. 5.17. Schema automatizare VANA 1 Amonte.

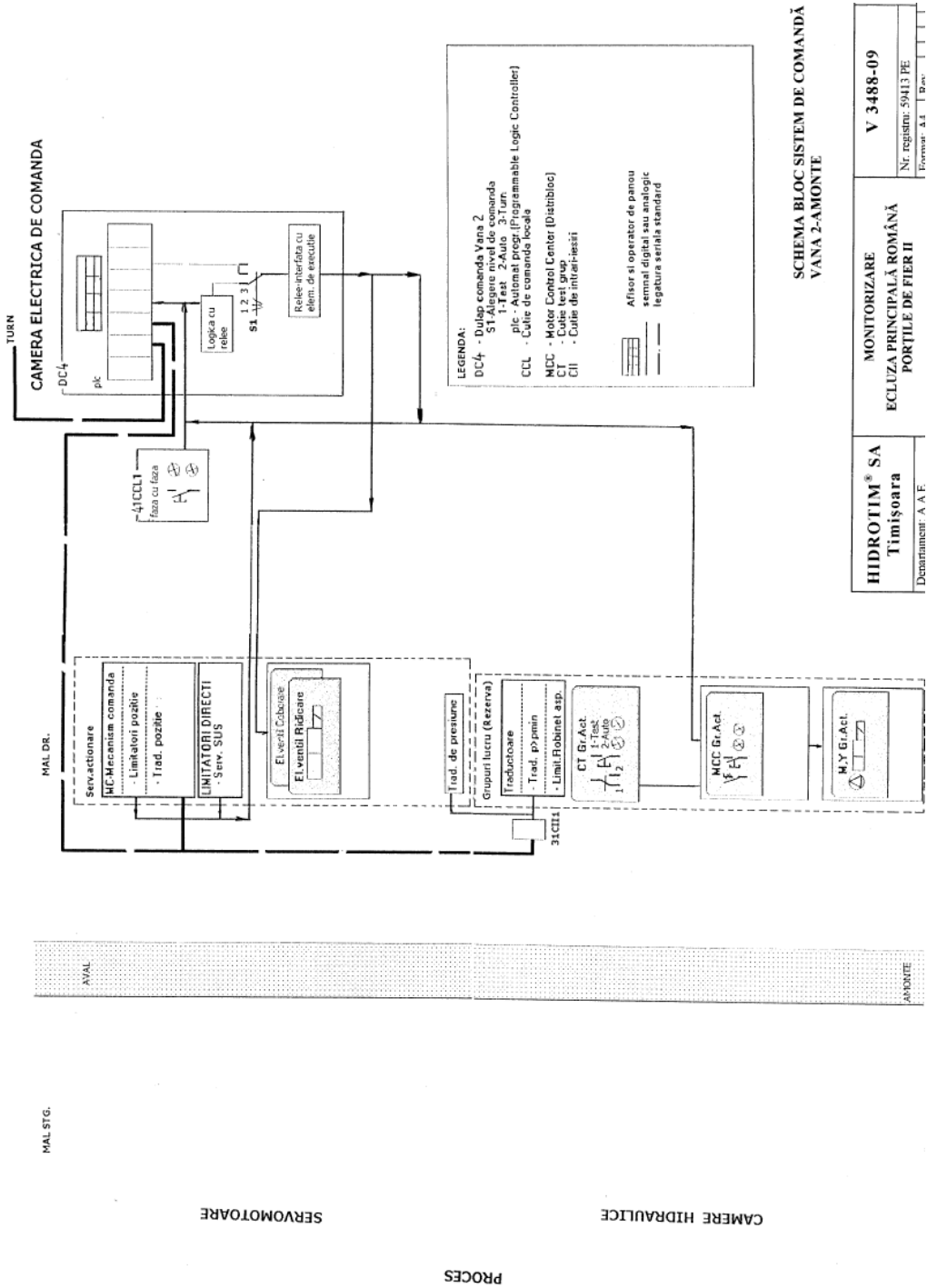


Fig. 5.18. Schema automatizare VANA 2 Amonte.

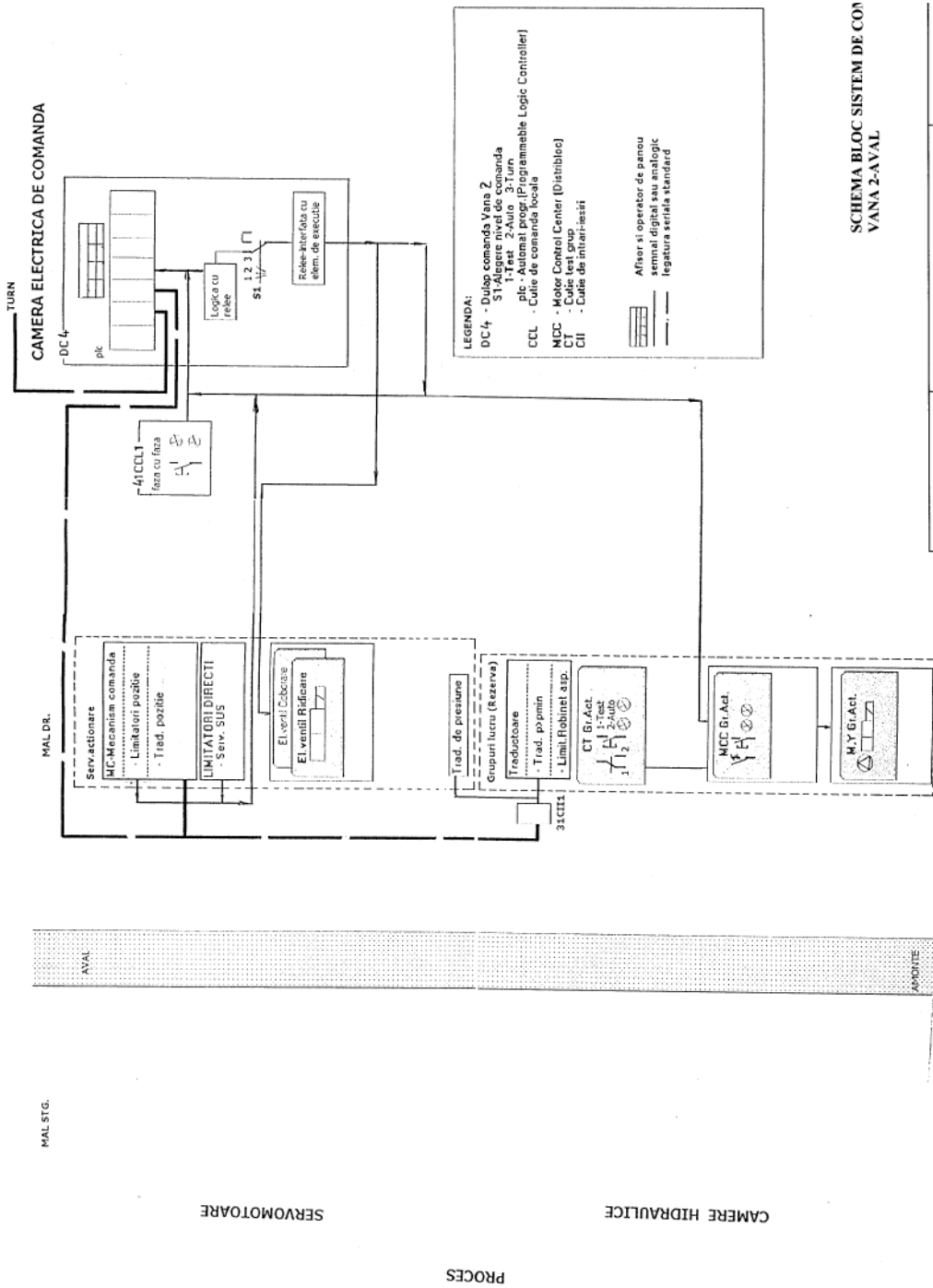


Fig. 5.20. Schema de automatizare VANA 2 Aval.

Instalație protecție poartă buscată (IPPBS)

Imaginea dinamică IPPBS va cuprinde o secțiune transversală a ecluzei în zona IPPBS precum și o secțiune prin IPPBS. Imaginea va cuprinde următoarele:

a) în formă grafică (dinamică):

- condiția de manevră a IPPBS prin luminița de culoare verde
- IPPBS în mișcare pe parcursul manevrării ei.
- O linie între cele două coloane rabatabile (CR) care să arate abaterea de la orizontalitate.
- Pe plasa IPPBS vor apărea săgeți pâlپătoare care vor indica sensul de deplasare (pentru fiecare mal în parte).
- Nava în plasă

b) în formă alfanumerică:

- poziția momentană pentru fiecare mal în parte în apropierea CR.
- Diferența dintre cele două poziții (desincronizarea)
- Orice alți parametri solicitați cum ar fi: presiunea uleiului din circuitul hidraulic, debit, nivelul uleiului în rezervorul de ulei, etc.
- Locul de comandă:
 - Turn
 - Camera electrică
 - Local
- Modul de comandă:
 - Automat
 - Manual
- Date generale:
 - Denumire imagine: IPPBS
 - Data, ora

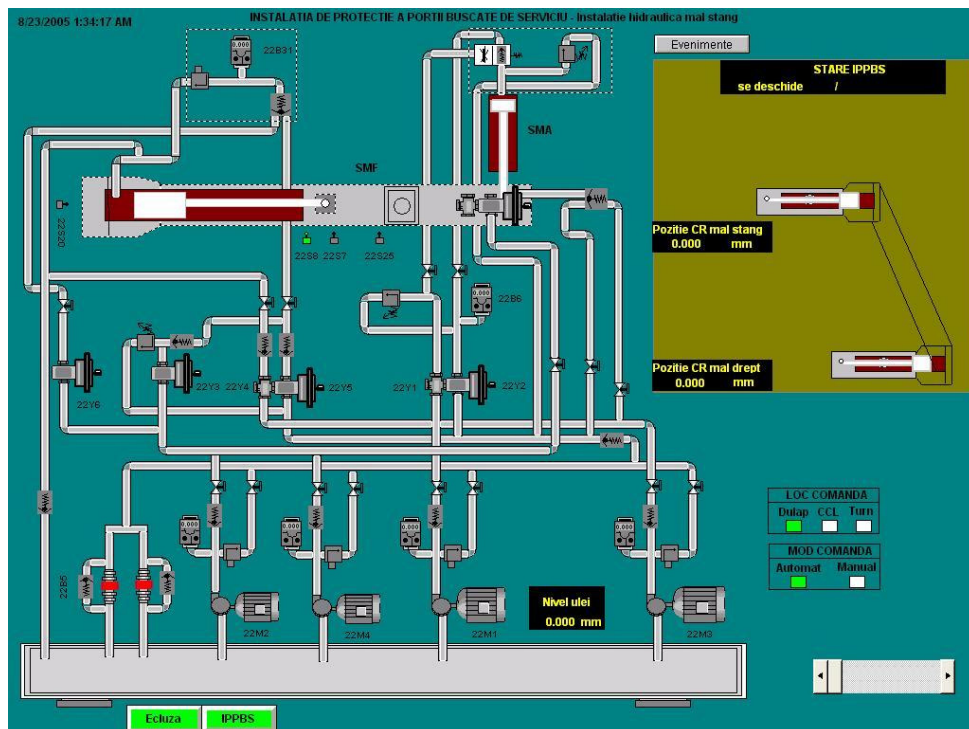


Fig.5.21. Schema sinoptică a automatizării protecției porții buscate.



Fig. 5.22. Instalația hidraulică exterioară.

5.3. Stări și semnale necesare monitorizării

Sub aspect funcțional sistemul de automatizare al ecluzei este conceput ca un sistem de AUTOMATE – MODULE FUNCȚIONALE.

Pentru o supervizare corespunzătoare a modului în care se desfășoară procesul de ecluzare este necesară cunoașterea, prelucrarea și stocarea pentru fiecare modul funcțional în parte a principalelor semnale care contribuie la derularea procesului automat.

Modulele tratate sunt:

- Procese automate generale (TURN);
- Cap ecluză (amonte/aval);
- Portă plană;
- Vane galerii umplere și golire;
- Instalație de protecție poartă buscată;
- Poartă buscată;
- Dulap general DC0 (amonte/aval);
- Tablou distribloc TD1 (amonte/aval).

Pentru fiecare modul funcțional în parte sunt specificate semnalele discrete care sunt necesare monitorizării, cum ar fi:

- Mărimi (variabile) de stare;
- Poziție de capăt de cursă, sau anumite poziții intermediare;
- Validitate pentru modulele subordonate, de exemplu grupurile hidraulice;
- Avarii;
- Chei și butoane de comandă;
- Condiții de manevră;
- Comenzi spre sau dinspre alte module, inferioare sau superioare din punct de vedere al poziției lor în sistemul ierarhizat de automatizare;

În afară de aceste semnale discrete sunt urmărite și mărimi cu variație continuă. Aceste mărimi sunt următoarele:

în cadrul modului general:

- Nivel apă amonte, SAS, aval;
- Viteză, direcție vânt;
- Temperatura mediului ambiant;

pentru fiecare echipament hidromecanic în parte se urmărește:

- Poziția servomotoarelor;
- Presiunea uleiului în circuitul hidraulic;
- Nivelul uleiului în rezervoarele hidraulice.

Dulapul instalației de automatizare este prezentat în fig. 5.23.

