

MINISTERUL EDUCATIEI SI INVATAMINTULUI  
UNIVERSITATEA TEHNICA TIMISOARA  
FACULTATEA DE MECANICA  
FACULTATEA DE ELECTRONICA SI TELECOMUNICATII

ING. BOGDANOV IVAN

OPTIMIZAREA SISTEMELOR AUTOMATE DE CONDUCERE  
A ACTIONARILOR ELECTRICE PENTRU ROBOTI INDUSTRIALI

TEZA DE DOCTORAT

CONDUCATOR STIINTIFIC  
PROF.DR.ING.FRANCISC KOVACS

BIBLIOTECA CENTRALĂ  
UNIVERSITATEA "POLITEHNICA"  
TIMIȘOARA

T I M I S O A R A  
1 9 9 2

MSH...  
568331  
191 11  
tamexa  
111  
568.33  
7

BUP



## CUPRINS

	Pag.
Capitolul 1. INTRODUCERE .....	1
Capitolul 2. PROBLEMA CONDUCERII ROBOTILOR INDUSTRIALI .....	8
2.1. Generalități. Problematika abordată în teza de doctorat .....	8
2.2. Aspectele problemei conducerii roboților industriali. Domeniul de preocupare și obiectivul tezei de doctorat .....	10
2.3. Conducerea nemijlocită a axelor roboților cu acționare electrică. Stadiul actual de dezvoltare a sistemelor de acționare electrică .....	16
2.4. Sisteme de acționare "exclusiv numerice" pentru conducerea roboților industriali, domeniu actual de cercetare .....	21
Capitolul 3. SISTEME CU MICROPROCESOARE UTILIZATE ÎN COMANDA ACȚIONARILOR ELECTRICE .....	24
3.1. Arhitectura standard a unui sistem cu microprocesor utilizat în comanda circuitelor de acționare electrică .....	24
3.2. Exemple de sisteme de comandă cu microprocesoare uzuale .....	30
3.2.1. Sisteme de comandă cu microprocesorul 8035 .....	31
3.2.2. Sistem de dezvoltare cu microprocesorul 8085 .....	34
3.2.3. Sistem de comandă cu microprocesorul Z80 .....	37
Capitolul 4. UTILIZAREA MICROPROCESOARELOR ÎN COMANDA ACȚIONARILOR CU MOTOARE DE CURENT CONTINUU .....	38
4.1. Probleme specifice cu privire la acționările cu mașini de curent continuu .....	38
4.2. Sisteme de acționare cu motoare de curent continuu alimentate de la redresoare comandate .....	41
4.2.1. Implementarea pe microsisteme a circuitelor de comandă pentru redresoare comandate .....	45

4.2.2. Exemple de aplicații pentru comanda redresoarelor cu microprocesor .....	46
4.2.2.1. Redresor monofazat în punte semicomandat cu microprocesor .....	46
4.2.2.2. Redresor trifazat în punte complet comandat cu microprocesor .....	54
4.3. Sisteme de acționare cu motoare de curent continuu alimentate prin variatoare de tensiune continuă .....	60
4.3.1. Circuite de comandă pentru choppere realizate cu microprocesor .....	65
4.3.2. Exemple de choppere comandate cu microprocesor .....	65
4.3.2.1. Chopper cu stingere forțată comandat cu microprocesor .....	65
4.3.2.2. Chopper cu tranzistoare comandat cu microprocesor .....	74
4.4. Exemple de sisteme de acționare cu motor de curent continuu comandate numeric .....	76
4.4.1. Principiul modulării în lățime de puls (PWM) .....	76
4.4.2. Sistem de acționare PWM "exclusiv numeric" .....	78
4.4.3. Echipamentul pentru conducerea mișcărilor CREONICS VME BUS MCC .....	80
Capitolul 5. SISTEM DE REGLARE NUMERICA A VITEZEI SI POZITIEI INTR-O ACȚIONARE CU MOTOR DE CURENT CONTINUU COMANDATA CU MICROPROCESOR .....	85
5.1. Structura sistemelor de reglare utilizate .....	85
5.1.1. Tipuri de structuri pentru sisteme de reglare automată a vitezei și poziției .....	85
5.1.2. Schemele bloc funcționale ale sistemelor de reglare automată proiectate .....	87
5.1.3. Procesul reglat .....	89
5.1.3.1. Modelul matematic operațional al MCC .....	89
5.1.3.2. Modelul matematic al elementului de execuție .....	90
5.1.3.3. Modelul matematic al traductorului TIRO și al numărătorului ....	90

5.1.4. Echipamentul de comandă numerică (ECN) .....	94
5.2. Proiectarea algoritmilor de reglare numerică ...	94
5.2.1. Proiectarea algoritmică a unui sistem convențional de reglare automată numerică .....	94
5.2.2. Discretizarea modelelor matematice ale elementelor de transfer continue .....	99
5.2.3. Proiectarea algoritmică a SRA în cascadă .....	100
5.2.4. Proiectarea concretă a SRA-V .....	100
5.2.5. Proiectarea SRA-VP .....	104
5.3. Implementarea algoritmului de reglare numerică a vitezei .....	107
5.3.1. Forma discretă finală pentru ARN-V .....	107
5.3.2. Sarcinile ECN. Structura ARN-V .....	108
5.3.3. Principalele subprograme constituente ale ARN-V .....	110
5.3.3.1. Subrutina de tratare a întreprerilor de la timer .....	110
5.3.3.2. Subrutina MASURA .....	112
5.3.3.3. Subrutina ARN. Calculul mărimii de comandă .....	113
5.3.3.4. Dialogul on-line cu operatorul uman - subrutina INTERRUPT-DAF .....	119
5.3.3.5. Dialogul off-line cu operatorul uman .....	119
5.3.4. Considerații asupra timpului mort total al buclei de reglare .....	120
5.3.5. Experimentarea SRA-V .....	121
5.4. Implementarea algoritmului de reglare numerică a vitezei și poziției .....	128
5.4.1. Funcțiile îndeplinite de microsistem. Structura programului ARN-VP .....	129
5.4.2. Principalele subprograme ale ARN-VP .....	131
5.4.2.1. Subrutina de tratare a întreprerilor de la timer .....	131
5.4.2.2. Subrutina SBR-2 pentru generarea intervalelor de conducție a chopperului ...	131

5.4.2.3. Preluarea poziției curente de la numărător .....	132
5.4.2.4. Subrutina SBR-1 - calculul mărimii de comandă .....	132
5.4.2.5. Dialogul on-line cu operatorul uman. Subrutina INTERRUPT-DAF .....	137
5.4.3. Experimentarea SRA-VP .....	137
5.5. Controlul curentului din înfășurarea rotorică a motorului .....	138
Capitolul 6. SISTEME DE REGLARE NUMERICA A VITEZEI PENTRU TREI MOTOARE DE CURENT CONTINUU CU UN SINGUR MICROPROCESOR DE COMANDA .....	149
6.1. Structura sistemului de reglare automată .....	149
6.2. Scuația discretă corespunzătoare algoritmului de reglare a vitezei .....	151
6.3. Funcțiile îndeplinite de microsistem .....	152
6.4. Gestiunea timpului real .....	154
6.5. Măsurarea vitezei curente de rotație .....	161
6.6. Dialogul cu sistemul ierarhic superior .....	164
6.7. Subrutina de calcul a algoritmilor de reglare numerică ARN .....	166
6.8. Subrutinele matematice .....	172
6.8.1. Subrutina DIF .....	172
6.8.2. Subrutina MLT1 .....	174
6.8.3. Subrutina MLT2 .....	175
Capitolul 7. SISTEM DE REGLARE NUMERICA A POZITIEI INTR-O ACTIONARE CU MOTOR DE CURENT CONTINUU FOLOSIND ALGORITME DE REGLARE MODALA ALUNECĂTOARE ...	176
7.1. Introducere .....	176
7.2. Analiza reglării modale-alunecătoare (RMA) .....	178
7.2.1. Principiul reglării modale alunecătoare .	178
7.2.2. Considerații generale privind proiectarea algoritmilor de reglare modală alunecătoare (ARMA) .....	180
7.2.3. Metode comenzii echivalente pentru proiectarea ARMA .....	180
7.3. Variante de ARMA .....	182
7.3.1. Viteza absolută a variabilei de comutație .....	182

7.3.2. Interconectarea algoritmilor de reglare modală alunecătoare cu algoritme de reglare PI .....	183
7.3.3. ARMA cu compensarea perturbațiilor .....	184
7.4. Analiza ARMA cu viteză absolută constantă a variabilei de comutație pentru reglarea poziției într-o acționare cu motor de curent continuu .....	185
7.4.1. Descrierea generală a algoritmului .....	185
7.4.2. Analiza fazei de atingere a regimului modal alunecător .....	186
7.4.3. Analiza fazei de regim modal alunecător .....	187
7.4.4. Analiza modului alunecător real .....	189
7.5. Aspecte privind acordarea ARMA .....	190
7.6. Proiectarea algoritmilor de reglare modal- alunecătoare a poziției pentru un sistem de acționare cu motor de curent continuu .....	195
7.6.1. Modelul matematic al procesului extins .....	197
7.6.2. Proiectarea algoritmului de reglare modală alunecătoare ARMA-1 în condițiile neglijării inerției în producerea acțiunii ponderomotoare .....	198
7.6.2.1. Modelul matematic al procesului condus extins, în condițiile neglijării inerției la producerea acțiunii ponderomotoare .....	198
7.6.2.2. Proiectarea ARMA-1 fără estimarea perturbațiilor .....	200
7.6.3. Introducerea estimatorului de perturbații .....	202
7.6.4. Proiectarea ARMA-2 în condițiile luării în considerare a inerției la producerea acțiunii ponderomotoare .....	203
7.7. Simularea cu ajutorul calculatorului a ARMA ....	206
7.7.1. Simularea procesului condus extins în condițiile neglijării producerii acțiunii ponderomotoare PCE-1 .....	206

7.7.2. Simularea procesului condus extins în condițiile luării în considerare a inerției producerii acțiunii ponderomotoare PCE-2 .....	208
7.7.3. Programul de simulare .....	208
7.7.4. Simularea ARMA-1 cu PCE-1 .....	210
7.7.5. Simularea ARMA-1 cu PCE-2 .....	212
7.7.6. Simularea ARMA-2 cu PCE-2 .....	213
7.8. Implementarea algoritmului de reglare modală alunecătoare pe model experimental .....	214
7.8.1. Executivul de timp real ETR .....	215
7.8.1.1. Funcțiile programului .....	215
7.8.1.2. Sincronizarea execuției funcțiilor .....	216
7.8.1.3. Structura programului .....	218
7.8.1.4. Preluarea poziției curente de la numărător .....	220
7.8.1.5. Preluarea referinței de poziție de la operatorul uman .....	220
7.8.2. Algoritmul de reglare ARMA-1 .....	221
7.8.3. Algoritmul de reglare ARMA-2 .....	224
7.8.4. Algoritm de reglare combinat ARMA+PI ....	227
7.8.5. Testarea prin simulare a programelor de implementare a ARN .....	229
7.8.6. Testarea pe model experimental a algoritmilor de reglare modală alunecătoare ....	230
7.8.6.1. Interpretorul de comenzi .....	231
7.8.6.2. Generarea tabelelor .....	233
7.8.6.3. Rularea programului de implementare a ARN în regim de testare ....	236
7.8.6.4. Afișarea tabelelor .....	239
Capitolul 8. CONCLUZII SI CONTRIBUTII .....	242
8.1. Concluzii și direcții de cercetare .....	242
8.2. Contribuții .....	256
BIBLIOGRAFIA .....	260
ANEXA A1. Program de reglare numerică a turației unui motor de curent continuu. Varianta chopperului cu tiristoare .....	A1.0-A1.13



- ANEXA A2. Program de reglare numerică a turației  
unui motor de curent continuu. Varianta  
chopperului cu tranzistoare ..... A2.0-A2.17
- ANEXA A3. Program de reglare numerică a vitezei și  
poziției pentru un sistem de acționare cu  
motor de curent continuu ..... A3.0-A3.12
- ANEXA A4. Program de reglare numerică a vitezei pentru  
trei sisteme de acționare cu motor de curent  
continuu independente comandate de un  
singur microprocesor uzual ..... A4.0-A4.18
- ANEXA A5. Programele de simulare pe calculator a  
metodei de reglare numerică modal-alunecătoare  
a poziției aplicată într-un sistem de  
acționare cu motor de curent continuu ..... A5.0-A5.8  
A5.1. Program cu 82 de pași de simulare  
A5.2. Program cu 847 de pași de simulare
- ANEXA A6. Rezultate experimentale obținute prin  
simulare pentru metoda de reglare numerică  
modal-alunecătoare aplicată la un sistem de  
acționare cu motor de curent continuu ..... A6.0-A6.39
- ANEXA A7. Programul de implementare concretă a metodei  
de reglare numerică modal-alunecătoare a  
poziției pentru un sistem de acționare cu  
motor de curent continuu ..... A7.0-A7.24  
A7.1. Executivul de timp real (ETR) ..... A7.1-A7. 9  
A7.2. Subprogramul de implementare a  
algoritmului ARMA-1 cu neglijarea inerției  
producerii acțiunii ponderomotoare ..... A7.10-A7.14  
A7.3. Subprogramul de implementare a  
algoritmului ARMA-2 cu luarea în considerare  
a inerției producerii acțiunii  
ponderomotoare ..... A7.15-A7.20  
A7.4. Subprogramele din biblioteca  
matematică ..... A7.21-A7.24
- ANEXA A8. Programul de implementare concretă a procedurii  
de utilizare a algoritmilor de reglare  
numerică în mod combinat ..... A8.0-A8. 9  
A8.1. Algoritm de reglare numeric clasic PI . A8.1-A8. 4

A8.2. Combinația algoritmilor ARMA-1 și ARN-PI .....	A8.5-A8.7
A8.3. Combinația algoritmilor ARMA-2 și ARN-PI .....	A8.8-A8.9
ANEXA A9. Programul TEST de implementare și testare pe model experimental a algoritmilor de reglare clasice și modal alunecătoare .....	A9.0-A9.22
ANEXA A10. Rezultate experimentale obținute cu programul TEST implementat pe model experimental .....	A10.0-A10.3
ANEXA A11. Program de verificare pe calculator a subrutinei de implementare în limbaj de asamblare a algoritmului ARMA-1 .....	A11.0-A11.9
A11.1. Programul principal - în limbaj FORTRAN .....	A11.1-A11.3
A11.2. Subprogramul ARMA-1 - în limbaj de asamblare .....	A11.4-A11.9
ANEXA A12. Rezultatele verificării pe calculator a subprogramelor de implementare în limbaj de asamblare a algoritmului de reglare ARMA-1 ....	A12.0-A12.3

## CAPITOLUL 1.

### INTRODUCERE

Pe măsura dezvoltării științei și tehnicii, procesele de producție și instalațiile tehnologice, în care acestea au loc, au înregistrat o transformare continuă, reflectată în reducerea treptată a muncii fizice a omului, concomitent cu creșterea ponderii muncii intelectuale pentru dirijarea sau conducerea mașinilor, dispozitivelor și instalațiilor respective, în condițiile creșterii productivității muncii /25 /.

Principala cale pe care progresul tehnic contribuie la creșterea eficienței muncii în industrie este mecanizarea și automatizarea /55 /. În condițiile tehnico-științifice contemporane, acest lucru înseamnă, înainte de toate, mecanizarea avansată și automatizarea flexibilă a proceselor de fabricație /53 /.

În producția industrială cu specific de construcție de mașini, în care domină operațiile tehnologice - prelucrări mecanice, asamblări, depozitări, ambalări, sudări, vopsiri - realizarea unei eficiențe economice ridicate pe calea automatizării producției creează o serie de probleme tehnice. Acestea sînt specifice modului de desfășurare a procesului tehnologic și de organizare a producției și sînt dificil, uneori chiar imposibil, de rezolvat prin utilizarea sistemelor clasice de automatizare folosite în procesele de producție cu caracter continuu (ca de exemplu energetică, chimie) sau semicontinuu, datorită, în principiu, lipsei de flexibilitate a acestor sisteme. Dificultățile sînt determinate de specificul proceselor industriale de tip construcții de mașini, care presupun în special operații diverse - de manipulare a unor piese și scule - , de necesitatea trecerii frecvente de la fabricarea unui produs la fabricarea altui produs cu aceleași utilaje tehnologice, de necesitatea efectuării unor operații de control tehnic de calitate interfațate și a sortării produselor etc. /25 /.

Aspectele mai sus menționate, la care se adaugă cele două obiective majore de creștere, pe de o parte, a eficienței economice a producției și de reducere a activității nemijlocit umane în sectoare cu dificultate și/sau monotonie marcanta, au condus, în cele din urmă, la apariția roboților industriali și a automatizării flexibile.

De la primele aplicații ale instalațiilor de manipulare programabilă s-a scurs o perioadă în care progresul în concepția și realizarea roboților și oportunitățile pe care industria le-a oferit automatizării flexibile s-au alimentat reciproc, accelerându-se pe de o parte ritmul construcției roboților, iar pe de altă parte diversitatea aplicațiilor.

Robotul este un produs "mechatronic", combinând tehnologia mecanică cu cea electronică, iar robotizarea industriei este etapa care urmează firesc mecanizării și automatizării. Robotul este o componentă evoluată de automatizare, care combină electronica de tip calculator cu sisteme avansate de acționare pentru a realiza un echipament independent de mare flexibilitate /53 /, /66 /.

Un studiu efectuat în industria constructoare de mașini în SUA /69 / în 1980 arată că "timpul de parcurgere" al unui produs - de la extragerea materialului/semifabricatelor din magazie până la expedierea produsului - este constituit din circa 5% timp de prelucrare efectivă și 95% timp de depozitare, transport uzinal, manipulare, așteptare etc. Ca urmare se întrevade o creștere mare a eficienței prin reducerea "timpului de parcurgere", deci implicit modernizarea logisticii (depozitare/regăsirea obiectelor de lucru, transportul uzinal) și a manipularii obiectelor și sculelor în cadrul unor procese tehnologice flexibile. În consecință, creșterea productivității muncii nu se poate îndeplini fără extinderea importantă a proceselor și sistemelor de fabricație flexibilă mecanizate avansate (cu folosirea manipuletoarelor) respectiv automatizate. Automatizarea eficientă a producției se poate realiza doar cu ajutorul unei automatizări suplă, în cadrul unor celule și linii de fabricație flexibile, bazate, în principal, pe mașini unelte cu comandă numerică și roboți industriali.

Uzina viitorului se va baza pe celule flexibile de fabricație deservite de roboți industriali sub controlul și conducerea micro și minicalculatoarelor electronice. Operațiile de asam-

blare-montaj, precum și cele de reglaj final, de asigurare a calității și de întreținere vor fi asigurate tot de roboți industriali. Intreaga funcționare va fi condusă de o rețea de calculatoare, care vor asigura atât coordonarea cât și optimizarea fluxurilor de fabricație. Vor fi automatizate la nivel foarte ridicat și operațiile de proiectare a produselor, de pregătire tehnică și tehnologică a fabricației. Nu se prevede în nici un caz excluderea omului din aceste linii de fabricație ale viitorului, cerându-i-se acestuia un nivel superior de pregătire, pentru a realiza conducerea procesului de producție de la terminale de calculator și pentru a asigura buna funcționare a unor utilaje complexe prin activități de întreținere preventivă, depanare și reparare.

Robotul modern este un sistem complex, implementat pe calculatoare, microprocesoare, senzori, sisteme de acționare, structuri mecanice, care are capacitate de acțiune, de percepere, de decizie și de comunicare /25 /.

Ramura de știință care se numește "Robotică" sau mai larg "Sisteme de fabricație flexibilă" (Flexible Manufacturing Systems - FMS) este prin excelență multidisciplinară îmbinând într-un ansamblu fascinant cunoștințe din Teoria mecanismelor, Dinamica sistemelor mecanice mobile, Dispozitive și scule, Tehnologia construcțiilor de mașini, Acționări (electrice, hidropneumatice), Electronică, Automatică, Construcție și programare a calculatoarelor, Organizarea întreprinderilor industriale, Management, Ergonomie și multe altele /55 /.

Această pluridisciplinaritate rezidă din însăși structura bloc informațională a unui robot industrial, a unei instalații pentru operații humanoide în general, structură compusă din trei sisteme distincte integrate organic și funcțional într-o corelație și corespondență continuă și numai principial divizibilă (vezi fig.1.1).

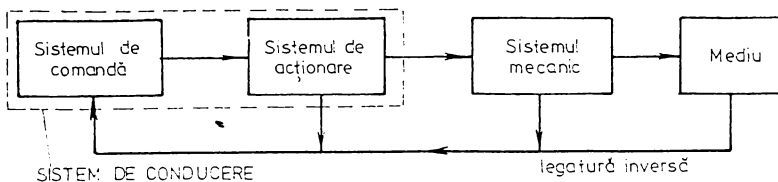


Fig.1.1.Schema bloc informațională a unei instalații pentru operații humanoide

Înțelegînd în sistemul de comandă unul sau mai multe calculatoare electronice, în sistemul de acționare un ansamblu de echipamente electrice și/sau hidropneumatice, în sistemul mecanic un complex de mecanisme și dispozitive din ce în ce mai sofisticate în ultimul timp, iar prin mediu întregul ansamblu implicat de organizare, tehnologia și condiții specifice, urmărite cu tractoare și senzori adecvați, se obține o imagine rezumativă dar plină de semnificații cu privire la ceea ce înseamnă robotica sau meseria, cu totul specială, de robotician.

Teza de doctorat de față își propune să abordeze în detaliu numai anumite aspecte legate de sistemul de acționare, cu scopul de a prezenta contribuții ale autorului în domeniul optimizării sistemelor de acționare electrică pentru roboți industriali, încercînd să încadreze aceste realizări în spațiul atât de amplu și complex, cu posibilități de perfecționare și modernizare practic nelimitate, oferit de cercetările din domeniul roboticii.

Acționarea electrică a roboților industriali, a instalațiilor pentru operații humanoide în general, devine astăzi cea mai răspîndită în aplicații. Acest fapt se explică prin cîteva argumente indiscutabile:

1. Disponibilitatea cvasigenerală a energiei electrice în majoritatea covârșitoare a mediilor de acțiune ale roboților industriali.
2. Robustețea motoarelor electrice.
3. Posibilitățile simple de racordare a motoarelor la rețeaua de distribuție a energiei electrice.
4. Performanțele dinamice deosebite ce se pot obține.
5. Randament mare.
6. Fiabilitate ridicată.
7. Preț de cost scăzut.
8. Posibilitatea interfațării directe, fără conversii ale formei de energie, între echipamentul de comandă (electronic) și echipamentul de execuție (electric) ș.a. /54 /, /95/.

În ultimii 10-15 ani, apariția și răspîndirea microprocesoarelor a determinat o extindere a ariei de preocupare a specialiștilor din cele mai diverse domenii de activitate. Cunoașterea microprocesoarelor și a programării tinde să devină astăzi o chestiune de cultură generală /62,100/, iar domeniile de aplicație pentru micro sisteme se consideră limitate numai de imaginația utilizatorilor /99/.

În condițiile amintite, s-au extins și aplicațiile industriale ale microprocesoarelor. Cele mai potrivite situații de implementare cu microprocesor a circuitelor de comandă și control, pentru procese industriale, par să fie acelea în care numărul de unități de proces care trebuie supravegheate, comandate sau reglate nu depășește câteva zeci (10-20 tipic). Pentru un număr mic de unități de proces, este mai ieftină utilizarea rețelelor, tranzistoarelor, a circuitelor electronice clasice /100/. Dacă numărul unităților de proces este mare, sute sau mii, apare necesitatea de a folosi minicalculatoare sau calculatoare de proces, deși și acestea sînt treptat înlocuite cu microprocesoare din ce în ce mai performante.

Din punctul de vedere al complexității impuse de comanda circuitelor de acționare electrică, utilizarea, în structura lor, a sistemelor cu microprocesoare apare ca fiind cea mai îndreptățită întrucît circuitele de acționare implică o complexitate medie în ceea ce privește numărul de comenzi și operații necesare. Acest aspect se evidențiază și în cazul sistemelor de acționare electrică din sistemele de conducere pentru roboți industriali.

În condițiile în care problema sintetizării circuitelor de comandă pentru sistemele de acționare în variante electronice tradiționale, analogice sau numerice, este generalizată, introducerea microprocesoarelor impune schimbarea conceptelor de proiectare și de sinteză a secvențelor de comandă în vederea obținerii de avantaje incontestabile.

Utilizarea microprocesoarelor în sistemele de acționare se justifică nu numai din punctul de vedere al complexității operațiilor și comenzilor necesare ci și dintr-o serie de alți factori cum sînt /99, 100, 17, 62, 24, 85, 61, 75, 76, 41/:

1. Micșorarea timpului de răspuns al circuitului de comandă și realizarea de circuite de acționare rapidă.

2. Creșterea preciziei și siguranței în lucru, prin însăși folosirea unui sistem de calcul ca circuit de comandă.

3. Asigurarea flexibilității și posibilității de a extinde numărul de elemente controlate și de a efectua modificări și corecții asupra funcției de comandă prin simpla schimbare a unui program și menținerea neschimbată a structurii hardware.

4. Reducerea numărului de componente din circuitele de comandă, în situația în care și prețul de cost al circuitelor LSI a scăzut simțitor.

5. Posibilitatea subordonării directe a circuitelor de comandă ale sistemelor de acționare calculatoarelor ierarhic superioare.

6. Îmbunătățirea fiabilității și asigurarea unei mai bune insensibilități la perturbații pentru circuitele de comandă, fiind posibilă includerea de programe de test sau autodepanare.

7. Scăderea consumului de energie.

Având în vedere considerentele de mai sus, spre sfârșitul anilor '70 și începutul anilor '80, studiile, cercetările și aplicațiile care includ microprocesoarele în circuite de comandă în acționările electrice au cunoscut un avânt deosebit.

Cercetările s-au orientat atât în sensul folosirii microprocesoarelor în cele mai diverse circuite de acționare, cât și în cel al utilizării de diferite familii de microprocesoare, începând cu microprocesorul microsistem într-o singură capsulă și pînă la cele mai noi și mai puternice microprocesoare de astăzi /85, 47, 27/.

O direcție importantă de cercetare este aceea de a studia posibilitățile de utilizare cât mai eficientă a structurilor cu microprocesoare uzuale în sistemele de acționare electrică /19, 38/. Abordarea problemei implementării pe microsisteme uzuale a circuitelor de comandă pentru acționări electrice, determinată în principal de dorința de a evita utilizarea procesoarelor și procesoarelor specializate sau dedicate, permite ca prin efort de inteligență să se obțină sisteme bune de comandă la prețuri și disponibilități de aprovizionare accesibile.

În poșta realizărilor multiple în domeniu, problemele apărute sînt departe de a fi fost elucidate. Cercetările în curs vizează creșteri de viteză și precizie, simplificări în sistemele de comandă, optimizări din punctul de vedere al structurilor hardware și software, atât în domeniul acționării motoarelor de curent continuu cât și în acționările cu motoare de curent alternativ /102, 103, 107, 104, 105, 56/.

Conducerea numerică a roboților industriali a generat cerința adaptării elementelor componente ale sistemelor de acționare (compus din sistemele de reglare afarante fiecărei axe), la tehnica numerică. Înlocuirea elementelor analogice din structura sistemelor de acționare oferă, pe lîngă asigurarea caracterului unitar al prelucrării informației, următoarele avantaje /67/:

1. Simplificarea structurii echipamentelor de reglare prin eliminarea reguletoarelor analogice, convertoarelor numeric-ana-



logice și tahogeneratoarelor.

2. Creșterea fiabilității și reducerea substanțială a influenței condițiilor de mediu și a dispersiei tehnologice asupra performanțelor statice și dinamice ale sistemului de reglare.

3. Îmbunătățirea caracterului adaptiv autoacordabil al reguletoarelor.

4. Micșorarea semnificativă a prețului de cost.

Factorii care limitează performanțele sistemelor de conducere nemijlocită a axelor "exclusiv numerice" /67 / derivă din necesitatea discretizării în durată și amplitudine a semnalelor. Discretizarea în durată afectează mai ales comportarea dinamică, iar discretizarea în amplitudine influențează precizia de reglare. Utilizarea unor algoritme de reglare adecvate și, eventual, a unor procesoare rapide, precum și extinderea lungimii cuvântului binar prin care se exprimă mărimile ce intervin în procesul de reglare, contribuie la îmbunătățirea indicatorilor de calitate ai sistemului de reglare.

Obiectivul tezei de doctorat a fost acela de a concretiza eforturile de cercetare ale autorului în domeniul comenzii cu microprocesor a sistemelor de acționare cu motoare electrice de curent continuu. S-a urmărit utilizarea microprocesoarelor de cost redus și uz general, cu intenția de a se obține și cu acestea performanțe competitive și, implicit, avantajelor mai sus menționate. Teza de doctorat se dorește a deveni o contribuție la realizarea sistemelor de acționare comandate exclusiv numeric, în vederea optimizării sistemelor de acționare pentru roboți industriali.

Teza de doctorat elaborată prezintă în prima sa parte câteva principii și metode specifice de abordare a problematicii utilizării microprocesorului în comanda sistemelor de acționare cu motoare electrice de curent continuu, cu contribuții și realizări, până la nivel experimental, din partea autorului. În cea de a doua parte sînt prezentate sisteme de acționare cu motor de curent continuu originale, concepute și realizate concret în variante exclusiv numerice. Se analizează performanțele obținute și se indică posibilitățile de implementare a acestor realizări în echipamentele de comandă pentru roboți industriali.

## CAPITOLUL 2.

### PROBLEMA CONDUCERII ROBOTILOR INDUSTRIALI

#### 2.1. Generalități. Problematika abordată în teza de doctorat

Robotul modern este un sistem complex, implementat pe structuri mecanice avansate, (micro) calculatoare, sisteme de acționare perfecționate și este dotat cu diverse categorii de senzori. În acest fel robotului i se conferă capacitate de acțiune, de percepere, de decizie și de comunicare cu mediul și operatorul uman.

Din punct de vedere informațional, structura bloc a unui robot industrial, a unei instalații pentru operații humanoide în general, se compune din trei sisteme integrate organic și funcțional într-o corelație și intercomunicație continuă și numai principial divizibilă (vezi fig.1.1, cap.1).

Sistemul mecanic este sistemul condus din structura robotului.

Sistemul de acționare are rolul de a transforma o anumă formă de energie potențială (hidraulică, pneumatică, electrică) în energie mecanică și de a transmite, sub control automat, mișcarea mecanică la cuplurile cinematice conducătoare ale lanțului cinematic deschis ce constituie sistemul mecanic.

Sistemul de comandă, realizat în jurul unui sistem de calcul, are drept scop prescrierea succesiunii și parametrilor mișcărilor lanțului cinematic, în funcție de condițiile impuse în realizarea unui anumă proces și de reacțiile sesizate în mediu ca urmare a acțiunii robotului asupra acestuia.

Sistemul de comandă și de acționare constituie sistemul de conducere al robotului industrial. Privit din punctul de vedere al circuitului informațional structura bloc a unui sistem de conducere poate fi redusă, principial, la cea redată în fig.2.1 /87,7/.

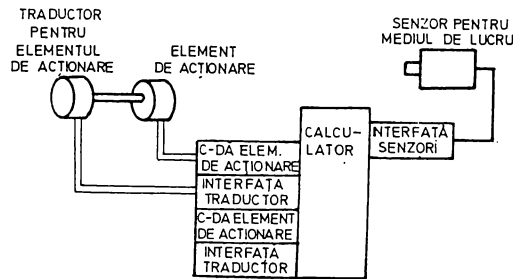


Fig.2.1. Componentele structurii bloc de principiu a unui sistem de conducere al unui robot industrial.

În această structură bloc sistemul de comandă este numit calculator, întrucît majoritatea funcțiilor ce trebuie să le realizeze sînt similare celor executate de calculatoarele numerice. Cu toate acestea, în anumite cazuri unele funcțiuni pot fi îndeplinite de circuitele hardware specializate, iar în alte situații sarcinile de comandă sînt îndeplinite de mai multe microcalculatoare interconectate.

Pentru fiecare cuplă cinematică conducătoare din lanțul cinematic, ce constituie sistemul mecanic, este prevăzut cîte un element de acționare (electric, hidraulic sau pneumatic).

Urmărirea situației din mediu este realizată cu senzori și traductoare, care comunică, prin interfețe adecvate, cu sistemul de conducere.

Sistemului de conducere al unui robot industrial îi revine rolul de a conduce și supraveghea întreaga activitate desfășurată de robot, asigurînd totodată și dialogul dintre utilizator și robot prin intermediul limbajului de programare. Conducerea unui robot implică un număr mare de sarcini și funcțiuni, cu un grad de complexitate din ce în ce mai ridicat pe măsura perfecționării roboților. Principalele sarcini care trebuie îndeplinite de sistemul de conducere pot fi împărțite în trei grupe mari [23] /:

- A. Modelarea mediului exterior.
- B. Specificarea, generarea și controlul mișcărilor.
- C. Asigurarea dialogului cu utilizatorul uman și integrarea într-o rețea de calculatoare.

Cele trei grupe de sarcini pot fi îndeplinite de calculatoare separate, aferente fiecărei grupe de sarcini, între care

trebuie asigurată o intercomunicație. Mai mult chiar, și cadrul unei anumite grupe sarcinile pot fi rezolvate prin structuri multiprocesor. De exemplu, în cadrul grupei B, unui procesor coordonator îi revine rolul specificării și coordonării mișcărilor - apelînd la generatoare de mod și de traiectorie - și unui alt (micro)procesor, numit procesor de ax, îi revine rolul comenzii și reglării mișcării propriu-zise.

Cele trei grupe de sarcini au fost analizate și prezentate în / 7 /. În teza de doctorat de față preocupările, cercetările, rezultatele obținute sînt orientate numai pe grupa B de sarcini ale unui sistem de conducere, în încercarea de a aduce contribuții cu privire la optimizarea sistemelor de conducere nemijlocită a axelor unui robot, realizarea de sisteme de acționare performante, în variante exclusiv numerice.

## 2.2. Aspectele problemei conducerii roboților industriali. Domeniul de preocupare și obiectivul tezei de doctorat

În vederea reprezentării matematice și apoi prin programul calculatorului a pozițiilor și orientărilor elementelor dispozitivului de ghidare se procedează la definirea ca punct caracteristic a unui punct fizic al obiectului manipulat de robot.

Parcursarea etapelor unui proces tehnologic robotizat înseamnă, din punctul de vedere al conducerii dispozitivului de ghidare, deplasarea acestuia în raport cu obiectele din mediu astfel încît punctul caracteristic să ocupe în fiecare etapă poziții determinate sau impuse.

Se atașează un sistem de coordonate triortogonal obiectului manipulat cu originea în punctul caracteristic. Acest sistem de coordonate are, în fiecare etapă, o poziție și o orientare în raport cu un sistem de coordonate de referință. La trecerea punctului caracteristic printr-un anumit punct din spațiul cartezian și sistemele de coordonate atașate elementelor dispozitivului de ghidare dobîndesc poziții și orientări distincte în raport cu sistemul de referință /23, 74, 14, 26, 39, 59/.

Pentru explicitate, trecerea punctului caracteristic printr-un punct din spațiul cartezian se exprimă, în robotică, cu matricea de situare  $T/23$  /. Aceasta reprezintă poziția și orientarea, impuse de procesul tehnologic, sistemului de coordonate tri-

ortogonal, atașat punctului caracteristic, (în raport cu sistemul de coordonate de referință) în momentul trecerii prin punctul respectiv din spațiu.

În conducerea unui robot, pentru realizarea unui proces, anumite puncte din spațiu sînt numite puncte țintă (goal points) și reprezintă puncte urmărite ca scop final în deplasare, de cele mai multe ori fiind impusă și oprirea în acestea. La trecerea de la un punct țintă la altul se pot impune și puncte intermediare sau puncte via cu, sau, de regulă, fără oprire prin acestea. Dintre punctele via unele sînt puncte via propriu-zise (through via points), prin care trecerea este obligatorie, iar altele sînt puncte pseudo-via de trecere numai prin vecinătatea lor.

Trecerea punctului caracteristic printr-un anumit punct țintă sau via subînțelege, evident, poziționarea particulară a elementelor lanțului ce constituie dispozitivul de ghidare. Atingerii unui punct țintă sau via din spațiul cartezian îi corespunde, deci, un set de valori particulare pentru coordonatele poziționale relative ale elementelor cuplelor cinematice conducătoare. Setul de coordonate de la nivelul cuplelor cinematice conducătoare se obține cu analiza cinematică inversă, din matricea de situație ce exprimă trecerea punctului caracteristic prin punctul țintă sau via cartezian. Valorile obținute reprezintă tot puncte țintă sau via dar exprimate în coordonatele cuplelor cinematice conducătoare.

Cu alte cuvinte, trecerea punctului caracteristic printr-un punct din spațiu se poate exprima fie în coordonate carteziene, cu matricea de situație aferentă; fie în coordonatele poziționale relative ale elementelor cuplelor cinematice conducătoare, prin setul de valori corespunzătoare.

Conducerea dispozitivului de ghidare, constă, în esență, din două aspecte, numai aparent distincte. Pe de o parte trebuie asigurată deplasarea dispozitivului de ghidare astfel încît punctul caracteristic să treacă prin punctele din spațiu impuse de procesul tehnologic sau de alte criterii. În al doilea rînd se impune ca deplasarea elementelor sistemului mecanic să se desfășoare lin, fără smucituri și porniri sau opriri bruște. Acest din urmă deziderat este impus, pe de o parte, de anumite procese tehnologice (paletizare, montaj ș.a.), pe de altă parte se impune eliminarea șocurilor mecanice în structura mecanică a robotului. O deplasare lină este asigurată dacă se realizează după o funcție

lină de timp, adică după o funcție continuă de timp care are și, cel puțin, primele două derivate continue în timp.

Prin urmare conducerea unui robot constă în a asigura poziția și orientarea impuse sistemului de coordonate cu originea în punctul caracteristic de obligativitate de a trece prin punctele țintă sau punctele via și sintetizarea deplasărilor de la un punct la altul cu funcții line de timp.

Pentru realizarea conducerii sistemului mecanic este, principial, necesară parcurgerea a două etape:

1. Specificarea și generarea elementelor mișcării.
2. Conducerea nemijlocită a sistemului mecanic.

Conducerea nemijlocită este realizată de sistemul de acționare al robotului în cadrul căruia, la nivelul fiecărei cuple cinematice conducătoare este prevăzut câte un motor (electric, hidraulic sau pneumatic) care produce mișcarea mecanică, și care este integrat într-un sistem automat, local, de reglare destinat conducerii nemijlocite a cuplei cinematice respective.

În literatură se folosesc frecvent termenii sistem de reglare automată pentru conducerea nemijlocită a unei axe /36/, sistem de conducere nemijlocită a unei axe /16 / sau sistem de acționare a unei axe /45 / cu semnificație identică.

Sintetic o schemă bloc de reprezentare pentru conducerea unui robot este cea din fig.2.2.

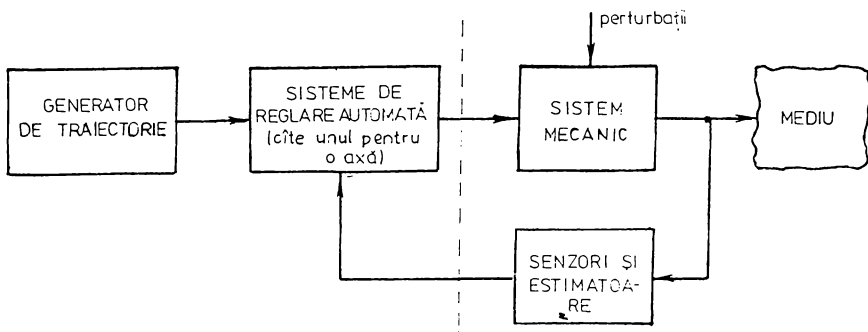


Fig.2.2. Schema bloc pentru conducerea sistemului mecanic al unui robot.

Ansamblul sistemelor de reglare automată constituie sistemul de acționare al robotului. Prezenta teză de doctorat își propune să abordeze detalii ale acestor sisteme de reglare pentru

cazul acționării electrice a roboților industriali.

Pentru reglatoarele din structura sistemelor de reglare automată mărimile prescrise (poziții, viteze, accelerații) sînt furnizate de către calculatorul care realizează specificarea și generarea elementelor mișcării (simbolizat în fig.2.2 cu generatorul de traiectorie).

În consecință etapa de specificare și generare a elementelor mișcării are ca sarcină principală generarea, în timp, a unei secvențe de puncte prescrise (set points) pentru sistemele de reglare.

Curba din spațiul cartezian descrisă de punctul caracteristic în deplasare se numește traiectoria sa. În vederea sintetizării ei este necesar să se apeleze la un formalism matematic adecvat pentru a defini și descrie mișcările dorite între punctele țintă și via și pe baza acestuia să se genereze punctele prescrise.

Formalismul matematic adoptat în robotică se bazează pe procedeul de interpolare. Acesta constă în adoptarea unor funcții, adecvate, de timp cu care se calculează secvențele de puncte prescrise care "interpolează" ("aproximează") traiectoria punctului caracteristic. Cu alte cuvinte, în conducerea unui robot, traiectoria este sintetizată prin interpolare, între puncte țintă și via, care sînt numite, fie, ca în matematică, noduri sau puncte de interpolare / 89/, fie puncte de definire a traiectoriei (name path points) / 23/.

Funcțiile de interpolare se aleg de variabilă de timp pentru că interesează evoluția în timp a sistemului condus. Se aleg funcții line de timp din necesitatea de a asigura deplasări line, după cum s-a arătat anterior. Alegerea se face dintr-o clasă de funcții de interpolare specifice și cunoscute din matematică / 31/, /91 /, de regulă polinomiale. Funcțiile de interpolare de variabilă timp au fost denumite, în robotică, funcții conducere / 52/. Valorile numerice concrete, în diferite momente, ale funcției conducere și derivatelor sale (poziții, viteze, accelerații generalizate) sînt valorile concrete ale parametrilor cinematici ai mișcărilor elementelor mecanice în acele momente.

Legătura dintre funcția conducere (de timp) și punctele (spațiale) țintă sau via este realizată prin aceea că, la anumite momente impuse, valorile numerice rezultate din funcția conducere trebuie să coincidă cu valorile corespunzătoare trecerii prin punctele respective.

generator de traiectorie (trajectory planner) și care calculează, fie în coordonate carteziene, fie în coordonatele poziționale relative ale elementelor cuplelor cinematice conducătoare, valori numerice discrete pentru o anumită funcție conducere, la intervale de timp constante (cuprinse de regulă între 1/200 Hz și 1/20 Hz).

Aceste valori numerice corespund unor poziții succesive ocupate de elementele dispozitivului de ghidare. În acest mod, generatorul de traiectorie calculează, pe baza funcțiilor conducere, locații succesive în care trebuie să se situeze în timp dispozitivul de ghidare, realizând de fapt corespondența spațiu-timp în mișcare. Se asigură concomitent trecerea prin punctele țintă și via și o deplasare fără șocuri.

Schematic sarcinile blocului generator de traiectorie pot fi reprezentate ca în fig.2.3.

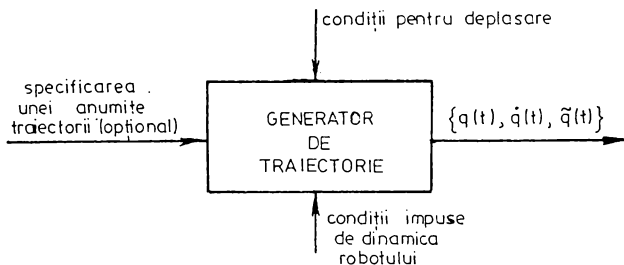


Fig.2.3. Schema bloc pentru sarcinile generatorului de traiectorie.

Generatorul de traiectorie primește ca date de intrare condițiile impuse pentru deplasare și cele impuse de dinamica sistemului mecanic și, în unele cazuri, chiar expresia analitică a unei traiectorii carteziene pe care punctul caracteristic trebuie să o descrie în spațiu. La ieșire se obțin, cu o anumită frecvență de repetiție, poziții, orientări, viteze și accelerații, în coordonate carteziene sau în coordonatele poziționale relative ale elementelor cuplelor cinematice conducătoare, care sînt prescrise succesiv pentru deplasarea elementelor mecanice dintr-o poziție inițială într-o poziție finală.

Pentru generatorul de traiectorie sînt cunoscute /74/ două modalități de operare :

A. Fără sintetizarea unei traiectorii carteziene prescrise.

B. Cu sintetizarea unei traiectorii impuse pentru punctul caracteristic.



În primul mod, prin deplasarea elementelor mecanice conform secvenței de parametri calculați de generatorul de traiectorie. punctul caracteristic descrie în spațiu o traiectorie greu previzibilă. Din acest punct de vedere denumirea de generator de traiectorie, în acest mod de operare, este aparent improprie.

În cazul A generatorul de traiectorie primește condițiile impuse trecerii prin puncte țintă și via, alege o funcție conducere dintr-o clasă de astfel de funcții, calculează coeficienții variabilei timp, din condiții impuse, și generează, la intervale constante de timp, coordonatele carteziene sau ale pozițiilor relative ale elementelor cuplelor cinematice conducătoare, vitezele și accelerațiile necesare deplasării robotului conform condițiilor impuse și după funcția conducere aleasă.

Generatorul de traiectorie este utilizat în modul A de operare în cazul metodelor de conducere secvențială, punct cu punct sau multipunct /52/.

În cazul celui de al doilea mod de operare generatorul de traiectorie primește la intrare și expresia analitică a unei traiectorii carteziene impuse în deplasarea punctului caracteristic. Sînt și situații în care traiectoria impusă se obține prin aproximarea ei și pe porțiuni cu segmente de curbe simple (liniare, circulare, parabolice, etc.).

Modul de operare B este cel utilizat în cazul conducerii pe traiectorie continuă a robotului cînd generatorul de traiectorie este utilizat în sensul propriu denumirii sale.

Și în acest mod de operare sînt alese funcții conducere, fie în coordonate carteziene, fie în coordonatele poziționale relative ale elementelor cuplelor cinematice conducătoare, din care se calculează și apoi se generează, la intervale constante de timp, parametrii cinematici impuși mișcării elementelor mecanice. Pozițiile obținute din funcția conducere trebuie să coincidă sau să aproximeze cît mai precis punctele carteziene impuse.

Indiferent de modul de operare al generatorului de traiectorie, A sau B, funcțiile conducere după care acesta execută generarea se aleg astfel încît sa fie cît mai simple, să asigure deplasarea lină și să necesite un timp de calcul cît mai scurt (cît mai apropiat de timpul real). Mărimile de la ieșirea generatorului de traiectorie reprezintă puncte prescrise pentru sistemele de reglare automată, din structura sistemului de acționare, care asigură conducerea ne-

mijlocită a robotului.

Obiectivul tezei de doctorat este de a studia posibilități de optimizare pentru sistemele de conducere nemijlocită a cuplelor cinematice conducătoare ale roboților industriali cu acționare electrică. Domeniul de preocupare este la nivelul sistemului de acționare, a sistemelor de reglare automată ce îl constituie, și urmărește realizarea de sisteme de acționare în variante exclusiv numerice.

### 2.3. Conducerea nemijlocită a axelor roboților cu acționare electrică. Stadiul actual de dezvoltare a sistemelor de acționare electrică.

După cum s-a arătat în capitolul I, acționarea electrică a roboților industriali cunoaște o extindere continuă /54/. Motivele care determină acest fenomen, înșirate în capitolul introductiv, sînt în general cunoscute, unanim acceptate și se încadrează în situația generală determinată de dezvoltarea automatizării industriale, care a impus ca astăzi acționările electrice să consume 50 - 60 % /45/ din totalul energiei electrice disponibile.

Structura unui sistem de reglare automată care conduce nemijlocit o cuplă cinematică conducătoare din sistemul mecanic, pentru roboții industriali cu acționare electrică se încadrează, desigur, în structura, general acceptată /45/ pentru sistemele de acționare electrică și poate fi reprezentată ca în schema bloc din fig.2.4 /7/.

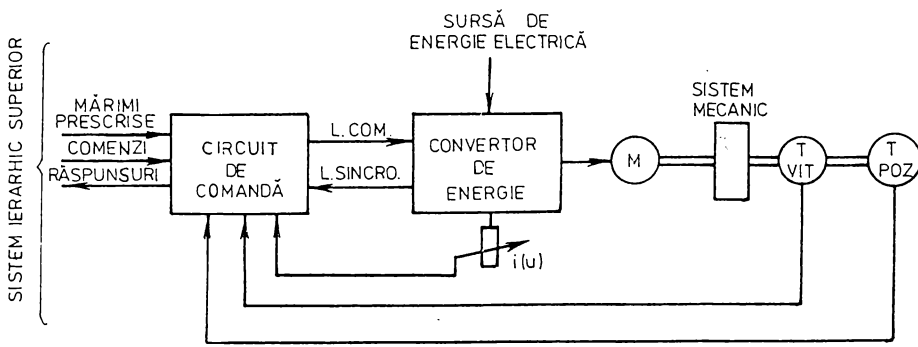


Fig.2.4. Schema structurală a unui sistem de acționare electrică (pentru o axă).

508.331  
19111

Sistemul de acționare, în ansamblu, al întregului robot, reprezintă totalitatea sistemelor de reglare automată pentru conducerea nemijlocită a fiecărei cuple cinematis conducătoare, cu câte-o structură ca cea din fig.2.4.

Schema din fig.2.4 reprezintă o structură cu caracter general, în care motorul electric M poate fi un motor de curent continuu, un motor de curent alternativ sau motor pas cu pas.

În cazul roboților industriali /54/ la sistemul de acționare se consideră inclus și dispozitivul mecanic de adaptare a mișcării de rotație a arborelui motorului electric la cupla cinematică conducătoare, corespunzătoare, din dispozitivul de ghidare. În conducerea sistemului de acționare, în calculele necesare reglării, trebuie însă luate în considerare toate elementele din sistemul mecanic puse în mișcare de motorul electric /25/.

În ceea ce privește tipul de motor electric utilizat în aplicații mai frecvent este motorul de curent continuu. Acesta aduce cu sine, ca principal argument favorabil în utilizare, facilitatea conferită de a-și modifica turația la simpla modificare a tensiunii la borne, în condițiile menținerii unui cuplu motor constant. Rezultă de aici o structură mai simplă și mai ieftină a echipamentului de comandă și o pronunțată flexibilitate a sistemului de acționare, ceea ce prezintă importanță în cazul roboților industriali.

Motorul asincron, în schimb, prezintă o robustețe mai ridicată, un preț de fabricație mai scăzut, conferă posibilitatea alimentării de la rețeaua de curent alternativ, poate fi lipsit de perii colectoare (pentru varianta cu rotor în scurtcircuit) și deci poate fi utilizat în medii explozive. Principalul dezavantaj al motoarelor asincrone este imposibilitatea reglării turației lor prin modificarea amplitudinii tensiunii la borne, fiind necesară modificarea simultană a amplitudinii și frecvenței tensiunii de alimentare. Deși astăzi au apărut, în afara metodei de reglare  $U/f=ct$  /45/, și alte metode cum sînt: cea bazată pe principiul orientării după cîmp /49/ sau metoda accelerării cîmpului slaborată de Yamamura /97/, /98/, problema reglării motoarelor asincrone rămîne deschisă, iar echipamentele necesare sînt complexe și costisitoare.

În ceea ce privește utilizarea motoarelor pas cu pas, aria lor de utilizare se restrînge astăzi, deși la un moment dat MPP apărea ca și un traductor biunivoc impuls electric de comandă - deplasare.

În majoritatea covârșitoare a realizărilor industriale concrete roboții de astăzi sînt acționaiți cu motoare de curent continuu / 7 /.

Blocul convertor de energie, din fig.2.4, este în fapt un bloc de conversie a formei de energie electrică de la cea disponibilă de la sursa de energie, la cea necesară alimentării corespunzătoare a motorului utilizat. Sursa de energie este de regulă rețeaua de curent alternativ, monofazată sau trifazată, sau, mai rar, surse autonome de c.c. sau c.a. Dacă motorul utilizat este de curent continuu la bornele sale trebuie aplicată o tensiune continuă de valoare variabilă. Pentru un motor asincron este necesară la borne o tensiune alternativă cu amplitudinea și frecvența variabile. În cazul motoarelor pas cu pas alimentarea se face cu impulsuri de tensiune cu factor de repetiție variabil.

Circuitul de comandă al acționării asigură prin linii de comenzi semnalele de închidere și deschidere a diferitelor căi de curent din blocul convertor, în conformitate cu cerințele de deplasare impuse robotului și în condiții de sincronizare de evenimente (L.SINCRO) între cele două blocuri. Totodată circuitului de comandă îi revine și sarcina determinării vitezelor de deplasare și pozițiilor realizate, prin preluarea informațiilor furnizate de traductoarele de viteză și poziție, de a le compara cu mărimile prescrise, furnizate de nivelul ierarhic superior - generatorul de traiectorie, în cazul roboților industriali - și de a genera comenzi în vederea atingerii parametrilor prescrși. De aceea circuitul de comandă al acționării trebuie să permită un dialog cu sistemul ierarhic superior (comenzi-răspunsuri) și să realizeze și supravegherea procesului (de exemplu calea de supraveghere i) pentru a asigura reglajele și protecțiile necesare.

Detaliind primele două blocuri din fig.2.4 se ajunge la fig.2.5, reprezentată pentru cazul alimentării de la rețea. Blocul convertor din fig.2.4 este reprezentat de Electronica de putere. Acest bloc este constituit, în cazul utilizării unui motor de curent continuu, dintr-un redresor comandat sau un chopper, dintr-un grup redresor urmat de un invertor (contactor static de frecvență) dacă motorul este asincron, respectiv din comutatoare electronice de putere dacă motorul este pas cu pas. Exemple de scheme de acest tip și explicații cu privire la funcționarea, cerințele impuse în comandă, moduri concrete de implementare au fost prezentate pe larg în / 5 /.

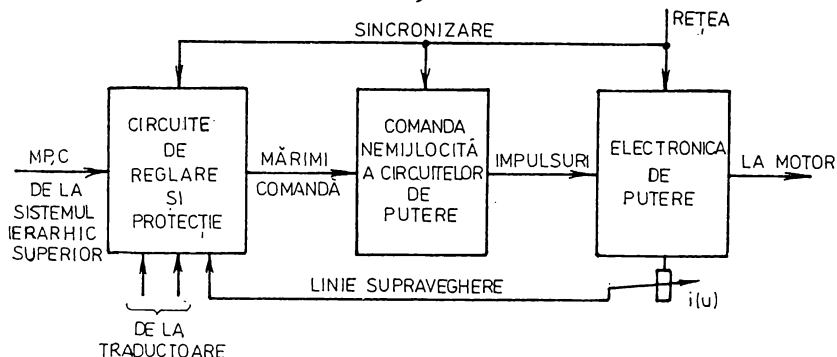


Fig.2.5. Schema bloc a circuitului de comandă al unui sistem de acționare.

Circuitul de comandă reprezentat în fig.2.4 cuprinde, după cum rezultă din detalierea din fig.2.5, două blocuri distincte. Unul dintre blocuri realizează comanda nemijlocită a electronicii de putere prin impulsuri de comandă, iar cel de al doilea este constituit de reglatoarele (de poziție, viteză curent sau tensiune) necesare, circuitele de protecție, circuitele de interfață (dialog) cu traductoarele și sistemul ierarhic superior.

În varianta tradițională, reglatoarele utilizate sînt analogice, iar circuitele de interfață sînt realizate cu circuite integrate analogice și digitale clasice. Problema alegerii tipului de regulator adecvat, a proiectării, implementării și acordării acestuia este o problemă de importanță majoră, cu efect direct asupra preciziei robotului.

În cazul acționărilor cu motor de curent continuu, cel mai frecvent utilizat și la roboții industriali, ansamblul din fig. 2.5 este numit variator și cunoaște o largă răspîndire în industrie. Este realizat ca un echipament bine pus la punct, cu bună fiabilitate.

Apariția și răspîndirea microprocesoarelor a condus la o serie de mutații în modalitățile de implementare a sistemelor automate. În acest sens trebuie menționată realizarea numerică a diferitelor sisteme de reglare automată a proceselor industriale.

Și în sistemele de acționare electrică microprocesoarele au pătruns din ce în ce mai mult. În comanda roboților industriali realizarea în variante numerice a sistemelor de acționare este deosebit de potrivită prin prisma faptului că dialogul dintre sistemul de comandă al robotului și sistemul de acționare se realizează direct, fără interfațări suplimentare. Se asigură, totodată, precizii sporite, fiabilitate mai mare și, nu în ultimul rînd, posibilitatea de a realiza mai simplu sisteme flexibile sau adaptive.

Pentru variantele numerice de comandă ale sistemelor de acționare, s-a generalizat în ultimul timp structura reprezentată în fig.2.6.

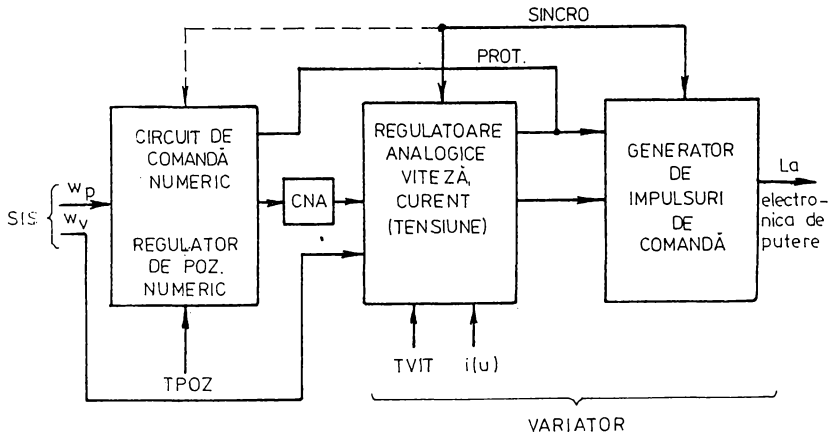


Fig.2.6. Sistem de acționare electrică cu comandă numerică.

Un sistem de comandă numeric servește la implementarea regulatorului numeric de poziție, la dialogul cu sistemul ierarhic superior și la generarea semnalelor de comandă. Și în acest caz sînt utilizate variatoare ce conțin regulatoare analogice, fiind deci necesare convertoare numeric-analogice (CNA) ca circuite de interfață între comandă și reglare.

În cazul roboților industriali, rolul circuitului de comandă din fig.2.6 poate fi preluat, în principiu, de către sistemul de comandă propriu-zis al robotului / 7 /. În cazul roboților moderni calculatorul care asigură specificarea și generarea elementelor mișcării este urmat de un microcalculator dedicat conducerii sistemului de acționare / 58 /, / 7 /.

În / 7 / au fost prezentate cîteva structuri de sisteme de conducere pentru roboți industriali. În absolut toate cazurile structurile de acționare sînt realizate în conformitate cu cele descrise mai sus, cu variator electronic în variantă analogică. La expoziția mondială de roboți industriali de la Brno - Cehoslovacia, 5-9 martie 1990, toate echipamentele de comandă expuse au prezentat astfel de variante pentru sistemul de acționare. Robotul ROMAT-76 expus la firma germană CLOOS la facultatea de Meca-

nică a Universității Tehnice Timișoara, este de asemenea construit cu variator analogic de turație.

2.4. Sisteme de acționare "exclusiv numerice" /67 /  
pentru conducerea roboților industriali, domeniu  
actual de cercetare

După cum s-a arătat în capitolul I din teza de doctorat de față, în ultimul timp cercetările cu privire la optimizarea sistemelor de acționare electrică, în special pentru roboții industriali, sînt orientate în direcția adaptării elementelor componente ale sistemelor de reglare la tehnica numerică și de a realiza sisteme de acționare în variante "exclusiv numerice".

În teza de doctorat de față, în / 6 /, în / 7 /, în / 5 /, se propune structura din fig.2.7 ca sistem de conducere pentru o axă a unui robot industrial, componentă a unui sistem de acționare cu motoare de curent continuu pentru roboți industriali.

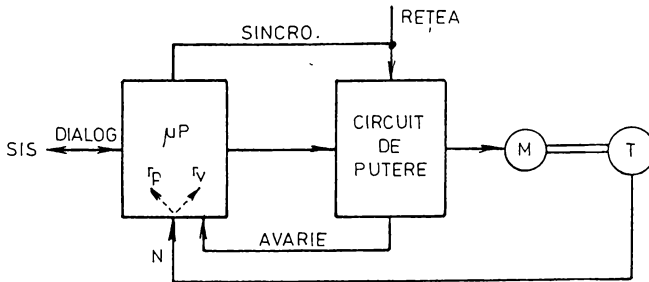


Fig.2.7. Structura unui sistem de acționare, în variantă "exclusiv numerică", pentru o axă a unui robot industrial acționat cu MCC.

În schema din fig.2.7 un singur sistem cu microprocesor de uz general, cu dezvoltare obișnuită, din familia 8030 sau Z80, îndeplinește toate funcțiile a două sau trei blocuri din variantele de sisteme de acționare tradiționale. În afara microsistemului, a circuitului de putere, a motorului și a unui singur traductor - incremental de deplasare - nu mai este necesar nici un alt bloc electronic la realizarea unui sistem de acționare pentru o axă a unui robot.

Simplificarea propusă se justifică din mai multe puncte de vedere.

În primul rînd, circuitul electronic de putere trebuie comandat în impulsuri. Acest lucru se poate realiza în mod direct

de către microsistem. În / 5 / sînt prezentate, detaliat, diferite modalități de comandă directă pentru mai multe variante de circuite de putere, specifice acționărilor cu motor de curent continuu, motor asincron și motor pas cu pas, și se arată că această funcție este comod realizabilă cu microprocesor întrucît pe liniile unui port de ieșire se poate sintetiza orice secvență necesară de impulsuri de comandă. Sînt citate lucrări / 80/, / 82/, / 1 /, /64 /, /65 /, / 93/ în care se folosește procedeul amintit.

În teza de doctorat sînt prezentate numai cazurile circuitelor aferente acționărilor cu motor de curent continuu.

În afara comenzii directe a circuitelor de putere, cu microsistemul ce comandă o structură ca cea reprezentată în fig.2.7 pot fi implementate prin mijloace software și alte funcții. De exemplu se pot sintetiza, prin program, funcții de reglare pentru poziție și/sau viteză.

Traductorul de poziție este de tip incremental. Ca urmare, informația furnizată de acesta poate fi preluată direct pe un port de intrare al microsistemului. Informația de viteză se poate obține prin calcul din informația de poziție și astfel se poate evita utilizarea traductorului de viteză dedicat.

Pe parcursul său, teza de doctorat de față, demonstrează posibilitățile de a efectua, la nivelul unui microsistem uzual, toate calculurile implicate de realizarea funcțiilor amintite mai sus și chiar a unor funcții suplimentare. În acest sens, se demonstrează că se poate asigura prin mijloace software, simultan cu funcțiile de bază, și controlul curentului din înfășurarea de comandă a motorului, fără a fi necesar un traductor de curent.

Circuitele auxiliare necesare completării sistemului se rezumă doar la cele care asigură sincronizarea cu rețeaua și la un circuit pentru sesizarea, printr-un semnal pe o linie, a avariei de curent survenită accidental.

Circuitele analogice, convertoarele numeric-analogice și tahogeneratorul pot fi, practic, evitate. Se obține, astfel, o schemă ieftină și cu fiabilitate ridicată.

De menționat este și avantajul reprezentării numerice a constantelor regulatorului, ceea ce permite schimbarea lor cu ușurință, conferind o mare flexibilitate sistemului de acționare.

Utilizarea microprocesoarelor de uz general în conducerea sistemelor de acționare necesită, desigur, o investiție de inteligență suplimentară (software), dar permite realizarea unor sis-



tame de acționare performante și în condițiile lipsei de procesoare și coprocesoare specializate sau dedicate. În literatură se cunosc eforturi depuse în cercetare în țările mai puțin dezvoltate economic /19 /, /93 / în această direcție. Se remarcă /93 / în care un singur microprocesor microsistem într-o singură capsulă (vezi cap.3) asigură comanda și reglajul unui sistem de acționare cu motor asincron / 5 /. Și în teza de doctorat de față sînt prezentate realizări ale autorului în ceea ce privește utilizarea, în condițiile unei exploatare cît mai eficiente, a microsistemelor uzuale în conducerea sistemelor de acționare cu motor de curent continuu.

### Capitolul 3.

#### SISTEME CU MICROPROCESOARE UTILIZATE ÎN COMANDA ACŢIONARILOR ELECTRICE

##### 3.1. Arhitectura standard a unui sistem cu microprocesor utilizat în comanda circuitelor de acţionare electrică

Structura unui microsistem utilizat în comanda acţiunilor electrice se încadrează, desigur, în arhitectura standard acceptată pentru acest tip de sisteme. Schema bloc a unui sistem cu microprocesor este reprezentată în fig.3.1 /100/.

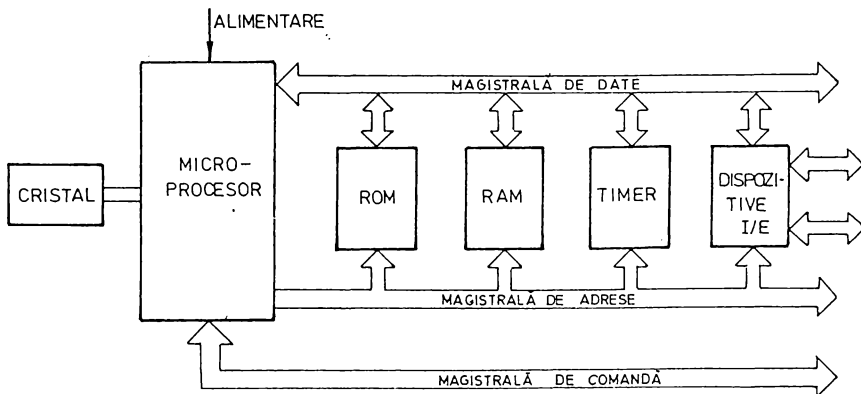


Fig.3.1.Arhitectura standard a unui microsistem.

Dispozitivul fundamental din sistem este microprocesorul care conține, în esență, elementele unei unități centrale dintr-un (micro) calculator. Operând cu instrucții și date, rolul microprocesorului este de a decodifica informația recepționată, de a vehicula corect adresele și datele, de a efectua operații în unitatea aritmetică și logică internă și de a asigura schimbul de semnale de comandă interne și externe sistemului, în ritmul impulsurilor de tact.

Intercomunicația între microprocesor și celelalte compo-

nenta de sistem este asigurată prin cele trei magistrale:

- a) magistrala de date
- b) magistrala de adrese
- c) magistrala de comandă.

Pentru microprocesoarele care operează prin circuite de 1 octet magistrala de date este constituită din 8 linii. Magistrala de adrese are, de regulă, 16 linii permițând adresarea directă a unei memorii de maxim 64 kocteți. La unele microprocesoare magistrala de date nu este realizată fizic separat întrucât cele 8 linii corespunzătoare biților cel mai puțin semnificativi de adresă, se utilizează, prin multiplexare, și ca magistrală de date. Magistrala de comandă conține un număr variabil de linii ce depinde de tipul concret de microsistem și servește la transmiterea semnalelor de comandă de tip SCRIE, CITEȘTE, START, STOP, furnizate de microprocesor către celelalte componente de sistem, respectiv la transmiterea semnalelor de răspuns de la componentele de sistem la microprocesor /99, 72, 62, 94/.

Memoria ROM, nevolatilă, componentă a sistemului constituie spațiul de depozitare al unui set minim de subrutine necesare pentru lucrul cu sistemul, set cunoscut în literatură / 6 / sub denumirea de nucleu. Având în vedere faptul că structura de microsistem descrisă este utilizată la comanda nemijlocită a circuitelor de acționare electrică, nucleul nu trebuie să fie deosebit de extins. În acest sens, subrutinele absolut necesare pentru ca sistemul să fie operativ se pot rezuma la: subrutine de inițializare a sistemului, subrutine utilitare de lucru în timp real, subrutine de prelucrare a semnalelor implicate în proces, subrutine de tratare a întreruperilor și subrutine de comunicație (tip intrări - ieșiri). Utilizarea unui nucleu redus presupune că programele de lucru au fost, în general, puse la punct pe alte sisteme de calcul mai dezvoltate și că aplicația concretă necesită doar o adaptare minimă. Flexibilitatea sistemului nu este afectată întrucât orice program poate fi completat cu subrutine înscrise în memoria RAM prin care se asigură particularizările, eventual, necesare,

Având în vedere considerentele amintite, capacitatea memoriei ROM din microsistem poate fi și de numai 1-2 kocteți.

Memoria RAM permite ca informația să fie atât citită cât și înscrisă în oricare locație a sa. În aplicații, în acest spațiu de memorie se depun programele curente și tabelele de date necesare în lucru. Lungimea programelor este, desigur, variabilă, dar

experiența a dovedit că în aplicații complexe, din domeniul acționărilor electrice, o memorie de 16 Kocteți RAM dinamic este oportunoasă ca extindere.

Necesitatea reîmprospătării memoriei RAM dinamic, la intervale constante de timp, are drept consecință directă lungirea duratei de acces la informația păstrată de aceasta. Pentru realizarea de viteze cât mai mari în efectuarea calculului se impune includerea în sistem și a unui spațiu de memorie RAM static. În această memorie se păstrează subrutinele de calcul cu numere reprezentate pe mai mulți octeți. Un minim de 256 octeți memorie RAM static s-a dovedit suficient pentru subrutinele prin care se efectuează operațiile matematice care apar în programe uzuale.

Un bloc component important în sistem îl constituie circuitele de interfață intrări/ieșiri (I/O) pentru comunicație cu exteriorul microsistemului. Aceste circuite se pot realiza fizic ca circuite integrate dedicate de tip 8251, 8255, 8212, 8155, 8355 ș.a. / 73,24/ și constituie baza fizică pentru porturile de intrare și ieșire ale sistemului. Numărul porturilor necesare este variabil, de la o aplicație la alta, dar se poate aprecia că acest număr este limitat superior la 10.

Prin porturile de intrare-ieșire se realizează, pe de o parte, comunicația cu utilizatorul (în general cu un nivel de conducere ierarhic superior), pe de altă parte, se asigură semnalul de semnale direct implicate în comanda acționării - semnale de proces.

Semnalele direct legate de proces se grupează în numărul de intrare, ce sosesc la porturi de intrare ale microsistemului, și semnale de comandă către circuitele de putere din structura acționării, generate prin porturi de ieșire ale sistemului.

În comanda circuitelor de acționare electrică, la porturile de intrare ale sistemului utilizat sosesc semnalele sub formă unor impulsuri sau niveluri de tensiune TTL. Aceste semnale se pot împărți în două categorii: mărimi de prescriere și date de proces.

În prima categorie sînt cuprinse, în funcție de aplicație, de exemplu: poziția și/sau viteza impusă pentru deplasarea la un moment dat, semnal de avans, frecvența de comandă prescriată, constante de proces etc.

Datele de proces mai des întîlnite, în acționările electrice, sînt: poziția și/sau viteza curentă, impulsuri de sincronizare cu frecvența rețelei, semnale de stare din proces ș.a.

Mărimile urmărite în proces sînt, în general, analogice. Transformarea lor în semnale utile calculatorului de comandă implică, de regulă, utilizarea de convertoare analog-numeric (CAN).

Numărul de CAN utilizate depinde de tipul traductoarelor care se folosesc, de modul în care a fost concepută structura acționării și de numărul de unități de proces urmărite. La proiectarea sistemului de acționare se recomandă alegerea unei configurații cu un număr minim de convertoare, atît pentru reducerea prețului de cost cît și pentru creșterea fiabilității sistemului. Reducerea numărului de convertoare este posibil mai ales în cazul acelor mărimi care pot fi determinate de microcalculator indirect, prin calcul, cu ajutorul mărimilor măsurate și a constantelor, cunoscute, de proces. Utilizarea circuitelor de multiplexare este o metodă hardware de reducere a numărului de CAN. Prin multiplexare un sistem CAN poate fi folosit la interfașarea mai multor mărimi analogice din proces.

În capitolele 5, 6 și 7 din teza de doctorat se prezintă realizarea unor sisteme de reglare numerică a vitezei și poziției, într-o acționare cu motor de curent continuu în care utilizarea convertoarelor a fost complet evitată.

Semnalele de comandă furnizate prin porturile de ieșire ale sistemului sînt, de asemenea, semnale TTL din care, cu circuite adecvate, se formează impulsurile de comandă propriu-zise pentru tranzistoarele și tiristoarele circuitului de putere. Între portul de ieșire al sistemului și circuitul comandat se impune o izolare galvanică prin optocuploare sau transformatoare de impulsuri. Structura circuitului comandat de microsistem și programul de comandă trebuie astfel întocmite încît semnalele generate la porturile de ieșire să fie active pe nivelul 0 logic.

În caz contrar, orice întrerupere fizică a unui fir de legătură de la sistem la circuitul de forță poate provoca intrarea în conducție, în momente nedorite, a dispozitivului corespunzător de putere.

În figurile 3.2 și 3.3 se prezintă două exemple de conectare, la o linie a unui port de ieșire, a unui tranzistor, respectiv a unui tiristor din circuitul de putere /11, 80, 79/.

Tot la circuitele I/E se conectează și echipamentele periferice care asigură comunicația microsistemului cu utilizatorul, de exemplu lector de bandă sau casetofon (pentru încărcarea programelor de lucru), tastatura și circuitele de afișare. Dacă sis-

temul de comandă al unei acționări electrice este parte integrantă a unei mașini cu comandă numerică sau a unui robot, tastatura și circuitele de afișare de pe panoul operatorului se folosesc și la intervenția în circuitul de comandă al acționării.

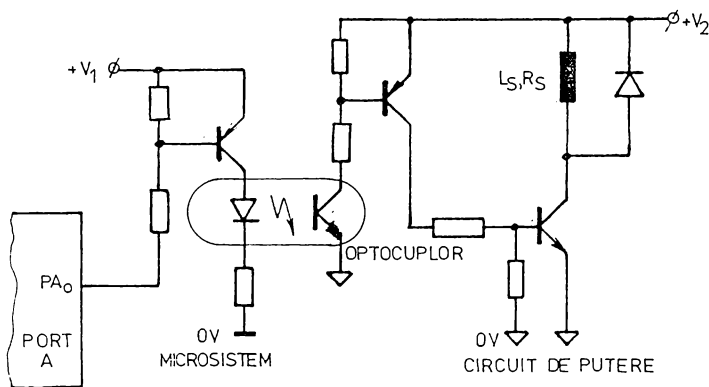


Fig.3.2. Model de circuit de comandă pentru tranzistoare de putere activat de o linie de ieșire din microsystem.

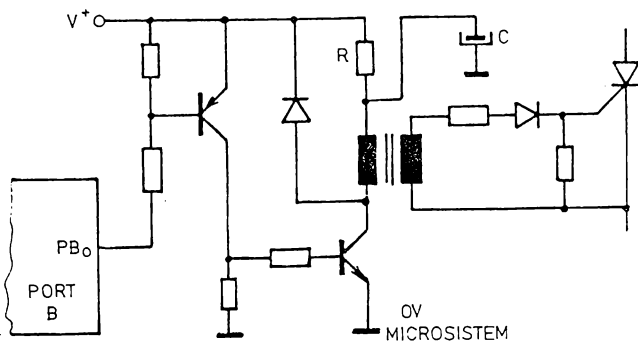


Fig.3.3. Model de circuit de comandă pentru tiristoarele din circuitul de putere activat de o linie a unui port de ieșire.

Deosebit de util pentru punerea în funcțiune și depanarea sistemului este să se includă în structura sa și o interfață de I/E de comunicație cu un DAF / 71/.

Rolul variabilei "timp" este esențial în conducerea proceselor. Acest rol rezultă din desfășurarea în timp real a acestora, ceea ce presupune existența, în cadrul (micro)sistemului de conducere, a următoarelor facilități /42 /:

- a) de a măsura timpul scurs între două evenimente,
- b) de a genera semnale care să marcheze intervale de timp de durată prestabilită, variabilă sau fixă, în decursul operării sistemului.

Aceste două funcții sînt îndeplinite de circuite specializate din sistem, generatoare de timp real sau timere (de exemplu de tip 8253) care sînt, în esență numărătoare programabile. Funcționarea generatoarelor de timp real este sincronă cu semnalul de tact, după care lucrează întregul sistem și a cărui constanță este asigurată de cristalul de cuarț. Prin introducerea timerelor în sistem, microprocesorul poate fi degrevat și de sarcina contorizărilor de lungă durată, necesare deseori în aplicații.

Dintre metodele de utilizare ale generatoarelor de timp, în circuitele de comandă din acționările electrice, cele mai utile par să fie următoarele:

- a) utilizarea unui contor de timp real absolut, pornit, de exemplu, la începutul controlării procesului și citirea conținutului acestui contor la apariția evenimentelor urmărite din proces,
- b) utilizarea timerelor pentru declanșarea unor semnale de întrerupere la intervale de timp cu care au fost programate, și contorizarea evenimentelor apărute în aceste intervale de timp.

Un microsistem utilizat în comanda circuitelor de acționare electrică trebuie prevăzut cu 1-3 generatoare de timp real. Exemplul caracteristic de utilizare a timerelor în ganul de aplicație amintit este determinarea vitezei de avans a organului mobil al acționării, prin contorizarea deplasării efectuate de acesta într-un interval de timp stabilit.

În activitatea sa de comandă și control a acționării microsistemul conlucrează cu o serie de dispozitive externe unității centrale și memoriei, iar dialogul cu acestea este intermitent. Dispozitivele externe trebuie să primească, la momentul potrivit, atenția unității centrale prin intermediul unui sistem de întreruperi corespunzător.

Sistemul de întreruperi este constituit din structura hardware special construită și setul de subrutine din nucleul păstrat în memoria ROM. Principalele sarcini ale sistemului de întreruperi sînt:

- acceptarea semnalelor de întrerupere,
- determinarea cererii cu prioritate maximă,
- compararea nivelului de prioritate al întreruperii cu

cal al programului curent executat de către unitatea centrală,  
- întreruperea unității centrale când nivelul de prioritate al întreruperii este adecvat,

- informarea unității centrale asupra adresei de start care servește întreruperea (de obicei o instrucție CALL).

În aplicațiile specifice acționărilor electrice semnalele care solicită întrerupere se pot împărți în trei categorii: semnale sosite de la proces, semnale de la unitatea de comandă de nivel superior (sau operatorul uman) și semnale de sincronizare generate de timerele sistemului.

În funcție de aplicația concretă, utilizatorul stabilește o ordine a priorităților semnalelor de întrerupere și asigură conectarea hardware corespunzătoare a dispozitivelor care le generează. În acest mod, microprocesorul își desfășoară activitatea curentă pînă la apariția unei cereri de întrerupere, când prelucrările curente se întrerup, pentru perioada tratării solicitării sosite, prin subrutine adecvate aflate la adrese prestabilite. Un asemenea mod de rezolvare a cererilor de întrerupere este cel mai eficient în aplicațiile concrete.

Avînd în vedere numărul mediu relativ restrîns de semnale de întrerupere ce pot apare în comande circuitelor de acționare electrică este suficient un număr maxim de 8 nivele de prioritate a întreruperilor, care pot fi gestionate cu o singură capsulă de tip 8259 / 42 /.

### 3.2. Exemple de sisteme de comandă cu microprocesoare uzuale.

În cercetările noastre cu privire la utilizarea microprocesoarelor în comanda acționărilor electrice s-au folosit microprocesoarele: 8035, din familia 8048, 8085, din familia 8080 și Z80. Fiecare dintre cele trei microprocesoare folosite este reprezentant al cîte uneia din trei generații. În cele ce urmează se prezintă, pe scurt, structurile sistemelor construite și utilizate, în vederea unei discuții în paralel asupra rezultatelor practice obținute și a diferențelor specifice apărute în diverse aplicații, precum și pentru a se stabili avantajele și dezavantajele rezultate din utilizarea de microprocesoare din generații diferite.



### 3.2.1. Sisteme de comandă cu microprocesor 8035

Microprocesorul 8035 face parte din familia 8048, familie cunoscută în literatură sub denumirea de "low-end microprocessor" tradus în l.română "microsistem într-o singură capsulă" /99,100,24/. Schema bloc internă a microprocesorului 8048 este redată în fig. 3.4 /72/. Din această structură se poate observa că microprocesorul 8048 conține, intern, toate blocurile ce intră în componența unui sistem de calcul.

