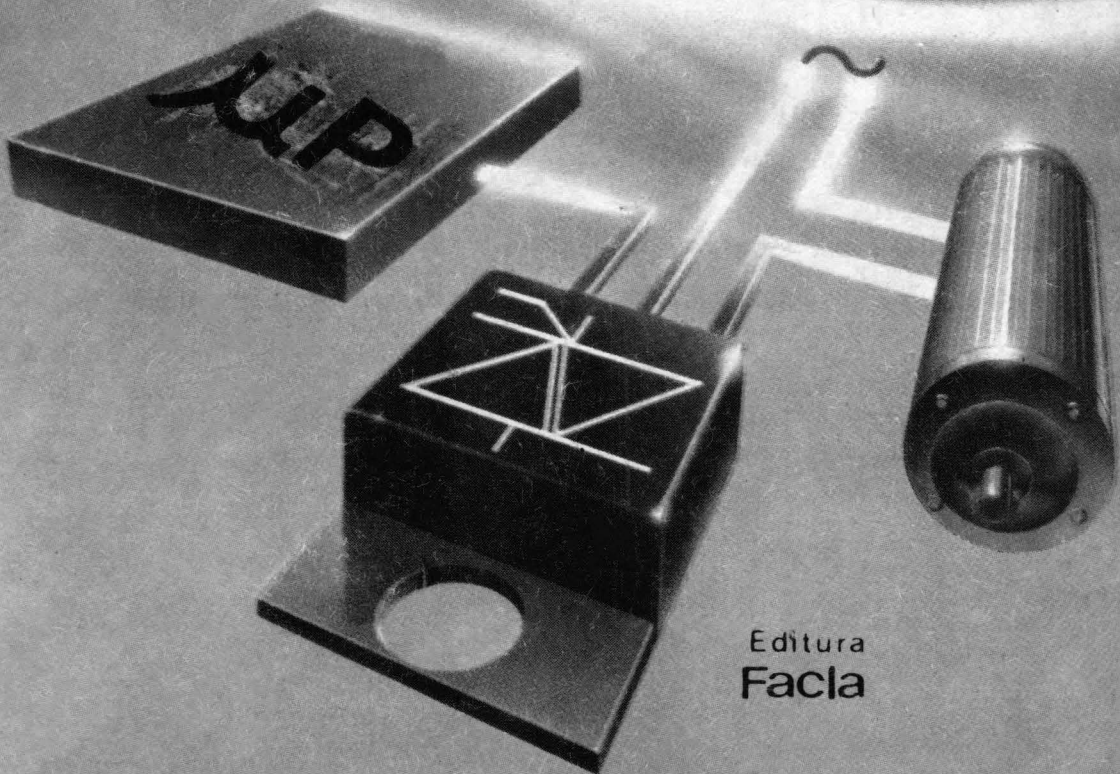


IVAN BOGDANOV

**MICROPROCESORUL
ÎN COMANDA
ACȚIONĂRILOR
ELECTRICE**



Editura
Facla

IVAN BOGDANOV

MICROPROCESORUL
ÎN COMANDA
ACȚIONĂRIILOR ELECTRICE

MICROPROCESORUL
CUPRÎN COMANDA
A ACȚIONĂRIILOR ELECTRICE

ISBN 973-36-0330-7

Redactor: ION MIH
Tehnoredactor: IOAN I. IANCU

Bun de tipar: 25.10.1988
Agost 1988
Cant. de tipar: 1125

Tiparul executat la
INTREPRINDEREA POLIGRAFICA CUB
Municipal Cluj-Napoca
Bv. Lenin nr. 146
sub contract nr. 112



ITAN BOBANEV
MIRROPROCESSORUL
IN COMANDA
ACIUNARILOR ELECTRICE

ISBN 973-36-0030-X

Redactor: ION ILIN
Tehnoredactor: IOAN I. IANCU

Bun de tipar: 27.10.1989.
Apărut 1989.
Coli de tipar: 11,25

Tiparul executat la
ÎNTRERINDERA POLIGRAFICĂ CLUJ
Municipiul Cluj-Napoca,
Bv. Lenin nr. 146
sub comanda nr. 112



IVAN BOGDANOV

INTRODUCERE

MICROPROCESORUL ÎN COMANDA ACȚIONĂRILOR ELECTRICE

EDITURA FACLA

Timișoara, 1989

IVAN BOGDANOV

MICROPROCEZORUL
IN COMANDA
ACȚIONĂRIILOR ELECTRICE

TEHNOLOGIA
DE
PROIECTARE
ȘI
CONSTRUCȚIE

DE
PROIECTARE
ȘI
CONSTRUCȚIE

TEHNOLOGIA
DE
PROIECTARE
ȘI
CONSTRUCȚIE

TEHNOLOGIA

Referent științific:

Prof. dr. ing. TIBERIU MUREȘAN

Institutul Politehnic „Traian Vuia” Timișoara

CAPITOLUL I

INTRODUCERE

În ultimii zece ani, apariția și răspândirea 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 [47, 79], iar domeniile de aplicație pentru micro sisteme se consideră limitate numai de imaginația utilizatorilor [78].

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 releelor, tranzistoarelor, a circuitelor electronice clasice [79]. Dacă numărul unităților de proces este mare, sute sau mii, apare necesitatea de a folosi minicalculatoare sau calculatoare de proces, deși și acestea tind să fie 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 drept cea mai îndreptățită, întrucât circuitele de acționare implică o complexitate medie în ceea ce privește numărul de comenzi și operații necesare.

Desigur astăzi, comanda circuitelor de acționare electrică cu circuite electronice tradiționale, analogice sau numerice, este, practic, generalizată. Se pune problema folosirii microprocesoarelor în circuite de comandă pentru acționări electrice, a schimbării conceptelor de proiectare, de sinteză a secvențelor de comandă în vederea realizării de avantaje incontestabile.

Utilizarea microprocesoarelor în domeniul amintit se justifică nu numai din punctul de vedere al complexității operațiilor și comenzilor necesare în circuitele electrice de acționare, ci și dintr-o serie de alți factori, cum sînt:

1. Micșorarea timpului de răspuns a circuitelor de comandă, întrucât viteza de lucru a microprocesorului permite acest fapt, și realizarea de circuite de acționare rapide.

2. Creșterea preciziei și siguranței în lucru, prin însăși folosirea unui sistem de calcul ca circuit de comandă. Se elimină integral (sau aproape integral) elementele analogice din circuitele de comandă.

3. Flexibilitatea — posibilitatea extinderii numărului de elemente controlate, a efectuării de modificări sau 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 integrate *LSI* a scăzut simțitor.

5. Posibilitatea subordonării directe a circuitelor de comandă pentru acționare sistemelor de calcul ierarhic superioare, la mașini cu comandă numerică, roboți etc., evitându-se convertoarele numeric-analogice sau alte circuite de interfață.