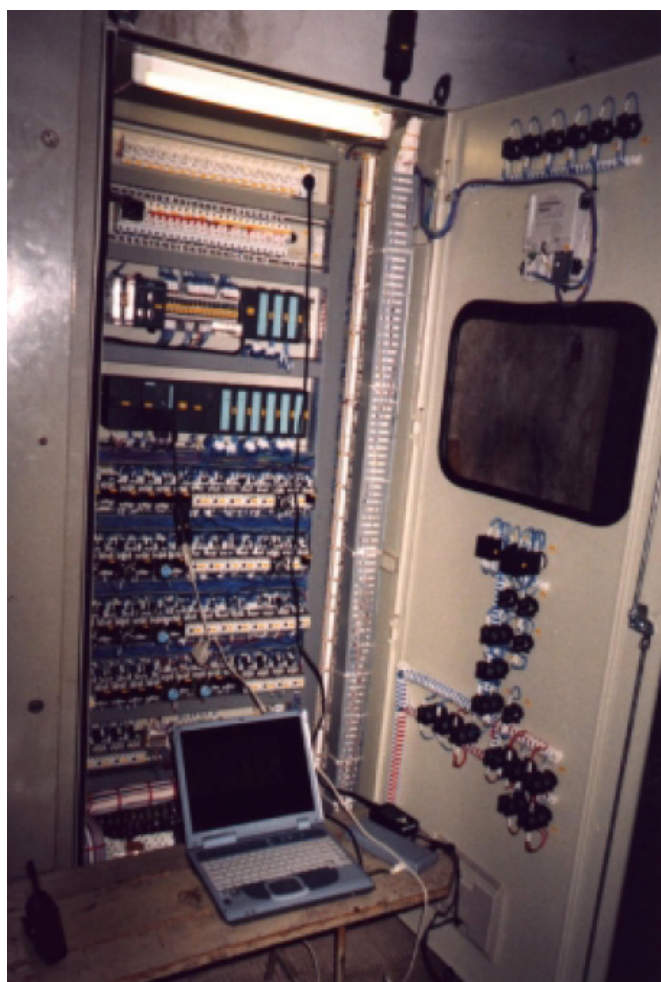


Fig. 5.23. Dulapul instalației de automatizare.

Stări și semnale preluate de la modulul procese automate generale (turn)

Stările procesului de ecluzare automată:

- C1 – Ciclul ecluzare RIDICARE în derulare.

- C2 – Ciclul ecluzare COBORÂRE în derulare.
Pașii ciclului de ecluzare automată RIDICARE:
- C3 – Toate luminile de semaforizare intrare – ieșire sunt pe roșu.
- C4 – Închiderea poartă buscată.
- C5 – Deschidere vane amonte.
- C6 – Deschidere poartă plană de serviciu și închidere instalație de protecție poartă buscată.
- C7 – Închidere vane amonte.
- C8 – Lumini de navigație ieșire amonte – verde.
- C9 – Sfârșit secvență automată de RIDICARE.
Pașii ciclului de ecluzare automată COBORÂRE:
- C10 – Toate luminile de semaforizare intrare – ieșire sunt pe roșu.
- C11 – Închidere poartă plană de serviciu.
- C12 – Deschidere vane aval.
- C13 – Deschidere poartă buscată și deschidere instalație de protecție poartă buscată.
- C14 – Închidere vane aval.
- C15 – Lumini de navigație ieșire aval – verde.
- C16 – Sfârșit secvență de COBORÂRE.

Celelalte mărimi SCADA care vor fi preluate de la automatul programabil al turnului sunt prezentate în tabelul anexat.

Stări și semnale discrete preluate de la poarta plană (PPS, PPA)

De la modulul POARTĂ PLANĂ se preiau următoarele stări:

- C1 – AVARIE poartă plană.
- C2 – Poarta plană SE DESCHIDE.
- C3 – Poarta plană SE ÎNCHIDE.

Situațiile de manevră fază cu fază sunt caracterizate de asemenea de câte o variabilă de stare proprie. Aceste variabile de stare, necesare să fie preluate în vederea monitorizării PORȚII PLANE, sunt următoarele:

- C4 – Poarta plană SE RIDICĂ.
- C5 – Poarta plană SE COBOARĂ.
- C6 – Poarta plană SE COBOARĂ PE FURCĂ.
- C7 – Poarta plană SE BLOCHEAZĂ.
- C8 – Poarta plană SE DEBLOCHEAZĂ.

Semnalele de validare a grupurilor hidraulice: POARTA PLANĂ este deservită de opt grupuri hidraulice de lucru și de opt grupuri hidraulice de rezervă. De la fiecare grup hidraulic se preia semnalul discret: grup VALID.

Celelalte mărimi SCADA care vor fi preluate de la automatele programabile aferente porților plane sunt prezentate în tabelul anexat.

Stări și semnale discrete preluate de la vane (VPG)

Fiecare cap al ecluzei este echipat cu câte două vane galerie de umplere a SAS-ului ecluzei, respectiv de golire a SAS-ului. Întrucât modulele de automatizare sunt identice, de la fiecare vană se preiau același stări prezentate în continuare pentru o singură vană:

Stările vanei:

- C1 – AVARIE vană.
- C2 – Vana SE DESCHIDE.
- C3 – Vana SE ÎNCHIDE.
- C4 – vana SE MENȚINE DESCHISĂ.

Semnale de validitate a grupurilor hidraulice: vana este deservită de două grupuri hidraulice, unul de lucru și unul de rezervă. De la fiecare grup hidraulic se preia semnalul discret: grup VALID.

Celelalte mărimi SCADA care vor fi preluate de la automatele programabile aferent vanelor plane sunt prezentate în tabelul V40766-24.

Stări și semnale discrete preluate de la poarta buscată (PBS)

De la modulul poartă buscată se preiau următoarele stări :

- C1 – AVARIE poartă buscată.
- C2 – Poarta buscată SE DESCHIDE.
- C3 – Poarta buscată SE ÎNCHIDE.

Semnale de validitate a grupurilor hidraulice: poarta buscată este deservită de două grupuri hidraulice de lucru și de două grupuri hidraulice de rezervă. De la fiecare grup hidraulic se preia semnalul discret: grup VALID.

Celelalte mărimi SCADA care vor fi preluate de la automatul programabil aferent PBS sunt prezentate în tabelul anexat.

Stări și semnale discrete preluate de la instalația de protecție a porții buscate (IPPBS)

De la modulul Instalației de protecție poartă buscată se preiau următoarele semnale:

Stările instalației de protecție poartă buscată:

- C1 – AVARIE instalație de protecție poartă buscată.
- C2 – Instalația de protecție poartă buscată SE DESCHIDE.
- C3 – Instalația de protecție poartă buscată SE ÎNCHIDE.

Situațiile de manevră fază cu fază sunt caracterizate de asemenea de câte o variabilă de stare proprie. Aceste variabile de stare, necesar să fie preluate în vederea monitorizării instalației de protecție poartă buscată, sunt următoarele:

- C4 – Instalația de protecție poartă buscată SE RIDICĂ.
- C5 – Instalația de protecție poartă buscată SE COBOARĂ.
- C6 – Instalația de protecție poartă buscată SE TENSIONEAZĂ.
- C7 – Instalația de protecție poartă buscată SE DETENSIONEAZĂ.

Situațiile de automenținere sunt caracterizate și ele de câte o variabilă de stare proprie. Aceste variabile de stare, necesar să fie preluate în vederea monitorizării instalației de protecție poartă buscată, sunt următoarele:

- C8 – Instalația de protecție poartă buscată SE AUTOMENȚINE TENSIONATĂ SUS.
- C9 – Instalația de protecție poartă buscată SE AUTOMENȚINE TENSIONATĂ JOS.

Semnale de validare a grupurilor hidraulice: instalația de protecție poartă buscată este deservită de patru grupuri hidraulice de lucru și de patru grupuri hidraulice de rezervă. De la fiecare grup hidraulic se preia semnalul discret: grup VALID.

Celelalte mărimi SCADA care vor fi preluate de la de la automatul programabil aferent IPPBS sunt prezentate în tabelul anexat.

Stări și semnale discrete preluate de la modulul cap (DC0)

În cadrul structurii ierarhizate a sistemului de conducere a unei ecluze, nivelul CAP este un nivel de automatizare situat între nivelul TURN și nivelul ECHIPAMENT HIDROMECHANIC. La nivelul CAP distingem două module funcționale de același rang: CAP AMONTE și CAP AVAL. Întrucât cele două module sunt relativ similare, în cele ce urmează sunt detaliate principalele semnale culese de la modulul CAP AMONTE.

Semnalele de STOP de avarie:

- C1 – Stop de avarie din camera electrică.
- C2 – Stop de avarie de pe bajoaier mal stâng.

- C3 – Stop de avarie de pe bajoaier mal drept.
- C4 – Stop de avarie de la vana 1.
- C5 – Stop de avarie de la vana 2.

Semnalele de la chei și butoane de comandă:

- C6 – Alegere loc de comandă.
- C7 – Loc de comandă: TURN.
- C8 – Loc de comandă: CAMERA ELECTRICĂ.
- C9 – Loc de comandă: Pe bajoaier.

Semnalele de sesizare prezenta nave:

- C10 – aval
- C11 – amonte

Stării și semnale preluate de la tabloul distribloc (TD1)

De la modulul tablou distribloc se preiau următoarele semnale:

Stări ale automatului de anclanșare rapida :

- C1 AAR funcționează
- C2 AAR a lucrat
- C3 AAR indisponibil

Celelalte mărimi SCADA vor fi preluate de la de la automatul programabil aferent TD1.

Algoritmi de filtrare semnale

Valorile prelucrate de către PC vor fi prelucrate după diverși algoritmi urmând ca rezultatele acestor prelucrări să fie înmagazinate în baza de date în vederea utilizării ulterioare de către pachetul de programe.

Prelucrarea datelor achiziționate la fiecare citire este necesară pentru evitarea stocării unui volum inutil de mare de informații care să ducă la depășirea capacității de stocare în scurt timp.

Algoritmii de prelucrare a datelor achiziționate în vederea memorării lor diferă de la un tip de mărime la altul după cum urmează:

- a) Pentru mărimile cu variație continuă setul de date ce sunt citite la un moment dat vor fi memorate numai dacă cel puțin unul dintre ele diferă de cel citit anterior cu o valoare ce reprezintă rezoluția cu care se dorește stocarea evoluției în timp a parametrilor. Programul de prelucrare va oferi posibilitatea modificării acestor rezoluții în vederea optimizării raportului între acuratețea analizării procesului de manevrare a echipamentelor hidromecanice.
- b) Pentru semnalele discrete culese din instalație rezultă un vector de stare al fiecărui echipament în parte alcătuit din n biți, vector ce va fi memorat împreună cu momentul citirii numai dacă diferă de vectorul citit anterior prin cel puțin un bit. Pentru fiecare vector se va crea în baza de date un tabel cu corespondența dintre fiecare câmp al vectorului și semnificația tehnologică a semnalului recepționat în acel câmp (bit). La depășirea limitelor de funcționare ale parametrilor, semnalele de alarmă vor fi memorate împreună cu momentul apariției lor și cu toți ceilalți parametri citiți indiferent de algoritmi prezențați anterior.

Ciclul de citire

Procesele monitorizate prin sistemul de achiziție SCADA se împart (după durata ciclului de achiziție) în două mari categorii:

- procese rapide (1 secundă)
- procese lente (1 minut)

În categoria proceselor rapide intră monitorizarea pozițiilor echipamentelor, valorile presiunilor și nivelele de ulei din instalații. În categoria proceselor lente se află restul proceselor monitorizate prin sistemul SCADA .

În anexele acestei lucrări vor fi prezentate schemele detaliate ale tuturor instalațiilor realizate în cadrul aplicației de la Porțile de Fier II.

5.4. Concluzii

Modernizarea și îmbunătățirea sistemelor hidroenergetice de navigație și transport este o necesitate dictată de creșterea siguranței în exploatarea acestora, dar și de strategia energetică a României. Modernizarea acestor activități presupune, pe lângă investiții mari în infrastructură, stabilirea unor algoritmi corecți pentru funcționarea sistemului.

Creșterea randamentului întregului ansamblu este garantată în primul rând de cunoașterea tuturor datelor și parametrilor care intervin în funcționarea ansamblului. Acest obiectiv poate fi atins prin implementarea unor sisteme de telemanagement informatizat la nivel de sistem hidroenergetic și de navigație. Această modernizare presupune monitorizarea în timp real a tuturor informațiilor de interes din sistem, controlul unor parametri de sistem și gestionarea acestor informații la toate nivelele.

S.C. HIDROTIM S.A. este una dintre firmele care dispun de o vastă experiență în acest domeniu, autorul fiind direct implicat în elaborarea a numeroase sisteme de automatizare, măsură și control.

Una dintre particularitățile esențiale ale soluțiilor tehnice descrise în acest capitol constă în implementarea unor sisteme de tip SCADA bazate pe calculatoare industriale de proces, ceea ce conferă o robustețe și o fiabilitate sporită a sistemului în comparație cu soluțiile tehnice utilizate până în prezent, care se bazează pe calculatoare personale ce înglobează diverse plăci de achiziție sau diverse alte interfețe. Sistemul hidrotehnic este văzut ca un proces industrial complex, a cărui funcționare trebuie monitorizată și controlată.

Soluția unui sistem SCADA dispus pe mai multe nivele ierarhice este o altă particularitate a soluțiilor tehnice descrise până acum. Sistemul câștigă astfel o mai mare fiabilitate, o modularizare sporită, dar și flexibilitate în exploatare.

Simplificarea interfeței OM - MAȘINĂ (interfața dintre utilizator și sistem) este una din caracteristicile fundamentale ale acestor aplicații. Dezvoltarea software-ului dedicat a condus la realizarea unui dialog eficient între utilizatorul uman și sistemul implementat. Conceptul modern de „user friendly - sistem prietenos cu utilizatorul”, care a stat la baza filozofiei acestui pachet de aplicații software, permite o comunicare performantă, în condițiile în care sistemul este supervizat de către operatori umani care nu au cunoștințe speciale de programare, ci doar un minim necesar de cunoștințe legate de utilizarea calculatorului și a software-ului respectiv.

Aplicația de la Porțile de Fier II demonstrează posibilitatea conceperii unui sistem unic de monitorizare și dispecerizare a sistemelor hidrotehnice, conceput în acest caz pe trei nivele ierarhice, având un subsistem de automatizare, unul de măsurare și monitorizare precum și un subsistem de comunicație și dispecerizare.