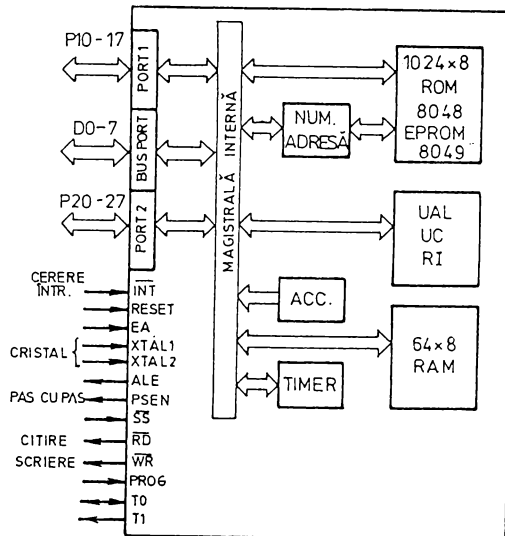


Fig.3.4. Structura microprocesorului 8048.

Microprocesorul 8035 prezintă o serie de caracteristici comune familiei 8048: poate funcționa ca dispozitiv de sine stătător sau ca parte a unei configurații multiprocesor; are o singură sursă de alimentare de +5V; conține unitatea centrală a unui microcalculator cu toate elementele componente, o memorie RAM cu 64 de octeți care poate fi folosită și îndeplinind rolul de registre generale, porturi de intrare-ieșire și un generator de timp real programabil. Deosebirea esențială între microprocesorul 8035 și 8048 este că 8035 nu are memorie ROM internă, ceea ce implică necesitatea de a conecta în sistem o capsulă de memorie ROM și un amplificator de magistrală. Schema de principiu a configurației realizate este prezentată în fig.3.5 /8, 13/.

Microprocesorul 8035 este prevăzut cu trei porturi bidirecționale de câte 8 biți. În cazul sistemului de față "BUS-PORT-ul"

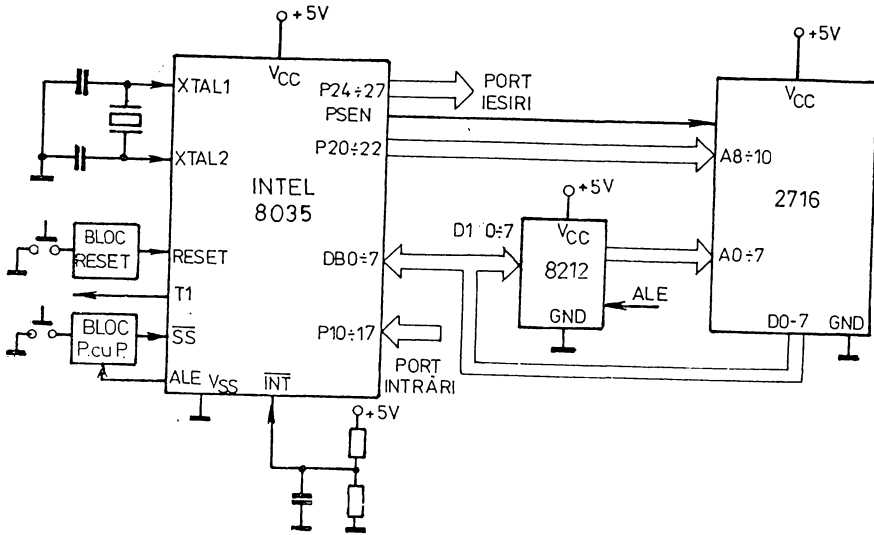


Fig.3.5.Sistem cu microprocesorul 8035.

microprocesorului este utilizat, prin multiplexare, ca magistrelă de date și adrese. Astfel pe liniile acestui port se generează cel mai puțin semnificativi 8 biți de adresă dar pe aceleași linii se sesc și instrucțiunile programului din memoria EPROM externă. Liniile P20-23 din portul P2 sînt de asemenea utilizate pentru magistrala de adrese a sistemului. Pentru dialogul cu exteriorul au rămas la dispoziție patru linii din portul P2 (P24-P27) și portul P1. Aceste linii sînt programabile ca linii de intrare sau ieșire.

Pe liniile de intrare sînt preluate mărimile de prescriere și datele de proces de tipul celor enumerate în paragraful 3.1. Aceste semnale pot fi primite de la un sistem de calcul ierarhic superior, în cazul mașinilor unelte cu comandă numerică sau a roboților industriali, sau de la operatorul uman, prin circuite de interfață corespunzătoare.

Pe liniile de ieșire se obțin impulsurile de comandă pentru tranzistoarele sau tiristoarele circuitelor de putere componente ale sistemului de acționare.

Microsistemul este prevăzut cu circuite logice externe pentru realizarea comenzii RESET, care apelează adresa de început a subrutinei de inițializare și permite lansarea execuției programelor curente. S-a prevăzut, de asemenea, un bloc logic pentru

posibilitatea rulării programelor în modul "pas cu pas", deosebit de util la punerea în funcțiune sau depanarea sistemului.

În afara structurii de microsistem cu procesorul 8035, prezentată anterior, se pot realiza și configurații de sisteme mai simple. În situația în care componentele necesare sînt disponibile se pot construi sisteme cu un singur circuit integrat, eventual, cu două. Se obțin sisteme capabile să îndeplinească toate funcțiile cerute unui circuit de comandă pentru anumite circuite de acționare în condiții de fiabilitate și preț de cost extrem de avantajoase.

În fig.3.6 se prezintă o configurație de microsistem complet, obținut prin conectarea la microprocesorul 8035 a unei capsule 8355. Semnalele neimplicate direct în dialogul dintre cele două circuite integrate nu s-au reprezentat pe figură.

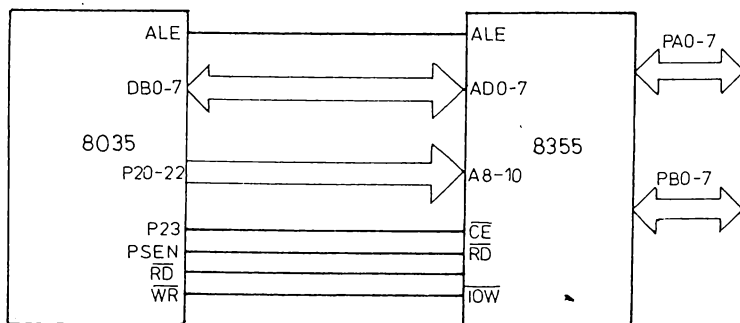


Fig.3.6.0 variantă de sistem cu microprocesorul 8035.

Circuitul 8355 conține 2 Kocteți memorie PROM și două porturi intrare-ieșire de câte 8 biți și prezintă avantajul de a putea fi direct conectat la microprocesorul 8035 / 72/.

Microsistemul complet obținut cu numai două circuite integrate are avantajul de a putea lucra în două moduri distincte. Primul mod este identic cu cel descris anterior, programul fiind înscris în circuitul 8355. Informațiile exterioare și (sau) de la sistemul ierarhic superior pot sosi fie prin porturile de intrare ale microprocesorului, fie prin porturile circuitului 8355. Cel de al doilea mod de lucru permite executarea unor secvențe întregi de program sosite din exterior prin porturile lui 8355.

Configurația de microcalculatoar într-un singur circuit integrat în sensul cel mai restrîns al termenului se obține utili-

zînd un alt microprocesor din seria 8048; microprocesorul 8749. Acest circuit are cei 2 Kocteți de memorie EPROM incluși în capsulă (fig.3.7) / 8 /.

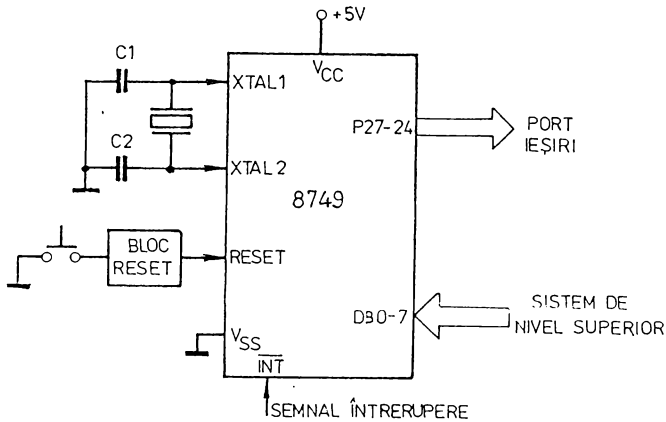


Fig.3.7. Sistem într-o singură capsulă cu microprocesorul 8749.

Din cele prezentate rezultă posibilitatea de a realiza sisteme de calcul extrem de simple. După cum se va vedea în cele ce urmează, capacitatea de memorie și viteza de calcul a acestui tip de microsistem sînt suficiente pentru îndeplinirea funcțiilor impuse unui circuit de comandă în suficient de multe circuite de acționare electrică. În acest fel, realizarea circuitelor de comandă pentru acționări devine extrem de simplă, ieftină, flexibilă, fiabilă și, practic, tipizată.

### 3.2.2. Sistem de dezvoltare cu microprocesorul 8085

Microprocesorul 8085 se consideră succesorul direct al microprocesorului 8080. Din punct de vedere hardware, circuitul 8085 înglobează într-un singur circuit integrat funcțiile microprocesorului 8080, ale generatorului de tact 8224 și ale circuitului de comandă 8228, acestea din urmă absolut necesare în cadrul unui sistem cu 8080. În plus microprocesorul 8085 este alimentat de o singură tensiune (+5V), lucrează la o frecvență de 3 MHz (mai mare decît cea de la 8080), are magistrală comună de date și adrese, are cinci intrări pentru tratarea întreruperilor, permite accesul datelor serie direct la microprocesor prin conexiunile SID - SCD

și are prevăzute trei semnale specifice de comandă pentru controlul vehiculării informației pe magistrale /24, 72, 62/.

Din punct de vedere software microprocesorul 8085 este perfect compatibil cu 8080, avînd doar două instrucții suplimentare (RIM, SIM) /62, 42/.

Se consideră însă că, saltul calitativ esențial făcut de microprocesorul 8085, îl constituie realizarea, simultană cu aceasta, a două circuite speciale, circuitele 8155 și 8355, cu ajutorul cărora se poate realiza un sistem cu performanțe foarte bune prin simpla interconectare a numai trei circuite integrate.

Circuitul 8155 conține o memorie RAM cu o capacitate de 256 octeți și un circuit de interfață tip intrări-ieșiri (I/O) cu trei porturi. Două dintre porturi (A și B) sînt de opt biți și pot fi programate fie ca porturi de intrare fie ca porturi de ieșire, iar cel de-al treilea (C) este de 6 biți. În același circuit se află și un timer programabil /24, 72/.

Circuitul 8355 conține o memorie PROM de 2 Kocteți și un circuit interfață cu două porturi de cîte opt biți /24, 72/.

Liniiile porturilor acestor circuite servesc pentru schimbul de semnale cu circuitul de acționare comandat de sistem, semnale de tipul celor menționate în paragraful 3.1.

În sistemul realizat, la structura de bază s-a atașat inițial o tastatură hexazecimală și un afișaj cu 7 segmente, comandate prin intermediul unui circuit 8279.

În acest mod s-a obținut un sistem cu o structură minimală, dar care a corespuns pentru realizarea unor aplicații de complexitate medie /106/.

Ulterior microsistemul a fost extins. S-a adăugat o memorie suplimentară de 32 Kocteți RAM dinamic și 4 Kocteți EPROM. Au fost incluse în sistem și cîte un circuit 8155 și 8355, suplimentare, pentru realizarea de mai multe porturi I/O și dotarea cu a unui al doilea timer. S-a construit, apoi, un circuit interfață pentru display și consolă de tip DAB local paralel. De asemenea, cu ajutorul unui circuit 8255 (trei porturi programabile intrare-ieșire) s-a realizat o interfață de comunicare cu lector și perforator de bandă.

Configurația sistemului realizat și utilizat în aplicații este dată în fig.3.8./82, 13, 11/.

Din punct de vedere software sistemul a fost dotat cu un sistem de operare minimal interactiv păstrat în memoria ROM. Nucleul sistemului de operare conține subrutina de inițializare,

subrutinale de tratare I/E la nivel fizic, subrutina de gestionare a întreruperilor și un număr de subrutine utilitare (temporizări, intercomunicații etc.). Celelalte blocuri software din cadrul sistemului de operare sînt: un interpretor de comenzi, un sistem de gestiune al fișierelor, un editor, un asamblor în doi pași și un depanator.

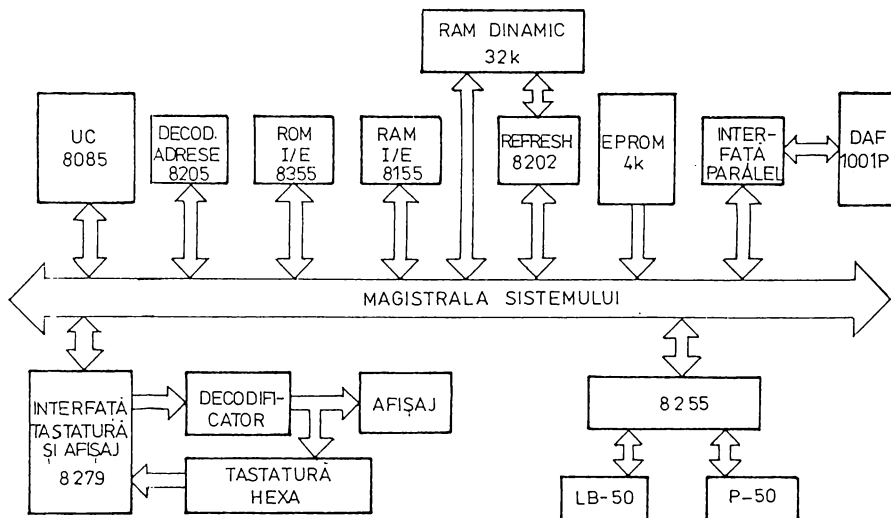


Fig.3.8.Sistem cu microprocesorul 8085.

Printr-o astfel de dotare operatorul poate lucra de la consola DAF-ului și are la dispoziție o serie de facilități pentru alcătuirea programelor. Dintre aceste facilități se pot menționa: posibilitatea de lucru pe maxim 6 fișiere, utilizarea unui set de comenzi ale editorului (LIST, DELETE) sau ale depanatorului (DUMP, ENTR, BREAK), posibilitatea de a scrie programul în limbaj de asamblare inclusiv utilizarea etichetelor sau a unor pseudoinstrucțiuni ale asamblorului (ORG, EQU, DB) ș.a. Programele pot fi citite sau salvate pe bandă perforată.

Pe parcursul lucrării, în capitolele 4,5,6,7, se prezintă programe întocmite pe sistem și se evidențiază detalii cu privire la facilitățile oferite de acesta.

Sistemul prezentat, succint, în acest paragraf este capabil să realizeze conducerea unor procese complexe, cu un număr suficient de mare de unități de proces controlabile și implicînd calcule laborioase. Configurația construită și utilizată reprezintă

mai mult decât o structură strict concepută pentru comanda nesmițlocită a circuitelor de acționare electrică, de tipul prezentat, principial, în paragraful 3.1. Microcalculatorul descris poate fi, însă, folosit la întocmirea, punerea la punct și depanarea unor programe pentru alte sisteme, mai restrânse, componente propriu-zise ale circuitelor de acționare electrică.

### 3.2.3. Sistem de comandă cu microprocesorul Z80

Structura sistemului utilizat este dată în fig.3.9 / 13/ și reprezintă o structură standard de microcalculator de uz general.

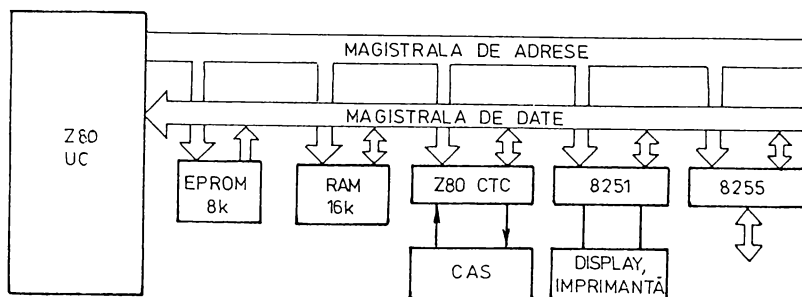


Fig.3.9. Sistem cu microprocesor Z80.

Microprocesorul Z80 este considerat primul reprezentant al familiilor de microprocesoare moderne. Deși descendent din familia 8080, microprocesorul Z80 are resurse mult sporite mai ales din punct de vedere software. Dintre aceste posibilități sînt de amintit: posibilitatea adresării indexate cu ajutorul a două registre index, posibilitatea adresării relative și pe bit, existența setului alternativ de registre precum și a instrucțiunilor de transfer de date în bloc /72,61/.

Memoria sistemului este formată din 8 Kocteți EPROM și 16 Kocteți RAM dinamic. Comunicația cu utilizatorul este asigurată prin DAF, imprimantă și casetofon legate la sistem printr-o interfață realizată cu circuitele Z80 CTC și 8251. Programele se încarcă de pe casetofon în memoria RAM. Schimbul de semnale, similar cu cele de la paragraful 3.1, cu circuitele de acționare propriu-zise se face prin circuitul 8255.

## Capitolul 4.

### UTILIZAREA MICROPROCESOARELOR IN COMANDA ACTIONARILOR CU MOTOARE DE CURENT CONTINUU.....

#### 4.1. Probleme specifice cu privire la acționările cu mașini de curent continuu

Structura unui sistem de acționare se prezintă schematic ca în fig.2.7 din capitolul 2.

Sarcina mecanică dezvoltată în arborele motorului un cuplu static rezistent  $M_S$ . Pentru învingerea acestuia motorul de acționare trebuie să producă cuplu motor  $M$ . În regim staționar  $M=M_S$ . La o modificare a unuia din cele două cupluri rezultă schimbarea regimului de funcționare, se modifică viteza de lucru și deci energia cinetică a maselor în mișcare.

Variația energiei cinetice a maselor, raportată la arborele motorului în unitatea de timp reprezintă puterea inerțială  $\beta_6 /$ :

$$P_J = \frac{dW_c}{dt} \quad (4.1)$$

Dar

$$W_c = \frac{1}{2} J \Omega^2 \quad (4.2)$$

unde  $J$  este momentul de inerție al maselor în mișcare ce se rotesc cu viteză unghiulară  $\Omega$ .

Deci

$$P_J = J\Omega \frac{d\Omega}{dt} + \frac{\Omega^2}{2} \frac{dJ}{d\alpha} \cdot \frac{d\alpha}{dt} = J\Omega \frac{d\Omega}{dt} + \frac{\Omega^3}{2} \frac{dJ}{d\alpha} \quad (4.3)$$

(se admite, în general, că  $J$  variază cu poziția unghiulară  $\alpha$ ).

Cuplul dinamic, ce reprezintă diferența dintre cuplul motor și cel rezistent, este:

$$M_d = M - M_S = \frac{P_J}{\Omega} = J \frac{d\Omega}{dt} + \frac{\Omega^2}{2} \frac{dJ}{d\alpha} \quad (4.4)$$



În practică, de cele mai multe ori,  $J=ct.$  și ecuația de mișcare devine:

$$M - M_S = J \frac{d\Omega}{dt} \quad (4.5)$$

cu consecințele imediate:

- a)  $M > M_S \Rightarrow \frac{d\Omega}{dt} > 0$  - accelararea acționării,
- b)  $M < M_S \Rightarrow \frac{d\Omega}{dt} < 0$  - încetinirea acționării,
- c)  $M = M_S \Rightarrow \Omega = ct.$  - regim staționar.

Funcționarea unei acționări în timpul proceselor tranzitorii depinde de modul de variație a cuplului motor și a cuplului rezistent în funcție de viteză.

Pentru motoare, definiții în acest sens sînt caracteristicile mecanice  $\Omega = f(M)$  /86/. În cazul motorului de curent continuu cu excitație separată caracteristica mecanică naturală se obține din ecuația:

$$u = u_a + (R_i + R)i_i + L \frac{di_i}{dt} \quad (4.6)$$

unde:  $u$  este tensiunea la borne,  $u_a$  tensiunea indusă,  $R_i$  și  $i_i$  rezistența și curentul prin indus,  $R$  rezistența exterioară conectată în serie cu indusul,  $L$  inductivitatea indusului /86/.

Dar :

$$u_a = k \phi \Omega \quad /78/ \quad (4.7)$$

cu  $\phi$  - fluxul de excitație,  $\Omega$  - viteza unghiulară,  $k$  - o constantă care depinde de construcția motorului (nr.de perechi de poli, nr.de spire din indus etc.).

Rezultă în regim stabilizat:

$$u = k \phi \Omega + i_i (R_i + R)$$

$$\Omega = \frac{u - i_i (R_i + R)}{k \phi} = \frac{u}{k \phi} \left( 1 - \frac{R_i + R}{u} i_i \right) \quad (4.8)$$

Considerînd că  $\phi$  se menține la valoarea de la mersul în gol  $\phi_0$  avem:

$$\begin{aligned} \Omega &= \frac{u}{k \phi_0} \left( 1 - \frac{(R_i + R)i_i}{u} \right) = \Omega_0 \left( 1 - \frac{(R_i + R)i_i}{u} \right) = \Omega_0 - \frac{(R_i + R)i_i}{k \phi_0} = \\ &= \Omega_0 - \frac{(R_i + R)i_i}{K} = \Omega_0 - \Delta \Omega \end{aligned} \quad (4.9)$$

cu  $K = k \phi_0$ .

Ecuația (4.9),  $\Omega = f(i_i)$  se reprezintă de obicei în coordonate  $\Omega = f(M)$ , deoarece cuplul:

$$M = \frac{P_g}{\Omega} = \frac{\bar{u}_g i_i}{\Omega} = \frac{k\phi \Omega i_i}{\Omega} = k\phi i_i \quad (4.10)$$

este direct dependent de curent.

Caracteristica  $\Omega = \Omega_0 - \Delta\Omega$  se numește caracteristică mecanică a mașinii / 86/ și este o dreaptă de alura din fig.4.1.

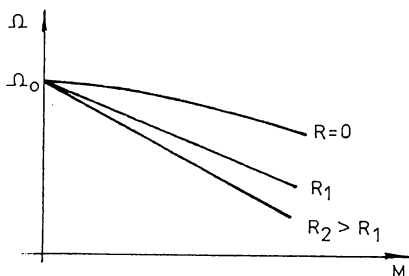


Fig.4.1. Forme caracteristicilor mecanice ale mașinii de curent continuu.

Caracteristica naturală corespunde lui  $R=0$  iar  $\Delta\Omega = R_1 i_i / K$  reprezintă căderea de viteză. Pentru  $R \neq 0$  se obțin caracteristicile artificiale.

O astfel de caracteristică mecanică are un caracter rigid, în sensul că viteza de rotație descrește relativ puțin prin mărirea cuplului.

Din ecuația (4.8) rezultă și metodele de modificare a turației motorului de curent continuu cu excitație separată:

1. Prin intercalarea de rezistențe în serie cu indusul.
2. Prin șuntarea indusului cu o rezistență.
3. Prin slăbire de câmp (a curentului de excitație).
4. Prin schimbarea tensiunii la bornele indusului, excitația rămânând constantă / 78, 30/.

Dintre aceste metode, cea mai răspândită este cea prin schimbarea tensiunii la borne, pentru că prin această metodă caracteristicile artificiale rămân paralele cu cea naturală și nu apare nici dezavantajul pierderilor de energie / 45, 78/.

Din (4.8) :

$$\Omega = \frac{u - R_i i_i}{k\phi} \quad (4.11)$$

cînd  $u$  scade, scade și termenul  $u/k\phi$ . Se obține familia de caracteristici din fig.4.2, adică o deplasare a caracteristicii mecanice momentane.

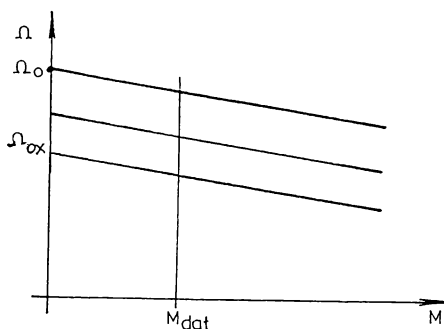


Fig.4.2. Familia de caracteristici mecanice obținută prin modificarea tensiunii la bornele motorului de c.c.

În concluzie, motorul de curent continuu prezintă două avantaje majore pentru acționările **electromecanice**: permite modificarea turației în limite largi în condițiile menținerii cuplului la o valoare ridicată și suficient de constantă, iar modificarea turației se poate face relativ simplu cu variația tensiunii de alimentare.

#### 4.2. Sisteme de acționare cu motoare de curent continuu alimentate de la redresoare comandate

Acționările reglabile de curent continuu s-au impus astăzi, practic, definitiv /45 /. Există acum experiență vastă și un larg spațiu bibliografic în acest domeniu /85, 78, 29, 46/.

În preocupările cu privire la utilizarea microprocesoarelor în comanda acționărilor electrice am folosit două tipuri de redresoare: redresorul monofazat în punte semicomandată și redresorul trifazat în punte complet comandată. Modul de abordare a problemei, rezultatele obținute, concluziile desprinse se pot extinde cu ușurință asupra tuturor tipurilor de redresoare cunoscute.

Rolul redresorului este de a realiza conversia tensiunii alternative de la rețea în tensiune continuă /45 /. În cazul redresoarelor (semi)comandate, tensiunea de la ieșire  $U_d$  este reglabilă. Reglajul se realizează prin comanda corespunzătoare a tiristoarelor din circuit. Tiristoarele trebuie comandate pe intervalul în care tensiunea lor anod-catod este pozitivă. Momentul în care tensiunea pe tiristor devine nulă și urmează să devină pozitivă în sensul anod-catod, se alege ca moment de referință pentru co-

manda pe grilă. În raport cu referința, impulsul de aprindere al fiecărui tiristor poate fi aplicat la un interval de timp variabil. Exprimiind acest interval în radiani, se definește, în literatură, unghiul de comandă ( $\alpha$ ) pentru tiristoare. Valoarea medie a tensiunii redresate se modifică, dacă și unghiul de comandă se modifică.

Momentul de referință pentru unghiul de comandă este momentul trecerii prin zero a tensiunii rețelei în cazul redresorului monofazat, respectiv momentul egalității, în domeniul valorilor pozitive, între două tensiuni de fază ale rețelei, în cazul redresorului trifazat /85/.

Structura circuitelor utilizate este prezentată în fig.4.3 și fig.4.5.

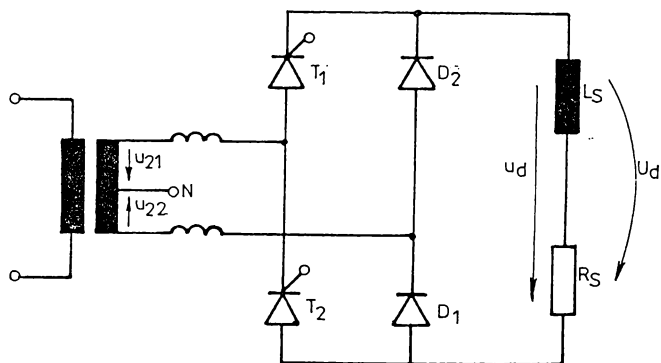


Fig.4.3. Redresor monofazat în punte semicomandată.

Funcționarea schemei din fig.4.3 poate fi urmărită și cu ajutorul formelor de undă reprezentate în fig.4.4 /78 /.

În punctul A al diagramei se presupune că începe ieșirea din conducție a tiristorului  $T_2$ . Intră în conducție dioda  $D_1$ . Cum pînă în acest moment se afla în conducție și dioda  $D_2$ , rezultă că tensiunea la bornele rezistenței de sarcină este de valoare scăzută. Diodele se mențin în conducție pînă cînd la momentul  $\omega t = \alpha$  se aplică un impuls de comandă tiristorului  $T_1$ . Pe durata intervalului de comutație  $\gamma_2$  tensiunea  $U_d$  este practic egală cu zero. După unghiul  $\gamma_2$  curentul de sarcină trece în întregime prin  $T_1$  și  $D_1$ . La trecerea prin zero a tensiunii din anodul lui  $T_1$ , acesta începe să se blocheze (punctul B). Dioda  $D_2$  intră în conducție și pînă la o nouă comandă tensiunea redresată  $U_d$  este de o valoa-

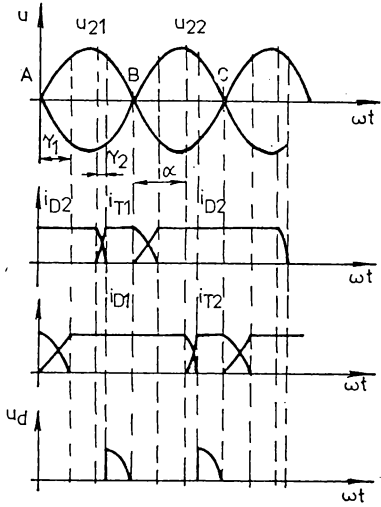


Fig.4.4. Formele de undă reprezentative pentru redresorul monofazat în punte semicomandată.

$\alpha < \omega t < \alpha + \pi/3$ , conduc tiristoarele  $T_1$  și  $T_6$ , iar terminalele sarcinii sînt conectate la fazele  $u_{s1}$  și  $u_{s2}$ . În momentul  $\omega t = \alpha + \pi/3$ , se comandă tiristorul  $T_2$ , iar tiristorul  $T_6$  se blochează fiind polarizat invers (comutația naturală). Curentul din tiristorul  $T_6$  este preluat de tiristorul  $T_2$  și, ca urmare, tensiunea la bornele sar-

re scăzută, considerată, de obicei, nulă.

Se observă că diodele conduc atît ca diode de nul cît și ca diode redresoare, iar durata de conducție a lor este mai mare decît aceea a tiristoarelor.

Funcționarea schemei din fig.4.5 /85,29 / poate fi urmărită cu ajutorul formelor de undă reprezentate în fig. 4.6.

La momentul  $\omega t = \alpha$  se comandă tiristorul  $T_1$ . Tiristorul  $T_6$  se află în acest moment în conducție, întrucît el a fost comandat anterior. Astfel în intervalul de timp

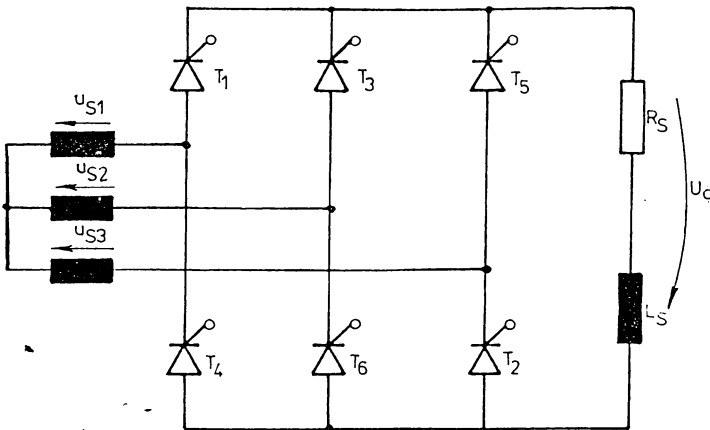


Fig.4.5. Redresor trifazat în punte complet comandată.

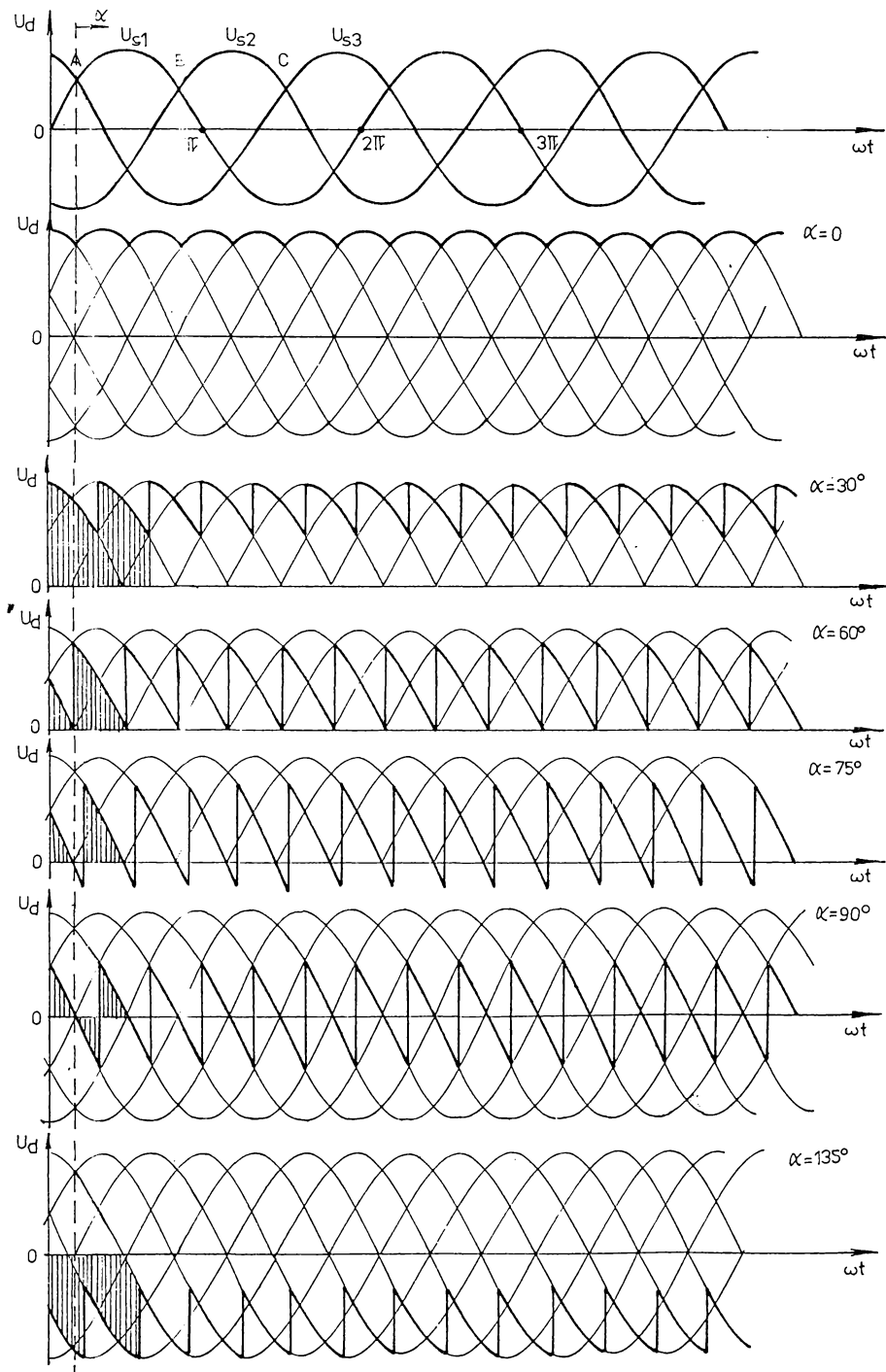


Fig.4.6. Forma tensiunii de ieșire la diferite unghiuri de comandă pentru redresorul trifazat.

cinii este:  $u_d = u_{s1} - u_{s3}$ . Acest proces se repetă după câte 60 grade sexagesimale ori de câte ori se comandă câte un tiristor. Tiristoarele din schema reprezentată în fig.4.5 au fost numerotate în ordinea în care trebuie comandate.

În fig.4.6, e reprezentată tensiunea la bornele sarcinii pentru diferite unghiuri de comandă. Pentru un unghi  $\alpha > 90^\circ$ , tensiunea  $U_d$  devine negativă. Această situație corespunde funcționării numite "în regim de invertor" /78, 37/ și este posibilă numai dacă în locul sarcinii se conectează o sursă de tensiune negativă, și deci energia este dirijată din partea de curent continuu spre partea de curent alternativ a circuitului.

Dependența valorii medii a tensiunii redresate  $U_d$ , la ieșire, de unghiul de comandă este:

$$U_d = \frac{\sqrt{2}}{\pi} U_S (\cos \alpha + 1) \quad (4.12)$$

pentru redresorul monofazat în punte semicomandată, și

$$U_d = \frac{3\sqrt{6}}{\pi} U_S \cos \alpha \quad (4.13)$$

pentru redresorul trifazat în punte complet comandată. În ambele relații  $U_S$  este valoarea efectivă a tensiunii din secundarul transformatorului /85 /.

Dacă sarcina redresorului, în principiu arbitrară, este indusul unui motor de curent continuu cu excitație separată, prin reglarea valorii medii a tensiunii redresate (a tensiunii la bornele motorului) se realizează modificarea turației motorului, așa cum s-a arătat în paragraful 4.1. Astfel, prin modificarea unghiului de comandă  $\alpha$  al tiristoarelor din redresor, se obține modificarea turației motorului de curent continuu de acționare. Această metodă asigură un reglaj continuu al turației motorului de la zero la valoarea maximă, fără pierderi de putere activă /45, 78 /.

#### 4.2.1. Implementarea pe micro sisteme a circuitelor de comandă pentru redresoare comandate

Variatoarele de turație pentru motoare de curent continuu, în care circuitul de forță este realizat cu redresor, sînt larg răspîndite în industrie. Circuitele de comandă, realizate în variante electronice clasice, sînt considerate ca fiind puse la punct /78 /.

Pentru a realiza sisteme de acționare comandate exclusiv

numeric, direct subordonabile la un sistem de calcul ierarhic superior, fără interfațări suplimentare, precum și din dorința de a simplifica cât mai mult posibil circuitele de comandă, în comparație cu configurațiile clasice, se pune problema utilizării microprocesorului în circuite de comandă pentru redresoare. Desigur utilizarea unui microsistem nu trebuie să însemne o simplă înlocuire a circuitelor clasice de comandă, cu realizarea de performanțe egale sau superioare, ci și posibilități de implementare de noi funcții pentru circuitul de comandă. Un exemplu în acest caz poate fi realizarea unei bucle de reglare de viteză cu același circuit care asigură și comanda propriu-zisă a redresorului.

În cazul utilizării sale exclusive pentru comanda redresoarelor, microcalculatorul trebuie să realizeze următoarele funcții:

(F1) să sesizeze momentul de referință pentru unghiul de comandă al tiristoarelor;

(F2) să realizeze temporizări variabile corespunzătoare pentru toată plaja de reglare impusă unghiului de comandă;

(F3) să genereze câte un impuls de comandă scurt (tipic  $t_i = 100 \mu S$ ) pentru fiecare tiristor din redresor, la momentul și în conformitate cu secvența necesară funcționării corecte a circuitului de putere.

Configurația circuitului de comandă cu microprocesor poate fi în oricare dintre variantele prezentate în cap.3.

#### 4.2.2. Exemple de aplicații pentru comanda redresoarelor cu microprocesor

În acest paragraf se prezintă modalitățile concrete de implementare pe microprocesor a circuitelor de comandă pentru redresoarele prezentate anterior.

##### 4.2.2.1. Redresor monofazat în punte semicomandată cu microprocesor / 80 /.

Structura circuitului de putere este prezentată în fig.4.3. Impulsurile de comandă pentru tiristoare se obțin pe o linie de ieșire a unuia dintre porturile microcalculatorului utilizat ca circuit de comandă. În cazul sistemului cu 8085 se poate utiliza, în acest scop, terminalul SOD al microprocesorului, dezafectî-



du-se un port de ieșire /13 /.

Momentele trecerilor prin zero ale tensiunii rețelei monofazate de alimentare sînt sesizate fie pe o linie a unui port de intrare al sistemului, fie pe intrarea SID, în cazul microprocesorului 8085.

Pentru că microprocesorul lucrează cu niveluri de tensiune TTL, tensiunea alternativă este transformată într-o tensiune dreptunghiulară prin intermediul unui circuit de forma din fig.4.7, conectat în exteriorul sistemului.

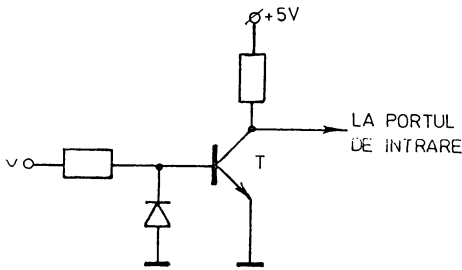


Fig.4.7. Detector de nul pentru tensiunea rețelei monofazate.

Evident o asemenea configurație introduce o eroare în ceea ce privește determinarea cu precizie a momentelor de trecere prin zero ale tensiunii rețelei, eroare cauzată de pragul,  $U_{be\gamma}$ , de deschidere al tranzistorului și, eventual, de viteza de comutare a acestuia. Principial, acest dezavantaj se poate elimina utilizînd alte variante, mai puțin simple dar mai costisitoare, pentru detectorul de nul sau adăugînd, la schema din fig.4.7, o tensiune de alimentare negativă /78, 81/.

În cazul utilizării microprocesorului nu mai este, însă, necesară o complicație hardware, întrucît erorile introduse de detector pot fi compensate. În momentul tranziției tranzistorului T, din starea blocată în starea de conducție, tensiunea rețelei a trecut prin valoarea zero înspre valori pozitive. Microprocesorul sesizează tranziția și ține seama de întîrzierea acesteia față de momentul real de trecere prin zero al tensiunii alternative. Astfel, numărul utilizat în contorizarea intervalelor de timp pentru unghiurile  $\alpha$ , este corectat la valoarea  $\alpha - \alpha_K$ , unde  $\alpha_K$  are valori de ordinul zecimilor de grad sexazecimal și este prestabilit în așa fel încît să compenseze erorile introduse de detector.

Aparent acest mod de lucru introduce erori la capetele intervalului de valori pentru  $\alpha$  ( $0^\circ$  și, respectiv,  $180^\circ$ ). Comanda redresorului monofazat monoalternanță la unghiuri mici nu se utilizează, însă, în cazul în care sarcina este un motor, întrucît comanda corectă a acestui tip de redresor se asigură dacă  $\alpha > \alpha_1$  (fig. 4.4). La unghiuri de comandă mari, amorsarea tiristcurelor nu este posibilă decît pînă la momentele în care nivelul tensiunii

alternativa depășește nivelul tensiunii electromotoare induse. Nici în cazul sarcinilor pur rezistive comanda redresorului la un unghi  $\alpha=180^\circ$  (absolut) nu se poate realiza pentru că la OV anod - catod tiristorul nu poate fi comandat /43/. Valorile extreme pentru unghiul  $\alpha$ , din punct de vedere practic, sînt cuprinse între  $25^\circ$  și  $155^\circ$  /78/, iar în acest interval, corecțiile cu  $\alpha_K$  sînt ușor de realizat.

Din momentul sesizării trecerii prin zero a tensiunii alternative sistemul așteaptă un interval de timp corespunzător unghiului de comandă  $\alpha$  ce trebuie realizat pentru o anumită tură a motorului. La sfîrșitul intervalului  $\alpha$  sistemul generează impulsul de comandă propriu-zis unuia dintre tiristoare ( $T_1$  de exemplu pe alternanță pozitivă).

Pentru a se asigura izolarea galvanică între circuitul de putere și microprocesor, impulsurile generate de microsistem se transmit către redresor prin intermediul unor optocuploare sau prin amplificatoare cu transformatoare de impulsuri care formează impulsuri de formă și amplitudine corespunzătoare unei comenzi sigure (vezi paragraful 3.1).

Valoarea unghiului  $\alpha$ , convertită într-un număr pe 16 biți poate fi tabelată în memoria sistemului sau poate fi adusă, din exterior, la un port al sistemului. În ambele cazuri valoarea impusă pentru  $\alpha$  poate fi generată de utilizator (prin consolă, panou de comandă etc.) sau de un alt calculator care dirijează procesul la nivel superior.

Organigrama bloc a programului de comandă este prezentată în fig.4.8.

Determinarea alternanței și comanda succesivă a tiristoarelor nu este absolut necesară. Cele două tiristoare pot fi comandate și simultan pe fiecare alternanță, întrucît nu intră în conducție decît acal tiristor care are tensiunea anod-catod cu polaritatea corespunzătoare /78/.

Organigrama unui program concret întocmit și utilizat pentru comanda redresorului monofazat monoalternanță este cea din fig.4.9, /80/.

Programele detaliate pentru comanda redresorului monofazat cu microprocesoarele 8035, 8085 și Z80 sînt cele de mai jos /80,13/:

8035	8085	Z80
DIS TONT I	BI: MVI A,70H	BI: XOR A,A
JMP fo	SIM	OUT 62H,A

IN A,P1	RIM	IN A,61H
MOV R6,A	ANI 80H	AND A,01
RSTR	MOV B,A	LD B,A
40: MOV A,#80	42: RIM	42: IN A,61H
OUTL P2,A	ANI 80H	AND A,01
ORL P1,#FF	CMP B	CP A,B
JN I	JZ B2	JR Z,B2
IN A,P2	LXI H,2200H	LD HL,(3000H)
MOV R5,A	MOV B,M	AST:DCI HL
41: IN A,P2	INR L	LD A,L
XRL A,R5	MOV C,M	OR A,H
JZ B1	INR L	JR NZ,AST
MOV A,R6	MOV D,M	INC A
MOV T,A	43: DCX B	OUT 62H,A
STRT T	MOV A,C	LD B,t <sub>i</sub>
46: JTF B5	ORA B	IM: DJNZ IM
JMP B6	JNZ B3	JR B1
45: STOP TONT	MVI A,0FDH	
ORL P2,#00	SIM	
MOV R7,#t <sub>i</sub>	44: DCI D	
47: DJNZ R7,B7	JNZ B4	
JMP B0	JMP B1	

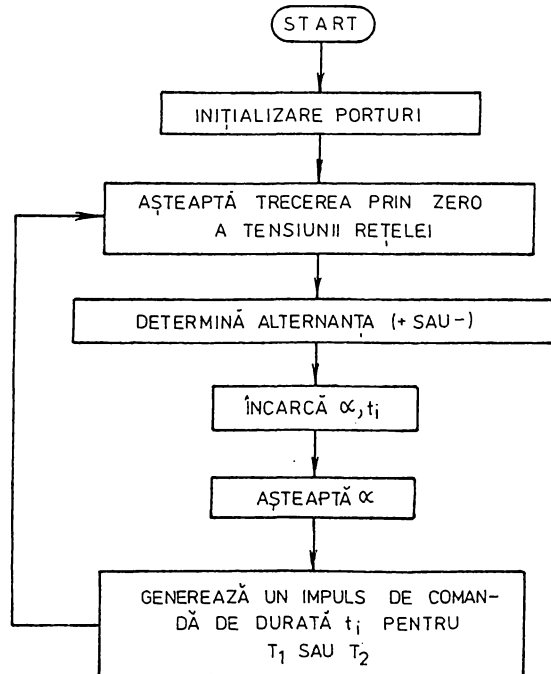


Fig.4.8. Program de comandă pentru redresor monofazat monoalternanță cu participarea curentă a unității centrale.

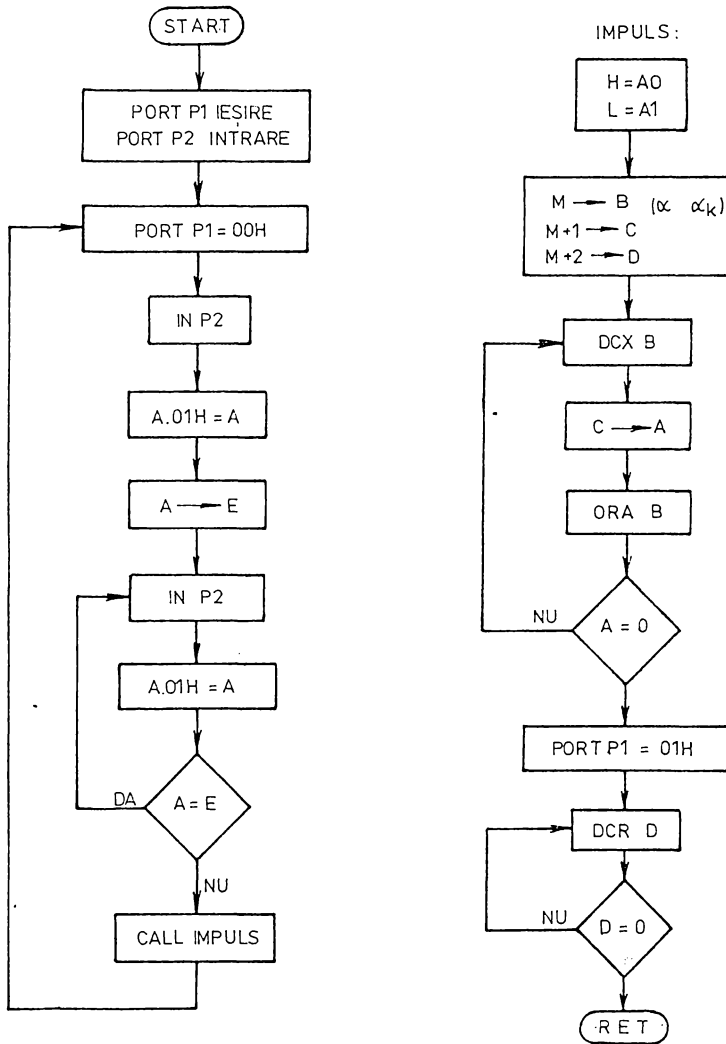


Fig.4.9. Organigrama unui program utilizat la comanda redresorului monofazat.

Exemplele date reprezintă numai porțiunile de program care realizează comanda propriu-zisă a redresorului (instrucțiunile de inițializare s-au omis).

Pentru microprocesoarele 8035 și Z80 se folosesc linii ale două porturi I/E pentru comunicația cu procesul, în timp ce la microprocesorul 8085 se utilizează terminalele acestuia SID și SOD, activate prin instrucțiunile RIM și SIM.

Pentru evitarea unei erori în sesizarea trecerilor prin zero ale tensiunii rețelei, la fiecare citire a semnalului de sincronizare se efectuează câte o deparazitare software, cu operația logică SI, de tipul A.01=A sau A.80H=A - funcție de linia conectată la detectorul de nul.

Numărul de instrucții, în limbaj de asamblare, este egal în programele scrise pentru 8035 și 8085. Pentru Z80 programul este mai scurt cu cinci instrucții.

În condițiile în care, la aceste aplicații, vitezele de lucru sînt comparabile, rezultă că microsistemul cu 8035 (în varianta 8749 mai ales - vezi paragraful 3.2.) constituie un extrem de simplu circuit de comandă pentru redresoare, în orice caz mult mai simplu decît variantele tradiționale.

Prin utilizarea, la acest tip de microprocesor, a timerului intern pentru generarea intervalelor  $\alpha$  și  $t_1$ , unitatea centrală și registrele pot executa și alte programe.

Un asemenea mod de lucru este convenabil și mai evident, în cazul celorlalte două tipuri de microprocesoare. Utilizînd generatorul de timp real (timer) și sistemul de întreruperi microprocesorul trebuie ca în timpul dintre două impulsuri de trecere prin zero a tensiunii rețelei (10 mS), să genereze numai comenzile de încărcare și pornire a timerului și poate, astfel, executa alte operații cum ar fi de exemplu: rularea unui program de autogenerare a valorilor unghiului  $\alpha$ , program de regulator de viteză (de tipul celui prezentat în capitolul 5).

Pentru cazul utilizării timerului și sistemului de întreruperi structura hardware și software, cu microprocesor 8085, este cea din fig.4.10. În acest exemplu, impulsurile de sincronizare cu trecerile prin zero ale tensiunii rețelei trebuie să fie pozitive și scurte. Ele se obțin cu un redresor de mică putere și un monostabil. Erorile introduse de detectorul de nul sînt compensate de microprocesor, la fel ca în aplicația anterioară.

Metodele prezentate permit realizarea comenzii unui redresor monofazat în condiții avantajoase. Valoarea medie a tensiunii de la ieșire se poate modifica în trepte. Mărimea unei trepte este egală cu durata decrementării cu o unitate a numărului de 16 biți ce reprezintă  $\alpha$ . Această durată este de maxim 10  $\mu$ s când contorizarea este făcută software, cum este în primul exemplu, și de circa 0,5  $\mu$ s când contorizarea este efectuată de timer, ca în cel de al doilea exemplu.

Tinând seama de faptul că  $\alpha$  este, în aplicațiile practice, de ordinul milisecundelor (1 ms corespunde la 18° sexagesimale), o treaptă de reglare a valorilor sale mai mică de 10  $\mu$ s asigură o rezoluție foarte bună (0,18° grade).

Astfel, pentru un unghi de comandă  $\alpha = 90^\circ$  (la mijlocul alternanței) valoarea tensiunii medie redresate este:

$$U_d = \frac{\sqrt{2}}{\pi} U_S (\cos 90^\circ + 1) = \frac{\sqrt{2}}{\pi} U_S \quad (4.14)$$

Dacă, prin comandă, se modifică  $\alpha$  cu o unitate (un increment = 10  $\mu$ s) tensiunea la ieșirea redresorului devine:

$$U_d = \frac{\sqrt{2}}{\pi} U_S (\cos 90,18^\circ + 1) = \frac{\sqrt{2}}{\pi} U_S \cdot 0,996858 \quad (4.15)$$

O asemenea variație pentru tensiunea la bornele motorului este ne semnificativă. Cele două caracteristici mecanice ale motorului, cea corespunzătoare valorii (4.14), a tensiunii la borne și cea corespunzătoare valorii tensiunii dată de (4.15), se confundă practic.

Pentru a se realiza schimbarea turației motorului este necesar ca numărul binar care îl reprezintă pe  $\alpha$ , în program, trebuie modificat cu 8-10 unități și, deci, rezoluția obținabilă cu microprocesor este cu circa un ordin de mărime mai bună decât este propriu-zis necesar. În comparație, circuitele tradiționale de comandă permit reglajul unghiului  $\alpha$  cu rezoluție de ordinul gradelor /21, 4/.

Cel mai mic unghi de comandă realizabil este sub 0,54 grade sexagesimale (circa 30  $\mu$ s) și este condiționat de necesitățile de determinare cu precizie a trecerilor prin zero ale tensiunii rețelei, în principal. După cum am arătat anterior însă, unghiuri atât de mici de comandă nu interesează în aplicații concrete pentru că la unghiuri de comandă mai mici de 30 grade acest tip de redresor nici nu poate fi comandat ( $\alpha > \gamma_1$ ). De altfel se cunosc aplicații con-

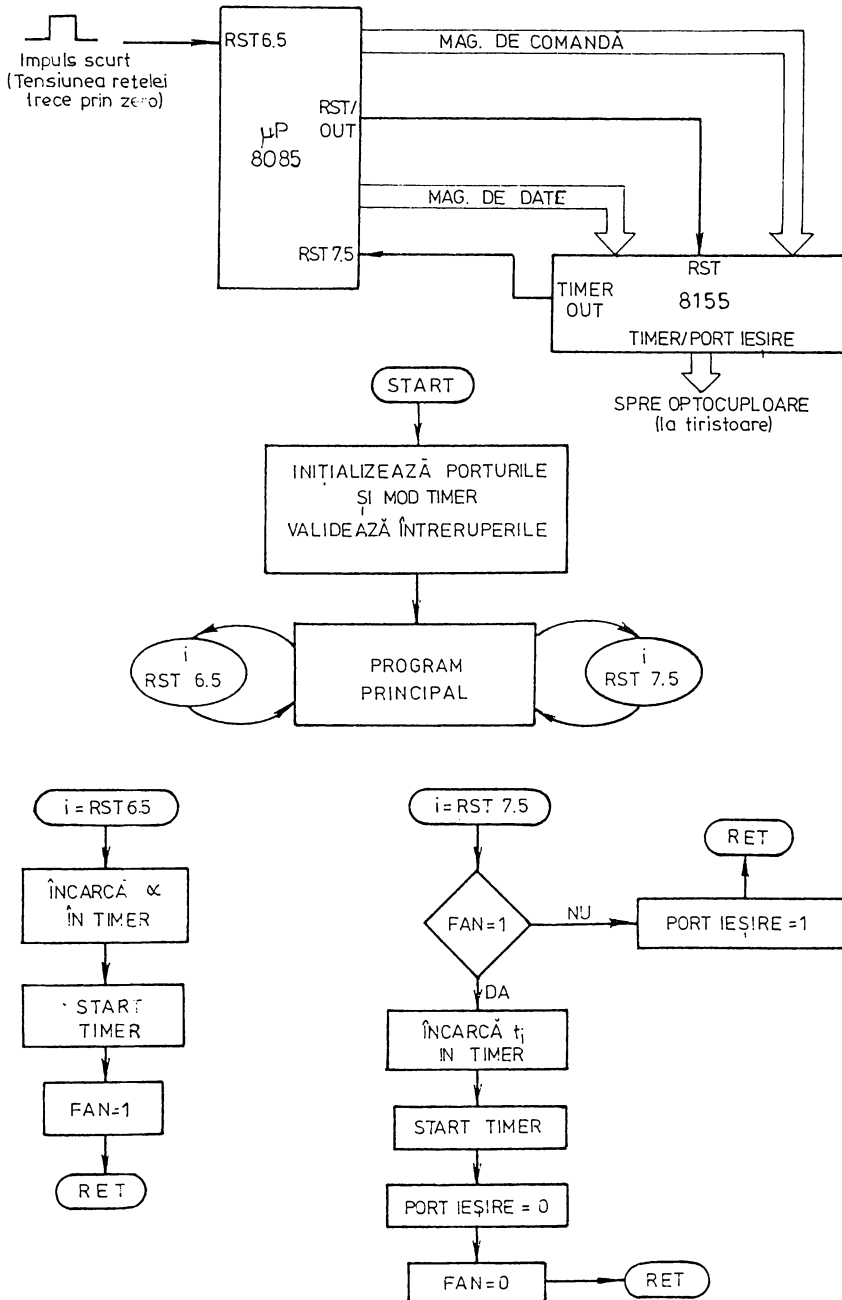


Fig.4.10. Comanda redresorului monofazat, utilizând sistemul de întreruperi pentru microprocesorul 8085.

sacrate, ce utilizează circuitul integrat  $\beta$ AA 145, dedicat comenzii pe grilă a tiristoarelor, în care nu se obțin unghiuri de comandă mai mici decât  $30^\circ$  /20,4/.

În același fel unghiuri de comandă mari, apropiate de  $180^\circ$ , deși realizabile cu microprocesor (până la  $179,46^\circ$ ), nu sînt utile practic, deoarece, chiar acceptînd că tiristoarele mai pot intra în conducție la sfîrșitul alternanței, motorul nu se mai poate roti dacă tensiunea la bornele sale este foarte aproape de zero.

Metodele descrise au fost folosite la modificarea turației unui motor de curent continuu cu excitația separată de tip CE 42 E, de fabricație IMEB. Parametrii de catalog pentru acest motor sînt  $U_a = 220V$ ,  $P_n = 2,7kW$ ,  $n_{max} = 2300$  rot/min. Încercările s-au efectuat cu motorul în gol. S-au obținut turații între 0 și cea maximă. Practic limitări în privința turației ce trebuie realizată nu există.

#### 4.2.2.2. Redresor trifazat în punte complet comandată cu microprocesor

Structura circuitului de putere este prezentă în fig.4.5. În acest caz impulsurile de comandă pentru tiristoare se obțin pe 6 linii ale unui port de ieșire din microsistem.

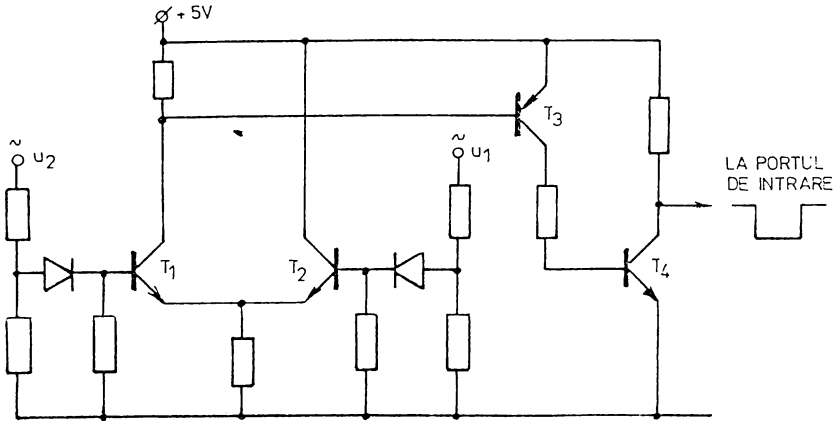


Fig.4.11. Circuit de sincronizare cu rețeaua pentru redresorul trifazat

La acest tip de redresor, momentul de referință, pentru



unghiul de comandă, este considerat momentul egalității tensiunii fazelor 1 și 2 în domeniul valorilor pozitive. Cu un circuit de forma din fig.4.11. se obține un impuls care este preluat de sistem pe un bit al unuia din porturile sale de intrare. În cazul microprocesorului 8085, se poate utiliza și facilitatea creată de existența pinului SID.

Cu toate că se redresează tensiune alternativă trifazată, este suficient un singur impuls de sincronizare /85/. Si în acest caz apare o eroare cu privire la precizia de determinare a momentului egalității celor două faze și deci microprocesorul trebuie să facă o corecție,  $\alpha_K$ , la fel ca în cazul redresorului monofazat.

Din momentul egalității tensiunilor celor două faze, microsistemul așteaptă un interval de timp corespunzător unghiului de comandă  $\alpha$ , impus pentru stabilirea unei anumite valori medii a tensiunii de ieșire. La sfârșitul intervalului  $\alpha$  sistemul generează impulsul de comandă pe grilă pentru tiristoare. Primul impuls de comandă se aplică tiristorului  $T_1$  /27/.

Desigur, și la această aplicație, sistemul trebuie izolat galvanic de circuitul de forță, prin optocuploare sau etaje cu transformatoare de impulsuri.

După comanda lui  $T_1$  trebuie generate pe rînd impulsurile de comandă pentru celelalte cinci tiristoare, la intervale de cîte 60 grade sexagesimale între ele (10/3 ms) /85/. Acest lucru se realizează relativ simplu, încărcînd în numărătorul, hardware (timer) sau software, utilizat pentru contorizarea intervalului de 60 grade, un număr  $N$  ce reprezintă echivalentul celor 3,33 ms. Valoarea lui  $N$  depinde de durata tactului în sistem.

Un ciclu de comandă se încheie după ce s-a comandat tiristorul  $T_6$ . Ciclul următor începe după încă un interval de 60 grade, cînd trebuie, din nou, comandat  $T_1$  și, apoi, pe rînd, celelalte tiristoare (tot după cîte 60 grade). Principial, o secvență corectă de impulsuri se poate obține prin utilizarea, de fiecare dată, a aceluiași numărător, cu conținut inițial  $N$  care se pornește după comanda fiecărui tiristor. Frecvența de tact a sistemului este fixă și deci comenzile pot fi, la prima vedere, corecte un timp îndelungat. Apar însă două probleme:

1. Frecvența de numărare pentru contor trebuie să fie exact multiplu întreg al frecvenței rețelei, ceea ce se realizează destul de greu.

2. Frecvența rețelei nu este constantă în timp.

Prin urmare după câteva **alternanțe** ale tensiunii rețelei apar erori în comandă.

Din acest motiv se impune sincronizarea cu rețeaua la anumite intervale de timp, de pildă la începutul fiecărui ciclu. Comanda tiristorului  $T_1$  nu se face la 60 grade interval după comanda lui  $T_6$  din ciclul precedent ci se așteaptă întâi impulsul de sincronizare cu rețeaua.

Organigrama programului, întocmit pentru un microprocesor Z80, este cea din fig.4.12.

După inițializările porturilor din sistem, programul detaliat este următorul:

LD HL,ADR	ET7: DEC B
ET1: LD D,OFFH	JR NZ,ET7
LD A,D	LD A,OFFH
OUT (PORT2),A	OUT (PORT2),A
ET2: IN A,(PORT1)	LD BC,N
AND 01	ET8: DEC BC
JR Z,ET2	LD A,B
ET3: IN A,(PORT1)	OR C
AND 01	JR NZ,ET8
JR NZ,ET3	SCF
LD B,(HL)	RL E
INC L	LD A,E
LD C,(HL)	CP A,0BFH
ET4: DEC BC	JR NZ,ET6
LD A,B	DEC D
OR C	JR NZ,ET5
JR NZ,ET4	INC L
ET5: LD E,0FEH	JR NZ,ET1
LD A,E	RST 08
ET6: OUT (PORT2),A	
LD B,7	

În literatură /27/, Dewan, personalitate cunoscută în domeniul acționărilor electrice, prezintă un exemplu de utilizare a microprocesorului 6802 la comanda redresorului în punte complet comandată.

Se utilizează un mod de lucru interesant, cu două timere. Unul dintre acestea este utilizat, la început, drept contor pentru  $\alpha$ . După primul impuls de comandă la  $T_1$ , același timer (ti-

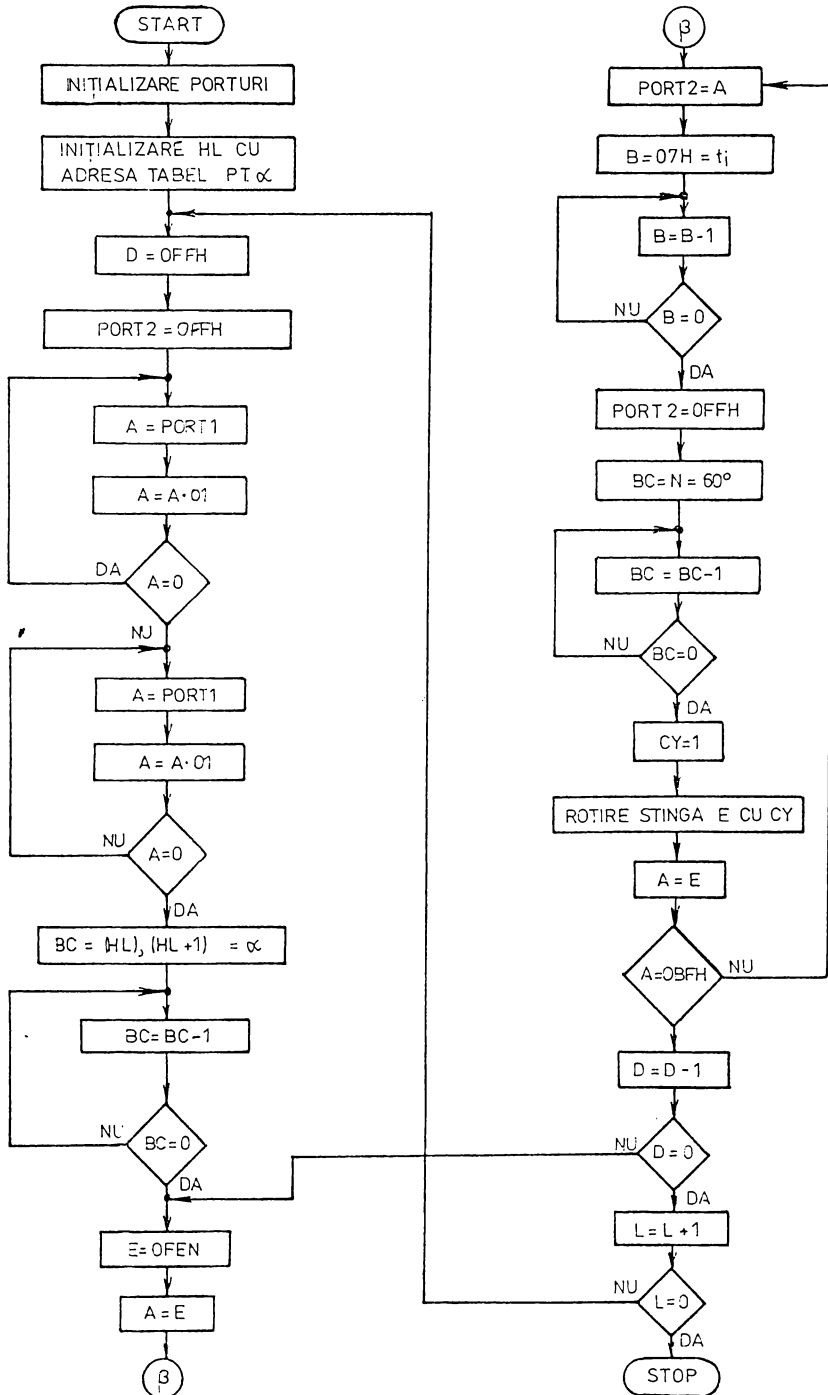


Fig.4.12. Organigrama programului de comandă a redresorului trifazat

mer 1) devine contorul pentru intervalul de 60 grade dintre impulsurile următoare, interval ce se folosește apoi tot timpul în comandă (deci și pentru comanda lui  $T_1$  după  $T_6$ ). Cel de al doilea timer (timer 2) este sincronizat cu rețeaua, la începutul fiecărui ciclu, și se utilizează drept contor de timp real.

În momentul aplicării fiecărui impuls de comandă, pentru câte unul dintre tiristoare, se citește numărul la care a ajuns timerul 2 și se compară cu o valoare obținută prin însumarea conținutului aceluiași timer (2) la comanda precedentă și numărul  $N$  utilizat la contorizarea celor 60 grade. Dacă cele două valori nu sînt egale se corectează  $N$ . Se observă că  $N$  corectat după un impuls se utilizează, de fapt, abia la comanda următoare.

În acest mod de lucru, microprocesorul stă în marea majoritate a timpului în starea HALT, întrucît timerele sînt cele care lucrează propriu-zis. Sub acest aspect, aplicația descrisă se apropie, cu excepția calculului erorilor, de ceea ce ar fi o implementare, cu circuite specifice microprocesorului, a unei variante de circuit de comandă cu numărătoare tradiționale.

În lipsa programului de corecție a intervalelor de  $60^\circ$  dintre două comenzi succesive pentru tiristoare pot apare erori în contorizarea intervalelor respective.

Astfel, dacă pe durata unui ciclu de comandă se admite o variație a frecvenței rețelei de 0,5 Hz, două tiristoare vor fi comandate succesiv la intervale diferite de  $60^\circ$ . Eroarea cea mai mare apare pe ultimul interval din cadrul unui ciclu (între comanda tiristorului  $T_5$  și comanda tiristorului  $T_6$ ), pînă la care erorile se cumulează, și este de  $3,6^\circ$ .

Frecvența rețelei, de valoare nominală  $f_0 = 50$  Hz, devine:

$$f = f_0 \pm \Delta f = f_0 \pm 0,5 \text{ Hz} \quad (4.16)$$

Corespunzător, intervalul dintre două impulsuri de sincronizare va avea o variație:

$$T = \frac{1}{f_0 \pm 0,5} = T_0 \pm \Delta T = T_0 \pm 0,2 \text{ ms} \quad (4.17)$$

unde  $T_0$  este, evident, 20 ms.

Rezultă o eroare absolută de sintetizare a intervalului de  $60^\circ$  de:

$$\Delta\alpha_1 = \frac{\Delta T}{T_0} \cdot 360^\circ = \frac{0,2}{20} \cdot 360^\circ = 3,6^\circ \quad (4.18)$$

eroare ce apare pe ultimul interval din cadrul unui ciclu de cc-

menzi.

Precizia de realizare a unui interval de  $60^\circ$  este, cum am arătat, influențată și de faptul că numărarea, în microsystem, se face cu o frecvență a impulsului de tact care nu este multiplu întreg al frecvenței rețelei. Se poate admite o eroare absolută de  $10 \mu s$  în contorizare datorată acestei neconcordanțe, pentru că  $10 \mu s$  reprezintă valoarea maximă posibilă a unui increment în numărare. Un interval de  $60^\circ$  este eronat cu:

$$\Delta\alpha'_2 = \frac{10 \mu s}{10 ms} \cdot 180^\circ = 0,18^\circ \quad (4.19)$$

Pe durata unui ciclu această eroare se cumulează, rezultând pentru ultimul interval, dintre comanda tiristoarelor  $T_5$  și  $T_6$ , o eroare absolută de:

$$\Delta\alpha_2 = 5 \cdot \Delta\alpha'_2 = 0,9^\circ \quad (4.20)$$

Așadar, pentru o variație de  $0,5$  Hz a frecvenței rețelei, pe durata unui ciclu de comandă, și o numărătoare eronată cu o unitate între două comenzi succesive, tiristoarele  $T_5$  și  $T_6$  vor fi comandate la  $60^\circ$  interval cu o eroare absolută maximă totală de:

$$\Delta\alpha = \Delta\alpha_1 + \Delta\alpha_2 = 4,5^\circ \quad (4.21)$$

Eroarea relativă maximă în contorizarea intervalului dintre două comenzi este:

$$\xi_r = \frac{\Delta\alpha}{60^\circ} = \frac{4,5^\circ}{60^\circ} = 0,075 \quad (4.22)$$

Eroarea obținută nu este foarte mare și poate fi complet compensată prin mijloace software (program de corecție a erorilor).

Utilizînd acest tip de redresor alimentat de la rețeaua trifazată și comandat cu microsystemele realizate cu microprocesoarele 8085 și Z80, cu programe realizate după organigrama din fig.4.12. s-a realizat practic modificarea turației unui motor de curent continuu cu excitație separată de tip CE 42 E IMEB. Parametrii acestui motor sînt:  $U_g = 220$  V,  $P_n = 2,7$  kW,  $n_{max} = 2300$  rot/min. Modificînd unghiul de comandă între  $60$  și  $90$  grade s-au obținut turații între cea maximă și  $0$ . Incărcarea motorului s-a efectuat prin cuplarea la arborele lui a unui al doilea motor, asincron și trifazat, devenit generator, ce a fost la rîndul său încărcat cu un grup de rezistențe variabile de putere conectate în stea.

Unghiul de comandă nu a fost coborât sub  $60^\circ$  pentru a se evita alimentarea motorului de curent continuu cu tensiune peste cea nominală. Dacă se utilizează, la alimentarea redresorului, un transformator coborîtor de tensiune, de exemplu cu raportul de transformare 1:2, se mărește plaja de reglaj pentru unghiul de comandă de la  $0$  la  $90^\circ$ , dar la unghiuri de comandă de peste  $75^\circ$  poate apare regimul, nedorit, de curent întrerupt.

Depășirea unui unghi de comandă de  $90^\circ$ , implică pentru redresor, intrarea în regim de inverter /37/ caz în care motorul devine generator (dacă și sarcina mecanică poate furniza energia necesară rotirii arborelui motor), Studii asupra acestui regim nu s-au făcut, ele urmînd să constituie obiectul unor cercetări viitoare /64, 70/.

#### 4.3. Sisteme de acționare cu motoare de curent continuu alimentate prin variatoare de tensiune continuă

Variatorul de tensiune continuă, sau chopperul, se utilizează frecvent la reglarea turației motoarelor de curent continuu, fiind un convertor care transformă o tensiune continuă, aplicată la intrare, în impulsuri dreptunghiulare la ieșire. Valoarea medie a tensiunii de la ieșirea unui chopper se poate modifica între zero și

cea a tensiunii de alimentare folosind unul din următoarele principii:

- modificarea frecvenței de repetiție a unor impulsuri de durată constantă;

- modularea în durată a unor impulsuri

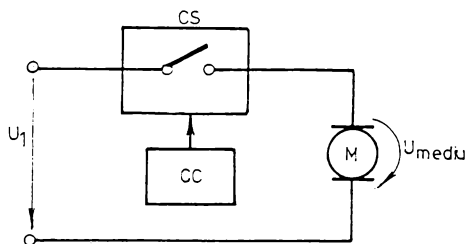


Fig.4.15. Schema de principiu a variatorului de tensiune continuă

de frecvență constantă.

Schema de principiu a unui astfel de sistem de acționare este prezentată în fig.4.15. În această figură, CS reprezintă un contactor static (chopper), CC circuitul său de comandă și M inductorul motorului de curent continuu.

Inchizînd și deschizînd contactorul CS după o secvență repetată periodic motorul va fi alimentat cu tensiunea medie:

$$U_m = \frac{T_c}{T} U_1 \quad (4.23)$$

unde  $T_c$  este durata de conectare a contactorului CS, iar  $T$  este perioada de repetiție a comenzii. Pentru a asigura continuitatea curentului prin motor este necesar ca perioada de comandă a contactorului static să fie mult mai mică decît constanta de timp electromagnetice a motorului /45/.

O variantă concretă pentru contactorul static din variator este chopperul cu stingere forțată, prezentat în fig.4.14.

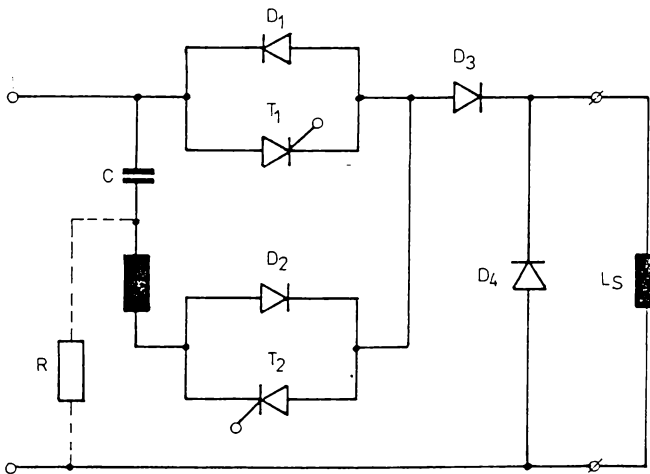


Fig.4.14. Chopper cu stingere forțată

Principalele elemente din schemă au fost notate astfel:  $T_1$  - tiristorul principal,  $T_2$  - tiristorul de stingere,  $C$  - condensatorul de stingere,  $L$  - inductivitatea de stingere,  $D_4$  - dioda de nul.

Funcționarea chopperului se urmărește mai ușor dacă se împarte durata  $T$  de repetiție a comenzilor în șase intervale de timp, marcate de cîte o schimbare de stare (conducție sau blocare) pentru dispozitivele din circuit /81/.

În fig.4.15. sînt reprezentate schemele echivalente ale chopperului pe fiecare dintre cele șase intervale de timp, iar în fig.4.16. sînt reprezentate formele de undă explicative.

În momentul  $t_0$  se comandă tiristorul principal  $T_1$ . Sarcina este alimentată cu tensiunea  $U_1$ . Pentru simplitate se presu-

pune  $L_S$  suficient de mare și curentul prin sarcină constant ( $i_S = I_S$ ).

În momentul  $t_1$  se comandă tiristorul de stingere  $T_2$ . Condensatorul  $C$  este încărcat la o tensiune cu polaritatea din fig.

4.15.b. (Condensatorul s-a încărcat în momentul conectării sursei de alimentare). Curentul rezonant de descărcare a lui  $C$  se anu -

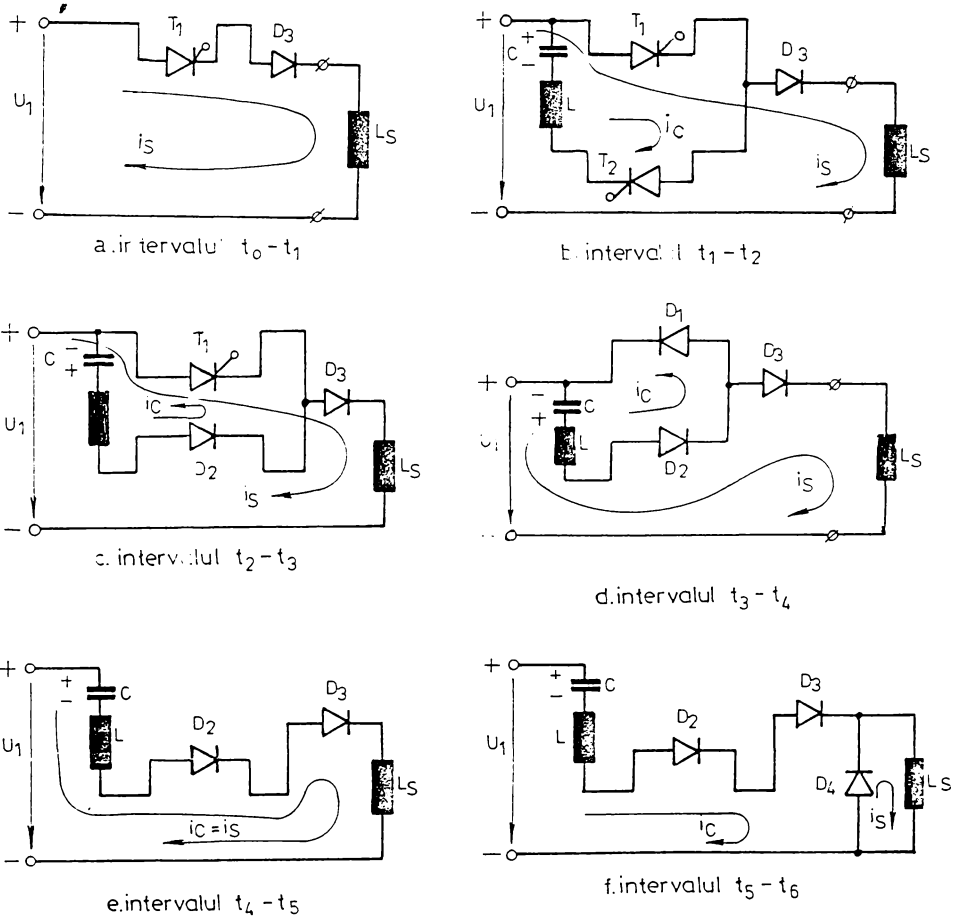


Fig.4.15. Circuitul echivalent al chopperului în diferite momente ale funcționării

lează în momentul  $t_2$  și, prin urmare,  $T_2$  se blochează. Schimbarea sensului curentului  $i_C$  determină, în momentul  $t_3$ , și stingerea tiristorului  $T_1$ , când suma curentilor  $i_C$  și  $i_S$  devine mai mică decât curentul de menținere în conducție al acestui tiristor.