6. Îmbunătățirea fiabilității și asigurarea unei mai bune insensibilități la perturbații pentru circuitele de comandă. Se pot include și programe de test sau autodepanare.

7. Scăderea consumului de energie [78, 79, 15, 47, 19, 69, 46, 57, 60, 30].

Având în vedere considerentele arătate, spre sfârșitul anilor '70 și începutul anilor '80, studiile, cercetările și aplicațiile cu microprocesoare ca circuite de comandă în acționările electrice au cunoscut un avânt deosebit.

Cercetările s-au orientat atît în sensul folosirii microprocesoarelor în cele mai diverse circuite de acționare, cît și în cel al utilizării de diferite familii de microprocesoare, începînd cu microprocesorul microsistem într-o capsulă și pînă la microprocesoarele de 16 biți sau bit-slice [69, 38, 20].

În pofida realizărilor multiple din 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 software, atît în domeniul acționărilor cu motoare de curent continuu, cît și, mai ales, în acționările cu motoare de curent alternativ [81, 82, 87, 83, 84, 43].

Acționările electrice au cunoscut în țara noastră o dezvoltare amplă și multilaterală. Este de la sine înțeles, deci, că și problema, de mare actualitate, de a utiliza microprocesorul în comanda acționărilor electrice, a fost abordată în majoritatea institutelor de învățămînt superior, de cercetare și proiectare, întreprinderilor de profil [38, 75, 18, 86, 80].

La Institutul Politehnic „Traian Vuia” din Timișoara, Facultatea de Electrotehnică, în cadrul catedrelor de electronică aplicată și automatizări și calculatoare există preocupări cu privire la acționările electrice comandate cu microprocesor, din anul 1982.

În lucrarea de față sînt prezentate, sintetic, rezultatele obținute, teoretice și practice, concluziile ce se impun, precum și o privire în perspectivă cu referire la posibilitățile de extindere și dezvoltare a realizărilor de pînă în prezent.

SISTEME CU MICROPROCESOARE UTILIZATE ÎN COMANDA ACȚIONĂRILOR ELECTRICE

2.1. ARHITECTURA STANDARD A UNUI SISTEM CU MICROPROCESOR UTILIZAT ÎN COMANDA CIRCUTELOR DE ACȚIONARE ELECTRICĂ

Structura unui microsistem utilizat în comanda acționărilor electrice se încadrează, desigur, în arhitectura standard acceptată pentru acest tip de sisteme. Schema bloc a unui sistem cu microprocesor este reprezentată în figura 2.1. [79].

Dispozitivul fundamental din sistem este microprocesorul, care conține, în esență, elementele unei unități centrale dintr-un (micro) calculator. Operînd cu instrucții și date, rolul microprocesorului este de a decodifica

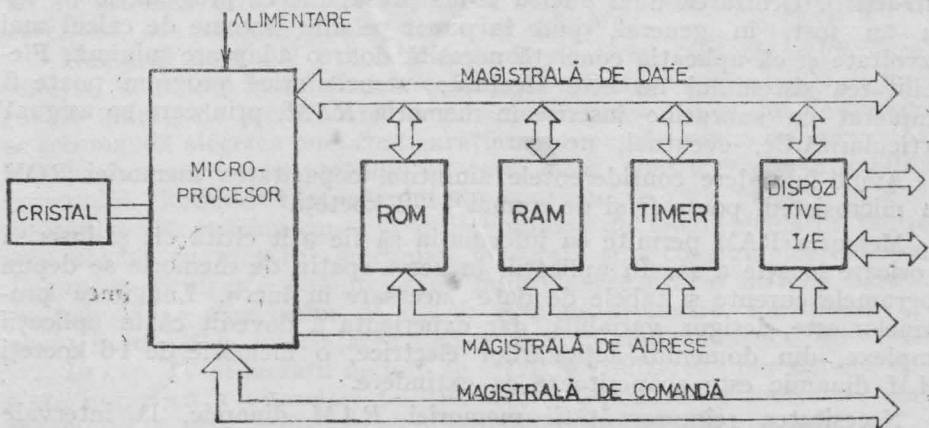


Fig. 2.1. Arhitectura standard a unui microsistem.

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 ale sistemului, în ritmul impulsurilor de tact.

Intercomunicația între microprocesor și celelalte componente 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 cuvinte 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 maximum 64 kocteți. La unele microprocesoare, magistrala de date nu este realizată fizic separat, întrucît cele 8 linii corespunzătoare biților cei 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 [78, 55, 47, 74].

Memoria ROM, nevolatilă, componentă a sistemului, constituie spațiul de depozitare al unui set minim de subrutine, necesare pentru lucrul cu sistemul, set cunoscut în literatura de specialitate [34] sub denumirea de nucleu. Avînd în vedere faptul că structura de microsistem descrisă este utilizată la comanda nemijlocită a circuitului 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: subrutina 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, întrucît orice program poate fi completat cu subrutine înscrise în memoria RAM, prin care se asigură particularizările, eventual, necesare.

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

Memoria RAM permite ca informația să fie atît citită cît și înscrisă în oricare locație a sa. În aplicații, în acest spațiu de memorie se depun programele curente și tabele 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 acoperitoare ca extindere.

Necesitatea reîmprospătării memoriei RAM dinamic, la intervale constante de timp, are drept consecință directă lungirea duratei de acces

la informația pătrată de aceasta. Pentru a se obține viteze cât mai mari în efectuarea calculelor 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 programele uzuale.

Un bloc component important în sistem îl constituie circuitele de interfață intrări/ieșiri (*I/E*) pentru comunicația cu exteriorul sistemului. Aceste circuite pot fi circuite integrate dedicate de tip 8251, 8255, 8212, 8155 ș.a. [56, 19] 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 schimbul de semnale direct implicate în comanda acționării — semnale de proces.

Semnalele direct legate de proces se grupează în mărimi de intrare, ce sosesc la porturile 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 semnale sub forma 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, funcție de aplicație, de exemplu: poziția și/sau viteza impusă pentru deplasare la un moment dat, sensul de avans, frecvența de comandă prescrisă, 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-numerice (*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 posibilă 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 singur *CAN* poate fi folosit la interfațarea mai multor mărimi analogice din proces.

În cap. IV al lucrării de față 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.

Deosebit de util pentru punerea în funcțiune și depanarea sistemului este să se includă în structura sa și o interfață I/E de comunicație cu un *DAF* [54].

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 [32]:

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, necesară 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 folosit în comanda circuitelor de acționare electrică trebuie prevăzut cu 1 ÷ 3 generatoare de timp real. Exemplul caracteristic de întrebuințare a timerelor în genul 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 cel al programului curent, executat de către unitatea centrală,
- întreruperea unității centrale cînd 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 întreruperea se pot împărți în trei categorii: semnalele 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 caz, microprocesorul își desfășoară activitatea curentă, pînă la apariția unei cereri de întrerupere, cînd 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 comanda circuitelor de acționare electrică, este suficient un număr maxim de 8 niveluri de prioritate a întreruperilor, care pot fi gestionate cu o singură capsulă de tip 8259 [32].