Eficiența ansamblului este garantată prin introducerea, alături de cele mai moderne sisteme de măsurare (cu traductori de ultimă generație) a echipamentelor de comandă cu logică programată (controller-ele logice programabile -PLC) precum și a calculatoarelor industriale de proces.

Această soluție tehnică este cea mai avantajoasă din punct de vedere al exploatarei, dovedind eficiență și fiabilitate sporită în condițiile unei mentenanțe mai puțin pretențioase.

Este de dorit ca existența unor astfel de sisteme unitare de automatizare și control în unele sisteme hidrotehnice să conducă la generalizarea acestor soluții tehnice la cât mai mulți operatori de asemenea aplicații din România.

În practică s-a demonstrat funcționalitatea acestor sisteme, încă din cursul anului 2005, acest sistem funcționând fără avarii încă din cursul anului 2005, realizând, practic, mii de ecluzări. Mai mult chiar, implementarea lor este însoțită de avantaje economice care decurg din creșterea randamentului sistemului de navigație, creșterea siguranței în exploatare și diagnosticarea eficientă a stării sistemului. Investiția se amortizează singură în scurt timp.

Nu există o soluție singulară, general valabilă și absolut viabilă pentru tratarea problemelor legate de automatizarea sistemelor hidrotehnice și de navigație.

5.5. Contribuții personale

Din multitudinea de soluții tehnice pentru sisteme de automatizare, măsură și control aplicate în sfera sistemelor hidrotehnice din România, s-a ales pentru exemplificare unul dintre cele mai noi și mai complexe astfel de sisteme, în materializarea cărora autorul a adus o serie de contribuții originale esențiale pentru finalizarea acestor lucrări, dintre care amintim:

- identificarea parametrilor, mărimilor și a altor elemente esențiale pentru automatizare;
- elaborarea arhitecturii sistemelor de automatizare, monitorizare și dispecerizare a sistemelor hidrotehnice și de navigație;
- alegerea aparaturii necesare și stabilirea protocoalelor de comunicare între componente;
- proiectarea schemelor de conexiuni, atât pentru alimentare, cât și pentru comunicare;
- configurarea și utilizarea în schemele de automatizare a unor calculatoare industriale;
- posibilitatea integrării sistemului într-un ansamblu complet, alături de automatizarea pompelor, vanelor și porților;
- adaptarea sistemului la configurațiile existente, precum și posibilitatea unei funcționări flexibile a ansamblului;
- determinarea algoritmilor de automatizare;
- propunerea soluției de măsurare - achiziție date;
- elaborarea algoritmilor pentru prelucrarea și vizualizarea datelor;
- conceperea unei interfețe operator - sistem capabilă să conducă la eficientizarea funcționării, baza conceptului „user friendly - sistem prietenos cu utilizatorul”;
- realizarea pachetului software, o parte din facilitățile, procedurile și funcțiile acestui program fiind pe larg descrise în acest capitol sau detaliate în Anexe;
- coordonarea directă a montării sistemului la beneficiar precum și analiza funcționării sistemelor în condițiile reale de exploatare;
- propunerea unor soluții de completare a sistemelor existente, pentru a răspunde cât mai bine cerințelor beneficiarului.

6. CONCLUZII ȘI PERSPECTIVE

6.1. Rezultatele cercetării

În cadrul acestui capitol rezervat concluziilor vom prezenta mai întâi rezultatele cercetării, urmând ca apoi să expunem concluziile și contribuțiile personale ale autorului.

Rezultatele studiilor teoretice

Dintre cele mai importante rezultate ale studiilor teoretice efectuate amintim:

- Stabilirea principalilor parametri și a principalelor mărimi care intervin în domeniul sistemelor hidrotehnice;
- Descrierea principalelor tipuri de traductoare utilizate în acest domeniu;
- Studiul sistemelor informatice de proces și al posibilelor configurații ale acestora;
- Analiza aplicațiilor distribuite în timp real;
- Introducerea conceptului de sistem SCADA în domeniul sistemelor hidrotehnice, concept preluat din sfera electroenergeticii;
- Descrierea principalelor metode de comunicație între elementele sistemului;
- Selecția tipurilor de baze de date utilizabile în acest domeniu precum și a algoritmilor necesari pentru filtrarea informațiilor conținute în aceste baze de date;
- Sinteza noțiunilor teoretice legate de urmărirea și conducerea proceselor tehnologice;
- Formularea matematică a problemelor de optimizare;
- Delimitarea nivelurilor de automatizare întâlnite în conducerea proceselor industriale;
- Calculul optimizării fără restricții;
- Justificarea raționamentului sistemului expert;
- Rezolvarea posibilelor conflicte în colaborarea pe verticală;
- Definirea noțiunii de sistem expert aplicat în domeniul sistemelor hidrotehnice;

Rezultatele cercetării aplicative

Cele mai importante aplicații ale studiilor teoretice sunt:

- Introducerea indicatorilor de fiabilitate în evaluarea performanțelor traductoarelor;
- Conceperea și programarea aplicațiilor de tip distribuit;
- Formularea funcțiilor pe care trebuie să le îndeplinească un sistem Dispecer implementat în cazul sistemelor hidrotehnice;
- Elaborarea unei structuri ierarhice pe trei nivele, deosebit de utilă în situația sistemelor de automatizare, control și monitorizare a sistemelor hidrotehnice;
- Definirea arhitecturii de tip SCADA, aplicată unor sisteme hidrotehnice;
- Implementarea bazelor de date în timp real în structura unui sistem de automatizare control și monitorizare a sistemelor hidrotehnice;
- Conceperea schemelor de automatizare a porților, vanelor și tuturor instalațiilor adiacente;
- Stabilirea unor programe de automatizare implementate pe PLC-uri;
- Stabilirea unor principii de bază pentru conceperea unei interfețe om - mașină cât mai simplă și cât mai performantă;
- Determinarea metodelor de calcul pentru optimizarea fără restricții a proceselor specifice acestui domeniu;

- Analiza unor soluții legate de sistemul expert;
- Aplicarea celor mai noi concepte privind filtrarea informațiilor provenind din procesul tehnic;
- Construirea unui sistem expert aplicabil în situația sistemelor hidrotehnice.

Rezultatele tehnice și experimentale

Cele mai importante soluții tehnice propuse, sau realizări practice și experimentale, sunt:

- Alegerea unor traductoare specifice performante pentru măsurarea diferitelor mărimi importante pentru activitatea din sectorul hidrotehnic;
- Interfațarea traductoarelor cu structurile ierarhice superioare;
- Introducerea și configurarea calculatoarelor de proces tip EAD destinate controlului acestor procese;
- Detalierea soluțiilor de comunicare între calculatoarele EAD și calculatorul DISPECER;
- Configurarea hardware a unui punct dispecer;
- Definirea arhitecturii reale a unui sistem SCADA aplicabil în acest domeniu;
- Structurarea bazelor de date în timp real;
- Filtrarea informațiilor conținute în bazele de date prin utilizarea unei interfețe simple și eficiente între om și sistemul de gestiune a bazei de date, bazată pe diverse tipuri ferestre - ecran;
- Scrierea secvențelor de program necesare pentru materializarea acestor ferestre, incluse în pachetul software;
- Ierarhizarea componentelor unui sistem informatic de proces destinat automatizării, controlului și monitorizării sistemelor hidrotehnice;
- Conceperea unor astfel de sisteme „la cheie”;
- Proiectarea, pentru S.C. HIDROTIM S.A. a întregului sistem de automatizare, monitorizare a activității de gestionare a nodului hidrotehnic și de navigație;
- Supervizarea implementării ansamblului la beneficiar;
- Verificarea experimentală a funcționării sistemului implementat;

6.2. Concluziile principale

Așa cum se observă din structura tezei, fiecare capitol are în încheiere un subcapitol care detaliază principalele concluzii specifice capitolului respectiv. De aceea nu vom insista asupra tuturor concluziilor, ci doar asupra celor mai importante dintre ele, relevante pentru întregul demers.

Dintre acestea enumerăm:

- Existența unor soluții tehnice destinate instalațiilor de măsură și control având grad mare de complexitate, care rezultă și din numărul mare de variante constructive posibile pentru traductori de presiune, temperatură, debit etc.;
- Lipsa unui „algoritm” general valabil, pentru alegerea soluției tehnice pentru instalațiile de măsură și control, alegerea acestora fiind uneori empirică și bazându-se de cele mai multe ori pe criterii pur economice, nu întotdeauna justificate tehnic;
- Cunoașterea și mai ales utilizarea celor mai noi modele de traductori și de aparate de măsură și control, produse atât în străinătate, cât și în România, integrate în aplicații specifice, conduce la creșterea randamentului în exploatare, dar și la scăderea costurilor de fabricație și reducerea pierderilor, precum și la creșterea siguranței în exploatare;

- Conceperea unui sistem informatic de proces cât mai performant presupune gruparea tuturor elementelor pe trei nivele ierarhice, anume Nivelul "0" sau nivelul de bază (format din totalitatea traductorilor și elementelor de execuție conectate la procesul urmărit și condus), Nivelul „1” (destinat cuplurului de proces) și Nivelul „2” care ca suport fizic echipamentele de calcul (calculator, imprimantă, MODEM, UPS) de la punctul dispecer și de la nivelul de conducere sau supervizare;
- Bazele de date relaționale (tabelare) reprezintă soluția cea mai adecvată pentru stocarea, afișarea și manipularea valorilor diverselor mărimi întâlnite în cadrul ansamblului, iar pentru a asigura funcția de manipulare a bazei de date este necesară existența unui SGBD care să facă apel la funcții și proceduri generate prin intermediul unor limbaje de nivel mai înalt, ca de exemplu limbajele din familia C (Visual C, C++, Borland C etc.);
- Folosirea unor ferestre dedicate pentru dialogul om – mașină (om – bază de date) este extrem de necesară pentru prelucrarea și valorificarea informației, informație care este vitală pentru buna funcționare a automatizării în ansamblu;
- Conceperea unei scheme de automatizare performante nu poate fi făcută decât luând în calcul și modelul matematic al procesului controlat precum și cel al metodei de control. Formularea matematică a problemei de optimizare trebuie făcută astfel încât să permită conceperea soluției de optimizare, fără a genera un număr important de restricții.
- Interfața OM - MAȘINĂ, care este localizată la nivelul calculatoarelor cu funcție de dispecer sub forma unor aplicații software, în situația automatizărilor din domeniul hidrotehnic, trebuie concepută într-un mod cât mai simplu (conceptul „user – friendly”), dar fără a diminua din eficiența procesului de automatizare în ansamblu.
- Aplicația de la Porțile de Fier II demonstrează posibilitatea conceperii unui sistem unic de monitorizare și dispecerizare a sistemelor hidrotehnice, conceput în acest caz pe trei nivele ierarhice, având un subsistem de automatizare, unul de măsurare și monitorizare precum și un subsistem de comunicație și dispecerizare. Eficiența și fiabilitatea ansamblului sunt garantate prin introducerea, alături de cele mai moderne sisteme de măsurare (cu traductori de ultimă generație) a echipamentelor de comandă cu logică programată (controller-ele logice programabile -PLC) precum și a calculatoarelor industriale de proces;
- În cazul unei aplicații complexe, cum este cea de la Porțile de Fier II, sistemul SCADA cu structura ierarhizată specifică este conceput să opereze în directă colaborare cu sistemul de comandă al pompelor și vanelor (echilibrare hidraulică și automatizare pompe), într-un sistem unitar și complet. Această soluție tehnică este avantajoasă și din punct de vedere al exploatării, dovedind eficiență și fiabilitate în condițiile unei mentenanțe mai puțin pretențioase.

6.3. Contribuțiile personale și originale

Deoarece, așa cum se observă și din structura tezei, fiecare capitol se încheie cu un subcapitol dedicat contribuțiilor personale și acolo unde este cazul, chiar originale, nu vom mai insista asupra fiecăreia dintre contribuțiile evidențiate în capitolele tezei.

În plus, la începutul acestui capitol s-au enumerat rezultatele cercetării, acestea toate fiind consecințe directe ale contribuțiilor personale ale autorului.

Se va face doar o prezentare succintă a contribuțiilor personale cele mai importante, alături de o descriere a demersului efectuat.

Lucrarea a debutat cu un studiu teoretic, al soluțiilor de măsurare aplicabile pentru determinarea mărimilor neelectrice care intervin în activitatea sistemelor hidrotehnice.

Cele mai importante contribuții personale ale autorului sunt date de sinteza bibliografică și analiza critică a soluțiilor tehnice existente sau a standardelor în domeniu, dar mai ales propunerea unor soluții originale.

Studiile bibliografice au vizat:

- Principalele mărimi întâlnite în acest domeniu de activitate;
- Standardele și reglementările juridice sau tehnice care intervin în această ramură de activitate;
- Construcția traductoarelor destinate achiziționării acestor mărimi;
- Determinarea unor indicatori sintetici de evaluare a performanțelor acestor traductoare.

În acest capitol autorul a încercat să dea unele răspunsuri la întrebările:

„Ce sunt sistemele informatice de proces?”

„Care sunt funcțiile dispecerului unui sistem hidrotehnic?”

„Care este structura hardware a unui astfel de sistem informatic de proces cu rol de dispecer?”

Chiar dacă în acest capitol se întâlnesc și numeroase sinteze bibliografice, contribuțiile originale ale autorului apar destul de proeminente.

Acestea sunt:

- stabilirea unei structuri ierarhizate pentru un astfel de sistem informatic de proces conceput pentru automatizarea, controlul și măsurarea parametrilor sistemelor hidrotehnice;
- determinarea funcțiilor specifice ale dispecerului sistemului hidrotehnic;
- conceperea unei structuri hardware performante, bazată pe elemente robuste și fiabile;
- configurarea elementelor hardware ale sistemului;
- stabilirea soluțiilor pentru comunicare în interiorul sistemului;
- propunerea unor soluții tehnice pentru implementarea acestor sisteme la beneficiar;

Lucrarea continuă cu o analiză a componentei software a unui sistem informatic de proces.