Încărcarea condensatorului  $C$  se continuă și pe intervalele  $t_3 - t_4$ ,  $t_4 - t_5$ ,  $t_5 - t_6$ . Tensiunea la care se încarcă este



mai mare decît cea a tensiunii de alimentare  $U_1$  datorită energiei înmagazinată în inductivitățile  $L$  și  $L_S$ . Din acest motiv se deschide și dioda  $D_4$  - dioda de nul - în momentul  $t_5$ . În momen-

mul  $t_6$ ,  $i_C$  devine nul, iar curentul prin sarcină se menține numai prin  $D_4$ , pînă la o nouă comandă a lui  $T_1$ .

Funcționarea descrisă în detaliu și relațiile de proiectare pentru acest tip de chopper sînt cunoscute [78, 48, 51].

Circuitul utilizat la comanda chopperului trebuie să furnizeze impulsurile de amorsare pentru tiristorul principal  $T_1$  cu o frecvență de repetiție  $T$ . Intrarea în conducție a lui  $T_1$  echivalează cu "închiderea contactului contactorului static". După un interval  $T_C$ , de la comanda lui  $T_1$ , circuitul de comandă generează impulsul de amorsare a tiristorului  $T_2$ , iar sarcina (motorul) este deconectată de la tensiunea de alimentare.

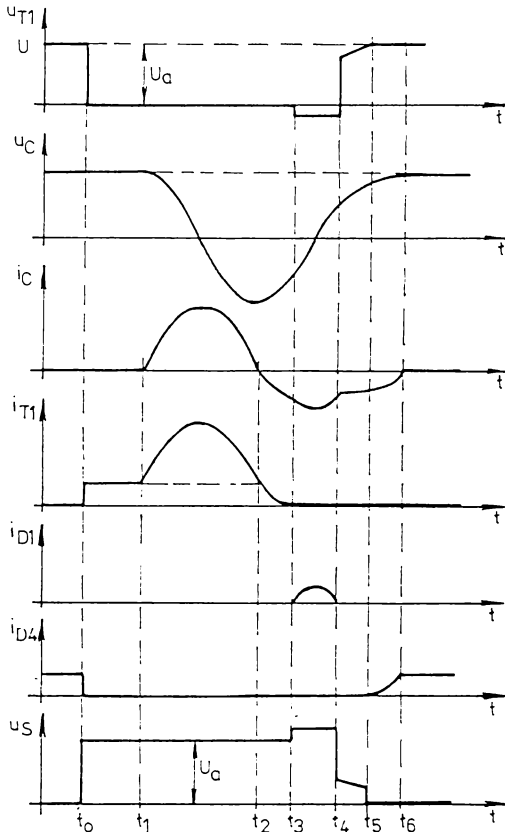


Fig. 4.16. Formele de undă explicative pentru chopperul cu stingere forțată

În fig. 4.17. se prezintă tensiunea (idealizată) la bornele motorului și secvența de impulsuri de comandă pentru chopperul descris.

Menținînd constantă perioada  $T$  de repetiție a impulsurilor de comandă la tiristorul principal și modificînd momentul de apariție al impulsului destinat tiristorului de stingere se

realizează modificarea valorii medii a tensiunii de ieșire și,

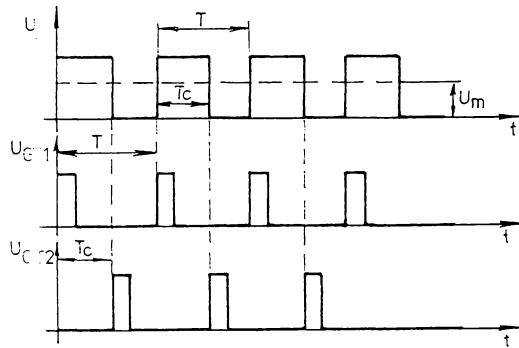


Fig.4.17. Diagrame explicative pentru chopperul cu stingere forțată

implicit, modificarea turației motorului.

Chopperul prezentat are dezavantajul funcționării într-un singur cadran. Pentru a se realiza o acționare reversibilă se utilizează un alt tip de chopper. Există scheme de chopper în două și patru cadrane cu largă răspândire în practică /77/.

În figura 4.18. se prezintă o variantă de chopper care funcționează în patru cadrane, fiind realizată cu tranzistoare /82/.

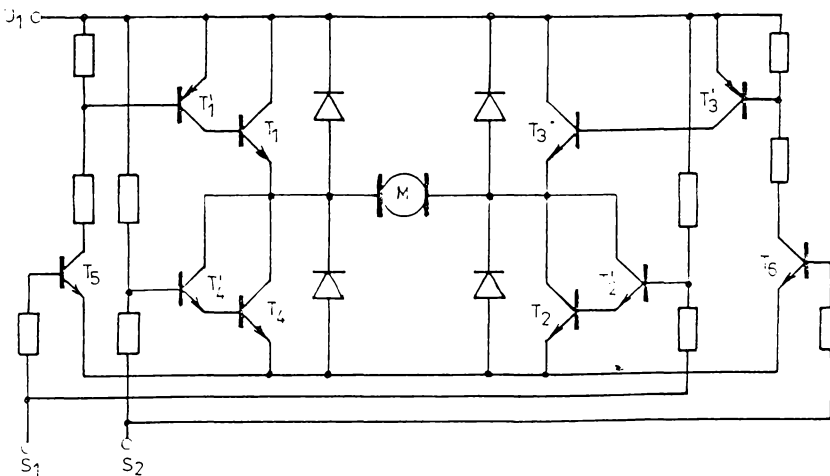


Fig.4.18. Chopper în patru cadrane realizat cu tranzistoare

Motorul de curent continuu, conectat în diagonala de ieșire a punții cu tranzistoare, se poate roti în ambele sensuri. Cele două sensuri se obțin pentru o combinație de semnale logice "0 și 1" respectiv "1 și 0" aplicată la bornele, pentru sens,  $S_1$  și  $S_2$ . Prin intermediul tranzistoarelor  $T_5$  și  $T_6$ , se comandă fie perechea de tranzistoare  $T_1$ - $T_2$ , pentru un sens de rotire, fie perechea  $T_3$ - $T_4$  pentru celălalt sens de rotație, în funcție de combinația aplicată la  $S_1$ - $S_2$ .

#### 4.3.1. Circuite de comandă pentru choppere realizate cu microprocesor

Ca și în cazul redresoarelor, circuitele de comandă pentru choppere, realizate cu mijloace electronice tradiționale, sînt utilizate azi pe scară largă /78, 28/.

Analiza implementării pe microsisteme a circuitelor de comandă pentru choppere se impune nu din necesitatea unei simple înlocuiri a circuitelor tradiționale, ci și din dorința de a realiza circuite de comandă flexibile, care să poată fi incluse în sisteme de conducere numerice mai complexe, fără interfațări suplimentare. Utilizînd microprocesorul, ca circuit de comandă, se pune problema de a realiza și funcții suplimentare sau comanda simultană pentru mai multe motoare. Acest lucru este perfect posibil, întrucît în comanda chopperului microprocesorul trebuie să realizeze numai următoarele funcții:

- (F1) generarea intervalelor de timp  $T$  și  $T_C$  variabile;
- (F2) comanda propriu-zisă a tiristoarelor - cu impulsuri scurte, sau tranzistoarelor - cu impulsuri de durată  $T_C$ .

#### 4.3.2. Exemple de choppere comandate cu microprocesor

În acest paragraf sînt prezentate realizările concrete cu privire la implementarea pe microsistem a circuitelor de comandă pentru choppere.

##### 4.3.2.1. Chopper cu stingere forțată comandat cu microprocesor

Structura circuitului de putere utilizat în aplicația descrisă în acest paragraf este cea din fig.4.4. Impulsurile

de comandă pentru tiristoare se obțin pe două linii ale unuia dintre porturile de ieșire ale microsistemului. Izolarea galvanică între sistem și circuitul de putere se realizează prin amplificatoare cu transformatoare de impulsuri (paragraful 3.1. fig.3.2.).

Mărimile de intrare pentru circuitul de comandă sînt perioada de repetiție  $T$  și durata de conducție  $T_C$  pentru tiristorul principal. Aceste mărimi pot fi depuse în memoria sistemului de către utilizator prin consolă /36/ sau de către un sistem ierarhic superior /11/ respectiv, pot fi aduse, din exterior, și pe unul dintre porturile de intrare ale sistemului.

Secvența de impulsuri necesară se poate obține prin mijloace software sau prin utilizarea generatorului de timp real și a sistemului de întreruperi.

În organigrama din fig.4.19. este reprezentat un exemplu de program care realizează comanda chopperului.

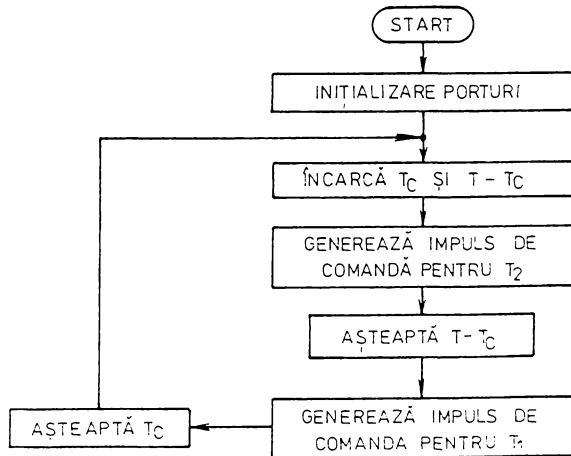


Fig.4.19. Organigrama programului de comandă pentru chopper

Programele detaliate pentru comanda chopperului utilizînd microprocesoarele 8035, 8085 și Z80 sînt cele ce urmează /13.11/.

Se constată, ca și la redresor, o diferență nesemnificativă între numărul de instrucții pentru cele trei variante de program, corespunzătoare celor trei familii de microprocesoare. Ca urmare, este mai avantajoasă utilizarea microprocesorului 8035 ca circuit simplu de comandă pentru chopper.

<u>8035</u>	<u>8085</u>	<u>Z80</u>
DIS TCNT I	MVI A,03	LD C,61H
JMP EO	OUT 20H	LD D,02
IN A,P2	E1: LXI H,83AOH	LD HL,T-T <sub>C</sub>
JB7 E1	MOV D,M	EXX
IN A,P1	INR L	LD C,61H
MOV R6,A	MOV E,M	LD D,01
JMP E2	MVI A,01	LD HL,T <sub>C</sub>
E1: IN A,P1	OUT 21H	E1: OUT (C),D
MOV R7,A	CALL IMP	LD B,t <sub>i</sub>
E2: JFO E3	INR L	IM: DJNZ IM
RETR	MOV B,M	OUT (C),B
EO: CPL FO	INR L	PUSH HL
MOV A,#80	MOV C,M	AST: DEC HL
OUTL P2,A	CALL AST	LD A,L
ORL P1,#FF	MVI A,02	OR A,H
ENI	OUT 21H	JR NZ,AST
HLT: JMP HLT	CALL IMP	POP HL
E3: CLR A	MOV C,E	EXX
MOV PSW,A	MOV B,D	JR E1
E4: ORL P1,FF	CALL AST	
ENI	JMP E1	
E8: MOV R4,#t <sub>i</sub>	AST: DCX B	
E5: DJNZ R4,E5	MOV A,C	
A : ANL P2,#80	ORA B	
MOV T,A	JNZ AST	
STRT T	RET	
E7: JTF E6	IMP: LXI B,00t <sub>i</sub>	
JMP E7	CALL AST	
E6: STOP TCNT	OUT 21H	
RETR	RET	
MOV A,#90		
OUTL P2,A		
MOV A,R7		
CALL E8		
MOV A,#A0		
OUTL P2,A		
MOV A,R6		
CALL E8		
JMP E4		

Dacă se folosesc mijloacele software pentru contorizarea intervalelor de timp, microprocesorul este utilizat neeconomic, cu rol de numărător simplu. Un microsistem poate executa, desigur, programe mai complexe.

În /11/ se prezintă, de exemplu, un program care realizează trei trepte de turație diferite pentru motor. Fiecare treaptă este menținută un anumit interval de timp, conform unui ipotetic proces tehnologic. Atât valorile turațiilor cât și durata de menținere a unei anumite trepte pot fi modificate după necesități. Organigrama programului este dată în fig.4.20. și 4.21. Ulterior acest program a fost extins la mai multe trepte de turație realizabile /63, 90/. O variantă de program care realizează 20 de trepte de viteză pentru motor este cea de mai jos:

ORG 8000H

0001 E1 EQU 9190H	0190 CPI 93H
0002 E2 EQU 9194H	0200 JZ A1
0003 E3 EQU 919AH	0210 CPI 99H
0004 E4 EQU 91A0H	0220 JZ A2
0010 START MVI A,01H	0230 CPI 9FH
0020 OUT 28H	0240 JZ A3
0030 IAR LXI H,E1	0250 CPI 0A5H
0040 REF MOV D,M	0260 JZ A4
0050 INR L	0270 CPI 0ABH
0060 MOV E,M	0280 JZ A5
0070 MVI A,01H	0290 CPI 0B1H
0080 OUT 29H	0300 JZ A6
0090 CALL IMP	0310 CPI 0B7H
0070 INR L	0320 JZ A7
0080 MOV B,M	0330 CPI 0BDH
0090 INR L	0340 JZ A8
0100 MOV C,M	0350 CPI 0C3H
0110 CALL AST	0360 JZ A9
0120 MVI A,02H	0370 CPI 0C9H
0130 OUT 29H	0380 JZ A10
0140 CALL IMP	0390 CPI 0CFH
0150 MOV C,E	0400 JZ A11
0160 MOV B,D	0410 CPI 0D5H
0170 CALL AST	0420 JZ A12
0180 MOV A,L	0430 CPI 0DEH

0440	JZ A13	0830	CALL TR
0450	CPI OE1H	0840	A15 LXI H,E16
0460	JZ A14	0850	CALL TR
0470	CPI OE7H	0860	A16 LXI H,E17
0480	JZ A15	0870	CALL TR
0490	CPI OEDH	0880	A17 LXI H,E18
0500	JZ A16	0890	CALL TR
0510	CPI OF3H	0900	A18 LXI H,E19
0520	JZ A17	0910	CALL TR
0530	CPI OF9H	0920	A19 LXI H,E20
0540	JZ A18	0930	CALL TR
0550	CPI OFFH	0940	LXI H,E20
0560	JZ A19	0950	CALL INC
0570	A1 LXI H,E2	0960	LXI H,E3
0580	CALL TR	0970	CALL INC
0590	A2 LXI H,E3	0980	LXI H,E4
0600	CALL TR	0990	CALL INC
0610	A3 LXI H,E4	1000	LXI H,E5
0620	CALL TR	1010	CALL INC
0630	A4 LXI H,E5	1020	LXI H,E6
0640	CALL TR	1030	CALL INC
0650	A5 LXI H,E6	1040	LXI H,E7
0660	CALL TR	1050	CALL INC
0670	A6 LXI H,E7	1060	LXI H,E8
0680	CALL TR	1070	CALL INC
0690	A7 LXI H,E8	1080	LXI H,E9
0700	CALL TR	1090	CALL INC
0710	A8 LXI H,E9	1100	LXI H,E10
0720	CALL TR	1110	CALL INC
0730	A9 LXI H,E10	1120	LXI H,E11
0740	CALL TR	1130	CALL INC
0750	A10 LXI H,E11	1140	LXI H,E12
0760	CALL TR	1150	CALL INC
0770	A11 LXI H,E12	1160	LXI H,E13
0780	CALL TR	1170	CALL INC
0790	A12 LXI H,E13	1180	LXI H,E14
0795	CALL TR	1190	CALL INC
0800	A13 LXI H,E14	1200	LXI H,E15
0810	CALL TR	1210	CALL INC
0820	A14 LXI H,E15	1220	LXI H,E16

123o	CALL INC	152o	DCR L
124o	LXI H,E17	153o	DCR L
125o	CALL INC	154o	DCR L
126o	LXI H,E18	155o	MOV A,C
127o	CALL INC	156o	ORA B
128o	LXI H,E19	157o	RZ
129o	CALL INC	158o	POP D
130o	LXI H,E2o	159o	JMP REF
131o	CALL INC	160o	INC MVI M,OH
132o	JMP JAR	161o	INR L
133o	AST DCX B	162o	MVI M,1OH
134o	DCX D	163o	RET
135o	MOV A,C	164o	E5 EQU 91A6H
136o	ORA B	165o	E6 EQU 91ACH
137o	JNZ AST	166o	E7 EQU 91B2H
138o	RET	167o	E8 EQU 91B8H
139o	IMP MVI C,07H	168o	E9 EQU 91BEH
140o	MVI B,OH	169o	E1o EQU 91C4H
141o	CALL AST	170o	E11 EQU 91CAH
142o	OUT 29H	171o	E12 EQU 91DOH
143o	RET	172o	E13 EQU 91D6H
144o	TR MOV B,M	173o	E14 EQU 91DCH
145o	INR L	174o	E15 EQU 91E2H
146o	MOV C,M	175o	E16 EQU 91E8H
147o	DCX B	176o	E17 EQU 91EEH
148o	MOV M,C	177o	E18 EQU 91F4H
149o	DCR L	178o	E19 EQU 91FAH
150o	MOV M,B	179o	E2o EQU 920OH
151o	DCR L		

Utilizând sistemul de întreruperi și timerul din sistem, microprocesorul este degrevat complet de sarcina contorizărilor și poate executa un alt segment al programului principal. In capitolul 5 în care se tratează regulatoarele de viteză și poziție, realizate cu microsistem, este descris un asemenea mod de lucru.

In realizările concrete, perioada de repetiție utilizată a fost  $T = 10 \text{ ms}$ . S-a ales acest interval de timp întrucât constanta de timp electromecanică a motoarelor comandate este mai mare de  $150 \text{ ms}$  și utilizând o perioadă de repetiție, în comandă, de



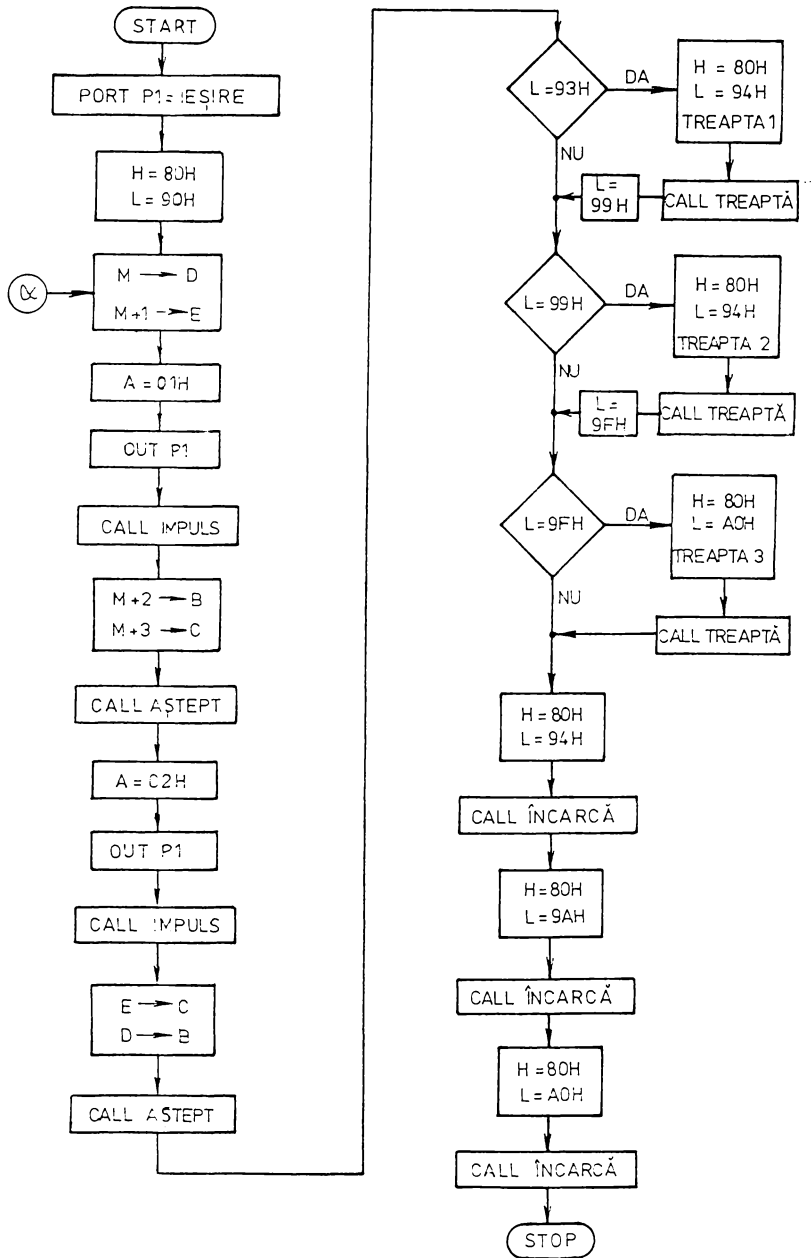
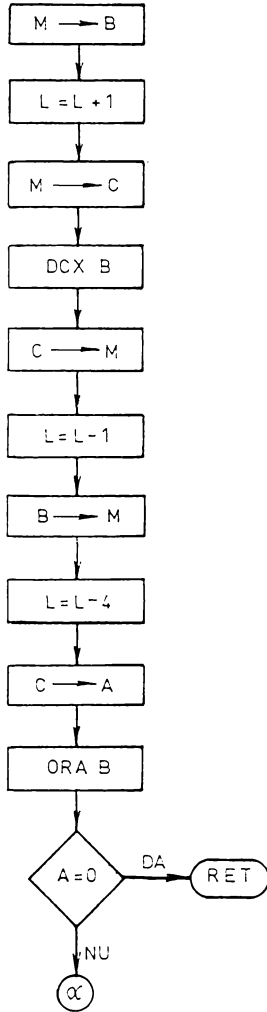
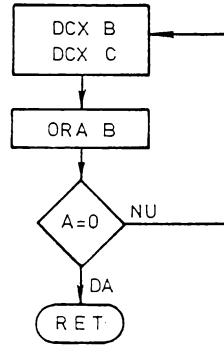


Fig.4.2o. Organigrama programului de comandă prin chopper a 3 trepte de viteză pentru un motor de c.c.

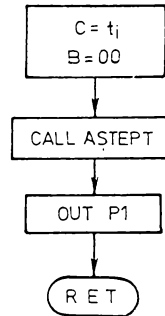
TREAPTĂ:



AȘTEPT:



IMPULS:



ÎNCARCĂ:

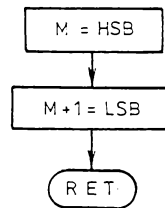


Fig.4.21. Subrutinele programului din fig.4.20

15 ori mai mică nu se ajunge la funcționarea motorului în regim de curent întrerupt /37/. Intervalele, variabile, de timp  $T_C$ , de conducție pentru tiristorul principal, s-au realizat de valori cuprinse între 0 și 9,496 ms.

Valoarea maximă pentru  $T_C$  este limitată de intervalul de timp minim dintre comanda tiristorului de stingere și comanda tiristorului principal, pentru a se asigura o stingere sigură a acestuia din urmă. Durata minimă necesară pentru stingerea tiristorului principal este fixată de elementele de stingere L-C din schemă (fig.4.14.). În montajul concret utilizat, aceste elemente au valorile:  $L = 25 \mu\text{H}$  și  $C = 105 \mu\text{F}$ , valori ce au fost calculate avîndu-se în vedere datele de catalog ale tiristoarelor din schemă, tocmai în vederea asigurării stingerii sigure a tiristorului principal. În aceste condiții, tiristorul principal poate fi comandat numai după intervalul minim:

$$\Delta T = 2\pi\sqrt{LC} = 322 \mu\text{S} \quad (4.24)$$

de la comanda tiristorului de stingere. S-a ales pentru siguranță  $\Delta T = 500 \mu\text{S}$  (în fapt  $504 \mu\text{S}$ ) rezultînd un factor de umplere  $T_C/T \cdot 100$  de maximum 95%.

Valoarea incrementului, la contorizarea duratei  $T_C$ , s-a obținut de  $8 \mu\text{S}$  în cazul numărării cu mijloace software, iar în aplicația în care se utilizează timerul incrementul pentru  $T_C$  a fost de  $0,33 \mu\text{S}$ , folosind microprocesorul 8085 cu frecvența impulsului de tact de  $3,14 \text{ MHz}$ . Rezoluția obținută fiind:

$$\frac{8 \cdot 10^{-6}}{10 \cdot 10^{-3}} = 0,8 \cdot 10^{-3} \quad (4.25)$$

într-un caz și

$$\frac{0,33 \cdot 10^{-6}}{10 \cdot 10^{-3}} = \frac{1}{30} \cdot 10^{-3} = 3,33 \cdot 10^{-6} \quad (4.26)$$

în cel de al doilea caz.

Variatorul de turație de tip chopper cu stingere forțată, a fost folosit pentru comanda a două motoare de curent continuu:

Un motor de mică putere, tip EP 211 (IME București) cu excitația cu magnet permanent și următoarele date de catalog:  $U_n = 24 \text{ V}$ ,  $I_n = 3,5 \text{ A}$  și  $n_{\text{max}} I = 1620 \text{ rpm} \pm 12\%$ , respectiv  $n_{\text{max}} II = 2160 \text{ rpm} \pm 12\%$  fiind prevăzut cu priză pe înfășurarea rotorică; sursa de tensiune stabilizată utilizată la alimentarea variatorului pentru acest motor este de tip I 4104 (IEMI

București).

Cel de al doilea motor utilizat este un motor de curent continuu cu excitație separată de tip CE 24 E (IME București) cu parametrii  $U_n = 220$  V,  $P = 2,7$  KW și  $n_{max} = 2300$  rpm. Tensiunea continuă de alimentare a fost obținută de la rețea cu un redresor în punte necomandată.

Experimentările efectuate, cu ambele motoare, au demonstrat că nu există nici un fel de limitări cu privire la turația obținabilă.

#### 4.3.2.2. Chopper cu tranzistoare comandat prin microprocesor /82/

Structura circuitului de putere este prezentată în fig. 4.18.

Semnalele de comandă pentru tranzistoare se obțin prin programarea corespunzătoare a două linii ale unuia dintre porturile de ieșire din microsistem. Izolarea galvanică între circuitul de putere și microsistem se realizează utilizând optocuploare /82/.

În funcție de sensul de rotație a motorului, necesar la un moment dat, una dintre cele două linii, dintr-un port de ieșire, utilizate la comanda chopperului, este poziționată pe nivelul 0 logic (TTL). O a doua linie este poziționată pe 1 logic un timp  $T_C$  - timp de conducție, apoi pe 0 logic un timp  $T - T_C$  - timp de pauză, după un ciclu care se repetă de câte ori este necesar. Pe cea de a doua linie de ieșire se obține, astfel, un tren de impulsuri cu frecvența de repetiție  $1/T$  constantă, dar cu factor de umplere variabil, funcție de viteza cu care trebuie să se rotească motorul. Pentru celălalt sens de rotație, impulsuri de comandă necesare se generează pe prima linie a portului de ieșire utilizat, iar cea de a doua linie se menține pe 0 logic.

Fiecare secvență de impulsuri, pentru câte un sens, poate fi menținută un anumit interval de timp ce se impune din necesități tehnologice. Pot fi realizate, în acest mod, diferite trepte de turație pentru motor, iar fiecare treaptă se poate menține pe intervale de timp diferite.

Organigrama unui astfel de program este prezentată în fig.4.22.

Programul utilizează o listă de date din memoria siste-

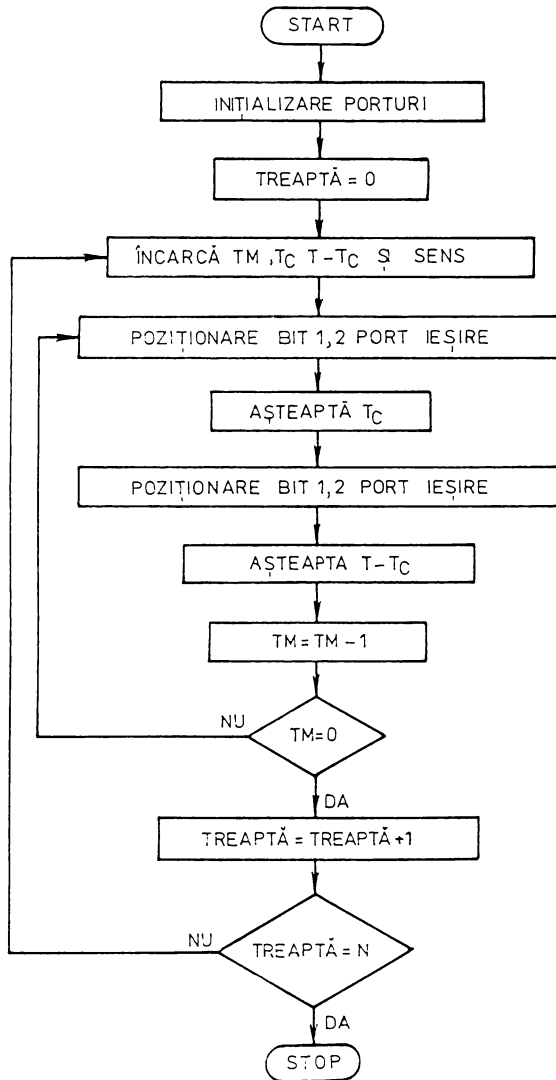


Fig.4.22. Organigrama programului de comandă a chopperului cu tranzistoare

mului. Lista conține 16 biți (2 cuvinte de memorie) ce reprezintă durata de menținere a unei anumite trepte de viteză (TM), 16 biți pentru durata de conducție ( $T_C$ ), 16 biți pentru durata pauzei  $T-T_C$ , și 8 biți pentru sens, pentru fiecare treaptă de viteză.

Datele din memorie sînt aduse în unitatea de comandă a sistemului succesiv și semnalele de comandă necesare se trimit la portul de ieșire. După epuizarea timpului de menținere al unei trepte, programul știe care treaptă a fost tocmai terminată și selectează imediat lista de date următoare sau se oprește dacă toate treptele s-au efectuat.

Chopperul cu tranzistoare comandat prin microprocesor s-a folosit ca variator de turație pentru un motor de curent continuu cu excitație prin magnet permanent tip EP 211, prezentat și în paragraful precedent. S-au obținut practic toate treptele de viteză între zero și cea maximă, fără nici un fel de limitări. Considerațiile cu privire la rezoluția de realizare a unei trepte de viteză sînt similare cu cele din cazul chopperului cu stingere forțată.

#### 4.4. Exemple de sisteme de acționare cu motor de curent continuu comandate numeric

##### 4.4.1. Principiul modulării în lățime de puls (PWM)

Acționările de curent continuu fac posibilă reglarea vitezei /38/ grație controlului tensiunii și a curentului rotorului. În ultimii ani s-a observat /38/ generalizarea a două metode de comandă a motoarelor de c.c.:

Prima metodă folosește un redresor comandat. Pentru funcționare în patru cadrane, sînt folosite două redresoare antiparalele, făcînd posibilă frînarea recuperativă atunci cînd viteza scade.

Cea de a doua metodă utilizează variatorul cunoscut sub numele de sistem cu modulație în lățime a pulsului (pulse width modulation - PWM). Frecvența maximă de comutare a tranzistoarelor din chopperul final poate fi 4 kHz cînd tranzistoarele sînt bipolare și 20 kHz cînd se folosesc tranzistoare MOSFET.

În varianta cunoscută și utilizată pe scară largă, inclusiv în echipamentele de conducere pentru roboți industriali,

sistemul PWM este utilizat în structuri hibride numeric-analogice (conform structurii bloc din fig.2.6.). Schema bloc a unui sistem de acționare numeric-analogic este cea din fig.4.23./15/.

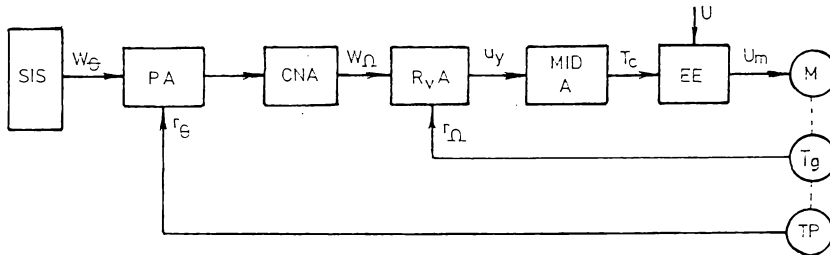


Figura 4.23. Schema bloc a unui sistem de acționare numeric-analogic pentru o axă

Blocul PA (procesor de ax) construit cu microprocesor, asigură conducerea sistemului de acționare. Acest bloc preia mărimea de prescriere pentru poziție ( $W_{\theta}$ ) de la sistemul ierarhic superior, preia informația numerică de reacție ( $r_{\theta}$ ) pentru poziția curentă de la traductorul incremental TP, realizează calculele aferente regulatorului numeric de poziție, generează, în formă numerică, mărimea de prescriere pentru regulatorul de viteză.

Blocul următor CNA este un convertor numeric-analogic ce transmite regulatorului analogic  $R_{vA}$  de viteză mărimea de prescriere  $W_{\Omega}$  pentru turația impusă motorului. Reglarea de viteză se asigură analogic cu blocul  $R_{vA}$ , pentru care mărimea de reacție este culeasă de la tahogeneratorul  $T_g$  cuplat la axul motorului.

Principiul modulării în durată (MID) sau în lățime de puls (PWM) constă în comanda unui chopper cu tranzistoare, în patru cadrane (vezi fig.4.18.) cu impulsuri, cu frecvența de repetiție constantă și durata variabilă funcție de turația ce trebuie realizată la un moment dat.

În fig.4.24. se prezintă modul în care se sintetizează impulsuri cu durată variabilă prin mijloace analogice.

Mărimea de comandă analogică  $u_y$  generată de regulatorul analogic de viteză se compară cu tensiunea în dinte de fierăstrău  $u_t$ , generată intern în modulatorul analogic (MID).

Între  $u_y$  și  $u_t$  există relația /66/:

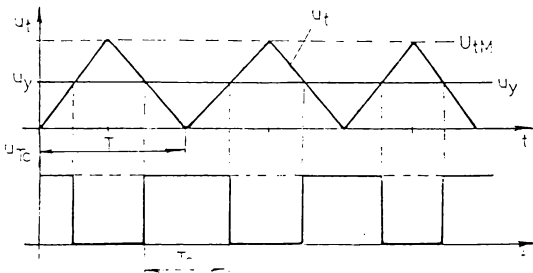
$$u_t = U_{tM} \frac{t}{T/2} \quad (4.27)$$

pentru  $t = \frac{T_c}{2}$

$$u_y = U_{tM} \frac{T_c}{T} \quad (4.28)$$

sau

$$T_c = \frac{u_y}{U_{tM}} T \quad (4.29)$$



In acest fel durata  $T_c$  a impulsurilor de comandă pentru chopperul cu tranzistoare (blocul EE - element de execuție) este proporțională cu tensiunea  $u_y$  pentru că  $T$  și  $U_{tM}$  sînt constante.

Tensiunea la bornele motorului de curent continuu este:

Figura 4.24. Sintetizarea analogică a impulsurilor cu lățime variabilă în sistemul PWM.

$$U_m = \frac{T_c}{T} U \quad (4.30)$$

dacă  $U$  este tensiunea continuă (constantă) de alimentare a chopperului (EE).

#### 4.4.2. Sistem de acționare PWM "exclusiv numeric"

In fig.4.25. se prezintă schema bloc a unui sistem de acționare cu motor de curent continuu, pentru o axă a unui robot

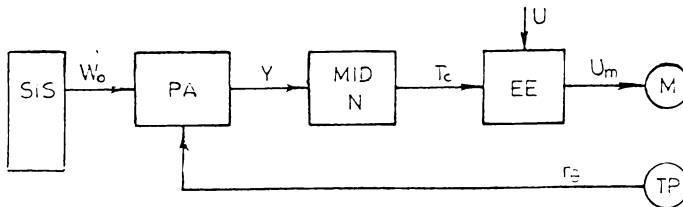


Figura 4.25. Schema bloc a unui sistem de acționare numeric pentru o axă a unui robot.



industrial, realizat în variantă "exclusiv numerică" /67/.

În această schemă modulatorul numeric în durată (MIDN) este realizat ca bloc independent neinclus în procesorul de axă PA. Cu  $T_c$  s-a notat și în acest caz durata impulsurilor, a căror perioadă de repetiție este  $T$  și cu care se comandă chopperul cu tranzistoare în patru cadrane care constituie elementul de execuție EE.  $U_m$  este valoarea medie a tensiunii la bornele motorului de curent continuu. Traductorul incremental de poziție TP măsoară poziția curentă  $\theta$ . Procesorul de ax asigură prin mijloace software reglarea de poziție și viteză. Rolul MIDN este de a transforma mărimea numerică de comandă  $Y$  în durata de conducție  $T_c$  a tranzistoarelor chopperului.

Schema bloc a modulatorului numeric în durată este cea din fig.4.26.

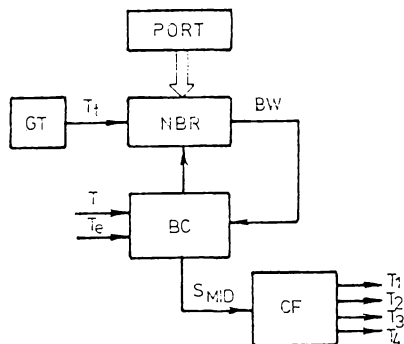


Figura 4.26. Modulator în durată numeric

De la portul de ieșire al procesorului de ax se preia mărimea  $Y$  (un număr) obținută prin calcul de reglatoare.

În principiu MIDN constă dintr-un numărător reversibil NBR inițializat cu numărul binar  $Y$  și decrementat cu impulsurile generate de generatorul de tact GT. Perioada de repetiție  $T_t$  a impulsurilor de tact este stabilă și impusă de realizarea condiției:

$$T_c = Y T_t \quad (4.31)$$

Intrucât timpii de comutație a tranzistoarelor din chopperul final EE sînt, în condițiile disponibilităților de astăzi, de ordinul microsecunde perioada de repetiție  $T$  a impulsurilor de comandă furnizate la ieșirea MIDN trebuie să fie de ordinul sutei sau sutelor de microsecunde. O astfel de valoare pentru perioada de repetiție a impulsurilor de comandă este acoperitoare (fiind mai mică decît constanta de timp electrică a motoarelor utilizate în robotică) și pentru asigurarea regimului de curent neîntrerupt prin motor. Totodată, această valoare asigură un răspuns rapid al modulatorului în durată și al elementului de execuție fiind mai mică decît perioada de eșantionare cu ca-

re operează procesorul de ax.

Rezoluția  $R$  a sistemului de modificare a tensiunii la bornele motorului este egală cu numărul de valori discrete pentru valoarea medie a tensiunii la bornele motorului  $0 \leq U_m \leq U$ . În acest fel perioada de repetiție  $T_t$  a semnalului de tact este determinată de:

$$T_t \leq \frac{T}{R} \quad (4.32)$$

În cazul aplicațiilor de robotică se consideră satisfăcătoare o rezoluție  $R$  la modificarea vitezei de ordinul a 1000 până la 2000, cu (4.32) și considerînd  $T = 100 \mu s$  rezultă ca necesară o frecvență de repetiție de  $10 \div 20$  MHz.

Blocul BC comandă reîncărcarea NBR cu numărul  $Y$  la fiecare perioadă de repetiție  $T$  și citirea unui nou număr  $Y$  la începutul fiecărei perioade de eșantionare (utilizată în procesorul de ax)  $T_e$ .

Blocul comutator final CF este un distribuitor de impulsuri, de la care se obțin impulsurile de comandă nemijlocită a tranzistoarelor  $T_1 - T_4$  ce constituie chopperul etaj final.

Diagrama de impulsuri, explicativă, pentru funcționarea MIDN este reprezentată în fig.4.27.

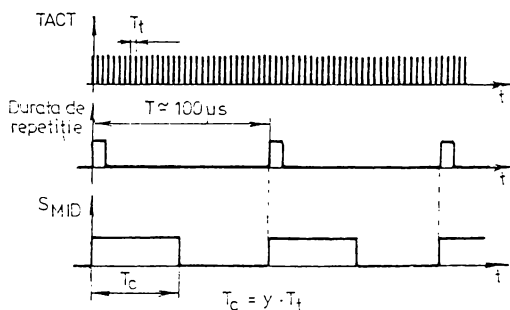


Figura 4.27. Diagrama de impulsuri pentru MIDN

#### 4.4.3. Echipamentul pentru conducerea mișcărilor Creonics WME bus MCC

Placa MCC (Motion Control Card) realizată în 1987 de firma Creonics reprezintă un echipament de comandă construit cu microprocesor, care asigură comanda simultană a două axe acționate cu motor de curent continuu, utilizînd mijloace exclu-

siv numerice /ol/. Schema de principiu a echipamentului este cea din fig.4.28.

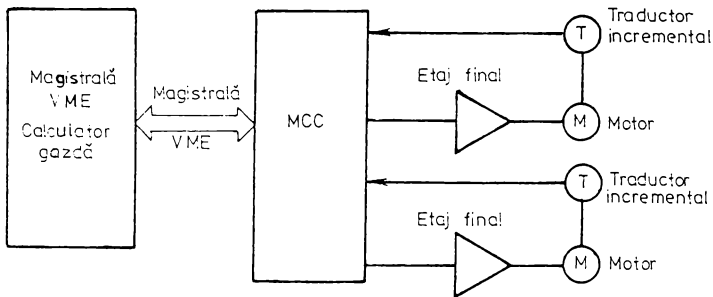


Figura 4.28. Schema bloc de principiu a echipamentului MCC Creonics

Conform filei de catalog, echipamentul realizează următoarele performanțe:

- comanda simultană pentru două axe
- bucle de reglare serie, complet numerice, de performanță, cu regulator PI de viteză și cu reacție în avans (feed forward) - (nu se folosesc algoritme de reglare moderne)

- perioadă de eșantionare și calcul: lms
- reprezentarea poziției pe 32 biți
- viteze și accelerații programabile cu precizie specificată pe 32 de biți

(nu se arată modalitatea prin care se realizează etajele finale de tip chopper și modul de implementare a comenzii PWM cu reprezentare pe 32 de biți a vitezei prescrise; decrementarea unui număr  $2^{32} = 4,29 \cdot 10^9$  într-un interval uzual de  $100 \mu s$  ( $10 \text{ kHz}$ ) ar necesita un impuls de tact de cca  $4 \cdot 10^{13} \text{ Hz}$  în etajul final !)

- posibilitatea acordării software a reguletoarelor
- funcții de protecție în caz de avarii
- funcții auxiliare: - acțiune sincronă pentru cele două axe

- generarea unui profil de poziții cu 8000 de puncte de precizie pe o axă sau câte 4000 de puncte de precizie pe cele 2 axe.

Schema bloc a structurii echipamentului este cea din fig. 4.29.

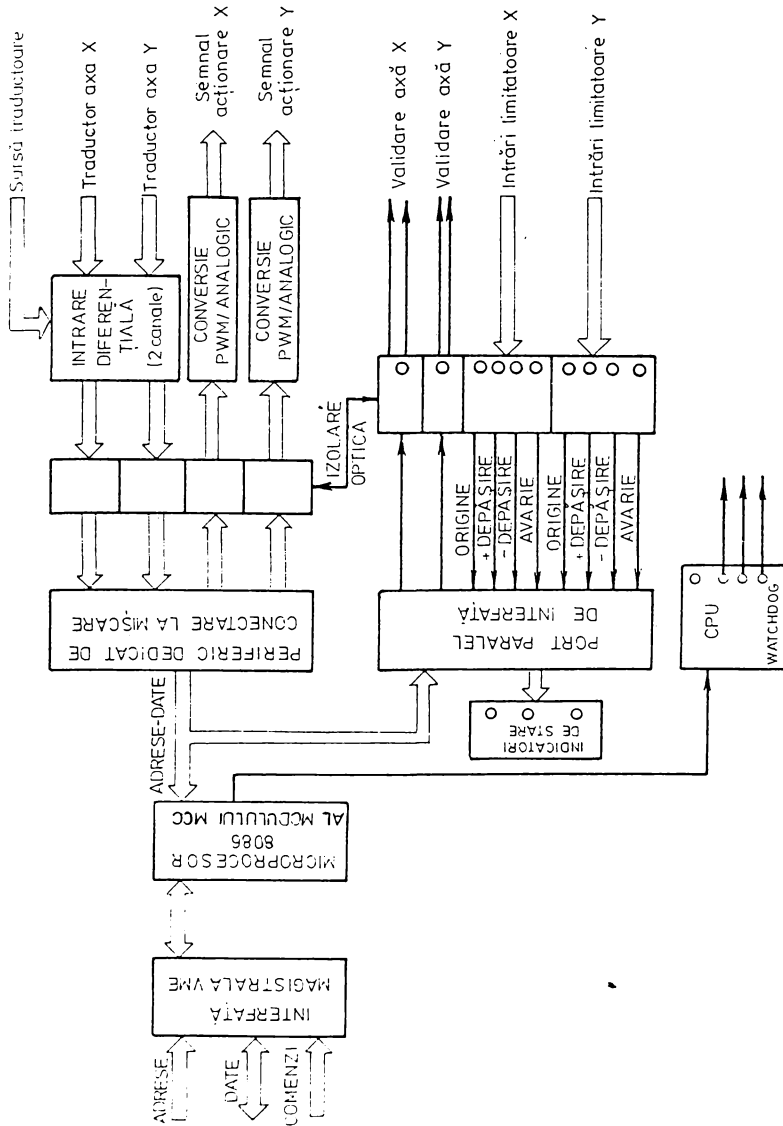


Figura 4.29. Schema bloc a echipamentului MMC Cronics

Echipamentul este construit cu microprocesorul 8086 care realizează funcțiile aferente reguletoarelor, sincronizării celor două axe și generării punctelor de precizie în cazul realizării unui anumit profil (camă). Conține procesorul dedicat CX 2216 (LSI) care realizează interfațarea traductoarelor incrementale, generarea impulsurilor de comandă PWM pentru etajele finale de putere și alte funcții (nespecificate).

Echipamentul a fost conceput spre a fi direct conectabil cu o magistrală standard VME prin care primește comenzi de la un alt calculator, conținând o interfață pentru magistrala respectivă.

Preluarea referinței de poziție se realizează direct de la traductoare incrementale (izolate optic) sau analogic de la traductoarele analogice (de tip rezolver).

Generarea comenzii pentru etajul final este realizabilă în două moduri:

- direct numeric (nivel TTL sau CMOS), cu semnal modulat în durată (PWM) pentru comanda unui chopper (fără să se specifice modalitatea concretă, și dificil de intuit metoda folosită);

- analogic, cu un semnal 0 - 10 V, pentru comanda unor variatoare cu intrare analogică. Pe schema bloc din fig.4.29.

101/ firma nu a reprezentat decât ieșirile analogice.

Suplimentar echipamentul permite un schimb de semnale de intrare (limită de cursă, poziție inițială, avarie în etajul final) și de ieșire (validare acționare) și are prevăzut un circuit cu ceas de gardă ("watchdog") pentru sesizarea defectării sale de către calculatorul gazdă.

De la calculatorul gazdă echipamentul MCC Creonics primește, prin magistrala VME, comenzi codificate numeric, programabile software de tipul: MOVE, noua poziție, noua viteză, citire poziție, citire viteză, introducere constante algoritmi, oprire în caz de eroare ș.a.

Cu toate scăpările în prezentare, din fila de catalog se poate concluziona că modulul MCC Creonics este un echipament de mare complexitate, un întreg microsistem realizat cu un număr mare de componente, între care un microprocesor pe 16 biți și un procesor dedicat, ce realizează performanțe deosebite și un număr mare de funcții. Prin modul cum este conceput, MCC Creonics reprezintă un pas important în realizarea sistemelor

de acționare "exclusiv numerice", deși la un preț de cost apreciabil și cu aspecte încă insuficient clarificate.

În finalul filei de calog se specifică: "Informația conținută în acest document poate fi subiectul unor schimbări fără a fi observate și acestea nu trebuie să fie interpretate ca fiind comise de Creonics Inc. Creonics nu își asumă nici o responsabilitate pentru orice eroare ce poate apare".

CAPITOLUL 5.

SISTEM DE REGLARE NUMERICA A VITEZEI SI POZITIEI  
INTR-O ACTIONARE CU MOTOR DE CURENT CONTINUU  
COMANDATA CU MICROPROCESOR

5.1. Structura sistemelor de reglare utilizate

În alegerea schemelor ce au fost proiectate și realizate, în varianta numerică, s-a pornit de la principalele tipuri de structuri ale sistemelor de reglare a vitezei și poziției, cunoscute în acționările cu motor de curent continuu.

5.1.1. Tipuri de structuri ale sistemelor de reglare automată a vitezei și poziției /3 /

În scopul reglării automate a vitezei și poziției unui motor de curent continuu (MCC), se folosesc sisteme ce pot avea diverse structuri, în funcție de cerințele impuse reglării /60, 43/.

Cea mai simplă structură, folosită doar pentru reglarea vitezei, este prezentată în fig.5.1.

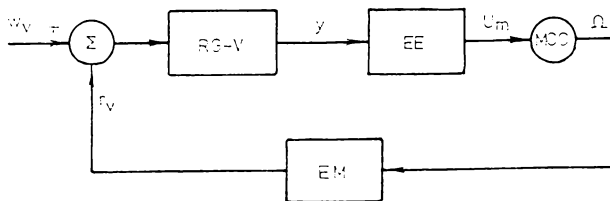


Figura 5.1. Structura unui sistem de reglare a vitezei unui motor de c.c.

Notațiile de pe figură, conforme cu cele curent folosite în automată, semnifică:

- $w_v$  - viteza prescrisă,
- $r_v$  - viteza măsurată (reacția de viteză),
- RG-V - regulatorul de viteză,
- $y$  - mărimea de comandă,
- EE - elementul de execuție,
- $U_m$  - mărimea de execuție (tensiunea medie aplicată la bornele motorului),
- $\Omega$  - viteza unghiulară,
- EM - elementul de măsură.

Pentru a se asigura atât reglarea vitezei cât și a poziției, sînt necesare două bucle de reglare. Acestea se pot lega în cascadă (fig.5.2.) sau în paralel (fig.5.3.). Notațiile de pe figuri au semnificații similare cu cele prezentate pentru fig. 5.1.

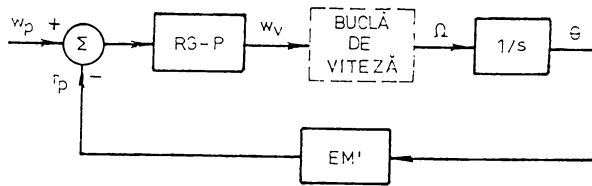


Figura 5.2. Structura unui sistem de reglare a vitezei și poziției cu două bucle de reglare legate în cascadă

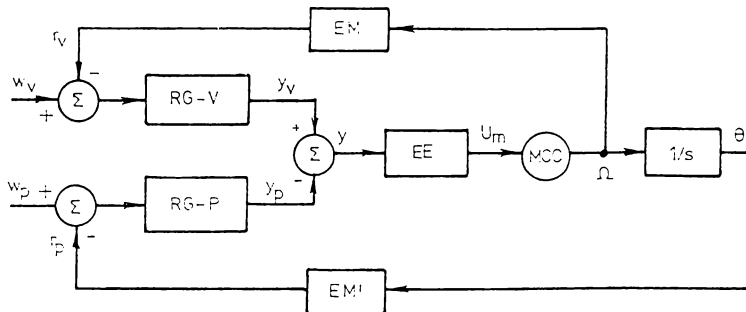


Figura 5.3. Structura unui sistem de reglare a vitezei și poziției cu două bucle de reglare legate în paralel

Reglatoarele folosite uzual sînt de următoarele tipuri:



- proporțional (P), cu funcția de transfer:  $H_{RG}(s) = K_R$ ,
- integrator (I),  $H_{RG}(s) = K_R/s$ ,
- proporțional-integrator (PI),  $H_{RG}(s) = K_R(1 + 1/sT_i)$ ,
- proporțional-derivator (PD),  $H_{RG}(s) = K_R(1 + sT_d)$ ,
- proporțional-integrator-derivator (PID),  $H_{RG}(s) = K_R(1 + 1/sT_i + sT_d)$ .

Structura lor poate fi analogică sau numerică. O variantă analogică de realizare a schemei din fig.5.2., care include și o buclă internă de curent, este prezentată în fig.5.4. Proiectarea concretă a reguletoarelor schemei este prezentată în /57/.

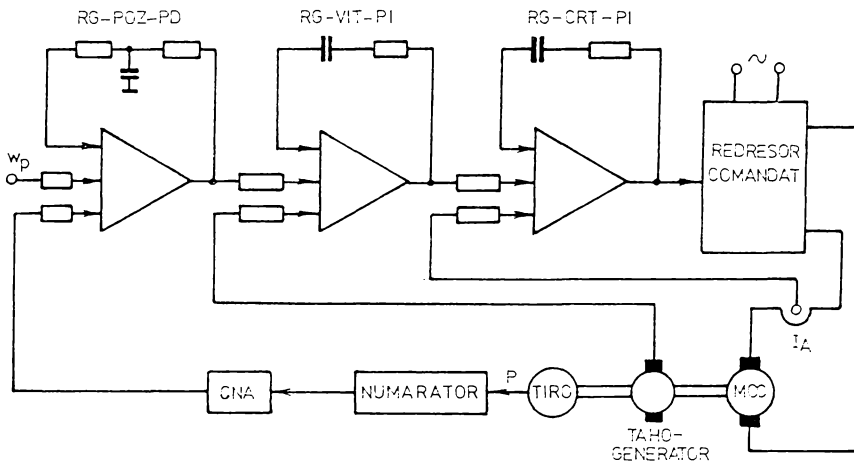


Figura 5.4. Schema de reglare a vitezei și poziției cu reguletoare analogice

### 5.1.2. Schemele bloc funcționale ale sistemelor de reglare automată proiectate

Cele două structuri concret implementate sînt: un sistem de reglare automată numerică a vitezei (SRAN-V) și un sistem de reglare automată numerică a vitezei și poziției (SRAN-VP).

Schemele funcționale ale celor două SRAN sînt prezentate în figurile 5.5. și 5.6.

În ambele scheme procesul (P) este o acționare cu motor de curent continuu (MCC), comandată în impulsuri de tensiune de către elementul de execuție (EE). EE este un chopper comandat

de către echipamentul de comandă numeric (ECN), construit în jurul unui microprocesor 8085, același cu cel descris în paragraful 3.2.2.

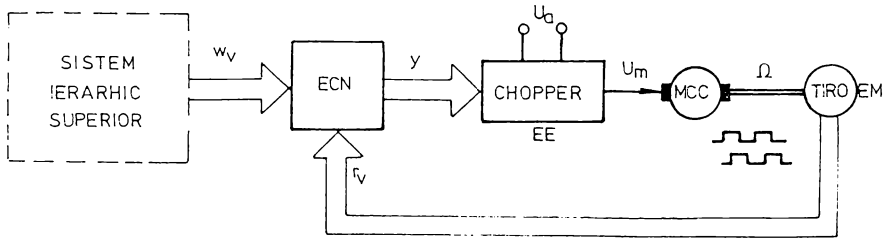


Figura 5.5. Structura sistemului de reglare automată numerică a vitezei (SRA-v) realizată

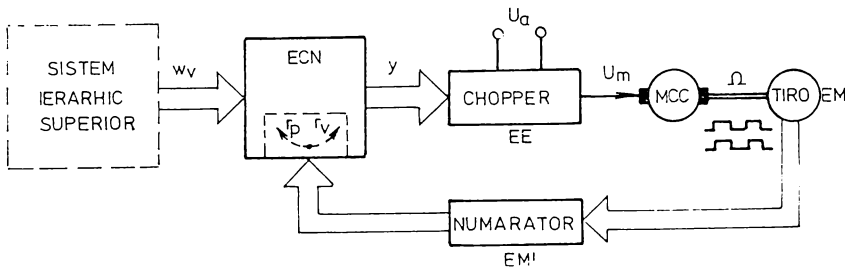


Figura 5.6. Structura sistemului de reglare automată numerică a vitezei și poziției (SRAN-vP) realizată

Elementul de măsurare a turației (EM) este un traductor incremental rotativ (TIRO). În cadrul SRAN-v, ECN asigură și funcția de măsurare a vitezei, prin contorizarea impulsurilor date de TIRO. În SRAN-vP, EM este completat cu un numărător (EM'), care furnizează la ieșirea sa, în orice moment, codul numeric al poziției curente. Această informație este utilizată de ECN, atât ca atare, cât și ca informație asupra vitezei curente, proporțională cu diferența dintre pozițiile curente din perioade de eșantionare succesive.

Mărimea prescrisă  $w$  este furnizată SRAN de către sistemul ierarhic superior, dacă acesta există, sau de către operatorul uman, prin consola ECN.

### 5.1.3. Procesul reglat

#### 5.1.3.1. Modelul matematic operațional al MCC

În capitolul 4 am arătat ecuațiile care descriu funcționarea unui motor de curent continuu. Ecuația (4.6) din paragraful 4.1., dacă se ține seama de faptul că motorul se alimentează în impulsuri de tensiune și deci intervine și inductivitatea înfășurării, devine:

$$u = u_e + R_i i_i + L_i \frac{di_i}{dt}, \quad (5.1)$$

cu

$$u_e = k \phi \Omega = K \Omega \quad (4.7) / 45/$$

Aceste ecuații sînt liniare. Aplicînd transformata Laplace (în condiții inițiale nule) se obține modelul matematic operațional:

$$U(s) = (R_i + sL_i)I_i(s) + K\Omega(s). \quad (5.2)$$

Ecuația de mișcare a motorului este:

$$M - M_g = J \frac{d\Omega}{dt}. \quad (4.5) / 45/$$

Dar

$$M_g = K i_i, / 45/ \quad (5.3)$$

Aplicînd și ecuației (4.5) transformata Laplace, se obține  $K I_i(s) = s J \Omega(s)$ , cu  $M_g = 0$  (5.4)

Din (5.2) și (5.4) se obține funcția de transfer a procesului:

$$H_p(s) = \frac{\Omega(s)}{U(s)} = \frac{1}{[(R_i + sL_i) \frac{sJ}{K^2} + 1]} \cdot K \quad (5.5)$$

În / 37, 45 / se notează:

$$T_m = \frac{JR_i}{K^2}, \quad (5.6)$$

- constanta de timp electromecanică a motorului și

$$T_e = \frac{L_i}{R_i}, \quad (5.7)$$

- constanta de timp electrică a motorului. Se obține astfel:

$$H_p(s) = \frac{\frac{1}{K}}{s^2 T_m T_e + s T_m + 1} \quad (5.8)$$

Pentru un timp uzual de motor de curent continuu  $T_m \gg T_e$  și se poate aproxima  $T_m \simeq T_m + T_e$ . Se obține, în final, funcția de transfer:

$$H_p(s) = \frac{\frac{1}{K}}{(1 + s T_m)(1 + s T_e)} \cdot \quad (5.9)$$

### 5.1.3.2. Modelul matematic al elementului de execuție

Elementul de execuție reprezentat în fig.5.2. și 5.3. este un variator de tensiune continuă de tipul celor descrise în paragraful 4.3.2.. Rolul său este de a alimenta motorul de curent continuu cu impulsuri de tensiune de amplitudine  $U_1$  și durată  $T_c$ , cu frecvența de repetiție  $T$ . Poate fi în variantă cu tiristoare (fig.4.14.) sau cu tranzistoare (fig.4.18). În cel de al doilea caz, acționarea este reversibilă.

Indiferent de structura de chopper utilizată, valoarea medie a tensiunii de alimentare pentru motor este:

$$U_m = \frac{T_c}{T} U_1 \cdot \quad (4.23)$$

Caracterizarea chopperului din punct de vedere dinamic, în condițiile în care el se comandă prin  $T_c$  variabil la  $T = \text{constant}$  se face prin funcția de transfer:

$$H_c(s) = \frac{U_m(s)}{T_c(s)} = \frac{U_1}{T} \cdot \quad (5.10)$$

### 5.1.3.3. Modelul matematic al traductorului TIRO și al numărătorului

În cadrul oricărui sistem automat este necesar să se dispună, la intrare, de valoarea mărimii reglate, ca mărime de reacție. În acest scop, se utilizează diferite tipuri de traductoare. Acestea sînt dispozitive ce au rolul de a stabili o corespondență între o mărime de măsurat și o mărime aptă de a fi utilizată de echipamente de prelucrare a datelor /88/.

Traductorul utilizat în cazul de față este un traductor incremental rotativ optic TIRO 1000, produs de Intreprinderea

Mecanică Fină București. Acest tip de traductor furnizează două trenuri de impulsuri de nivel TTL, de frecvență proporțională cu turația, decalate între ele cu un defazaj de  $\pm T/4$ , în funcție de sensul de rotație. La o turație completă, lungimea unui tren este de  $N = 1000$  impulsuri. Prin urmare, rezoluția în determinarea deplasării, cu un astfel de traductor, este de  $0,36^\circ$  sexagesimale. Există în fabricația românească și traductoare de tip TIRO 2000 cu rezoluție de  $0,18^\circ$  sexagesimale.

Principiul de funcționare al traductorului se explică pe baza schițelor și diagramelor din fig.5.7. /66/.

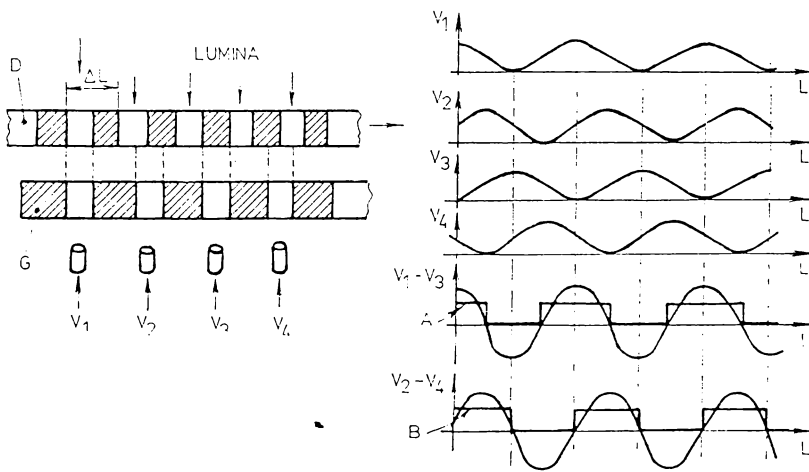


Figura 5.7. Principiul de funcționare al traductorului TIRO

Pe discul D, din material transparent, este depusă o rețea de linii echidistante, care formează un sistem de fante opace. Grila fixă G permite accesul la dispozitivele fotodioduri FC. Semnalele de la ieșirea acestora,  $V_1 \div V_4$ , variază aproximativ sinusoidal, pentru o mișcare uniformă a discului D.

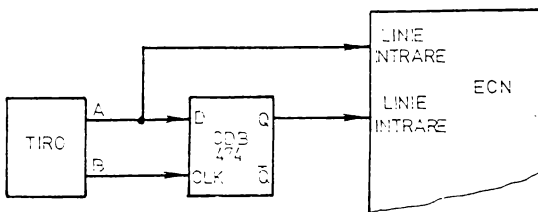


Figura 5.8. Determinarea sensului de rotație al TIRO cu bistabil D

Prin combinarea lor rezultă semnalele  $V_1 - V_3$  și  $V_2 - V_4$ , care sînt formate la niveluri TTL, prin cîte un comparator cu reacție pozitivă. Se obțin astfel cele două semnale de ieșire, A și B, defazate între ele

în funcție de sensul de rotație.

Traductorul este cuplat cu motorul cu un raport de transmisie 1:1. Pentru cazul acționării reversibile (cu chopper cu tranzistoare) este necesară decodificarea sensului de rotație. Aceasta se realizează simplu, cu ajutorul unui bistabil de tip D, conectat în conformitate cu fig.5.8. Ieșirea Q a bistabilului este pe 0 logic, respectiv 1 logic, în funcție de sensul de rotație.

Pentru determinarea sensului de rotație al TIRO, corect, fără erori, și în situația în care organul mobil al acționării, de care este legat traductorul, oscilează în jurul unei poziții fixe, se pot concepe și scheme mai complexe. În fig.5.9. se prezintă un exemplu în care se utilizează două circuite integrate: 7475 - patru bistabile D și 74153 - două multiplexoare. Semnalele de intrare pentru schemă sînt cele două semnale A și B de la TIRO și impulsul de tact al microsistemului. La ieșirile multiplexoarelor, care se leagă la câte o linie de port de intrare în sistem, se obțin impulsurile în concordanță cu sensul de rotație.

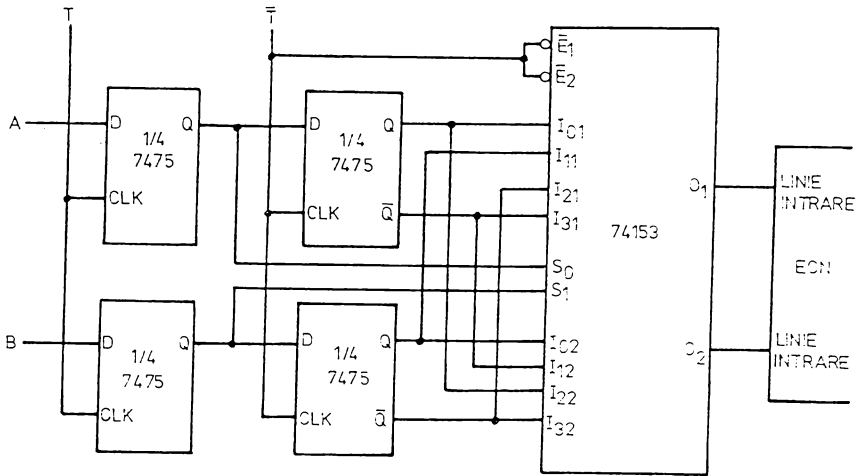


Figura 5.9. Schemă pentru determinarea sensului de rotație al TIRO

În cadrul SRAN-VP, unitatea centrală a ECN a fost eliberată de funcția de contorizare a impulsurilor venite de la traductor cu scopul de a i se asigura o rezervă suficientă de timp de execuție, pentru celelalte operații necesare în cadrul reglării. Funcția de măsurare permanentă a poziției este preluată de un numărător, care este incremetat sau decremetat cu câte o unitate

te, la fiecare deplasare unghiulară de  $0,36^\circ$ , într-un sens, respectiv celălalt, a axului traductorului, deplasare corespunzătoare unui impuls generat de TIRO.

Numărătorul este realizat cu 4 circuite CDB 4193, legate unul de celălalt, conform unei configurații clasice. Se obține codul numeric pe 16 biți al poziției curente. Ieșirile numărătoarelor utilizate sînt direct legate la cîte 4 linii a două porturi de intrare, de cîte 8 biți, ale ECN. Capacitatea unui astfel de numărător corespunde la circa 66 de turații ale arborelui motorului, legat cu TIRO.

Utilizînd mijloace numerice de implementare a reguletoarelor din sistem se subînțelege necesitatea alegerii unei perioade de eșantionare în proces.

Turația măsurată în cursul unei perioade de eșantionare este, de fapt, mărimea de reacție doar în pasul următor. Apare, deci, o întîrziere egală cu diferența dintre lungimea perioadei de eșantionare  $T$  și jumătatea intervalului de timp de măsurare  $T_M$ , dacă se acceptă că valoarea medie a vitezei este atînsă la mijlocul perioadei de măsură. Ca urmare, măsurarea este afectată de un timp mort  $U_M$ :

$$U_M = T - \frac{T_M}{2}. \quad (5.11)$$

Funcția de transfer a TIRO, considerînd viteza unghiulară ca mărime de intrare și numărul de impulsuri contorizate de ECN ca mărime de ieșire, este dată relația:

$$H_T(s) = \frac{N(s)}{Q(s)} = K_T e^{-s U_M}, \quad (5.12)$$

în care  $K_T$  (coeficientul de transfer) și  $U_M$  depind de modul concret de utilizare.

În cazul SRA-V, timpul de măsură  $T_M$  este egal cu jumătate din perioada de eșantionare  $T$ . În acest interval de timp sînt contorizate atît fronturile crescătoare, cît și cele descrescătoare ale impulsurilor sosite de la traductor. Rezultă, deci, o dublare a numărului contorizat față de numărul de impulsuri furnizate de TIRO. În consecință  $K_T$  are expresia:

$$K_T = 2T_M \cdot \frac{N}{2\pi} = \frac{NT}{2\pi} \quad (5.13)$$

iar funcția de transfer este:

$$H_T(s) = \frac{NT}{2T} e^{-s(T - \frac{T_M}{2})} = \frac{NT}{2T} e^{-\frac{3T}{4}s} \quad (5.14)$$

În cazul SRA-VP, măsurarea făcîndu-se pe toată durata perioadei de eşantionare, expresia funcţiei de transfer a traductorului este:

$$H_T(s) = \frac{NT}{2T} e^{-\frac{T}{2}s} \quad (5.15)$$

Numărătorul folosit are funcţia de transfer:

$$H_N(s) = \frac{N(s)}{\Omega(s)} = \frac{N}{2T} \quad (5.16)$$

#### 5.1.4. Echipamentul de comandă numerică (ECN)

ECN utilizat în SRA-V și SRA-VP este un microsistem realizat în jurul microprocesorului 8085. Acest sistem construit și dezvoltat în laboratorul de Electronică industrială al Facultății de Electronică din cadrul Universității Tehnice Timișoara, este descris în capitolul 3, paragraful 3.2.2..

### 5.2. Proiectarea algoritmilor de reglare numerică

#### 5.2.1. Proiectarea algoritmică a unui sistem convențional de reglare automată numerică /3 , 32/

Un sistem de reglare automată numerică (SRAN) convențional are structura din fig.5.10., /32, 96/. Pe figură, semnificațiile

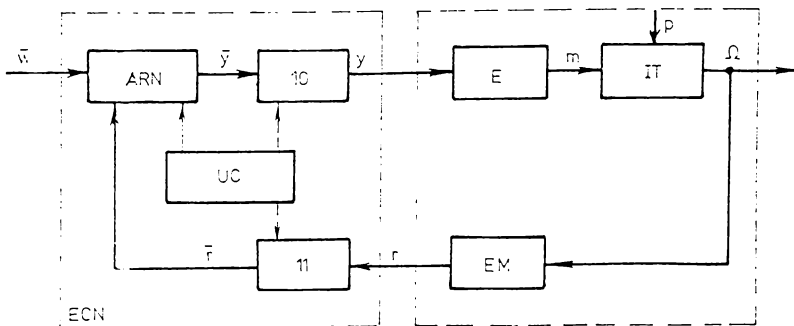


Figura 5.10. Structura unui SRAN convențional pentru ECN, E, EM, y, Ω, r au fost prezentate în paragraful 5.1.



iar celelalte notații sînt:

- ARN - algoritm de reglare numerică,
- II - interfață intrare,
- IO - interfață ieșire,
- UC - unitate de comandă,
- IT - instalația tehnologică,
- m - mărime de execuție,
- p - perturbație

$\bar{w}$ ,  $\bar{y}$ ,  $\bar{r}$  - codurile numerice corespunzătoare lui  $w$ ,  $y$ ,  $r$ .

Interfața de intrare (II) conține, ca element central, un convertor analog-numeric (CAN). Din punct de vedere informațional, II se reduce la eșantionorul EES (circuitul de eșantionare - memorare), înzestrat totodată cu capacitatea de conversie analog-numerică.

Elementul constitutiv principal al IO este convertorul numeric-analogic (CNA), care se reduce la elementul de tastare ETS, înzestrat și cu proprietatea de conversie N/A, înseriat cu elementul de reținere ER, destinat refacerii semnalului de comandă continuu. În general, eșantionorul și elementul de tastare lucrează sincron, cu aceeași perioadă, numită perioadă de eșantionare (T).

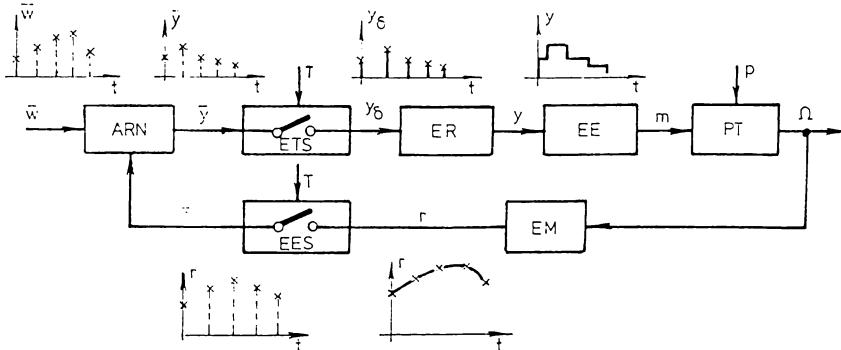


Figura 5.11. Schema bloc a SRAN convențional

Unitatea de comandă organizează și sincronizează în timp diferitele acțiuni care au loc în sistem. Teoretic, se presupune o sincronizare totală și o operare instantanee. Aceasta înseamnă că: EES și ETS lucrează sincron, conversiile A/N și N/A se execută instantaneu, ARN se efectuează instantaneu. În realitate, aceste ipoteze nu sînt valabile și conduc la abateri mai mult sau

mai puțin importante față de performanțele calculate.

ER reprezintă un extrapolator de ordinul zero, care pe intervalul de o perioadă de eșantionare își menține constantă mărimea de ieșire, la o valoare egală cu cea aplicată la intrare, la începutul perioadei respective.

Rezultă, prin urmare, schema bloc din fig.5.11. În figură se sugerează și modul de variație în timp a mărimilor ECN: cele reprezentate cu linie continuă au semnificația de mărimi analogice, iar cele reprezentate cu linie întreruptă corespund codurilor numerice menționate.

ER se asimilează cu un element, având funcția de transfer:

$$H_{ER}(s) = \frac{1 - e^{-sT}}{s} . \quad (5.17)$$

În SRAN, rolul regulatorului îl joacă ARN, ECN elaborând pe baza lui, în funcție de  $w$  și  $r$ , codul numeric al mărimii de comandă  $y$ . Codurile cu care operează ARN corespund, datorită eșantionării, unor momente discrete:

$$t_K = mT , \quad m = \text{întreg} , \quad (5.18)$$

și a unui anumit mod de cuantificare a mărimilor continue de același nume. În acest context, vom considera mărimile discrete:

$$y_K = \bar{y}|_{t=KT}; \quad w_K = \bar{w}|_{t=KT}; \quad r_K = \bar{r}|_{t=KT} , \quad (5.19)$$

respectiv ARN de forma:

$$y_K = f(w_K, r_K, y_{K-1}) . \quad (5.20)$$

Principial, problema proiectării algoritmice a unui SRAN convențional este similară cu problematica proiectării algoritmice a unui SRA continuu convențional /32/.

În funcție de dinamica procesului condus, procedeele de proiectare algoritmică a ARN se împart în două categorii:

- procedee bazate pe obținerea ARN prin discretizarea legilor de reglare continuă,
- procedee bazate pe proiectarea directă a ARN.

Prima categorie de procedee presupune o perioadă de eșantionare  $T$  de valoare redusă, în comparație cu dinamica impusă SRAN și cu dinamica procesului condus. În această situație, ARN aproximează legea de reglare de așa manieră încît comportarea SRAN diferă practic foarte puțin de comportarea unui SRA, care ar lucra cu un regulator ce implementează legea de reglare continuă de la care s-a plecat. ARN obținut astfel poartă denumirea

de algoritm de reglare numerică cvasicontinuă (ARNC).

Ipooteza că SRAN reproduce comportarea sistemului continuu este esențială în proiectarea ARNC. Acest lucru trebuie să se realizeze la nivelul tuturor elementelor componente. De aceea, se impune ca, pentru  $T \rightarrow 0$ , atât ansamblul format din ETS și ER, cât și EES, să aibă funcția de transfer  $H_{EER}(s) = 1$ . În acest scop, grupul ETS + ER se echivalează cu un sigur element de transfer, EER, numit element de tastare și reținere, având funcția de transfer:

$$H_{EER}(s) = \frac{1 - e^{-sT}}{sT} \quad (5.21)$$

Atunci când  $T$  are ordinul de mărime al constantelor de timp mici, în calculele de proiectare  $H_{EER}(s)$  se poate aproxima prin:

$$H_{EER}(s) = e^{-\frac{sT}{2}} \quad (5.22)$$

Proiectarea ARNC decurge conform organigramei din fig.

5.12. /32/.

Corespunzător acesteia, pe baza caracteristicilor procesului condus și a performanțelor impuse, se adoptă  $T$  și se apreciază dacă EER poate fi tratat ca un element de transfer neinerțial. În caz afirmativ, regulatorul se proiectează considerînd  $H_{EER}(s) = 1$ , adică omițînd prezența EER. În caz contrar, se adoptă pentru  $H_{EER}(s)$  expresia (5.21) sau o expresie de aproximare, de exemplu (5.22). Se proiectează apoi regulatorul continuu, în ipoteza că, în raport cu mărimea de comandă, procesul condus prezintă funcția de transfer:

$$H_p(s) = H_{EER}(s) \cdot H_R(s) \quad (5.23)$$

Pentru proiectarea funcției de transfer a regulatorului continuu,  $H_R(s)$ , se pot utiliza diferite procedee, cum sînt: metoda caracteristicilor de frecvență ale sistemului deschis; metoda alocării polilor și zerourilor funcției de transfer; metoda modulului - varianta Kessler /16/ ș.a.

În continuare, pe baza funcției de transfer  $H_R(s)$  calculate, se determină ARNC, folosind o metodă de discretizare a algoritmilor de reglare continuă. SRAN, astfel proiectat este supus, mai departe, validării. În funcție de rezultatul ultimei operații, se consideră ca soluție posibilă ARNC obținut sau se reia

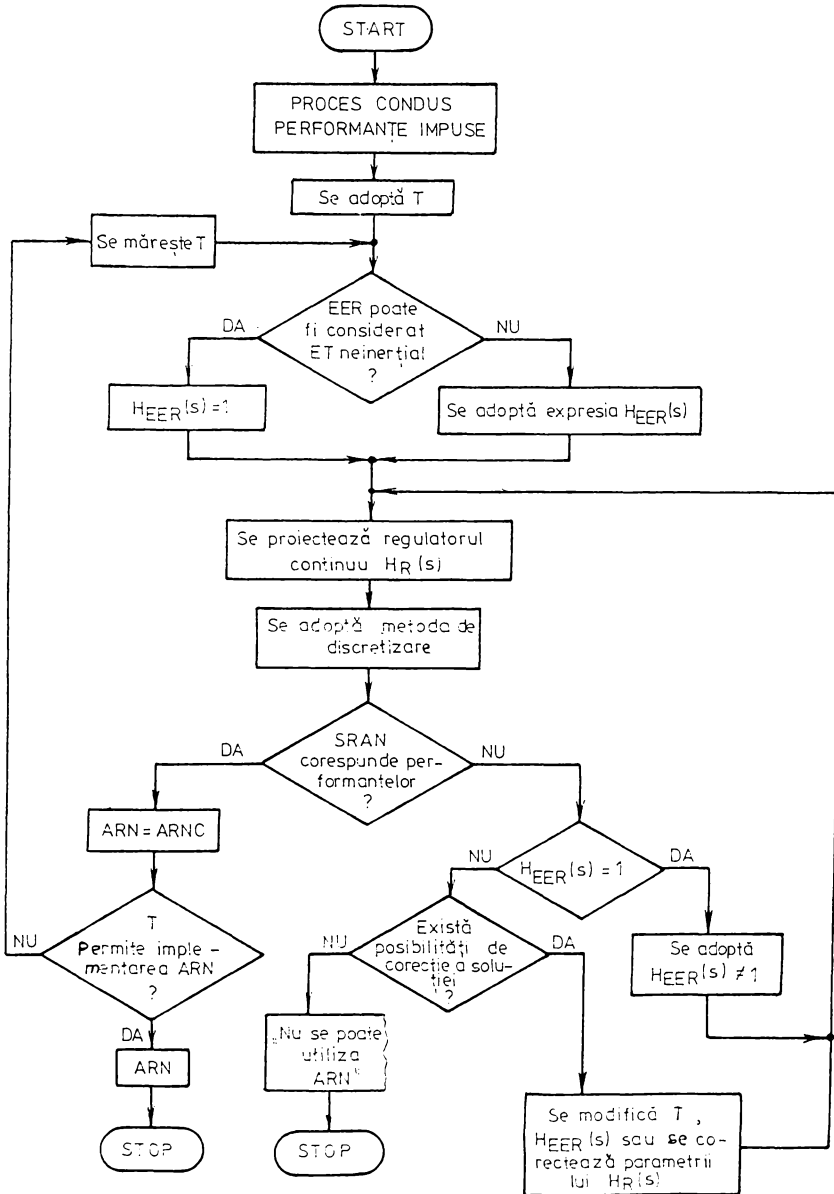


Figura 5.12. Procedura de proiectare a ARNC

proiectarea. In final, soluția posibilă este definitiv validată, numai dacă ARN poate fi implementat pe ECM, în intervalul de timp  $T$ .