În cercetările de la IPTVT, 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 întrebuițate 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.

2.2. SISTEM DE COMANDĂ CU MICROPROCESORUL 8035

Microprocesorul 8035 face parte din familia 8048, familie cunoscută în literatura de specialitate sub denumirea de „low-end microprocessor”, tradus în limba română, „microsistem într-o singură capsulă” [78, 79, 19]. Schema bloc internă a microprocesorului 8048 este redată în figura 2.4 [55]. Din această structură se poate observa că microprocesorul 8048 conține, intern, toate blocurile ce intră în componența unui sistem de calcul.

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 figura 2.5. [8, 11].

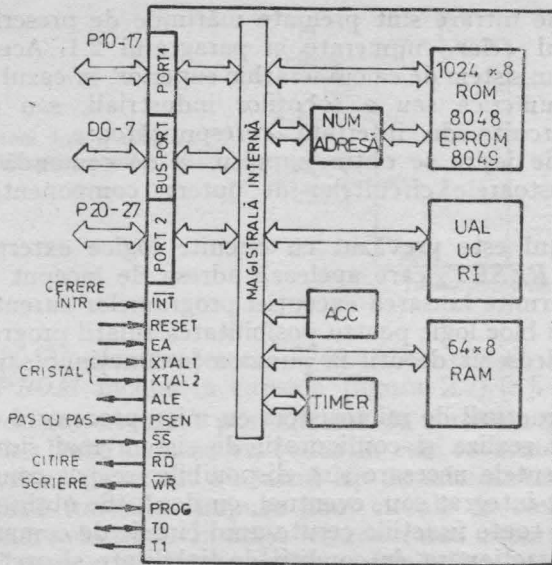


Fig. 2.4. Structura microprocesorului 8048.

Microprocesorul 8035 este prevăzut cu trei porturi bidirecționale de câte 8 biți. În cazul sistemului de față, „BUS-PORT”-ul microprocesorului este utilizat, prin multiplexare, ca magistrală de date și adresă. Astfel, pe liniile acestui port se generează cei mai puțin semnificativi 8 biți de adresă, pe aceleași linii sosesc și instrucțiunile programului din memoria EPROM externă. Liniile P20-23 din portul P2 sînt, de asemenea, folosite 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.

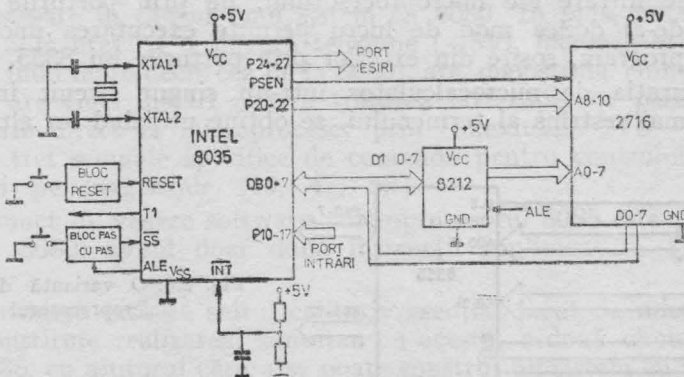


Fig. 2.5. Sistem cu microprocesorul 8035.

Pe liniile de intrare sînt preluate mărimea de prescriere și datele de proces de tipul celor enumerate în paragraful 2.1. 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 microprocesorul 8035 prezentată anterior, se pot realiza și configurații de sistem mai simple. În situația în care componentele necesare sînt disponibile, se pot construi sisteme cu un singur circuit integrat sau, 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 figura 2.6 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 prezentat pe figură.

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 [55].

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 microcalculator într-un singur circuit integrat, în sensul cel mai restrîns al termenului, se obține utilizînd un alt micropo-

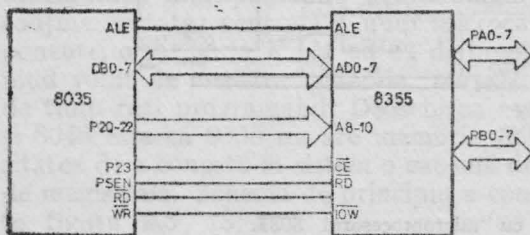
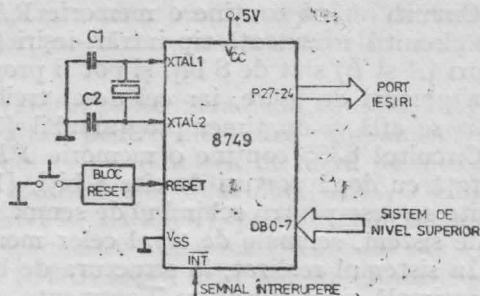


Fig. 2.6. O variantă de sistem cu microprocesorul 8035.

Fig. 2.7. Sistem într-o singură capsulă cu microprocesorul 8749.



cesor din seria 8048, microprocesorul 8749. Acest circuit are cei 2 kocteți de memorie EPROM incluși în capsulă (figura 2.7). [8].

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 multe circuite de acționare electrică. În acest fel, realizarea circuitelor de comandă pentru acționări devine extrem de simplă, ieftină, flexibilă, fiabilă și, practic, tipizată.

2.3. 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 cu 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 trătarea întreruperilor permite accesul datelor serie direct la microprocesor prin conexiunile SID—SOD și are prevăzute trei semnale specifice de comandă pentru controlul vehiculării informației pe magistrale. [19, 47, 55].

Din punct de vedere software, microprocesorul 8085 este perfect compatibil cu 8080, avînd doar două instrucții suplimentare (RIM, SIM) [47, 32].

Se consideră însă că saltul calitativ esențial făcut de microprocesorul 8085, îl constituie realizarea, simultan cu acesta, a două circuite speciale, 8155 și 8355, cu ajutorul cărora se poate construi 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/E) cu trei porturi. Două dintre porturi (A și B) sînt de 8 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 [19, 55].

Circuitul 8355 conține o memorie PROM de 2 kocteți și un circuit interfață cu două porturi de cîte 8 biți, [19, 55]. Liniile porturilor acestor circuite servesc pentru schimbul de semnale cu circuitul de acționare comandat de sistem, semnale de tipul celor menționate în paragraful 2.1.

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 efectuarea unor aplicații de complexitate medie [85].

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/E și dotarea cu un al doilea timer. S-a construit, apoi, un circuit interfață pentru display și consolă de tip DAF 1001 paralel. De asemenea, cu ajutorul unui circuit 8255 (trei porturi programabile intrare-ieșire) s-a efectuat o interfață de comunicare cu lector și perforator de bandă.