Bazele de date sunt instrumentul cel mai fiabil pentru materializarea componentei software.

Întrebările centrale pe care s-a axat demersul acestui capitol sunt:

„Ce fel de baze de date pot fi utilizate în acest domeniu?”

„Cum trebuie concepută o interfață performantă om - mașină?”

Contribuțiile autorului devin esențiale, meritând să amintim următoarele:

- Conceperea unui software specializat, flexibil, una din componentele acestui pachet software fiind și un Sistem de Gestionare a Bazelor de Date;
- Elaborarea, în cadrul acestui pachet software, a unei interfețe performante om-mașină (om-bază de date), axată pe conceptul de „user friendly”, utilizând ferestre de dialog;
- Conceperea algoritmilor și a programelor în mediul Visual C, programe destinate manipulării și prelucrării datelor, numeroase proceduri fiind chiar exemplificate pe parcursul capitolului.

O atenție sporită este acordată în continuare problemelor legate de optimizarea automatizărilor industriale și conducerea eficientă a proceselor tehnologice.

S-au avut în vedere următoarele întrebări:

„Cum se poate optimiza conducerea proceselor tehnologice?”

„Care este relația om - mașină?”

„Ce este un sistem expert?”

Chiar dacă, în această secțiune, ponderea revine studiului teoretic, contribuțiile originale nu sunt de loc neglijabile. Cele mai importante sunt:

- Formularea și studiul unor modele matematice capabile să contribuie la definitivarea soluției de automatizare aleasă;
- Analiza obiectivă a modului de conlucrare OM - MAȘINĂ;
- Evaluarea soluțiilor de automatizare aplicabile în cazul sistemelor hidrotehnice;
- Propunerea unor soluții de automatizare bazate pe conceptul de sistem expert;

Întregul demers anterior este exemplificat în finalul tezei prin prezentarea aplicației de la Porțile de Fier II.

Întrebarea centrală a acestui capitol este:

„Cum arată practic un astfel de sistem de automatizare control și monitorizare?”

În această parte a lucrării nu putem vorbi decât de contribuții personale ale autorului, cele mai importante fiind:

- elaborarea arhitecturii sistemelor de automatizare, monitorizare și dispecerizare a sistemelor hidrotehnice;
- identificarea parametrilor, mărimilor și a altor elemente esențiale pentru automatizare;
- alegerea aparaturii necesare și stabilirea protocoalelor de comunicare între componente;
- proiectarea schemelor de conexiuni, atât pentru alimentare, cât și pentru comunicare;
- configurarea și utilizarea în schemele de automatizare a unor calculatoare industriale;
- configurarea și utilizarea în schemele de automatizare a unor calculatoare industriale;
- posibilitatea integrării sistemului într-un ansamblu complet, alături de automatizarea pompelor circuitelor și vanelor;
- conceperea unei interfețe operator - sistem capabilă să conducă la eficientizarea funcționării, baza conceptului „user friendly - sistem prietenos cu utilizatorul”;
- realizarea pachetului software;

6.4. Perspectivele continuării studiului

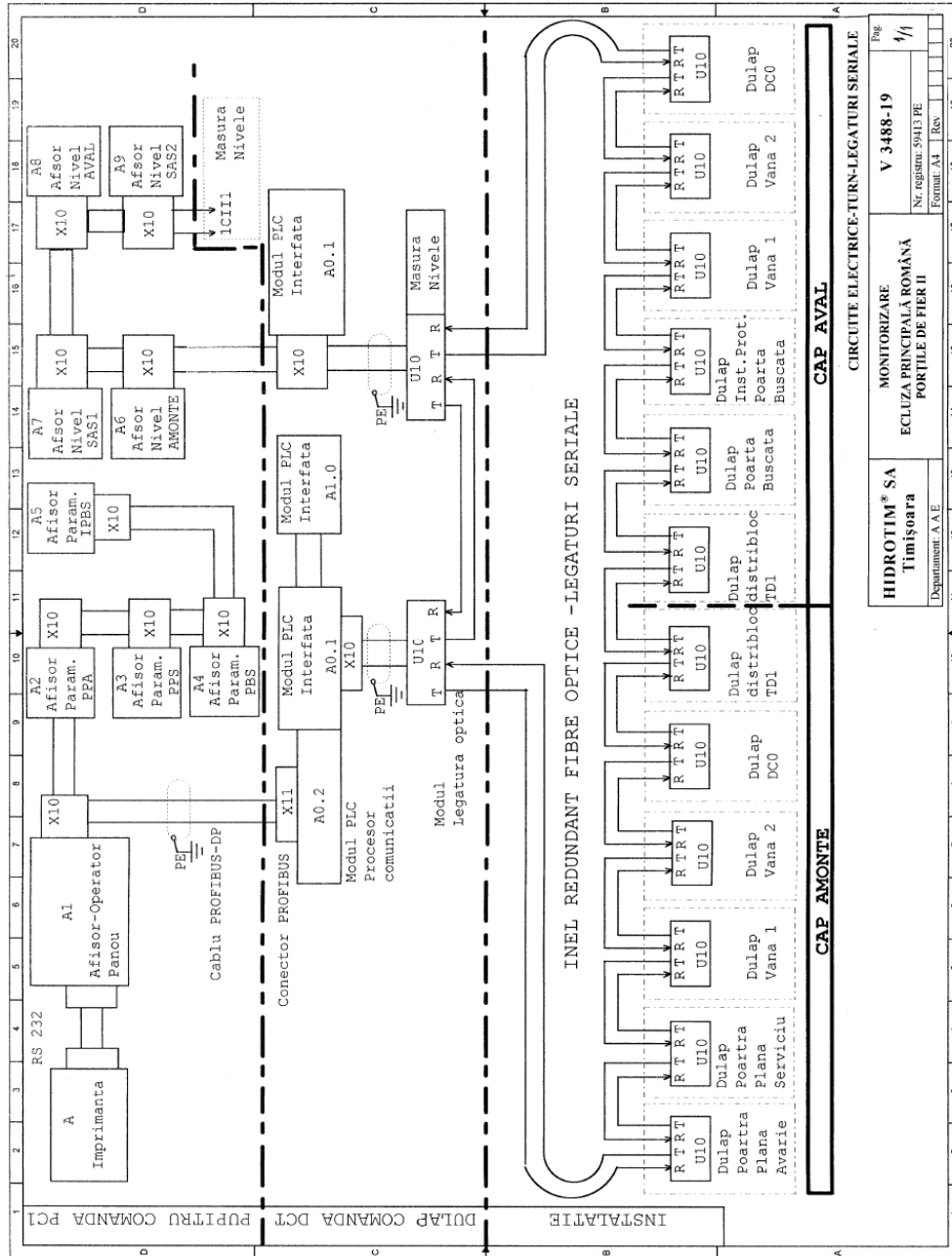
Procesul de modernizare a infrastructurii din domeniul navigației fluviale va continua și în următorii ani, în cadrul strategiei de naționale transport. Modernizarea infrastructurii impune și introducerea unor sisteme moderne de automatizare, măsurare și control al întregului ansamblu.

Autorul își propune ca și pe viitor să conceapă astfel de aplicații, implementate la beneficiar de către S.C. HIDROTIM S.A., numeroase proiecte de acest fel fiind deja în desfășurare. Toate aceste aplicații trebuie să țină cont de particularitățile aplicației

beneficiarului, soluția finală fiind oferită "la cheie". S.C. HIDROTIM S.A. dorește ca pe viitor să-și păstreze poziția pe care o ocupă în rândul companiilor din România producătoare de sisteme de automatizare pentru prestatorii de utilități publice.

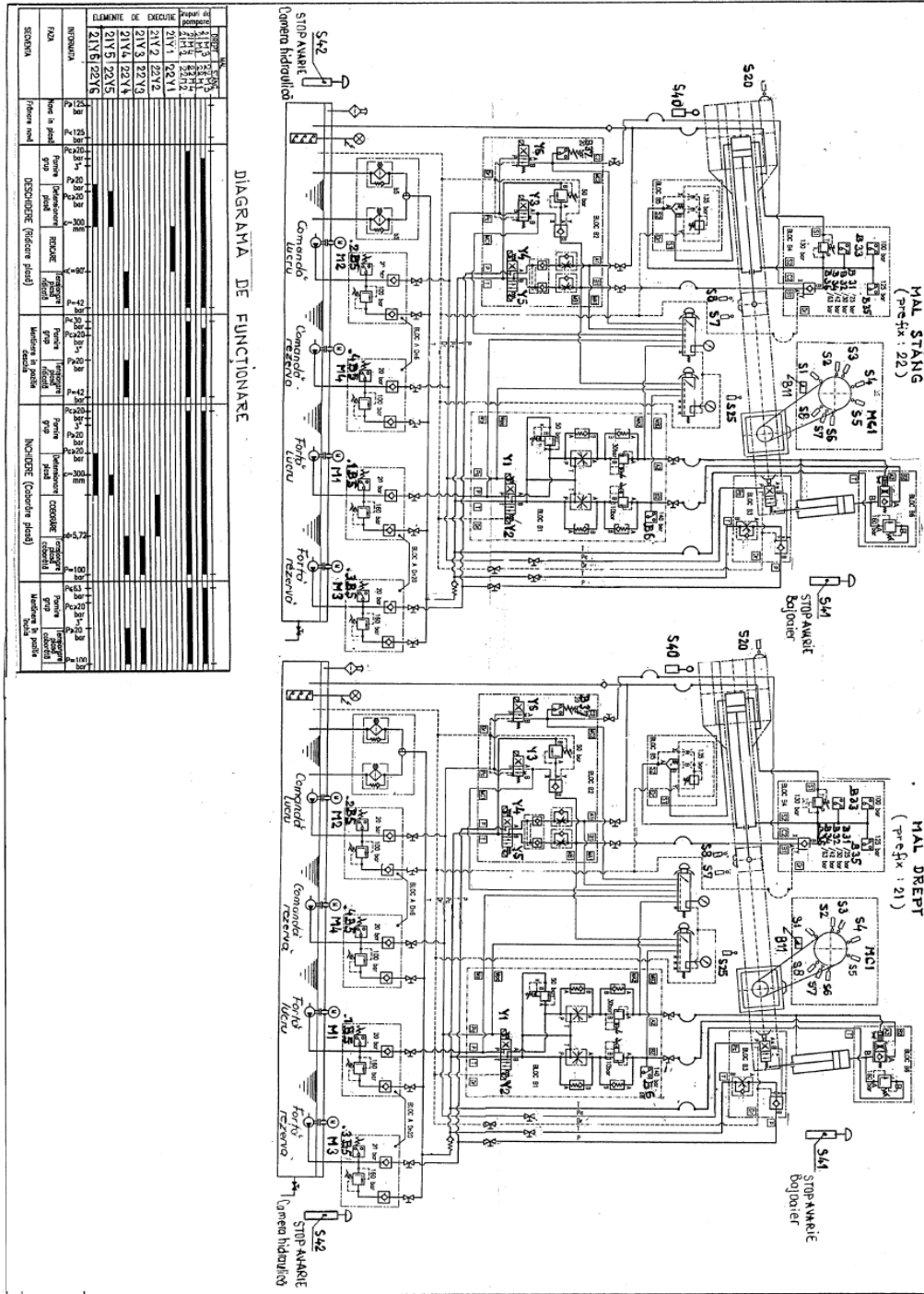
O dată cu progresele apărute în domeniul tehnicii de calcul și al IT -ului în general (știut fiind faptul că acest domeniu înregistrează o dinamică extrem de importantă), autorul, împreună cu specialiștii firmei S.C. HIDROTIM S.A. își propun implementarea celor mai noi și mai moderne soluții tehnice din domeniu, în scopul asigurării unei funcționări cât mai fiabile a întregului ansamblu monitorizat, continuând colaborarea și cu specialiștii din cadrul Catedrei de Electroenergetică a Facultății de Electrotehnică și Electroenergetică de la Universitatea „Politehnica” din Timișoara.