Adoptarea perioadei de eşantionare reprezintă o operație extrem de importantă. Practic,  $T$  trebuie să satisfacă o condiție de forma:

$$T \leq 0,1 \sum_j T_j, \quad (5.24)$$

în care  $T_j$  reprezintă constantele de timp semnificative (dominante) ale procesului condus.

### 5.2.2. Discretizarea modelelor matematice ale elementelor de transfer continue

Dintre procedeele de discretizare prezentate în /32, 16/, am utilizat metoda trapezului.

Prin această metodă se asociază unui element de transfer continuu un model matematic discret, aplicînd ecuației acestuia o transformare liniară. Procedeeul constă în următoarele:

Se integrează de  $n$  ori ecuația elementului de transfer pe intervalul  $[t - T, t]$ , folosind următoarea relație de aproximare:

$$\int_{t-T}^t x(\tau) d\tau = \frac{T}{2} [x(t) + x(t-T)]. \quad (5.25)$$

Se particularizează în rezultatul obținut  $t = t_K$  și se scrie ecuația discretă, înlocuind  $x(t_i)$  cu  $x_i$ .

Fie o aplicație a metodei trapezului pentru cazul unui element de transfer rațional, de ordinul I, cu funcția de transfer:

$$H(s) = \frac{\beta_0 + \beta_1 s}{\alpha_0 + \alpha_1 s}, \quad (5.26)$$

căreia îi corespunde ecuația diferențială:

$$\alpha_0 y(t) + \alpha_1 \dot{y}(t) = \beta_0 a(t) + \beta_1 \dot{a}(t) \quad (5.27)$$

Dacă se integrează ecuația (5.27), utilizînd metoda trapezului de discretizare se obține:

$$\begin{aligned} & \alpha_0 \frac{T}{2} [y(t_K) + y(t_{K-1})] + \alpha_1 [y(t_K) - y(t_{K-1})] = \\ & = \beta_0 \frac{T}{2} [a(t_K) + a(t_{K-1})] + \beta_1 [a(t_K) - a(t_{K-1})] \end{aligned} \quad (5.28)$$

În acest fel, pentru funcția de transfer (5.26), ARN corespunzător obținut prin discretizare, este:

$$y_K = d_0 a_K + d_1 a_{K-1} + c_1 y_{K-1} = d_0 (w_K - r_K) + d_1 (w_{K-1} - r_{K-1}) + c_1 y_{K-1}, \quad (5.29)$$

cu

$$d_0 = \frac{\beta_0 \frac{T}{2} + \beta_1}{\alpha_0 \frac{T}{2} + \alpha_1}; \quad d_1 = \frac{\beta_0 \frac{T}{2} - \beta_1}{\alpha_0 \frac{T}{2} + \alpha_1}; \quad c_1 = \frac{\alpha_1 - \alpha_0 \frac{T}{2}}{\alpha_0 \frac{T}{2} + \alpha_1} \quad (5.30)$$

unde:

- a este mărimea de acționare (eroarea; abaterea),
- y este mărimea de comandă.

### 5.2.3. Proiectarea algoritmică a SRA în cascadă /3 /

SRA în cascadă asigură performanțe superioare SRA convenționale, prin compensarea mai bună a perturbațiilor și reducerea inerției pe care o prezintă procesul, în raport cu mărimea de comandă, deci micșorarea timpului de reglare.

SRA-VP din fig.5.6. se încadrează în acest tip de structură /32/. Bucla de viteză este o buclă de reglare internă (auxiliară), iar bucla de poziție este o buclă de reglare (principală).

Proiectarea algoritmică a unui SRA în cascadă cuprinde următoarele etape:

1. Analizarea schemei bloc a procesului condus și descompunerea acesteia folosind algebra schemelor bloc și respectînd principiul cauzalității, într-o conexiune serie de subsisteme, care au ca mărimi de legătură tocmai mărimile auxiliare.
2. Plecînd de la ansamblul performanțelor impuse, se formulează condiții de proiectare pentru fiecare din buclele de reglare.
3. Se proiectează succesiv buclele de reglare, începînd cu bucla internă. De fiecare dată, după proiectarea regulatorului unei bucle de reglare auxiliare, se procedează la "reducerea" schemei informaționale a SRA, prin echivalarea buclei calculate printr-un element de transfer, ce redă comportarea buclei în raport cu mărimea ei de conducere. Schema de reglare rezultată în urma reducerii va avea în interior o nouă buclă de reglare convențională, care, după proiectarea regulatorului aferent, poate fi din nou redusă ș.a.m.d.

### 5.2.4. Proiectarea concretă a SRA-V /3 , 90, 36/

SRA-V are schema bloc din fig.5.13., în care blocurile au funcțiile de transfer stabilite în paragraful 5.1.

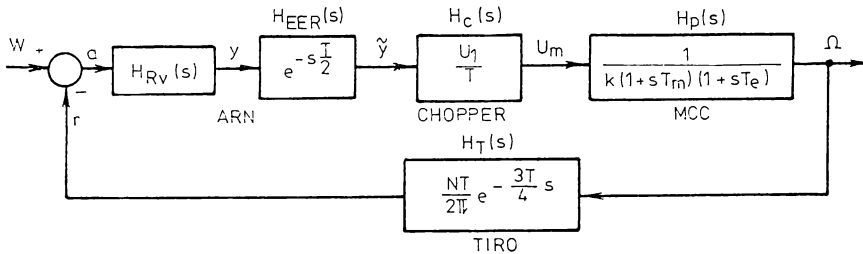


Figura 5.13. Schema bloc a SRA-V cu f.d.t. ale elementelor componente reprezentate

Pentru stabilirea valorii perioadei de eşantionare  $T$ , se porneşte de la determinarea constantelor  $T_m$  şi  $T_e$  din  $H_p(s)$  (modelul matematic al motorului de curent continuu), MCC folosit este de tip EP 211 (IME Piteşti), avînd excitaţia cu magnet permanent şi următoarele date de catalog:

$$U_n = 24 \text{ V}, I_n = 3,5 \text{ A}, M_n = 0,417 \text{ kgf.cm},$$

$$n_n = \text{I:}1620 \text{ rpm } \pm 12\% \text{ (tehnologic),}$$

$$\text{II:}2160 \text{ rpm } \pm 12\% \text{ (tehnologic),}$$

$$M_{\text{pornire}} = 8,33 \text{ kgf.cm},$$

$$R_i = 1,8 \Omega, L_i = 8,5 \text{ mH}.$$

Momentul de inerţie al motorului este:

$$J_m = \frac{mD^2}{8} = \frac{9 \frac{D^2}{4} \cdot 1 \cdot D^2}{8}, \quad (5.31)$$

$$J_m \approx 8,5 \cdot 10^{-4} \text{ kg.m}^2$$

Constanta motorului este:

$$K = \frac{U_n - R_i I_n}{2\pi n_n} \cdot 60 \approx 0,1 \frac{\text{V}}{\text{rad/s}} \quad (5.32)$$

Constantele de timp au următoarele valori:

$$T_e = \frac{L_i}{R_i} \approx 5 \text{ ms} \quad (5.33)$$

$$T_m = \frac{J_m R_i}{K^2} \approx 150 \text{ ms} \quad (5.34)$$

Cu aceste valori funcţia de transfer  $H_p(s)$  a motorului utilizat este:

$$H_p(s) = \frac{1}{K(1+sT_m)(1+sT_e)} = \frac{10}{(1+0,15s)(1+0,005s)} \quad (5.35)$$

Din calcule rezultă că valoarea cea mai semnificativă o are constanta mecanică  $T_m$  a motorului (0,15 s). Ca urmare, perioada de eșantionare trebuie să satisfacă condiția:

$$T \leq 0,1 T_m = 0,015 \text{ s}$$

Se adoptă  $T = 0,01 \text{ s} = 10 \text{ ms}$  (5.36)

Timpul de măsură  $T_M$ , se adoptă  $\frac{T}{2} = 5 \text{ ms}$ . (5.37)

La baza metodei caracteristicilor de frecvență ale sistemului deschis stau următoarele rezultate ale unor studii cu privire la legătura dintre aceste caracteristici de frecvență și calitatea SRA /32/:

- a) SRA este asimptotic stabil, dacă rezerva de față  $\varphi_M \in (50^\circ, 70^\circ)$ . În acest caz, amortizarea este bună și procesele tranzitorii se caracterizează prin oscilații relativ reduse, cu suprareglaje mici, corespunzând cerințelor din practică;
- b) În primă aproximație, un SRA se prezintă în raport cu mărimea de intrare ca un sistem de ordinul I, cu constanta de timp  $T_t = 1/\omega_t$ , timpul de reglare fiind aproximabil prin relațiile:
 
$$t_r \approx 0,05 \approx 4 T_t = 4/\omega_t, \quad (5.38)$$

$$t_r \approx 0,02 \approx 5 T_t = 5/\omega_t. \quad (5.39)$$
- c) Dacă sistemul deschis are un caracter integrator, atunci în regim staționar SRA satisface, în general, condiția de anulare asimptotică a ieșirilor de apreciere /16, 3 /.

În esență, proiectarea SRA prin această metodă constă în determinarea unui regulator, care să conducă la obținerea unor caracteristici de frecvență ale sistemului deschis care să satisfacă performanțele impuse, ținând cont de rezultatele menționate mai sus.

Pentru a asigura o eroare nulă în regim staționar, se adoptă o reglare de tip PI, corespunzătoare unui regulator cvasicontinuu cu funcția de transfer:

$$H_{RV}(s) = K_V \left(1 + \frac{1}{sT_i}\right) \quad (5.40)$$

Constanta de timp a regulatorului,  $T_i$ , se alege egală cu cea mai mare constantă de timp a procesului:

$$T_i = T_m = 0,15 \text{ s}. \quad (5.41)$$

Funcția de transfer a sistemului deschis (fig.5.13.) este:



$$H_L(s) = H_{RV}(s) \cdot H_{EER}(s) \cdot H_C(s) \cdot H_p(s) \cdot H_T(s) \quad (5.42)$$

În relația (5.42), fiecare model matematic a fost determinat în paragraful 5.1. Detaliind se obține:

$$H_L(s) = K_V \frac{1 + sT_i}{sT_i} \cdot e^{-\frac{T}{2}s} \cdot \frac{U_1}{T} \cdot \frac{1}{K(1+sT_m)(1+sT_e)} \cdot \frac{N}{2T} \cdot e^{-\frac{3T}{4}s} \quad (5.43)$$

Înlocuind numeric, pentru aplicația concretă, se obține:

$$H_L(s) = K_V \frac{254648}{s(1 + 0,005s)} \cdot e^{-0,0125s} \quad (5.44)$$

Coefficientul de reglare al regulatorului,  $K_V$ , se determină grafoanalitic. Acesta constă în reprezentarea caracteristicilor de frecvență ale sistemului deschis,  $H_L$  și  $\varphi_L$ , pentru o valoare arbitrară  $K_{V_0}$  a coeficientului de amplificare și determinarea lui  $K_V$  cu relația:

$$K_V = K_{V_0} \cdot 10^{-0,05 |H_L|^{\#}} \text{ dB} \quad (5.45)$$

în care  $|H_L|^{\#}$  dB este ordonata caracteristicii amplitudine-pulsatie, la  $\omega = \omega_t^{\#}$  (fig.5.14.), corespunzătoare unei rezerve de fază de  $60^\circ$ .

Reprezentarea caracteristicilor de frecvență s-a făcut alegînd:

$$K_{V_0} = 1/254648 = 3,93 \cdot 10^{-6}, \quad (5.46)$$

obținîndu-se pentru  $K_V$  valoarea:

$$K_V = K_{V_0} \cdot 10^{-0,05 |H_L|^{\#}} = \omega_t^{\#} \cdot K_{V_0} = 30,0 \cdot 3,93 \cdot 10^{-6} = 1,18 \cdot 10^{-4} \text{ s} \quad (5.47)$$

ARNC se determină prin discretizarea, cu ajutorul metodei trapezului, a legii de reglare de tip PI (5.40), identificată cu ecuația (5.26), în care:

$$\begin{aligned} \beta_0 &= K_V = 1,18 \cdot 10^{-4} \text{ s;} \\ \beta_1 &= K_V T_i = 1,77 \cdot 10^{-5} \text{ s;} \\ \alpha_0 &= 0; \alpha_1 = T_i = 0,15 \text{ s} \end{aligned}$$

Folosind formulele de calcul (5.30), se obțin pentru coeficienți, valorile:

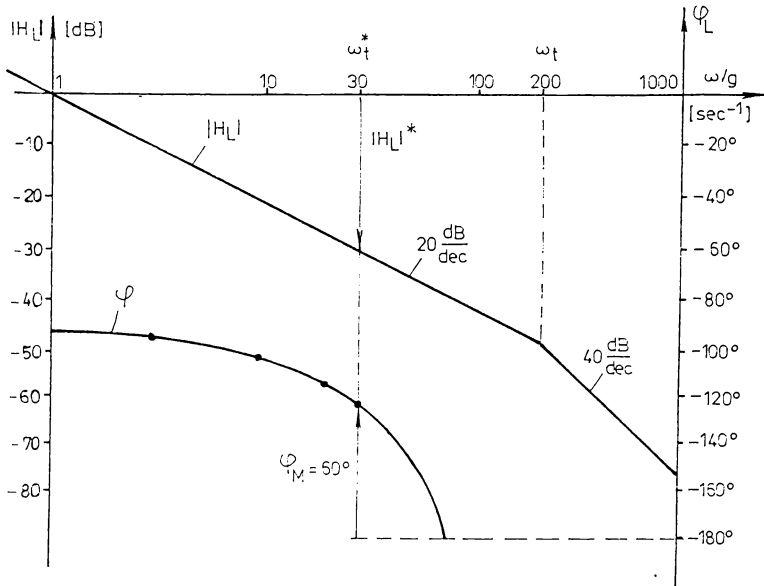


Figura 5.14. Caracteristica de fază și frecvență ale sistemului deschis, pentru o valoare arbitrară a lui  $K_{V0}$

$$\begin{aligned}
 d_0 &= 1,22 \cdot 10^{-4} \text{ s;} \\
 d_1 &= -1,14 \cdot 10^{-4} \text{ s;}
 \end{aligned}
 \tag{5.48}$$

$$c_1 = 1.$$

ARNC este deci de forma :

$$Y_K = d_0 a_K + d_1 a_K + Y_{K-1} \tag{5.49}$$

### 5.2.5. Proiectarea SRA-VP

SRA-VP are schema bloc din fig.5.15, în care blocurile au funcțiile de transfer stabilite în paragraful 5.1.3.

După cum am arătat în paragraful 5.2.3, proiectarea SRA în cascadă decurge similar cu proiectarea SRA convențională.

Ca urmare, se pot utiliza coeficienții calculați în paragraful 5.2.4, pentru bucla de viteză, cu observația că, datorită micșorării timpului mort total al acestei bucle, de la 12,5 ms la 10 msec, se obține o margine de fază mai mare :

$$\varphi_{MV} = 64,3^\circ \quad (5.50)$$

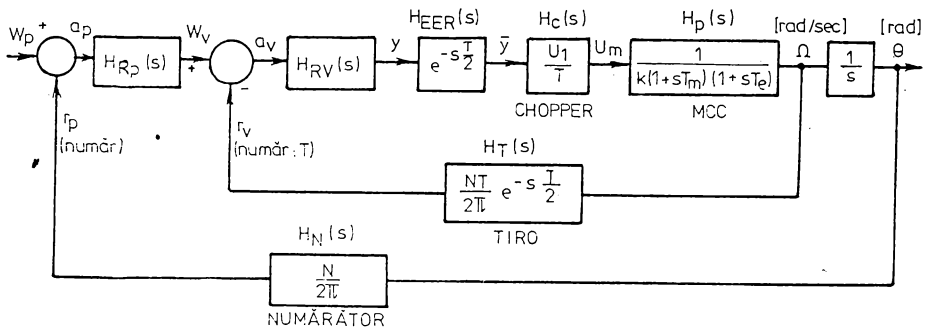


Figura 5.15. Structura bloc a SRA-VP cu f.d.t. ale elementelor detaliate

Acest lucru este de dorit, pentru că prin această creștere a marginii de fază, crește și amortizarea SRA /36/.

Bucula internă de viteză are funcția de transfer:

$$H_V(s) = \frac{H_{RV}(s) \cdot H_{EER}(s) \cdot H_C(s) \cdot H_P(s)}{1 + H_{RV}(s) \cdot H_{EER}(s) \cdot H_C(s) \cdot H_P(s) \cdot H_T(s)} \quad (5.51)$$

Înlocuind valorile numerice calculate în (5.24) rezultă:

$$H_V(s) = \frac{3776 e^{-0,005s}}{s^2 + 200s + 600} \quad (5.52)$$

Pentru bucla de poziție se adoptă o lege de reglare de tip proporțional (P), deci  $H_{RP}(s) = K_P$ . Nu se utilizează un regulator PI, datorită caracterului integrator al dependenței dintre viteză și poziție.

Funcția de transfer a sistemului deschis este:

$$H_L(s) = H_{RP}(s) \cdot H_V(s) \cdot \frac{H_N(s)}{s} \quad (5.53)$$

Numeric, concret:

$$H_L(s) = \frac{60000 \cdot K_P \cdot e^{-0,005s}}{s(s^2 + 200s + 600 \cdot e^{-0,01s})} \quad (5.54)$$

Deoarece timpii morți apar doar în interiorul buclei de viteză, care asigură o reacție negativă, ei pot fi neglijați inițial, urmînd să se evalueze ulterior consecințele acestor neglijări.

Pulsațiile de ~~fr~~ ~~în~~ ~~care~~ ale funcției de transfer (5.54) sînt:

$$\omega_{1,2} = -s_{1,2} = 100 \pm \sqrt{10000 - 6010}$$

$$\omega_1 = 36,8 \text{ s}^{-1}, \quad \omega_2 = 163,2 \text{ s}^{-1}$$

Caracteristicile de frecvență ale sistemului deschis sînt reprezentate în fig.5.16; s-a ales pentru  $K_P$  valoarea inițială  $K_{P0} = 0,01$ .

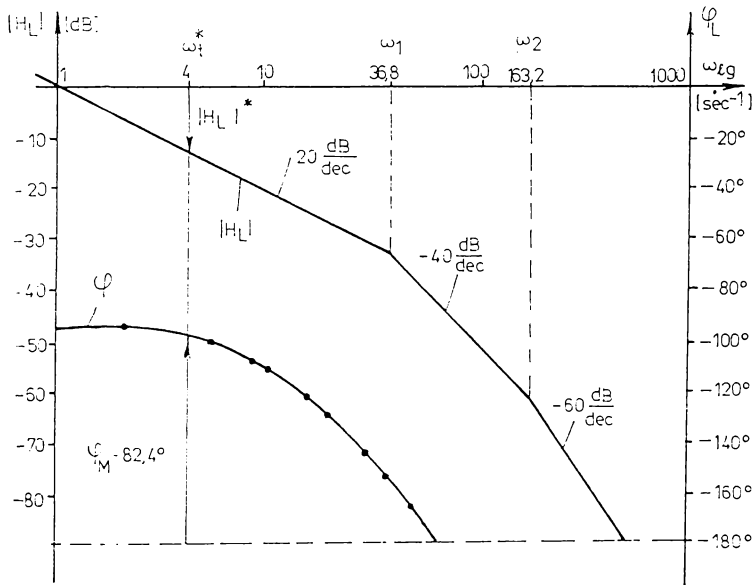


Figura 5.16. Caracteristicile de frecvență ale sistemului deschis pentru SRA-P

Pentru a asigura eliminarea oscilațiilor din jurul poziției de echilibru, răspunsul SRA trebuie să fie aperiodic. Acest tip de răspuns se obține pentru o margine de fază  $\varphi_M > 80^\circ / 32,16/$ .

Din fig.5.16 se observă că pentru  $\omega_t^* = 4 \text{ rad/S}$ ,  $\varphi_{M1} = 82,4^\circ$ . La această pulsație rezultă :

$$K_P = K_{P0} \cdot 10^{-0,05 \cdot |\bar{H}_L|^*} = \omega_t^* \cdot K_{P0} = 0,04 \quad (5.55)$$

Trebuie efectuat calculul exact al marginii de fază, pentru  $\omega_t^* = 4 \text{ s}^{-1}$  considerînd și efectul timpilor morți. Cu  $s = j\omega$  se obține :

$$H_L(4) = \frac{24040(1-j/50)}{4j[-16 + 800 + 6010(0,999 - j/25)]} = \frac{6010(1-j/50)}{j(5989 + 560j)} \quad (5.56)$$

$$\varphi_L(4) = \arg H_L(4) = \frac{180}{\pi} \left( -\frac{\pi}{2} - \frac{1}{50} - \frac{560}{5989} \right) = -96,5^\circ \quad (5.57)$$

Rezultă o margine de fază:

$$\varphi_M = 180^\circ - 96,5^\circ = 83,5^\circ > 80^\circ \quad (5.58)$$

Deci, răspunsul SRA se păstrează aperiodic.

In cazul buclei interne de viteză, ARNC este identic cu cel stabilit pentru SRA-V:

$$y_K = y_{K-1} + d_0(w_{VK} - r_{VK}) + d_1(w_{VK-1} - r_{VK-1}) \quad (5.59)$$

Pentru regulatorul de poziție se poate scrie:

$$w_{VK} = K_P(w_{PK} - r_{PK}) \quad (5.60)$$

Informația de viteză se extrage din cea de poziție:

$$r_{VK} = r_{PK} - r_{PK-1} \quad (5.61)$$

Rezultă:

$$y_K = y_{K-1} + d_0[K_P(w_{PK} - r_{PK}) - (r_{PK} - r_{PK-1})] + d_1[K_P(w_{PK-1} - r_{PK-1}) - (r_{PK-1} - r_{PK-2})]$$

$$y_K = y_{K-1} + d_0 K_P w_{PK} + d_1 K_P w_{PK-1} - d_0(1 + K_P)r_{PK} + [d_0 - d_1(1 + K_P)]r_{PK-1} + d_1 r_{PK-2}$$

Se obține astfel un ARN de ordinul doi de forma:

$$y_K = y_{K-1} + D_0 w_{PK} + D_1 w_{PK-1} + D_2 r_{PK} + D_3 r_{PK-1} + D_4 r_{PK-2} \quad (5.62)$$

In acest ARN, coeficienții au valorile:

$$D_0 = d_0 K_P = 4,88 \cdot 10^{-6} \text{ s}$$

$$D_1 = d_1 K_P = -4,56 \cdot 10^{-6} \text{ s}$$

$$D_2 = -d_0(1 + K_P) = -1,27 \cdot 10^{-4} \text{ s} \quad (5.63)$$

$$D_3 = d_0 - d_1(1 + K_P) = 2,41 \cdot 10^{-4} \text{ s}$$

$$D_4 = d_1 = -1,14 \cdot 10^{-4} \text{ s}$$

### 5.3. Implementarea algoritmului de reglare numerică a vitezei /36/

#### 5.3.1. Forma discretă finală pentru ARN-V

ARN-V a rezultat sub forma ecuației discrete (5.49):

$$y_K = d_0 a_K + d_1 a_{K-1} + y_{K-1},$$

cu coeficienții de valori calculate în (5.48).

În relația (5.49)  $y_K$  și  $y_{K-1}$ , mărimea de comandă curentă și cea precedentă, sînt exprimate în secunde și reprezintă intervale de conducție  $T_c$  pentru chopper. Implementarea corectă a algoritmului (5.49), pe sistemul SDK-85 utilizat, implică exprimarea mărimilor  $y_K$  și  $y_{K-1}$  în cod numeric normat, adică în numere de 14 biți (capacitatea timerului din sistem). Numerele de 14 biți se decrementează pînă la zero, obținîndu-se în acest fel intervalele de timp necesare în comanda chopperului. Pasul de decrementare a numărătorului este egal cu durata impulsului de tact din sistem (CLK), 325,52 ns. Rezultă pentru coeficienții ARN-V valorile:

$$d_0 = \frac{1,22 \cdot 10^{-4}}{325,52 \cdot 10^{-9}} = 375 \quad (5.64)$$

$$d_1 = -350 \quad (5.65)$$

Forma finală sub care ARN-V poate fi implementat pe micro-sistem este:

$$y_K = y_{K-1} + 375(w_K - r_K) - 350(w_{K-1} - r_{K-1}) \quad (5.66)$$

### 5.3.2. Sarcinile ECN. Structura programului ARN-V

ECN este realizat cu microsistemul SDK-85, construit în jurul microprocesorului 8085, prezentat în paragraful 3.2.2. Pentru realizarea SRAN-V microsistemul trebuie să îndeplinească următoarele funcții:

(F1) - preluarea mărimii prescrise,  $w$ , de la operator, prin consola DAF, utilizînd nivelul de întrerupere RST 6.5;

(F2) - măsurarea vitezei curente  $r_K$ , de la TIRO, utilizînd o subrutină de frecvențmetru;

(F3) - calculul mărimii de comandă  $y_K$ , conform ecuației (5.66) pentru ARN-V;

(F4) - conversia mărimii numerice  $y_K$  într-un interval de timp real  $T_{CK}$ , timp de conducție pentru chopper, utilizînd unul din timerele sistemului;

(F5) - comanda propriu-zisă a chopperului.

Cele cinci funcții trebuie îndeplinite de către ECN pe parcursul unei perioade de eșantionare  $T$ . În funcție de mărimea

duratei  $T_{CK}$  în raport cu  $T$  și cu valoarea minimă a intervalului de conducție  $T_{C,min}$  permisă de chopper (0 s pentru chopperul cu tranzistoare), sînt posibile trei regimuri de gestionare a perioadei de eșantionare, reprezentate în fig.5.17.

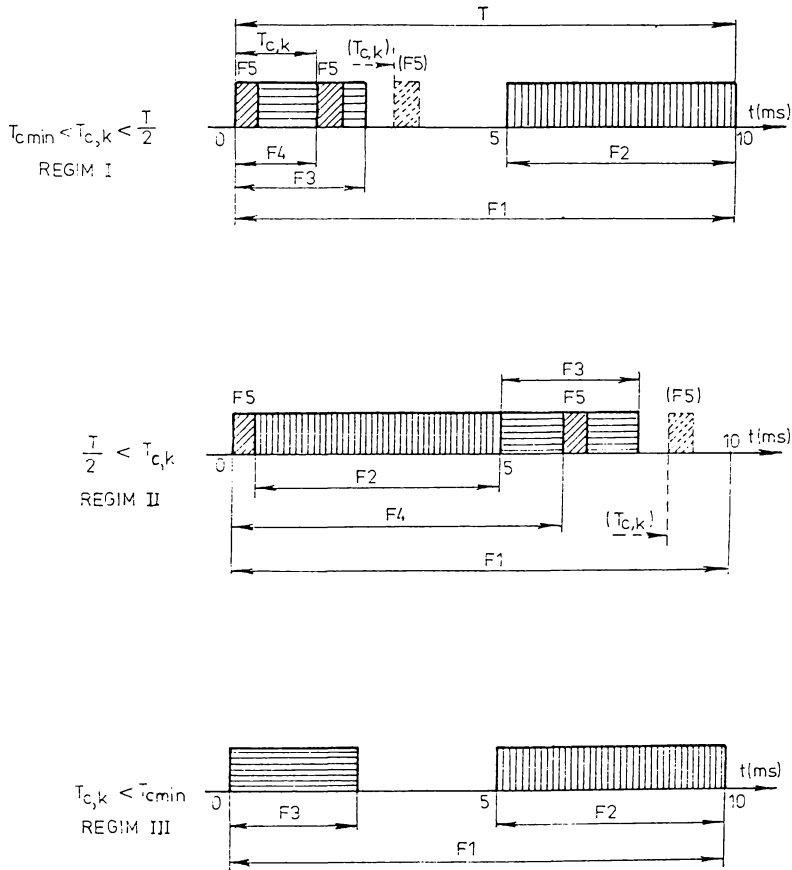


Figura 5.17. Momentele de execuție a celor cinci funcții de către sistemul de comandă pe parcursul perioadei de eșantionare  $T$ .

În fig.5.17. sînt reprezentate intervalele de timp în care se efectuează de către microsistem operațiile corespunzătoare celor cinci funcții care trebuie să le îndeplinească. Funcția  $F_4$ , contorizarea intervalului de conducție pentru chopper,  $T_{C,K}$ , se efectuează cu unul din timerele sistemului și, deci, microprocesorul este, pe această durată, liber spre a executa alte operații.

Intervalul  $T_{C,K}$  poate fi mai mic decît  $T/2$  - regimul I sau mai mare decît  $T/2$  - regimul II. Intervalul de timp în care se efectuează măsurarea vitezei curente a motorului de către microprocesor (F2) este egal cu  $T/2$ . De aceea, în regimul I, măsurarea se face după ce comanda chopperului s-a efectuat, iar în regimul II, măsurarea se efectuează în timpul contorizării lui  $T_{C,K}$ . Calculul mărimii de comandă se efectuează în timpul liber disponibil pe durata  $T$  și este întrerupt cînd trebuie generate comenzile propriu-zise pentru chopper. Mărimea prescrisă,  $w$ , poate fi preluată din exterior în orice moment prin întreruperea, pe nivelul RST 6.5, a oricărei operații aflate în curs de desfășurare cu excepția (F2). În regimul III, chopperul nu mai este comandat, adică motorul rămîne nealimentat.

Schema logică-bloc a programului este prezentată în fig.5.18.

Programele detaliate, atît pentru varianta cu chopper cu tiristoare, cît și pentru varianta cu chopper cu tranzistoare sînt date în anexa A1 /83/ și anexa A2.

### 5.3.3. Principalele subprograme constituente ale ARN-V

#### 5.3.3.1. Subrutina de tratare a întreruperilor generate de timer

Timerul utilizat, este partea constitutivă a circuitului 8155 și este folosit în modul de lucru 2, /79, 73/. Pinul de ieșire TIMER OUT este legat direct la intrarea de întrerupere RST 7.5 a microprocesorului 8085. Această intrare este activată cu frontul crescător al unui sennal de comandă.

În momentul activării intrării RST 7.5 se execută subrutina reprezentată în organigrama din fig.5.19.

Prin variabila BYTE se memorează tipul regimului de lucru (I, II sau III) curent și etapa în curs de parcurgere.

Astfel, după validarea întreruperilor și salvarea în stivă a acumulatorului și indicatorilor de condiții, se trece la testarea valorii lui BYTE. În cazul în care acesta are valoarea  $\gamma$ , înseamnă că s-a încheiat etapa de măsură și, după prelucrarea rezultatelor măsurării (valoare și semn), se re poziționează indicatorul de stivă, se ignoră adresa de întoarcere în programul apelant și se revine în programul principal prin instrucțiuni de



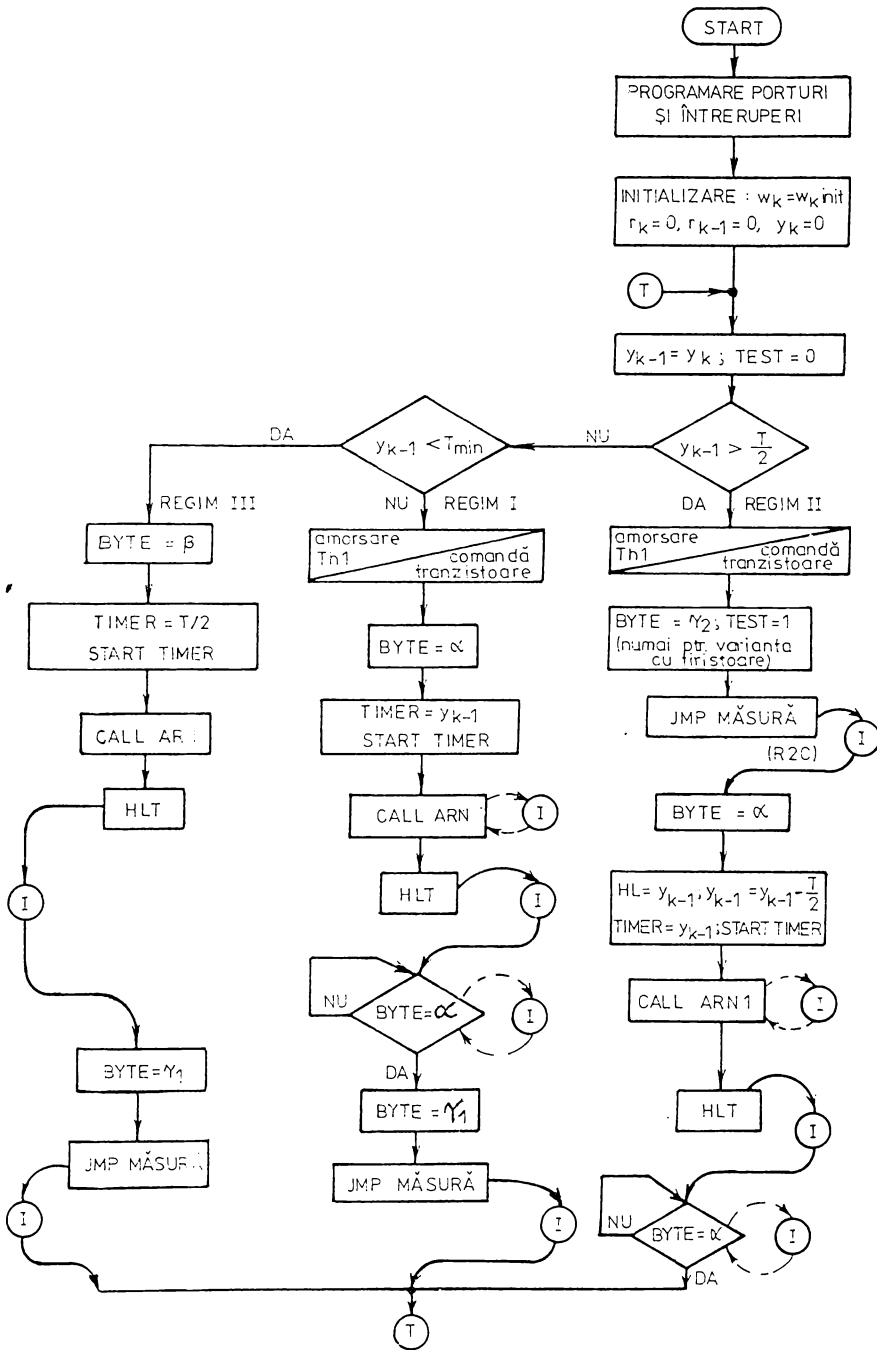


Figura 5.18. Organigrama programului pentru ARII-V

tip JMP.

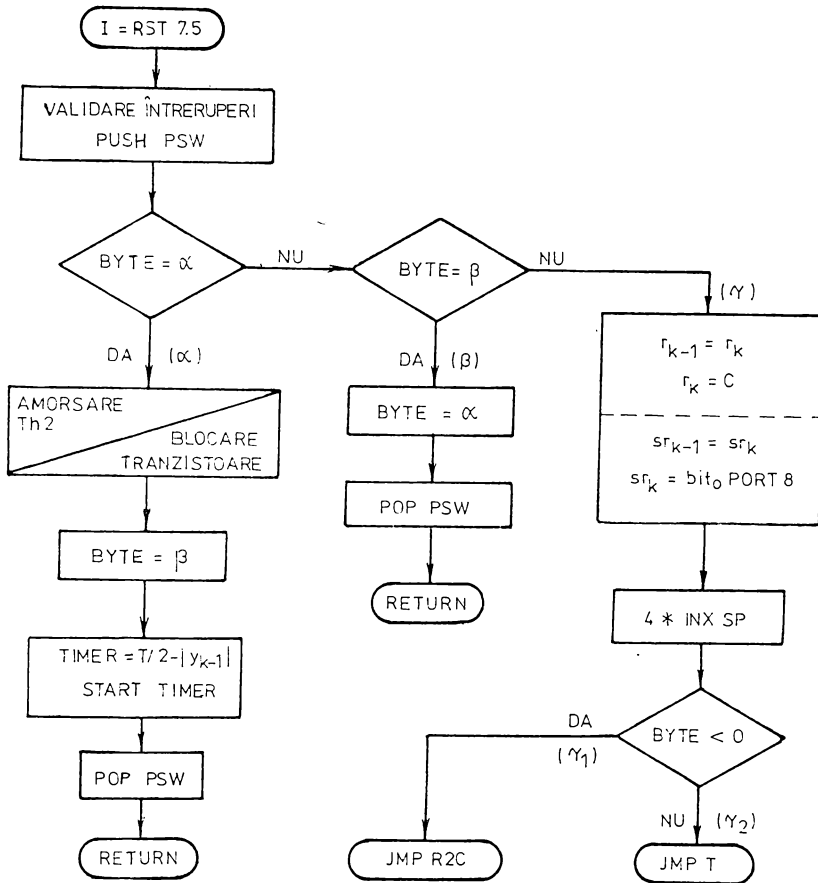


Figura 5.19. Organigrama programului de tratare a întreruperii RST 7.5

### 5.3.3.2. Subrutina MASURA

Această subrutină se utilizează pentru măsurarea turației curente, executată pe principiul de măsurare al frecvenței. Într-un interval de timp fix,  $T_{m\grave{a}s}$ , se numără impulsurile generate de TIRO, rezultatul obținut fiind proporțional cu frecvența acestora. Prin contorizarea ambelor fronturi ale unui impuls se obține o precizie dublă.  $T_{m\grave{a}s}$  este egal cu 5 ms, jumătatea perioadei de eșantionare.

Intre numărul obținut  $r_K$  și frecvența de rotație a axului motorului există următoarea legătură:

$$r_K = 2 N f T_{\text{măs}} \quad (5.67)$$

unde  $N$  este numărul de impulsuri generate de TIRO la o tură completă.

Pentru că, la aplicația concretă  $N = 1000$  și  $T_{\text{măs}} = 5 \cdot 10^{-3}$  s,

$$f[\text{Hz}] = \frac{r_K}{10} \quad (5.68)$$

iar

$$n[\text{rpm}] = 60 f = 6 r_K \quad (5.69)$$

Rezultatul contorizării se află în registrul C (fig.5.20.).

Subrutina se află înscrisă în memoria RAM static pentru ca precizia în numărare să nu fie influențată de întreruperile necesare pentru operațiunile de reîmprospătare (refresh) ale memoriilor RAM dinamic.

Subrutina nu se încheie printr-o instrucțiune de tip RETURN, ci ea se încheie când sosește semnalul de întrerupere generat de TIMER.

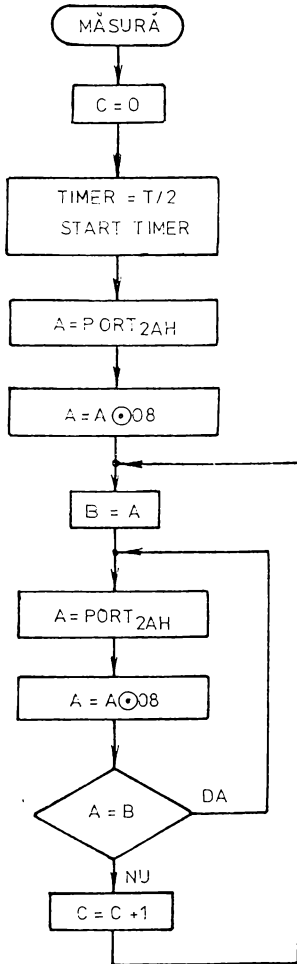


Figura 5.20. Organigrama subrutinei de măsură

### 5.3.3.3. Subrutina ARN. Calculul mărimii de comandă

Mărimea de comandă  $y_K$  se determină, după cum am arătat, cu relația (5.66):

$$y_K = y_{K-1} + 375(w_K - r_K) - 350(w_{K-1} - r_{K-1})$$

În această ecuație,  $y_K$ ,  $y_{K-1}$ , 350 și 375 sînt numere în-

tregi pe cîte 16 biți, considerînd bitul cel mai semnificativ bit de semn. Mărimile  $w_K$ ,  $w_{K-1}$ ,  $r_K$  și  $r_{K-1}$  sînt numere întregi pe 8 biți, cu semnul memorat separat în cîte o variabilă suplimentară atașată.

Valoarea maximă teoretică a mărimii de comandă  $y_K$  corespunde duratei perioadei de eșantionare:

$$y_{\max} = \frac{10 \text{ ms}}{325,52 \text{ ns}} = 30720 = 7800 \text{ H} \quad (5.70)$$

Alegerea regimului I sau II de lucru în program depinde de mărimea lui  $y_K$  - mai mare sau mai mică decît  $T/2$ , adică numeric:

$$y_{\text{Klim}} = \frac{y_{\text{Kmax}}}{2} = 15360 = 3000 \text{ H} \quad (5.71)$$

Subrutina ARN diferă pentru cele două tipuri de choppere utilizate și de aceea se prezintă separat fiecare caz.

#### a) Chopper cu tiristoare

Acționarea nu este reversibilă. Ca urmare,  $w_K$ ,  $w_{K-1}$ ,  $r_K$  și  $r_{K-1}$  sînt numere pozitive. Subrutina ARN este o subrutină dedicată, care execută exclusiv calculele, conform relației (5.66), cu valori fixe pentru coeficienți (375 și 350).

Organigrama pentru această subrutină este prezentată în fig.5.21.

Rezultatul scăderii dintre  $w_K$  și  $r_K$ , respectiv dintre  $w_{K-1}$  și  $r_{K-1}$  este un număr pe 8 biți, cu semnul memorat separat în variabila SIGN.

Valoarea obținută pentru  $y_K$  trebuie verificată în sensul încadrării ei în limite corecte.

În primul rînd, intervalul de siguranță între comenzile succesive pentru cele două tiristoare, după cum am arătat în paragraful 4.3.2.1., este de 500  $\mu$ s. Acestui interval îi corespunde o valoare admisă pentru  $y_K$  de 7200 H.

O a doua posibilitate de depășire a limitelor admise pentru  $y_K$  este în sensul obținerii de valori negative.

Intrucît un număr negativ, exprimat în codul complement față de 2, are bitul cel mai semnificativ egal cu 1, iar o depășire superioară echivalentă cu un rezultat  $y_K > 7200 \text{ H}$ , rezultă o posibilitate de manifestare în același fel a celor două tipuri de depășiri dacă programul testează bitul cel mai semnificativ al valorii obținute pentru  $y_K$ .

În scopul deosebirii celor două situații, este folosită

variabila fanion TEST în programul principal (fig.5.18.). Prin această variabilă se indică dacă se lucrează în regimul II

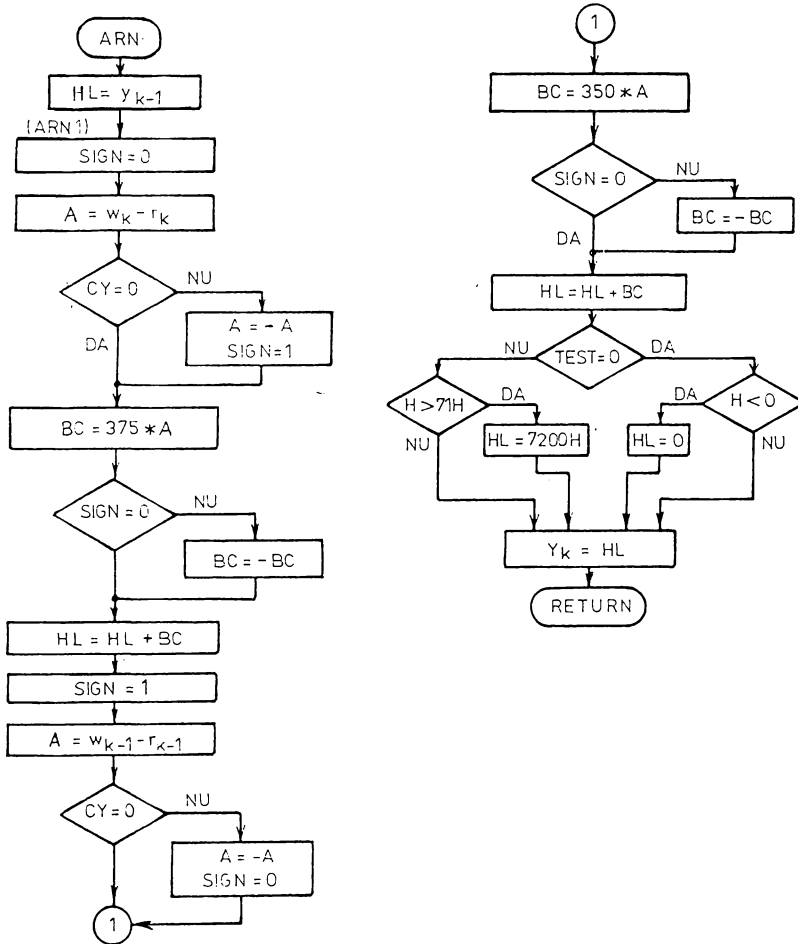


Figura 5.21. Organigrama subrutinei ARN în cazul utilizării chopperului cu tiristoare

$(y_K > \frac{T}{2} = 3000 H)$  sau nu. În cazul în care se lucrează în regimul II, se testează non-depășirea limitei superioare, iar în caz contrar, se testează încadrarea lui  $y$  în domeniul valorilor pozitive. Se elimină, astfel, alternativa unei decizii eronate.

În ceea ce privește calculul propriu-zis al mărimii  $y_K$ , conform relației (5.66), aceasta constă numai din adunări, scăderi și înmulțiri.

În vederea scurtării duratei calculului, s-a ales pentru efectuarea înmulțirilor un algoritm original, simplu, care pornește de la forma binară concretă a coeficienților 375 și 350. Cu acest algoritm s-a redus înmulțirea la numai câteva adunări și scăderi /83/. Algoritmul utilizat este însă rigid, în sensul că este conceput conform valorilor particulare ale coeficienților și deci nu se poate utiliza în cazul schimbării acestora. Subrutina ARN, în această variantă, se execută într-un interval de timp fix, de 225 μs.

b) Chopper cu tranzistoare

Utilizând chopperul cu tranzistoare, acționarea este reversibilă. Din acest motiv, semnele pentru mărimile reprezentate pe câte 8 biți sînt păstrate separat. Utilizînd relația(5.49)

$$y_K = y_{K-1} + d_0(w_K - r_K) + d_1(w_{K-1} - r_{K-1}),$$

determinată în paragraful 5.2.3., pot să apară cazuri în care valorile termenilor diferență din (5.49) să nu se încadreze în forma de reprezentare pe 8 biți. Acest lucru se întîmplă dacă semnul mărimii de prescriere și al celei de reacție nu coincid. În aceste situații, se impune necesitatea limitării rezultatelor parțiale ce se obțin.

Modul în care se efectuează scăderea  $w_K - r_K$ , în funcție de semnele acestora este prezentat în tabelul 5.1.

TABELUL 5.1.

Semn $w_K$	Semn $r_K$	Operația care se efectuează	SIGN	Dacă se face test de depășire	Limitare în caz de depășire $a_K$
+(o)	+(o)	$ w_K  -  r_K $	dat de calcul	nu	-
+(o)	-(1)	$ w_K  +  r_K $	+(o)	da	255 = FF H
-(1)	+(o)	$ w_K  +  r_K $	-(1)	da	255 = FF H
-(1)	-(1)	$ r_K  -  w_K $	dat de calcul	nu	-

Scăderea  $r_{K-1} - w_{K-1}$  se efectuează conform tabelului 5.2.

Cu variabila SIGN se memorează semnul rezultatului scăderilor  $w_K - r_K$  și  $w_{K-1} - r_{K-1}$ .

În cazul chopperului cu tranzistoare, subrutina ARN utilizează o metodă flexibilă pentru efectuarea înmulțirilor. Me-

toda, deși folosește un algoritm similar celui utilizat în înmulțirea manuală, prezintă originalitate în ceea ce privește modul concret de implementare software. Coeficienții  $d_0$  și  $d_1$  din relația (5.49) nu sînt ficși, ca și la chopperul cu tiristoare, ci pot lua orice valori de numere reprezentabile pe 16 biți. Semnele acestor coeficienți se consideră întotdeauna pozitive pentru  $d_0$  și negative pentru  $d_1$ .

TABELUL 5.2.

Semn $w_{K-1}$	Semn $r_{K-1}$	Operația care se efectuează	SIGN	Dacă se face test de depășire	Limitare în caz de depășire $a_{K-1}$	
+(0)	+(0)	$ r_{K-1}  -  w_{K-1} $		dat de calcul	nu	-
+(0)	-(1)	$ r_{K-1}  +  w_{K-1} $	-(1)		da	255 = FF H
-(1)	+(0)	$ r_{K-1}  +  w_{K-1} $	+(0)		da	255 = FF H
-(1)	-(1)	$ w_{K-1}  -  r_{K-1} $		dat de calcul	nu	-

Organigrama subrutinei ARN pentru cazul utilizării chopperului cu tranzistoare este reprezentată în fig.5.22. Rezultatul final al calculului este depus în registrele HL.

Operația  $HL = HL + B$ . DE este executată într-o altă subrutină ajutătoare INM (fig.5.22.). INM operează după un algoritm similar cu cel folosit la înmulțirea manuală. După cum se știe, /24/, algoritmul de înmulțire, după model manual, a numerelor binare, presupune un șir de adunări de produse parțiale, obținute prin rotirea spre stînga, în acumulator, a unuia dintre factori. Minimizarea timpului de execuție a înmulțirilor s-a obținut cu utilizarea instrucțiunilor de tip DAD rp, de adunare a unui registru pereche la registrul HL, atît pentru însumarea produselor parțiale ( $HL = HL + DE$ ), cît și pentru deplasarea la stînga a registrului pereche DE (prin secvența XCHG, DAD H, XCHG).

Subrutina ARN descrisă în acest paragraf, se execută în intervale de timp ce pot fi variabile, dar care se încadrează sigur în limita de 500  $\mu$ s.

Si în acest caz se verifică dacă  $y_K$  obținut prin calcul nu depășește valoarea limită admisă (7200 H). Diferența esențială, față de cazul chopperului cu tiristoare, este că în acest caz  $y_K$  poate lua atît valori pozitive cît și negative. Semnul

lui  $y_K$  este memorat suplimentar și în variabila TEST, utilizată în programul principal, care indică, în funcție de acest semn, perechea de tranzistoare care urmează să fie comandate.

Variabila TEST este reactualizată doar în cadrul regimurilor I și III de lucru (fig.5.18.). În cadrul regimului II de lucru se testează dacă s-a ajuns la depășiri sau nu pentru  $y_K$ . În cazul în care TEST = 0, se verifică dacă  $y_K \leq y_{Kmax}$  și, dacă nu, se ia  $y_K = y_{Kmax}$ . În cazul în care TEST = 1 se verifică dacă  $y_K \geq y_{Kmin} = -y_{Kmax}$  și, dacă nu, se ia  $y_K = y_{Kmin}$ .

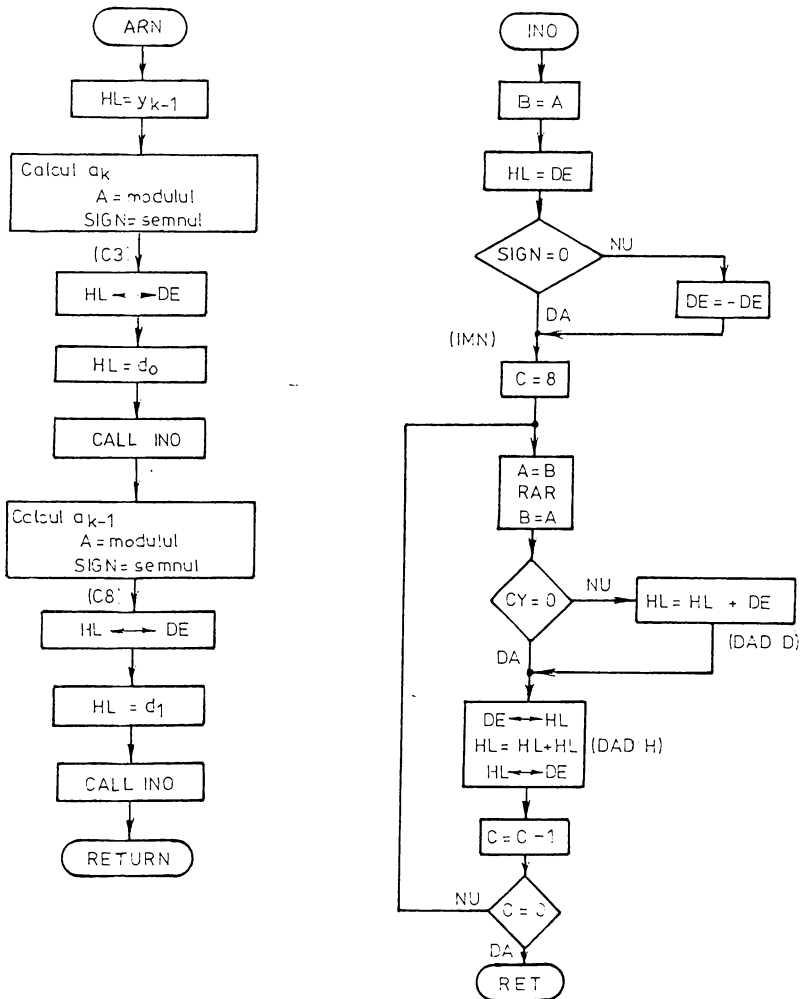


Figura 5.22. Organigrama subrutinei ARN în cazul chopperului cu tranzistoare



#### 5.3.3.4. Dialogul ON-LINE cu operatorul uman - subrutina INTERRUPT - DAF /83, 36/

Operatorul uman transmite microsistemului valoarea prescrisă a turației prin tastatura DAF. Deoarece DAF-ul lucrează în modul full-duplex (transmisie oarbă), microsistemul trebuie să retransmită, ca ecou, caracterele recepționate.

Interfața cu DAF-ul a fost concepută în așa fel încât acesta este tratat ca și o locație de memorie, cu adresa între 7000 H și 7FFFH. La apăsarea unei taste din claviatura DAF, se activează cererea de întrerupere RST 6.5 și se apelează subrutina INTERRUPT-DAF.

Subrutina este realizată în două variante. În ambele variante, variabila FAN asigură ca valoarea preluată să fie introdusă efectiv în locația destinată ei la începutul unei perioade de eșantionare. La începutul subrutinei se salvează în stivă toate registrele care vor fi afectate în timpul utilizării ei, pentru ca acestea să poată fi restaurate înainte de executarea instrucției RETURN.

Varianta folosită în cazul utilizării chopperului cu tiristoare acceptă la intrare două cifre hexazecimale. Atunci când acestea sînt recepționate, cursorul DAF-ului trece la rîndul următor, așteptînd o nouă comandă.

Varianta folosită pentru chopperul cu tranzistoare acceptă la intrare un număr zecimal cu semn, format din 1 ÷ 3 cifre, după care se tastează RETURN. În acest moment, subrutina convertește numărul zecimal în hexazecimal, trimite către DAF: "rezultat conversie H" și trece la rîndul următor, așteptînd o nouă comandă. Preluarea mărimii prescrise la începutul programului principal, se face într-una sau mai multe perioade de eșantionare, astfel încît modulul diferenței dintre mărimea prescrisă anterioară și cea curentă să fie de maximum 20, pentru a se evita salturi prea mari în  $y_K$ , care să conducă la depășiri necontrolabile prin software.

#### 5.3.3.5. Dialogul OFF-LINE cu operatorul uman

Dialogul off-line cu operatorul uman asigură acestuia po-

sibilitatea modificării parametrilor regulatorului, pentru optimizarea funcționării buclei de reglare.

Acest dialog se realizează prin intermediul unui subprogram ce se lansează în execuție de la adresa ClooH. El poate fi executat doar în cazul regulatorului pentru chopperul cu tranzistoare, caz în care subrutina ARN concepută (cazul b, paragraful 5.3.3.3.) permite modificarea coeficienților  $d_0$  și  $d_1$  ai ecuației discrete pentru  $y_K$ .

Execuția programului /83, 12/ începe prin editarea unui text introductiv, afișat pe ecranul DAF-ului, în care se arată legea de reglare implementată și semnificația coeficienților; apoi se cere introducerea parametrilor  $K_R$  și  $T_i$ , într-un format bine precizat:  $K_R \cdot 10^{-4}$ ,  $T_i$ (ms).

În continuare, subprogramul determină modulele coeficienților  $d_0$  și  $d_1$  din următorul sistem de ecuații:

$$\begin{cases} 2(d_0 + d_1) = 123 K_R & (5.72) \\ 2(d_0 - d_1) = (d_0 + d_1) \cdot 10/T_i \end{cases}$$

$$\begin{cases} d_0 = \frac{123 \cdot K_R}{4} + \frac{123 \cdot K_R \cdot 5}{4 T_i} \\ d_1 = \frac{123 \cdot K_R}{4} - \frac{123 \cdot K_R \cdot 5}{4 T_i} \end{cases} \quad (5.73)$$

Pentru erori de trunchiere minime, calculele se execută în ordinea următoare:

$$\begin{aligned} a &= 123 \cdot K_R; & b &= a/4; & c &= b \cdot 5; & d &= c/T_i \\ d_0 &= b + d; & d_1 &= b - d \end{aligned} \quad (5.74)$$

La sfârșitul subprogramului se editează un text final, în care se prezintă forma ecuației (5.66) pentru ARN și se afișează valorile pentru  $d_0$  și  $d_1$ , în format zecimal și hexazecimal. Valorile obținute sînt introduse în locațiile corespunzătoare lor, astfel încît programul de regulator să poată fi lansat.

#### 5.3.4. Considerații asupra timpului mort total al buclei de reglare

Timpul mort total al unei bucle de reglare se definește ca fiind suma tuturor întârzierilor care apar în transmiterea și prelucrarea informației vehiculate în cadrul acesteia /16/.

După cum s-a arătat în paragraful 5.2.4., timpul mort total al buclei are influență importantă asupra marginii de fază a SRA-V. Structura și valoarea acestui timp mort se pot determina doar după implementarea regulatorului pe microsistemul de calcul.

Structura timpului mort este diferită, funcție de durata de conducție necesară, adică funcție de mărimea de comandă  $y_K$ , rezultată din calcul. În fig.5.23. este reprezentată grafic această structură (graficele a, b, c, d) pentru cele 4 categorii de valori posibile pentru  $y_K$ . Se prezintă, de asemenea în această figură și graficul de dependență  $\tau_M = f(y_K)$ .

Se observă de pe graficele reprezentate, că timpul mort mediu este de 12,5 ms, adică valoarea adoptată inițial în proiectare. Abaterea de  $\pm 2,5$  ms nu influențează sensibil marginea de fază.

### 5.3.5. Experimentarea SRA-V

#### a) Experimentarea în buclă deschisă

Experimentele în buclă deschisă se efectuează în scopul verificării compatibilității reciproce a elementelor componente ale buclei de reglare /32/.

Configurația utilizată este prezentată în fig.5.24. În locul traductorului incremental (TIRO) se folosește un generator de semnale dreptunghiulare, cu frecvența reglabilă și de nivel TTL (în cazul concret aparatul VERSATESTER, de producție IEMI București). Elementul de execuție utilizat este chopperul cu tiristoare.

Pentru o mărime de comandă fixă  $w = 64 \text{ H} = 100$ , motorul trebuie să se rotească cu viteza de 10 rot/s, ceea ce corespunde unei frecvențe de 10 kHz la ieșirea traductorului TIRO, solidar cu axul motorului. În cazul experimentului în buclă deschisă se fixează frecvența generatorului la 10 kHz. Ieșirea generatorului se conectează la linia portului de intrare din sistemul de calcul, utilizat pentru măsură.

Dacă mărimea de comandă se menține fixă (64 H) și se micșorează frecvența semnalului dat de generator, regulatorul PI integrează abaterea (pozitivă), mărinđ mărimea de comandă  $y_K$ . Se obține o accelerare a motorului.

Cu aceeași valoare a mărimii de prescriere se mărește

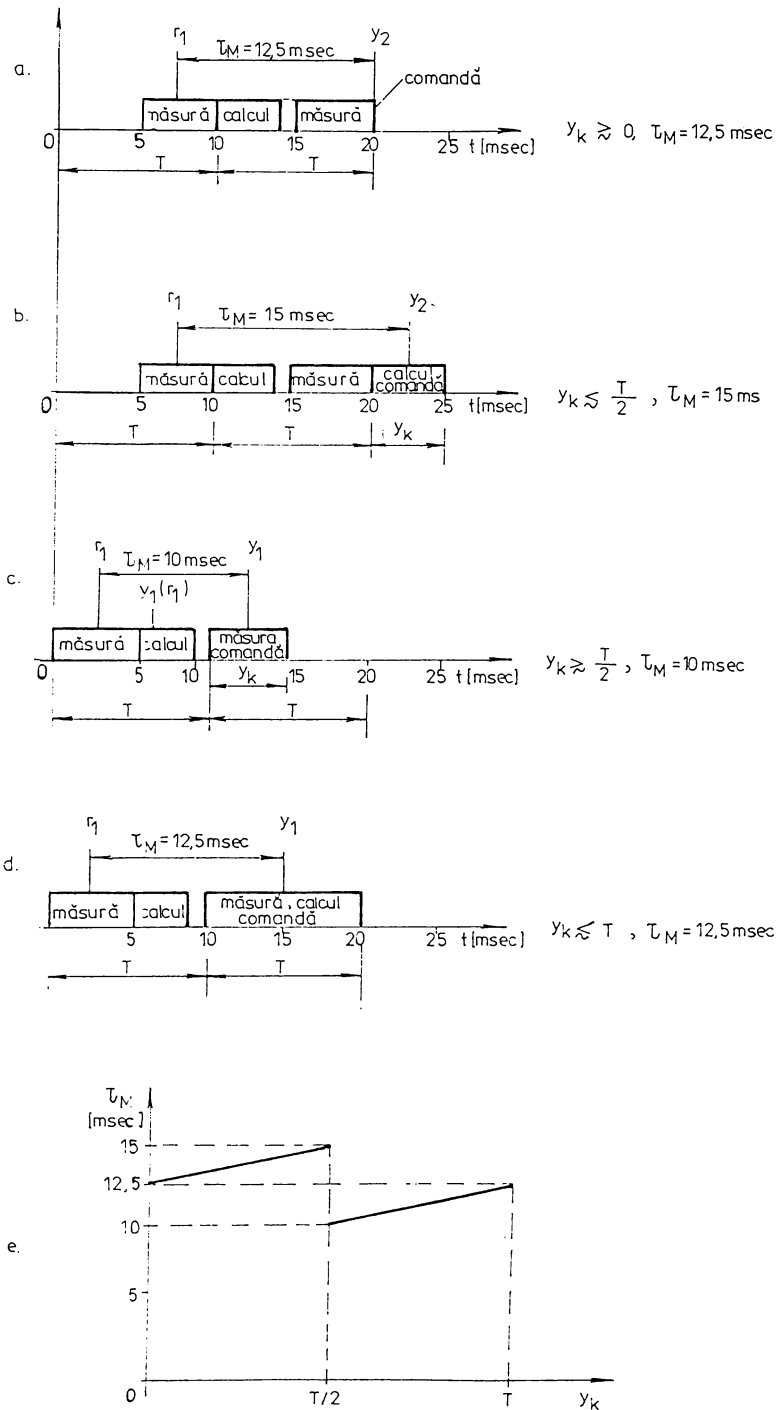


Figura 5.23. Structura timpului mort pentru diferite valori ale mărimii de comandă. Graficul dependenței  $\tau_M = f(y_k)$

frecvența semnalului furnizat de generator și rezultă o decelerare a motorului.

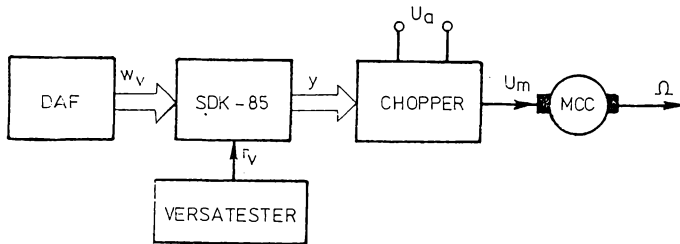


Figura 5.24. Schema bloc a sistemului experimental în buclă deschisă

Efecte similare se obțin și dacă se înscrie, prin DAF, o mărime de prescriere diferită de 64 H și se menține constantă frecvența generatorului. Pentru o prescriere mai mare de 64 H, se realizează o accelerare iar pentru o valoare mai mică decât 64 H se obține o decelerare a motorului.

b) Experimentarea în buclă închisă

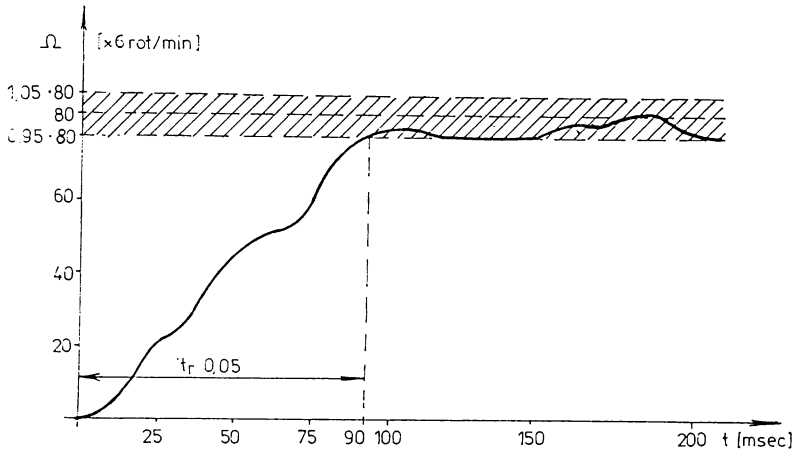
Experimentările efectuate cu sistemul în buclă închisă au confirmat calculele de proiectare.

În regim staționar au fost produse perturbații, prin modificarea bruscă a tensiunii de alimentare (între limitele 14 ÷ 24 V) și a sarcinii mecanice.

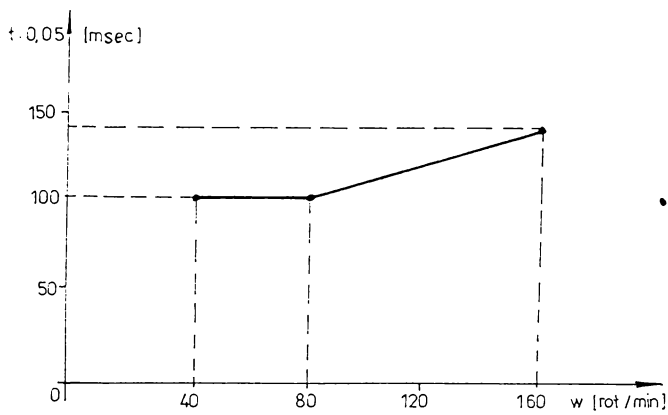
Turația motorului a suferit o ușoară variație (cca 1%) în momentul comutării tensiunii de alimentare și o revenire imediată la valoarea exactă prescrisă. S-a constatat, de asemenea, și o bună compensare a perturbațiilor prin variația sarcinii mecanice.

În regim dinamic, pentru o variație treaptă a mărimii prescrise, cu motorul pornit din repaus, răspunsul SRA a fost aperiodic pentru un sens de rotație, respectiv fie aperiodic, fie oscilant amortizat - funcție de amplitudinea treptei aplicate - pentru celălalt sens de rotație. Rezultatele cantitative ale experimentelor sînt reprezentate în fig.5.25. pentru un sens de rotație, respectiv fig.5.26. pentru celălalt sens de rotație.

În cazul unei variații treaptă a mărimii prescrise, aplicată în timp ce motorul se află în rotație, răspunsul SRA a fost oscilant amortizat. Principalii indicatori de calitate ai sistemului, /32/, sînt reprezentați în fig.5.27. și fig.5.28. S-au

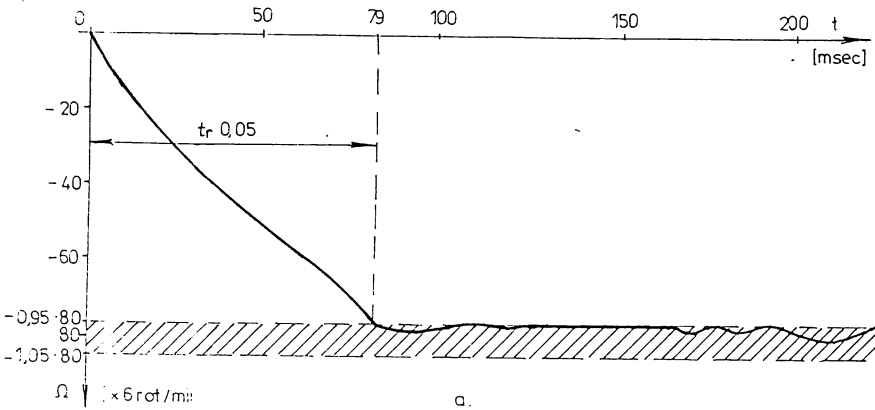


a.

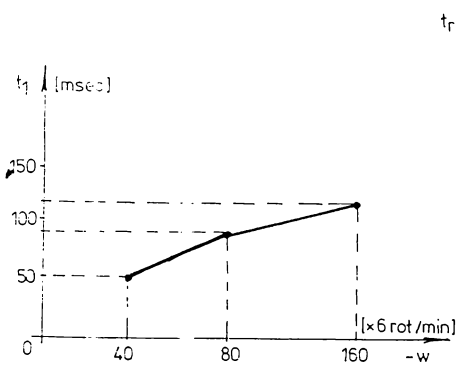


b.

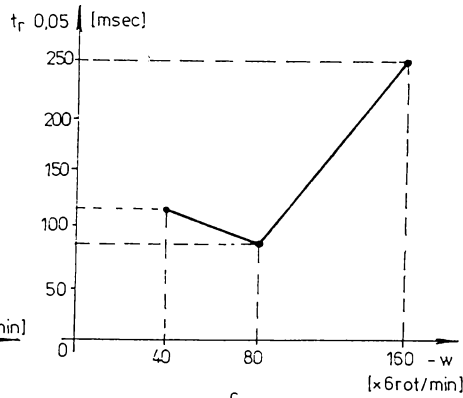
Figura 5.25. Răspunsul SRA la o variație treaptă a mărimii prescrise și dependența timpului de reglare de amplitudinea treptei de variație a mărimii prescrise în experimentul cu motorul pornind din repaus într-un sens de rotație



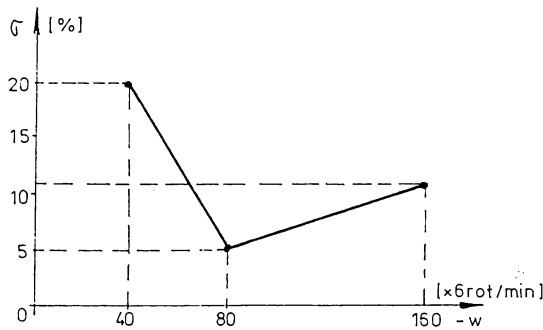
a.



b.



c.



d.

Figura 5.26. Răspunsul SRA la o variație treaptă a mărimii de prescriere și dependența principalilor indicatori de calitate ai SRA-V de amplitudinea treptei de variație a mărimii prescrise la experimentul cu motorul pornind din repaus în cel de al doilea sens de rotație: a - răspunsul SRA; b - timpul de primă reglare (timpul primei atingeri a valorii staționare); c - timpul de reglare (durata regimului tranzitoriu); d - supra-reglajul

aplicat diferite trepte de variație pentru mărimea prescrisă în jurul valorilor  $w_0 = 480 \text{ rot/min}$  (fig.5.27.) și, respectiv,  $w_0 = -480 \text{ rot/min}$  (fig.5.28.) ale acesteia.

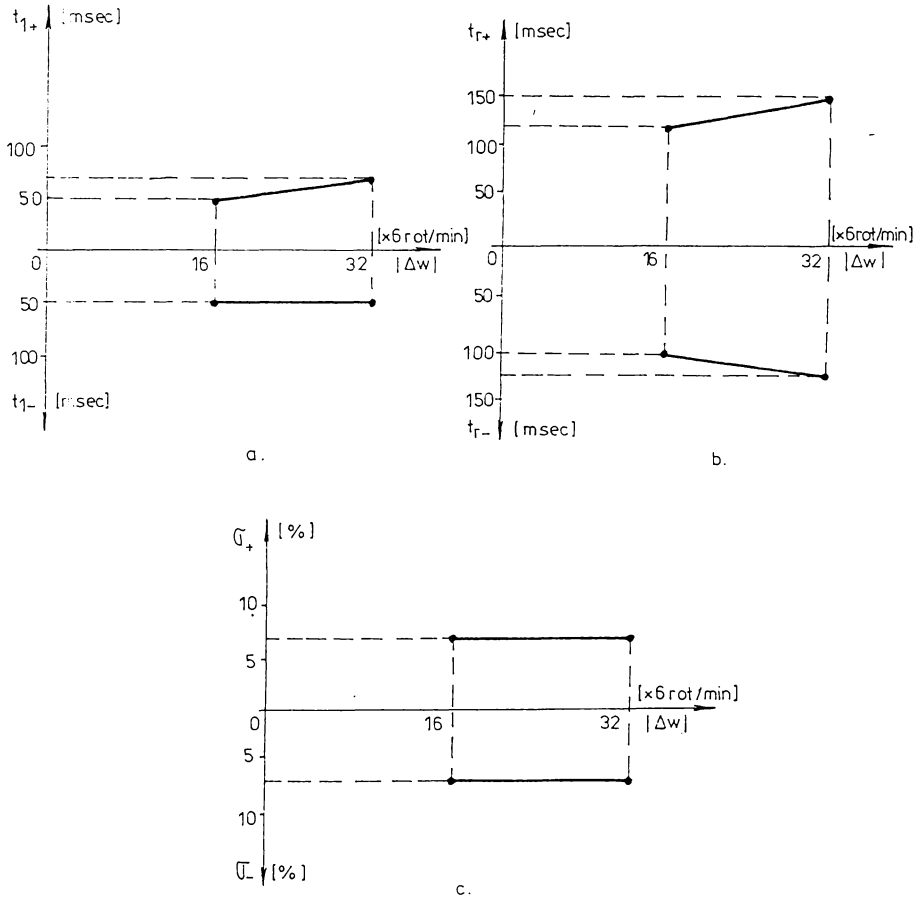
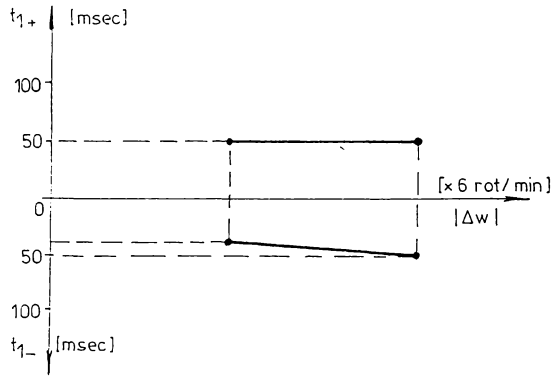
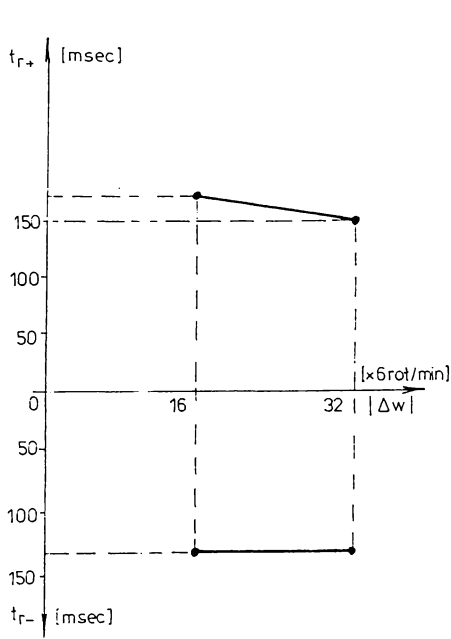


Figura 5.27. Dependența principalilor indicatori de calitate ai SRA-V de amplitudinea trepte de variație a mărimii prescrise, în experimentul cu motorul aflat în rotație într-un sens: a - timpul de primă reglare; b - timpul de reglare; c - suprareglajul

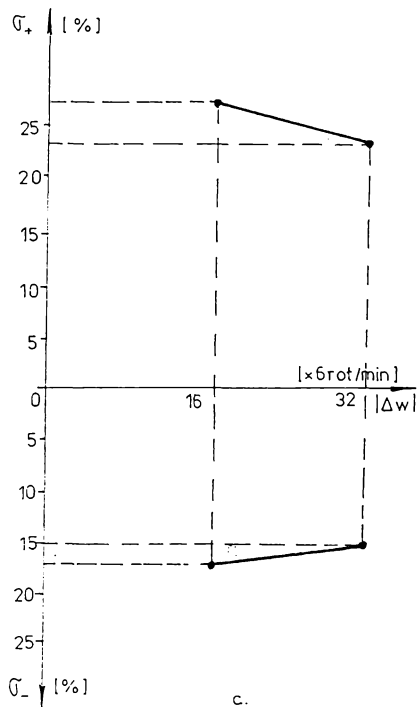




a.



b.



c.

Figura.5.28. Dependența principalilor indicatori de calitate ai SRA-V de amplitudinea treptei de variație a mărimii prescrise în experimentul cu motorul aflat în rotație în cel de al doilea sens

5.4. Implementarea algoritmului de reglare numerică a vitezei și poziției /35/

ARN-VP a rezultat de forma ecuației discrete (5.62):

$y_K = y_{K-1} + D_0 w_{PK} + D_1 w_{PK-1} + D_2 r_{PK} + D_3 r_{PK-1} + D_4 r_{PK-2}$ , unde coeficienții  $D_0, D_1, D_2, D_3, D_4$  au fost calculați în paragraful 5.2.5.

Valorile obținute pentru acești coeficienți trebuie normate, la fel ca în cazul ARN-V. Ținând seama de cele prezentate în paragraful 5.3.1., valorile normate pentru coeficienți sînt:

$$\begin{aligned} D_0 &= \frac{4,88 \cdot 10^{-6}}{325,52 \cdot 10^{-9}} = 15 ; & D_1 &= \frac{-4,56 \cdot 10^{-6}}{325,52 \cdot 10^{-9}} = -14 ; \\ D_2 &= \frac{-1,27 \cdot 10^{-4}}{325,52 \cdot 10^{-9}} = -390 ; & D_3 &= \frac{2,41 \cdot 10^{-4}}{325,52 \cdot 10^{-9}} = 739 ; \\ D_4 &= \frac{-1,14 \cdot 10^{-4}}{325,52 \cdot 10^{-9}} = -350 . \end{aligned} \quad (5.75)$$

Se obține astfel următoarea ecuație discretă, care trebuie implementată software:

$$y_K = y_{K-1} + 15w_K - 14w_{K-1} - 390r_K + 739r_{K-1} - 350r_{K-2} \quad (5.76)$$

În această formă trebuie efectuate 5 înmulțiri. Numărul de înmulțiri poate fi redus, utilizînd un simplu artificiu matematic, prin care se ajunge la forma:

$$\begin{aligned} y_K &= y_{K-1} + 390(w_K - r_K) - 375(w_K - r_{K-1}) - 364(w_{K-1} - r_{K-1}) + \\ &+ 350(w_{K-1} - r_{K-2}) \end{aligned} \quad (5.77)$$

În această formă sînt necesare doar patru înmulțiri. Dacă se face o analiză comparativă între cele două forme ale ecuației discrete (5.76), (5.77), este preferabilă totuși utilizarea formei (5.76), cu toate că necesită efectuarea unei înmulțiri suplimentare. Acest lucru este justificat de faptul că în forma (5.76) două înmulțiri se efectuează între un număr de 8 biți și unul de 16 biți, iar semnele tuturor produselor se cunosc dinainte, în timp ce în forma (5.77), toate înmulțirile sînt de tipul 16 biți x 16 biți, iar semnele produselor sînt variabile.

Important, în ceea ce privește efectuarea calculelor, este nu atît timpul mediu de efectuare a acestora, cît mai ales, timpul maxim de execuție, care trebuie să fie mai mic de 5 ms (1/2 din perioada de eșantionare).