Configurația sistemului realizat și utilizat în aplicații este dată în figura 2.8 [66, 11, 9]. Din punct de vedere software, sistemul a fost dotat cu un sistem de operare minimal interactiv în memoria ROM. Nucleul sistemului de operare conține subrutina de inițializare, subrutinele de tratare I/E la nivel fizic, subrutine de gestionare a întreruperilor și un număr de subrutine utilitare (temporizări, intercomunicații etc.). Celelalte blocuri soft-

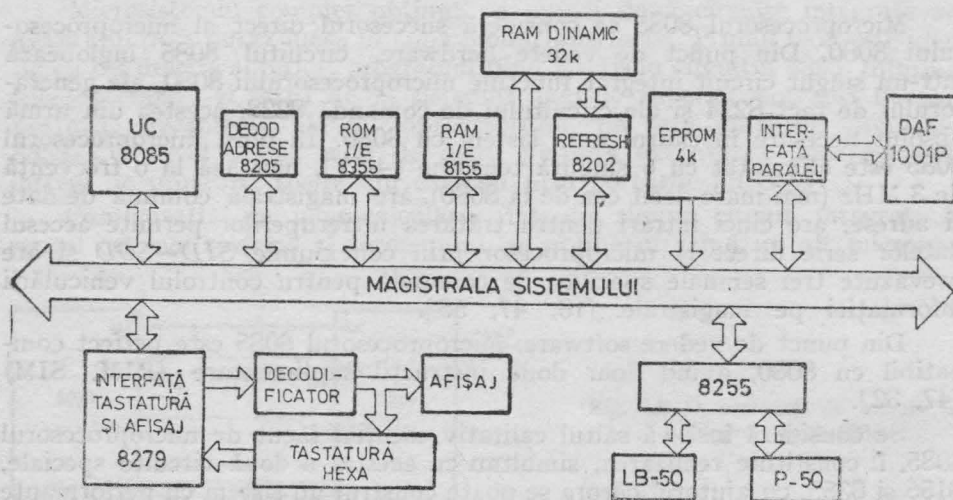


Fig. 2.8. Sistem cu microprocesorul 8085.

ware din cadrul sistemului de operare sînt: un interpretor de comenzi, un sistem de gestiune al fişierelor, un editor, un asamblor în doi paşi şi un depanator.

Printr-o astfel de dotare, operatorul poate lucra de la consola DAF-ului şi are la dispoziţie o serie de facilităţi pentru alcătuirea programelor. Dintre acestea facilităţi se pot menţiona: posibilitatea de lucru pe maximum 6 fişiere, 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 pseudoinstrucţiuni ale asamblorului (ORG, EQU, DB) ş.a. Programele pot fi citite sau salvate pe bandă perforată.

Pe parcursul lucrării, cu deosebire în cap IV, se prezintă programe întocmite pe sistem şi se evidenţiază detalii cu privire la facilităţile oferite de acesta.

Sistemul prezentat, succint, în acest paragraf, este capabil să realizeze conducerea unor procese complexe, cu un număr suficient de mare de unităţi de proces controlabile şi implicînd calcule laborioase. Configuraţia construită şi utilizată reprezintă mai mult decît o structură strict concepută pentru comanda nemijlocită a circuitelor de acţionare electrică, de tipul prezentat, principal, în paragraful 2.1. Microcalculatorul descris poate fi, însă, folosit la întocmirea, punerea la punct şi depanarea unor programe pentru alte sisteme, mai restrînse, componente propriu-zise ale circuitelor de acţionare electrică.

2.4. SISTEM DE COMANDĂ CU MICROPROCESORUL Z80

Structura sistemului utilizat este dată în figura 2.9 [11] şi reprezintă o structură standard de microcalculator de uz general. Microprocesorul Z80 este considerat primul reprezentat al familiilor de microprocesoare moderne. Deşi descendent din familia 8080, microprocesorul Z80 are resurse mult sporite, mai ales din punct de vedere software. Dintre aceste modalităţi sînt de amintit: posibilitatea adresării indexate cu ajutorul a două registre index, posibilitatea adresării relative şi pe bit, existenţa setului

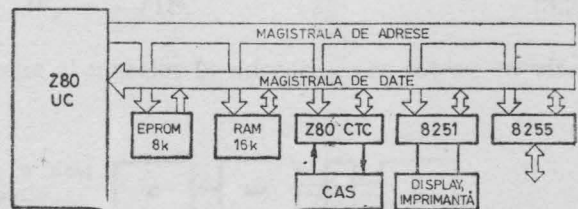


Fig. 2.9. Sistem cu microprocesorul Z80.

alternativ de registre, precum și a instrucțiilor de transfer de date în bloc, [55, 46].

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 cu circuitele de acționare propriu-zise similare cu cele amintite la paragraful 2.1 se face prin circuitul 8255.

2.1. SISTEM DE COMANDĂ CU MICROPROCESORUL 8085

Structura sistemului este prezentată în figura 2.1.1. Sistemul este o structură modulară, cu un procesor central de comandă și control, un procesor de date și un procesor de ieșire. Sistemul este controlat de un procesor de comandă și control, care este conectat la un procesor de date și un procesor de ieșire. Sistemul este controlat de un procesor de comandă și control, care este conectat la un procesor de date și un procesor de ieșire.



UTILIZAREA MICROPROCESOARELOR ÎN COMANDA ACȚIONĂRILOR CU MOTOARE DE CURENT CONTINUU

3.1. PROBLEME SPECIFICE CU PRIVIRE LA ACȚIONĂRILE CU MAȘINI DE CURENT CONTINUU

Structura unei acționări electromecanice se prezintă schematic ca în figura 3.1. [62, 28, 70], unde L este mașina de lucru (mecanismul executor), MA motorul de acționare, C convertorul de energie prin care este alimentat motorul, iar V concretizează masele în mișcare (volant).

Mașina de lucru dezvoltă în arborele motorului un cuplu static 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 [70].

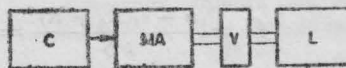
$$P_J = \frac{dW_c}{dt} \quad (3.1)$$

Dar

$$W_c = \frac{1}{2} J \Omega^2, \quad (3.2)$$

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

Fig. 3.1. Structura bloc a unei acționări electromecanice.



Deci

$$P_J = J\Omega \frac{d\Omega}{dt} + \frac{\Omega^2}{2} \frac{dJ}{d\alpha} \cdot \frac{d\alpha}{dt} = J\Omega \frac{d\Omega}{dt} + \frac{\Omega^2}{2} \frac{dJ}{d\alpha}, \quad (3.3)$$

(se admite, în general, că J variază cu poziția unghiulară α).