ANEXE

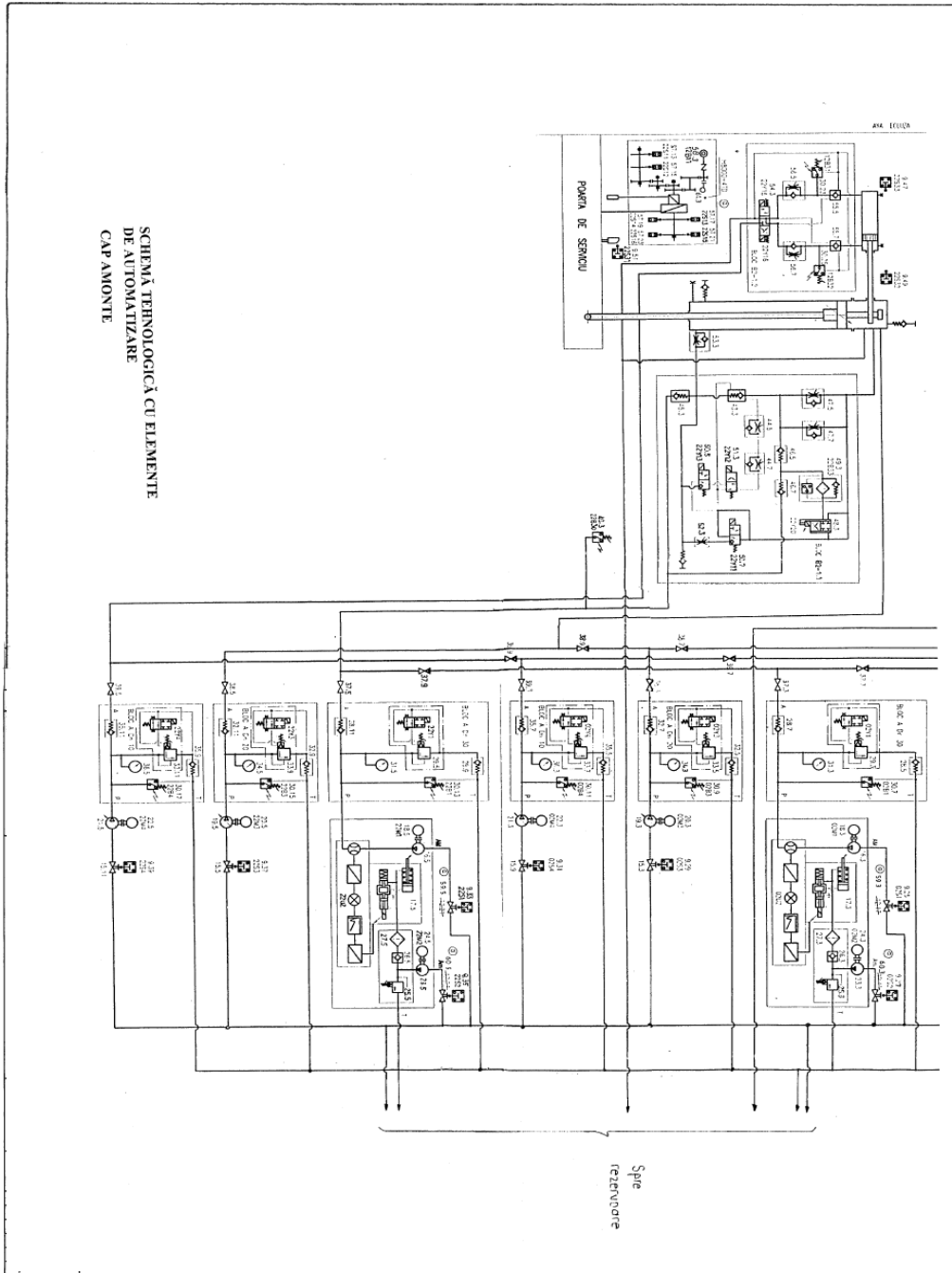


Anexa 1. Schema bloc pentru monitorizarea ecluzei române Pdf II.

HIDROTIM SA Timisoara Departament A A E		MONITORIZARE ECLUZA PRINCIPALA ROMANA PORTILE DE FIER II		Pag. V 3488-19 Nr. registr. SPH PE Format: A4 Rev.
--	--	---	--	---

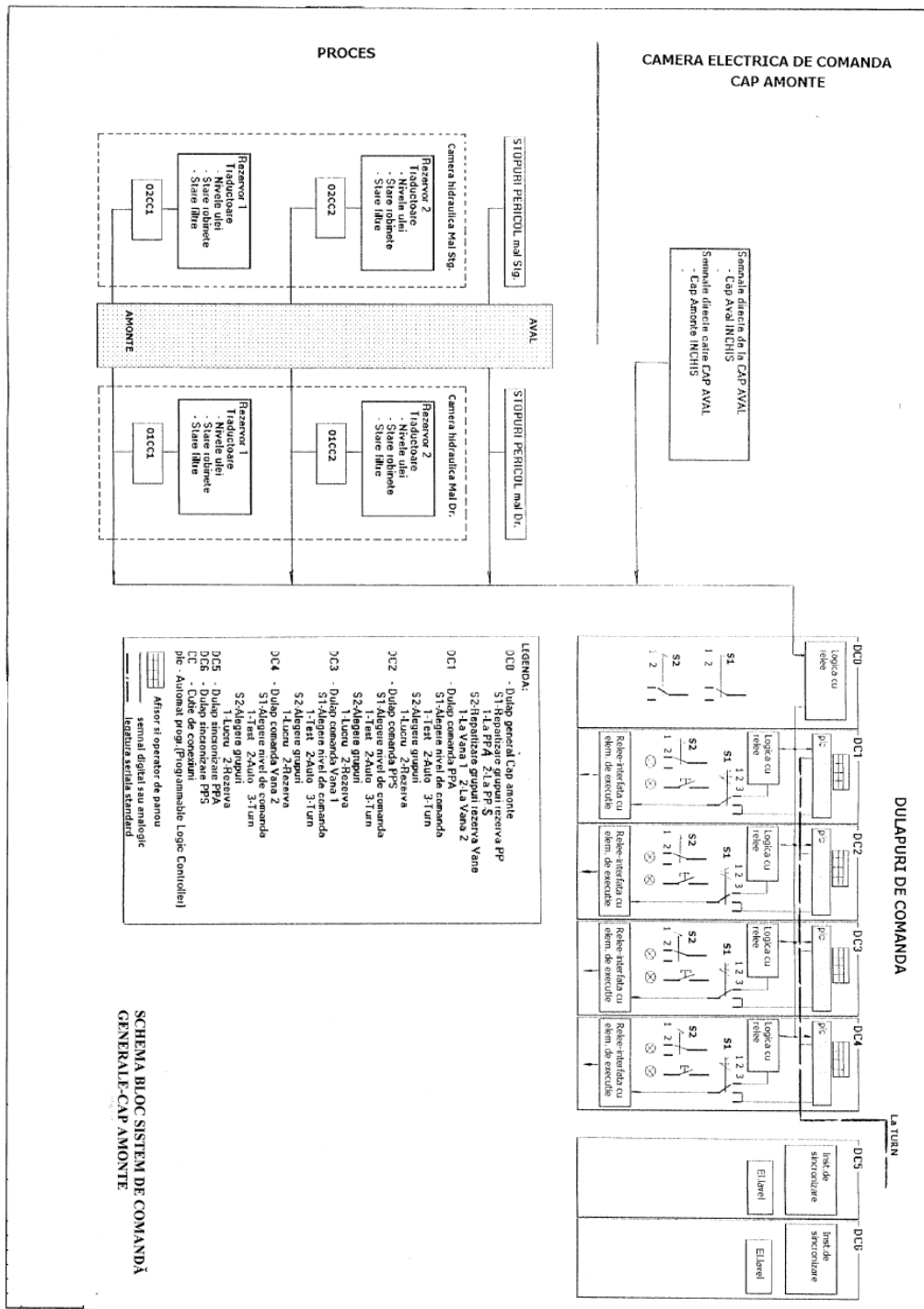


Anexa 2. Schema electrică desfășurată.

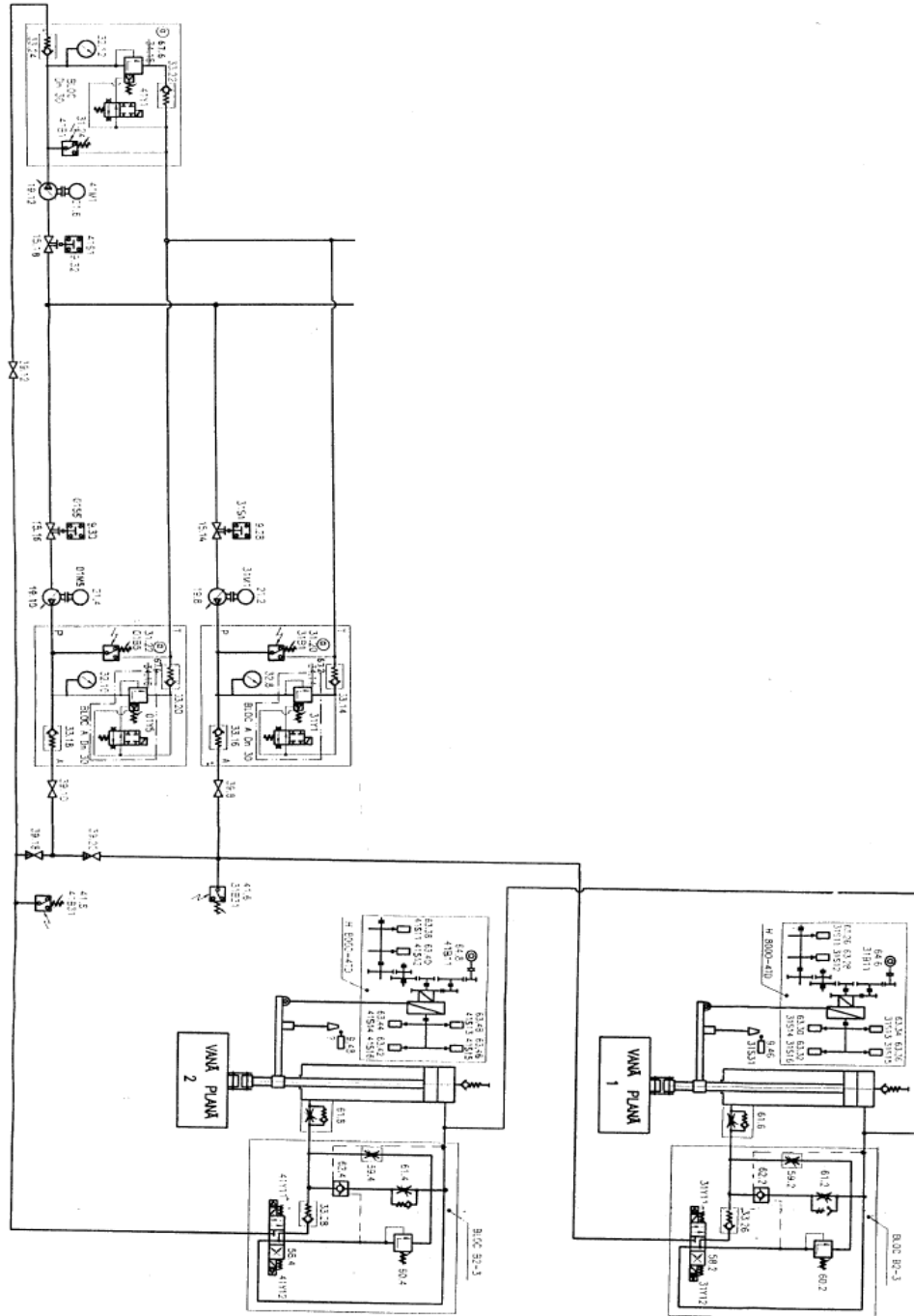


SCHEMĂ TEHNOLOGICĂ CU ELEMENTE
DE AUTOMATIZARE
CAP AMONTE

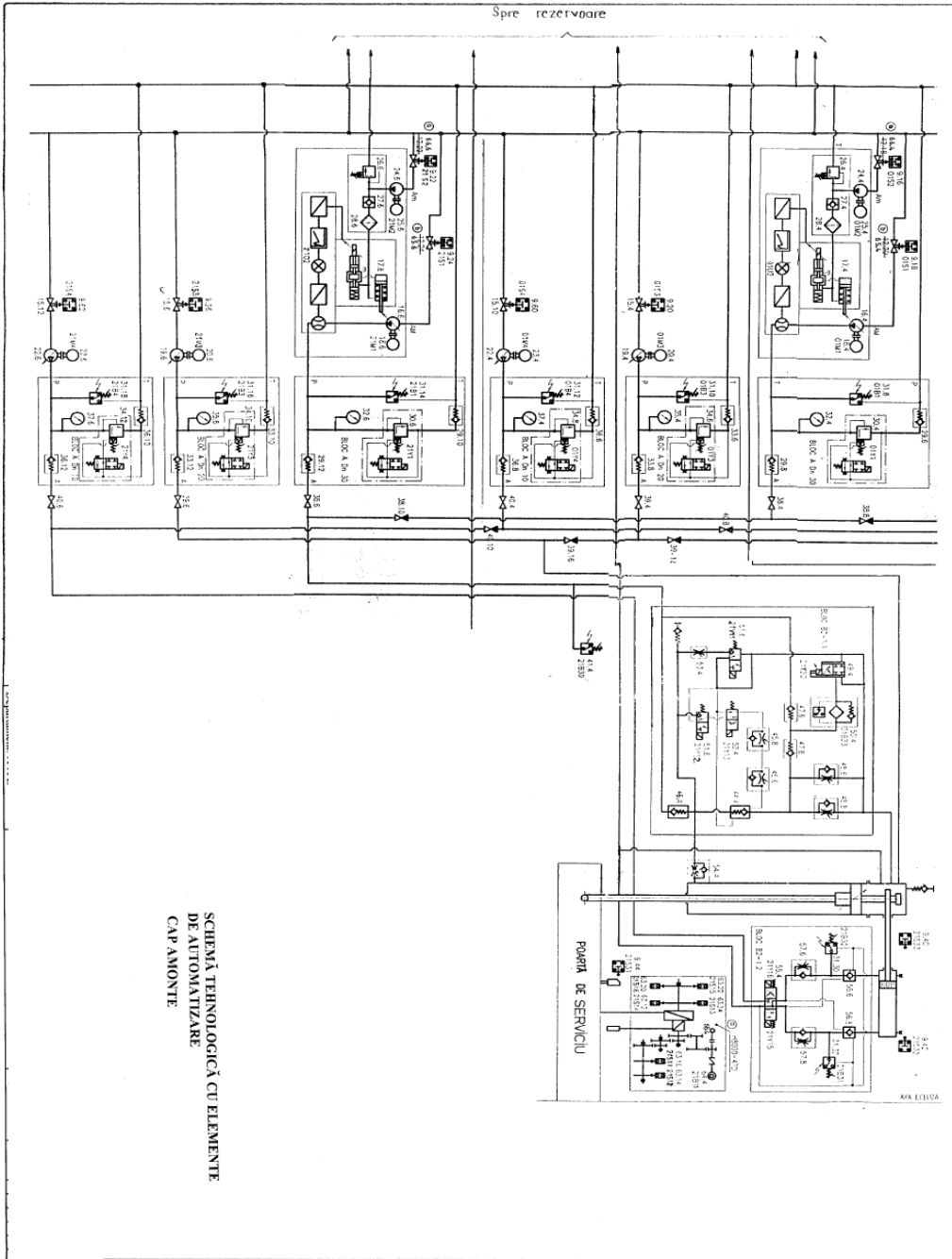
Anexa 3. Schema tehnologică cu elemente de automatizare CAP AMONTE.