5.4.1. Funcțiile îndeplinite de microsystem.  
Structura programului ARN-VP

ECN utilizat în SRA-VP este microsystemul SDK-85 și asigură îndeplinirea următoarelor funcții:

- (F1) - preluarea mărimii prescrise  $w_K$ , de la DAF, printr-o întreprere RST 6.5;
- (F2) - preluarea mărimii de reacție  $r_K$  de la numărător;
- (F3) - calculul mărimii de comandă  $y_K$ , conform ARN (5.76);
- (F4) - conversia numeric-analitică a acesteia, într-un interval de timp real  $T_{c,K}$ , (utilizând timerul);
- (F5) - comanda propriu-zisă a chopperului.

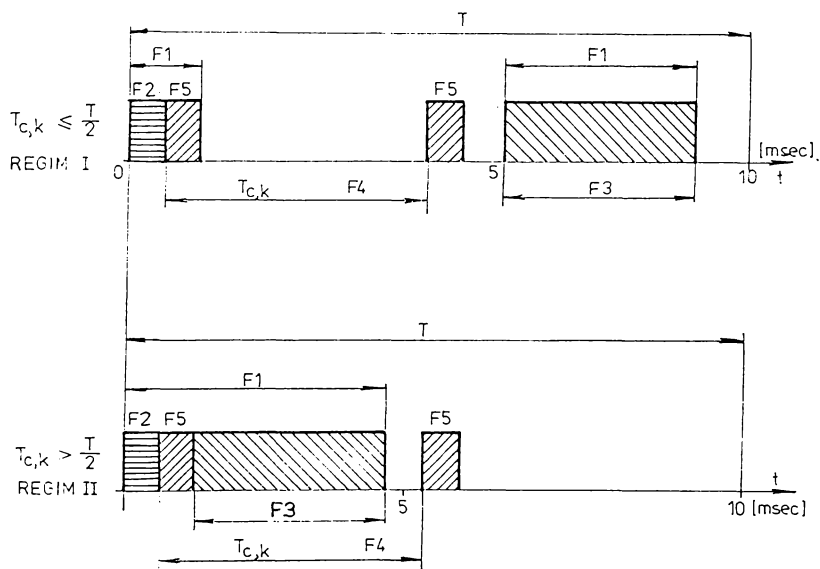


Figura 5.29. Modul de desfășurare în timp (de-a lungul perioadei de eșantionare) al momentelor de execuție a celor cinci funcții.

În funcție de valoarea lui  $T_{c,K}$ , mai mică sau mai mare decât 5 ms, pot exista două regimuri de gestionare a perioadei de eșantionare (fig.5.29).

În cazul  $y_K = 0$  (motor oprit), se utilizează regimul I,  $T_{c,K}$  generat în acest caz este de o perioadă a timerului (325,52 ns), interval la care motorul oricum nu poate răspunde.

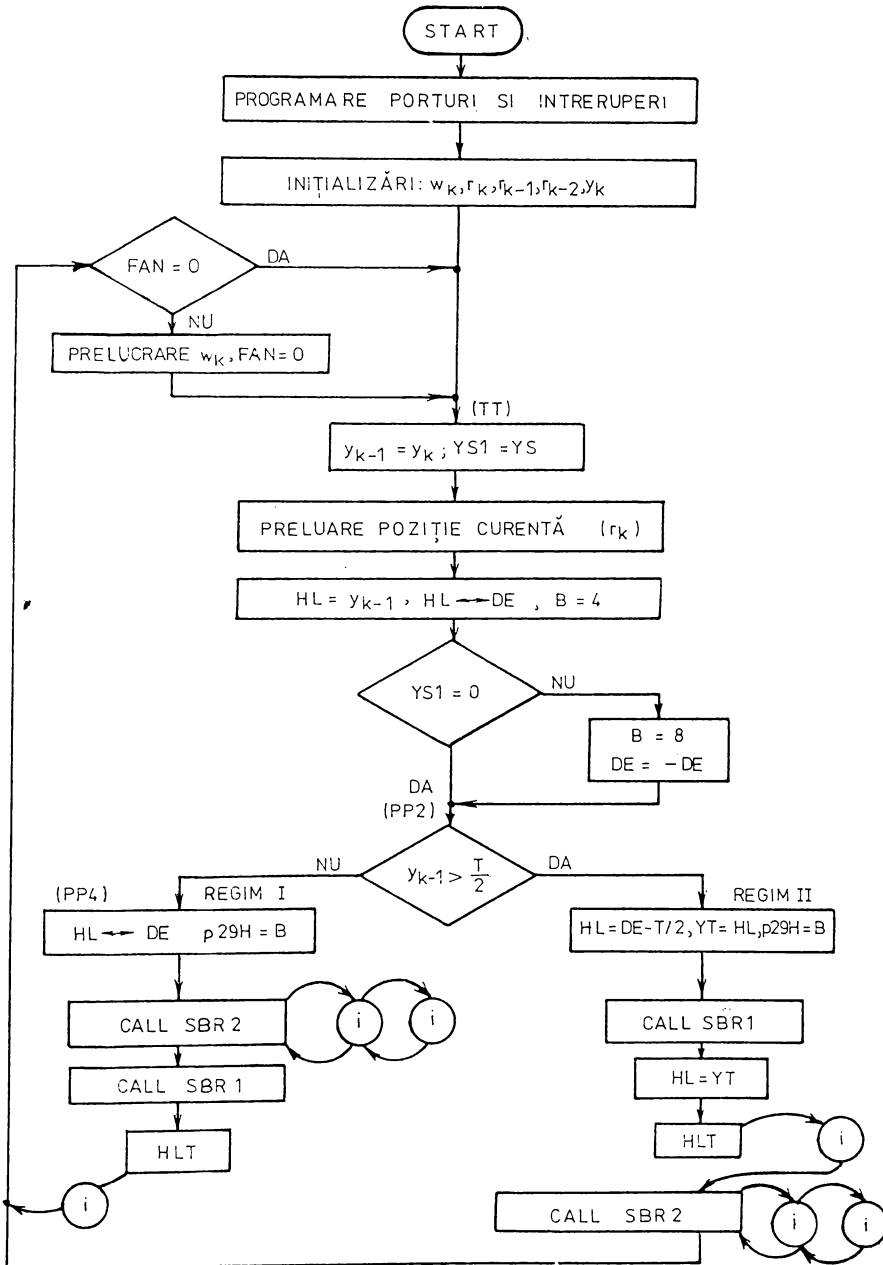


Figura 5.30. Organigrama ARN-VP

Organigrama programului este prezentată în fig.5.30. Funcțiile (F3) și (F5) sînt realizate în subrutinele SBR 1 și SBR 3.

#### 5.4.2. Principalele subprograme ale ARN-VP

##### 5.4.2.1. Subrutina de tratare a întreruperilor de la timer

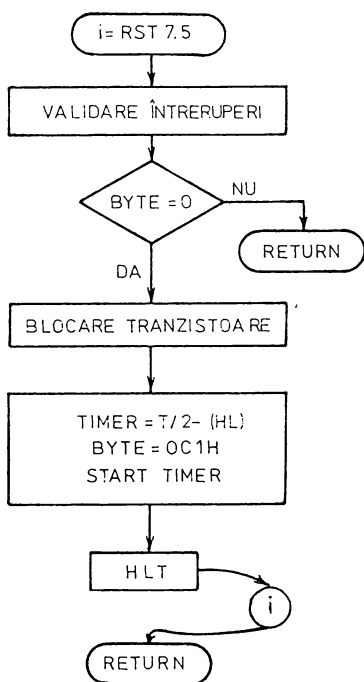


Figura 5.31. Subrutina de tratare a întreruperilor

Organigrama subrutinei apelate în momentul cînd timerul generează o cerere de întrerupere este prezentată în fig.5.31.

Subrutina utilizează variabila BYTE. În funcție de valoarea acestei variabile se revine la programul apelant (pentru  $BYTE \neq 0$ ), sau se comandă blocarea tranzistoarelor chopperului prin portul 29 H. În cel de al doilea caz, după comanda de blocare a chopperului, se încarcă în timer durata  $T/2 - (HL)$ , se pornește din nou timerul și se așteaptă în starea HLT o nouă cerere de întrerupere din partea timerului. Prin urmare, subrutina este de tip recursiv, apelîndu-se pe a însăși. La o nouă apelare însă, valoarea lui BYTE este diferită de zero și execuția subrutinei se încheie (RETURN).

##### 5.4.2.2. Subrutina SBR 2 - pentru generarea intervalului de conducție al chopperului

Intervalul de conducție pentru chopper este calculat pe baza ARN și valoarea acestuia este încărcată în registrul pereche (HL). În momentul apelării SBR 2, conținutul perechii de registre HL este transferat în timer și după pornirea acestuia,

se așteaptă în HLT o altă cerere de întrerupere (fig.5.32.).

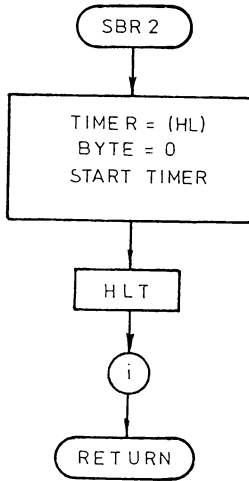


Figura 5.32. Subrutina SBR-2

#### 5.4.2.3. Preluarea poziției curente de la numărător

La începutul fiecărei perioade de eșantionare se citește de la numărător valoarea poziției curente ( $r_K$ ). Ieșirile celor 16 bistabile din numărător sînt legate la două porturi de intrare (0 și 1) ale microsistemului. Conținutul celor două porturi se citește secvențial.

Citirea conținutului numărătorului se efectuează asincron în raport cu impulsurile generate de TIRO - care se numără - și de aceea există următoarele

surse de erori:

- citirea unui port să se efectueze exact în momentul unei tranziții în numărător. De exemplu, la o tranziție de la 7FH spre 8oH se poate citi, în mod eronat, valoarea ooH;
- apariția unei tranziții în numărător, în intervalul de timp dintre citirile celor două porturi. De exemplu, la o tranziție de la o5FF H spre o6oo H în numărător, se poate citi întâi valoarea FFH (prin portul 0 - biții cei mai puțin semnificativi) și apoi valoarea o6H (prin portul 1 - biții cei mai puțin semnificativi), rezultînd în mod eronat poziția o6FF H.

Aceste surse de erori se elimină utilizînd citiri succesive ale valorilor înscrise în fiecare dintre porturi.

Organigrama subrutinei este prezentată în fig.5.33.

#### 5.4.2.4. Subrutina SBR 1 - calculul mărimii de comandă

La începutul subrutinei se inițializează și se pornește timerul, pentru a genera un interval de timp  $t = T/2 = 5$  ms, durata maximă admisă pentru efectuarea calculelor (fig.5.34., a., b.).

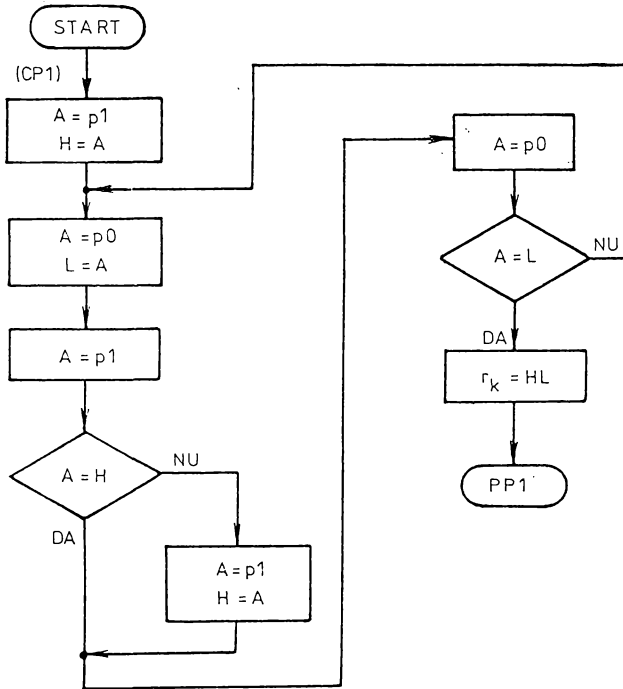


Figura 5.33. Subrutina de citire a poziției curente

Tinînd cont de (5.62):

$$Y_K = Y_{K-1} + D_0 w_{PK} + D_1 w_{PK-1} + D_2 r_{PK} + D_3 r_{PK-1} + D_4 r_{PK-2}$$

cu valorile concrete din (5.76):

$$Y_K = Y_{K-1} + 15w_K - 14w_{K-1} - 390 r_K + 739r_{K-1} - 350 r_{K-2}$$

obținute pentru ARN-VP, se continuă cu efectuarea calculului mărimii de comandă. Rezultatul calculului este reprezentat sub forma unui număr de 24 biți, depus în registrele C, H și L. Se asigură astfel posibilitatea sesizării corecte a depășirilor (vezi paragraful 5.3.3.3.).

Înmulțirile de tipul 8 biți x 24 biți = 24 biți se efectuează cu subrutina I 24, care execută operația  $CHL = CHL + B \times x \text{ ADE}$ . Pentru coeficienții  $D_2$ ,  $D_3$  și  $D_4$ , care sînt reprezentați pe 16 biți, s-a ales soluția adunărilor repetate pentru înmulțirea octetului superior cu înmulțitorul. Această metodă, realizată

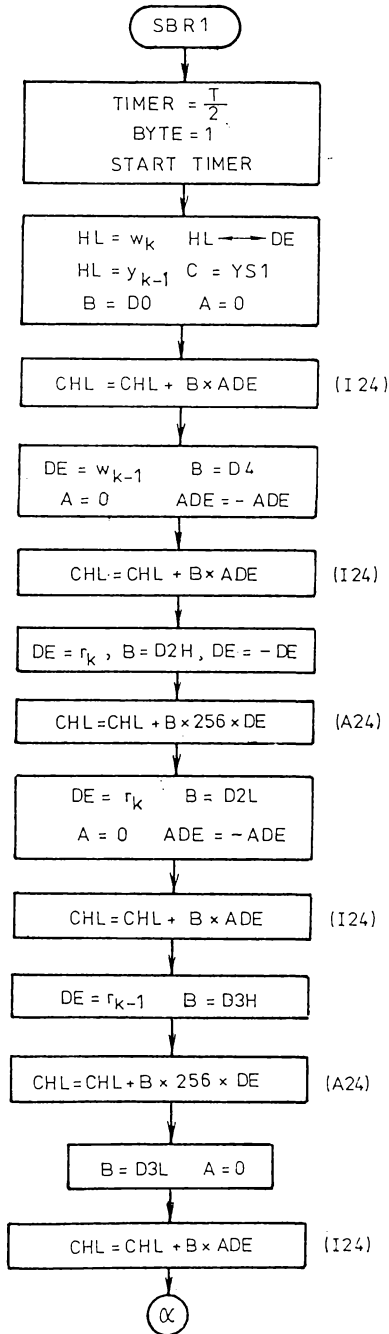


Figura 5.34. Organigrama subrutinei de calcul amărării de comandă

cu subrutina A 24, în care operația  $CHL = CHL + DE \times 256$  se execută de B ori, este rapidă, întrucât valorile octeților superiori ai deînmulțitului sînt mici.

Operațiile  $CHL = CHL - B \times ADE$  respectiv de B ori  $CHL = CHL - DE \times 256$ , se efectuează tot cu subrutinele I 24 și A 24, dar începînd de la adresele I2M, respectiv A2M. Pentru complementarea registrului pereche DE se utilizează subrutina AIN, care asigură și posibilitatea complementării pe 24 de biți a registrelor A, D, E.

Subrutinele I 24, A 24 și AIN sînt prezentate în organigramele din fig.5.35. și în listingul cu programul complet de la sfîrșitul lucrării. Aceste subrutine sînt înscrise în memoria RAM static a sistemului, pentru a scurta timpul de execuție.

În I 24, înmulțirile se efectuează după un algoritm similar cu cel utilizat în înmulțirea manuală. Insumarea produselor parțiale și rotirile spre stînga sînt în acest caz mai dificil de efectuat decît în cazul operațiilor pe 16 biți, prezentate în paragraful 5.3.3.3., dar principiul utilizat este similar. Se folosește și aici instrucția DAD rp, iar timpul de lucru cu stiva, absolut necesar, se scurtează cu ajutorul instrucției XTHL.

Complementarea față de



2, pe 24 de biți, se realizează prin complementarea registrelor A, D, E, urmată de incrementarea numărului pe 24 de biți obținut.

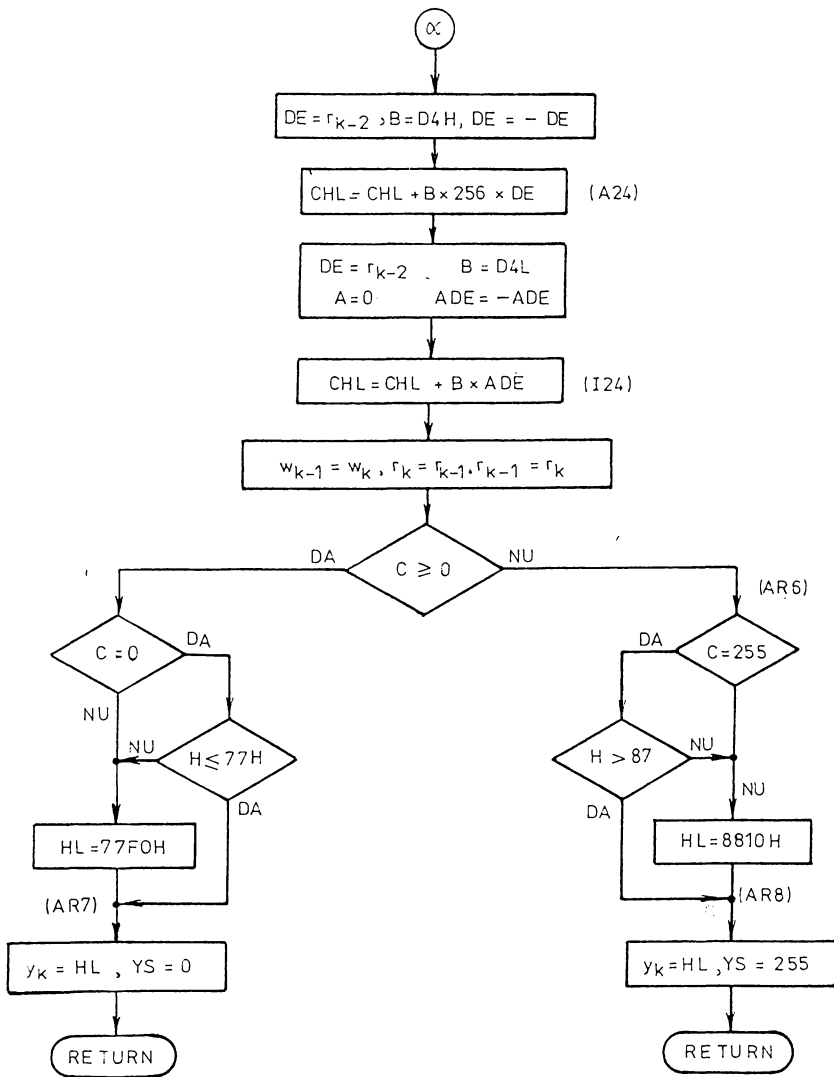


Figura 5.34. (continuare)

! Apare necesitatea transferării spre A a bitului de transport (CY), în situația în care, în urma incrementării, conținutul perechii DE devine zero. Instrucțiunile de incrementare, însă, nu afectează fanionul CY și de aceea acesta trebuie poziționat prin program.

Subrutina SBR 1 se încheie cu efectuarea testelor de depășire asupra valorii obținute prin mărimea de comandă (similar cu cazul SRAN-V, paragraful 5.3.3.3.) și limitat în cazul depășirii valorilor limită.

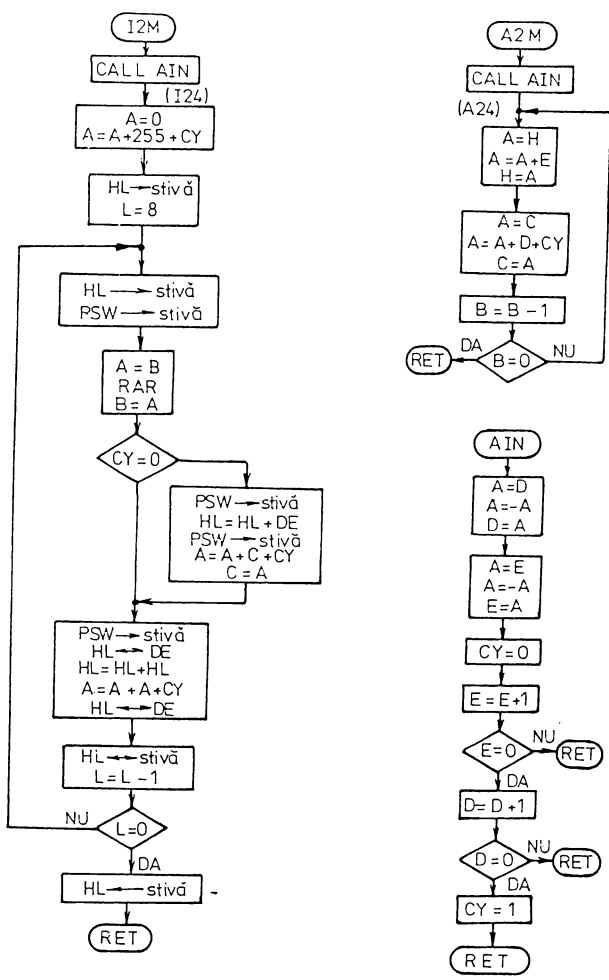


Figura 5.35. Subrutinele I24, A24 și AIN

#### 5.4.2.5. Dialogul ON-LINE - cu operatorul uman. Subrutina INTERRUPT-DAF

Mărimea prescrisă  $w_K$  este transmisă ECN, de către operatorul uman, prin intermediul consolei DAF-ului. Operatorul uman poate urmări pe ecranul DAF-ului confirmarea recepției corecte. Acest dialog este realizat prin intermediul subrutinei INTERRUPT-DAF.

Operatorul uman furnizează valoarea mărimii prescrise sub forma unui număr zecimal de  $1 \div 5$  cifre. Apăsarea tastei RETURN, de pe consolă, semnalează încheierea transmisiei.

Nivelul de întrerupere corespunzător DAF-ului este validat la începutul perioadei de eşantionare și este mascat în celelalte intervale de pe parcursul unei perioade de eşantionare (fig.5.29. - F1).

Subrutina semnalează programului principal terminarea recepției prin poziționarea variabilei FAN pe 1 logic. La începutul perioadei de eşantionare, imediat următoare, se execută conversia zecimal-hexazecimal, se introduce rezultatul în locațiile de memorie corespunzătoare lui  $w_K$  și se trimite către DAF valoarea recepționată, convertită în format hexazecimal, pentru confirmare.

Subrutina acceptă de la tastatură numai codurile pentru cifrele zecimale și pentru comenzile RETURN (cod ASCII = 13) și CLEAR (cod ASCII = 24). Celelalte coduri recepționate sînt ignorate.

#### 5.4.3. Experimentarea SRA-VP

Experimental, SRA-VP a avut o comportare satisfăcătoare în buclă închisă. S-a observat o bună compensare a perturbațiilor: modificînd, manual, poziția de echilibru a axului motorului cu fracțiuni de tură, respectiv una sau mai multe ture, acesta revine în poziția impusă de comandă.

Răspunsul SRA, la o variație treaptă a mărimii prescrise este aperiodic, confirmînd calculele de proiectare (fig.5.36.a.). Măsurătorile s-au efectuat ca și în cazul sistemului de reglare a vitezei, prin mijloace software, tabelînd în memoria micro-calculatorului, răspunsul sistemului în fiecare perioadă de eşantionare.

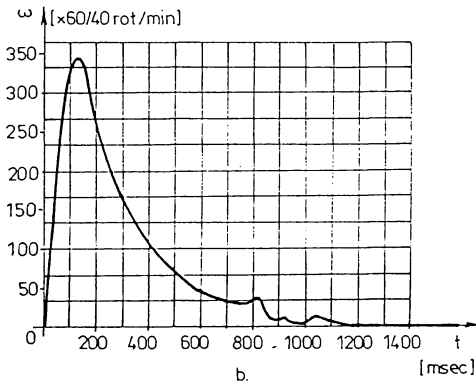
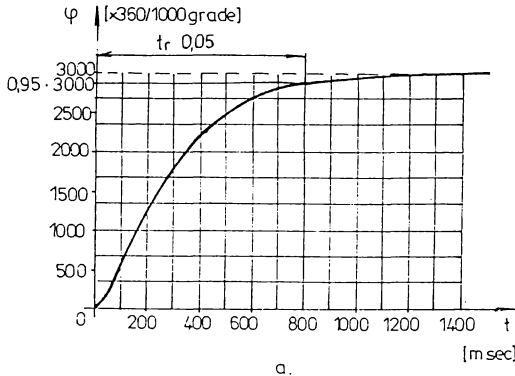


Figura 5.36. Răspunsul sistemului de reglare a vitezei și poziției la o variație treaptă a mărimii prescrise: a - poziția; b - viteza

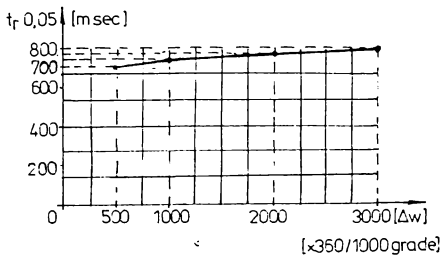


Figura 5.37. Dependența timpului de reglare obținut experimental de valoarea treptei de variație a mărimii prescrise

În fig.5.37. este prezentată dependența timpului de reglare obținut, în funcție de mărimea treptei de variație a mărimii prescrise. Nu s-au constatat diferențe în cazul aplicării unor trepte de variație cu polarități diferite.

### 5.5. Controlul curentului din înfășurarea rotorică a motorului

În cele două sisteme de acționare, prezentate în paragrafele anterioare din capitolul 5., se utilizează comanda în tensiune a motorului. Un sistem de acționare presupune, însă, și controlul valorii curentului din înfășurarea de comandă a motorului, cu limitarea acestuia în regimuri staționare. Un astfel de control este important în primul rând pentru protejarea dispozitivelor electronice de putere, din elementul de execuție, și a înfășurării, dar asigură și o protecție împotriva eventualelor suprasolicități în curent pentru sursa de alimentare. În sistemele de acționare tradiționale /86/ se include, supli-

mentar, o buclă de reglare automată a curentului alcătuită dintr-un traductor și un regulator de curent.

În ideea de a exclude complet utilizarea traductoarelor de curent și, implicit, a circuitelor de izolare galvanică, respectiv a convertoarelor numeric analogice, a fost pus la punct un procedeu original de limitare a curentului prin înfășurarea rotorică a motorului, utilizând exclusiv mijloace software.

Procedeu, aplicat în sistemele de acționare realizate, asigură controlul valorii curentului  $i_i$  din indus prin determinarea unei plaje:

$$\Delta U(\Omega) = [U_{\min}(\Omega), U_{\max}(\Omega)] \quad (5.78)$$

de valori admise pentru tensiunea  $u$  aplicată la bornele indusului, dependentă de viteza unghiulară  $\Omega$  a motorului.

Prin asigurarea menținerii valorilor tensiunii în domeniul (5.78), corespunzător relației:

$$i_i = \frac{u - K\Omega}{R_i} \quad (5.79)$$

dintre curentul din înfășurarea rotorică și tensiunea la bornele motorului, valabilă în regim staționar, rezultă:

$$i_i \in [I_{i\min}(\Omega), I_{i\max}(\Omega)] = I_i(\Omega) \quad (5.80)$$

cu

$$I_{i\min} = \frac{U_{\min}(\Omega) - K\Omega}{R_i} \quad (5.81)$$

$$I_{i\max} = \frac{U_{\max}(\Omega) - K\Omega}{R_i} \quad (5.82)$$

Notațiile avînd semnificațiile:  $U_{\min}$ ,  $U_{\max}$  marginile, inferioară, respectiv superioară, pentru plaja de valori  $\Delta U$ ,  $R_i$  - rezistența indusului,  $K$  - constanta motorului care apare în expresia tensiunii induse la cîmp de excitație constant,  $I_{i\min}$ ,  $I_{i\max}$  - marginile, inferioară, respectiv superioară, ale plajei de valori admise pentru curent în regim staționar.

Din ecuația (5.79) se poate deduce dependența tensiunii la bornele motorului de turația rotorului, în condițiile impunerii de valori limită pentru curentul  $i_i$ . Impunînd acestuia o valoare maximă  $I_{MAX}$  - și, respectiv, o valoare minimă  $-I_{MAX}$  (corespunzătoare unui regim de frînare în contracurent) - se obține o dependență de forma:

$$u_{\max}(\Omega) = R_i I_{\text{MAX}} + K\Omega \quad (5.83)$$

și

$$u_{\min}(\Omega) = -R_i I_{\text{MAX}} + K\Omega \quad (5.84)$$

Cele două ecuații delimitează valorile maximă și minimă pentru tensiunea la bornele motorului, în funcție de turație, și se pot reprezenta grafic ca în fig.5.38.

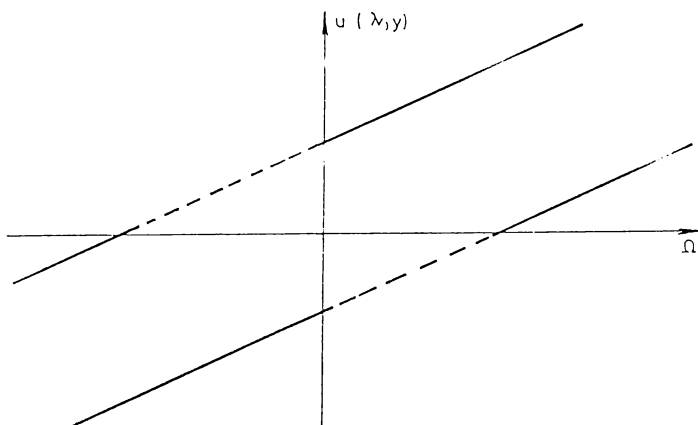


Figura 5.38. Graficul dependenței tensiunii la bornele motorului de turație, în regim staționar, la valorile limită impuse curentului rotoric

În cazul folosirii unui chopper cu tranzistoare cu funcționare în patru cadrane și prevăzut cu circuite de supresare, de tipul din fig.5.18., semnificație practică au numai porțiunile de grafic din cadranele I și III din fig.5.38. Chopperul de tipul menționat admite conducție de curent numai în același sens cu cel al tensiunii de comandă. În cazul când sarcina trece în regim de generator, tinzînd să forțeze curent în sens opus comenzii, circuitul de supresare pune motorul în scurtcircuit. Pentru cazul chopperului cu funcționare într-un singur cadran (cu tiristoare - fig.4.14.) nu interesează decît porțiunea de grafic din cadranul I din fig.5.38.

Limitarea valorii curentului rotoric în domeniul  $[-I_{\text{MAX}}, I_{\text{MAX}}]$  presupune, așadar, menținerea tensiunii de comandă la bornele motorului, în funcție de turație, între limitele impuse de cele două drepte reprezentate în graficul din fig.5.38.

Avînd în vedere relația (4.23) de dependență a tensiunii

de la ieșirea chopperului, aplicată motorului, de tensiunea de alimentare  $U_1$ :

$$U_m = \frac{T_C}{T} U_1 \quad (4.23)$$

și notînd

$$\lambda = \frac{T_C}{T} \quad (5.85)$$

unde  $T_C$  este durata de conducție, iar  $T$  este perioada de repetiție a unui ciclu de comenzi aplicate chopperului (egal cu perioada de eșantionare  $T = 10 \text{ ms}$ ), se constată că dependența  $\lambda(\Omega)$ , în condițiile limitării curentului rotoric, are aceeași alură:

$$\lambda_{\max}(\Omega) = \frac{R_i I_{\text{MAX}}}{U_1} + \frac{K\Omega}{U_1} \quad (5.86)$$

$$\lambda_{\min}(\Omega) = \frac{R_i I_{\text{MAX}}}{U_1} + \frac{K\Omega}{U_1} \quad (5.87)$$

întrucît  $u = U_m$ .

În acest fel graficul din figura 5.38. reprezintă și dreptele limitative pentru valorile lui  $\lambda$ .

După cum s-a arătat în paragrafele precedente (vezi și figurile 5.5. și 5.6.) mărimea de comandă  $y$ , calculată cu algoritmele de reglare, este o măsură a duratei  $T_C$  de conducție impusă chopperului pentru a se asigura reglarea corespunzătoare. Conversia număr-timp ( $y \rightarrow T_C$ ) este realizată fizic de timerul din structura ECN:

$$y = \frac{T_C}{325,52 \cdot 10^{-9}} \quad (5.88)$$

întrucît tactul sistemului este, după cum s-a arătat, de  $325,52 \cdot 10^{-9} \text{ s}$ .

Avînd în vedere perioada de eșantionare aleasă  $T = 10 \text{ ms}$

$$\lambda = \frac{T_C}{10 \cdot 10^{-3}} \quad (5.89)$$

și consecința imediată

$$y = 3,072 \cdot 10^4 \quad \lambda = \varepsilon \lambda \quad (5.90)$$

Așadar, între variabila  $\lambda$  și mărimea de comandă  $y$  relația este de directă proporționalitate, iar graficul  $y(\Omega)$  are, dacă valorile curentului se limitează, de asemenea alura din fig.5.38.

Concluzia ce se desprinde din succesiunea de raționamente de mai sus este rezumabilă, așadar, astfel: limitarea valorilor curentului rotoric în domeniul  $[-I_{MAX}, I_{MAX}]$  se realizează prin menținerea tensiunii la bornele motorului între limitele reprezentate de dreptele de pe graficul din fig.5.38. Intrucît

$$u = U_m = \lambda U_1 \quad (5.91)$$

limitarea curentului rotoric este asigurată dacă  $\lambda$  se menține într-o plajă de valori situată între două drepte de alura celor prezentate în graficul 5.38. În fine, avînd în vedere (5.90), dacă  $y$  este limitată la valori cuprinse între două drepte ca cele de pe graficul 5.38., se realizează implicit și limitarea valorii curentului din înfășurarea rotorică a motorului.

În consecință, dacă sistemele de reglare, descrise în paragrafele anterioare ale capitolului 5 se completează cu un bloc limitator pentru valorile mărimii de comandă  $y$  și se asigură pentru valorile limitative o dependență de turație de forma reprezentată grafic în fig.5.38., se realizează implicit limitarea curentului din înfășurarea rotorică în regim staționar.

Schemele bloc pentru cele două sisteme de acționare completate cu blocurile ce asigură limitarea curentului rotoric sînt cele din figurile 5.39 și 5.40.

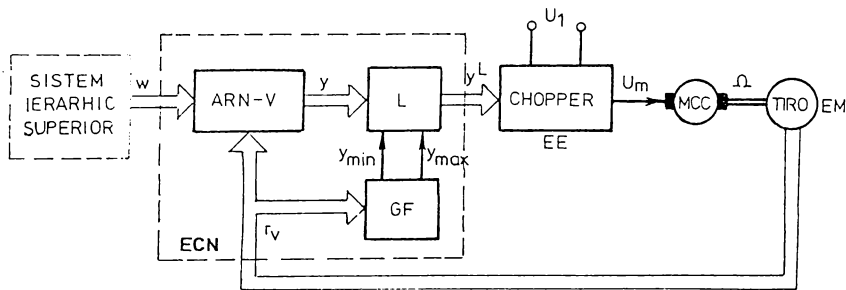


Figura 5.39. Structura sistemului de reglare automată numerică a vitezei, cu limitarea curentului rotoric în regim staționar

După cum este arătat în schemele bloc, limitarea pentru valorile mărimii de comandă se realizează prin mijloace exclusiv software la nivelul ECN, pe care sînt implementate și algoritmele de reglare numerică.



În esență, procedeul utilizat, include un program generator de funcții (GF), care generează, în funcție de turație, valori limită pentru  $y$  în concordanță cu graficul 5.38., și un bloc de comparație (limitator - L) a valorilor pentru  $y$  - calculate cu algoritmele de reglare - cu valorile extreme generate de GF.

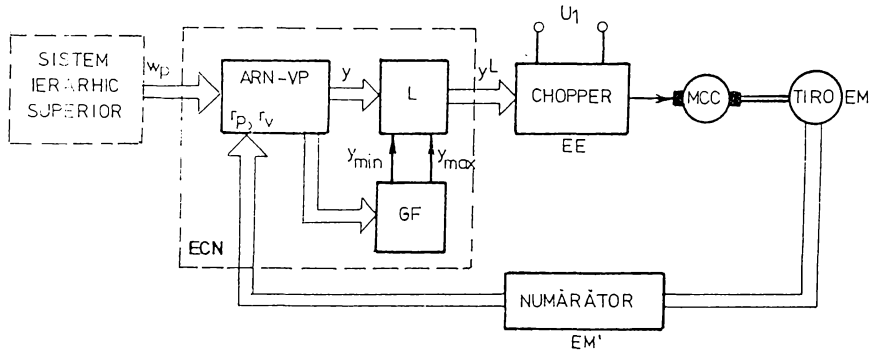


Figura 5.40. Structura sistemului de reglare automată numerică a vitezei și poziției, cu limitarea curentului rotorului în regim staționar

Dacă valorile lui  $y$  rezultate din calculele efectuate conform ARN-V sau, respectiv, ARN-VP, cu relațiile (5.66), respectiv (5.76), se încadrează între limitele  $y_{\min}$ ,  $y_{\max}$  calculate de generatorul de funcții pentru o valoare momentană  $r_v$  a turației, chopperul se comandă cu mărimea  $y$  rezultată din algoritmul de reglare. Dacă rezultatul calculului efectuat conform algoritmului nu se încadrează între limitele calculate de GF, chopperul este comandat cu mărimea  $y_{\min}$  sau  $y_{\max}$ , după caz.

Relațiile cu care se calculează valorile  $y_{\min}$  și  $y_{\max}$  deduc pornind de la (5.83) și (5.84):

$$u_{\max}(\Omega) = R_i I_{\max} + K\Omega \quad (5.83)$$

$$u_{\min}(\Omega) = -R_i I_{\max} + K\Omega \quad (5.84)$$

Tinînd seama de (5.86), (5.87) și (5.90) se obține:

$$y_{\max}(\Omega) = \frac{\xi R_i I_{\max}}{U_1} + \frac{\xi K}{U_1} \Omega \quad (5.92)$$

și

$$y_{\min}(\Omega) = - \frac{\varepsilon R_i I_{MAX}}{U_1} + \frac{\varepsilon K}{U_1} \Omega \quad (5.93)$$

Mijloacele de reglare utilizate sînt exclusiv numerice. Viteza unghiulară este măsurată direct în cazul ARN-V și calculabilă din măsura poziției în cazul ARN-VP. Procedeele de măsură utilizat este acela de numărare a impulsurilor generate de traductorul incremental în intervale de timp strict constante  $T_{m\ddot{a}s}$ . În cazul ARN-V mărimea de reacție  $r_k$ , dintr-un pas de eșantionare, este un număr ce depinde de viteza de rotație a traductorului TIRO, legat solidar cu axul motorului într-un raport de transmisie 1:1.

După cum s-a arătat în paragraful 5.3.3.2., între numărul  $r_k$  și frecvența de rotație a axului motorului există următoarea legătură:

$$r_k = 2N f T_{m\ddot{a}s} \quad (5.67)$$

unde  $N$  este numărul de impulsuri generate de TIRO la o tură completă.

Intrucît la aplicația concretă  $N = 1000$  și  $T_{m\ddot{a}s} = 5 \cdot 10^{-3}s$

$$f [Hz] = \frac{r_k}{2N T_{m\ddot{a}s}} = \frac{r_k}{10} \quad (5.68)$$

Așadar viteza unghiulară a axului motorului este calculabilă cu relația:

$$\Omega = 2\pi f = \frac{\pi r_k}{N T_{m\ddot{a}s}} = \frac{\pi}{5} r_k \quad (5.94)$$

Expresia (5.94) pentru  $\Omega$ , înlocuită în ecuațiile (5.92) și (5.93) conduce la determinarea dependenței mărimii de comandă discrete  $y_k$  de mărimea de reacție  $r_k$  astfel:

$$y_k \max(r_k) = \frac{\varepsilon R_i I_{MAX}}{U_1} + \frac{\varepsilon K}{U_1} \cdot \frac{\pi}{N T_{m\ddot{a}s}} r_k \quad (5.95)$$

$$y_k \min(r_k) = - \frac{\varepsilon R_i I_{MAX}}{U_1} + \frac{\varepsilon K}{U_1} \cdot \frac{\pi}{N T_{m\ddot{a}s}} r_k \quad (5.96)$$

După cum s-a arătat în paragrafele anterioare mărimea de comandă  $y_k$  se calculează ca un număr reprezentat pe 14 biți. Operarea, în porțiunea de program ce asigură limitarea, cu un număr reprezentat pe doi octeți ar conduce la un consum de timp de calcul suplimentar și inutil din punct de vedere practic. Este suficient ca în calcule și în comparații să se opereze numai cu

octetul cel mai semnificativ  $y_{kH}$ , al numărului ce reprezintă  $y_k$ , întrucît conținutul celui de al doilea octet este nesemnificativ ca influență asupra ordinului de mărime al valorii lui  $y_k$ .

Relațiile de calcul pentru  $y_{kHmax}$  și  $y_{kHmin}$  se obțin din (5.95) și (5.96) prin împărțirea acestor ecuații cu  $2^8 = 256$ , consecința operării exclusive cu octetul cel mai semnificativ:

$$y_{kHmax}(r_k) = \frac{\varepsilon R_i I_{MAX}}{256 \cdot U_1} + \frac{\varepsilon K}{256 \cdot U_1} \cdot \frac{\pi}{N T_{m\acute{a}s}} r_k \quad (5.97)$$

$$y_{kHmin}(r_k) = -\frac{\varepsilon R_i I_{MAX}}{256 \cdot U_1} + \frac{\varepsilon K}{256 \cdot U_1} \cdot \frac{\pi}{N T_{m\acute{a}s}} r_k \quad (5.98)$$

Cu relațiile (5.97) și (5.98) se calculează valorile extreme pentru octetul semnificativ  $y_{kH}$  al mărimii de comandă în funcție de o anumite valoare a reacției  $r_k$ , extreme impuse de limitarea valorii curentului din înfășurarea rotorică.

Pentru motorul concret utilizat, cu datele de catalog prezentate în paragraful 5.2.4., și admitînd  $I_{MAX} = 1$  A ecuațiile de mai sus devin:

$$y_{kHmax}(r_k) = 9 + \frac{5}{16} r_k \quad (5.99)$$

$$y_{kHmin}(r_k) = -9 + \frac{5}{16} r_k \quad (5.100)$$

Procedeu prezentat nu este limitat la o formă a caracteristicilor ca cea din fig.5.38 pentru dependența  $y(\Omega)$ ,  $u(\Omega)$ ,  $\lambda(\Omega)$ . Aceasta poate avea și o alură în trepte ca în fig.5.41, alură ce este mai ușor de sintetizat prin mijloace software.

Avantajul utilizării unui generator de funcții cu variația valorilor în trepte constă în reducerea semnificativă a timpului necesar efectuării calculelor conform relațiilor (5.99) și (5.100). Astfel, valorile pentru  $r_k$  se prelucrează grupat în trepte de domenii de valori de câte 16 unități (10 H), prin neglijaarea ultimilor 4 biți, ceea ce permite apoi efectuarea împărțirii cu 16, necesară, prin simple rotații ale conținutului acumulatorului. Se obțin în acest mod trepte de valori pentru  $y_{kHmax}$  și  $y_{kHmin}$ .

**Organigrama** porțiunii de program ce realizează calculele (5.99) și (5.100), compară  $y_{kH}$  calculat de algoritmul de reglare cu valorile limită impuse de nedepășirea curentului maxim admis,

și asigură limitarea cînd aceasta e necesară, este cea din fig. 5.42.

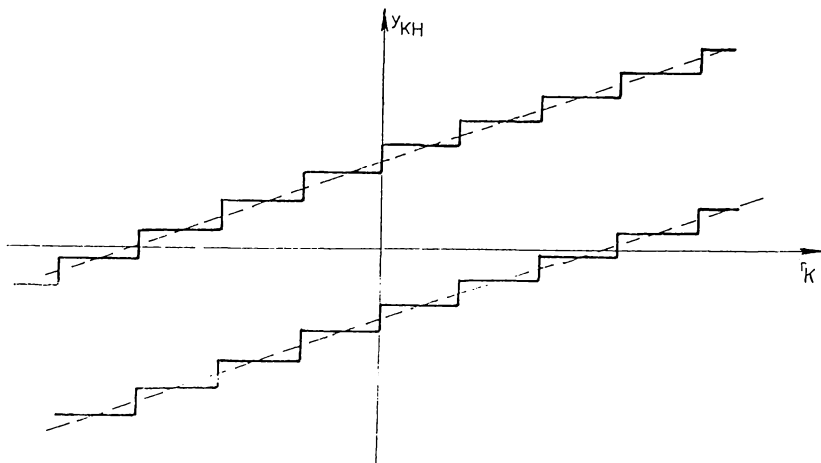


Figura 5.41. Graficul dependenței mărimii de comandă de turație, în regim staționar, cu limitare în trepte impusă de valorile extreme ale curentului rotoric

Programul detaliat este după cum urmează:

```
LDA RK                MOV A,B
ANI OFOH              ADI 9
RRC                  CMP C
RRC                  JNC ET1
MOV B,A              STA YKH
RRC                  JMP CONTINUARE
RRC                  ET1 SUB 18
ADD B                JC CONTINUARE
MOV B,A              CMP C
LDA YKH              JC CONTINUARE
MOV C,A              STA YKH
CONTINUARE ...
```

Porțiunea de program prezentată se include în programul principal înainte de testarea regimurilor de funcționare (vezi organigramele 5.18. și 5.30.).

După cum se observă programul operează numai cu mărimi aparținînd cadranelui I din graficul reprezentat în fig.5.41. După cum s-a arătat anterior funcționarea în cadranele II și IV

! nu are, însă, corespondent practic. La funcționarea chopperului

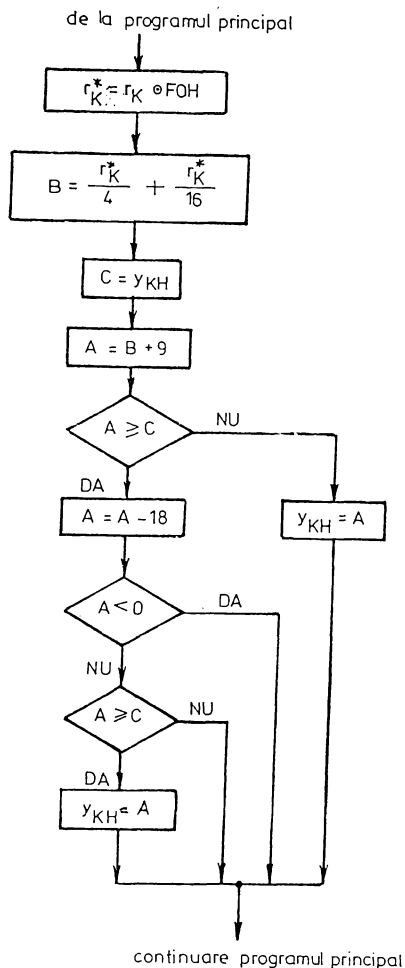


Figura 5.42. Organigrama porțiunii de program pentru controlul curentului rotoric

în cadranul III, mărimile  $y_k$  și  $r_k$  sînt tratate în modul, iar semnul este memorat separat (vezi 5.3.3.3.b). În consecință programul descris anterior se poate utiliza în aceeași variantă și în cazul funcționării în cadranul III.

În cazul algoritmului de reglare a vitezei și poziției, mărimea de reacție este poziția curentă a axului motorului (ci-

tită de la numărător la începutul fiecărei perioade de eşantionare - vezi 5.4.2.) și

$$r_{P_k} = \frac{\theta_k}{2\pi} \cdot N \quad (5.101)$$

unde  $\theta_k$  este poziția unghiulară a rotorului în pasul de eşantionare  $k$ , iar  $N$  numărul de impulsuri generate de TIRO la o rotație completă.

Analog

$$r_{P_{k-1}} = \frac{\theta_{k-1}}{2\pi} \cdot N \quad (5.102)$$

Deplasarea motorului pe durata perioadei de eşantionare  $k-1$ , de durată  $T$ , este:

$$r_{P_k} - r_{P_{k-1}} = \frac{N}{2\pi} (\theta_k - \theta_{k-1}) \quad (5.103)$$

sau

$$r_{P_k} - r_{P_{k-1}} = \frac{NT}{2\pi} \cdot \frac{\theta_k - \theta_{k-1}}{T} = \frac{NT}{2\pi} \Omega \quad (5.104)$$

Conform (5.94)

$$\Omega = \frac{r_k}{N T_{\text{măs}}} \quad (5.94)$$

Se obține în final pentru mărimea de reacție de viteză, necesară în controlul curentului din înfășurare, relația de calcul:

$$r_k = \frac{2T_{\text{măs}}}{T} (r_{P_k} - r_{P_{k-1}}) \quad (5.105)$$

După cum s-a arătat în paragraful 5.3.  $T_{\text{măs}} = \frac{T}{2}$ , adică:

$$r_k = r_{P_k} - r_{P_{k-1}} \quad (5.61)$$

S-a prezentat, astfel, și deducerea relației (5.61) din 5.2.5.

În consecință în cazul utilizării ARN-VP, înainte de rula-rea programului din fig.5.40. se procedează la efectuarea calculului (5.61) pentru determinarea  $r_k$ .

Prin includerea programului prezentat în acest paragraf în sistemele de reglare numerică descrise în capitolul 5, se realizează, cu mijloace exclusiv software, în afara funcțiilor de reglare a vitezei și/sau poziției și controlul curentului din înfășurarea rotorică a motorului.

## CAPITOLUL 6.

### SISTEM DE REGLARE NUMERICA A VITEZEI PENTRU TREI MOTOARE DE CURENT CONTINUU CU UN SINGUR MICROPROCESOR DE COMANDA

#### 6.1. Structura sistemului de reglare automată

Realizările descrise în capitolul 5 au fost continuate și dezvoltate în continuare /9 /, /18/. În capitolul de față se prezintă un sistem conceput și realizat pentru a asigura comanda și controlul vitezei de deplasare pentru trei sisteme de acționare complet independente între ele, deși circuitul lor de comandă este comun și constituit de un unic microsistem cu microprocesor de uz general pe 8 biți. Motoarele de acționare sînt de curent continuu și pot fi utilizate la acționarea a trei axe diferite ale unui robot industrial.

Si în această aplicație se presupune că un sistem, ierarhic superior, furnizează codurile vitezelor impuse, la un moment dat, de necesitățile tehnologice. Aceste coduri sînt preluate de unicul microsistem de comandă, care asigură reglajul și comanda, simultană și independentă, pentru toate cele trei motoare.

Structura sistemului de reglare numerică automată a vitezei este redată în fig.6.1. Acționările sînt reversibile prin folosirea chopperelor în patru cadrane, de același tip cu cel utilizat în aplicațiile din capitolul 5 (vezi și capitolul 4, fig.4.18.).

Notațiile de pe figură au semnificații similare cu cele din capitolul precedent:

$w_i$  - viteza prescrisă;

$r_i$  - viteza măsurată (reacția de viteză);

- $c_{1i}, c_{2i}$  - mărimi de comandă;  
 $U_{mi}$  - mărimea de execuție (tensiunea medie aplicată motorului);  
 $U_{ai}$  - tensiunea continuă de alimentare;  
 $\Omega_i$  - viteza unghiulară;  
 $MCC_i$  - motor de curent continuu;  
 $TIRO_i$  - traductor incremental de deplasare cu  $i = 1, 2, 3$ ;  
 $ECN$  - echipament de comandă numerică;  
 $EE$  - element de execuție (chopper);  
 $EM$  - element de măsură;  
 $P$  - procesul reglat.

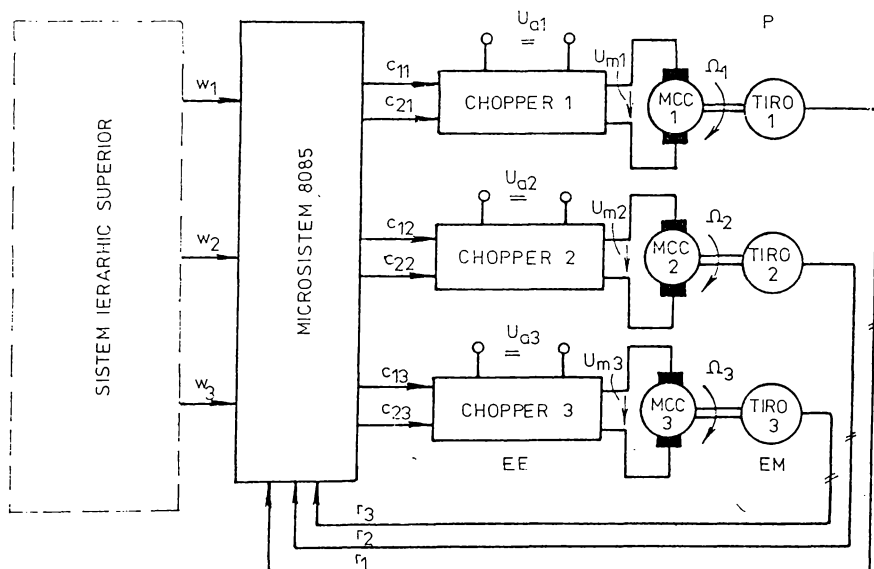


Figura 6.1. Schema bloc a celor trei sisteme de acționare comandate cu un singur microprocesor

În această schemă sînt prezentate cele trei sisteme de acționare cu motor de curent continuu. În sistemul realizat practic fiecare dintre motoare este cîte un motor de tipul EP 211 (IME Pitești), același cu cel folosit în aplicațiile din capitolul 5 (vezi paragraful 5.2.4.). Tensiunea medie aplicată indusului motorului este obținută sub forma unor impulsuri de tensiune generate de elementul de execuție (EE) de tip chopper în patru cadrane cu tranzistoare. Comanda fiecăruia dintre choppere este



realizată direct de către echipamentul de comandă numerică(ECN).

ECN este un microsistem cu microprocesorul 8085, același cu cel folosit în aplicațiile din capitolul 5, și descris în detaliu în capitolul 3, paragraful 3.2.2.. Un singur microsistem cu microprocesor uzual pe 8 biți asigură comanda și reglajul pentru toate cele trei sisteme de acționare /9 /.

Elementul de măsură EM, din fiecare sistem de acționare, este câte un traductor incremental de deplasare de tip TIRO(descriș în paragraful 5.1.3.3.). Echipamentul de comandă numeric asigură și funcția de măsurare a vitezei prin contorizarea simultană în intervale fixe de timp a impulsurilor furnizate de toate cele trei traductoare.

Echipamentul de comandă numerică generează semnalele de comandă  $c_{1i}$ ,  $c_{2i}$  ( $i = 1, 2, 3$ ), impulsuri TTL cu durata  $T_C$  de conducție impusă la un moment dat fiecăruia dintre chopere,  $c_{1i}$  corespunzând unui sens de rotație iar  $c_{2i}$  sensului contrar. Se asigură izolare galvanică prin optocuploare între ECN și chopere.

Timpul de conducție  $T_C$  reprezintă, din punct de vedere informațional, mărimea de comandă a fiecăruia dintre cele trei sisteme de acționare independente conduse de același microcalculator.

Mărimile de prescriere  $w_i$  ( $i = 1, 2, 3$ ) sînt furnizate ECN de către un sistem ierarhic superior care poate fi în structura unui robot industrial calculatorul pentru modelarea mediului ori calculatorul pentru specificarea mișcărilor. In cazul aplicației concret realizate mărimile de prescriere sînt furnizate de un operator uman prin tastatura consolei echipamentului de calcul.

## 6.2. Ecuția discretă corespunzătoare algoritmului de reglare a vitezei

Algoritmemele utilizate pentru reglarea vitezei celor trei motoare sînt aceleași cu cele proiectate în capitolul 5 (5.2.4.) cu forma discretă:

$$y_K = d_0 a_K + d_1 a_{K-1} + c_1 y_{K-1} \quad (5.49)$$

în care:  $y_K$ ,  $y_{K-1}$  reprezintă mărimea de comandă exprimată în secunde, avînd semnificația de interval de conducție pentru chopper

în perioada de eşantionare curentă respectiv anterioară,  $a_K$ ,  $a_{K-1}$  reprezintă codul numeric al mărimii de acţionare (diferenţa dintre mărimea prescrisă -  $w$  şi cea de reacţie -  $r$ ) în perioada de eşantionare curentă, respectiv anterioară.

Coeficienţii  $d_0$ ,  $d_1$ ,  $c_1$  sînt cei calculaţi în (5.48) rezultînd pentru (5.49) forma:

$$y_K = y_{K-1} + 1,22 \cdot 10^{-4} a_K - 1,14 \cdot 10^{-4} a_{K-1} \quad (6.1)$$

După cum s-a arătat în 5.3.1., implementarea corectă a algoritmului (5.49), presupune exprimarea mărimilor de comandă  $y_K$ ,  $y_{K-1}$  în cod numeric normat. Acest cod reprezintă un număr pe 14 biţi (capacitatea timerului din microsistem) şi care este decrementat pînă la zero în timpul funcţionării pentru a se genera intervalul de timp  $T_C$  necesar.

Tinînd cont că decrementarea timerului se efectuează cu impulsuri de durată  $325,52 \cdot 10^{-9}$  sec. (durata de tact din microsistem) normarea se asigură astfel:

$$y_K = \frac{y_K [\text{sec}]}{325,52 \cdot 10^{-9}} \quad (6.2)$$

Rezultă în consecinţă forma finală (5.66) pentru algoritmul de reglare numerică a vitezei:

$$y_K = y_{K-1} + 375(w_K - r_K) - 350(w_{K-1} - r_{K-1}) \quad (5.66)$$

Acest algoritm este calculat de trei ori într-o perioadă de eşantionare, pentru fiecare din cele trei motoare cîte o dată.

### 6.3. Funcţiile îndeplinite de microsistem

Si în această aplicaţie se foloseşte sistemul de întreruperi, specific modului de lucru în timp real.

Evenimentele din procesul condus de microsistem se succed la diferite intervale de timp care sînt contorizate cu timerul (8155) din sistem. La sfîrşitul fiecărui interval de timp contorizat, timerul generează un semnal de întrerupere către microprocesor. Acesta recepţionează şi înţelege semnificaţia întreprerii şi generează comenzile corespunzătoare în momentele potrivite.

Principalele funcţii pe care le îndeplineşte microsistemul sînt:

(F1) - preluarea mărimilor de prescriere - codurile numerice  $w_{Ki}$

( $i = 1, 2, 3$ ) - pentru fiecare motor, de la sistemul ierarhic superior. Dialogul este inițializat printr-un nivel de întrerupere hardware, diferit de cel ce corespunde timerului, conectat la sistemul ierarhic superior care semnalează transmiterea codurilor respective;

- (F2) - măsurarea vitezei curente  $r_{Ki}$  ( $i = 1, 2, 3$ ) pentru cele trei motoare simultan - cu un subprogram de numărare simultană a impulsurilor generate de traductoarele incrementale, timp de o jumătate de perioadă de eșantionare;
- (F3) - calcularea codului numeric normat  $y_{Ki}$  ( $i = 1, 2, 3$ ) al mărimii de comandă  $y_{Ki}$  conform algoritmului (5.66);
- (F4) - conversia codurilor normate  $y_{Ki}$  în durate de conducție pentru choppere. Conversia se realizează prin încărcarea în timer a codului  $y_{Ki}$  și decrementarea conținutului acestuia pînă la zero cu frecvența  $f_0 = 3,072$  MHz de tact a sistemului;

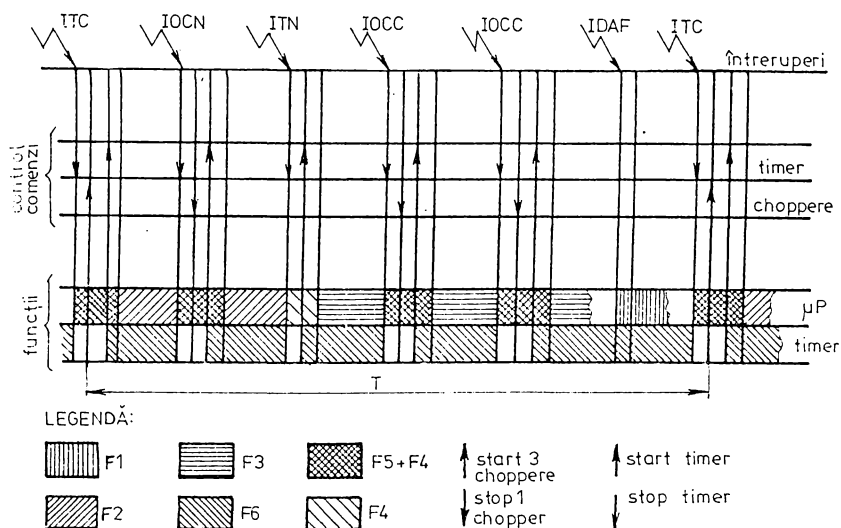


Figura 6.2. Eșalonarea în timpul unei perioade de eșantionare a funcțiilor executate de microprocesor

- (F5) - generarea impulsurilor de comandă pentru intrarea în conducție și blocarea tranzistoarelor chopperului. Aceste impulsuri, generate la un port de ieșire al microsistemului asigură comanda directă, cu izolare galvanică corespunzătoare, a circuitului de putere. Se evită utilizarea conver-

toarelor numeric-analogice.

În vederea scurtării execuției, cele 5 funcții de bază se execută "întreșut". Pentru a se realiza acest mod de lucru microprocesorul îndeplinește și o funcție suplimentară:

(F6) - interpretarea și tratarea întreruperilor generate de timer sau gestiunea timpului real. Întrucât generarea tuturor intervalelor de timp se realizează cu un singur timer (soluția cea mai simplă din punct de vedere hardware), fiecare semnal de întrerupere generat de timer trebuie "înțeles" corect de microprocesor. În acest scop se utilizează subrutina de tratare a întreruperilor care decide ce anume funcție trebuie executată la sosirea unui semnal de întrerupere.

În fig.6.2. este reprezentat un exemplu de eșalonare a execuției diferitelor funcții pe parcursul unei perioade de eșantionare  $T$ .

#### 6.4. Gestiunea timpului real

Generarea tuturor intervalelor de timp: pentru conducția chopperelor, pentru activarea funcției de numărare a impulsurilor sosite de la TIR0 și pentru reluarea periodică a întregului algoritm de conducere, sînt generate de către timerul inclus în circuitul 8155 din sistem. Ieșirea TIMER OUT a acestuia este conectată hardware la pinul corespunzător nivelului de întrerupere RST 7.5 al microprocesorului.

În timer se încarcă valoarea numerică  $t/f_0$  corespunzătoare unui interval de timp ce trebuie contorizat și se decrementează pînă la zero, cu frecvența  $f_0$  (de tact a sistemului). La sfîrșitul decrementării timerul generează semnalul de întrerupere către microprocesor. Acesta produce un alt salt în program la subrutina de tratare a întreruperilor.

În cadrul acestei subrutine trebuie să se decidă care anume funcție urmează să fie executată. Analizînd toate cazurile posibile, se constată că există patru tipuri de întrerupere. Se atașează fiecărui tip de întrerupere un cod reprezentat pe 2 biți după cum urmează:

IOCN - cod 00 - marchează momentele în care trebuie întreruptă numărarea impulsurilor de la traductoare pentru a se genera comenzi de blocare a chopperelor;

- IOCC - cod 10 - marchează momentele în care se întrerupe efectuarea calculelor, conforme algoritmului (5.66), pentru a se genera comenzi de blocare a chopperelor;
- ITN - cod 01 - marchează epuizarea duratei afectate numărării impulsurilor de la traductoare;
- ITC - cod 11 - marchează epuizarea unei perioade de eşantionare. Pentru explicitate în prezentare perioada de eşantionare alocată execuției întregului algoritm de conducere este denumită "cuantă";
- ITC survine după  $T = 10 \text{ ms}$  de la începerea cuantei curen-te;
- ITN survine după  $T/2 = 5 \text{ ms}$  (exact). Semnale de întrerupere pentru blocarea chopperelor sînt generate după intervale de timp  $y_{Ki}$  și de aceea ele sînt de tip IOCN dacă  $y_{Ki} < T/2$  și de tip IOCC dacă  $y_{Ki} > T/2$ .

În fig.6.2. au fost reprezentate fiecare dintre aceste tipuri de întreruperi pentru un exemplu arbitrar și s-au arătat comenzile generate de microprocesor către choppere și timer. Dialogul cu nivelul ierarhic superior (consola operatorului uman în cazul aplicației concrete) este deservit la sfîrșitul perioadei de eşantionare cînd toate celelalte funcții au fost executate.

Schema logică de principiu a programului principal este prezentată în fig.6.3.

Cele trei choppere sînt direct comandate de liniile portului 23 H din microsistem. Legătura, izolată galvanic cu optocuploare, între liniile portului 23 H și tranzistoarele chopperelor este ilustrată în fig.6.4.

Comanda de pornire se realizează prin înscrierea unui cuvînt de 8 biți în registrul de comandă al acestui port. Cuvîntul înscris are poziționați pe 1 biții ce corespund tranzistoarelor  $T_{1i}$  ( $i = 1, 2, 3$ ) pentru realizarea unui sens de rotație dorit sau biții ce corespund tranzistoarelor  $T_{2i}$  ( $i = 1, 2, 3$ ) pentru realizarea sensului de rotație contrar pentru motorul  $i$ . Octetul de comandă a pornirii chopperului - un vector de pornire - este stabilit de către subrutina de calcul și depozitat la locația VPRN de unde programul de gestiune a timpului real îl preia și îl înscrie în registrul portului 23 H.

Comanda de blocare a unuia sau mai multor choppere este precedată de o operație logică "SI" între conținutul registrului

portului 23 H înscris la pornire și un octet mască de oprire. Octetul de mascare conține valori 0 pe toate pozițiile corespunzătoare liniilor conectate la  $T_{1i}$  și  $T_{2i}$  pentru realizarea blocării chopperului de pe axa  $i$  și 1 pe celelalte poziții. Rezultatul operației "SI" este depus apoi în registrul portului 23 H.

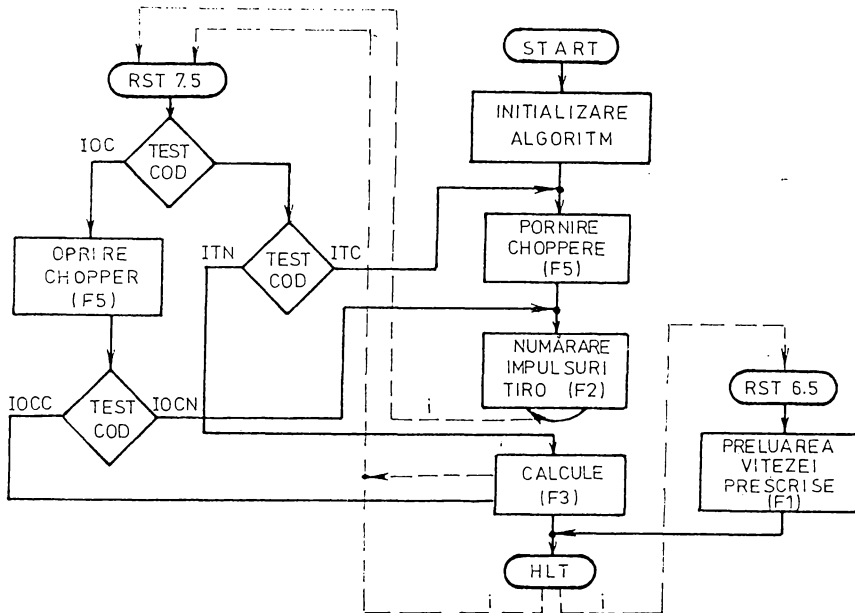


Figura 6.3. Organigrama principală a programului principal

Se realizează oprirea chopperului după necesități, indiferent de sensul de rotație comandat anterior.

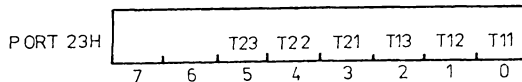


Figura 6.4. Modul de conectare a liniilor portului de ieșire 23 H la tranzistoarele celor trei choppere

Pentru cele trei axe sînt folosite cuvintele mască:

- axa 1 : MSKA = 11110110 = F6H
  - axa 2 : MSKB = 11101101 = EDH
  - axa 3 : MSKC = 11011011 = DBH
- (6.3)

Succesiunea întreruperilor se contabilizează cu un șir de evenimente, reprezentat în memorie cu un tabel cu structura din

fig.6.5., începînd de la adresa TAB I.

Subrutina de tratare a întreruperilor parcurge acest tabel o dată în fiecare cuantă, în concordanță cu cele 5 tipuri de întrerupere reprezentate în fig.6.2. După deservirea unei întreruperi se memorează la adresa ADRTAB poziția întreruperii următoare ce trebuie deservită, adică adresa din tabelul prezentat în fig.6.5. ce corespunde constantei ce urmează să fie încărcată în timer în următorul interval de temporizare.

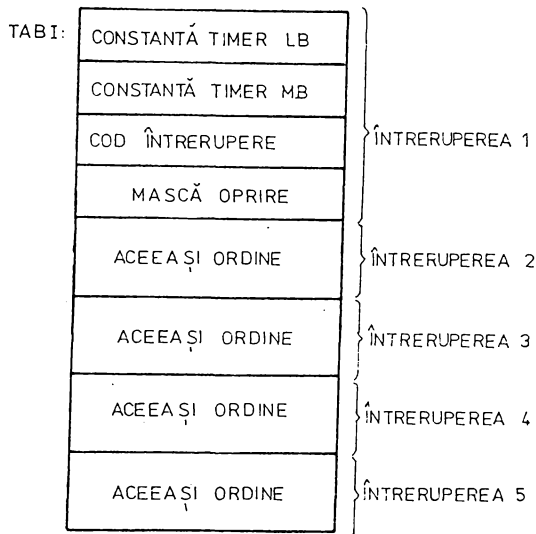


Figura 6.5. Tabelul din memorie cu înșiruirea evenimentelor

Subrutina de tratare a întreruperii execută următoarele operații:

- adună patru la ADRTAB, pregătind astfel deservirea întreruperii următoare;
- încarcă în timer constanta intervalului de timp ce trebuie contorizat în continuare (de la adresa ADRTAB nou calculată);
- aduce din tabel codul întreruperii curente (pentru care execută o revenire în tabel cu 3 poziții);
- analizează codul întreruperii prin testarea bitului mai puțin semnificativ ( $b_0$ ) dintre cei doi ai codului, astfel:
  - α) dacă  $b_0 = 0$ , întreruperea este de tip IOCC sau IOCN și:
    - aduce din tabel masca de oprire,

- comandă oprirea unui chopper prin mascarea codului înscris în registrul portului 23 H,
  - testează al doilea bit din codul întreruperii ( $b_1$ ):
    - dacă  $b_1 = 0$  întreruperea este de tip IOCN; în acest caz comandă pornirea timerului și revine la subrutina de numărare prin instrucțiunea JMP,
    - dacă  $b_1 = 1$  întreruperea este de tip IOCC; în acest caz comandă pornirea timerului, reface registrele interne ale microprocesorului și revine la subrutina de calcul prin instrucțiunea RET;
- β) dacă  $b_0 = 1$ , întreruperea este de tip ITN sau ITC și:
- testează al doilea bit din codul întreruperii ( $b_1$ ):
    - dacă  $b_1 = 1$  întreruperea este de tip ITC, încarcă timerul cu constanta corespunzătoare primului interval de temporizare din următoarea cuantă (de la adresele TABI și TABI+1) și reia algoritmul de conducere prin salt la adresa INIT; pornirea timerului nu se execută aici ci după pornirea chopperului în cadrul programului principal.

Execuția corectă a calculelor presupune, în momentul în care survine o întrerupere IOCC, salvarea în stivă a registrelor microprocesorului. Subrutina de tratare a întreruperilor este, de aceea, prevăzută la început cu instrucțiuni de salvare în stivă a registrelor microprocesorului. Dacă întreruperea este de tip IOCC, după deservirea acesteia, se readuc întâi din stivă informațiile salvate și se revine la calcule cu instrucțiunea RET. În cazul celorlalte tipuri de întreruperi, conținutul stivei nu interesează și de aceea se execută de șase ori incrementarea INX SP și se pierde informația depusă în stivă, iar revenirea din întrerupere se efectuează cu salt necondiționat (JMP).

În programul principal se disting următoarele blocuri:

- o rutină care asigură execuția corectă a primei cuante, la inițializarea algoritmului de conducere,
- o secvență de instrucțiuni ce se execută în fiecare cuantă pentru inițializările contoarelor subrutinei de numărare (registrele B, C, D) și pentru ADRTAB, urmate de comenzi pentru pornirea chopperelor, conform vectorului VPRN, și pornirea timerului.

Rutina de inițializare construiește un tabel cu structu-



ra din fig.6.6., inițializează ADRTAB cu valoarea 4A H, corespunzătoare locației TABI+12 și încarcă în timer valoarea 3COOH ce corespunde unui interval de timp  $T/2 = 5$  ms.

TABI+12	00	CONSTANTĂ LB	} ÎNTRERUPEREA 4
	00	TIMER HB	
	01	COD ÎNTRERUPERE	
	00	MASCĂ OPRIRE	
	00	CONSTANTĂ LB	} ÎNTRERUPEREA 5
	3C	TIMER HB	
	03	COD ÎNTRERUPERE	
	00	MASCĂ OPRIRE	

Figura 6.6. Tabelul din memorie utilizat de rutina de inițializare

Cînd algoritmul de conducere se rulează pentru întîia oară se execută următoarele etape:

- numărarea impulsurilor de la traductoare exact 5 ms,
- întreruperea ITN,
- întreruperea ITC după 5 ms de la generarea ITN.

Se trece apoi la executarea propriu-zisă a programului suplimentar, rutina inițializează cu valoarea zero zona de date corespunzătoare variabilelor folosite de ARN și contorul CCR folosit pentru dialogul cu consola operatorului.

În fig.6.7., se prezintă schema bloc detaliată a programului principal în care sînt evidențiate toate subprogramele, inițializările, operațiile cu stiva, validările întreruperilor și comenzile pentru choppere și timer.

La deservirea întreruperii RST 7.5 există o secțiune critică ce nu poate fi întreruptă de alte semnale de întrerupere și anume secvența de instrucțiuni de inițializare, executată pe fiecare cuantă și subrutinele propriu-zise de tratare a întreruperilor.

Pentru întreruperea RST 6.5, secțiunea critică include și subrutina de numărare a impulsurilor de la traductoare ce nu

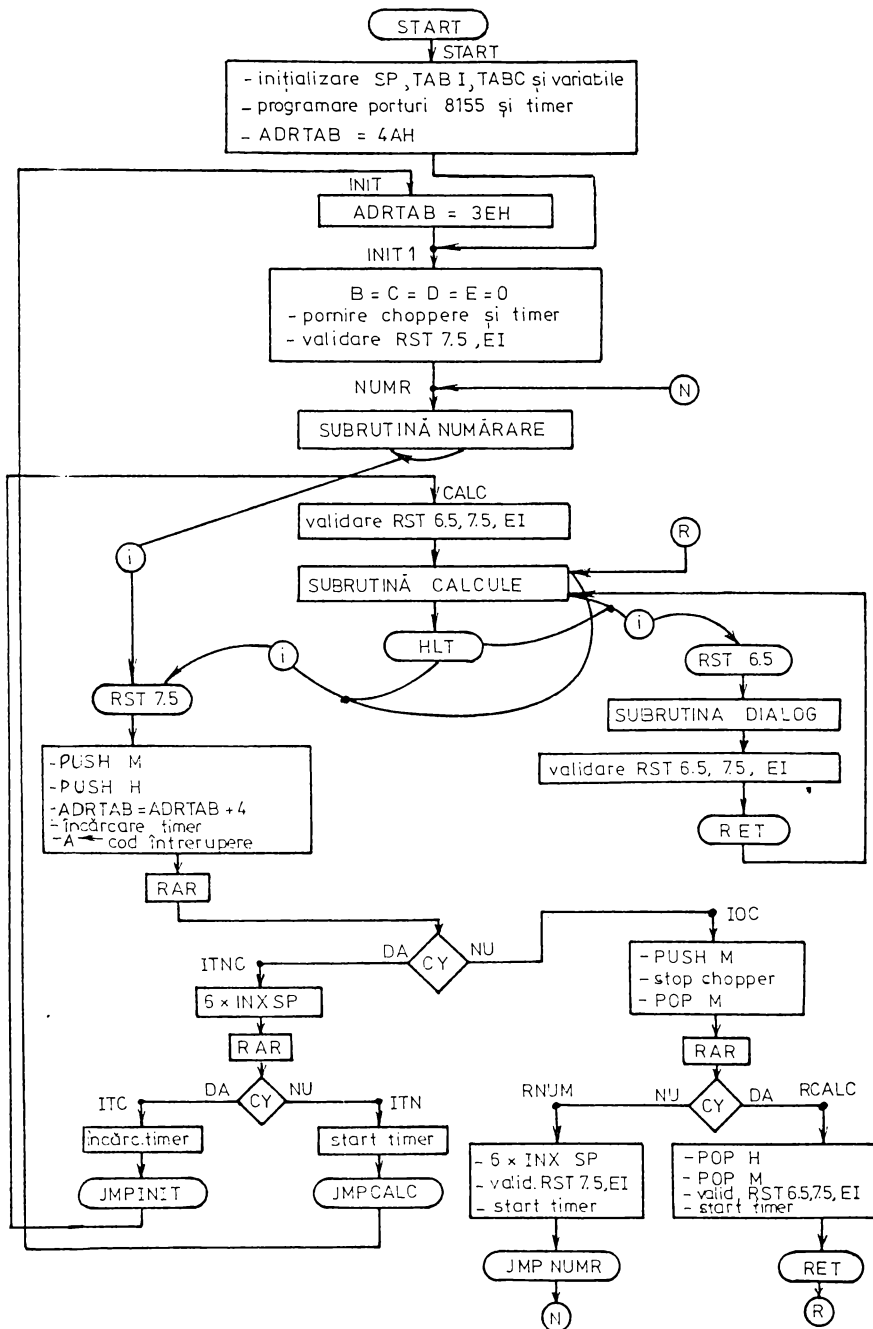


Figura 6.7. Organigrama detaliată a programului principal

poate fi întreruptă excesiv de des.

### 6.5. Măsurarea vitezei curente de rotație

Determinarea vitezei de rotație curente a fiecărui motor se realizează prin numărarea impulsurilor generate de traductorul TIRO aferent, cuplat pe arborele fiecărui motor. Se procedează la numărarea ambelor fronturi (crescător și descrescător) ale fiecărui impuls pe durata  $T/2 = 5$  ms. Restul de timp  $T/2$  este, în acest fel, disponibil pentru efectuarea calculelor implicate de algoritmul de reglare.

Pentru realizarea cât mai operativă, a numărării, s-a elaborat o subrutină de numărarea simultană a fronturilor impulsurilor generate de cele trei traductoare incrementale. Liniile  $A_1$ ,  $A_2$ ,  $A_3$  pe care sosesc impulsurile de la cele trei traductoare sînt conectate la trei linii ale portului de intrare 21 H din sistem (vezi fig.6.8.).

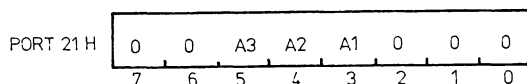


Figura 6.8. Modul de conectare la portul 21 H a liniilor de la traductoarele incrementale

Liniile portului care rămîn neutilizate sînt legate la nivelul 0 logic prin rezistoare pentru a se elimina erorile de citire a valorii numerice înscrise la un moment dat în port.

Pentru numărare se utilizează registrele interne ale microprocesorului astfel:

- Registrul B, motorul 1
- Registrul C, motorul 2
- Registrul D, motorul 3

Numărarea se efectuează prin citiri repetate la diferite intervale de timp ale conținutului portului 21 H. Sesizarea unei tranziții survenite, pentru unul sau mai multe traductoare, între două citiri succesive se realizează prin efectuarea operației logice SAU EXCLUSIV între valori citite în două momente consecutive (instrucțiunea XRA). Se obține, astfel, un cuvînt de opt biți care se folosește la formarea unei adrese de salt, prin concatenarea cu valoarea înscrisă în registrul H ce reprezintă cei mai semnificativi 8 biți de adresă (28 H). Saltul se execută

cu instrucțiunea PCHL. La adresa respectivă se află instrucțiunile de **incrementare** a registrelor care trebuie incrementate urmate de o instrucție de salt pentru continuarea numărării.