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

$$M_d = M - M_S = \frac{P_J}{\Omega} = J \frac{d\Omega}{dt} + \frac{\Omega^2}{2} \frac{dJ}{d\alpha}. \quad (3.4)$$

În practică, de cele mai multe ori, $J = \text{const.}$ și ecuația de mișcare devine:

$$M - M_S = J \frac{d\Omega}{dt}, \quad (3.5)$$

cu consecințele imediate:

a) $M > M_S \Rightarrow \frac{d\Omega}{dt} > 0$ — accelerarea acționării,

b) $M < M_S \Rightarrow \frac{d\Omega}{dt} < 0$ — încetinirea acționării,

c) $M = M_S \Rightarrow \Omega = \text{const.}$ — regim staționar.

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

Pentru motoare, definatorii în acest sens sînt caracteristicile mecanice $\Omega = f(M)$ [70]. În cazul motorului de curent continuu cu excitație separată, caracteristica mecanică naturală se obține din ecuația:

$$u = u_e + (R_i + R)i_i + L \frac{di_i}{dt}, \quad (3.6)$$

unde: u este tensiunea la borne, u_e — tensiunea indusă, R_i și i_i — rezistența și curentul prin indus, R — rezistența exterioară conectată în serie cu indusul, iar L — inductivitatea indusului [70].

Dar

$$u_e = k\Phi\Omega, \quad [62], \quad (3.7)$$

cu: Φ — fluxul de excitație, Ω — viteza unghiulară, k — o constantă care depinde de construcția motorului (numărul de perechi de poli, numărul de spire din indus etc.).

Rezultă, în regim stabilizat:

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

$$\Omega = \frac{u - i_i(R_i + R)}{k\Phi} = \frac{u}{k\Phi} \left(1 - \frac{R_i + R}{u} i_i \right). \quad (3.8)$$

Considerînd că Φ se menține la valoarea de la mersul în gol Φ_0 , avem :

$$\begin{aligned}\Omega &= \frac{u}{k\Phi_0} \left(1 - \frac{(R_i + R)i_i}{u} \right) = \Omega_0 \left(1 - \frac{(R + R_i)i_i}{u} \right) = \\ &= \Omega_0 - \frac{\Omega_0(R_i + R)i_i}{k\Phi_0\Omega_0} = \Omega_0 - \frac{(R_i + R)i_i}{K} = \Omega_0 - \Delta\Omega,\end{aligned}\quad (3.9)$$

cu $K = k\Phi_0$.

Ecuția (3.9), $\Omega = f(i_i)$, se reprezintă de obicei în coordonate $\Omega = f(M)$, deoarece cuplul :

$$M = \frac{P_e}{\Omega} = \frac{u_e i_i}{\Omega} = \frac{k\Phi\Omega i_i}{\Omega} = k\Phi i_i \quad (3.10)$$

este direct dependent de curent.

Graficul relației $\Omega = \Omega_0 - \Delta\Omega$ se numește caracteristica mecanică a mașinii [70] și este o dreaptă de alura din figura 3.2.

Caracteristica naturală corespunde lui $R = 0$, iar $\Delta\Omega = \frac{R_i i_i}{K}$ 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 (3.8) 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 cîmp [(a curentului de excitație) ;
4. Prin schimbarea tensiunii la bornele indusului, excitația rămînînd constantă, [62, 23].

Dintre aceste metode, cea mai răspîndită este cea prin schimbarea tensiunii la borne, pentru că prin această metodă caracteristicile artifi-

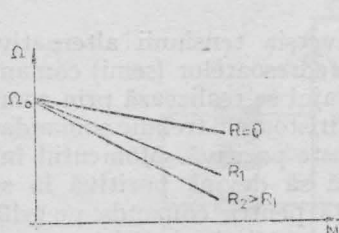


Fig. 3.2. Forma caracteristicilor mecanice ale mașinii de curent continuu.

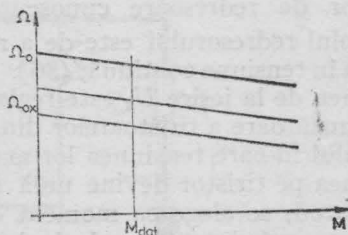


Fig. 3.3. Familia de caracteristici mecanice obținută prin modificarea tensiunii la bornele motorului de c.c.

ciale rămân paralele cu cea naturală și nu apare nici dezavantajul pierderilor de energie [36, 62].

Din relația (3.8):

$$\Omega = \frac{u - Ri_i}{h\Phi}, \quad (3.11)$$

cînd u scade, scade și termenul $\frac{u}{h\Phi}$. Se obține familia de caracteristici din figura 3.3, adică o deplasare a caracteristicii mecanice momentane.

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.

3.2. SISTEME DE ACȚIONARE CU MOTOARE DE CURENT CONTINUU ALIMENTATE DE LA REDRESOARE COMANDATE

Acționările reglabile de curent continuu s-au impus astăzi, practic, definitiv [36]. Există acum experiență vastă și un larg spațiu bibliografic în acest domeniu [22, 69, 62, 37].

În preocupările noastre 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, [36]. În cazul redresoarelor (semi) comandate, tensiunea de la ieșire U_a 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 comanda pe grilă. În raport cu referința, impulsul de aprindere al fiecărui tiristor poate fi aplicat la un interval de timp variabil. Exprimînd acest interval în radiani, se definește unghiul de comandă (α) pentru tiristoare. Valoarea medie a tensiunii redresate se modifică, dacă și unghiul de comandă se modifică.

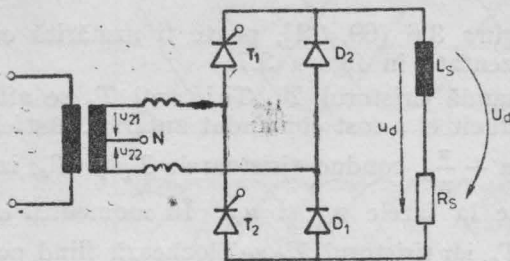


Fig. 3.4. Redresor monofazat în punte semicomandată.

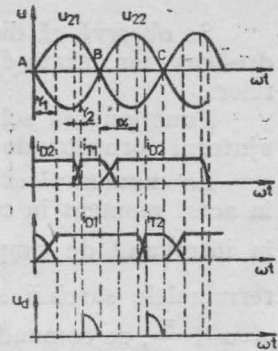


Fig. 3.5. Formele de undă reprezentative pentru redresorul monofazat în punte semicomandată.

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 [69].

Structura circuitelor utilizate este prezentată în figurile 3.4 și 3.6. Funcționarea schemei din figura 3.4 poate fi urmărită cu ajutorul formelor de undă reprezentate în figura 3.5. [62].

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 γ_2 , tensiunea u_d este practic egală cu zero. După unghiul γ_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 valoare scăzută, considerată, de obicei, nulă.