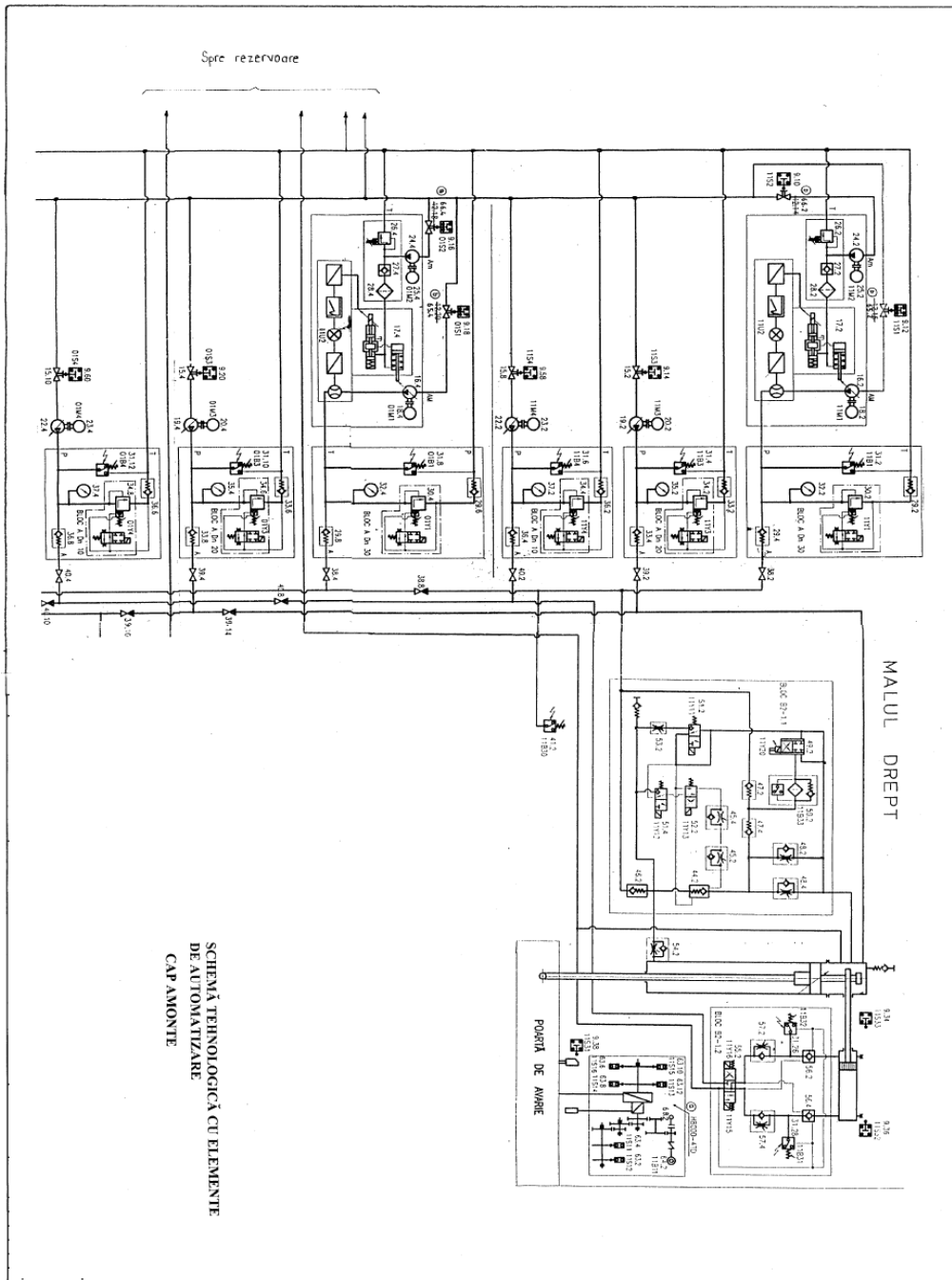
Anexa 4. Schema bloc sistem de comandă CAP AMONTE.



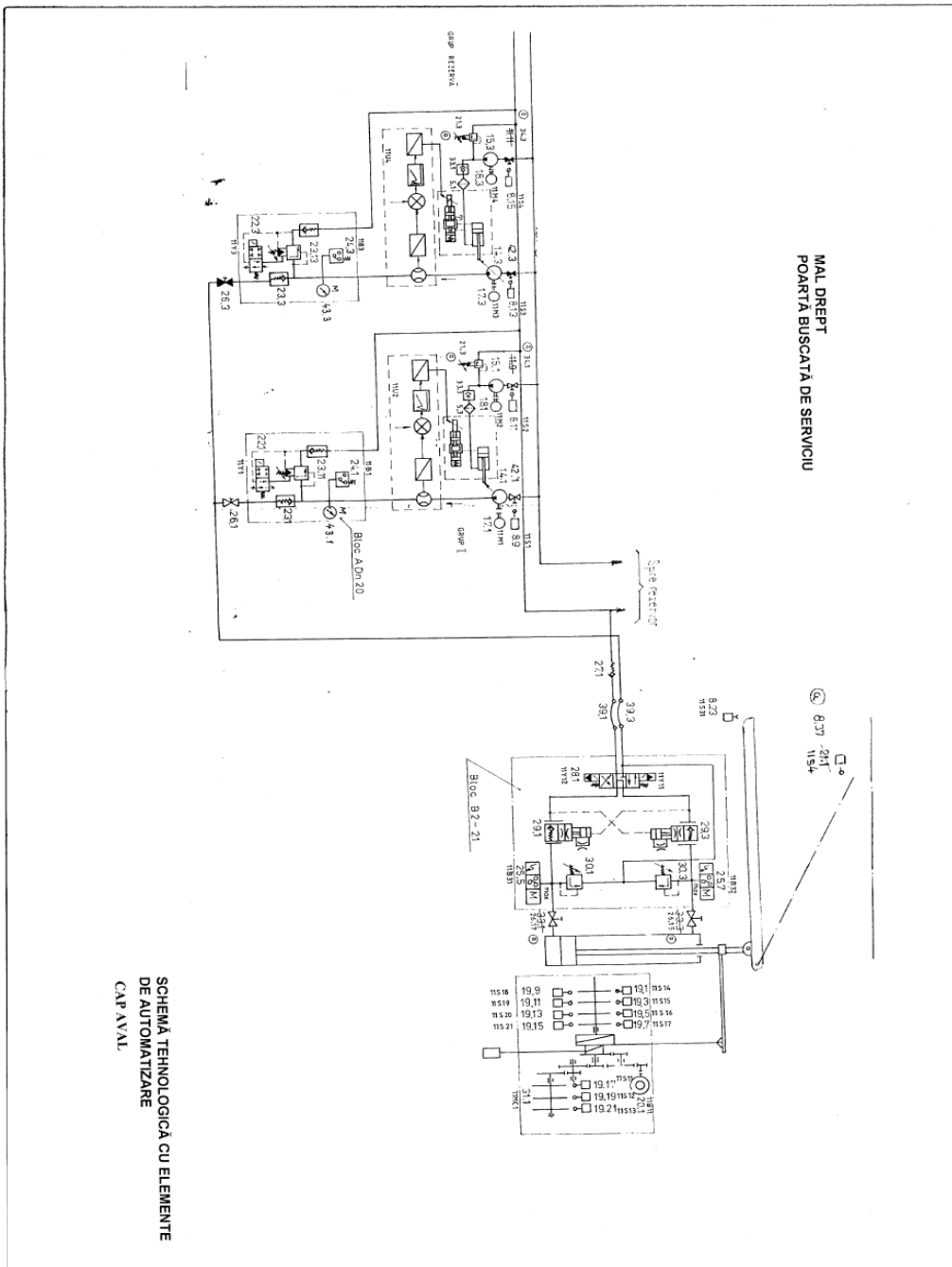
Anexa 5. Schema tehnologică cu elemente de automatizare CAP AMONTE – Continuare I.



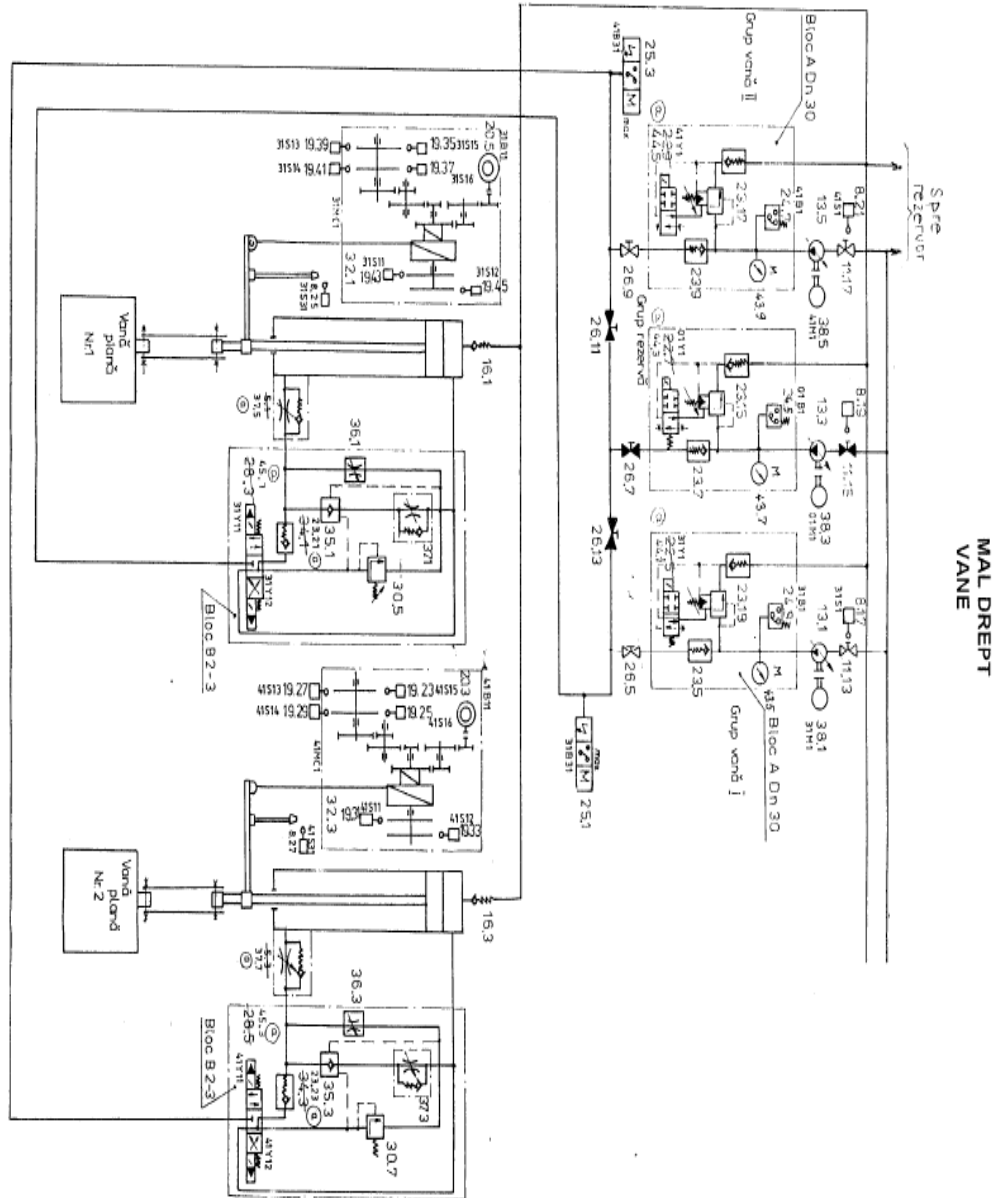
Anexa 6. Schema tehnologică cu elemente de automatizare CAP AMONTE – Continuare II.



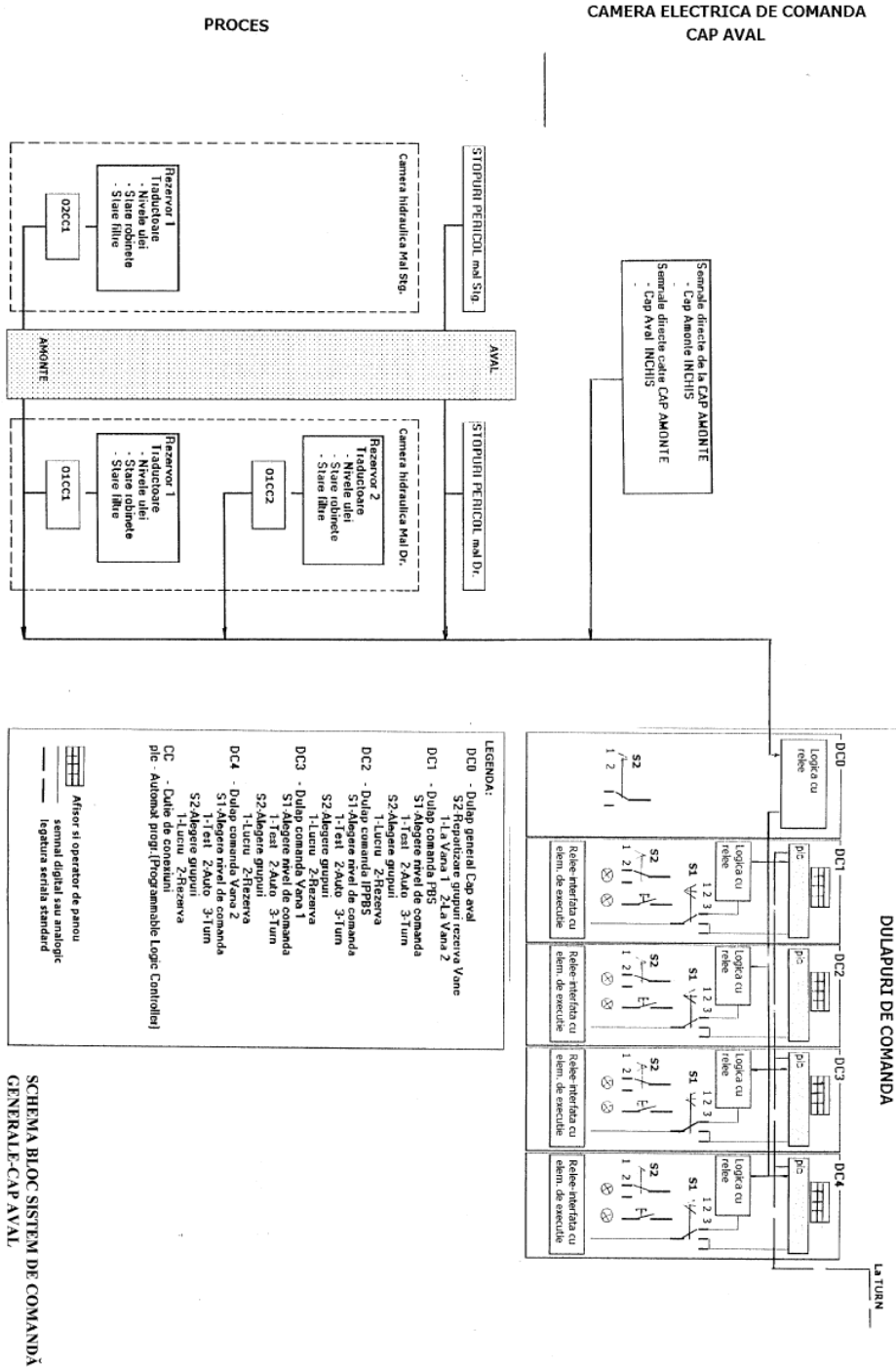
Anexa 7. Schema tehnologică cu elemente de automatizare CAP AMONTE – Continuare III.