Subrutina de numărare se execută în buclă închisă, efectuându-se citiri ale portului 21 H cu frecvența maxim realizabilă, pînă la sosirea semnalului de oprire a numărării (ITN) furnizat de timer.

Organigrama subrutinei de numărare este prezentată în fig.6.9.

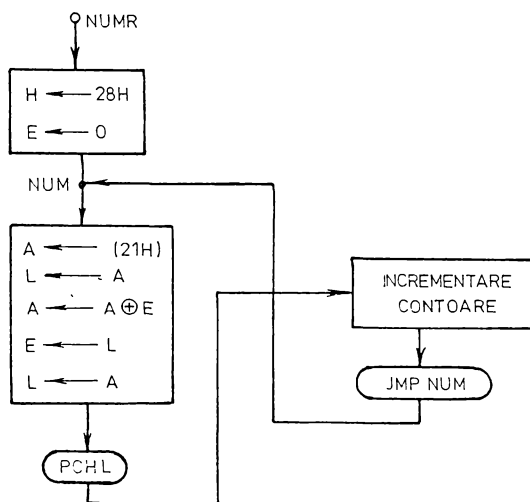


Figura 6.9. Organigrama subrutinei de numărare a impulsurilor de la traductoarele incrementale

În registrul E al microprocesorului este păstrat conținutul portului 21 H de la citirea anterioară. Apariția unei tranziții la unul, două sau toate cele trei traductoare incrementale pot determina 8 cazuri distincte în care să fie necesară incrementarea a câte unuia, a câte două sau a tuturor celor trei registre folosite la numărare. În consecință, începînd de la adresa 2800H sînt amplasate opt secvențe de incrementare, distanțate între ele cu câte opt locații de memorie (vezi programul din ANEXA A4).

Avînd în vedere mecanismul de derulare a numărării simultane a impulsurilor generate de la traductoare trebuie efectuată o analiză cu privire la turația maximă a motoarelor pentru care

nu se pierd fronturi la numărare.

Cazul cel mai defavorabil, din punctul de vedere al duratei contorizărilor, este acela când trebuie incrementate toate cele trei registre. În acest caz o execuție a subrutinei de numărare durează 49 de stări, adică un timp:

$$T_N = 49 \cdot \frac{1}{F_0} = 15,95 \text{ } \mu\text{sec} \quad (6.4)$$

Frecvența maximă a impulsurilor generate de traductoare ce pot fi contorizate este deci:

$$f_m = \frac{1}{2T_N} = 31,347 \text{ KHz} \quad (6.5)$$

ceea ce corespunde unei **turații** maxime:

$$n_{m1} = \frac{f_m}{N} \cdot 60 = 1880,8 \text{ rpm} \quad (6.6)$$

Capacitatea de 8 biți a registrelor B, C, D folosite în contorizare limitează și ea turația maximă măsurabilă:

$$n_{m2} = \frac{2^8 \cdot 60}{NT} = 1536 \text{ rpm} \quad (6.7)$$

În relațiile (6.6) și (6.7)  $N = 1000$  reprezintă numărul de impulsuri generate de un traductor TIRO la o rotație completă, iar  $T = 10 \text{ ms}$  este perioada de eșantionare.

În aceste condiții turația maximă măsurabilă cu sistemul descris este cea mai mică dintre cele două calculate în (6.6) și (6.7), adică

$$n_{\max} = 1536 \text{ rpm}$$

Turația minimă măsurabilă poate crește dacă se folosesc circuite divizoare programabile, comandate prin program. Soluția implică însă o scădere a rezoluției în numărare.

Rezoluția treptelor discrete de turație ce pot fi sesizate depinde de numărul de impulsuri pe tură generate de TIRO:

Dacă în  $T = 10 \text{ ms}$  se generează un singur impuls, acestuia îi corespunde o turație de

$$n_1 = \frac{60}{NT} = 6 \text{ rpm} \quad (6.9)$$

Deci, rezoluția obținabilă cu sistemul descris este

$$\Delta n = 6 \text{ rpm.}$$

Creșterea raportului  $n_{\max}/\Delta n$  se poate obține folosind contoare cu mai mult de 8 biți.

## 6.6. Dialogul cu sistemul ierarhic superior

În cazul aplicației realizată concret mărimile de prescriere sînt generate de la consola unui operator uman. S-a utilizat un terminal DAF lool-P, conectat la microsistemul SDK-85 printr-o interfață paralelă (vezi paragraful 3.2.2.). Într-o aplicație industrială sau un robot, rolul DAF-ului poate fi preluat de un alt calculator.

Terminalul este astfel conectat la microsistem încît apăsarea oricărei taste de pe claviatura sa generează o întrerupere hardware pe nivelul RST 6.5 al microprocesorului. Subrutina de tratare a acestei întreruperi realizează preluarea caracterului recepționat de la DAF și preluarea informației primite.

Intrucît microsistemul SDK-85 a fost dezvoltat din punct de vedere software pentru lucrul cu un terminal de tip DAF, din programul său monitor se reutilizează anumite subrutine specializate și în conducerea celor trei sisteme de acționare. Adresele de început ale subrutinelor utilizate sînt după cum urmează:

- 4336H - se depune în registrul B al microprocesorului codul ASCII al caracterului introdus de la tastatură
- 4437H - se afișează pe ecran caracterul al cărui cod ASCII se găsește în acumulator
- 4441H - se trimite la DAF caracterele de comandă <CR> și <LF>, pentru trecerea la o nouă linie
- 4565H - conversia din cod ASCII în hexazecimal a unui număr din acumulator.

Pentru generarea unei mărimi de prescriere de la tastatură operatorul uman trebuie să respecte secvența de apăsare a tastelor indicată în fig.6.10.

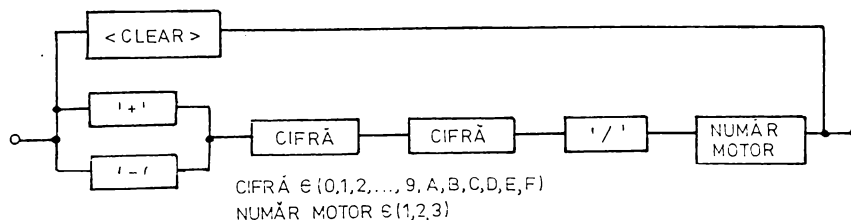


Figura 6.10. Secvența de apăsare a tastelor de către operatorul uman pentru introducerea mărimii de prescriere

După cum se observă din fig.6.10., se folosesc două cifre

pentru prescrierea referinței de viteză. Numărul de două cifre introdus reprezintă echivalentul în număr de impulsuri generate de traductorul TIRO în  $T = 10 \text{ ms}$  în cazul rotirii cu viteza respectivă prescrisă a motorului.

Relația între numărul introdus (notat cu  $w$ ) și turația  $n$ , în rotații pe minut, este dată de:

$$n = \frac{w}{N} \cdot \frac{60}{T} \quad (6.10)$$

sau

$$n \text{ [rpm]} = 6 \cdot w \text{ [hexazecimal]} \quad (6.11)$$

La fiecare caracter transmis de la tastatură se generează câte un semnal de întrerupere în urma căruia subrutina de dialog cu operatorul uman preia câte un singur caracter. Comanda completă de la DAF este executată după recepționarea tuturor celor cinci semnale de întrerupere generate de la o comandă completă, cu mențiunea că între două semnale de întrerupere, adică între două apăsări consecutive de taste, poate exista un timp de așteptare oricât de lung.

Numărul de caractere recepționate se contorizează utilizând variabila CCR - "contor caractere recepționate". Această variabilă este inițializată la valoarea zero la începutul programului și se incrementează la fiecare recepție a unui caracter. Numărarea se efectuează modulo 5, CCR devenind zero după ce a fost 5.

Subrutina de tratare a întreruperii RST 6,5 testează valoarea variabilei CCR.

Dacă CCR = 0 se așteaptă să sosească unul dintre caracterele "+", "-" sau "CLEAR". Orice alt caracter, eventual, recepționat este ignorat. În cazul în care se recepționează caracterul "CLEAR", acesta se retransmite DAF-ului în ecou, ceea ce determină numai ștergerea ecranului fără nici o altă consecință. Recepționarea unui caracter de semn este memorată la adresa SVIT astfel:

SVIT = 0, dacă s-a recepționat "+"

SVIT = 1, dacă s-a recepționat "-"

Dacă CCR = 1 se așteaptă codul ASCII al unei cifre hexazecimale, se convertește în cifră hexazecimală și se memorează, temporar, la locația VITPR.

Dacă CCR = 2 se așteaptă codul ASCII pentru cea de a doua

cifră, se convertește în hexazecimal și împreună cu prima cifră hexazecimală se formează valoarea propriu-zisă a vitezei prescrite, care se stochează în locația VITPR.

Dacă CCR = 3, se așteaptă caracterul "/" și orice alt caracter, eventual recepționat, se ignoră.

Dacă CCR = 4, se așteaptă o cifră (1, 2 sau 3) care semnifică numărul motorului, care după recepție se stochează la adresa NMOT. În continuare se parcurge o secvență de instrucțiuni care, în funcție de NRMOT și SVIT, stabilesc două măști cu care se modifică vectorul de pornire VPRN, corespunzător noii combinații de semne pentru vitezele impuse motoarelor:

- prima mască servește la modificarea codului înscris în portul de ieșire 23 H în sensul de a nu se mai aplica, la o nouă pornire, comandă de deschidere a tranzistoarelor ce s-au aflat în conducție în vechiul sens de rotație, pe axa pentru care s-a recepționat comanda. În consecință masca va conține un zero pe poziția corespunzătoare tranzistoarelor ce nu trebuie deschise și unu pe celelalte poziții, iar între acest octet mască și codul VPRN se execută operația logică "SI";

- cea de a doua mască servește la modificarea codului înscris în portul 23 H în sensul de a se aplica, la o nouă pornire, comandă de deschidere a tranzistoarelor ce asigură realizarea noului sens de rotație pe axa pentru care s-a recepționat comanda. De aceea octetul mască va conține în acest caz câte un 1 pe poziția corespunzătoare tranzistoarelor ce trebuie comandate și zero pe celelalte poziții, iar între octetul mască și codul VPRN se aplică funcția logică "SAU".

Cele două măști se aplică succesiv evitându-se intrarea în conducție, în mod simultan, a două tranzistoare de pe o aceeași ramură a chopperului și apariția surtcircuitului pe sursa de alimentare (vezi fig.6.4.).

Organigrama subrutinei de dialog cu DAF-ul este cea din fig.6.11.

### 6.7. Subrutina de calcul a algoritmilor de reglare numerică ARN

Această subrutină are ca sarcină calculul mărimii de comandă  $y_K$  conform algoritmului (5.66). Se utilizează același



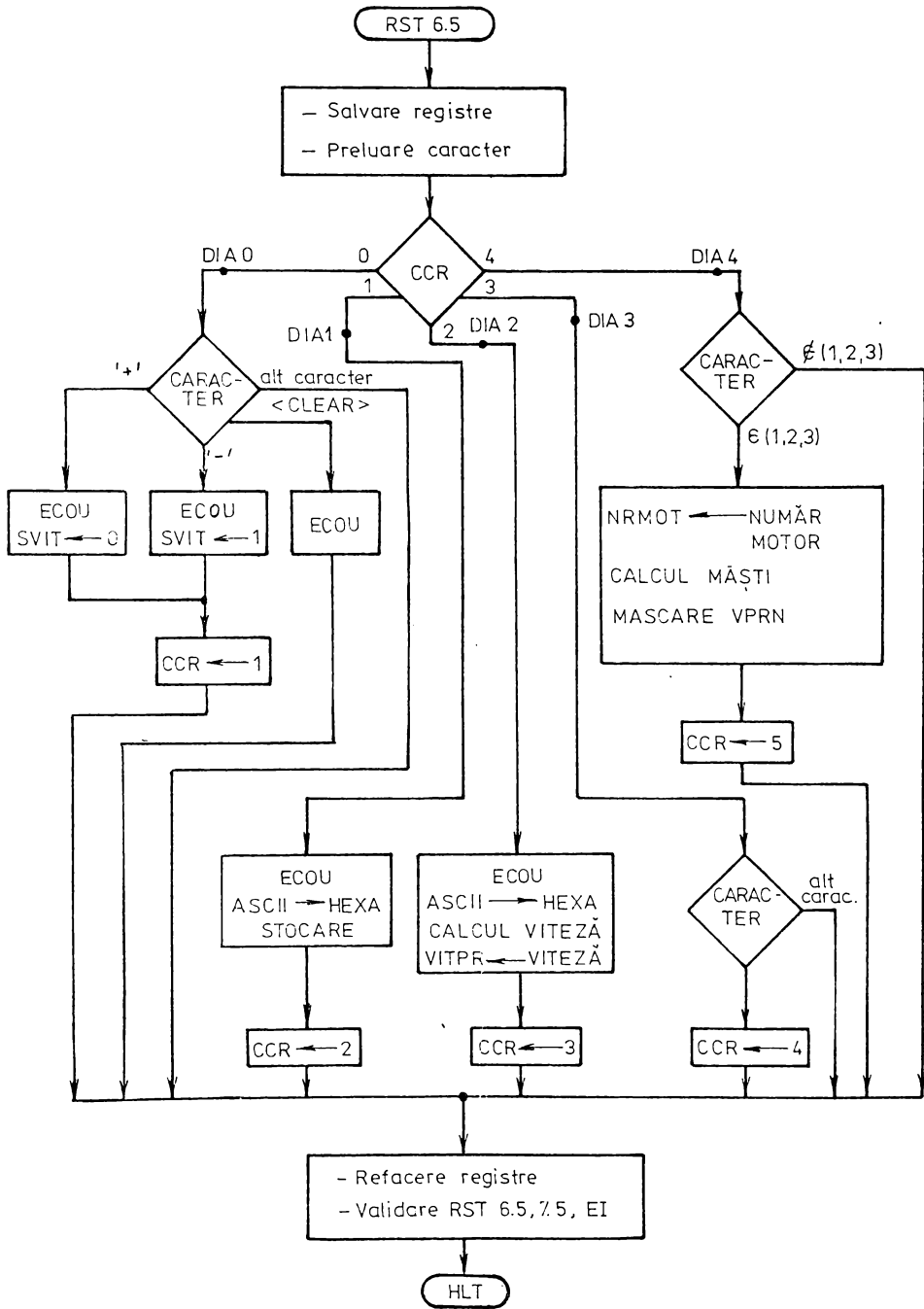


Figura 6.11. Organigrama subrutinei de dialog cu terminalul de tip DAF 1001-P

algoritm de reglare pentru toate cele trei motoare, dar calculele se efectuează independent pentru fiecare dintre motoare (se calculează de câte trei ori  $y_K$ , în fiecare cuantă).

Variabilele  $y_K, y_{K-1}, r_K, r_{K-1}, w_K, w_{K-1}$  din relația(5.66) se păstrează într-o zonă de memorie de 36 de octeți, cu o structură identică cu cele trei ARN, conform tabelului din fig.6.12.

Subrutina de calcul se compune din patru părți:

P.1) În prima parte se realizează atribuirile valorilor variabilelor în vederea trecerii la o nouă cuantă:

$$\begin{aligned} y_K &\longrightarrow y_{K-1} \\ \text{semn}(r_K) &\longrightarrow \text{semn}(r_{K-1}) \\ \text{semn}(w_K) &\longrightarrow \text{semn}(w_{K-1}) \\ r_K &\longrightarrow r_{K-1} \\ w_K &\longrightarrow w_{K-1} \end{aligned} \quad (6.12)$$

Se preiau din registrele utilizate la numărare mărimile  $r_{Ki}$  ( $i = 1, 2, 3$ ) obținute de subrutina de numărare. Aceste valori sînt depuse în tabelul din memorie la locațiile corespunzătoare.

În această parte a programului se determină și sensurile de rotație pentru cele trei motoare. Sensul de rotație al fiecărui traductor incremental este sesizat hardware cu câte o configurație cu bistabil D similară cu cea prezentată în capitolul 5 (vezi fig.5.8.). Cele trei ieșiri Q ale bistabilelor indică, prin starea lor logică, un sens de rotație sau altul și sînt conectate la câte o linie a portului de intrare 22 H, conform fig.6.13.

Dacă  $Q_i = 0$  se atribuie și  $\text{semn}(r_{Ki}) = 0$ ,  $i = 1, 2, 3$  iar dacă  $Q_i = 1$  se atribuie și  $\text{semn}(r_{Ki}) = 1$ ,  $i = 1, 2, 3$ . În acest fel prin citirea portului de intrare 22 H se cunoaște și valoarea variabilelor  $\text{semn}(r_{Ki})$ .

Subrutina testează apoi contorul CCR și dacă acesta are valoarea 5, ce indică recepționarea completă a mărimii de prescriere, depune în tabelul din memorie  $w_K$  și  $\text{semn}(w_K)$  la adresele corespunzătoare, calculate în funcție de variabila NRMOT.

În finalul acestei părți subrutina poziționează pe zero variabila CCR permițînd recepția unei noi mărimi de prescriere.

P.2) În partea a doua a sa subrutina calculează mărimile  $y_{Ki}$  ( $i = 1, 2, 3$ ) pentru cele trei motoare independent. Este

utilizată aceeași secțiune de program, care se execută de trei

AXA 1	AXA 2	AXA 3	NUME VARIABLE	SEMNICIFICAȚIE
TABC	TABC +12	TABC +24	(YK) <sub>L</sub>	} Y <sub>k</sub>
TABC +1	TABC +13	TABC +25	(YK) <sub>H</sub>	
⋮	⋮	⋮	SRK	SEMN(R <sub>k</sub> )
			SWK	SEMN(W <sub>k</sub> )
			RK	R <sub>k</sub>
			WK	W <sub>k</sub>
			(YK1) <sub>L</sub>	} Y <sub>k-1</sub>
			(YK1) <sub>H</sub>	
			SRK1	SEMN(R <sub>k-1</sub> )
			SWK1	SEMN(W <sub>k-1</sub> )
			RK1	R <sub>k-1</sub>
TABC +11	TABC +23	TABC +35	WK1	W <sub>k-1</sub>

Figura 6.12. Tabelul variabilelor din memorie pentru cele trei ARN

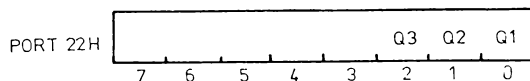


Figura 6.13. Modul de conectare la liniile portului de intrare 22 H a ieșirilor Q ale bistabilelor indicatoare de sens de rotație

ori. La fiecare parcurgere a secțiunii de calcul propriu-zis se utilizează variabilele din câte o "coloană" de 12 octeți din tabelul de variabile din fig.6.12. Calculele se efectuează apelând la subrutinele matematice (descrise în paragraful 6.8) conform algoritmului (5.66), executat în forma:

$$Y_K = Y_{K-1} \pm 375 \cdot |W_K - R_K| - \pm 350 \cdot |W_{K-1} - R_{K-1}| \quad (6.13)$$

Intrucât mărimile  $w_K$  și  $r_K$  sînt pozitive și reprezentate

pe 8 biți, iar diferențele pot fi și pozitive și negative, în relația (6.13) se vor efectua fie adunări, fie scăderi, după caz.

Cînd se calculează modulul diferenței (cu subrutina DIF), semnul acesta se memorează la locația SEMN și se ține seama de el în subrutinele de înmulțire MLT1 și MLT2 (vezi paragraful 6.8.). Mărimile  $y_K$  sînt pozitive și se reprezintă în modul pe doi octeți.

P.3) Partea a treia din subrutina de calcul ordonează, în ordine crescătoare valorile  $y_{Ki}$ . Acest lucru este necesar pentru a se stabili succesiunea întreruperilor pe parcursul unei cuante. În plus fiecărei valori  $y_{Ki}$  trebuie să i se atașeze masca de oprire ce corespunde motorului respectiv (vezi paragraful 6.4.).

Valorile calculate pentru  $y_{Ki}$  ( $i = 1,2,3$ ) se găsesc în registrele pereche ale microprocesorului după cum urmează:

$$\begin{aligned} y_{K1}, & \text{ în registrul pereche BC;} \\ y_{K2}, & \text{ în registrul pereche DE;} \\ y_{K3}, & \text{ în registrul pereche HL.} \end{aligned} \quad (6.14)$$

Registrele se compară două cîte două, după care se depun în stivă, în ordine descrescătoare valorile  $y_{Ki}$  și măștile asociate. Organigrama acestei părți din subrutina de calcul este prezentată în fig.6.14.

P.4) În cea de a patra parte a subrutinei de calcul se realizează construcția tabelului de întreruperi, cu structura prezentată în fig.6.6.

Intervalele de timp ce trebuie măsurate de la începutul cuantei au valorile numerice echivalente  $y_{K1}$ ,  $y_{K2}$ ,  $y_{K3}$ , 3C00H și 7800H. Ultimele două valori reprezintă intervalele de timp de 5 și 10 ms întrucît:

$$\frac{T}{1/f_0} = \frac{10 \text{ ms}}{325,52 \text{ ns}} = 30720 = 7800H \quad (6.15)$$

$$\frac{T/2}{1/f_0} = \frac{5 \text{ ms}}{325,52 \text{ ns}} = 15360 = 3C00H \quad (6.16)$$

În cazuri absolut particulare  $y_{Ki}$  pot rezulta chiar și de aceste valori, dintre care valoarea (6.15) reprezintă și valoarea teoretică maximă pentru  $y_{Ki}$ .

În tabelul de întreruperi se introduc valori numerice egale cu diferențele dintre valorile numerice corespunzătoare pentru două întreruperi ce trebuie să survină succesiv. Aceste valori reprezintă constantele ce urmează a se încărcă în timer succesiv.

După înscrierea constantei pentru timer se înscrie în tabel și codul tipului de întrerupere astfel:

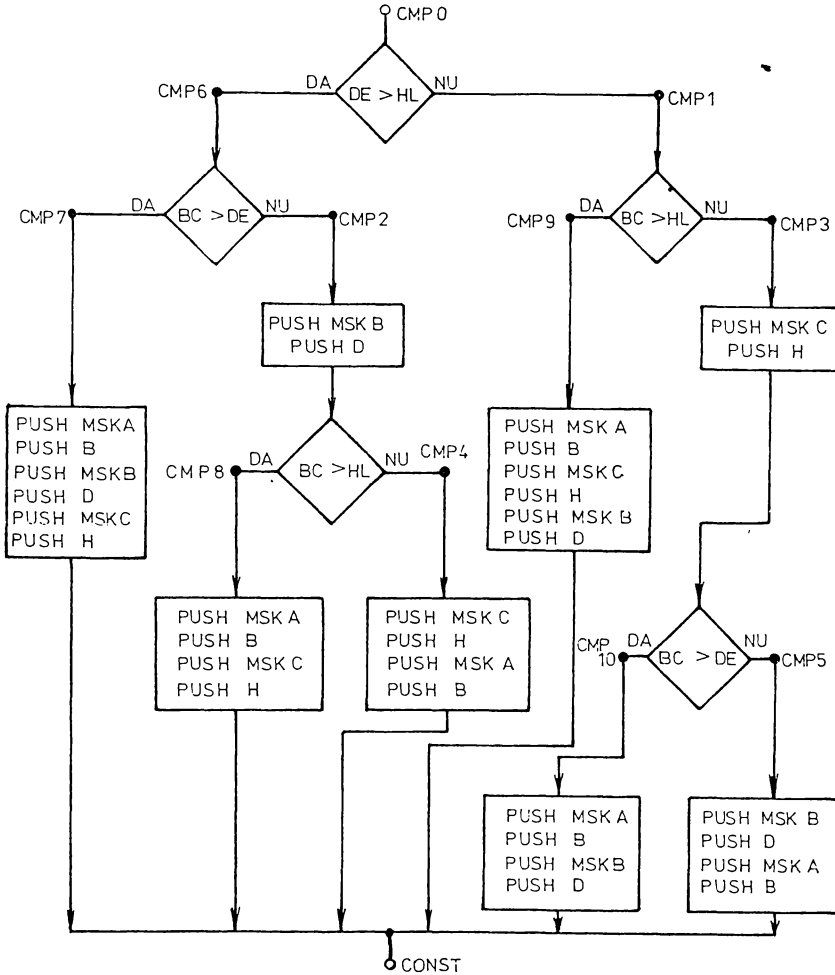


Figura 6.14. Ordonarea mărimilor  $y_{Ki}$  și asocierea corespunzătoare a măștilor de oprire în stivă.

- IOCN pentru  $y_{Ki}$ , dacă  $y_{Ki} < 3C00H$
  - IOCC pentru  $y_{Ki}$ , dacă  $y_{Ki} \geq 3C00H$
  - ITN pentru  $3C00H$
  - ITC pentru  $7800H$
- Pentru  $y_{Ki}$  ( $i = 1, 2, 3$ ) se înscrie și masca de oprire,

extrasă din stivă, unde a fost depusă o dată cu  $y_{Ki}$  în partea a treia a subrutinei.

Întreruperile de tip IOCN și IOCC introduse pe rînd în tabelul de întreruperi sînt numărate utilizînd variabila COCC - con-tor întreruperi oprire chopper.

Diferențele dintre valorile numerice, ce reprezintă inter-valele de timp, se execută cu subrutina DIFY. În cadrul acestei subrutine se efectuează operația  $DE = DE - HL$  și se limitează in-ferior rezultatul diferenței la valoarea 30H, pentru ca două în-treruperi succesive să survină suficient de distanțate în timp.

Organigrama părții a patra a subrutinei de calcul este prezentată în fig.6.15. Registrul BC se utilizează pentru a me-mora constanta pentru timer utilizată în întreruperea anterioa-ră.

Fiind ultima parte a subrutinei de calcul ARN, ea se încheie cu instrucția HLT ce menține microprocesorul în aștepta-re pînă la apariția întreruperii ITN, cu care programul se reia.

Timpul necesar efectuării tuturor calculelor și operații-lor logice din subrutina de calcul ARN este și în cazul cel mai îndelungat mai mic de 5 ms și deci subrutina se va termina înain-te de apariția întreruperii ITN /18/.

## 6.8. Subrutinele matematice

### 6.8.1. Subrutina DIF

Această subrutină execută scăderea a două numere de cîte 8 biți, în reprezentare modul + semn, cu modulul separat. Con-venția de semne utilizată este:

SEMN = 0, dacă mărimea este pozitivă sau nulă

SEMN = 1, dacă mărimea este negativă.

Subrutina preia din registrul HL al microprocesorului adresa din tabelul de variabile corespunzătoare motorului pentru care se fac calcule. Din tabel se extrag apoi mărimile semn ( $r_K$ ), semn ( $w_K$ ),  $r_K$  și  $w_K$ .

Rezultatele calculelor sînt depuse după cum urmează:

- în acumulator modul diferenței ( $r_K - w_K$ )
- la locația de memorie SEMN, semnul diferenței  $w_K - r_K$
- în registrul HL, adresa variabilei  $w_K$  din tabelul de

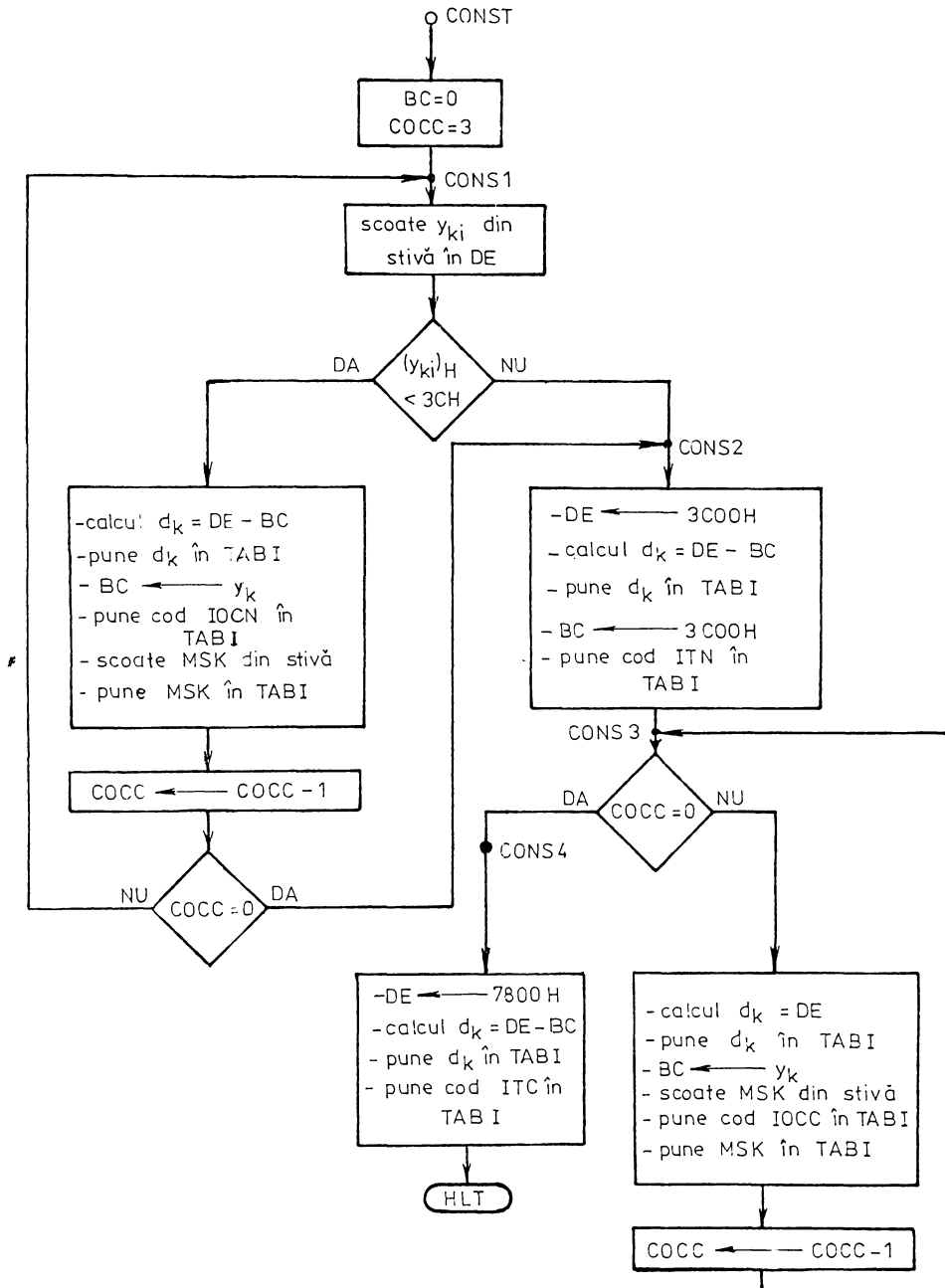


Figura 6.15. Intocmirea tabelului de întreruperi (partea a patra a subrutinei de calcul ARN)

variabile, deoarece aceasta se va folosi și la alte calcule.

În funcție de semnele lui  $w_K$  și  $r_K$  pot apare patru modalități distincte de efectuare a operațiilor și de prelucrare a rezultatului, după cum se arată în tabelul din fig.6.16.

Semn( $w_k$ )	Semn( $r_k$ )	$w_k$	$r_k$	Operația efectuată	Semn	Test depășire	Corecție depășire
+ (0)	+ (0)	$\geq 0$	$\geq 0$	$ w_k  -  r_k $	dat de calcul	nu	-
+ (0)	- (1)	$\geq 0$	$< 0$	$ w_k  +  r_k $	+ (0)	da	255 = FFH
- (0)	+ (0)	$< 0$	$\geq 0$	$ w_k  +  r_k $	- (1)	da	255 = FFH
- (1)	- (1)	$< 0$	$< 0$	$ r_k  -  w_k $	dat de calcul	nu	-

Figura 6.16. Modalitățile de efectuare a diferenței  $w_K - r_K$

Corecția ce trebuie efectuată asupra rezultatului diferenței constă în sesizarea depășirii valorii maxime reprezentabile pe 8 biți 255 = FFH, și limitarea rezultatului la această valoare maximă.

Organigrama subrutinei este cea din fig.6.17.

### 6.8.2. Subrutina MLTI

Cu subrutina denumită MLTI se realizează înmulțirea unui număr de 8 biți cu o constantă și o adunare sau o scădere pe doi octeți, conform relației:

$$DE = DE \pm 378 \times A \quad (6.17)$$

Adunarea se efectuează dacă la locația SEMN se găsește 0, iar scăderea se efectuează dacă la locația SEMN se găsește 1.

S-a folosit constanta 378 în locul constantei 375 din (5.66) deoarece numărul 378 poate fi calculat cu:

$$378 = 256 + 128 - 8 + 2 = 2^8 + 2^7 - 2^3 + 2^1 \quad (6.18)$$

în care intervin numai puteri ale lui 2.

Înmulțirea se efectuează extrem de rapid, adunînd pe rînd în registrul pereche DE conținutul acumulatorului (A) în poziții rotite conform puterilor lui 2 care îl înmulțesc. Eroarea introdusă în relația de calcul a ARN (5.66) s-a dovedit practic neglijabilă.



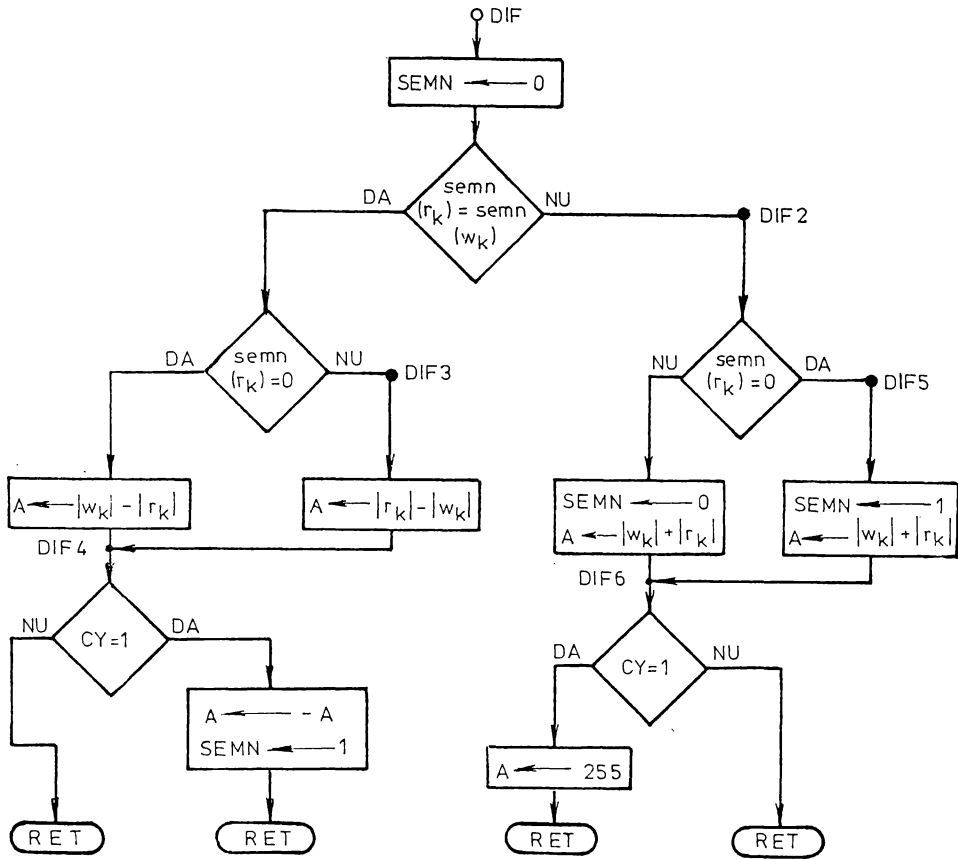


Figura 6.17. Organigrama subrutinei DIF

### 6.8.3. Subrutina MLT2

Această subrutină realizează aceeași operație ca și MLT1, dar valoarea și sensul constantei diferă:

$$DE = DE - (\pm 353 * A) \quad (6.19)$$

S-a folosit în acest caz descompunerea în factori pentru

353:

$$353 = 256 + 128 - 32 + 1 = 2^8 + 2^7 - 2^5 + 2^0 \quad (6.20)$$

## CAPITOLUL 7.

### SISTEM DE REGLARE NUMERICA A POZITIEI INTR-O ACTIONARE CU MOTOR DE CURENT CONTINUU FOLOSIND ALGORITME DE REGLARE MODALA ALUNECATOARE

#### 7.1. Introducere

În capitolele precedente s-au prezentat o serie de aplicații în care un microprocesor uzual de 8 biți, este utilizat în comanda unor sisteme de acționare electrică.

S-a folosit microprocesorul 8085 și s-a arătat că acesta este potrivit utilizării în comanda sistemelor de acționare electrică cu motor de curent continuu. Dintre argumentele favorabile utilizării acestui tip de microprocesor în acționări se pot aminti /24/, /73/:

- nu necesită circuite distincte pentru generarea impulsului de tact și interfațarea magistralelor;
- este prevăzut cu cinci nivele de întrerupere hardware mascabile separat;
- are prevăzută o intrare și o ieșire pentru semnale transmise serial;
- lucrează cu o frecvență de tact suficient de ridicată (3,125MHz);
- împreună cu încă două circuite integrate, 8155 și 8355, se poate construi un microsistem complet:
  - circuitul 8155 conține două porturi cu liniile programabile individual ca intrări și ieșiri, 256 de octeți de memorie RAM static și un timer programabil,
  - circuitul 8355 conține trei porturi intrare-ieșire și 2 octeți memorie PROM, iar varianta 8755 este prevăzută cu memorie EPROM.

În acest fel, cu numai trei circuite integrate se poate

construi un microsistem de comandă ieftin, fiabil, care poate servi la implementarea tuturor funcțiilor impuse în aplicațiile din acționările de curent continuu comandate numeric, ceea ce s-a arătat și în aplicațiile descrise anterior.

Utilizarea acestui tip de microprocesor uzual aduce cu sine, desigur, dezavantajul limitărilor de timp de calcul disponibil. Implementarea unor algoritme de reglare performante implică efectuarea unor cantități sporite de calcule în timp real, ceea ce se rezolvă principial prin utilizarea unor microprocesoare mai performante și/sau utilizarea procesoarelor specializate sau dedicate.

Intrucât în cercetările descrise în prezenta lucrare s-a urmărit în principal utilizarea microprocesoarelor uzuale și, în special, a microprocesorului 8085, considerat ca fiind foarte potrivit pentru utilizare în acționări, s-a pus problema de a studia și alte clase de algoritme de reglare, în afara acelor clase clasice de tip PID și de a se realiza implementarea lor pe microprocesorul 8085.

Urmărirea precisă, cu viteză mare a traiectoriei dorite reprezintă scopul principal al conducerii roboților industriali. Realizarea acestui deziderat este asociată cu necesitatea utilizării unor algoritme de conducere complexe, caracterizate, însă prin dificultăți de implementare în timp real.

O posibilitate de conducere corespunzătoare o oferă principiul reglării modale alunecătoare cunoscut în literatura de limbă engleză sub denumirea "the sliding-mode control".

Algoritmii de reglare modale alunecătoare reprezintă o **clasă** de algoritme de reglare moderne, recent apărută /92/, /40/ prin care se obțin performanțe superioare în reglare, asigură insensibilitatea parametrică și rejectia perturbațiilor pentru sistemul de reglare proiectat.

În capitolul de față se prezintă principiul reglării modale alunecătoare, principalele tipuri de algoritme, metodologia de proiectare concretă pentru două variante de algoritme de reglare de tipul respectiv, verificarea prin simulare a acestora și a performanțelor ce le asigură precum și modalitățile concrete de implementare pe un microprocesor uzual.

7.2. Analiza reglării modale alunecătoare (RMA)  
/92/, /84/, /68/

7.2.1. Principiul reglării modale alunecătoare

Proceselor conduse caracterizate printr-un grad de complexitate mediu sau ridicat nu le este caracteristică reglarea unei singure mărimi, ci reglarea simultană a mai multor mărimi, modelele matematice ale procesului condus fiind reprezentate prin sisteme multivariabile /32/.

Fie  $n$  numărul variabilelor de stare care descriu procesul condus. In acest caz spațiul stărilor, avînd drept coordonate variabile de stare, este spațiul  $n$  dimensional  $R^n$ . Fiecărei stări a procesului condus îi corespunde, în acest spațiu, punctul reprezentativ  $x \in R^n$ .

Fie  $u \in R^m$  vectorul variabilelor de comandă. In aceste condiții ecuația de stare a procesului condus este următoarea:

$$\dot{x} = f(x, t, u); f : R^{n \times 1 \times X \times m} \rightarrow R^n \tag{7.1}$$

Obiectivul de conducere a procesului poate fi exprimat prin următoarele ecuații:

$$s(x, t) = 0 ; s \in R^m \tag{7.2}$$

Se observă că numărul ecuațiilor care pot fi satisfăcute simultan de către punctul reprezentativ este egal cu dimensiunea vectorului comenzii. Fiecare din aceste ecuații descrie o hipersuprafață în spațiul stărilor  $R^n$ . Obiectivul de conducere este deci reprezentat prin intersecția celor  $m$  hipersuprafețe, iar evoluția stării procesului în conformitate cu obiectivul de conducere este echivalentă cu evoluția punctului reprezentativ la intersecția acestor hipersuprafețe.

Principiul reglării modale alunecătoare (RMA) constă în a constrînge punctul reprezentativ să se mențină într-un domeniu determinat, la intersecția hipersuprafețelor, prin intermediul unei legi de comandă cu structură variabilă, de forma următoare:

$$u_i(x, t) = \begin{cases} u_i^+(x, t) & \text{pentru } s_i(x, t) > 0 \\ u_i^-(x, t) & \text{pentru } s_i(x, t) < 0 \end{cases} ; i = \overline{1, m} \tag{7.3}$$

Condițiile  $s_i(x, t) > 0$  și  $s_i(x, t) < 0$  exprimă situarea punctului reprezentativ de o parte, respectiv de cealaltă parte

a hipersuprafeței de comutație descrise de ecuația  $s_i(x, t) = 0$ . În consecință, aceasta se va numi hipersuprafață de comutație (HSC), iar  $s(x, t)$  va fi numită variabilă de comutație.

Obiectivul RMA, de a forța evoluția punctului reprezentativ spre hipersuprafața de comutație, se exprimă prin următoarea condiție:

$$\lim_{s_i \rightarrow 0} s_i \cdot \dot{s}_i < 0, \quad i = \overline{1, m}$$

unde

(7.4)

$$\dot{s}_i = \frac{ds_i(x, t)}{dt}$$

Relația (7.4) se numește condiția de atingere. Se disting două cazuri:

1. Dacă  $s_i > 0$ , cerința de apropiere a punctului reprezentativ de HSC impune scăderea în timp a valorii variabilei de comutație, deci  $\dot{s}_i < 0$ .

2. Dacă  $s_i < 0$ , un raționament analog conduce la condiția  $\dot{s}_i > 0$ .

După atingerea HSC, legea de comandă va determina punctul reprezentativ să execute oscilații pe de o parte și de alta a HSC, evoluție denumită mod alunecător. Oscilațiile sînt caracterizate prin amplitudine mică și frecvență relativ ridicată, iar abaterea de la HSC este redusă. Se poate deci considera că traiectoria punctului reprezentativ este inclusă în HSC sau, altfel spus, "alunecă" pe HSC. Această comportare a sistemului este denumită mod alunecător ideal (MAI) și este descrisă prin ecuațiile următoare:

$$\begin{cases} s_i(x, t) = 0 \\ \dot{s}_i(x, t) = 0 \end{cases}; \quad i = \overline{1, m} \quad (7.5)$$

Prima ecuație reprezintă condiția de situare a punctului reprezentativ pe HSC. A doua ecuație se deduce din prima, avînd în vedere îndeplinirea condiției de mai sus la două momente oarecare:  $t$  și  $t + \Delta t$ :

$$\left. \begin{aligned} s_i(x, t) &= 0 \\ s_i(x, t + \Delta t) &= 0 \end{aligned} \right\} \Rightarrow \Rightarrow \dot{s}_i(x, t) = \lim_{\Delta t \rightarrow 0} \frac{s_i(x, t + \Delta t) - s_i(x, t)}{\Delta t} = 0 \quad (7.6)$$

Datorită abaterilor sistemului de la modelul matematic

descriu prin ecuația (7.1) (timpuri morți, întârzieri, histerezis etc.), punctul reprezentativ va evolua în vecinătatea HSC. Această comportare este denumită mod alunecător real (MAR).

### 7.2.2. Considerații generale privind proiectarea algoritmilor de reglare modal alunecătoare (ARMA)

Din descrierea principiului RMA, făcută anterior, rezultă că proiectarea ARMA impune alegerea corectă a variabilei de comutație (7.2) și a comenzii cu structură variabilă (7.3).

Variabila de comutație (respectiv variabilele, pentru cazul  $m > 1$ ) trebuie aleasă astfel încât anularea ei să reprezinte satisfacerea obiectivului de conducere, în orice situație posibilă.

Forma comenzii se va alege astfel încât să asigure evoluția punctului reprezentativ înspre HSC, pentru orice stare inițială posibilă a sistemului. Parametrii care intervin în expresia comenzii vor influența performanțele sistemului de reglare, ca de exemplu: timpul de atingere a HSC dintr-o stare inițială dată, amplitudinea oscilațiilor punctului reprezentativ în jurul HSC etc. Valori limită pentru aceste mărimi vor fi impuse la proiectare.

Se observă că, după atingerea MAI, evoluția punctului reprezentativ, deci a stării procesului, este determinată numai de obiectivul de conducere (7.2), fiind mai puțin influențată de modelul matematic al procesului (7.1). În consecință, metoda de reglare este caracterizată prin insensibilitate parametrică, printr-o bună rejecție a perturbațiilor (deoarece acestea se includ tot în ecuația de stare (7.1)).

La proiectare este deci suficientă cunoașterea aproximativă a modelului matematic al procesului și estimarea ordinului de mărime al perturbațiilor, ceea ce constituie un avantaj al acestui tip de algoritm.

### 7.2.3. Metoda comenzii echivalente pentru proiectarea ARMA

Metoda comenzii echivalente este o metodă de proiectare a modului alunecător ideal și constă în determinarea unei forme

a comenzii care să asigure îndeplinirea condiției exprimate de a doua ecuație din (7.5), din momentul satisfacerii primei ecuații.

În continuare se vor prezenta etapele proiectării MAI cu metoda comenzii echivalente /84/, /68/, /34/.

1. Se stabilește ecuația de stare a procesului condus sub forma (7.1).
2. Se exprimă dezideratul de comandă, printr-o ecuație de forma (7.2). Această etapă presupune alegerea formei variabilelor de comutație  $s_i(x, t)$ .
3. Se calculează  $\dot{s}_i(x, t)$ , ținînd cont de relația (7.1), astfel:

$$\dot{s}_i(x, t) = \frac{ds_i[x(t), t]}{dt} = \frac{\partial s_i}{\partial x} \dot{x} + \frac{\partial s_i}{\partial t} = \frac{\partial s_i}{\partial x} f(x, t, u) + \frac{\partial s_i}{\partial t} \quad (7.7)$$

4. Se rezolvă sistemul algebric (7.5), cu necunoscuta  $u$ . Dacă există, soluția  $u_{eq}$  reprezintă comanda echivalentă, adică aceea formă a comenzii care va asigura evoluția sistemului în MAI cu respectarea condiției exprimate de a doua ecuație din (7.5). Trebuie menționat că nu în toate cazurile sînt îndeplinite condițiile de existență a MAI. Studiul acestor condiții este făcut în /84/, /33/, /34/.
5. Se înlocuiește  $u_{eq}$  în ecuația (7.1), care apoi se integrează. Soluția  $x(t)$  descrie modul de evoluție a sistemului în MAI. Aplicarea metodei în cazul sistemelor liniare în raport cu comanda este descrisă în /84/, /33/, /34/.

În modul alunecător real, datorită abaterii punctului reprezentativ de la HSC, apare o componentă suplimentară în forma comenzii:

$$u = u_{eq} + \Delta u \quad (7.8)$$

Aici  $u_{eq}$  reprezintă comanda echivalentă, determinată prin metoda descrisă anterior, deci furnizează forma de variație a mărimii de comandă în condițiile evoluției sistemului în MAI. În cazul, real,  $u_{eq}$  reprezintă echivalentul valorii medii a comenzii care menține sistemul în MAR.

Componenta  $\Delta u$  are rolul de a forța punctul reprezentativ spre HSC. În conformitate cu principiul RMA, această componentă va avea o structură variabilă, cu o formă similară cele descrise de relația (7.3):

$$\Delta u_i = \begin{cases} \Delta u_i^+ & \text{pentru } s_i(x, t) > 0 \\ \Delta u_i^- & \text{pentru } s_i(x, t) < 0 \end{cases}; \quad i = \overline{1, m} \quad (7.9)$$

Calculul lui  $u_{eq}$  prin această metodă și includerea lui în forma comenzii (7.8) + (7.9) conduce la o comandă totală cu variații de amplitudine mai mici decât în cazul general descris de (7.3). Se realizează astfel o compensare anticipativă. Componenta  $\Delta u$ , are rolul de a corecta micile abateri ale punctului reprezentativ în jurul HSC, datorate neidealităților sistemului, putând avea deci valori mai mici decât comanda globală (7.3) din cazul general.

Datorită frecvenței mari a oscilațiilor punctului reprezentativ în jurul HSC, se poate considera că  $u_{eq}$  reprezintă componenta de frecvență joasă a comenzii, iar  $\Delta u$  - componenta de frecvență ridicată.

### 7.3. Variante de ARMA /84/, /68/, /33/, /34/

#### 7.3.1. Viteza absolută a variabilei de comutație

Condiția de atingere (7.4) poate fi rescrisă sub următoarea formă:

$$\dot{s}_i[x(t), t] = -P_i[x(t), t] \operatorname{sgn} \{s_i(x(t), t)\} \quad i = \overline{1, m} \quad (7.10)$$

unde

$$P_i[x(t), t] > 0$$

Factorul  $P_i[x(t), t]$  se stabilește conform strategiei de comandă.

În continuare se va analiza doar cazul  $m = 1$ , deoarece procesul condus care face obiectul cercetării de față are o singură intrare de comandă. Din relația (7.10) se obțin prin particularizare câteva cazuri importante:

a) ARMA cu viteză absolută constantă a variabilei de comutație.

Acest tip de algoritm este caracterizat de următoarea formulă:

$$P[x(t), t] = P = \text{constant} > 0 \quad (7.11)$$

b) ARMA invariant în timp, cu viteză absolută a variabilei de comutație dependentă de stare.

Formula pentru factorul  $P$  este în acest caz:

$$P[x(t), t] = P(x) > 0 \quad (7.12)$$

și reprezintă un algoritm adaptiv.

În continuare se prezintă un exemplu de strategie de modi-



ficare a parametrului  $P$ , necesitînd un timp de calcul redus, (în ipoteza unei implementări numerice).

Se definește un interval  $\Delta$  în jurul HSC:

i) Dacă traiectoria a rămas în interiorul intervalului  $\Delta$  pe parcursul ultimei perioade de comutare  $T_s$ ,  $P$  nu se schimbă.

ii) Dacă traiectoria a ieșit din intervalul  $\Delta$  în decursul ultimei perioade  $T_s$ , după ce, în anterioarele două perioade fusese în interiorul acestui interval,  $P$  este multiplicat cu un factor  $k_2 > 1$ .

iii) Dacă traiectoria a traversat intervalul  $\Delta$  în ultimul interval  $T_s$ ,  $P$  este multiplicat cu un factor  $k_1 < 1$ .

Pentru a asigura o creștere rapidă a lui  $P$ , se alege:

$$k_2 > 1/k_1 \quad (7.13)$$

### 7.3.2. Interconectarea algoritmilor de reglare modală alunecătoare cu algoritme de reglare PI

Caracteristic algoritmilor de reglare modală alunecătoare este faptul că mărimea de ieșire prezintă, în regim staționar, oscilații de mică amplitudine, întrucît punctul reprezentativ al procesului oscilează în jurul hipersuprafeței de comutație. În cazul conducerii roboților această comportare constituie un dezavantaj major. De aceea este necesar să se recurgă la utilizarea, în completare, a două tipuri de algoritme de reglare, un ARMA combinat cu algoritm de reglare PI (caracterizat prin eroare nulă în regim staționar).

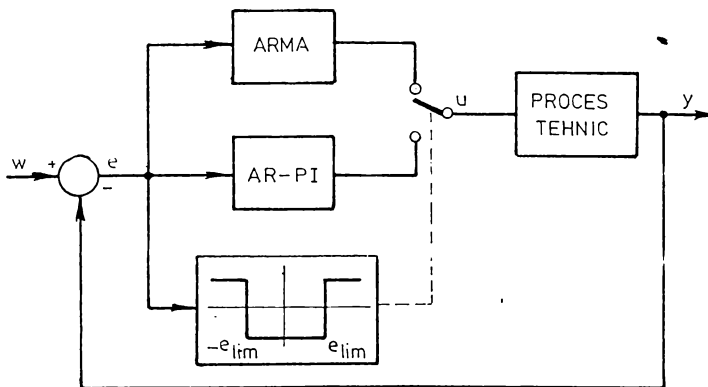


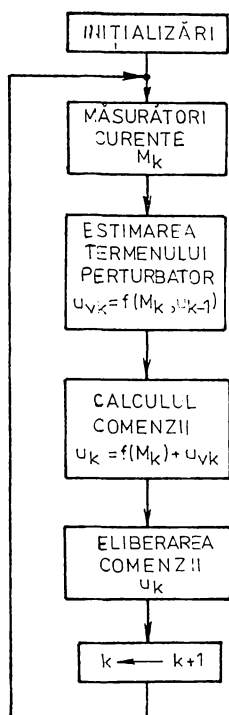
Figura 7.1. Modalitatea de interconectare a unui ARMA cu un algoritm clasic de tip PI

În fig.7.1. este prezentată modalitatea de interconectare, mai exact de comutare a procesului de reglare pe ARMA sau pe ARPI. Astfel:

- a) pînă cînd mărimea de ieșire  $y$  prezintă abateri mari în raport cu mărimea de prescriere  $w$  se procedează la reglare cu ARMA;
- b) din momentul în care abaterile dintre  $y$  și  $w$  sînt mai mici în modul decît o valoare  $e_{lim}$  se comută pe algoritmul PI.

Prin acest procedeu se obține o însumare a avantajelor aduse de fiecare dintre algoritme: performanțe superioare (viteză de deplasare mai mare pe traiectorie, insensibilitate parametrică și rejectia perturbațiilor specifice ARMA), respectiv eroare nulă de regim staționar, specifică ARPI.

### 7.3.3. ARMA cu compensarea perturbațiilor



Cu scopul de a compensa perturbațiile în formula generală a comenzii (7.8) se introduce un termen suplimentar  $u_v$ :

$$u = u_{eq} + \Delta u + u_v \quad (7.14)$$

Valoarea acestui termen se poate obține fie prin măsurarea directă, dacă procesul condus permite acest lucru, fie prin estimare. În cazul implementării numerice, estimarea perturbațiilor se bazează pe ipoteza că mărimea acestora nu suferă modificări semnificative pe parcursul unei perioade de eșantionare. Valoarea perturbațiilor pentru perioada de eșantionare precedentă se calculează în funcție de comanda ce a fost aplicată în perioada de eșantionare precedentă și efectul acesteia măsurat în perioada de eșantionare curentă.

În cazul în care se implementează software un ARMA cu compensarea perturbațiilor schema bloc de principiu a programului este cea din fig.7.2.

Fig.7.2. Organigrama bloc de implementare a ARMA cu compensarea perturbațiilor

7.4. Analiza ARMA cu viteză absolută constantă a variabilei de comutație pentru reglarea poziției într-o acționare cu motor de curent continuu

7.4.1. Descrierea generală a algoritmului

În fig.7.3. se prezintă modelul dinamic simplificat al procesului condus, ce reprezintă în acest caz sistemul de acționare.

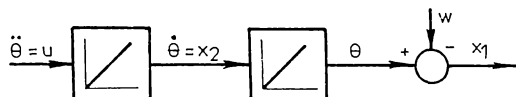


Figura 7.3. Modelul simplificat al sistemului de acționare cu MCC

Mărimea  $\theta$  reprezintă poziția unghiulară a axei motorului de acționare. Accelerația unghiulară  $\ddot{\theta}$  se consideră în acest model ca fiind mărime de comandă. În cazul unui sistem de acționare (cu MCC) real, mărimea de comandă este tensiunea aplicată motorului. Aceasta se poate calcula pornind de la valoarea dorită a accelerației unghiulare, pe baza modelului matematic al sistemului de acționare. În consecință, pentru simplificarea analizei, se poate folosi modelul simplificat prezentat în fig.7.3.

Mărimea de prescriere pentru ARMA se notează  $w$ . În aceste condiții, variabilele de stare se aleg după cum urmează:

$$\begin{cases} x_1(t) = \theta(t) - w(t) \\ x_2(t) = \dot{\theta}(t) \end{cases} \quad (7.15)$$

Necesitatea introducerii celor două mărimi de stare va rezulta din modul de definire a obiectivului de conducere, care se va face ulterior.

În conformitate cu (7.1), din relațiile (7.15) se deduc ecuațiile de stare, sub următoarea formă:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = u(t) \end{cases} \quad (7.16)$$

Se definește variabila de comutație:

$$s(t) = Cx_1(t) + x_2(t) ; \quad C > 0 \quad (7.17)$$

Obiectivul de conducere, exprimat prin ecuația (7.2),

primește acum următoarea formă concretă:

$$Cx_1(t) + x_2(t) = 0 \quad (7.18)$$

echivalentă cu:

$$\dot{\theta}(t) = -C[\theta(t) - w(t)] \quad (7.19)$$

Sînt posibile următoarele situații:

a) Dacă  $\theta = w$ , rezultă  $\dot{\theta} = 0$ , deci mărimea de ieșire este egală cu valoarea prescrisă.

b) Dacă  $\theta > w$ , rezultă  $\dot{\theta} < 0$ , deci  $\theta$  scade, urmînd a se ajunge pînă la urmă în situația a).

c) Dacă  $\theta < w$ , rezultă  $\dot{\theta} > 0$ , deci  $\theta$  crește, cu aceeași consecință de a ajunge în situația a).

În concluzie, obiectivul de conducere, exprimat prin relația (7.18), poate fi formulat astfel: mărimea de ieșire are sensul de variație care o apropie ca valoare de mărimea de prescriere  $w$ ; viteza de variație este proporțională cu eroarea de poziționare (în valori absolute).

Deoarece s-a propus o variantă de ARMA cu viteză absolută constantă a variabilei de comutație, obiectivul de conducere se formulează, conform relațiilor (7.10) și (7.11), în modul următor:

$$\dot{s}(t) = -P \operatorname{sgn} \{s(t)\} ; \quad P = \text{constant}; \quad P > 0 \quad (7.20)$$

#### 7.4.2. Analiza fazei de atingere a regimului modal alunecător

Din relațiile (7.20), (7.17), (7.16) rezultă următoarea ecuație diferențială:

$$\dot{x}_2 + Cx_2 = -P \cdot \operatorname{sgn} \{s(t)\} \quad (7.21)$$

Prin integrarea ei se obține forma de variație pentru  $x_2$  în faza de atingere:

$$x_2(t) = -\frac{P}{C} \operatorname{sgn} \{s(t)\} + \left[ \frac{P}{C} \operatorname{sgn} \{s(t)\} + x_{20} \right] e^{-Ct} \quad (7.22)$$

Integrînd prima ecuație din sistemul (7.16) se obține forma de variație pentru  $x_1$ :

$$x_1(t) = -\frac{P}{C} \cdot t \cdot \operatorname{sgn} \{s(t)\} - \frac{1}{C} \left[ \frac{P}{C} \cdot \operatorname{sgn} \{s(t)\} + x_{20} \right] e^{-Ct} + \frac{1}{C} \left[ \frac{P}{C} \cdot \operatorname{sgn} \{s(t)\} + x_{20} \right] + x_{10} \quad (7.23)$$

În formulele deduse anterior,  $x_{10} = x_1(0)$  și  $x_{20} = x_2(0)$

reprezintă condițiile inițiale, deci starea din care sistemul pornește spre RMA.

A doua ecuație din (7.16) furnizează forma legii de comandă care asigură atingerea regimului modal alunecător:

$$u(t) = - [P \cdot \operatorname{sgn} \{s(t)\} + Cx_{20}]e^{-Ct} \quad (7.24)$$

Deoarece în faza de atingere  $\operatorname{sgn} \{s(t)\}$  este constant, rezultă că mărimea din paranteză în relația de mai sus are valoare constantă,

$$P \cdot \operatorname{sgn} \{C \cdot x_{10} + x_{20}\} + C \cdot x_{20} = u_0 \quad (7.25)$$

și este dependentă numai de starea inițială a procesului. Legea de comandă poate fi scrisă deci sub forma:

$$u(t) = -u_0 \cdot e^{-Ct} \quad (7.26)$$

În cazul unei implementări numerice, se poate folosi următoarea procedură de calcul a comenzii pentru faza de atingere:

a) Se memorează (tabelar) funcția:

$$\hat{u}(t) = e^{-Ct} \quad (7.27)$$

b) Se calculează  $u_0$  conform relației (7.25).

c) Se calculează comanda:

$$u(t) = -u_0 \cdot \hat{u}(t) \quad (7.28)$$

Această metodă presupune efectuarea unei singure înmulțiri în timp real, rezultând astfel reducerea timpului de calcul.

Durata fazei de atingere, numită timp de atingere ( $t_a$ ), se calculează impunând condiția de atingere

$$s(t_a) = 0 \quad (7.29)$$

Calculînd  $s(t)$  conform relațiilor (7.17), (7.23), (7.22) și impunînd condiția (7.29) se obține:

$$t_a = \frac{C \cdot x_{10} + x_{20}}{P \operatorname{sgn} \{s(t)\}} \quad (7.30)$$

Tinînd cont că în faza de atingere  $\operatorname{sgn} \{s(t)\}$  este constant, rezultă:

$$t_a = \frac{|C \cdot x_{10} + x_{20}|}{P} \quad (7.31)$$

#### 7.4.3. Analiza fazei de regim modal alunecător

Înlocuind în ecuațiile (7.5), care descriu comportarea sistemului în regim modal alunecător ideal, forma variabilei de

comutație dată de relația (7.17) și ținând cont de ecuația de stare (7.16), rezultă ecuația diferențială pentru  $x_2$  în faza de regim modal alunecător:

$$C \cdot x_2 + \dot{x}_2 = 0 \quad (7.32)$$

Se consideră drept condiții inițiale valorile variabilelor de stare la sfârșitul fazei de atingere:

$$x_{1a} = x_1(t_a); \quad x_{2a} = x_2(t_a) \quad (7.33)$$

Integrând ecuația (7.32) cu aceste condiții inițiale se obține forma de variație a variabilei de stare  $x_2$ :

$$x_2 = x_{2a} \cdot e^{-C(t-t_a)}; \quad t \geq t_a \quad (7.34)$$

Integrând prima ecuație din (7.16) și ținând cont de (7.34) rezultă:

$$x_1 = \frac{x_{2a}}{C} [1 - e^{-C(t-t_a)}] + x_{1a}; \quad t \geq t_a \quad (7.35)$$

Din relațiile (7.29) și (7.17) rezultă că în momentul atingerii regimului modal alunecător este îndeplinită condiția:

$$x_{1a} + \frac{x_{2a}}{C} = 0 \quad (7.36)$$

Formula pentru  $x_1$  în faza de regim modal alunecător devine:

$$x_1 = x_{1a} \cdot e^{-C(t-t_a)} \quad (7.37)$$

Legea de comandă care asigură această evoluție se obține din a doua ecuație din (7.16):

$$u(t) = -C \cdot x_{2a} \cdot e^{-C(t-t_a)} \quad (7.38)$$

Starea finală la care conduce regimul modal alunecător se obține trecînd la limită pentru  $t \rightarrow \infty$ :

$$x_{2\infty} = \lim_{t \rightarrow \infty} x_2(t) = 0 \quad (7.39)$$

$$x_{1\infty} = \lim_{t \rightarrow \infty} x_1(t) = \frac{x_{2a}}{C} + x_{1a} = 0 \quad (7.40)$$

$$u_{\infty} = \lim_{t \rightarrow \infty} u(t) = 0 \quad (7.41)$$

Sistemul evoluează deci spre starea de echilibru staționar caracterizată prin  $\theta = w$ .

Considerăm că această stare atinsă cu aproximație atunci cînd este îndeplinită condiția:

$$|x_1| < \Delta \quad (7.42)$$

Ținând cont de forma de variație a lui  $x_1$  dată de (7.37) rezultă formula pentru timpul de reglare (de la atingerea fazei de RMA pînă la atingerea cu aproximație a stării stabile):

$$t_r = \frac{1}{C} \ln \frac{|x_{1a}|}{\Delta} \quad (7.43)$$

#### 7.4.4. Analiza modului alunecător real

Abaterea sistemului de reglare real de la modelul matematic (întîrzierea produsă de frecvența de comutare finită, neglijarea constantelor de timp mici etc.) conduce la evoluții oscilante în jurul traiectoriilor de RMA ideal. Minimizarea amplitudinii oscilațiilor se face prin creșterea frecvenței de comutare, această frecvență este însă limitată superior de către timpul necesar efectuării calculelor. Se va analiza în continuare efectul alegerii valorii constantei  $P$  asupra amplitudinii oscilațiilor.

Variația valorii variabilei de comutație în decursul unei perioade de comutare  $T_c$  se calculează cu aproximație pe baza relației (7.10):

$$\frac{s(t_0 - T_c) - s(t_0)}{T_c} = -P \operatorname{sgn} \{s\}$$
$$|s(t_0 + T_c) - s(t_0)| = P \cdot T_c \quad (7.44)$$

Impunînd o limită pentru amplitudinea oscilațiilor variabilei de comutație sub forma:

$$|s| \leq \Delta \quad (7.45)$$

rezultă următoarea condiție de proiectare pentru  $P$ :

$$P \leq \frac{\Delta}{T_c} \quad (7.46)$$

care limitează viteza de variație a variabilei de comutație.

Din definiția lui  $s$  dată de formula (7.17) și din constatarea că  $x_1$  și  $x_2$  au același semn, rezultă că îndeplinirea condiției (7.45) garantează îndeplinirea condiției:

$$|x_1| \leq \frac{\Delta}{C}, \quad (7.47)$$

care limitează amplitudinea oscilațiilor poziției  $\theta$  în jurul valorii prescrise  $w$ .

În analiza de pînă acum s-a considerat domeniul de variație a lui  $u$ , nelimitat. În realitate,  $u$  poate lua valori într-un

interval  $[-u_{\max}, u_{\max}]$ . În /84/ se face analiza fazei de atingere în acest caz. Se poate demonstra că, dacă valoarea inițială  $x_{20}$  a variabilei de stare  $x_2$  este situată în intervalul:

$$\left[ -\frac{P + u_{\max}}{C}, -\frac{P - u_{\max}}{C} \right]$$

pentru  $s > 0$ , respectiv:

$$\left[ \frac{P - u_{\max}}{C}, \frac{P + u_{\max}}{C} \right]$$

pentru  $s < 0$ , atunci sistemul evoluează spre regimul modal alunecător cu comanda nesaturată. Pentru  $x_{20}$  situat în afara acestor intervale, sistemul evoluează în prima fază cu comanda saturată, după care se trece la o fază de atingere cu comanda nesaturată. În concluzie domeniul de accesibilitate (domeniul din planul stărilor în care sistemul poate ajunge la regimul modal alunecător) este întreg planul  $\langle x_1, x_2 \rangle$ . În plus, dacă este îndeplinită condiția:

$$P \leq u_{\max} \tag{7.48}$$

sistemul va evolua cu comanda nesaturată, indiferent de semnul lui  $s$ , atunci când pornește din repaus ( $x_{20} = 0$ ).

### 7.5. Aspecte privind acordarea ARMA

În acest paragraf se vor estima performanțele ARMA cu scopul de a deduce unele indicații pentru acordare. Calcularea performanțelor ARMA, pornind de la condiții inițiale oarecare, prin metode analitice, este dificilă. Din această cauză, se preferă a se studia evoluția sistemului pornind din starea inițială:

$$x_{10} > 0 ; \quad x_{20} = 0$$

caracterizată prin viteză nulă. Forma de variație a mărimilor este reprezentată în fig.7.4.

Evoluția punctului reprezentativ în spațiul stărilor este prezentată în fig.7.5.

În continuare se studiază modul în care constanta  $C$  influențează timpul total de reglare. Particularizînd relația (7.31) în condițiile inițiale menționate anterior, rezultă valoarea timpului de atingere a fazei de RMA:

$$t_a = \frac{C|x_{10}|}{P} \tag{7.49}$$



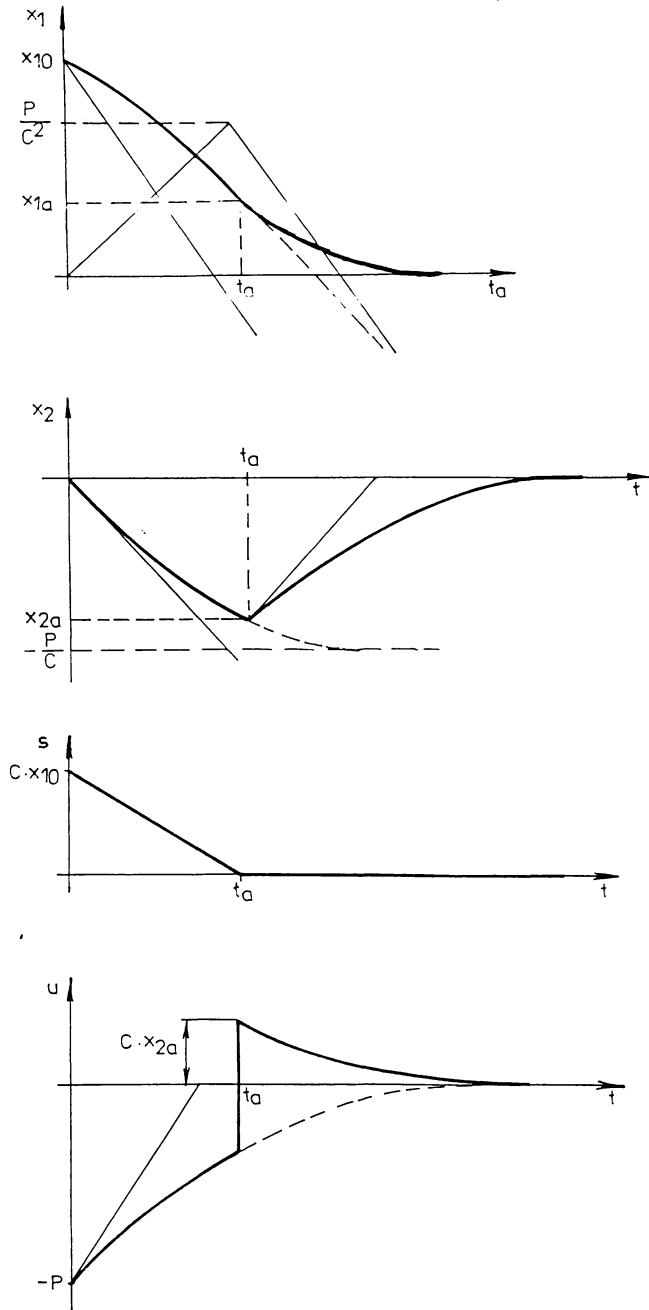


Figura 7.4. Variația în timp a mărimilor de stare, a variabilei de comutație și a comenzii

Variația timpului de atingere în funcție de valoarea inițială a abaterii de poziție ( $x_{10}$ ), pentru diverse valori ale constantei C este prezentată în fig.7.6.

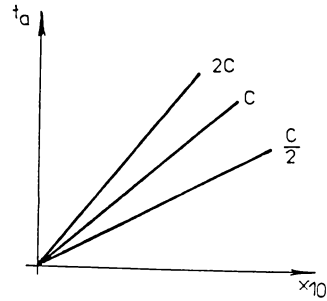
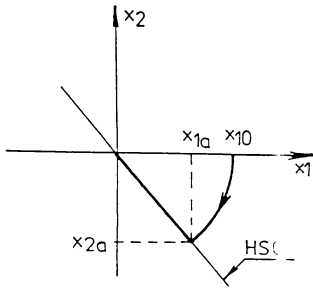


Figura 7.5. Evoluția punctului reprezentativ în spațiul stărilor

Figura 7.6. Dependența timpului de atingere de abatere de poziție inițială

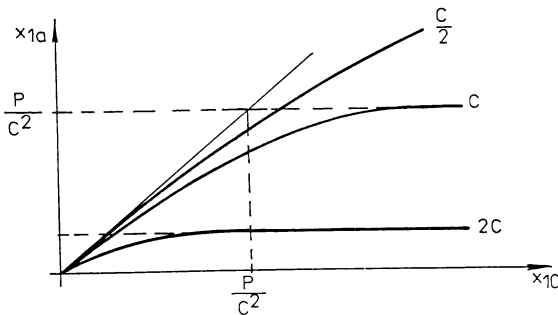


Figura 7.7. Valoarea finală a mărimii de stare  $x_1$

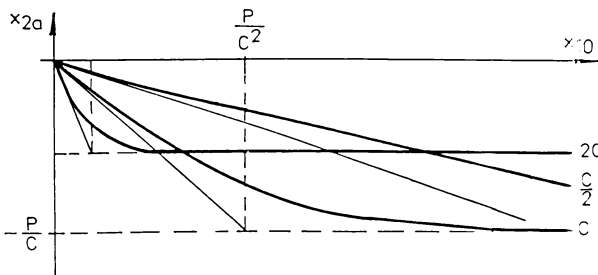


Figura 7.8. Valoarea finală a mărimii de stare  $x_2$

Valorile finale pentru variabilele de stare sînt date de formulele:

$$x_{1a} = \frac{P}{C^2} \left[ 1 - e^{-\frac{C^2}{P} x_{10}} \right] \quad (7.50)$$

$$x_{2a} = \left[ -\frac{P}{C} 1 - e^{-\frac{C^2}{P} x_{10}} \right] \quad (7.51)$$

și sînt reprezentate grafic în figurile 7.7. și 7.8.

Se observă că, pentru  $x_{10} > n \frac{P}{C^2}$ , unde  $n = 3 \div 6$ , aceste valori finale variază puțin, avînd valori apropiate de maxim:

$$|x_{1a}|_{\max} = \frac{P}{C^2} \quad (7.52)$$

$$|x_{2a}|_{\max} = \frac{P}{C} \quad (7.53)$$

Timpul de reglare, conform relațiilor (7.37), (7.51), are expresia:

$$t_r = \frac{1}{C} \left[ \ln \frac{1}{\Delta} \frac{P}{C^2} + \ln \left( 1 - e^{-\frac{C^2}{P} x_{10}} \right) \right] \quad (7.54)$$

Dependența calitativă între timpul de reglare  $t_r$  și abaterea inițială  $x_{10}$  este prezentată în fig.7.9.

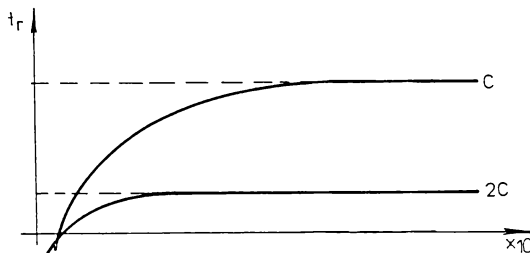


Figura 7.9. Dependența timpului de reglare de abaterea inițială

Se remarcă scăderea lui  $t_r$  la creșterea constantei  $C$ , spre deosebire de  $t_a$ , care este proporțional cu  $C$ . Rezultă necesitatea găsirii unei valori a lui  $C$  care să permită obținerea unui timp total de reglare:

$$t_{tr} = t_a + t_r \quad (7.55)$$

de valoare mică, pentru condiții inițiale într-o plajă de variație dată.

Pentru valori  $x_{10} > n \frac{P}{C^2}$ , ( $n = 3 \div 6$ ), timpul de reglare are o valoare apropiată de cea maximă:

$$t_r \max = \frac{1}{C} \left( \ln \frac{P}{C^2} \cdot \frac{1}{\Delta} \right), \quad (7.56)$$

obținută din formulele (7.54) și (7.52). În acest caz, timpul total de reglare este:

$$t_{tr} \cong \frac{C}{P} x_{10} + \frac{1}{C} \ln \left( \frac{P}{C^2} \cdot \frac{1}{\Delta} \right). \quad (7.57)$$

Modul în care acesta depinde de  $x_{10}$ , pentru diverse valori ale constantei  $C$  este prezentat calitativ în fig.7.10.

Se observă că pentru valori mari ale lui  $x_{10}$  este avantajoasă o valoare mică a constantei  $C$ . În schimb, pentru deplasări mici sînt avantajoase valori mari ale lui  $C$ . În concluzie alegerea lui  $C$  va depinde de mărimea deplasărilor ce se vor efectua. De exemplu, în cazul unui robot industrial care execută o mișca-

re punct cu punct pe distanțe mari, se va alege o valoare mică pentru C. Pentru un robot care execută o deplasare pe traiectorie continuă și în cazul cărui regulator de poziție primește valori succesive ale referinței de la generatorul de traiectorie, deplasările succesive care se execută sînt mici; în consecință, se recomandă valori mari pentru C.

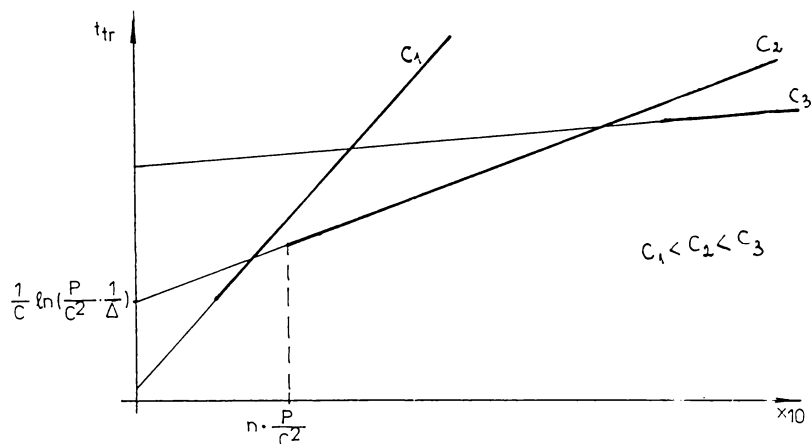


Figura 7.10. Dependența timpului total de reglare de abaterea inițială

Se analizează în continuare efectul constantei P asupra performanțelor ARMA. Se constată că valorile lui P au influență contrară, decît valorile alese pentru C și anume: valori mari pentru P se recomandă în cazul deplasărilor mari în timp ce valori mici pentru P sînt recomandabile pentru deplasări mici.

La pornirea din repaus, pentru a se asigura evoluția sistemului cu comanda nesaturată, se vor alege pentru P valori mai mici decît valoarea maximă a comenzii ( $u_{\max}$ ). Acest lucru rezultă și din analiza formei de variație a comenzii prezentată în fig.7.4., în care se vede că valoarea maximă a acesteia este P.

Un alt aspect important este acela de a asigura o comandă nesaturată la începutul fazei de regim modal alunecător, unde apare schimbarea semnului comenzii. Dacă nu se respectă această cerință nu se asigură o evoluție a sistemului în regim modal alunecător și poate apare suprareglajul. Studiind fig.7.4., rezultă că trebuie îndeplinită condiția:

$$|C \cdot x_{2a}| \leq u_{\max} \quad (7.58)$$

Inlocuind  $x_{2a}$  cu valoarea dată de (7.51) se obține:

$$P(1 - e^{-Ct_a}) \leq u_{\max} \quad (7.59)$$

Intrucât în această relație paranteza este subunitară, condiția mai sus amintită este îndeplinită, pentru orice  $t_a$ , dacă:

$$P \leq u_{\max} \quad (7.60)$$

În paragraful 7.4.4., la analiza modului alunecător real, s-a constatat că pentru valori:

$$P \leq \frac{\Delta}{T_C} \quad (7.46)$$

se obțin oscilații ale variabilei de comutație de amplitudine maximă egală cu  $\Delta$ . În consecință amplitudinea oscilațiilor de poziție va rezulta cu siguranță mai mică decât  $\Delta/C$ .

O altă consecință a măririi valorii lui  $P$  este creșterea consumului de la sursa de energie ce alimentează motorul de curent continuu. Analiza acestei dependențe este legată de modelul matematic complet al sistemului de acționare și se va efectua ulterior.

În concluzie se înțelege că determinarea valorilor optime, prin calcul analitic, pentru constantele  $P$  și  $C$ , este dificilă. Se pot face aprecieri calitative ale performanțelor algoritmului de reglare modală alunecătoare în funcție de mărimea valorii acestor constante și se pot evidenția valorile limită care se impun. Acordarea propriu-zisă implică obligatoriu simularea și apoi testarea pe model experimental.