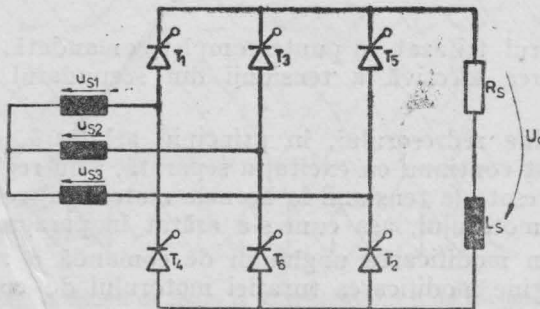


Fig. 3.6. Redresor trifazat în punte comandată.

Se observă că diodele conduc, atât ca diode de nul, cât și ca diode redresoare, iar durata de conducție a lor este mai mare decât aceea a tiristoarelor.

Funcționarea schemei din figura 3.6 [69, 22], poate fi urmărită cu ajutorul formelor de undă reprezentate în figura 3.7.

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 $\alpha < \omega t < \alpha + \frac{\pi}{3}$, conduc tiristoarele T_1 și T_6 , iar terminalele sarcinii sînt conectate la fazele u_{s1} și u_{s2} . În momentul $\omega t = \alpha + \frac{\pi}{3}$, se comandă tiristorul T_2 , iar tiristorul T_6 se blochează fiind polarizat invers (comutație naturală). Curentul din tiristorul T_6 este preluat de tiristorul T_2 și, ca urmare, tensiunea la bornele sarcinii este: $u_d = u_{s1} - u_{s3}$. Acest proces se repetă după cîte 60 grade sexagesimale, ori de cîte ori se comandă cîte un tiristor. Tiristoarele din schema prezentată în figura 3.6 au fost numerotate în ordinea în care trebuie comandate.

În figura 3.7 a fost 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 inverter”, [62, 28] 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.

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

$$U_d = \frac{\sqrt{2}}{\pi} U_s (\cos \alpha + 1), \quad (3.12)$$

pentru redresorul monofazat în punte semicomandată și

$$U_d = \frac{3\sqrt{6}}{\pi} U_s \cos \alpha, \quad (3.13)$$

pentru redresorul trifazat în punte complet comandată. În ambele relații, U_s este valoarea efectivă a tensiunii din secundarul transformatorului [69].

Dacă sarcina redresorului, în principiu arbitrară, este indusul unui motor de curent continuu cu excitația 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 3.1.

Astfel, prin modificarea unghiului de comandă α 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 [36, 62].

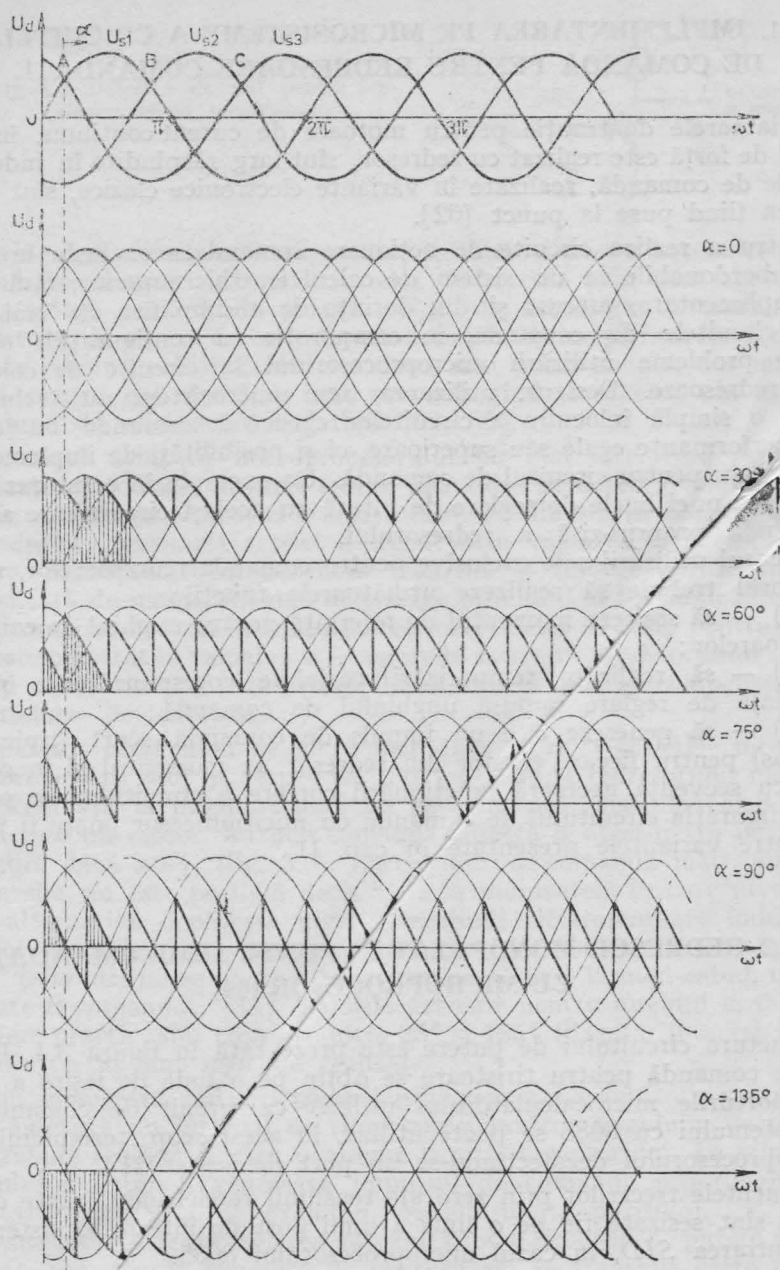


Fig. 3.7. Forma tensiunii de ieșire la diferite unghiuri de comandă pentru redresorul trifazat.

3.2.1. IMPLEMENTAREA PE MICROSISTEME 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 [62].

Pentru a realiza circuite de acționare comandate exclusiv numeric, direct subordonabile la un sistem de calcul ierarhic superior, fără interfațări suplimentare, precum și din dorința de a simplifica cît 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 cîte 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. II.

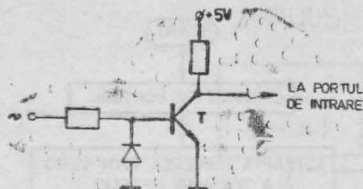
3.2.2. REDRESOR MONOFAZAT ÎN PUNTE SEMICOMANDATA CU MICROPROCESOR [64]

Structura circuitului de putere este prezentată în figura 3.4. Impulsurile de comandă pentru tiristoare se obțin pe o linie de ieșire a unui dintre porturile microcalculatorului utilizat ca circuit de comandă. În cazul sistemului cu 8085 se poate utiliza, în acest scop, terminalul *SOD* al microprocesorului dezafectîndu-se un port de ieșire [11].

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 figura 3.8, conectat cu exteriorul sistemului.