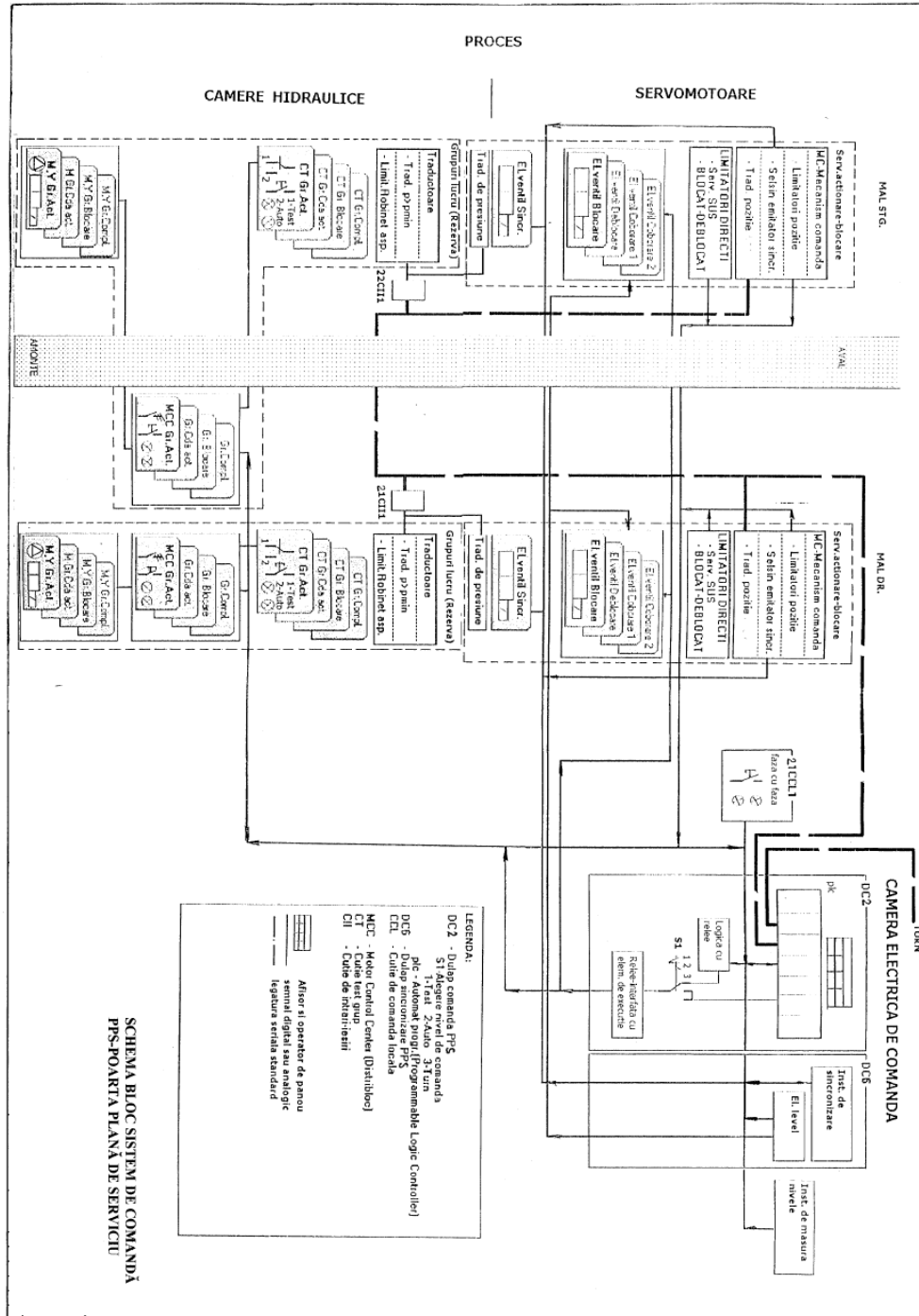
Anexa 8. Schema tehnologică cu elemente de automatizare CAP AVAL.



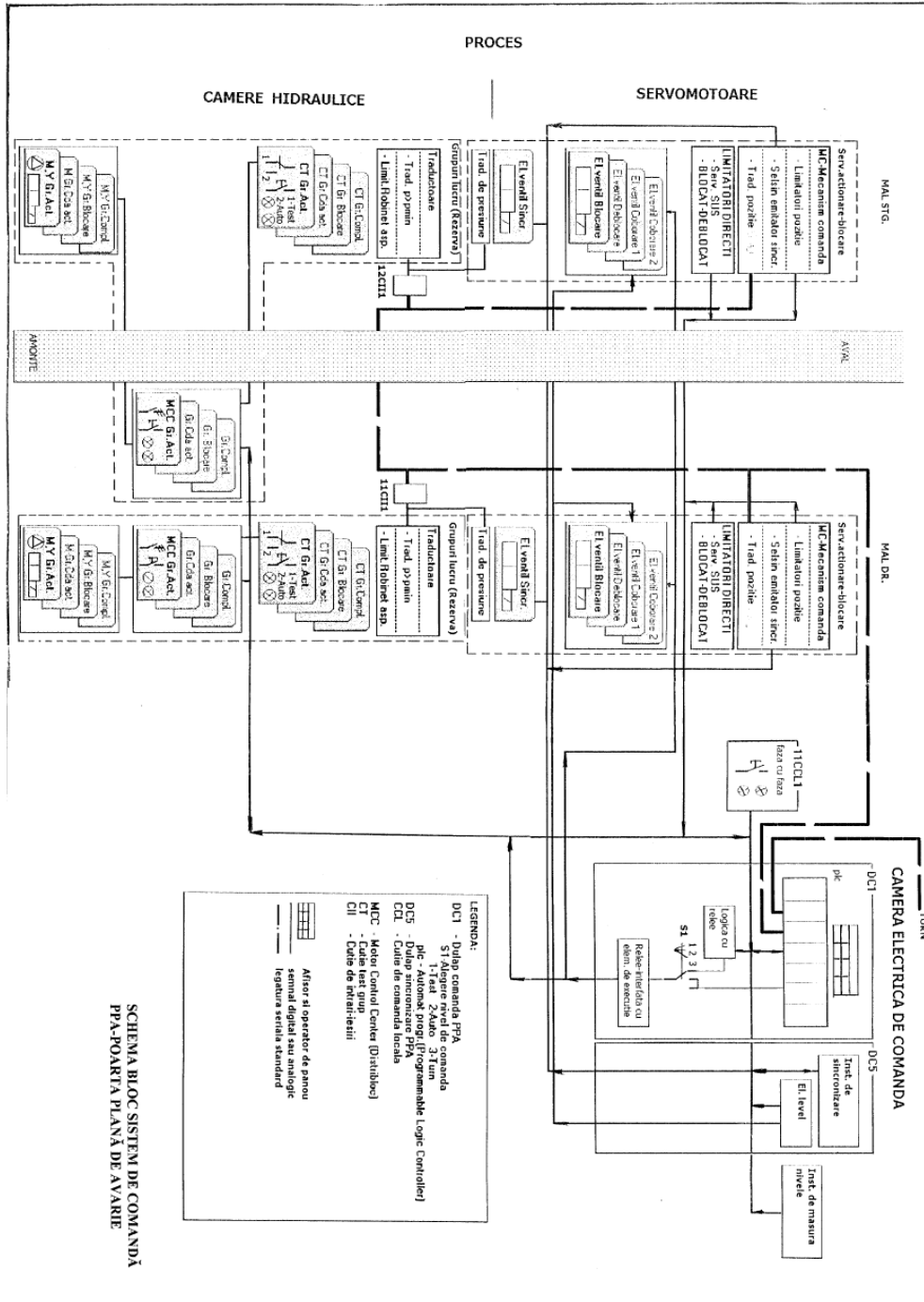
Anexa 9. Schema tehnologică cu elemente de automatizare CAP AVAL- Continuare I.



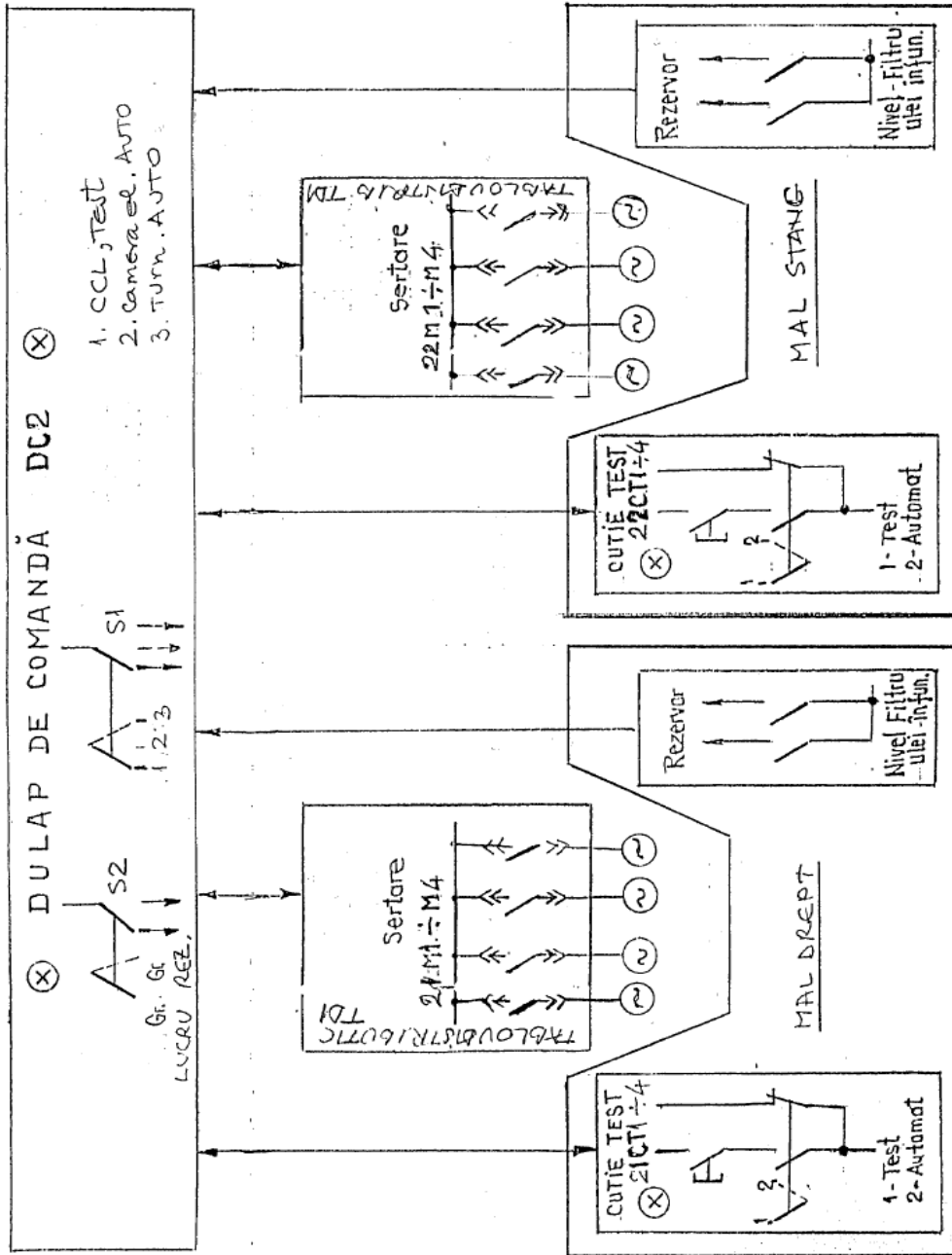
Anexa 11. Schema tehnologică cu elemente de automatizare CAP AVAL- Continuare III.



Anexa 12. Schema bloc comandă PPS.



Anexa 13. Schema bloc comandă PPA.



Anexa 14. Schema bloc a unui dulap de conexiuni.