#### 7.6. Proiectarea algoritmilor de reglare modală alunecătoare a poziției pentru un sistem de acționare cu motor de curent continuu

Sistemul de acționare utilizat are aceeași structură cu cea prezentată în capitolul 5, varianta din fig.5.6. Modelele matematice operaționale ale elementelor componente ale sistemului sînt, după cum s-a arătat în paragraful 5.1.3.:

- pentru motorul de curent continuu:

$$H_P(s) = \frac{\Omega(s)}{U(s)} = \frac{1/K}{s^2 T_m T_e + s T_m + 1} \quad (5.8)$$

cu  $\Omega$  - viteza unghiulară a motorului,  $U$  - tensiunea la borne, iar  $T_m$  și  $T_e$  - constantele de timp electromecanică și electrică a motorului.

Intrucît algoritmul de reglare modală alunecătoare urmărește reglarea poziției, mărimea de ieșire este poziția  $\theta$  a axului motorului. Intre viteza unghiulară și turație legătura este de tip integral și deci în operațional:

$$\Theta(s) = \frac{1}{s} \Omega(s) \quad (7.61)$$

Se obține pentru motor funcția de transfer:

$$H_{MCC}(s) = \frac{\Theta(s)}{U(s)} = \frac{1/K}{s^3 T_m T_e + s^2 T_m + 1} \quad (7.62)$$

- pentru chopper:

$$H_C(s) = \frac{U_m(s)}{T_c(s)} = \frac{U_1}{T} \quad (5.10)$$

unde  $U_1$  este tensiunea de alimentare continuă (24 V),  $T_c$  durata de conducție,  $U_m$  tensiunea medie la ieșire corespunzătoare unui anumit interval de conducție  $T_c$ ,  $T$  - perioada de eșantionare aleasă pentru implementare.

În capitolul 5, perioada de eșantionare utilizată a fost 10 ms. În cazul ARMA este previzibil un timp de calcul mai îndelungat și pentru a asigura o rezervă de timp suficient de largă se alege:

$$T = 12,5 \text{ msec.} \quad (7.63)$$

Și cu această perioadă de eșantionare relația (5.24):

$$T \leq 0,1 \sum_j T_j \quad (5.24)$$

este satisfăcută pentru că valoarea constantei de timp dominante (semnificative) din proces este  $T_m = 150 \text{ ms}$ .

- pentru ansamblul numărător și traductor incremental:

$$H_N(s) = \frac{N(s)}{\Omega(s)} = \frac{N}{2T} \quad (5.16)$$

cu  $N = 1000$  pentru traductorul tip TIRO 1000 (1000 de impulsuri pe tură).

Se observă că spre deosebire de situația din capitolul 5, unde trebuie ținut seama de momentul în care se efectuează măsurătoarea de turație și/sau pozițiile și momentul utilizării rezultatului acesteia, în acest caz nu este necesar să se țină

seama de timpii morți (de întârzieri) surveniți, deoarece, după cum se va vedea, algoritmul de reglare modal alunecător este practic independent de funcția de transfer a procesului condus.

### 7.6.1. Modelul matematic al procesului extins

În acest paragraf se aduce o completare la modelul matematic al sistemului, utilizat în capitolul 5, în sensul că se ține suplimentar seama de faptul că mărimea de ieșire, cu care se comandă chopperul, este un interval de timp  $T_c$ , adică o mărime nenumerică. Deoarece, practic, generarea acestui interval de timp se realizează, după cum s-a arătat, cu timerul din micro-sistemul utilizat în comandă, mărimea propriu-zisă numerică furnizată de algoritmul de reglare este codul numeric al perioadei de conducție  $T_c$ , cod care se încarcă în timer. Notăm cu  $u$  acest cod (în capitolele 5 și 6 a fost notat cu  $y$ ) și includem în schema bloc operațională a procesului condus și timerul. Se obține schema bloc operațională a procesului extins, reprezentată în fig.7.11.

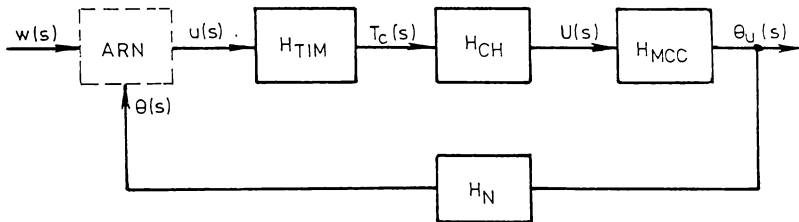


Figura 7.11. Schema bloc operațională a procesului extins

S-a arătat în capitolele 5 și 6 că generarea intervalului de timp  $T_c$  cu timerul se realizează astfel:

- se încarcă în timer codul numeric  $u$ ;
- se comandă intrarea în conducție a tranzistoarelor chopperului într-un sens de rotație;
- simultan cu semnalul de comandă pentru chopper se pornește decrementarea conținutului registrului timerului cu frecvența de tact a micro-sistemului;
- când conținutul registrului devine zero se comandă blocarea tranzistoarelor chopperului.

În consecință intervalul de conducție  $T_c$  se obține cu relația:

$$T_c = u \cdot T_s \quad (7.64)$$

unde  $T_s = 1/f = 325,52 \cdot 10^{-9}$  sec., este durata tactului utilizat în microsystem.

Prin urmare, timerul are funcția de transfer:

$$H_{TIM}(s) = \frac{T_c(s)}{u(s)} = T_s \quad (7.65)$$

Tinând cont de relațiile (5.8), (5.10), (5.16) și (7.65) funcția de transfer a procesului condus extins este:

$$\begin{aligned} H(s) &= \frac{\Theta(s)}{u(s)} = H_{TIM}(s) \cdot H_c(s) \cdot H_N(s) \cdot H_{MCC}(s) = \\ &= T_s \frac{U_1}{T} \cdot \frac{N}{2T} \cdot \frac{1/K}{s^3 T_m T_e + s^2 T_m + 1} \end{aligned} \quad (7.66)$$

### 7.6.2. Proiectarea algoritmului de reglare modală alunecătoare ARMA-1 în condițiile neglijării inerției la producerea acțiunii ponderomotoare

#### 7.6.2.1. Modelul matematic al procesului condus extins, în condițiile neglijării inerției la producerea acțiunii ponderomotoare

Varianta propusă în acest paragraf utilizează o formă simplificată pentru funcția de transfer a motorului de curent continuu și anume o formă în care cuplul motor este proporțional cu tensiunea la borne. Această simplificare permite includerea algoritmului utilizat în forma prezentată în paragraful 7.4. și este justificată de rezultatele obținute experimental.

Pornind de la ecuația (5.1) a motorului de curent continuu:

$$u = K\Omega + R_i \cdot i_i + L \frac{di_i}{dt} \quad (5.1)$$

și considerînd în regim staționar cu  $\Omega^* = ct$  corespunzător unei anumite tensiuni constante la borne  $u_1$ :

$$u_1 = K\Omega^* + R_i \cdot i_{i1} \quad (7.67)$$

Tinând cont de relația:

$$M = K \cdot i_i \quad (7.68)$$

relația (7.67) devine:

$$u_1 = K \Omega^* + \frac{R_i}{K} M_1 \quad (7.69)$$



Dacă turația  $\Omega^*$  se menține constantă, la o altă tensiune  $u_2$  aplicată la borne se modifică valoarea cuplului motor:

$$u_2 = K \Omega^* + \frac{R_i}{K} M_2 \quad (7.70)$$

În consecință, în condițiile menținerii turației constante și neglijând inerțiile producerii cuplului motor poate fi acceptată relația de aproximare:

$$\Delta u = \frac{R_i}{K} \Delta M \quad (7.71)$$

Ca urmare, prin neglijarea dinamicii de producere a cuplului motor, între tensiunea la bornele motorului și cuplul dezvoltat există o relație de proporționalitate.

Ecuția de mișcare a motorului este:

$$M - M_s = J \frac{d\Omega}{dt} = J \dot{\Omega} \quad (7.72)$$

unde  $M_s$  este cuplul de sarcină.

Tinând cont și de relația dintre poziție și turație:

$$\dot{\theta} = \Omega \quad (7.73)$$

pentru motorul de curent continuu se obține schema bloc operațională din fig.7.12., valabilă dacă se neglijează inerția producerii acțiunii ponderomotoare.

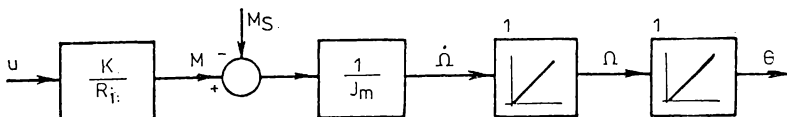


Figura 7.12. Schema bloc operațională a MCC cu neglijarea dinamicii producerii cuplului motor

Funcția de transfer care rezultă, în acest caz pentru MCC este:

$$H_{MCC}^*(s) = \frac{\theta(s)}{U(s)} = \frac{K}{R_i J} \cdot \frac{1}{s^2} \quad (7.74)$$

Pentru întregul proces condus extins se obține funcția de transfer, similară relației (7.66),

$$\begin{aligned} H_{\Sigma}^*(s) &= \frac{\theta(s)}{u(s)} = H_{TIM}(s) \cdot H_C(s) \cdot H_N(s) \cdot H_{MCC}^*(s) = \\ &= T_s \frac{U_1}{T} \cdot \frac{N}{2\pi} \cdot \frac{K}{R_i J} \cdot \frac{1}{s^2} \end{aligned} \quad (7.75)$$

Notînd cu:

$$d = \frac{R_i J}{K} \cdot \frac{2\pi}{N} \cdot \frac{T}{U_1 T_s} \quad (7.76)$$

rezultă pentru  $H_{\Sigma}^i(s)$ , relația:

$$H_{\Sigma}^i(s) = \frac{1}{d s^2} \quad (7.77)$$

Valoarea coeficientului  $d$  poate fi calculată numeric, cunoscând parametrii motorului (vezi paragraful 5.2.4.), tensiunea de alimentare a motorului (24 V), valoarea perioadei de eșantionare și a duratei impulsului de tact din sistem:

$$d = \frac{1,8 \cdot 8,5 \cdot 10^{-4}}{0,1} \cdot \frac{2 \cdot \pi}{1000} \cdot \frac{12,5 \cdot 10^{-3}}{24 \cdot 325,52 \cdot 10^{-9}} = 0,1538 \text{ sec}^2 \quad (7.78)$$

Dacă relația (7.77) se trece în domeniul timp, pentru mărimea de comandă se obține:

$$u(t) = d \cdot \ddot{\theta} \quad (7.79)$$

ceea ce arată că între accelerația axei MCC și mărimea de comandă calculată de algoritm există o relație de proporționalitate.

#### 7.6.2.2. Proiectarea ARMA-1, fără estimarea perturbatiilor

În cele prezentate în acest paragraf mărimile calculate sau măsurate vor fi cu un indice care arată numărul de ordine al perioadei de eșantionare în care se calculează sau se măsoară. Cu litera  $K$  se notează perioada de eșantionare curentă.

În concordanță cu notațiile utilizate în capitolele 5 și 6, mărimile de intrare pentru algoritm sînt:

$w_K$  - codul numeric al poziției prescrise (furnizat de sistemul ierarhic superior);

$\theta_K$  - codul numeric al poziției curente (furnizat de numărător);

Mărimea de ieșire a algoritmului este:

$u_K$  - codul numeric al intervalului de conducție pentru chopper.

Derivatele poziției, necesare în algoritmul de reglare, se calculează prin aproximările:

$$\Omega_K = \frac{\theta_K - \theta_{K-1}}{T} \quad (7.80)$$

$$\dot{\Omega}_K = \frac{\Omega_K - \Omega_{K-1}}{T} \quad (7.81)$$

unde T este perioada de eşantionare.

ARMA-1 a fost analizat în paragraful 7.4.1., de unde se vor prelua principalele formule.

Conform ecuațiilor (7.15), variabilele de stare sînt:

$$\begin{cases} x_1 = \theta - w \\ x_2 = \dot{\theta} \end{cases} \quad (7.82)$$

Se calculează deci eroarea de poziționare, cu formula:

$$e_K = \theta_K - w_K \quad (7.83)$$

variabila de comutație se calculează, conform relației (7.17), astfel:

$$s_K = C \cdot e_K + \Omega_K; \quad C > 0 \quad (7.84)$$

Obiectivul de conducere, dat de formula (7.20), va fi:

$$s_K = -P \cdot \operatorname{sgn} \{s_K\} \quad (7.85)$$

Din formulele (7.84) și (7.85) rezultă egalitatea:

$$C \cdot \Omega_K + \dot{\Omega}_K = -P \cdot \operatorname{sgn} \{s_K\} \quad (7.86)$$

de unde se calculează valoarea dorită a accelerației unghiulare:

$$\dot{\Omega}_{DK} = -C \cdot \Omega_K - P \cdot \operatorname{sgn} \{s_K\} \quad (7.87)$$

Tinînd cont de modelul matematic al procesului condus extins, exprimat de formulele (7.77), rezultă valoarea comenzii care va produce accelerația dorită:

$$u_K = d \cdot \dot{\Omega}_{DK} \quad (7.88)$$

Introducînd relația (7.87) în (7.88) rezultă:

$$u_K = -d \cdot C \cdot \Omega_K - d \cdot P \cdot \operatorname{sgn} \{s_K\} \quad (7.89)$$

Identificînd cu forma generală a comenzii, dată de (7.8), se observă că primul termen al formei comenzii date de relația anterioară reprezintă comanda echivalentă. Se poate demonstra aceasta, aplicînd metoda comenzii echivalente, descrisă în paragraful 7.2.3:

1. Ecuațiile de stare sînt (7.82).

2. Dezideratul de comandă este:

$$s = 0 \quad (7.90)$$

unde

$$s = C \cdot e + \Omega \quad (7.91)$$

3. Se calculează derivata variabilei de comutație:

$$\dot{s} = \frac{ds}{dt} = C \frac{de}{dt} + \frac{d\Omega}{dt} = C \cdot \Omega + \ddot{\theta} \quad (7.92)$$

ținând cont de (7.79), derivata se scrie:

$$\dot{s} = C \cdot \Omega + \frac{u}{d} \quad (7.93)$$

4. Se rezolvă prima ecuație din (7.5) cu necunoscuta  $u$ , obținând rezultatul:

$$u_{eq} = -d \cdot C \cdot \Omega \quad , \quad (7.94)$$

care în timp discret se scrie:

$$u_{eqK} = -d \cdot C \cdot \Omega_K \quad (7.95)$$

reprezentând primul termen din membrul 2 al egalității (7.89).

### 7.6.3. Introducerea estimatorului de perturbații

Conform celor arătate în paragraful 7.3.3., se va introduce în forma comenzii un termen suplimentar, cu rol de compensare a perturbațiilor.

Ca perturbație, se consideră cuplul rezistent la arborele MCC,  $M_g$ , incluzând în acesta și efectul celorlalte perturbații din sistem (de exemplu, abaterea tensiunii de alimentare a chopperului de la valoarea nominală, limitarea valorii comenzii, etc.).

În ipoteza unei perturbații nule ( $M_g = 0$ ), conform modelului matematic exprimat prin ecuația (7.79), comanda  $u_{K-1}$ , calculată în perioada de eșantionare anterioară și aplicată în perioada curentă, va produce în perioada curentă accelerația:

$$\dot{\Omega}_K = \frac{1}{d} u_{K-1}$$

Dacă există o perturbație, atunci diferența:

$$v_K = u_{K-1} - d \cdot \dot{\Omega}_K \quad (7.96)$$

va fi nenulă și va reprezenta valoarea estimată a acestei perturbații, măsurată în unități ale comenzii. În consecință, pentru compensarea perturbației valoarea estimată a acesteia,  $v_K$ , s-ar putea aduna la comanda  $u_K$  din perioada curentă. Semnul pe care termenul  $v_K$  îl va avea în sumă se deduce astfel: presupunând  $v_K > 0$ , rezultă că  $\dot{\Omega}_K$  este mai mic decât cel dorit, datorită unei perturbații  $M_g > 0$ . Va trebui deci crescută valoarea comenzii în perioada curentă, deci  $v_K > 0$  intervine cu semnul (+).

Pentru a evita efectele nedorite produse de perturbațiile cu frecvență mare (mai mare decât frecvența corespunzătoare perioadei de eșantionare), perturbația estimată se filtrează înain-

te de a fi adunată în comandă. Filtrarea se face cu un filtru numeric, obținut prin discretizarea ecuației unui element de transfer PT-1 cu coeficient de transfer unitar:

$$T_f \dot{\tilde{v}}(t) + \tilde{v}(t) = v(t) \quad (7.97)$$

unde  $v(t)$  reprezintă perturbația filtrată, iar  $T_f$  este constanta de timp a filtrului. Aplicînd metoda dreptunghiului pentru discretizare, se obține:

$$T_f \tilde{v}(t+T) - T_f \tilde{v}(t) + T \tilde{v}(t) = T v(t) \quad (7.89)$$

Cu notațiile:

$$k_1 = \frac{T_f}{T}; \quad v(t+T) = v_{K-1}; \quad v(t) = v_{K-2} \quad (7.99)$$

se obține formula pentru filtrul numeric:

$$\tilde{v}_{K-1} = \frac{1}{1+k_1} v_{K-1} + \frac{k_1}{1+k_1} \tilde{v}_{K-2} \quad (7.100)$$

Perturbația filtrată se include în forma comenzii, conform formulei:

$$u_K = d \cdot \dot{\Omega}_{DK} + \tilde{v}_{K-1} \quad (7.101)$$

Schema bloc a algoritmului ARMA-1 este prezentată în fig. 7.13.

#### 7.6.4. Proiectarea ARMA-2 în condițiile luării în considerare a inerției la producerea acțiunii ponderomotoare

Se utilizează forma completă a f.d.t. a procesului condus extins, dată de relația (7.66). Efectuînd notația:

$$A = T_s \frac{U_1}{T} \cdot \frac{N}{2W} \cdot \frac{1}{K} = 0,995 \quad (7.102)$$

rezultă f.d.t. sub următoarea formă:

$$H(s) = \frac{A}{s^3 T_m T_e + s^2 T_m + s} = \frac{\Theta(s)}{u(s)} \quad (7.103)$$

Aplicînd relației anterioare transformarea Laplace inversă, se obține legătura între mărimea de comandă și derivatele mărimii de măsură:

$$u = \frac{T_m T_e}{A} \ddot{\Theta} + \frac{T_m}{A} \dot{\Theta} + \frac{1}{A} \Theta \quad (7.104)$$

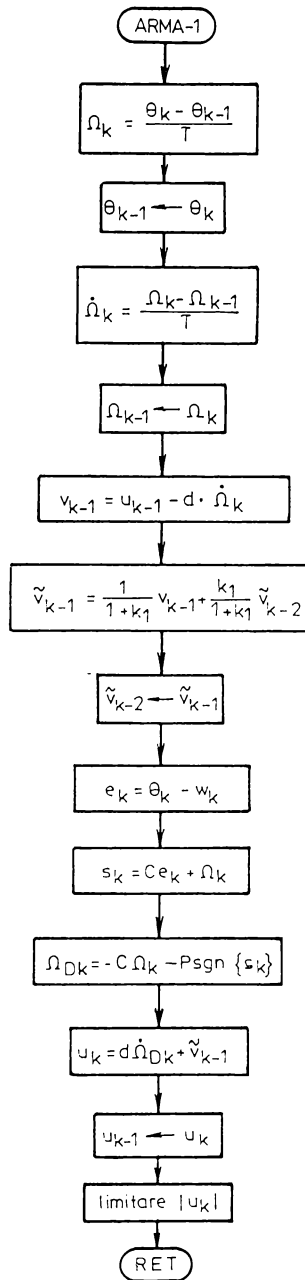


Figura 7.13. Organigrama de calcul pentru ARMA-1

Introducînd notațiile:

$$a_3 = \frac{T_m T}{A} e = 0,753 \cdot 10^{-3}$$

$$a_2 = \frac{T_m}{A} = 0,151 \quad (7.105)$$

$$a_1 = \frac{1}{A} = 1,005$$

formula (7.104) se scrie sub forma:

$$u = a_3 \ddot{\theta} + a_2 \ddot{\theta} + a_1 \dot{\theta} \quad (7.106)$$

Examinînd formula anterioară, rezultă necesitatea de a alege un număr de trei variabile de stare, astfel:

$$x_1 = \theta - w = e$$

$$x_2 = \dot{\theta} \quad (7.107)$$

$$x_3 = \ddot{\theta}$$

Ecuatiile de stare corespunzătoare sînt:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = x_3 \quad (7.108)$$

$$a_3 \dot{x}_3 + a_2 \dot{x}_2 + a_1 \dot{x}_1 = u$$

Variabila de comutație se va defini deci în funcție de cele 3 variabile de stare, astfel:

$$s = e + b_1 \dot{\theta} + b_2 \ddot{\theta} \quad (7.109)$$

Condiția de atingere se scrie sub forma (7.20):

$$s = -P \cdot \text{sgn} \{s\}; \quad P > 0; \quad P = \text{constant} \quad (7.110)$$

Substituind (7.109) în (7.110), rezultă egalitatea:

$$\dot{\theta} + b_1 \ddot{\theta} + b_2 \ddot{\theta} = -P \cdot \text{sgn} \{s\} \quad (7.111)$$

Această relație se interpretează astfel: pentru a se atinge regimul modal alunecător, este necesar să se obțină comanda  $u$  în așa fel încît derivatele poziției  $\theta$  să satisfacă relația (7.111). Cea mai simplă variantă este de a exprima valoarea dorită a derivatei întîii (pentru perioada de eşantionare următoare) în funcție de valorile celorlalte derivate (din perioada curentă):

$$\dot{\theta}_{DK} = b_1 \ddot{\theta}_K - b_2 \ddot{\theta}_K - P \cdot \text{sgn} \{s_K\} \quad (7.112)$$

Mărimea dorită a derivatei a doua va fi cea care produce variația derivatei întîii de la  $\dot{\theta}_K$  la  $\dot{\theta}_{DK}$ , în decursul unei perioade de

eșantionare, adică:

$$\ddot{\theta}_{DK} = \frac{\dot{\theta}_{DK} - \dot{\theta}_K}{T} \quad (7.113)$$

Similar, pentru valoarea derivatei a treia rezultă relația:

$$\ddot{\ddot{\theta}}_{DK} = \frac{\ddot{\theta}_{DK} - \ddot{\theta}_K}{T} \quad (7.114)$$

Mărimile calculate anterior se folosesc pentru a determina valoarea comenzii ce se va aplica în perioada de eșantionare următoare. Conform formulei (7.106), rezultă:

$$u_K = a_3 \ddot{\ddot{\theta}}_{DK} + a_2 \ddot{\theta}_{DK} + a_1 \dot{\theta}_{DK} \quad (7.115)$$

Se proiectează un estimator de perturbații, care introduce în comandă un termen suplimentar, cu rol de compensare anticipativă a perturbațiilor. Conform celor arătate în paragraful 7.6.3., perturbația din perioada de eșantionare anterioară se estimează cu formula:

$$v_{K-1} = u_{K-1} - a_3 \ddot{\ddot{\theta}} - a_2 \ddot{\theta} - a_1 \dot{\theta} \quad (7.116)$$

Formula utilizată pentru estimarea perturbațiilor este (7.100):

$$\tilde{v}_{K-1} = \frac{1}{1+k_1} v_{K-1} + \frac{k_1}{1+k_1} \tilde{v}_{K-2} \quad (7.117)$$

Forma completă a comenzii este:

$$u_K = a_3 \ddot{\ddot{\theta}}_{DK} + a_2 \ddot{\theta}_{DK} + a_1 \dot{\theta}_{DK} + \tilde{v}_{K-1} \quad (7.118)$$

Schema bloc a algoritmului este prezentată în fig.7.14.

Studiul analitic al comportării algoritmului este complicat, de aceea stabilirea parametrilor  $b_1$ ,  $b_2$ ,  $P$  se va face prin simularea numerică și prin studiu experimental.

## 7.7. Simularea cu ajutorul calculatorului a ARMA

### 7.7.1. Simularea procesului condus extins în condițiile neglijării inerției producerii acțiunii ponderomotoare PCE-1

Se folosește f.d.ț. simplificată a procesului condus extins (7.77):

$$H'_\Sigma(s) = \frac{\theta(s)}{u(s)} = \frac{1}{d} \frac{1}{s^2} \quad (7.119)$$



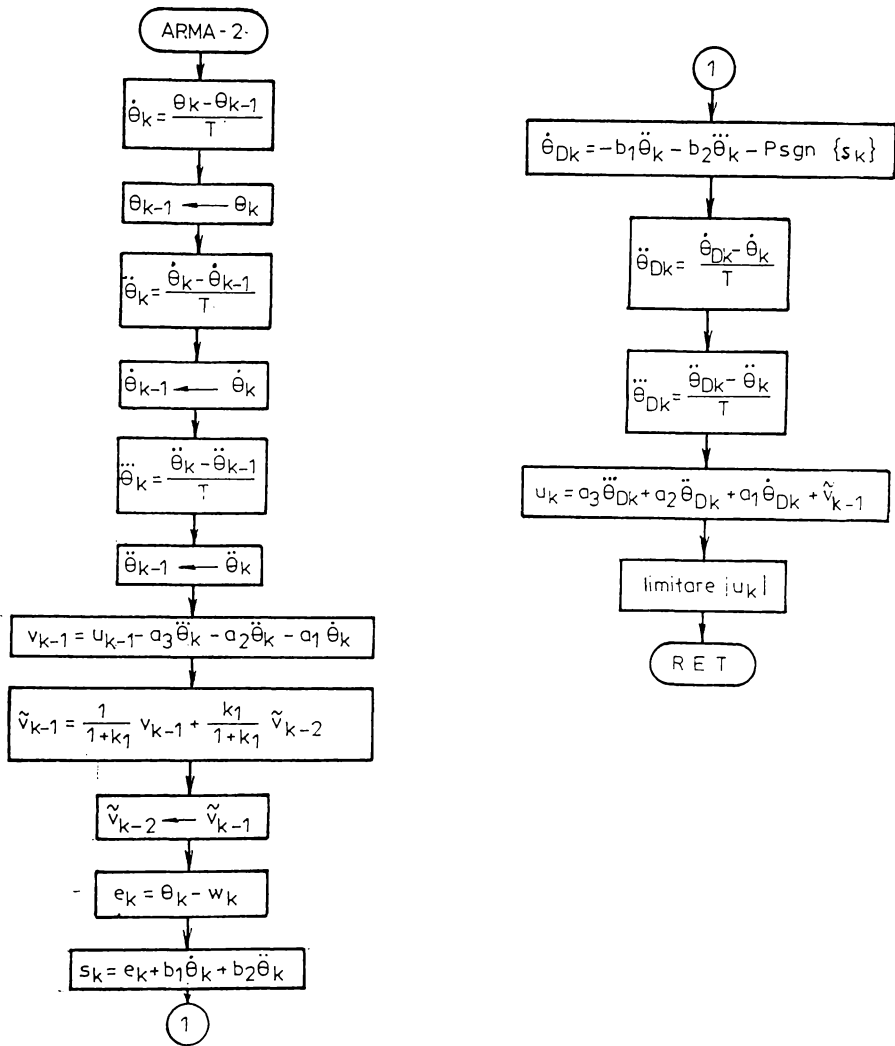


Figura 7.14. Organigrama de calcul pentru ARMA-2

Aplicînd transformarea Laplace inversă, rezultă legătura între comanda aplicată și accelerația unghiulară obținută:

$$\ddot{\theta}(t) = \frac{1}{d} u(t) \quad (7.120)$$

Se realizează discretizarea ecuației diferențiale prin metoda trapezului /32/, obținînd în final formula:

$$\theta_{k+1} = 2\theta_k - \theta_{k-1} + \alpha(u_k + 2u_{k-1} + u_{k-2}) \quad (7.121)$$

unde

$$\alpha = \frac{T^2}{4d} \quad (7.122)$$

7.7.2. Simularea procesului condus extins în condițiile luării în considerare a inerției producerii acțiunii ponderomotoare PCE-2

Se utilizează f.d.t. completă a procesului condus extins (7.103). Prin aplicarea transformării Laplace inverse, rezultă dependența între comanda aplicată și derivatele poziției unghiulare:

$$T_e T_m \ddot{\Theta}(t) + T_m \dot{\Theta}(t) + \dot{\Theta}(t) = A u(t) \quad (7.123)$$

Prin discretizare cu metoda trapezului se obține formula:

$$\Theta_{K+1} = \beta_1 \Theta_K + \beta_2 \Theta_{K-1} + \beta_3 \Theta_{K-2} + \beta_4 (u_K + 3u_{K-1} + 3u_{K-2} + u_{K-3}) \quad (7.124)$$

unde coeficienții au următoarele valori:

$$\beta_1 = \frac{24T_e T_m + 4TT_m - 2T^2}{8T_e T_m + 4TT_m + 2T} \quad (7.125)$$

$$\beta_2 = \frac{-24T_e T_m + 4TT_m + 2T^2}{8T_e T_m + 4TT_m + 2T} \quad (7.126)$$

$$\beta_3 = \frac{8T_e T_m - 4TT_m + 2T^2}{8T_e T_m + 4TT_m + 2T} \quad (7.127)$$

$$\beta_4 = \frac{AT^3}{8T_e T_m + 4TT_m + 2T^2} \quad (7.128)$$

7.7.3. Programul de simulare

Simularea decurge astfel: se consideră inițial SRA în repaus, caracterizat prin valori nule pentru poziție, viteză, accelerație și comandă. Se aplică la intrarea corespunzătoare referinței de poziție  $w$ , un semnal treaptă de amplitudine programabilă. Simultan, se prevede posibilitatea de simulare a unei perturbații de amplitudine programabilă. În schema bloc a SRA din fig.7.15., perturbația se aplică la intrarea procesului condus extins. Scopul aplicării perturbației este de a urmări modul în care aceasta este compensată de către estimatorul de perturbații.

Cu semnalele de intrare  $w$  și  $P$  se execută secvența de si-

mulare a ARMA, care calculează valoare comenzii  $u$ . In continuare se execută secvența de simulare a procesului condus extins, care calculează poziția pentru perioada de eșantionare următoare, ș.a.m.d..

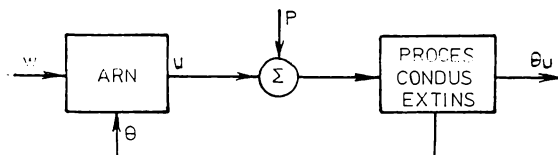


Figura 7.15. Schema bloc pentru sistemul de reglare automată

Evoluția valorii principalelor mărimi (poziția, viteza, comanda, variabila de comutație, perturbația estimată) se memorează, iar la încheierea simulării se reprezintă grafic.

Se asigură și posibilitatea programării parametrilor ARMA, în scopul acordării acestuia.

Programul de simulare este conceput în trei părți:

a) Programul principal, cu următoarele funcții:

a.1. funcția de prelucrare a următoarelor date de la operator:

- tipul de ARMA
- modelul PCE folosit
- valorile mărimilor de intrare  $w$  și  $P$
- parametrii algoritmilor de reglare

a.2. funcția de apelare succesivă a secvențelor de simulare a ARMA și procesului condus

a.3. funcția de memorare a valorilor mărimilor care interesează ( $\theta$ ,  $\Omega$ ,  $u$ ,  $s$ ,  $v$ )

a.4. funcția de a trasa graficele mărimilor memorate

b) Secvența de simulare a ARMA, care calculează valoarea comenzii  $u_k$ .

c) Secvența de simulare a procesului condus extins, care calculează poziția în perioada următoare  $\theta_{k+1}$ .

Interacțiunea celor trei părți ale programului de simulare este prezentată în fig.7.16.

Listiugul programului, scris în FORTRAN, se găsește în anexa A5.

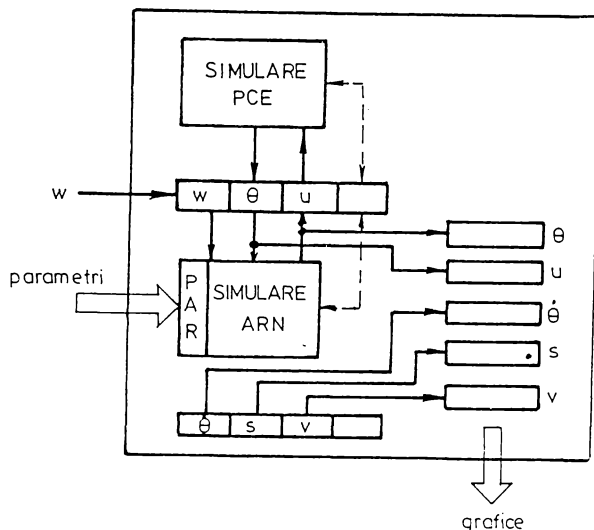


Figura 7.16. Structura programului de simulare

#### 7.7.4. Simularea ARMA-1 cu PCE-1

Acest tip de simulare utilizează modelul simplificat al funcției de transfer a procesului condus extins, deci nu urmărește simularea comportării reale a SRA avînd implementat ARMA-1. Se realizează în schimb verificarea analizei teoretice făcute în paragraful 7.4., precum și abaterile de la acest model datorate discretizării algoritmului.

Formele de variație pentru poziție (THETA), viteză (OMEGA), variabila de comutație (S), comandă (U) sînt prezentate în fig.de la pag. A6.2-A6.4 din anexa A6. Pasul de simulare utilizat este egal cu perioada de eșantionare:  $T = 12,5$  ms. Se efectuează 82 de pași, deci durata totală a simulării este de aproximativ 1 sec. În consecință, s-a utilizat o amplitudine mică ( $W = 1000$ ) pentru semnalul treaptă aplicat la intrarea referinței de poziție. Valorile parametrilor C și P s-au ales pentru a obține un răspuns rapid. Pînă la atingerea RMA, se observă că mărimile evoluează în conformitate cu analiza teoretică. În regimul modal alunecător, atins după  $t_a = 0,25$  s, variabila de comutație efectuează oscilații în jurul valorii  $s = 0$ . Cauza acestora o reprezintă discretizarea algoritmului de reglare. Consecința schimbării pe-

riodice a semnelui variabilei de comutație este evoluția oscilației a comenzii. Oscilațiile, de amplitudine mare, au loc simetric de o parte și de alta a valorii deduse analitic și reprezentate în fig.7.4. Viteza prezintă și ea oscilații mari, avînd ca valoare medie tot valoarea dedusă prin calcul analitic. Poziția în schimb, evoluează cu oscilații mici în jurul valorii calculate. Timpul total de reglare este  $t_{tr} = 0,75$  sec.

Pentru a studia efectul pe care îl are lungimea perioadei de eșantionare a algoritmului de reglare numerică, s-a efectuat o simulare cu perioada de eșantionare mai mică  $T = 5$  msec. și aceleași valori ale parametrilor ARMA-1. Graficele sînt prezentate în fig. A6.6 - A6.9. Comanda prezintă oscilații de amplitudine aproximativ egală cu cea din exemplul anterior, dar de perioadă de 2,5 ori mai mică. În consecință, oscilațiile variabilei de comutație, vitezei și poziției vor avea amplitudini mai reduse. Vor rezulta deci, în cazul implementării practice, amplitudini mai mici ale oscilațiilor poziției în jurul valorii de echilibru în regim staționar, precum și un consum mai redus de la sursa de alimentare a motorului de acționare. Se recomandă deci adoptarea unei valori cît mai reduse a perioadei de eșantionare (în limitele impuse de durata calculelor aferente algoritmului de reglare).

În scopul studierii influenței parametrilor ARMA-1 asupra performanțelor SRA, s-au efectuat simulări pentru diverse valori ale acestor parametri. Se prezintă, ca rezultat, doar graficele evoluției în timp a poziției, în paginile anexă A6.11 - A6.16.

Crescînd valoarea parametrului P se obține scăderea timpului de atingere și a timpului total de reglare. Crește în schimb amplitudinea oscilațiilor comenzii, deci consumul de la sursa de energie. Scăzînd valoarea lui P (pg.A6.12), efectul este invers. În consecință, P se va alege cît mai mare, dar se va evita supra-solicitarea sursei de alimentare.

Crescînd valoarea parametrului C, crește valoarea timpului de atingere, dar timpul de reglare rămîne aproximativ același. Scăzînd valoarea lui C (pg.A6.14), scade timpul de atingere a RMA, dar timpul total de reglare crește mult. Această comportare coincide cu cea dedusă analitic, ținînd cont că, în graficele din fig.7.10.,  $x_{10}$  se situează în zona valorilor  $n.P/C^2$ . Aceeași figură arată că valorile mici ale parametrului C sînt recomandate pentru răspunsul rapid la trepte mari de modificare a poziției.

Pentru a studia modificarea timpului total de reglare la diverse amplitudini ale semnalului treaptă pentru referința de poziție, s-au efectuat simulări pentru două valori diferite ale acesteia, cu graficele prezentate în A6.15-A6.16. Se observă că timpul total de reglare depinde puțin de amplitudinea lui  $w$ . Acest efect se accentuează pe măsura scăderii parametrului  $C$  conform fig.7.10.

#### 7.7.5. Simularea ARMA-1 cu PCE-2

Acest tip de simulare urmărește comportarea SRA experimental avînd implementat ARMA-1. Se pot evidenția astfel efectele aproximării expresiei funcției de transfer a procesului condus extins la proiectarea ARMA-1, asupra performanțelor SRA.

Comparînd răspunsul în poziție din A6.18 cu cel obținut pentru modelul PCE-1 (A6.6), la aceleași valori ale parametrilor, se observă o creștere a timpului de atingere. În schimb, nu se atinge valoarea prescrisă în intervalul de simulare de 1 sec. Sistemul se comportă ca și cînd ar avea un statistic negativ. Este posibil însă ca această comportare să fie doar aparentă, poziția prescrisă atingîndu-se după un timp mai mare decît intervalul de simulare.

Pentru a urmări efectul perturbațiilor, se realizează o simulare cu o perturbație introdusă conform fig.7.15. De menționat că perturbația estimată nu a fost introdusă în comandă. În A6.20

și se prezintă graficele pentru poziție și perturbația estimată. În ceea ce privește răspunsul, se remarcă un statistic pozitiv important (**abatere maximă +10%**). După atingerea RMA, variațiile mari în valoarea comenzii împiedică buna funcționare a estimatorului de perturbații, acesta devenind inefficient.

Graficele din A6.22 și A6.23 s-au obținut prin introducerea estimatorului de perturbații în algoritmul de reglare, prin introducerea perturbației estimate în forma comenzii, conform relației (7.101). Se observă că perturbația estimată de valoare negativă, cauzată de utilizarea modelului PCE-2 pentru simularea procesului condus, nu contribuie semnificativ la accelerarea răspunsului în faza inițială a reglării. În schimb, la atingerea unor valori mari, pozitive, pentru perturbația filtrată se aplică o corecție, avînd drept scop scăderea vitezei. Perturbația

estimată scade, avînd în continuare o evoluție oscilantă amortizată, pînă la intrarea în faza de RMA. Se observă o mică reducere a statismului, față de situația din A6.20 , dar acesta nu este eliminat complet.

În concluzie, estimatorul de perturbații nu aduce îmbunătățiri semnificative în acest caz. Efectul său este mai important în cazul unor timpi de reglare mai mari, deci al unor trepte, a poziției, de amplitudine mai mare.

#### 7.7.6. Simularea ARMA-2 cu PCE-2

Se simulează comportarea celui de-al doilea tip de algoritm, la proiectarea căruia s-a luat în considerare forma completă a funcției de transfer a procesului condus. Se pot astfel evidenția performanțele algoritmului, comparativ cu ARMA-1.

Forma de variație a mărimilor pentru un set de valori ale parametrilor este prezentată în A6.26-A6.29 . A fost necesară adoptarea unor valori foarte mici pentru parametri  $b_1$  și  $b_2$ , deoarece valori mai mari au condus la instabilitate. În consecință, ponderea mărimilor  $\dot{\theta}$  și  $\ddot{\theta}$  în formula variabilei de comutație (7.109) este redusă, iar s se anulează practic la îndeplinirea condiției  $\theta = w$ , moment în care începe faza de RMA. Tipul total de reglare este aproximativ egal cu cel obținut utilizînd ARMA-1, dar oscilațiile în regim staționar sînt mult mai mari. Mărimirea de comandă are la început o evoluție oscilantă, trecînd prin valori negative, fapt care are o influență nefavorabilă asupra vitezei de răspuns. Viteza prezintă și ea oscilații, după care se stabilizează la o valoare constantă.

Pentru a simula mai exact procesele care au loc în SRA real, s-a luat în considerare și limitarea comenzii, la valoarea  $w_{\max} = 40000$ . Graficele pentru  $\theta$  și  $u$  sînt prezentate în A6.31 și A6.32. Timpul total de reglare se păstrează același, iar amplitudinea oscilațiilor în regim staționar este în acest caz mai redusă.

Introducînd și estimatorul de perturbații, se obține pentru  $\theta$  graficul din A6.34 . Se observă scăderea amplitudinii oscilațiilor în regim staționar, aceasta avînd acum elongații maxime de  $+3 \div -26$ , ceea ce corespunde unor variații ale unghiului axei MCC de cca  $10^\circ$ . Frecvența oscilațiilor este de aproximativ

8 Hz.

Pentru a putea studia comportarea SRA cînd se aplică o mărime de referință sub formă de treaptă cu amplitudine mare, s-a realizat o modificare a programului de simulare. Se efectuează 850 de pași de simulare, iar mărimile se afișează din lo în lo pași. Listingul acestei variante de program este prezentat în anexa A5.5-A5.8.

#### 7.8. Implementarea algoritmului de reglare modală alunecătoare pe model experimental /lo/

Implementarea concretă a ARMA s-a realizat pe sistemul de acționare descris detaliat în capitolul 5, cu structura din fig. 5.6. Echipamentul de comandă numerică, realizat cu microprocesorul 8085, trebuie să execute atît calculele corespunzătoare algoritmului de reglare numerică cît și o serie de operații de dialog cu sistemul ierarhic superior (operatorul uman) și/sau cu procesul condus.

În cazul implementării ARMA s-a procedat la o întocmire modulară a programului de conducere a sistemului de acționare în sensul că s-au întocmit blocuri "software" specializate pentru realizarea diferitelor funcții ce trebuie realizate în conducere.

S-a conceput un program numit executiv de timp real ETR căruia îi revin sarcini legate de preluarea datelor de la sistemul ierarhic superior și din proces, respectiv de emiterie a comenzilor către chopper (elementul de execuție). Implementarea algoritmului de reglare numerică propriu-zis s-a realizat cu un subprogram dedicat, întocmit distinct și denumit ARN.

Realizarea programului în această structură modulară prezintă avantajul de a se putea testa experimental diferite tipuri de algoritme de reglare, fără să fie necesară rescrierea blocului ETR. În plus, cu o astfel de structură software, în memoria echipamentului de comandă numerică pot fi păstrate **mai multe variante de ARN ce pot fi apelate după necesitățile survenite la un moment dat în conducerea proceselor.**

Structura organizării conducerii procesului este reprezentată în fig.7.17.



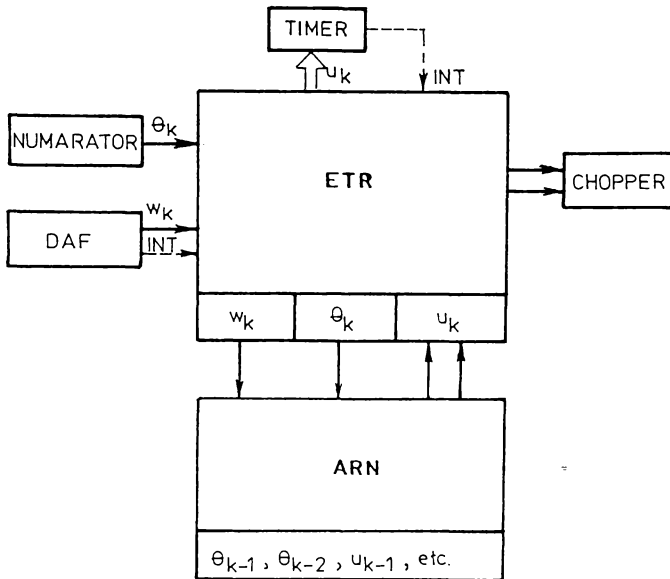


Figura 7.17. Organizarea conducerii sistemului de acționare după o structură software modulară

### 7.8.1. Executivul de timp real ETR

#### 7.8.1.1. Funcțiile programului

Funcțiile ETR sînt următoarele:

- (F1) - preluarea referinței de poziție  $w_k$  de la sistemul ierarhic superior; în cazul modelului experimental al SRA, referința de poziție se preia de la operatorul uman prin intermediul terminalului de tip DAF.
- (F2) - preluarea poziției curente  $\theta_k$  de la numărător.
- (F3) - gestiunea timpului real, în scopul generării duratelor de conducție și de blocare pentru chopper; această funcție presupune execuția a 2 operații:
  - conversia codului numeric reprezentînd durata de conducție  $u_k$  în intervalul de timp și marcarea sfîrșitului acestui interval;
  - marcarea sfîrșitului fiecărei perioade de eșantionare  $T$ .

- (F4) - comanda chopperului; se execută prin două linii diferite ale unuia dintre porturile de ieșire ale microsistemului.
- (F5) - activarea subprogramului care implementează ARN.

În plus, ETR execută inițializarea variabilelor la lansarea în execuție, precum și sincronizarea între ele a funcțiilor. Un exemplu privind desfășurarea în timp a funcțiilor ETR este prezentat în fig.7.18.

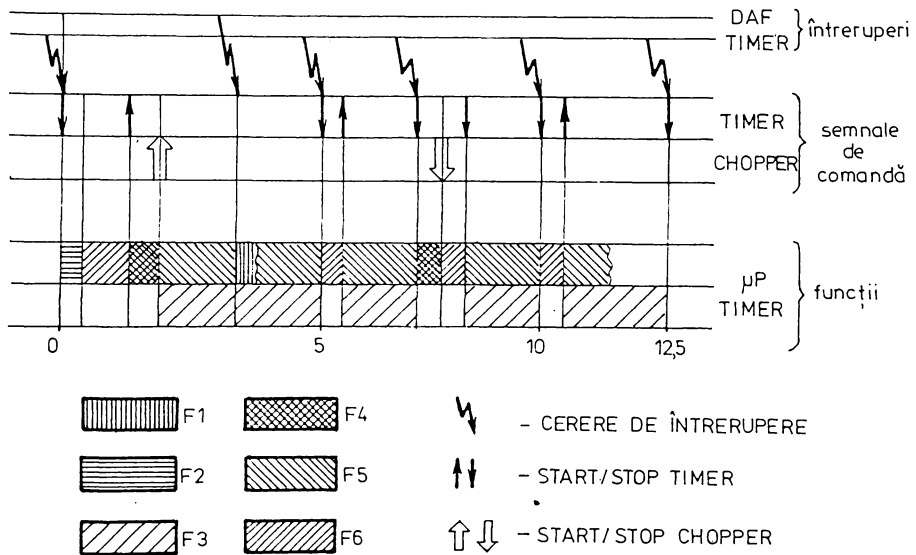


Figura 7.18. Execuția succesivă a funcțiilor executivului de timp real într-o perioadă de eșantionare

### 7.8.1.2. Sincronizarea execuției funcțiilor

Vom denumi evenimente momentele de timp la care se lansează în execuție o anumită funcție. Conform fig.7.18., evenimentele sînt marcate prin întreruperi. Deoarece microprocesorul 8085 dispune de mai multe niveluri de întrerupere hardware, au fost alocate niveluri diferite pentru DAF și timer.

Intreruperile de la DAF activează un subprogram a cărui funcționare va fi prezentată în paragraful 7.8.1.5.

Intreruperile de la timer sînt generate toate pe un același nivel, deși marchează, de regulă, evenimente diferite. Pentru

a le diferenția s-a adoptat câte un cod pentru fiecare tip de eveniment.

Operațiile care se execută în cazul fiecăruia dintre aceste tipuri de întrerupere sînt următoarele:

tip 0 - marchează un interval de 5 ms sau multiplu al acestei valori, de la începutul perioadei de eșantionare; activează sincronizarea.

tip 1 - marchează sfîrșitul perioadei de eșantionare; activează F2, F4 (start chopper) și sincronizarea.

tip 2 - marchează sfîrșitul perioadei de conducție a chopperului; activează F4 (stop chopper) și sincronizarea.

Intreruperile intermediare la 5 ms și 10 ms sînt folosite întrucît circuitul timer nu poate realiza temporizări mai lungi decît  $2^{14}$ .  $T_s = 5,33$  ms. Valoarea mare a perioadei de eșantionare (12,5 ms) a fost adoptată pentru a asigura timp de calcul suficient pentru implementarea unor algoritme de reglare complexe.

Modul de sincronizare al evenimentelor este următorul: la începutul fiecărei perioade de eșantionare se stabilește ordinea evenimentelor, în funcție de lungimea perioadei de conducție a chopperului, dată de valoarea numerică  $u_K$ . Evenimentele se înscriu în ordinea în care urmează să se producă într-un șir de așteptare, sub forma unui tabel în memorie, ca în fig.7.19.

TIP EVENIMENT 1
TEMPORIZARE EVENIMENT 2
TIP EVENIMENT 2
TEMPORIZARE EVENIMENT 3
TIP EVENIMENT 3
TEMPORIZARE EVENIMENT 4
TIP EVENIMENT 4

Figura 7.19. Tabelul de așteptare al evenimentelor dintr-o perioadă de eșantionare

Alături de codul care arată tipul evenimentului, se înscriu în tabel valorile numerice care, încărcate în timer, vor genera intervalele de timp pînă la declanșarea evenimentului respectiv. Valoarea corespunzătoare temporizării pînă la evenimentul 1 se va înscrie direct în registrul timer-ului, la construirea șirului de așteptare. De menționat că programul prezintă o mare flexibilitate, putînd fi adaptat pentru orice valoare a perioadei de eșantionare  $T$ , printr-o simplă modificare a procedurii de construcție a șirului de așteptare.

Schema logică a secțiunii de program care construiește tabelul din fig.

7.19. este prezentată în fig.7.20. Valorile numerice (în hexazecimal) care corespund unor intervale de temporizare sînt: 3C00H - 5 ms; 7800H - 10 ms; 9600H - 12,5 ms.

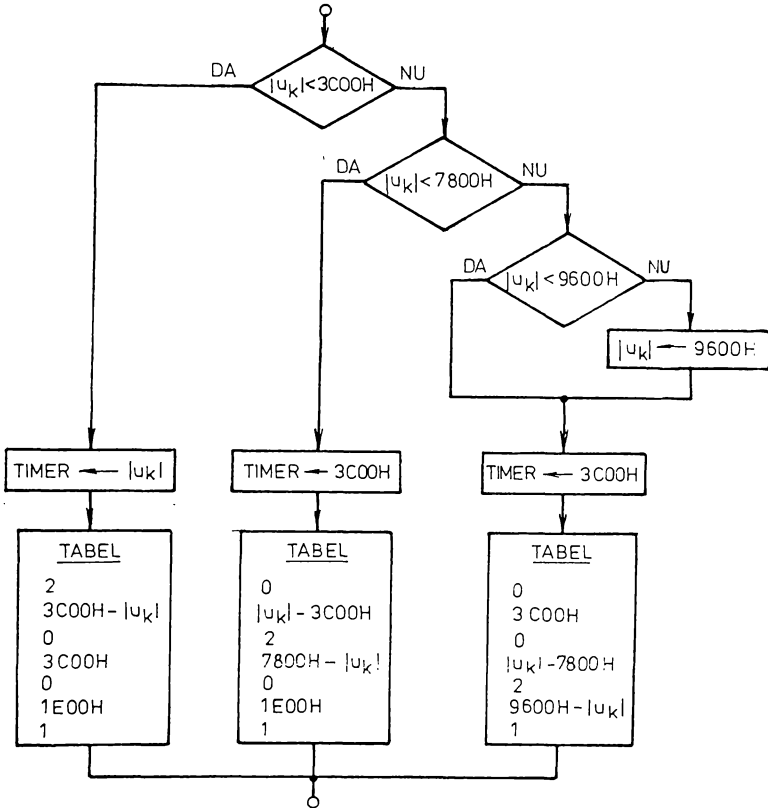


Figura 7.20. Secvența de program care construiește tabelul de așteptare.

### 7.8.1.3. Structura programului

Organigrama programului ETR este prezentată în fig.7.21. Variabilele din organigramă au următoarea semnificație:  
 SC = 1 - sfîrșit perioadă (cuantă) de eșantionare,  
 SR = 1 - sfîrșit recepție  $w_k$  de la DAF.

Detalii suplimentare asupra ETR se găsesc în listingul programului, din anexa A7. Subprogramul de tratare a întreruperilor salvează la început registrele în stivă și le reface în

final, putînd astfel îtrerupe programul ARN fără a-i afecta funcționarea corectă. Întreruperile se validează la sfîrșitul subruti-

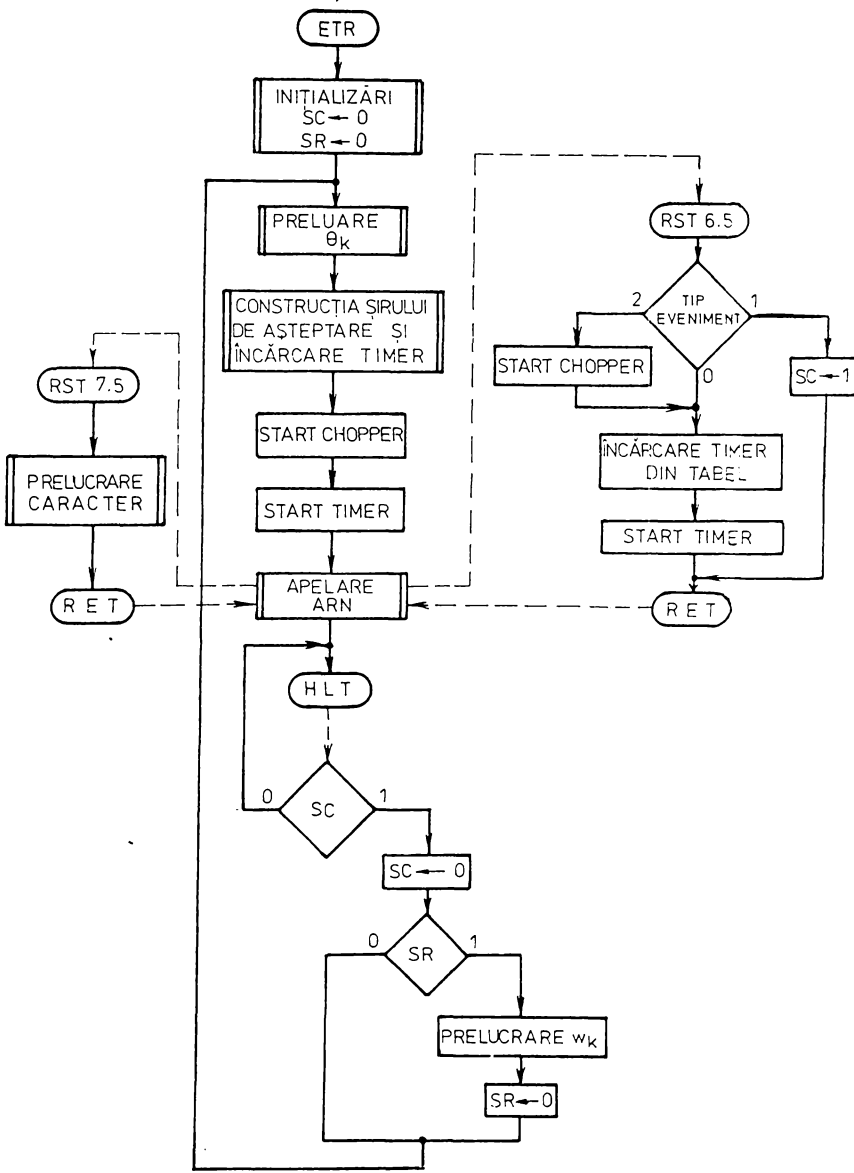


Figura 7.21. Organigrama BTR

nelor de tratare a întreruperilor, astfel încît aceste părți ale lor reprezintă o secțiune critică. De asemenea, este secțiune

critică (deci nu poate fi întreruptă) porțiunea inițială din programul principal, pînă la pornirea timerului.

#### 7.8.1.4. Preluarea poziției curente de la numărător

Citirea conținutului numărătorului, prin intermediul porturilor de intrare 0 și 1 ale sistemului, se execută la începutul fiecărei perioade de eșantionare. La citire pot apare erori, din următoarele cauze:

a) citirea unui port se efectuează în timp ce în numărător are tocmai loc o tranziție;

b) între citirile celor două porturi se produce o tranziție în starea numărătorului, în așa fel încît se schimbă și conținutul octetului deja citit (acest lucru se întîmplă cînd apare un transport între cei doi octeți).

Pentru a evita aceste erori, citirea celor două porturi se efectuează conform schemei logice din fig.7.22. Citirea se consideră corectă numai dacă ambii octeți au rămas neschimbați la două citiri succesive. Octetul mai puțin semnificativ se citește din portul 0, iar cel mai semnificativ, din portul 1.

#### 7.8.1.5. Prelucrarea referinței de poziție de la DAF

Referința de poziție este dată de către operatorul uman sub forma unui șir de 5 cifre zecimale, încheiate cu <CR>. Acest cod numeric reprezintă (în baza 10) conținutul numărătorului după atingerea stării stabile a SRA.

Recepția se face, de către programul ETR, caracter cu caracter. La apăsarea unei taste, terminalul generează o întrerupere, care activează subprogramul corespunzător din ETR. Dacă se constată recepționarea unei cifre zecimale, codul ASCII al acesteia se introduce într-un șir din memorie. Dacă s-a recepționat <CR>, se marchează sfîrșitul recepției prin SR=1. Dacă s-a recepționat <CLEAR>, se șterge ecranul. După aceste operații se revine din subrutină printr-o instrucțiune RET.

La sfîrșitul fiecărei perioade de eșantionare se verifică valoarea fanionului SR. Dacă se constată sfîrșitul unui ciclu de citire (SR = 1), se convertește șirul de caractere ASCII în numărul zecimal reprezentînd mărimea  $w_K$ , după care se reinițializează

completarea șirului.

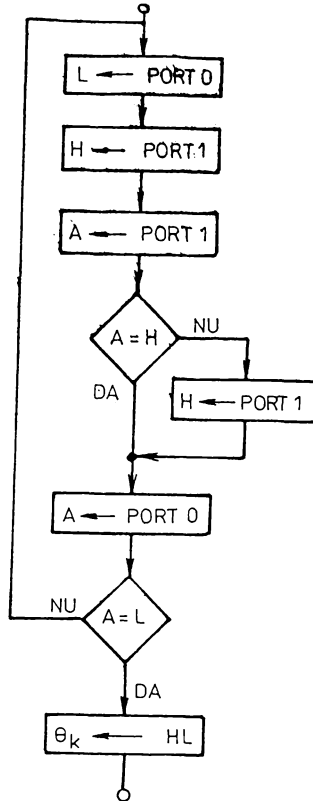


Figura 7.22. Organizarea subprogramului din ETR cu care se realizează citirea conținutului numărătorului.

Schema bloc a subprogramului de tratare a întreruperilor de la DAF este prezentată în fig.7.23., iar cea a segmentului din programul principal care realizează prelucrarea șirului de caractere, în fig.7.24. Pointerul șirului de recepție și fanionul SR sînt inițializate în cadrul segmentului de inițializări a programului ETR.

### 7.8.2. Algoritmul de reglare ARMA-1

Subprogramul implementează algoritmul descris prin schema logică din fig.7.13. În scopul simplificării calculelor, se grupează constantele care intervin în operațiile de înmulțire și

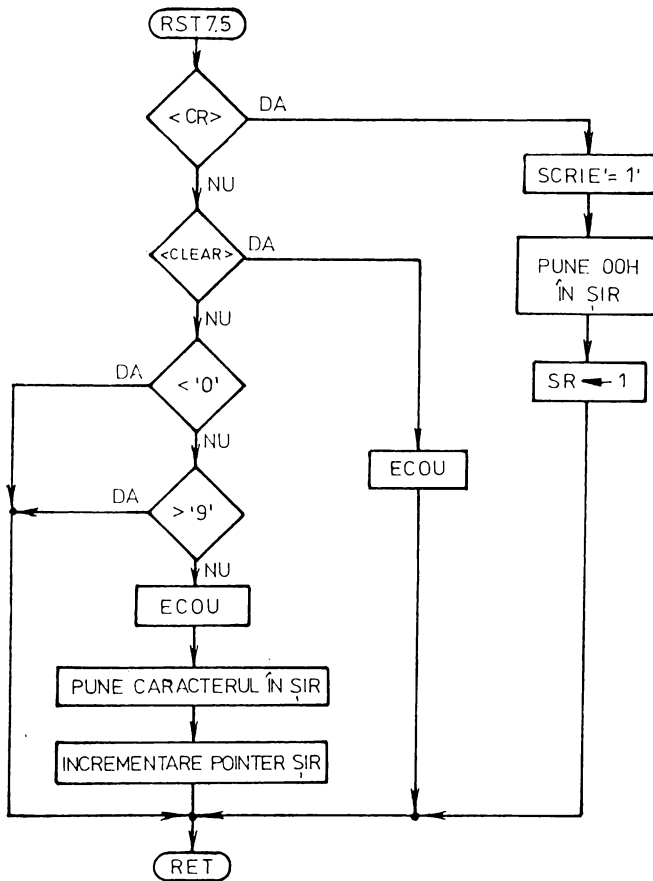


Figura 7.23. Subprogramul din ETR de citire de la DAF

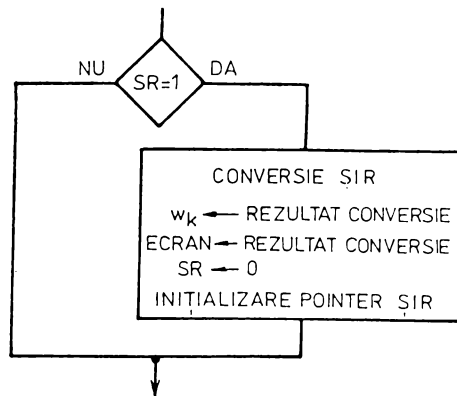


Figura 7.24. Segmentul de program din ETR care realizează prelucrarea caracterelor recepționate de la DAF



împărțire. Se fac următoarele notații:

$$\begin{aligned}
 \Omega_{K^T} &= \Omega_K^* \\
 \dot{\Omega}_{K^T} &= \dot{\Omega}_K^* \\
 s_{K^T} &= s_K^* \\
 \dot{\Omega}_{DK^d} &= \dot{\Omega}_{DK}^* \\
 \gamma_1 &= 1/(1 + k_1) \\
 \gamma_2 &= k_1/(1 + k_1)
 \end{aligned} \tag{7.128}$$

Folosind mărimile astfel definite, formulele ARN se rescriu în modul următor:

$$\begin{aligned}
 \Omega_K^* &= \theta_K - \theta_{K-1} \\
 \theta_{K-1} &\longrightarrow \theta_K \\
 \dot{\Omega}_K^* &= \Omega_K^* - \Omega_{K-1}^* \\
 \Omega_{K-1} &\longleftarrow \Omega_K \\
 v_{K-1} &= u_{K-1} - \frac{d}{T^2} \dot{\Omega}_K^* \\
 \tilde{v}_{K-1} &= \gamma_1 v_{K-1} + \gamma_2 \tilde{v}_{K-2} \\
 \tilde{v}_{K-2} &\longleftarrow \tilde{v}_{K-1} \\
 e_K &= \theta_K - w_K \\
 s_K^* &= C \cdot T \cdot e_K + \Omega_K^* \\
 \dot{\Omega}_{DK}^* &= -C \cdot d \frac{1}{T} - \Omega_K^* - P \cdot d \cdot \text{sgn} \{s_K\} \\
 u_K &= \dot{\Omega}_{DK}^* + \tilde{v}_{K-1} \\
 u_{K-1} &\longleftarrow u_K
 \end{aligned} \tag{7.129}$$

Se fac următoarele notații ale constantelor:

$$\begin{aligned}
 K_1 &= -\frac{d}{T^2} = -984 \cong -2^{10} \\
 K_2 &= C \cdot T \cdot 2^8 = 3,2 \cdot C \\
 K_3 &= P \cdot d = 0,1538 \cdot P \\
 K_4 &= -\frac{C \cdot d}{T} 2^{-4} = 0,769 \cdot C
 \end{aligned} \tag{7.130}$$

Pentru constanta care intervine în filtrul perturbației estimate, se adoptă valoarea  $k_1 = 0,6$ . În acest caz, constantele care intervin la filtrarea perturbației au valoarea:

$$\begin{aligned} \gamma_1 &= 0,625 = 5/8 = 2^{-1} + 2^{-3} \\ \gamma_2 &= 0,375 = 3/8 = 2^{-2} + 2^{-3} \end{aligned} \quad (7.131)$$

La implementarea formulelor de calcul intervin probleme specifice aritmeticii. Astfel, a fost necesară examinarea gamei de valori care intervin în algoritm și stabilirea adecvată a lungimilor de cuvânt prin care acestea sînt reprezentate. De asemenea, s-au executat scalări, pentru translatarea gamei de valori, în scopul simplificării calculelor (de exemplu, constantele  $K_2$  și  $K_4$ ). Prin aceste măsuri se evită erorile datorate trunchierilor. Lungimile de cuvânt alese sînt următoarele:

- 2 octeți pentru:  $\theta_K, \theta_{K-1}, \Omega_K^{\#}, \Omega_{K-1}^{\#}, \dot{\Omega}_K^{\#}, w_K, |e_K|, |u_K|$
- 3 octeți pentru:  $u_{K-1}, v_{K-1}, \tilde{v}_{K-2}, s_K, \dot{\Omega}_{DK}^{\#}$

Deoarece valoarea maximă a comenzii  $|u_K|$ , **corespunzătoare unui** factor de umplere unitar al tensiunii generate de chopper, este 9600H, la sfîrșitul calculelor se face trunchierea modulului lui  $u_K$  la 2 octeți.

Variabila  $e_K$  necesită 2 octeți numai pentru modul, de aceea semnul se păstrează separat pînă după realizarea primei înmulțiri, cu rezultatul pe 3 octeți. Acest rezultat va fi apoi complementat dacă semnul lui  $e_K$  a fost negativ.

Pentru creșterea vitezei de calcul, se alocă spațiu de memorie doar pentru variabilele care se păstrează de la o perioadă de eșantionare la alta:  $\theta_{K-1}, \Omega_{K-1}^{\#}, \tilde{v}_{K-2}, u_{K-1}$  și pentru cele folosite de mai multe ori în calcule:  $\Omega_K^{\#}$ . Pentru variabilele care fac legătura cu ETR, există spațiu de memorie alocat în cadrul respectivului modul de program:  $\theta_K, w_K, |u_K|, \text{sgn} \{u_K\}$ .

Detalii suplimentare sînt prezentate în comentariile din listingul programului, aflat în anexa A7. Subrutinele matematice utilizate sînt grupate într-o mică bibliotecă matematică, prezentată în anexa A7.

### 7.8.3. Algoritm de reglare ARMA-2

Schema logică a algoritmului este prezentată în fig.7.14.

În scopul grupării constantelor care intervin în înmulțiri și împărțiri, se fac următoarele notații:

$$\begin{aligned}
 \dot{\theta}_K^T &= \dot{\theta}_K^{\#} \\
 \ddot{\theta}_K^T &= \ddot{\theta}_K^{\#} \\
 \dddot{\theta}_K^T &= \dddot{\theta}_K^{\#} \\
 a_1/T &= a_1^{\#} \\
 a_2/T^2 &= a_2^{\#} \\
 a_3/T^3 &= a_3^{\#} \\
 b_1/T &= b_1^{\#} \\
 b_2/T^2 &= b_2^{\#} \\
 \dot{\theta}_{DK}^T &= \dot{\theta}_{DK}^{\#} \\
 \ddot{\theta}_{DK}^T &= \ddot{\theta}_{DK}^{\#} \\
 \ddot{\theta}_{DK}^T &= \ddot{\theta}_{DK}^{\#} \\
 K_1 &= P \cdot T = 0,0125 \cdot P
 \end{aligned}
 \tag{7.132}$$

Tinând cont și de notațiile (7.131) pentru constantele din estimatorul de perturbații, rezultă următoarele formule:

$$\begin{aligned}
 \dot{\theta}_K^{\#} &= \theta_K - \theta_{K-1} \\
 \theta_{K-1} &\longleftarrow \theta_K \\
 \ddot{\theta}_K^{\#} &= \dot{\theta}_K^{\#} - \dot{\theta}_{K-1}^{\#} \\
 \dot{\theta}_{K-1}^{\#} &\longleftarrow \dot{\theta}_K^{\#} \\
 \ddot{\theta}_K^{\#} &= \ddot{\theta}_K^{\#} - \ddot{\theta}_{K-1}^{\#} \\
 \ddot{\theta}_{K-1}^{\#} &\longleftarrow \ddot{\theta}_K^{\#} \\
 v_{K-1} &= u_{K-1} - a_3 \ddot{\theta}_K^{\#} - a_2 \ddot{\theta}_K^{\#} - a_1 \dot{\theta}_K^{\#} \\
 \tilde{v}_{K-1} &= \gamma_1 \tilde{v}_{K-1} + \gamma_2 \tilde{v}_{K-2} \\
 \tilde{v}_{K-2} &\longleftarrow \tilde{v}_{K-1}
 \end{aligned}
 \tag{7.133}$$

$$e_K = \theta_K - w_K$$

$$s_K = e_K + b_1^* \dot{\theta}_K^* + b_2^* \ddot{\theta}_K^*$$

$$\dot{\theta}_{DK}^* = -b_1^* \ddot{\theta}_K^* - b_2^* \ddot{\theta}_K^* - K_1 \cdot \text{sgn} \{s_K\}$$

$$\ddot{\theta}_{DK}^* = \dot{\theta}_{DK}^* - \dot{\theta}_K^*$$

$$\ddot{\theta}_{DK}^* = \ddot{\theta}_{DK}^* - \ddot{\theta}_K^*$$

$$u_K = a_3^* \ddot{\theta}_{DK}^* + a_2^* \ddot{\theta}_{DK}^* + a_1^* \dot{\theta}_{DK}^* + \tilde{v}_{K-1}$$

$$u_{K-1} \leftarrow u_K$$

Se adoptă valoarea  $k_1 = 0,6$  astfel încît constantele din estimatorul de perturbații au valorile:

$$\begin{aligned} \gamma_1 &= 0,625 = 5/8 = 2^{-1} + 2^{-3} \\ \gamma_2 &= 0,375 = 3/8 = 2^{-2} + 2^{-3} \end{aligned} \quad (7.134)$$

De asemenea constantele care depind de parametri procesului condus se scriu sub forma unor sume de puteri ale lui 2, astfel:

$$\begin{aligned} a_1^* &= 80 = 2^6 + 2^4 \\ a_2^* &= 966 \simeq 2^{10} - 2^6 \\ a_3^* &= 385,5 \simeq 2^8 + 2^7 \end{aligned} \quad (7.135)$$

S-au neglijat termenii de ordin mai redus, urmărind în același timp ca această neglijarea să nu introducă erori mai mari de 3%.

Pentru coeficienții  $b_1$  și  $b_2$  ai ARMA s-au adoptat astfel de valori, încît operațiile de înmulțire să se reducă la rotații și adunări.

$$\begin{aligned} b_1 &= T \\ b_2 &= T^2 \end{aligned} \quad (7.136)$$

Constanta  $k_1 = P \cdot T = 0,0125$ .  $P$  permite reprezentarea ei pe un octet (deși se alocă 2, din considerente de uniformitate a lungimii de reprezentare a constantelor care se modifică la acordare). De remarcat că această constantă nu intervine în algoritm printr-o operație de înmulțire. În consecință, s-a evitat complet utilizarea subrutinei de înmulțire, care consumă mult timp.

Lungimile de reprezentare pentru variabile sînt următoarele:

- 2 octeți:  $\hat{\theta}_K^*$ ,  $\hat{\theta}_K^*$ ,  $\hat{\theta}_K^*$ ,  $\hat{\theta}_{K-1}^*$ ,  $\hat{\theta}_{K-1}^*$ ,  $\theta_{K-1}$ ,  $w_K$ ,  $|e_K|$ ,  $|u_K|$

- 3 octeți:  $v_{K-1}$ ,  $\tilde{v}_{K-1}$ ,  $\tilde{v}_{K-2}$ ,  $s_K$ ,  $\hat{\theta}_{DK}^*$ ,  $\hat{\theta}_{DK}^*$ ,  $\hat{\theta}_{DK}^*$ ,  $u_{K-1}$

Pentru  $u_K$  și  $e_K$  sînt valabile precizările făcute în paragraful anterior. Se alocă spațiu în memorie numai pentru acele variabile care se transmit de la o perioadă de eșantionare la alta:

$\theta_{K-1}^*$ ,  $\hat{\theta}_{K-1}^*$ ,  $\hat{\theta}_{K-1}^*$ ,  $\tilde{v}_{K-2}$ ,  $u_{K-1}$  și pentru cele care se utilizează de mai multe ori în calcule sau nu pot fi păstrate pe parcurs în registrele microprocesorului:  $\hat{\theta}_K^*$ ,  $\hat{\theta}_K^*$ ,  $\hat{\theta}_K^*$ ,  $\hat{\theta}_{DK}^*$ ,  $\hat{\theta}_{DK}^*$ .

Listiugul programului se găsește în anexa A7. Comentariile din cadrul acestuia conțin detalii suplimentare asupra implementării algoritmului.

#### 7.8.4. Algoritm de reglare combinat: ARMA+PI

După cum s-a arătat în paragraful 7.3.2., se pot îmbina avantajele celor două tipuri de algoritme, dacă se utilizează ARMA pentru valori mari ale modulului erorii de poziționare  $e = \Theta - w$ , respectiv un algoritm de reglare de tip PI (ARN-PI) pentru valori mici ale erorii. Comutarea între cele două tipuri de algoritme se face conform fig.7.1. Pragul de comutare se realizează programabil, pentru a putea fi stabilit experimental la valoare optimă.

În scopul testării pe modelul experimental, s-a utilizat algoritmul de reglare numerică de tip PI, proiectat în capitolul 5. Formula de calcul a comenzii este următoarea:

$$u_K = u_{K-1} + D_0 w_K + D_1 w_{K-1} + D_2 \theta_K + D_3 \theta_{K-1} + D_4 \theta_{K-2} \quad (7.137)$$

Constantele au următoarele valori (vezi 5.4.1.):

$$D_0 = 15 = 2^4 - 2^0$$

$$D_1 = -14 = -(2^3 + 2^2 + 2^1)$$

$$D_2 = -390 \cong -(2^8 + 2^7) \quad (5.76)$$

$$D_3 = 739 \cong 2^9 + 2^8$$

$$D_4 = -350 \cong -(2^8 + 2^6 + 2^5)$$

Descompunerea în suma de puteri ale lui 2 s-a făcut pentru a înlocui înmulțirile prin câteva rotiri și adunări. Aproximările făcute introduc erori mai mici decît 4%. Semnele puterilor lui 2

au fost alese astfel încît să fie necesară o singură schimbare de semn pe parcursul calculării formulei (7.137). Calculele implementate sînt:

$$u_K = -[(2^3 + 2^2 + 2^1)w_{K-1} + (2^8 + 2^7)\theta_K + (2^8 + 2^6 + 2^5)\theta_{K-2} + w_K] + [2^4w_K + (2^9 + 2^8)\theta_{K-1} + u_{K-1}] \quad (7.138)$$

$$w_{K-1} \leftarrow w_K$$

$$\theta_{K-2} \leftarrow \theta_{K-1}$$

$$\theta_{K-1} \leftarrow \theta_K$$

$$u_{K-1} \leftarrow u_K$$

Lungimile de cuvînt pe care sînt reprezentate variabilele asigură compatibilitatea cu algoritmele de reglare modal alunecătoare descriși anterior. Actualizările exprimate în ultimele relații din (7.138) vor fi efectuate chiar în timpul calculării primei formule, în cazurile în care acest lucru este posibil. În acest mod se scurtează durata totală a calculelor.

Secvența care realizează comutarea între cele două subprograme, corespunzătoare algoritmilor de reglare, are schema logică prezentată în fig.7.25.

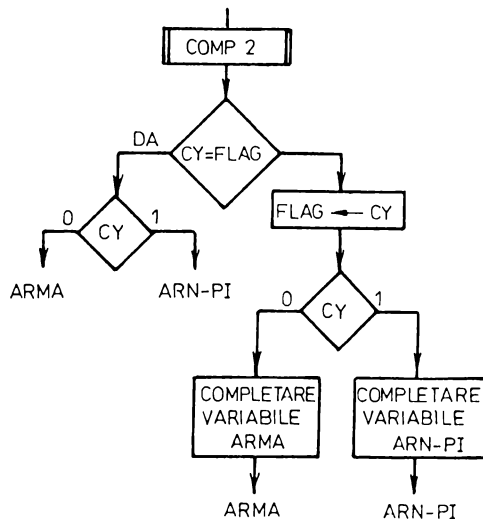


Figura 7.25. Secvența de program care realizează comutarea pe ARMA sau ARN-PI

Subrutina COMP2 furnizează valoarea "1" în fanionul CY dacă modulul erorii de poziționare este mai mic decît pragul de

comutare. Variabila FLAG servește la memorarea tipului de algoritm utilizat anterior: "0" pentru ARMA și "1" pentru ARN-PI. La schimbarea tipului de algoritm se completează locațiile de memorie ale variabilelor folosite în noul algoritm, care nu au fost actualizate de către celălalt algoritm:

a) La trecerea de la ARMA-1 sau ARMA-2 la ARN-PI:

$$\theta_{K-2} \leftarrow \theta_{K-1}$$

$$w_{K-1} \leftarrow w_K$$

b) La trecerea de la ARN-PI la ARMA-1:

$$\tilde{v}_{K-2} \leftarrow 0$$

$$\Omega_{K-1} \leftarrow 0$$

c) La trecerea de la ARN-PI la ARMA-2:

$$\tilde{v}_{K-2} \leftarrow 0$$

$$\dot{\theta}_{K-1} \leftarrow 0$$

$$\dot{\theta}_{K-2} \leftarrow 0$$

Completările cu zero se fac deoarece nu există valori apropiate disponibile; oricum, comutarea de la ARN-PI la ARMA are loc la variații bruște și cu amplitudini mari ale referinței de poziție. În regimul tranzistoriu instalat, erorile introduse pe parcursul primelor câteva perioade de eșantionare de către inițializările menționate mai sus nu sînt deranjante.

Listingul subrutinei, scrisă în limbaj de asamblare, care realizează implementarea ARN-PI și comutarea între acesta și a ARMA-1 este prezentat în anexa A8.

#### 7.8.5. Testarea prin simulare a programelor de implementare a ARN

Primul mod de testare constă în înlocuirea secvenței din programul de simulare care implementează ARN prin subrutina scrisă în limbaj de asamblare, destinată implementării ARN în cadrul modelului experimental al acționării.

Se utilizează o variantă simplificată a programului de simulare, avînd listingul prezentat în anexa A11.

Se folosește pentru procesul condus extins modelul PCE-2, deoarece simulează mai exact procesul real.

Comunicarea între modulele scrise în FORTRAN, respectiv în limbaj de asamblare, se face printr-o zonă de date comună. Variabilele prin care se comunică sînt de tip BYTE sau INTEGER. Tot în anexa All se prezintă zona de date modificată a celor 2 subprograme care implementează ARMA-1.

#### 7.8.6. Testarea pe model experimental a algoritmilor de reglare modală alunecătoare

Modelul matematic al procesului condus, folosit în proiectare reprezintă procesul real numai din punctele esențiale de vedere. În procesul real apar cupluri rezistente în axul rotorului, frecări, cuplaje imperfecte între motor și traductorul de poziție, variații ale tensiunii sursei de alimentare, rezistență de ieșire nenulă a acesteia, abateri ale parametrilor motorului de la datele de catalog ș.a. Se impune deci necesitatea de a realiza o testare pe un model real. Este necesară, totodată și acordarea algoritmilor de reglare numerică într-o situație concretă de funcționare.

În scopul testării pe model real a fost conceput și implementat un program specializat denumit TEST. Principalele facilități oferite de programul TEST sînt:

- posibilitatea de a selecta un anumit algoritm de reglare din mai multe variante, proiectate în prealabil și depozitate în memoria microsistemului de comandă;

- posibilitatea de a modifica parametrii algoritmilor de reglare, cu scopul acordării acestora;

- posibilitatea de a genera semnalele de prescriere în diferite forme de variație în timp: constantă, treaptă, rampă;

- posibilitatea de a simula prin software a unor perturbații, conform fig.7.15., cu diferite forme de variație în timp: constantă, treaptă, rampă;

- posibilitatea de lansare în execuție a programului de conducere a procesului, cu reglare automată pentru un anumit interval de timp (programabil);

- posibilitatea de afișare grafică a valorilor principalelor mărimi care interesează în proces: poziția, comanda, viteza.