Fig. 3.8. 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_{ber} , de deschidere a tranzistorului, și, eventual, de viteza de comutare a acestuia. Principial, acest dezavantaj se poate elimina utilizând alte variante, mai puțin simple, dar mai costisitoare, pentru detectorul de nul, sau adăugând la schema din figura 3.8 o tensiune de alimentare negativă [62, 65].

În cazul utilizării microprocesorului nu mai este, însă, necesară o complicație hardware, întrucât 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 întârzierea acesteia față de momentul real de trecere prin zero al tensiunii alternative. Astfel, numărul utilizat în contorizarea intervalelor de timp pentru unghiurile α este corectat la valoarea $\alpha - \alpha_k$, unde α_k are valori de ordinul zecimilor de grad sexazecimal și este prestabilit în așa fel încât 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, întrucât comanda corectă a acestui tip de redresor se asigură dacă $\alpha > \gamma_1$ (fig. 3.5). La unghiuri de comandă mari, amorsarea tiristoarelor nu este posibilă decît pînă la momentele în care nivelul tensiunii alternative 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 0 V anod-catod, tiristorul nu poate fi comandat, [33]. Valorile extreme pentru unghiul α , din punct de vedere practic, sînt cuprinse între 25° și 155° [62], iar în acest interval, corecțiile cu α_k sînt ușor de realizat.

Din momentul sesizării trecerii prin zero a tensiunii alternative, sistemul așteaptă un interval de timp corespunzător unghiului de comandă α ce trebuie realizat pentru o anume turație a motorului. La sfîrșitul intervalului α sistemul generează impulsul de comandă propriu-zis unuia dintre tiristoare (T_1 , de exemplu, pe alternanța 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 2.1).

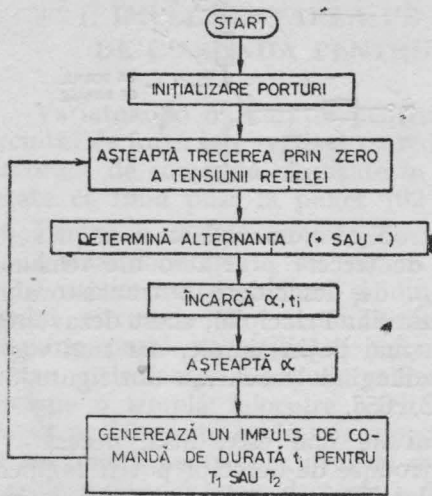


Fig. 3.9. Program de comandă pentru redresor monofazat monoalternanță cu participarea curentă a unității centrale.

Valoarea unghiului α , 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 α 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 figura 3.9.

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 acel tiristor care are tensiune anod-catod cu polaritate corespunzătoare [62].

Organigrama unui program concret întocmit și utilizat pentru comanda redresorului monofazat monoalternanță este cea din figura 3.10 [64].

Programele detaliate pentru comanda redresorului monofazat realizată cu microprocesoarele 8035, 8085 și Z80 sînt cele de mai jos.

8035	8085	Z80
DIS TCNT I	E1: MVI A,70H	E1: XOR A,A
JMP E0	SIM	OUT 62H,A
IN A,P1	RIM	IN A,61H
MOV R6,A	ANI 80H	AND A,01
RETR	MOV E,A	LD E,A
E0: MOV A,#80	E2: RIM	E2: IN A,61H
OUTL P2,A	ANI 80H	AND A,01
ORL P1,#FF	CMP E	CP A,E
EN I	JZ E2	JR Z,E2
IN A,P2	LXI H,2200H	LD HL,(3000H)
MOV R5,A	MOV B,M	AST: DEC HL
E1: IN A,P2	INR L	LD A,L
XRL A,R5	MOV C,M	OR A,H
JZ E1	INR L	JR NZ,AST
MOV A,R6	MOV D,M	INC A
MOV T,A	E3: DCX B	OUT 62H,A
STRT T	MOV A,C	LD B,t
E6: JTF E5	ORA B	IM: DJNZ IM
JMP E6	JNZ E3	JR E1
E5: STOP TCNT	MVI A,0FDH	
ORL P2,#C0	SIM	
MOV R7,#t _i	E4: DCR D	
E7: DJNZ R7,E7	JNZ E4	
JMP E0	JMP E1	