BIBLIOGRAFIE

- [1] ABDALLA - GHALLY, A., A. CHAN, B.,L., "Evaluation of competing reliability predictions" IEEE Transactions on Software Engineering, pag. 950-967, 1986
- [2] AGACHI, S., "Automatizarea proceselor chimice", Casa Cărții și a Științei; Cluj Napoca, 1994
- [3] BALOG, A., "Standardele ISO 9000 în domeniul calității software", Rev. PCWorld, Nr. 6/1995, Pag. 50
- [4] BARABOI, A., ș.a. "Tehnici moderne de comutație de putere", Editura A92, Iași, 1996
- [5] BENCHIMOL, G., ș.a. "Sisteme expert în întreprindere", Editura Tehnică București, 1993
- [6] BORANGIU, Th., ș.a. "Conducerea multiprocesor în timp real a structurilor flexibile de fabricație", Editura Tehnică, București, 1989
- [7] CĂPĂȚĂNĂ, O., ș.a. "Proiectarea cu microcalculatoare integrate", Editura Dacia, Cluj Napoca, 1992
- [8] CĂRȚINĂ, Gh. ș.a. "Rețele neuronale și sisteme expert în energetică", Editura Gh. Asachi, Iași, 1994
- [9] CHEN, J., "Izolarea și detectarea erorii bazată pe observare, rezistență și aplicații", 1992
- [10] CLARK, R.N., "A Simplified Instrument Failure Detection Scheme", 1994
- [11] CRISTEA, V., ș.a. "Rețele de calculatoare", Editura Teora, București 1992
- [12] DAVIDOVICIU, A., BĂRBAT, B., "Limbaje de programare pentru sisteme în timp real", Editura Tehnică, București, 1986
- [13] DAVIDOVICIU, A., ș.a. "Minicalculatoare și microcalculatoare în conducerea proceselor industriale", Editura Tehnică, București, 1983
- [14] DESAI, M., ROI, A., "A Fault Detection and Isolation Metodology", IEEE, 1993
- [15] DUMITRESCU, S., ș.a., "Aplicații inginerești ale calculatorului"
- [16] DUROCHER, D., "Langage: An Expert System for Alarm Processing", Proceedings of the Eleventh Biennial IEEE Workshop on Power System Control Centers, Montreal, Canada, 1990
- [17] EREMIA, M., SÂNDULEAC, M., "Introducerea sistemelor expert. Aspecte generale". Rev. Energetica, nr.5, Seria B, 1993
- [18] EREMIA, M., ș.a., "Sistem expert pentru controlul U - Q în sistemul energetic", Lucrare pentru Conferința de Rețele Electrice de foarte Înaltă Tensiune, Sibiu, 1995, Proceedings vol.3
- [19] FEHER, K, ș.a., "Comunicații digitale avansate", vol. I, Editura Tehnică, București, 1993
- [20] FELEA, I., "Ingineria fiabilității în electroenergetică", Editura Didactică și Pedagogică, R.A., București, 1996
- [21] FRANK, P., M., "Diagnoza de eroare bazată pe model cantitativ și analitic. O privire generală și câteva noi rezultate", IEEE, 1979
- [22] FRANK, P., M., "Fault Diagnostics in Dinamic System via Static Estimation", IEEE 1978
- [23] FRIGURĂ – ILIASA, F., VĂȚĂU, D., "Utilizarea automatelor programabile în schemele de comandă ale unor instalații electrice" Proceeding-ul Simpozionului Științific Internațional de la Petroșani, pag. 106-111, Petroșani, 1999

- [24] FRIGURĂ – ILIASA, F, VĂTĂU, D., "Implementarea automatelor programabile în comanda unui reactor chimic" Proceeding-ul celui de-al IV-lea Simpozion Internațional "Cercetarea Multidisciplinară Regională", Timișoara, 16-18 Noiembrie 2000, pag. 1313 – 1322, S.C. Infotim S.A., Editura Sudura, Timișoara, 2001
- [25] FRIGURĂ – ILIASA, F., M., MĂȚIU-IOVAN L., VĂTĂU, D., JUDE S., A., "A Few Aspects Concerning the Thermal Finite Element Modelling of a Low Voltage ZnO Based Varistor with an Additional Mass" Proceedings 3rd International Conference "From Scientific Computing to Computational Engineering 3rd IC-SCCE, Athens, Greece 9-12 July, 2008, paper 078
- [26] HORTOPAN, G., "Aparate electrice", Editura Tehnică, București 2000
- [27] HORTOPAN, G., "Aparate electrice de comutație", Editura Tehnică, 1993
- [28] IFTIME, V., FRIGURĂ – ILIASA F., "A Few Measuring Solutions and Some Intelligent Management Systems Used by Heating Distribution Companies and Power Plants in Romania", Buletinul Științific al Universității POLITEHNICA din Timișoara, Seria Energetică, Tom 46(60), Fascicola 1 – 2, Timișoara 2001, Proceedings of the Fourth International Power Systems Conference, Timișoara, 8-9 Noiembrie 2001, pag. 215 - 218
- [29] IGNEA, A., CHIVU, M., BORZA, I., "Măsurări electrice și electronice în instalații", Editura Orizonturi Universitare, Timișoara, 1998
- [30] IVAS, D., MUNTEANU, F., "Funcțiunile unui sistem expert în centrele de conducere ale rețelelor electrice și arhitecturi care pot prelua aceste funcțiuni", Rev. Energetica nr.1, Seria B, 1994
- [31] JIAN, I., GAVRILESCU, H., "Utilizarea bazelor de date", Editura Mirton, Timișoara, 1994.
- [32] JOHNES, H., L., "Failure Detection in Linear Systems", Ph.D. Thesis, MIT, Cambridge, M.A.
- [33] JUDE, S., A., EHEGARTNER, P., ANDEA, P., FRIGURĂ-ILIASA, F. M., "New Techniques for High Voltage Equipment Maintenance" Proceedings of the 10th International Symposium "Materials, Methods and Technologies", 2-6 iunie 2008, Sunny Beach Resort-Burgas, Bulgaria, pp.100-107
- [34] JUDE, S., A., EHEGARTNER, P., ANDEA, P., FRIGURĂ-ILIASA, F. M., „Considerations about the Thermal Stability Evaluation for Low Voltage MOV Surge-Arresters" Proceedings of the 2nd International Conference on Electrical and Electronics Engineering, ICEEE 08, Laghouat, 21-23 aprilie 2008, Algeria, pag 322 – 327, (apărut și în DIRASSAT-Revue Periodique, avril 2008)
- [35] JUDE, S., A., FRIGURĂ-ILIASA, F. M., EHEGARTNER, P., ANDEA, P., "Finite Element Modeling for Two ZnO Based Varistors Connected in Parallel and Thermally Coupled" Proceedings 3rd International Conference "From Scientific Computing to Computational Engineering 3rd IC-SCCE, Athens, Greece 9-12 July, 2008, paper 079
- [36] KNIGHT, J., K., LEVESTON, N., "An Experimental Evaluation of the Assumptions of Independence in Multiversion Programming". IEEE Transactions on software engineering, Vol. SE - 12, PP. 96-109, Jan. 1986
- [37] KNUTH, D., E., "Tratat de programare a calculatoarelor. Algoritmi fundamentali", Editura Tehnică, București, 1974
- [38] LANDAN, I., D., "Identificarea și comanda sistemelor", Editura Tehnică, București, 1997
- [39] LAUGHTON, M., A., "Expert Application in Power Systems", Prentice Hall International, 1990
- [40] LEONTE, P., "Aparate electrice", Vol. I și II, I.P. Iași, 1983
- [41] LITTLEWOOD, B., "Stochastic Reliability Growth, A Model for Fault Removal in Computer Programs and Hardware Designs", IEEE Transactions on Reliability, PP. 313 - 320, 1981

- [42] MANOLESCU, P., "Măsurări electrice și electronice", Editura Didactică și Pedagogică, București, 1980
- [43] MĂRZA, E., "Radiocomunicații mobile", Editura Orizonturi Universitare, Timișoara, 2001
- [44] MILLOT, P., "Supervision des procédés automatisés et ergonomie", Editura Hermes, Paris, 1988
- [45] MILLOT, P., "Configurations homme-machine dans les procédés automatisés", Editura Octares, Marseille, Paris, 1990
- [46] MOGA, M., "Sisteme inteligente pentru conducerea rețelelor electrice de distribuție", Editura AGIR, București, 2000
- [47] NAGEL, P., SKRIVAN, J., "Software reliability: Repetitive Run Experimentation and Modelling" NASA Contractor Report, 165836, feb. 1982
- [48] NIȚU, V., I., ș.a. "Echipamente electrice și electrotehnice de automatizare", Editura Didactică și Pedagogică, București, 1983
- [49] PĂUNESCU, F., ș.a., "Sisteme cu prelucrare distribuită și aplicațiile lor", Editura Tehnică, București, 1983
- [50] PĂUNESCU, F., GOLIȘTEAN, D., "Sisteme cu prelucrare distribuită și aplicațiile lor", Editura Tehnică, București, 1993
- [51] PETRESCU, S., ș.a., "Termotehnică și mașini termice", Editura Didactică și Pedagogică, București, 1978
- [52] POLICEC, A., MĂRZA, E., "Telecomunicații", Editura Orizonturi Universitare, Timișoara, 2001
- [53] POP, E., ș.a., "Măsurări în Energetică", Editura Facla, Timișoara, 1981
- [54] POPA, B., "Termotehnică și măsurări tehnice", Editura Didactică și Pedagogică, București, 1972
- [55] POPESCU, D., ș.a., "Automatizări industriale", Editura Tehnică, București, 1994
- [56] SIMA, V., VARGA, A., "Practica optimizării asistate de calculator", Editura Tehnică, București, 1986
- [57] STANCIU, D., "Senzori - Prezent și perspectivă", Editura Tehnică, București, 1987
- [58] SUCIU, I., "Bazele echipamentelor electrice", Editura Facla, Timișoara, 1980
- [59] TERTIȘCO, M., ș.a. "Identificarea asistată de calculator a sistemelor", Editura Tehnică, București, 1987
- [60] TODEREAN, G., ș.a. "Transputere și procesoare de semnal", Editura Microinformatica, Cluj-Napoca, 1993
- [61] TODEREAN, G., ș.a. "Rețele neuronale artificiale", Editura Albastră, Cluj-Napoca, 1995
- [62] TODEREAN, G., ș.a. "Rețele neuronale", Editura Microinformatica, Cluj-Napoca, 1994
- [63] TRIFU, R., BALOG, A., "Testarea și evaluarea produselor program", Rev. PCWorld, nr. 6/1995, pag. 52
- [64] TRUȘCĂ, V., POPESCU M., "Tehnologia de fabricație a aparatelor electrice", Editura ICPE, 1996
- [65] VASILIEVICI, Al., BĂLAȘIU, F., ș.a. "Implementarea echipamentelor digitale de protecție și comandă pentru rețele electrice", editura Tehnică, București, 2000
- [66] VASILIEVICI, Al., FRIGURĂ – ILIASA, F., "Exemple de implementare a automatelor programabile (PLC) în instalațiile de comandă ale marilor consumatori industriali", Analele Universității din Oradea, Fascicola Energetică, Vol. I, pag. 56-64, Oradea, 2000
- [67] * * * Automatizări și instrumentație; Anul X, 3/2001
- [68] * * * Automatizări și instrumentație; Anul X, 4/2001
- [69] * * * Automatizări și instrumentație; Anul X, 6/2001
- [70] * * * HIDROTIM S.A., Diverse documente tehnice, 2000/2008

SINTEZĂ PRIVIND LUCRĂRILE PROPRII

- [1] FRIGURĂ – ILIASA, F.M., MÂȚIU-IOVAN L., VĂȚĂU, D., JUDE S.,A., "A Few Aspects Concerning the Thermal Finite Element Modelling of a Low Voltage ZnO Based Varistor with an Additional Mass" Proceedings 3rd International Conference "From Scientific Computing to Computational Engineering 3rd IC-SCCE, Athens, Greece 9-12 July, 2008, paper 078
- [2] JUDE, S.,A., EHEGARTNER, P., ANDEA, P., FRIGURĂ-ILIASA, F.M., "New Techniques for High Voltage Equipment Maintenance" Proceedings of the 10th International Symposium "Materials, Methods and Technologies", 2-6 iunie 2008, Sunny Beach Resort-Burgas, Bulgaria, pp.100-107
- [3] JUDE, S.,A., EHEGARTNER, P., ANDEA, P., FRIGURĂ-ILIASA, F.M., „Considerations about the Thermal Stability Evaluation for Low Voltage MOV Surge-Arresters" Proceedings of the 2nd International Conference on Electrical and Electronics Engineering, ICEEE 08, Laghouat, 21-23 aprilie 2008, Algeria, pag 322 – 327, (apărut și în DIRASSAT-Revue Periodique, avril 2008)
- [4] JUDE, S.,A., FRIGURĂ-ILIASA, F.M., EHEGARTNER, P., ANDEA, P., "Finite Element Modeling for Two ZnO Based Varistors Connected in Parallel and Thermally Coupled" Proceedings 3rd International Conference "From Scientific Computing to Computational Engineering 3rd IC-SCCE, Athens, Greece 9-12 July, 2008, paper 079
- [5] EHEGARTNER P., JUDE A., ANDEA P., VATAU D., F.M. FRIGURA-ILIASA, A Model Concerning the High Voltage Systems Impact on the Environment inside a Romanian Power Substation, Proceedings of the 11th WSEAS International Conference on Automatic Control, Modelling and Simulation (ACMOS 09), Istanbul, Turkey, 30 mai – 1 iunie 2009, WSEAS Press, pp. 419-424
- [6] A. JUDE, P. EHEGARTNER, P. ANDEA, D. VATAU, F.M. FRIGURA-ILIASA, Power Quality Control on the Romanian Energy Market, Proceedings of the 11th WSEAS International Conference on Automatic Control, Modelling and Simulation (ACMOS 09), Istanbul, Turkey, 30 mai – 1 iunie 2009, WSEAS Press, pp. 413-418