Programul TEST, în variantă completă, este prezentat în Anexa A9. În cele ce urmează sînt prezentate principalele elemen-



te ale programului.

### 7.8.6.1. Interpretorul de comenzi

Pentru a se pune la dispoziția utilizatorului facilitățile anterior amintite, într-o formă convenabilă în utilizare, s-a conceput un interpretor de comenzi. Acesta prezintă pe ecranul DAF-ului diverse meniuri (liste de opțiuni) dintre care utilizatorul poate alege variantele de facilități după dorință, prin apăsarea unei taste.

Principalele meniuri sînt următoarele:

#### M1: regim de rulare

- A: acordare algoritme
- D: execuție cu prescriere de la DAF
- T: execuție în regim de testare
- Q: terminare program

Comanda A - permite modificarea valorii constantelor algoritme-

lor:  $K_2$ ,  $K_3$ ,  $K_4$  pentru ARMA-1

$K_1$  pentru ARMA-2

$K_5$  codifică tipul de ARN folosit, astfel:

- 1 - ARMA-1
- 3 - ARMA-2
- 2 - ARN-PI
- o - ARMA-1 + ARN-PI
- 4 - ARMA-2 + ARN-PI

Comanda D - lansează ETR, dar cu variabila MOD conținînd valoarea "o". În consecință, ARN va prelua referința de poziție de la DAF, ca și în cazul utilizării programelor ETR și ARN, fără TEST.

Comanda T - pune valoarea "1" în variabila MOD, corespunzător regimului de testare. În continuare, se afișează meniul M2.

#### M2: operații de testare

- G: generare tabele
- E: execuție algoritme
- A: afișare tabele
- Q: ieșire din meniu

Comanda G - activează meniul M3, pentru generarea tabelelor

Comanda E - lansează programul ETR în regim de testare. Funcțio-

narea în acest regim va fi descrisă în paragraful 7.8.6.3.

Comanda A - activează meniul M4, pentru afișarea tabelelor.

M3: generare tabele

W: referință poziție (o ... 65535)

P: perturbație

Q: ieșire din meniu

Comanda W - selectează tabelul din care se va prelua  $w_K$

Comanda P - afișează meniul M5, pentru selectarea sensului perturbației, după care se selectează tabelul pentru  $u_p$ . În continuare se efectuează un salt la secvența de program care generează tabelele, secvența descrisă în paragraful 7.8.6.3.

M4: afișare tabele

W: referință poziție

P: perturbație

T: poziție

O: viteză

U: comandă

Q: ieșire din meniu

După selectarea tabelului, se efectuează un salt la secvența de program pentru afișarea tabelelor, descrisă în paragraful 7.8.6.4.

M5: perturbație

P: în sens pozitiv

N: în sens negativ

A: în ambele sensuri

Se completează variabila SENSP, în funcție de comanda primită, astfel:

P: SENSP 0

N: SENSP 2

A: SENSP 1

Urmează saltul la secvența de generare a tabelelor.

Modul de înlănțuire a meniurilor este prezentat în fig.

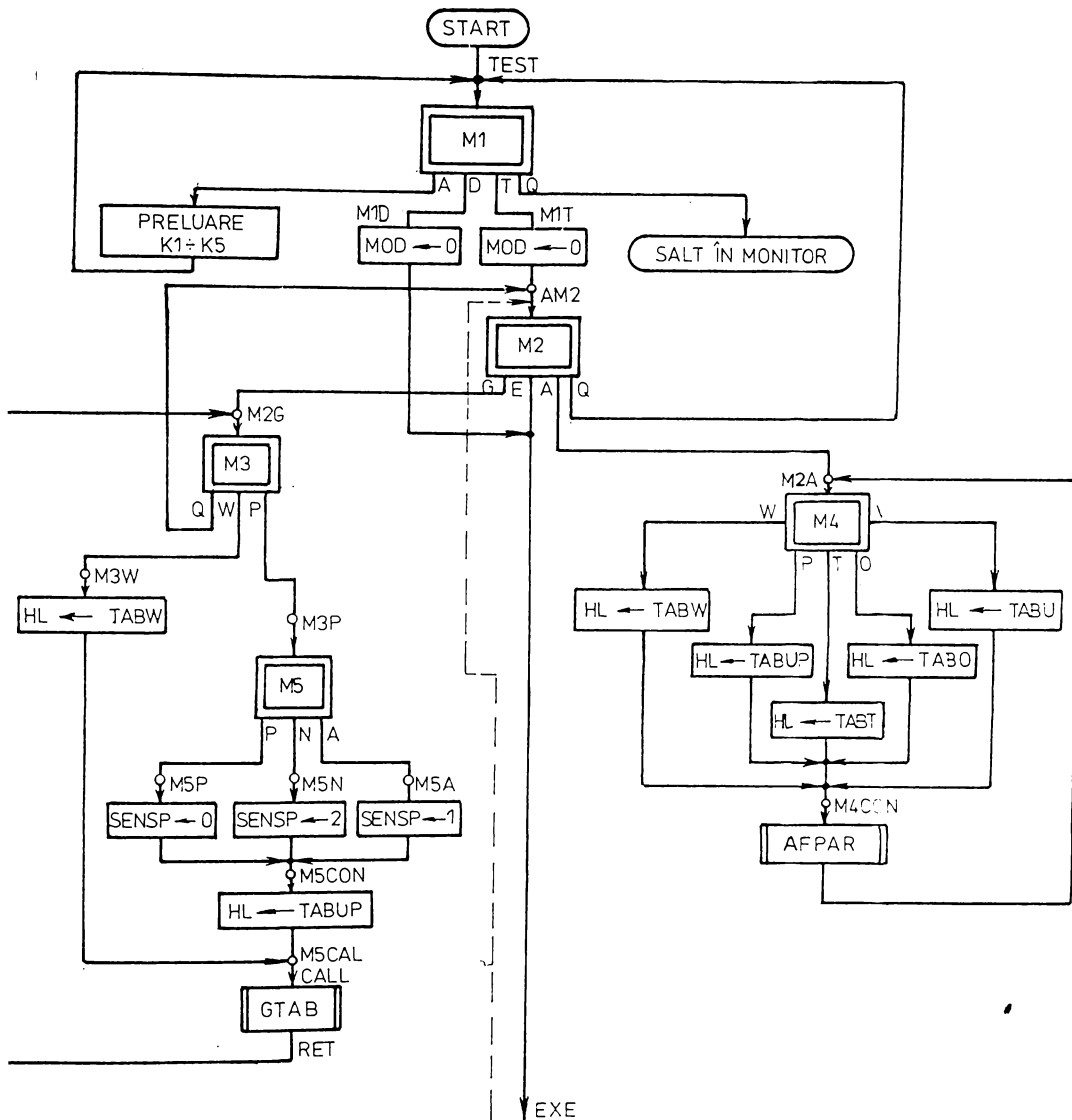


Figura 7.26. Modul de înlănțuire a meniurilor oferite de programul TEST

### 7.8.6.2. Generarea tabelor

După cum s-a arătat, programul TEST asigură generarea unor tabele care să conțină eşantioane ale unor semnale în formă de : constantă, treaptă, rampă. Valorile eşantioanelor sînt prezentate

în întreg, pe 2 octeți. Tabelul conține 200H = 512 valori, deci are o lungime de 400H octeți.

Generarea se realizează de către o subrutină care primește în registrul HL adresa de început a tabelului ( $N_1$ ). Restul parametrilor este solicitat de la operator, printr-un dialog. Pentru aceasta se afișează următorul meniu:

M6: forma

C: constantă

T: treaptă

R: rampă

În fig.7.27. se prezintă cele trei variante pentru modalitățile de generare a valorilor, pentru semnale, memorate în tabele.

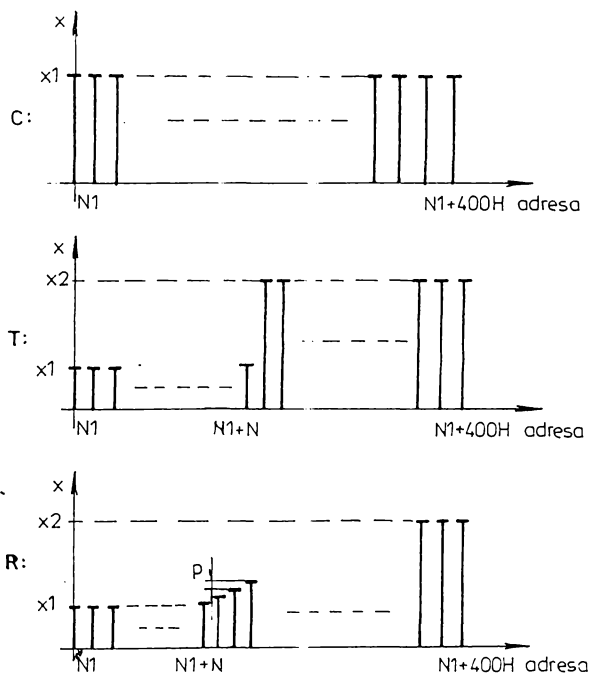


Figura 7.27. Modul de generare a valorilor memorate în tabele

Organigrama subrutinei este prezentată în fig.7.28. Prin notația (HL)  $\leftarrow x$ , se înțelege aici plasarea variabilei  $x$ , reprezentată pe 2 octeți, în locația de memorie adresată de HL

(octetul mai puțin semnificativ), și în locația următoare (octetul mai semnificativ). Celelalte notații sînt prezentate în fig. 7.27.

În cazul comenzii C, se solicită, apoi se preia de la operator valoarea (egală) a eșantioanelor ( $x_1$ ).

Dacă s-a selectat T, se solicită valoarea eșantioanelor din prima parte a semnalului ( $x_1$ ), numărul cuantei în care se generează treapta ( $N$ ), valoarea eșantioanelor din partea a doua a semnalului ( $x_2$ ).

În cazul comenzii R, se solicită: valoarea inițială ( $x_1$ ), cuanta în care începe rampa ( $N$ ), panta rampei ( $P$ ), valoarea finală ( $x_2$ ).

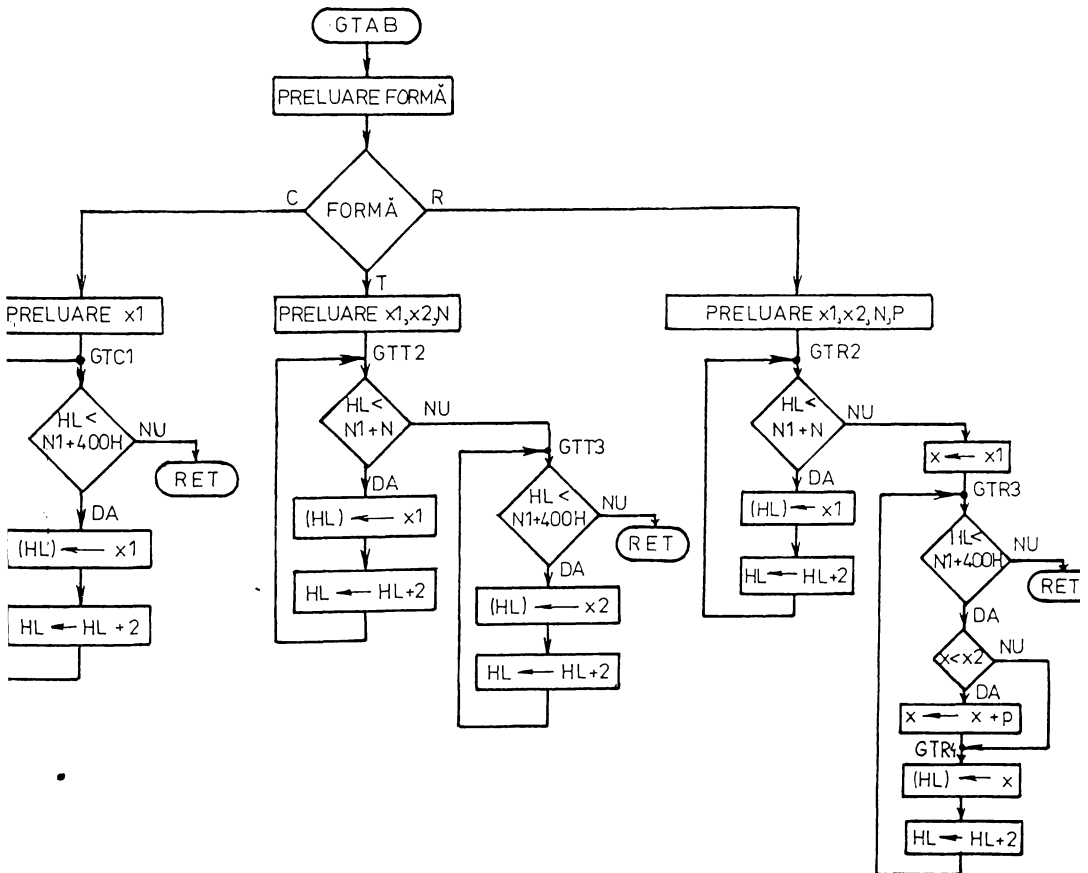


Figura 7.28. Subrutina de generare a tabelor

7.8.6.3. Rularea programului de implemmentare a ARN în regim de testare

Regimul de testare diferă de regimul de rulare obișnuit, cu prescriere de la DAF, prin următoarele: referința de poziție se preia din tabelul special construit în prealabil; se poate simula software o perturbație, ale cărei valori sînt conținute într-un alt tabel; valorile pe care le i-au poziția, viteza, comanda în fiecare perioadă de eșantionare se păstrează în tabel, iar conținutul acestora poate fi afișat, sub formă de grafice.

Modul de interfațare între programele TEST, ETR și ARN este prezentat în fig.7.29. Se prezintă mai jos succesiunea acțiunilor executate de cele 3 programe.

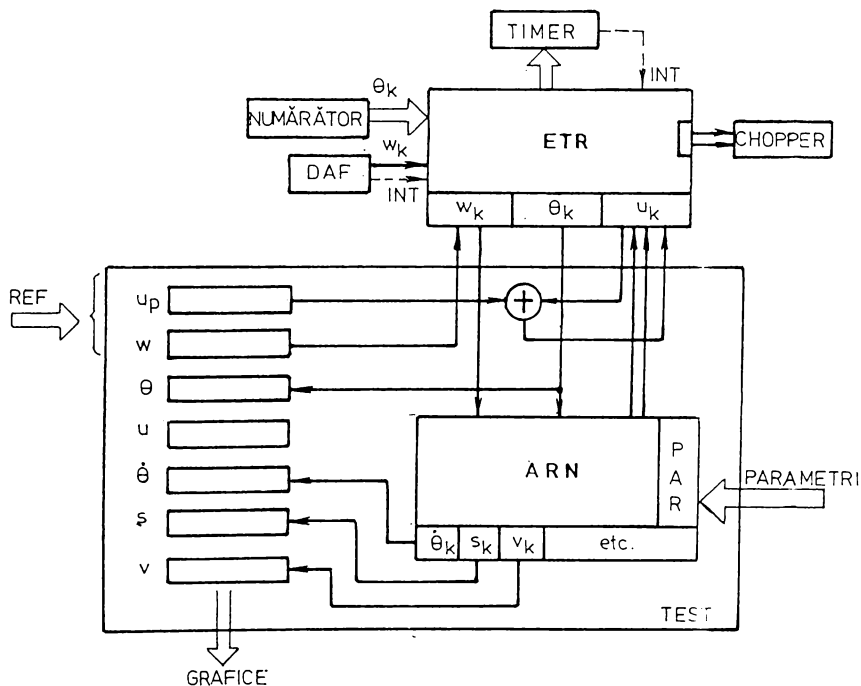


Figura 7.29. Interconexiunea programelor TEST, ETR și ARN

Programul TEST lansează ETR. Acesta efectuează operațiile de gestiune a timpului real și apelează, în mod normal, programul de implementare a ARN. În cazul regimului de testare, apelul se face la o secvență din programul TEST. Această secvență plasează în locația  $w_k$  a ETR o valoare preluată din tabel, după ca-

re apelează subprogramul ARN. ARN calculează valoarea comenzii, în modul și semn, completînd locațiile UK și US ale ETR.

Revenirea din ARN se face tot în TEST. Acesta modifică valoarea comenzii, prin însumarea unei mărimi  $u_p$ , care simulează software o perturbație apărută în SRA, conform fig.7.15. Se modifică valoarea comenzii, în funcție de sensul perturbației, astfel:

$$\text{- sens pozitiv: } u_K \longleftarrow u_K + u_{pK}$$

$$\text{- sens negativ: } u_K \longleftarrow u_K - u_{pK}$$

$$\text{- ambele sensuri: } |u_K| \longleftarrow |u_K| - u_{pK}$$

Perturbația care se manifestă în ambele sensuri este o perturbație de frînare. Se va ține cont că, în cadrul programului ETR, mărirea  $u_K$  este reprezentată prin modul și semn.

În continuare, TEST, completează tabelele cu valorile curente ale variabilelor  $\theta_K$ ,  $\dot{\theta}_K^*$  (sau  $\Omega_K^*$ ),  $u_K$ , după care se revine în ETR printr-o instrucțiune RET.

TEST trebuie să realizeze și numărarea perioadelor de eșantionare afectate pentru experiment, corespunzătoare numărului maxim de poziții în tabel (care este 200H). Pentru executarea în bune condiții a experimentării, s-a prevăzut și următoarea facilitate: la lansarea în execuție, în regim de testare, programul TEST nu incrementează pointerii tabelelor, furnizînd în fiecare perioadă de eșantionare valoarea referinței  $w$  conținută în prima linie a tabelului respectiv. Devine astfel posibilă stabilizarea buclei de reglare în respectiva poziție. Cînd operatorul observă stabilizarea apasă tasta S. Subprogramul de tratare a întreruperilor de la DAF detectează acest caracter și pune valoarea "1" în variabila START. Din acest moment, programul TEST incrementează pointerul tabelelor (STAB) la fiecare nouă perioadă de eșantionare, după care verifică dacă s-a ajuns la sfîrșitul tabelor (STAB = 200H) caz în care transferă controlul interpretorului de comenzi, la meniul M2, pentru a face posibilă afișarea tabelor.

După aceea, se poziționează pe zero porturile de ieșire care comandă chopperul și se realizează oprirea motorului. De asemenea, este necesară refacerea stivei (care conține adresa de întoarcere plasată de instrucțiunea CALL din programul ETR), precum și a adresei de salt în cazul întreruperii de la DAF, astfel încît controlul dialogului cu operatorul să fie redat

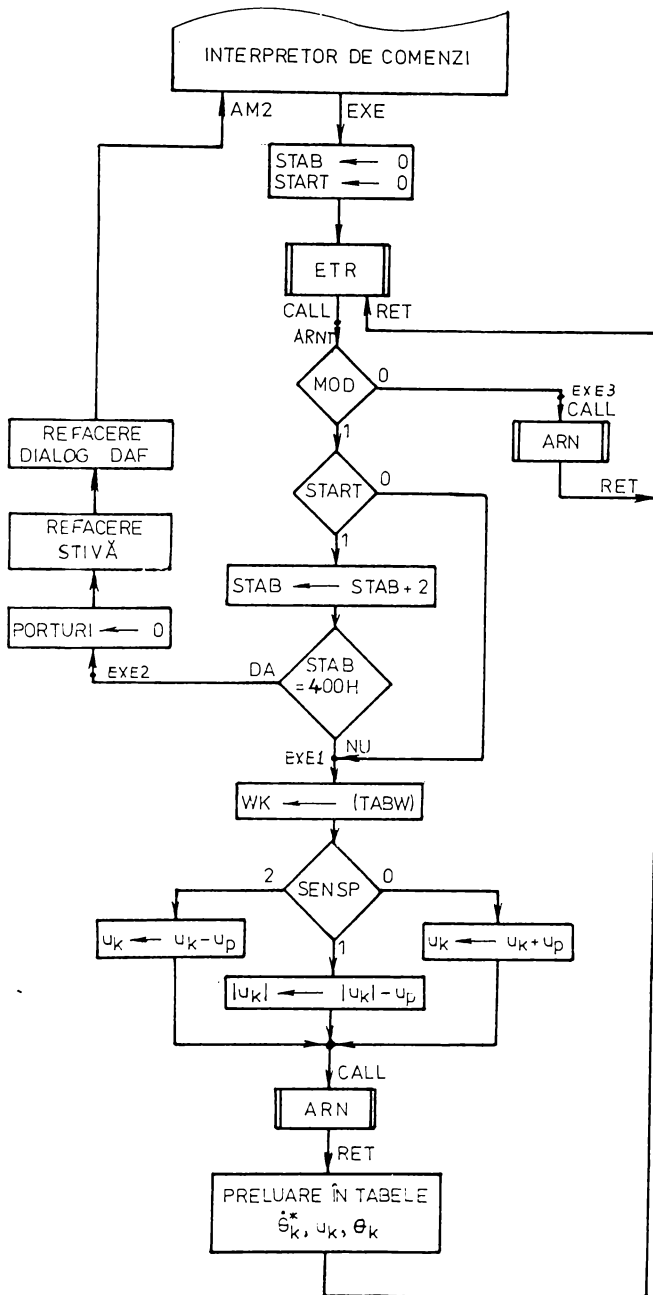


Figura 7.30. Organigram secțiunii din programul TEST care realizează rularea în regim de test



programului monitor.

Organigrama porțiunii din programul TEST care realizează rularea în regim de testare este prezentată în fig.7.30.

Subrutina de trasare a graficelor funcțiilor reprezentate în tabele consideră valorile acestor funcții în valoare absolută, în domeniul  $0 - 2^{16}$ . Intrucît mărimea  $\dot{\Theta}_K^{\#}$ , reprezentată în complement față de 2, poate fi atât pozitivă, cât și negativă, în tabel se va memora  $\dot{\Theta}_K^{\#} + 8000H$ . Se realizează astfel o translatore a graficului, axa timpului fiind plasată în dreptul valorii 8000H.

Mărimea  $u_K$  are modulul reprezentat pe 2 octeți, în variabila UK, și semnul reprezentat separat în variabila US. Pentru a o putea reprezenta, în tabel, pe 2 octeți, se va calcula valoarea:

$$u_K/2 + 2^{15} = (|u_K|/2) \cdot \text{sgn} \{u_K\} + 8000H \quad (7.139)$$

care va fi memorată în tabel. Se realizează astfel "comprimarea" la jumătate a modulului funcției și translatore pe ordonată cu 8000H.

La interpretarea graficelor pentru  $\dot{\Theta}_K^{\#}$  și  $u_K$  se va ține cont de modificările efectuate conform celor de mai sus.

Pentru a se putea realiza selectarea uneia dintre cele 5 variante de algoritme de reglare, în funcție de valoarea constantei  $K_5$ , (vezi 7.8.6.1.) s-a elaborat o scurtă secvență de program, începînd de la adresa ARNC, care transferă controlul uneia sau alteia dintre subrutinele care implementează cele 5 variante.

Pentru a se asigura rularea împreună a programelor ETR, TEST și subprogramelor de implementare a ARN, este necesar ca, în cadrul programului ETR, apelul subrutinei ARN să fie înlocuit cu instrucțiunea CALL ARNT.

#### 7.8.6.4. Afișarea tabelelor

Pentru analiza formei de variație în timp a mărimilor care intervin în algoritmul de reglare, este necesară afișarea sub formă de grafice a conținutului tabelelor completate de către programul TEST. Afișarea se realizează cu subrutina AFIS, care primește în registrul HL adresa de început a tabelului selectat. Lungimea tabelului este fixă (512 numere, pe 2 octeți fiecare). Cei-lalți parametri necesari afișării sînt preluați de către subrutina de la operatorul uman prin intermediul DAF-ului.

Terminalul de tip DAF loop, conectat la sistem, permite

afișarea numai cu caractere alfanumerice. Pe ecran se afișează 20 de linii, a câte 50 de caractere fiecare. Aceste limitări au impus utilizarea unei forme simplificate de afișare. Astfel, axa timpului se plasează pe verticală, de sus în jos, iar axa valorilor mărimii tabelate (pe care o vom nota  $Oy$ ), pe orizontală, de la stînga la dreapta. Fiecare linie de pe ecran va reprezenta o locație din tabel. În partea stîngă a liniei se înscrie valoarea conținută în locație, în zecimal. Restul liniei conține: un caracter "I" la începutul și sfîrșitul porțiunii rezervate graficului, precum și în dreptul valorii zero; un caracter "≠", la o distanță, față de margine, proporțională cu valoarea numerică conținută în locația din tabel. De exemplu, o linie poate fi astfel afișată:

"01597 I I ≠ I"

Prin alăturarea pe verticală a liniilor, rezultă graficul corespunzător tabelului selectat.

Datorită lungimii mari a tabelului (512 valori), s-a prevăzut posibilitatea de "comprimare" a reprezentării pe axa timpului, prin micșorarea numărului de eșantioane din tabel care se afișează. Astfel, nu se mai reprezintă fiecare locație din tabel pe o linie, ci, între locațiile reprezentate pe 2 linii succesive, se omit  $K-1$  locații. Valoarea lui  $K$  se introduce de către operatorul uman, la începutul operației de afișare, ca răspuns la mesajul:

" Afișare din  $K$  în  $K$  valori "

Tot în scopul unei afișări flexibile, s-a prevăzut și posibilitatea deplasării ferestrei de vizualizare de 19 linii, de-a lungul axei timpului. După afișarea unui ecran, prin comanda

$C < NN \dots N > < CR >$

se determină afișarea unui nou ecran, cu aceeași valoare a parametrelui  $K$ , dar în care prima linie va corespunde locației cu numărul  $NN \dots N$ . Aici  $NN \dots N$  este un număr de  $1 \div 5$  cifre, în zecimal.

S-a prevăzut, de asemenea, posibilitatea de a crea o fereastră de vizualizare pe axa  $Oy$ . Pentru a putea scrie simplu, în limbaj de asamblare, subrutina de afișare, s-a elaborat un algoritm care utilizează doar înmulțiri și împărțiri cu puteri ale lui 2. Se împarte domeniul maxim de variație pe  $Oy$  ( $0 - 2^{16}$ ) în  $2^N$  intervale,  $N = 0 \div 11$ . Se afișează intervalul cu numărul de ordine  $M$ ,  $M = 1 \div 2^N$  (numerotarea se face de la stînga la dreapta). Valorile  $N$  și  $M$  se dau de către utilizator, la începutul

operației de afișare a unui tabel, ca răspuns la mesajele:

" nr.intervale pe Oy = 2 ~~##~~ N "      N =

" afișare interval M "      M =

Pentru a ieși din regimul de afișare, după ce ecranul DAF-ului este completat, se folosește comanda Q. Aceasta determină afișarea meniului M4, permițînd afișarea altui tabel. Aceeași comandă se utilizează și atunci cînd se dorește afișarea aceluiași tabel, dar cu alte valori ale parametrilor K, M, N.

## CAPITOLUL 8.

### CONCLUZII SI CONTRIBUTII

#### 8.1. Concluzii și direcții de cercetare

Automatizarea flexibilă, axată pe linii și celule flexibile de fabricație, încadrate în sisteme complexe, s-a impus ca necesitate, astăzi, în procesele industriale și reprezintă, cu certitudine, o direcție de dezvoltare în viitor.

În sistemele moderne de fabricație roboții industriali ocupă și vor ocupa, din ce în ce mai mult, poziții centrale determinante. Efortul, prin excelență pluridisciplinar, pentru perfecționarea și modernizarea roboților industriali, reprezintă, în zilele noastre, o direcție majoră în cercetare pentru specialiști din cele mai diverse domenii de activitate ca: Teoria mecanismelor, Organe de mașini, Tehnologia construcțiilor de mașini, Acționări electrice și hidropneumatice, Electronică, Automatică, Construcția și programarea calculatoarelor, Organizarea întreprinderilor industriale, Management, Ergonomie, dacă se face o enumerare care nu este exhaustivă.

Teza de doctorat elaborată abordează aspecte esențiale ale sistemelor de acționare moderne și aduce contribuții la optimizarea acestora. Lucrarea se aliniază eforturilor conjugate ale cercetărilor, de cele mai diverse specializări, din domeniul complex și nelimitat al roboticii, impulsionat de tendința perpetuă de creștere a performanțelor roboților.

Preocupările tezei, din domeniul sistemelor de acționare electrică, sînt în consens cu deosebit de frecvența utilizare a acestora în conducerea roboților industriali, dar și cu o largă arie de răspîndire a acționărilor electrice în industrie, în general.

În condițiile tehnice de astăzi, la 10 - 15 ani după apariția și răspândirea microprocesoarelor, această familie de circuite a devenit esențială în structura echipamentelor de conducere a roboților industriali. Sistemele de comandă pentru roboți sînt implementate pe echipamente de calcul. Comanda numerică solicită și adaptarea structurii sistemelor de acționare - compuse din sistemele de reglare aferente fiecărei axe - la tehnica numerică. În consecință, tendința actuală, în robotică, este de a se realiza sisteme de acționare "exclusiv numerice" și de a înlocui elementele analogice din componența lor. În acest fel se asigură un caracter unitar mijloacelor de prelucrare a informației la care se adaugă și o serie de avantaje menționate în capitolul introductiv al tezei.

Lucrarea de față aduce, prin conținutul său, contribuții la modalitățile de utilizare a microprocesoarelor în comanda sistemelor de acționare electrică, realizarea unor sisteme de conducere nemijlocită a cuplelor cinematice conducătoare ca sisteme de reglare automată, în variante originale "exclusiv numerice". Rezultatele obținute aduc contribuții la simplificarea structurilor hardware folosite în comanda sistemelor de acționare cu motor de curent continuu - cel mai frecvent utilizate în cazul roboților industriali. Pentru aceste sisteme sînt elaborate soluții originale cu care se obțin performanțe competitive la prețuri de cost minime și disponibilități de aprovizionare accesibile. Pe parcursul tezei de doctorat sînt propuse numeroase metode de exploatare eficientă a microsistemelor uzuale. Se demonstrează că prin efort de inteligență (software) se pot obține performanțe superioare cu aceste microsisteme, prin exploatarea la maximum a resurselor lor. Aplicațiile prezentate arată că a fost posibil ca la nivelul unui singur microprocesor uzual să fie sintetizate toate operațiile, comenzile și calculele implicate în conducerea unor sisteme de acționare de curent continuu.

Ca principal argument justificativ al direcției urmate în cercetare, în lucrare se obține cu numai trei circuite integrate un sistem de comandă complet pentru acționarea unei axe. Sistemul conceput asigură reglarea vitezei și poziției, cu limitarea curentului din înfășurarea rotorică a motorului de curent continuu, folosind mijloace originale, exclusiv software. În aceeași

configurație minimală s-a realizat și un sistem de conducere după viteză pentru trei axe simultan. Soluțiile de implementare au la bază microprocesorul 8085 și recomandă folosirea acestui circuit în acționări.

Sistemele de comandă realizate asigură performanțe foarte bune în reglare și conferă flexibilitatea pretinsă în robotică, prin utilizarea de algoritme numerice, ușor de acordat, sau chiar de schimbat, în funcție de tipul sistemului condus sau la modificări survenite în proces.

Pe aceeași structură minimală au fost implementate atât algoritme de reglare clasice cât și algoritme moderne, de tip modal alunecător. Pentru acestea din urmă a fost concepută și o metodologie de proiectare și de acordare. A fost de asemenea elaborată o metodă originală de combinare a două algoritme diferite în ideea însumării efectelor lor pozitive în reglare.

În lucrare este redat, în detaliu, aparatul matematic folosit pentru modelarea și proiectarea sistemelor de reglare realizate. Sînt prezentate, critic, rezultatele obținute în experimentările efectuate pe un stand anume construit.

Implementările software folosite apelează la tehnici specifice programării în timp real. Sînt detaliate soluțiile originale găsite în organizarea structurată pe funcții distincte a programelor. Se subliniază modalitățile de rezolvare a problemelor referitoare la o gestionare judicioasă a informațiilor vehiculate și desfășurarea în succesiune corespunzătoare a operațiilor.

Teza de doctorat cuprinde 8 capitole, extinse pe 268 pagini, are inserate 135 de figuri și face referiri la un număr de 107 titluri bibliografice. Sînt atașate și un număr de 177 pagini listate pe calculator ce reprezintă 12 anexe cu programele elaborate în limbaj de asamblare, programele de simulare folosite în proiectare și rezultate experimentale.

În capitolul introductiv se prezintă, în sinteză, nivelul la care a ajuns și spre care tinde automatizarea flexibilă a proceselor industriale, rolul roboților industriali astăzi și în uzina viitorului. Se subliniază aria largă a cercetărilor implicate în robotică, accentuîndu-se pluridisciplinaritatea domeniului. Se specifică modul în care se încadrează preocupările din teză, cu privire la optimizarea sistemelor de acționare, în ro-

botică.

Avantajele utilizării sistemelor de acționare electrică a roboților industriali sînt considerate ca fiind unanim acceptate și valorificate printr-o foarte frecventă utilizare în realizările practice. În circuitele de comandă pentru acționările electrice se remarcă, tot mai frecvent, tendința de a se introduce microprocesoarele și de a se dezvolta sisteme "exclusiv numerice". Această tendință este cunoscută și în țările cu acces limitat la componente foarte avansate, unde eforturile se fac pentru a atinge performanțe competitive prin exploatarea software, la maximum, a posibilităților microprocesoarelor uzuale.

Sinteza bibliografică realizată în capitolul 2 al tezei aduce contribuții la sistematizarea și ordonarea principalelor aspecte privind problema conducerii roboților industriali.

Pornind de la structura bloc a unui sistem de conducere pentru roboți industriali, în lucrare se propune o împărțire în trei grupe de sarcini a ansamblului de funcții și operații ce îi revin spre execuție. Cele trei grupe, îndeplinite de blocuri dedicate, dar aflate într-o interconexiune organică, sînt:

- A. Modelarea mediului exterior
- B. Specificarea, generarea și controlul mișcărilor
- C. Dialogul cu operatorul uman și integrarea într-o rețea de calculatoare

Grupa B de sarcini realizează conducerea propriu-zisă a robotului, iar aceasta constă, în esență, din două aspecte, numai aparent distincte:

- a) asigurarea trecerii punctului caracteristic prin puncte calculate sau impuse, exprimate fie în coordonate carteziene (cu matricea de situare) fie în coordonatele poziționale relative **ale elementelor cuplelor cinematice conducătoare;**
- b) execuția unei deplasări line între punctele prescrise.

Se prezintă legătura spațiu-timp care se realizează în conducere și principiul sintetizării traiectoriei descrise de punctul caracteristic cu funcții conducere de timp. Se propune abordarea problemei conducerii în conformitate cu cele două etape de execuție:

1. Specificarea și generarea parametrilor cinematici ai mișcării
2. Conducerea nemijlocită a cuplelor cinematice conducătoare

toare ale dispozitivului de ghidare.

Prima etapă a conducerii este realizată de blocul denumit generator de traiectorie, pentru care sînt prezentate modurile de operare în funcție de opțiunea pentru metoda de conducere și de coordonatele urmărite în mișcare.

A doua etapă a conducerii, cea propriu-zisă, este executată de sistemul de acționare al robotului, compus din sistemele de reglare automată de la nivelul fiecărei cuple cinematice conducătoare. Acesta este domeniul în care se încadrează cercetările prezentate în ansamblul lucrării.

Sinteza bibliografică este continuată cu principalele aspecte ale sistemelor de acționare electrică pentru roboți industriali și analizează stadiul actual de dezvoltare a acestora. Se evidențiază rolul microprocesoarelor în noile structuri elaborate și tendința de a se implementa sisteme de acționare "exclusiv numerice". În finalul capitolului 2 este propusă, la nivel de schemă bloc, structura minimală pînă la care se pot optima sistemele de acționare electrică, structură spre care converg rezultatele implementărilor expuse ulterior în lucrare.

În cel de-al treilea capitol al tezei este prezentată arhitectura standard pentru un microsistem uzual pe 8 biți, ce poate fi utilizată cu circuite de interfață simple în comanda acționărilor electrice. Sînt descrise trei configurații realizate practic, fiecare corespunzînd unei familii de microprocesoare. Aceste microsisteme au fost construite cu scopul direct de a fi folosite în cercetarea experimentală, pentru verificarea soluțiilor de optimizare propuse pe parcursul tezei.

Se evidențiază varianta cu microprocesor 8085, recomandată în lucrare ca fiind potrivită, cu deosebire, pentru aplicație în sistemele de acționare cu motor de curent continuu. Argumentele justificative pentru această recomandare sînt aduse în primul rînd de dotările proprii microprocesorului 8085: cinci nivele de întrerupere hardware, intrare și ieșire serială de date, circuite interne de interfață cu magistralele, frecvență de lucru suficient de mare. La aceste utilități se adaugă și posibilitatea de a construi un microsistem complet dacă la microprocesor se conectează numai încă două circuite integrate: 8155 și 8355 (8755). Cele două circuite completează sistemul cu un timer, o capacitate de memorie suficientă și un număr de linii



intrare-ieşire acoperitor pentru controlul complet al unui proces de complexitate medie. Din punctul de vedere al numărului de conexiuni, comenzi și operații pe care le presupun și sistemele de acționare de curent continuu se situează la nivel mediu de complexitate.

Avînd în vedere importanța sistemelor de acționare cu motoare de curent continuu pentru roboți industriali, în capitolul următor sînt abordate principalele elemente specifice acestora și problematica sintetizării numerice a comenzilor ce le asigură funcționarea.

Motivul principal care susține utilizarea motorului de curent continuu este că acesta conferă flexibilitate sistemelor de acționare, prin cele două avantaje majore aduse de specificul funcționării sale: permite modificarea turației în limite largi, în condițiile menținerii cuplului la o valoare ridicată și suficient de constantă, iar modificarea turației se poate realiza relativ ușor, cu variația tensiunii de alimentare. În consecință, structura sistemelor de acționare cu motor de curent continuu este mai simplă, rezumîndu-se, în principal, la folosirea redresoarelor comandate și a chopperelor cu circuite electronice de putere ce asigură controlul tensiunii la bornele motorului.

Pe parcursul capitolului 4 sînt prezentate cîteva exemple, concepute de autor, pentru modul în care se poate realiza comanda cu microprocesor a redresoarelor și chopperelor. Sînt expuse, în paralel, versiuni implementate pe fiecare dintre cele trei variante de microsistem descrise în capitolul al treilea. Sînt analizate rezultatele experimentale obținute și se subliniază avantajele folosirii microprocesoarelor în raport cu procedeele de comandă tradiționale:

- simplificarea structurilor hardware pentru schemele de comandă;
- o plajă mai largă de valori sintetizabile pentru turația motorului;
- modificarea turației cu o rezoluție mai fină;
- facilitatea implementării ș.a.

În același capitol, cîteva exemple de aplicație originale demonstrează că microprocesoarele uzuale dispun de suficiente rezerve și pentru implementarea unor funcții mai complexe decît aceea de comandă nemijlocită a circuitelor de putere. Mijloace de extindere a capacității de operare se obțin cu precădere prin

utilizarea judicioasă a sistemului de întreruperi și a generatorului de timp real.

În ultimii ani, odată cu producerea tranzistoarelor bipolare și MOSFET de putere mare, tendința în acționările de curent continuu este de a se generaliza folosirea variatoarelor construite cu tranzistoare funcționând pe principiul modulării în lățime de puls (pulse width modulation). În consens cu această tendință, în finalul celui de al patrulea capitol al tezei se prezintă principiul PWM și modul de implementare al acestuia în variantă hibridă, **numeric-analogică**, respectiv în variantă "exclusiv numerică". Se accentuează avantajele celei de a doua variante și este analizat critic un sistem de acționare "exclusiv numeric" preluat din literatura de specialitate.

Mijloacele simple ce le conferă pentru sinteza semnalelor de comandă și capacitatea de operare suficient de mare a microprocesoarelor au condus la ideea de a implementa, la nivelul lor, și funcții mai complexe dintre cele solicitate de comanda sistemelor de acționare. Mai mult decât atât, s-a pus problema de a se obține, în variantă "exclusiv numerică", un circuit de comandă complet, folosind numai un microsistem uzual în configurație minimală.

În acest sens, în capitolul cinci al tezei se propun două structuri originale de sistem de reglare numerică automată pentru o acționare cu motor de curent continuu.

Realizate în jurul câte unui microprocesor uzual pe opt biți, sistemele propuse reprezintă versiunile concrete ale structurii optimale de principiu anticipată în capitolul 2. Cele două sisteme se recomandă ca fiind foarte potrivite și pentru conducerea nemijlocită a unei cuple cinematice conducătoare dintr-un robot industrial.

Configurația hardware propusă conține un număr minim de elemente. Circuitul de comandă este implementat cu microsistemul uzual. În varianta cu microprocesor 8085 acesta este alcătuit din numai trei circuite integrate: 8085, 8155 și 8355 (sau 8755). În afara microsistemului nu este necesar decât un chopper, un motor, un traductor incremental de **deplasare și sursele de alimentare** pentru a se obține un sistem de acționare complet.

Cu microprocesorul sînt implementate toate funcțiile necesare pentru comanda și controlul întregului sistem:

- (F1) Preluarea mărimii prescrise (viteză, poziție) de la un sistem ierarhic superior.
- (F2) Determinarea mărimii curente de reacție din informația furnizată de traductorul incremental, cuplat cu axul motorului.
- (F3) Calculul codului numeric normat al mărimii de comandă pe baza unui algoritm de reglare.
- (F4) Conversia mărimii de comandă în interval de timp de conducție pentru chopper.
- (F5) Comanda propriu-zisă a chopperului.
- (F6) Limitarea curentului din înfășurarea de comandă a motorului.

Toate cele șase funcții se execută pe durata fiecărei perioade de eșantionare aleasă pentru procedeul discret de conducere a sistemului. Sintetizarea tuturor funcțiilor în timp limitat presupune o judicioasă exploatare a capacităților de operare ale microprocesorului uzual.

Se prezintă un sistem de reglare numerică automată a vitezei (SRAN-V) și un sistem de reglare automată numerică a vitezei și poziției (SRAN-VP) comandate cu microprocesorul 8085.

În cadrul SRAN-V a fost implementat un algoritm numeric PI pentru reglarea turației motorului, iar în cel de-al doilea sistem s-a folosit un algoritm P în reglarea poziției și un algoritm PI pentru viteză. Sînt detaliate modelele matematice stabilite pentru componentele de sistem și etapele de proiectare - bazată pe metoda caracteristicilor de frecvență ale sistemului deschis - cu aplicație numerică pentru un anumit tip de motor utilizat practic. Se subliniază faptul că acordarea sau modificarea parametrilor reglatoarelor se poate realiza ușor prin simpla schimbare a unor coeficienți numerici din formula finală obținută pentru algoritmele de reglare.

S-a reușit elaborarea unei configurații hardware numerice și au fost excluse complet circuitele analogice, respectiv convertoarele numeric-analogice. În plus, informația pentru valoarea vitezei curente de deplasare se obține prin calcul, din informația de poziție, ceea ce a permis eliminarea traductorului de viteză, dedicat, din structură. Limitarea curentului din înfășurarea rotorică a motorului se realizează cu un procedeu original conceput prin care s-a exclus și traductorul de curent.

Implementarea software este prezentată în detaliu în lucrare. Aceasta se bazează pe utilizarea timerului - la generarea intervalelor de timp aferente evenimentelor din proces și conversia cod numeric-timp necesare în conducere - și a sistemului de întreruperi hardware ale microprocesorului 8085. Pentru fiecare sistem a fost conceput câte un program principal cu sarcina de a stabili succesiunea și momentele de lansare în execuție a celor șase funcții ce trebuie sintetizate. Se asigură astfel o gestiune judicioasă a duratei perioadei de eșantionare. Programul principal face apel la subrutine dedicate, anume concepute, pentru:

- determinarea mărimii de reacție (prin numărarea impulsurilor generate de traductorul incremental);
- calculele aritmetice conforme algoritmilor de reglare;
- dialogul on-line și off-line cu sistemul ierarhic superior;
- comanda nemijlocită a electronicii de putere.

Ambele sisteme au fost verificate și practic. În acest scop s-a folosit un stand de încercări realizat de autor pentru studiul experimental al sistemelor numerice de acționare cu motor de curent continuu. Standul a fost construit în jurul sistemului de dezvoltare cu microprocesor 8085 prezentat în paragraful 3.2.2. al tezei.

În cadrul experimentărilor s-au efectuat măsurători asupra principalilor indicatori de calitate ai reglării, în regim dinamic și staționar. Performanțele obținute, chiar în condițiile întrebuintării unui motor cu calități modeste, sînt competitive și recomandă utilizarea în robotică a variantelor de sistem propuse.

În concluzie, pe un microsistem compus din numai trei circuite integrate s-a reușit implementarea tuturor funcțiilor aferente conducerii unui sistem de acționare cu motor de curent continuu. Pe baza acestui microsistem ieftin și fiabil au fost realizate sisteme de acționare cu performanțe foarte bune, într-o structură exclusiv numerică, redusă la un număr minim de componente. Sinteza acestor structuri a fost posibilă prin elaborarea unor metode originale cu care se exploatează optimal mijloacele microprocesorului 8085. Principiile utilizate în concepție, procedeele originale propuse și experiența dobîndită în aceste realizări pot fi cu ușurință aplicate și la alte tipuri de microprocesoare sau controlere folosite în comanda sisteme-

lor de acționare.

În capitolul 6 al tezei de doctorat se prezintă alte contribuții ale autorului la optimizarea sistemelor de acționare. Pe parcursul acestui capitol sînt propuse soluții originale prin care microsistemele uzuale sînt și mai eficient exploatate, ceea ce permite utilizarea lor în structurarea unor noi variante de sisteme de acționare.

În acest sens, cu microsistemul utilizat și în capitolul precedent s-a reușit implementarea unui sistem în care se controlează simultan viteza de deplasare pentru trei motoare de curent continuu. Cu alte cuvinte, s-a realizat o structură originală pentru trei sisteme independente de reglare numerică automată a vitezei, comandate cu un singur microprocesor uzual.

Structura hardware pentru fiecare dintre cele trei sisteme este realizată în varianta optimală propusă în capitolul 5. Componentele de sistem necesare sînt utilizate de cîte trei ori în acest caz, cu excepția microsistemului care este unic în configurația propusă.

Conducerea, în paralel, a trei sisteme de acționare presupune efectuarea tuturor celor șase funcții de comandă și reglare de cîte trei ori pe parcursul unei perioade de eșantionare. Durata acesteia, însă, este limitată superior de mărimea constantelor de timp care intervin în fiecare proces. Dificultăți în implementare sînt cauzate și de faptul că se dispune de un singur generator de timp real în sistem. Efectuarea unui număr mare de calcule și comenzi în condiții de constrîngere temporală a fost posibilă prin aplicarea unui procedeu original de execuție "întreșesută" a operațiilor aferente celor trei sisteme. Procedeeul propus constă în:

- a) nominalizarea evenimentelor din cele trei procese cu cîte un cuvînt de cod specific;
- b) construirea în memoria microsistemului a unui "tabel de evenimente" pe durata fiecărei perioade de eșantionare;
- c) contorizarea intervalelor de timp, în care sau între care se petrec evenimente, cu timerul unic disponibil.

În "tabelul de evenimente" sînt ordonate valorile numerice corespunzătoare intervalelor de timp în/între care se desfășoară evenimentele, în succesiunea corespunzătoare din perioada de eșantionare următoare. Pe parcursul acesteia din urmă

valorile ordonate sînt transferate timerului pentru conversia succesivă în durată. Descifrînd conținutul codului specific (atașat în tabel fiecărei valori numerice cu corespondent timp) microprocesorul decide ce operație și pentru care dintre sisteme trebuie să servească în execuție.

În afara procedurii de mai sus, pentru două dintre funcțiile necesare în conducere a fost concepută o modalitate de execuție în paralel:

- a) preluarea celor trei mărimi prescrise de la nivelul ierarhic superior se execută prin "citirea" unui singur cuvînt de cod dedicat;
- b) măsurarea turației celor trei motoare se realizează cu un procedeu original de numărare simultană a impulsurilor generate de traductoarele aferente.

Regulatele implementate folosesc algoritmul PI proiectat în capitolul 5. Coeficienții numerici din formula finală obținută pentru acest algoritm sînt adaptabili la tipurile de motoare concret utilizate. Calculele aferente reglării se efectuează conform algoritmului, parcurs de cîte trei ori într-o perioadă de eșantionare în cadrul unei subrutine dedicate. Cele trei mărimi de comandă rezultate din calcule sînt depuse în "tabelul de evenimente" într-o ordine valoric crescătoare. Subrutina de calcul construiește și un al treilea cuvînt de cod cu care este sintetizată, ulterior, distribuirea corectă a semnalelor de comandă pentru chopere.

Procedurile software originale expuse în acest capitol au fost implementate pe microprocesorul 8085. Aceste proceduri sînt însă independente de structura particulară a microprocesorului folosit. Ele reprezintă un ansamblu de noi mijloace care permit utilizarea microprocesoarelor uzuale în aplicații de complexitate mai mare.

Structura de sistem de acționare propusă în capitolul 6, în care cu un singur microprocesor se comandă în viteză trei axe simultan este o contribuție adusă de teza de doctorat la optimizarea conducerii și acționării roboților industriali.

Elaborarea unor sisteme de reglare cu performanțe și mai bune, în condițiile utilizării microprocesoarelor uzuale la comanda lor, presupune implementarea unor algoritme care să asigure o reglare la indicatori de calitate superiori. De aceea în capi-

tolul 7 al lucrării este abordat principiul reglării modal alunecătoare, tot mai frecvent citat în literatura din ultimii ani, în legătură cu sinteza algoritmilor de reglare performante.

La începutul capitolului se prezintă, folosindu-se o formulare matematică sintetică, principiul RMA. Se subliniază avantajele conferite de aplicarea acestui principiu: asigură insensibilitate evoluției stărilor procesului în raport cu parametrul său și o bună rejecție a perturbațiilor. În cazul aplicării în sisteme de poziționare, RMA permite creșterea apreciabilă a vitezei de deplasare a punctului caracteristic pe traiectoria sa.

Pe baza concluziilor desprinse din analiza resurselor bibliografice se precizează expresia generală pentru legile de comandă ce conduc evoluția sistemului la modul alunecător real și ideal. Sînt deduse etapele de proiectare pentru algoritmele de reglare modal-alunecătoare.

Se prezintă și dezavantajul principal al principiului RMA: principiul utilizează o lege de comandă cu structură variabilă, iar la implementarea numerică variațiile sînt discrete ceea ce determină oscilații ale valorilor variabilelor de stare în jurul celor stabilite de obiectivul de conducere. În cazul sistemelor de poziționare numerică rezultatul final este că elementul condus prezintă oscilații de amplitudine mică, dar cu frecvență ridicată, în jurul poziției sale țintă. Pentru eliminarea acestui dezavantaj este propusă, între algoritmele prezentate în capitolul 7, și o variantă originală care utilizează, în mod combinat, un ARMA cu un algoritm PI. Varianta propusă îmbină avantajele reglării modal-alunecătoare cu posibilitatea de a obține eroare nulă în regim staționar.

Preocuparea centrală în capitolul 7 este de a implementa algoritme modal-alunecătoare pentru reglarea poziției în sistemul de acționare cu motor de curent continuu "exclusiv numeric", în configurație minimală, propus în capitolul 5.

În acest scop, se efectuează întîi o analiză de principiu pentru sistemele de reglare cu motor de curent continuu de acționare, sînt alese variabilele de stare și variabila de comutație, se stabilește (matematic) obiectivul de conducere și se calculează forma legii de comandă. Pe baza unui studiu efectuat asupra efectelor coeficienților ce intervin în formulările matematice, sînt propuse modalități principiale de acordare a re-

gulatorilor ce utilizează ARMA.

În proiectarea concretă se pornește de la modelul matematic al procesului condus extins și se parcurg etapele de proiectare deduse în analiza cu caracter general efectuată anterior.

Au fost proiectate două variante de ARMA:

- un algoritm obținut pe un model simplificat, cu neglijarea inerției producerii acțiunii ponderomotoare - ARMA-1;
- un algoritm proiectat pornind de la modelul matematic complet al procesului condus extins - ARMA-2.

S-a procedat și la introducerea unui estimator de perturbații pentru compensarea și mai eficientă a acestora.

Complexitatea algoritmilor proiectate a pretins, înainte de etapa implementării propriu-zise, și simularea pe calculator. A fost conceput un program în limbaj FORTRAN cu care s-a studiat comportarea celor două algoritme pe modelul simplificat și pe cel complet al procesului. Programul de simulare elaborat permite și verificarea formelor finale implementate în limbaj de asamblare pentru cele două algoritme.

În lucrare se prezintă detaliat soluțiile originale de implementare concretă a ARMA pe sistemul de acționare cu configurația minimală propusă în capitolul 5. Programul de conducere a sistemului a fost conceput într-o structură modulară compusă din subrutine dedicate execuției diferitelor funcții. A fost elaborat un program principal denumit executiv în timp real care îndeplinește cea mai mare parte dintre funcțiile aferente conducerii:

- (F1) preluarea poziției prescrise de la sistemul ierarhic superior;
- (F2) preluarea poziției curente de la numărătorul hardware ce contorizează impulsurile generate de traductorul incremental;
- (F3) gestiunea timpului disponibil pe durata perioadei de eșantionare, ceea ce presupune următoarele operații:
  - comanda timerului pentru conversia cod numeric-timp a mărimii de comandă și marcarea sfârșitului duratei acesteia;
  - comanda timerului pentru generarea unor intervale fixe de timp (durata perioadei de eșantionare, jumătatea acesteia)
- (F4) comanda nemijlocită a chopperului;
- (F5) activarea subrutinei care implementează algoritmul de re-



glare propriu-zis.

La implementarea celor cinci funcții au fost refolosite și unele dintre procedeele propuse în capitolul 6 cu privire la codificarea întreruperilor ce marchează evenimentele diferite și construcția unui "tabel de evenimente". Calculele aferente algoritmului propriu-zis se efectuează cu o subrutină dedicată.

Cu această subrutină au fost implementate trei algoritme de reglare:

- cele două algoritme de reglare modal-alunecătoare proiectate (ARMA-1 și ARMA-2);
- o variantă combinată în care un algoritm modal-alunecător se schimbă cu unul PI din momentul în care abaterea scade sub o valoare impusă.

Cele trei variante de ARMA propuse au fost studiate și experimental. În acest scop s-a folosit standul de încercări special construit, utilizat și în experimentările din capitolele precedente. Pentru a se realiza un studiu experimental cât mai complet a fost conceput un program amplu (TEST) cu care au fost implementate mai multe funcții utilitare. Principalele facilități conferite de programul TEST sînt:

- posibilitatea de a selecta un anumit algoritm de reglare din mai multe variante depozitate în memoria microsistemului de comandă;
- posibilitatea de a modifica parametrii algoritmilor (în scopul acordării);
- posibilitatea de a genera mărimi prescrise cu diferite forme de variație în timp: constantă, treaptă, rampă;
- posibilitatea de simulare prin software a unor perturbații;
- posibilitatea de a porni după dorință sistemul de acționare;
- posibilitatea de a opri sistemul de acționare cu o comandă voluntară sau în mod automat după diferite intervale de timp (programabile) de la pornire;
- posibilitatea de a afișa grafic evoluția în timp a valorilor pentru mărimile care interesează în proces.

Rezultatele experimentale obținute sînt prezentate în Anexa a lucrării.

## 8.2. Contribuții

Obiectivul tezei de doctorat a fost acela de a aduce contribuții la optimizarea sistemelor de conducere nemijlocită a axelor roboților industriali cu acționare electrică de curent continuu. Cercetările efectuate sînt în legătură cu sistemele de reglare automată de la nivelul cuplelor cinematice conducătoare ale dispozitivului de ghidare, care constituie sistemul de acționare al unui robot. S-a urmărit realizarea unor sisteme de acționare în variante exclusiv numerice, cu configurații hardware optime avînd la bază microprocesoare uzuale.

În concordanță cu obiectivele urmărite, principalele realizări cuprinse în lucrare sînt:

1. Realizarea unor sisteme pentru conducerea nemijlocită a unei axe ca sisteme de reglare automată "exclusiv numerice", cu o configurație hardware optimal simplificată. Soluțiile de implementare au permis eliminarea din structura sistemelor realizate a circuitelor analogice, a convertoarelor analog-numerice și numeric-analogice, respectiv a tahogeneratoarelor. Cu aceste sisteme s-au obținut performanțe foarte bune în reglare la prețuri de cost minime.

2. Elaborarea de noi modalități de utilizare a microprocesoarelor în comanda sistemelor de acționare cu motoare de curent continuu.

3. Sintetizarea unor procedee originale care permit exploatarea cu maxim de eficiență a microsistemelor uzuale.

4. Realizarea cu numai trei circuite integrate a unui sistem de comandă complet pentru acționarea unei axe. Sistemul asigură reglarea poziției și/sau vitezei elementului condus și limitarea curentului din înlășurarea rotorică a motorului de curent continuu de acționare.

5. Implementarea pe aceeași configurație minimală a circuitului de comandă pentru trei sisteme de acționare de curent continuu independente.

6. Implementarea numerică de algoritme clasice și moderne pe structurile de sistem propuse.

7. Realizarea practică și verificarea prin simulare și pe model experimental a șase variante originale de sistem de acțio-

nare cu motor de curent continuu a unei axe.

Contribuțiile originale ale autorului prezentate în lucrare sînt:

2.1. Sistemizarea principalelor aspecte tratate în literatura de specialitate cu privire la problema conducerii roboților industriali.

2.2. O împărțire sintetică în trei grupe de sarcini a ansamblului de funcții și operații ce revin spre execuție unui sistem de conducere pentru roboți industriali:

- a. modelarea mediului exterior;
- b. specificarea, generarea și controlul mișcărilor;
- c. dialogul cu operatorul uman.

2.3. Tratarea problemei conducerii roboților industriali în două etape:

- a. specificarea și generarea parametrilor cinematici ai mișcării;
- b. conducerea nemijlocită a cuplelor cinematice conducătoare ale dispozitivului de ghidare.

2.4. Analiza stadiului actual și a principalelor direcții de dezvoltare a sistemelor de acționare electrică a roboților industriali pe baza unei sinteze bibliografice în domeniu.

2.5. Propunerea unei structuri minimale pînă la care se pot optimiza sistemele de acționare.

3.1. Construirea a trei sisteme de comandă cu microprocesoare din familii diferite cu scopul de a fi utilizate în conducerea sistemelor de acționare.

3.2. Construirea unui stand pentru experimentarea sistemelor de acționare cu motor de curent continuu, folosind la comandă microprocesorul 8085.

4.1. Implementarea pe trei variante de microsisteme a circuitelor de comandă pentru redresoare și choppers.

4.2. Elaborarea programelor în limbaj de asamblare afezente circuitelor de comandă realizate.

4.3. Implementarea unor aplicații cu funcțiuni multiple pentru microprocesoare în comanda redresoarelor și chopperelor.

5.1. Realizarea și experimentarea unui sistem de reglare numerică a vitezei într-o acționare cu motor de curent continuu comandată cu microprocesor.

5.2. Realizarea și experimentarea unui sistem de reglare numerică a vitezei și poziției într-o acționare cu motor de cu-

rent continuu comandată cu microprocesor.

5.3. Realizarea unei structuri "exclusiv numerice" cu configurație minimală pentru sistemele de reglare realizate.

5.4. Sinteza tuturor funcțiilor de comandă a sistemelor de acționare propuse cu numai trei circuite integrate.

5.5. Eliminarea elementelor analogice și a convertoarelor analog-numerice și numeric-analogice din structurile de sistem propuse.

5.6. Elaborarea unui procedeu de calcul a vitezei din informația furnizată de traductorul incremental de deplasare, ceea ce a permis eliminarea tahogeneratorului din structura sistemelor realizate.

5.7. Concepția și implementarea unui procedeu de limitare a curentului din înfășurarea rotorică a motorului, folosind mijloace exclusiv **software**.

5.8. Implementarea software a funcțiilor de comandă aferente sistemelor de acționare realizate.

5.9. Elaborarea unui procedeu de gestionare judicioasă a duratei perioadei de discretizare pentru execuția tuturor funcțiilor de comandă în succesiunea impusă de proces la un moment dat.

5.10. Proiectarea și implementarea unor algoritme discrete pentru reglarea vitezei și, respectiv, reglarea simultană a vitezei și poziției.

6.1. Realizarea și experimentarea a trei sisteme de acționare cu motor de curent continuu cu reglare automată a vitezei comandate cu un singur microprocesor.

6.2. Elaborarea unui procedeu software de execuție "întrețesută" a funcțiilor de comandă aferente celor trei sisteme independente.

6.3. Elaborarea unui procedeu de numărare simultană a impulsurilor generate de trei traductoare incrementale de deplasare.

7.1. O sinteză bibliografică asupra principiului reglării modal-alunecătoare.

7.2. Propunerea unei metodologii de proiectare a algoritmilor de reglare modal-alunecătoare.

7.3. Propunerea unei tehnici de acordare a sistemelor de reglare modal-alunecătoare.

7.4. Aplicarea metodologiei de proiectare propuse în proiectarea unui sistem de reglare numerică modal-alunecătoare a poziției într-o acționare cu motor de curent continuu.

7.5. Realizarea și experimentarea unui sistem de reglare numerică automată modal-alunecătoare a unei acționări de curent continuu, comandate cu microprocesoare uzuale.

7.6. Proiectarea și implementarea a două algoritme de reglare modal-alunecătoare.

7.7. Elaborarea unui program de simulare în limbaj FORTRAN pentru sistemele de reglare realizate.

7.8. Elaborarea programului de conducere a sistemului de reglare modal-alunecătoare într-o structură modulară compusă din subrutine dedicate execuției diferitelor funcții.

7.9. Elaborarea unui program denumit "executiv de timp real" pentru sinteza principalelor funcții de comandă a sistemului, independent de tipul algoritmului de reglare utilizat.

7.10. Implementarea unei variante de sistem de reglare a poziției care utilizează un mod combinat un algoritm de reglare modal-alunecător cu un algoritm PI clasic.

7.11. Elaborarea unui program amplu în limbaj de asamblare - denumit TEST - pentru studiul experimental al sistemelor de acționare cu motor de curent continuu comandate numeric.

BIBLIOGRAFIE

1. Andreiciuc, D., Bogdanov, I., Popescu, V., Invertor trifazat cu tensiune de ieşire sintetică comandat cu microprocesorul 8085, Simpozionul "Aplicații ale electronicii industriale" Craiova, 17-18 nov., 1983, pp.27-33.
2. Athani, V.V., Deshpande, S.M., Microprocessor control of a three phase inverter in induction motor speed control system, IEEE Transactions on Industrial Electronics and Control Instrumentations, Vol.IECI-27, No.4, nov.1980, pp.291-298.
3. Babuția I., Dragomir, T.L., Mureșan, I., Proștean, O., Conducerea automată a proceselor, Editura Facla, Timișoara, 1985.
4. Bodea, M., Vătășescu, A., Tănase, G., Neagu, S., Năstase, A., Gheorghiu, V., Marinescu, N., Circuite integrate lineare. Manual de utilizate, vol.IV, Editura Tehnică, București, 1985.
5. Bogdanov, I., Conducerea cu microcalculatorul a sistemelor de acționare electrică, Referatul nr.1 în cadrul pregătirii la doctorat, IPTV Timișoara, 14 iul.1989.
6. Bogdanov, I., Microprocesorul în comanda acționărilor electrice, Editura Facla, Timișoara, 1989.
7. Bogdanov, I., Structuri de sisteme de acționare electrică cu comandă numerică implementate în sisteme de conducere a roboților industriali, Referatul nr.2 în cadrul pregătirii la doctorat, U.T.Timișoara, 18 martie 1991.
8. Bogdanov, I., Bătrînu, M., Acționare cu motor pas cu pas comandată prin microcalculatorul într-un singur circuit integrat 8035, CNEE'84, Craiova, 20-21 sept.1984, vol.7, pp.25-30.
9. Bogdanov, I., Neag, I., Single microprocessor system of speed digital control for three d.c. motors, Proceedings of the Sixth National Conference on Electrical Drives, Timișoara, May 13-14, 1988, pp.2.7-2.12.
10. Bogdanov, I., Neag, I., Literati, J., Precup, R., Implementarea software a unor algoritme numerice de reglare modală alunecătoare a poziției, Al IX-lea Simpozion Național de Roboți industriali, Baia-Mare, 26-28 oct.1989.

11. Bogdanov, I., Popescu, V., Andreiciuc, D., Comanda prin micro-procesor a unui chopper de putere utilizat în acționarea cu motoare de curent continuu, Simpozionul "Aplicații ale electronicii industriale", Craiova, 17-18 nov.1983, pp.6-14.
12. Bogdanov, I., Precup, A., Asupra implementării software a algoritmului de conducere a unui sistem de reglare a vitezei unui motor de curent continuu, realizat cu microprocesorul 8085, Al IV-lea Simpozion Național de Teoria sistemelor, Craiova, dec.1986, pp.391-397.
13. Bogdanov, I., Precup, A., Three microprocessor families in several electrical drives control. The 5<sup>th</sup> NCED, Iași, May 16-17 1986, pp.C7-C15.
14. Brady, M., Basics of robot motion planning and control. Robot motion, M.Brady, J.Hollenbach, T.Johnson, T.Lozano-Perez and M.Mason, Cambridge Mass M.J.T.Press, 1983, pp.1-50.
15. Buja, G., Fiorini, P., Microcomputer control of PWM Inverters. IEEE Transactions on Industrial Electronics, vol.IE-29, No.3, Aug.1982, pp.212-216.
16. Călin, S., Dumitrache, I., Reglatoare automate, Editura Didactică și pedagogică, București, 1985.
17. Căpățînă, O., Cornea-Hășegan, M., Pușca, M., Proiectarea cu microprocesoare, Editura Dacia, Cluj-Napoca, 1983.
18. Cernat, I.M., Sistem de reglare numerică a vitezei pentru trei motoare de curent continuu cu un singur microprocesor de comandă, Proiect de diplomă, IPTV Timișoara, 1988.
19. Chen Wie Ying, J., Eng Kee, G., Politehnica Ngee An Singapore, Controlul și comanda motoarelor electrice cu ajutorul microprocesoarelor de cost redus, Mașini și acționări electrice, tendințe actuale, Editura Tehnică, București, 1988.
20. Coita, V., Aparat pentru testarea circuitelor integrate tip  $\beta$  AAl45. Certificat de inovator nr.937, 04.06.1987, MIET-IAEM Timișoara.
21. Coita, V., Comandă numerică pentru un inverter trifazat în punte. Certif.de inovator nr.938, 04.06.1987, MIET-IAEM Timișoara.

22. Cojocaru, G., Kovacs, F., *Roboții în acțiune*, Editura Facla, Timișoara, 1985.
23. Craig, J.J., *Introduction to Robotics, Mechanics & Control*, Addison Wesley Publishing Company, Reading, Massachusetts, Menlo Park, California, Don Mills, Ontario, Wokingham, England, Amsterdam, Sydney, Singapore, Tokyo, Mexico City, Bogota, Santiago, San Juan, 1986.
24. Dancea, I., *Microprocesoare. Arhitectură internă, aplicații*, Editura Dacia, Cluj, 1979.
25. Davidoviciu, A., Drăgănoiu, Gh., Moangă, A., *Modelarea, simularea și comanda manipuletoarelor și roboților industriali*, Editura Tehnică, București, 1986.
26. Denavit, J., Hartenberg, R.S., *A Kinematic Notation for Lower - Pair Mechanism Based on Matrices*, J. Applied Mechanics, June 1955, pp.215-221.
27. Dewan, S.B., Dunford, W.G., *A microprocessor-based controller for a three-phase controlled rectifier bridge*. IEEE Transactions on Industry Applications, vol. IA-19, No. 4, Jan/Feb. 1983, pp.113-119.
28. Dewan, S.B., Mirbod, A., *Microprocessor-based optimum control for four quadrant chopper*. IEEE Transactions on Industry Applications, vol. IA-17, No. 1, 1981, pp.34-40.
29. Dewan, S.B., Stroughen, A., *Power semiconductor circuits*. Wiley Interscience Publication, New York, Toronto, Brisbane, 1975.
30. Dordea, T., *Mașini electrice. Editura Didactică și pedagogică, București, 1977.*
31. Dorn, W.S., McCracken, D.D., *Metode numerice cu programe în Fortran IV*, Editura Tehnică, București, 1976 (traducere din limba engleză).
32. Dragomir, T.L., *Regulate automate, curs*, Litografia IPTV, Timișoara, 1986.
33. Dragomir, T.L., Bogdanov, I., Literati, J., *Asupra unor variante de algoritme numerice de reglare modală alunecătoare a mișcării unei axe. Simpozionul Național de Calculatoare și Conducerea Automată a Proceselor*, Timișoara, 8-10 dec. 1988.



34. Dragomir, T.L., Bogdanov, I., Neag, I., Precup, R., Minoiu, D.,  
Asupra unor algoritme numerice de reglare modală a poziției pentru acționarea cu motor de c.c. a unei axe a unui robot industrial, Al IV-lea Simpozion Național de Roboți industriali, Baia-Mare, 26-28 oct.1989.
35. Dragomir, T.L., Bogdanov, I., Precup, A., Digital control system of speed and position in a DC motor electric drive, The Sixth National Conference on Electrical Drives, Timișoara, May 13-14, 1988, Proceedings, pp.2.47-2.52.
36. Dragomir, T.L., Bogdanov, I., Precup, A., Mihăescu, M., Szabo, M., Sistem de reglare a vitezei unei axe a unui robot, realizat cu microprocesorul 8085, Al IV-lea Simpozion Național de Teoria Sistemelor, Craiova, dec.1986, pp.385-391.
37. Fransua, A., Măgureanu, R., Mașini și acționări electrice. Elemente de execuție, Editura Tehnică, București, 1986.
38. Fransua, A., Măgureanu, R., Tendințe actuale în domeniul construcției și comenzii mașinilor și acționărilor electrice, Mașini și acționări electrice, Tendințe actuale, Editura Tehnică, București, 1988.
39. Gorla, B., Renaud, M., Robots. Manipulateurs - Cepandues Editions Toulouse, 1984.
40. Harashima, F., Ueshiba, T., Hashimoto, H., Sliding Mode Control for Robotic Manipulator, in EPE, Tokyo, 1986.
41. Houppis, C.H., Lamont, G.B., Digital control systems, theory hardware, software. McGraw-Hill, International Student Edition, 1985.
42. Ionescu, T., Sisteme și echipamente pentru conducerea proceselor. Editura Didactică și pedagogică, București, 1982.
43. Iosif, N., Luca, D.M., Udrea-Spenea, M., Popa, E., Boulescu, G., Primejdie, G., Tiristoare și module de putere. Catalog, Editura Tehnică, București, 1984.
44. Kaimoto, M., Hashii, M., Yanase, T., Nakano, T., Performance improvement of current source inverter-fed induction motor drives. IEEE Transaction on Industry Applications, vol. IA-18, No.6, nov./dec.1982, pp.703-710.

45. Kelemen,A., Acționări electrice, Editura Didactică și pedagogică, București, 1979.
46. Kelemen,A., Imecs,M., Electronică de putere, Editura Didactică și pedagogică, București, 1980.
47. Kelemen,A., Imecs,M., Incremental position control system with asynchronous motor. The Fifth National Conference on Electrical Drives, Iași, May 16-17, 1986, pp.B.17-B.19.
48. Kelemen,A., Imecs,M., Mutatoare, Editura Didactică și pedagogică, București, 1978.
49. Kelemen,A., Imecs,M., Sisteme de reglare cu orientare după câmp ale mașinilor de curent alternativ, Institutul Politehnic Cluj, 1987.
50. Kelemen,A., Imecs,M., Sisteme de reglare cu orientare după câmp ale mașinilor de curent alternativ, Editura Academiei, București, 1989.
51. Kelemen,A., Imecs,M., Matlac,I., Titz,G., Mutatoare. Aplicații, Editura Didactică și pedagogică, București, 1980.
52. Kovacs,Fr., Roboți industriali, Note de curs, U.T.Timișoara, 1991.
53. Kovacs,Fr., Unele probleme ale strategiei de implementare a sistemelor de fabricație flexibilă robotizate, Robot MEM'88, Al VIII-lea Simpozion Național de Roboți industriali, Cluj, 20-22 oct.1988, vol.I, pp.407-413.
54. Kovacs,Fr., Cojocaru,G., Manipulatoare, Roboți și aplicațiile lor industriale, Editura Facla, Timișoara, 1982.
55. Kovacs,Fr., Cojocaru,G., Pentru o știință interdisciplinară a producției, Lucrările celui de al IV-lea Simpozion "Robotizare în industrie, Mecanisme și transmisii mecanice", Timișoara, 29 nov.-2dec.1984, vol.III, pp.125-127.
56. Kuo,B.C., Kelemen,A., Crivii,M., Trifa,V., Sisteme de comandă și reglare incrementală a poziției. Editura Tehnică, București, 1981.
57. Kummel,F., Elektrische Antriebstechnik, Springes Verlag, Berlin, 1971.

58. Lane, J.D., Robot Welding, International Trends in Manufacturing Technology, IFS (Publications) LTD, UK, Springer-Verlag Berlin, Heidelberg, New-York, London, Paris, Tokyo, pp.281.
59. Lee, G.C.S., Robot and Kinematics, Dynamics and Control, University of Michigan, IEEE Transactions on Robotics, dec.1982.
60. Lungu, M.ș.a., Echipamente modulare cu semiconductoare de putere, vol.I, MICMUE-CIETA, București, 1970.
61. Lupu, C., Stăncescu, S., Microprocesoare. Circuite-Proiectare, Editura Militară, București, 1986.
62. Lupu, C., Tepelea, V., Purice, E., Microprocesoare. Aplicații, Editura Militară, București, 1982.
63. Mihăescu, M., Regulator numeric de turație cu microprocesor. Proiect de diplomă, IPTV, Timișoara, 1984.
64. Mirbod, A., El-Amawy, A., A General-Purpose Microprocessor Based Control Circuit for a Three Phase Controlled Rectifier Bridge, IEEE Transactions on Industrial Electronics, Aug.1986, Vol.IE-33, No.3, pp.310-317.
65. Mirbod, A., El-Amawy, A., Performance Analysis of a Novel Microprocessor Based Controller for a Phase-Controlled Rectifier Connected to a Weak AC System, IEEE Transaction on Industry Applications, 1987, No.1, pp.57-65.
66. Mureșan, T., Conducerea roboților industriali, note de curs, IPTV Timișoara, 1988.
67. Mureșan, T., Modulator în durată numeric pentru acționări de curent continuu condus de microcalculatoare, Al IX-lea Simpozion Național de Roboți industriali, Baia-Mare, 26-28 oct.1989.
68. Neag, I.A., Sistem de reglare numerică a poziției într-o acționare cu motor de curent continuu folosind algoritme de reglare modală alunecătoare, Proiect de diplomă, IPTV Timișoara, 1990.
69. Nevins, J.L., Witney, D.E., ș.a., Exploratory Research on Industrial Assembly Matting, Ch.Stark Draper Laboratory, Cambridge, Mass, 1980.

70. Olivier, G., Stefanovici, V., April, G.E., Microprocessor controller for a thyristor converter with an improved power factor. IEEE Transactions on Industrial Electronics and Control Instrumentation, vol.I, IECI-28, No.3, aug.1981, pp.188-194.
71. Osborne, A., An introduction to microcomputers, vol.0-1, Basic Concepts, Berkeley, California, 1979.
72. Osborne, A., An introduction to microcomputers, vol.2, Some real microprocessors, Berkeley, California, 1979.
73. Osborne, A., An introduction to microcomputers, vol.3, Support Devices, Berkeley, California, 1979.
74. Paul, R.P., Robot Manipulators, Mathematics, Programming and Control, The Computer Control of Robot Manipulators, The MIT Press, Cambridge, Massachusetts and London England, 1982.
75. Peñalver, C.M., Peire, J., Martinez, P.M., Microprocessor control of DC/AC static converters. IEEE Transactions on Industrial Electronics, vol.IE-32, No.3, Aug.1985, pp.186-191.
76. Plant, J.B., Jorna, S.J., Chan, J.T., Microprocessor control of position or speed of an SCR DC motor drive. IEEE Transactions on Industrial Electronics and Control Instrumentation, vol.IECI-27, No.3, aug.1980, pp.228-234.
77. Pomner, I., Electronică industrială. Editura Didactică și pedagogică, București, 1972.
78. Popescu, V., Electronică industrială. Curs, vol.1 și 2, Litografia IPTV Timișoara, 1983-1984.
79. Popescu, V., Andreiciuc, D., Bogdanov, I., Electronică industrială, Indrumător de lucrări de laborator, Litografia IPTV Timișoara, 1985.
80. Popescu, V., Andreiciuc, D., Bogdanov, I., Redresor cu tiristoare comandat prin microprocesor, Simpozionul "Aplicații ale electronicii industriale", Craiova, 17-18 nov.1983, pp. 24-27.
81. Popescu, V., Balaci, I., Electronică industrială, Indrumător de lucrări de laborator, Litografia IPTV Timișoara, 1980.

82. Popescu, V., Bogdanov, I., Andreiciuc, D., D.C. motor speed adjusting with microprocessor, Buletinul Stiintific și Tehnic al IPTV Timișoara, Tom 29, Electrotehnica, 1984, pp.73-76.
83. Precup, A., Sisteme de reglare a vitezei și poziției într-o acționare cu motor de curent continuu comandat cu microprocesor, Proiect de diplomă, IPTV Timișoara, 1986.
84. Precup, R.E., Variantă de sistem de reglare automată modală alunecătoare a poziției pe o axă la roboți industriali, Proiect de diplomă, IPTV Timișoara, 1987.
85. Sen, P.Ch., Thyristor DC Drives, Library of Congress Cataloguing in Publication Date, Wiley, Interscience Publication, USA, Canada, 1981.
86. Seracin, E., Acționări electrice, Curs, partea I-II, Litografia IPTV Timișoara, 1980.
87. Seracin, E., Popovici, D., Tehnica acționărilor electrice, Editura tehnică, București, 1985.
88. Simon, W., Conducerea numerică a mașinilor unelte (Traducere din germană), Editura Tehnică, București, 1967.
89. Snyder, W.E., Industrial Robots, Computer Interfacing and Control, North Carolina State University, Prentice/Hall International, Inc., 1985.
90. Szabo, M., Sistem de reglare automată a vitezei și poziției pentru mișcarea pe o axă a unui robot industrial realizat cu microprocesorul 8085, Proiect de diplomă, IPTV Timișoara, 1985.
91. Sabac, I.Gh., Cocârlan, P., Stănășilă, O., Topală, A., Matematici speciale, Editura didactică și pedagogică, București, 1983.
92. Sabanovici, A., Izosianov, D.B., Application of Sliding Modes to Induction Motor Control, IEEE Transactions, vol.IA-17, No.1, Jan.-Febr.1981.
93. Tae Kim, G., Wan Chun, T., Ki Sul, S., Ho Park, M., - Coreea de Sud - Design of Single-Chip Microprocessor-Based Controller for Current Source Inverter - Fed Induction Motor Drive, IEEE Transaction on Industrial Electronics, vol.IE-34, No.3, Aug.1987, pp.331-338.

94. Toacșe, Gh., Introducere în microprocesoare. Editura științifică și enciclopedică, București, 1985.
95. Trif, N.I., Contribuții la acționările electrice de curent continuu comandate cu micro sisteme - cu aplicații la roboți. Teză de doctorat, 1987.
96. Weihrich, G., Drehzalregelung von Gleichstromantrieben unter Verwendung eines Zustands und Strörgrößenbeobachter, Teil I, Regelungstechnik, 26 (1978), H11, pp.349-354.
97. Yamamura, S., AC Motors for High Performance Applications, Marcel Dekker, 1986.
98. Yamamura, S., Teoria modernă a motoarelor de curent alternativ, metoda separării fazelor, metoda vectorului rotitor, Mașini și acționări electrice, tendințe actuale, Editura tehnică, București, 1988.
99. Zaks, R., Introduction to microprocessors. Sybex Inc., Berkeley, California, 1978.
100. Zaks, R., Programming microprocessors. Sybex Inc., Berkeley, California, 1978.
101. \* \* \* Creonics VME Bus Motion Control Card - fișă de catalog, 1987.
102. \* \* \* 4<sup>th</sup> European Conference on Electrotechnics Eurocon'80, Stuttgart, Germany, 24-28 march, 1980.
103. \* \* \* Proceedings, International Conference on Electrical Machines, Budapest, Hungary, 5-9 sept.1982.
104. \* \* \* Proceedings. The Fifth National Conference on Electrical Drives, Iași, 16-17 may, 1986.
105. \* \* \* Proceedings. The Sixth National Conference on Electrical Drives, Timișoara, may 13-14, 1988.
106. \* \* \* SDK-85, System design kit. User's manual, Intel. Corporation, 1978.
107. \* \* \* 3<sup>rd</sup> 1 FAC Symposium Lausanne, Switzerland, 12-14 sept.1983.