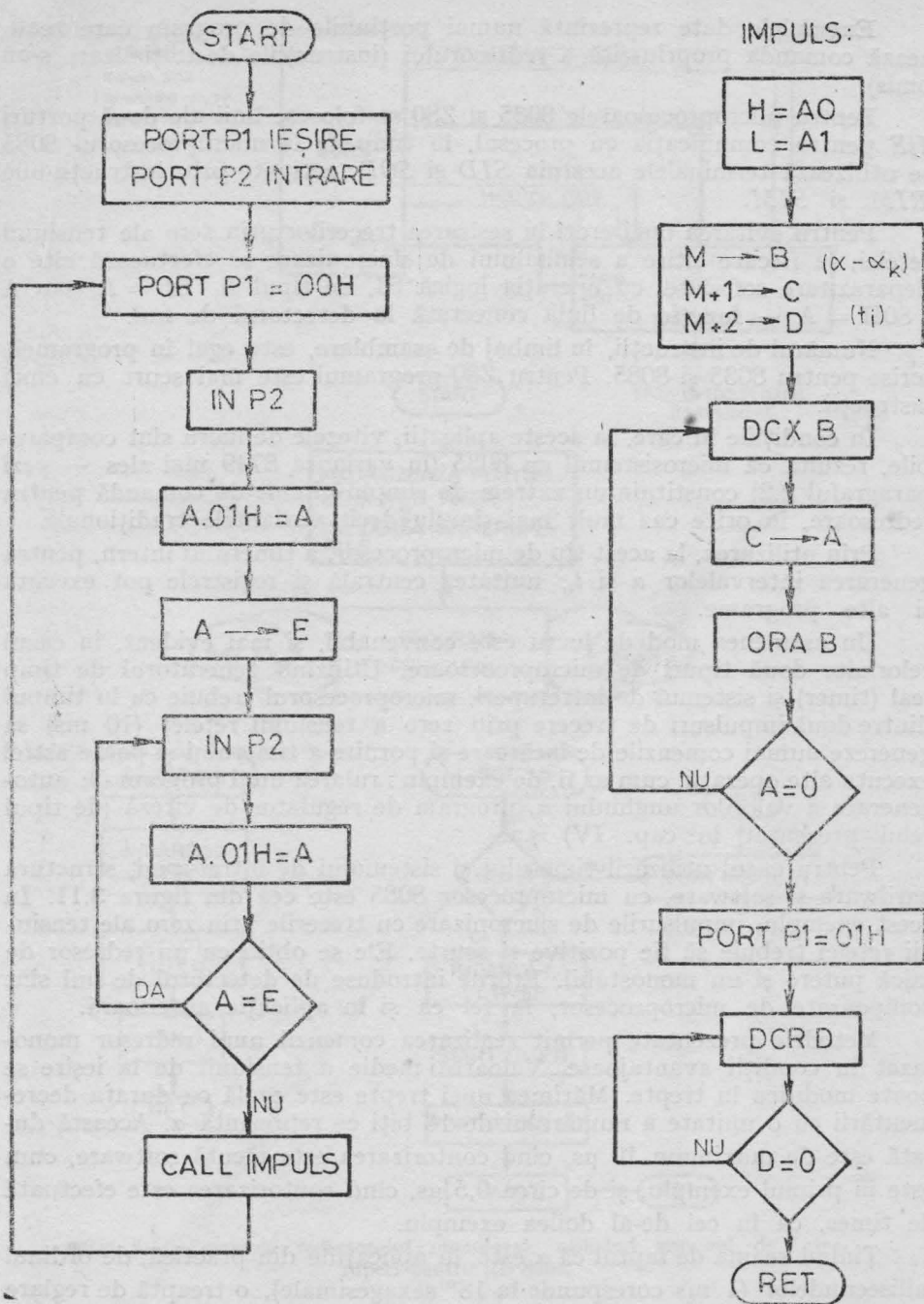


Fig. 3.10. Organigrama unui program utilizat la comanda redresorului monofazat.

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 $A \cdot 01 = A$, sau $A \cdot 80H = A$ — 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 2.2) 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 α și t_i , 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 auto-generare a valorilor unghiului α , program de regulator de viteză (de tipul celui prezentat în cap. IV) ș.a.

Pentru cazul utilizării timerului și sistemului de întreruperi, structura hardware și software, cu microprocesor 8085 este cea din figura 3.11. 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 și 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 α . Această durată este de maximum 10 μ s, cînd contorizarea este făcută software, cum este în primul exemplu, și de circa 0,5 μ s, cînd contorizarea este efectuată de timer, ca în cel de-al doilea exemplu.

Ținînd seama de faptul că α este, în aplicațiile din practică, de ordinul milisecundelor (1 ms corespunde la 18° sexagesimale), o treaptă de reglare a valorilor sale mai mică de 10 μ s asigură o rezoluție foarte bună (0,18°).

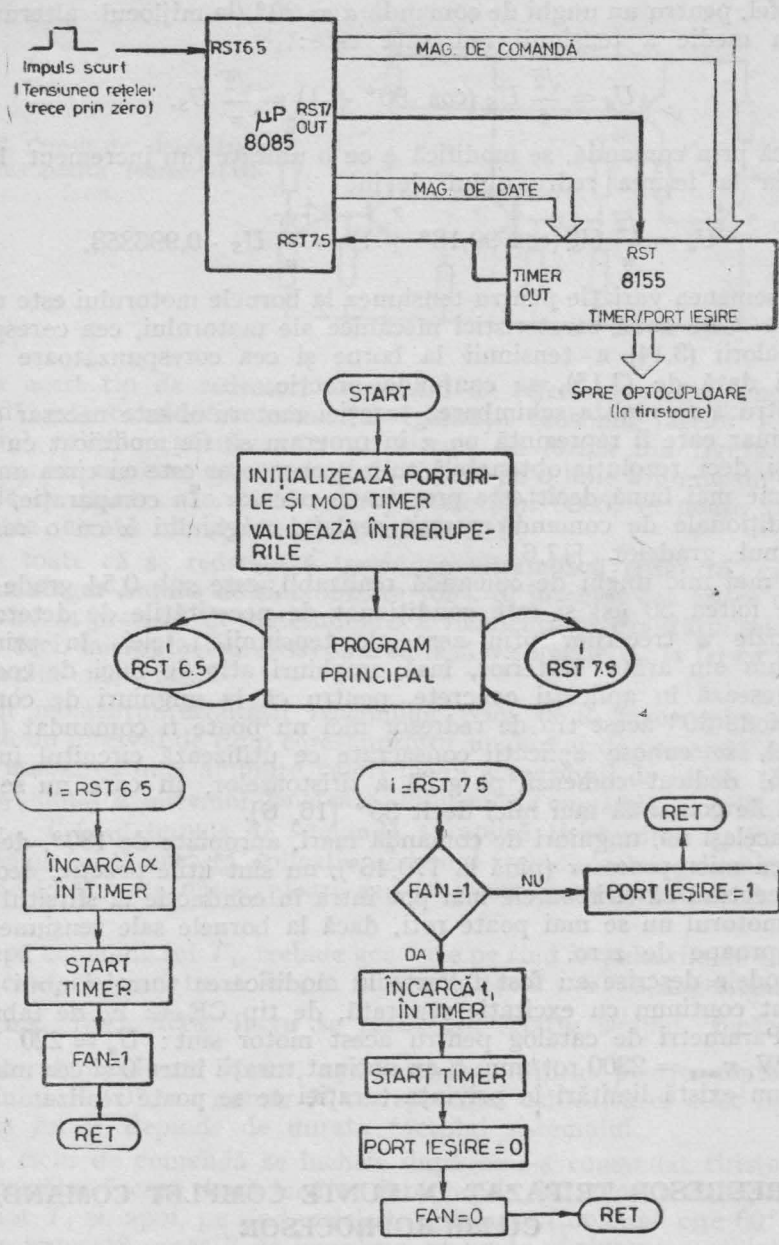


Fig. 3.11. Comanda redresorului monofazat, utilizând sistemul de întreruperi pentru μP 8085.

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

$$U_a = \frac{\sqrt{2}}{\pi} U_S (\cos 90^\circ + 1) = \frac{\sqrt{2}}{\pi} U_S. \quad (3.14)$$

Dacă prin comandă, se modifică α cu o unitate (un increment 10 μ s), tensiunea la ieșirea redresorului devine:

$$U_a = \frac{\sqrt{2}}{\pi} U_S (\cos 90,18^\circ + 1) = \frac{\sqrt{2}}{\pi} U_S \cdot 0,996858. \quad (3.15)$$

O asemenea variație pentru tensiunea la bornele motorului este nesemnificativă. Cele două caracteristici mecanice ale motorului, cea corespunzătoare valorii (3.14) a tensiunii la borne și cea corespunzătoare valorii tensiunii dată de (3.15), se confundă practic.

Pentru a se realiza schimbarea turației motorului este necesar ca numărul binar care îl reprezintă pe α în program să fie 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 α cu o rezoluție de ordinul gradelor [17,6].

Cel mai mic unghi de comandă realizabil, este sub 0,54 grade sexagesimale (circa 30 μ 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 atît de mici de comandă nu interesează în aplicații concrete, pentru că la unghiuri de comandă mai mici de 30° acest tip de redresor nici nu poate fi comandat ($\alpha > \gamma_1$). De altfel, se cunosc aplicații consacrate ce utilizează circuitul integrat β A A 145, dedicat comenzii pe grilă a tiristoarelor, în care nu se obțin unghiuri de comandă mai mici decât 30° [16, 6].

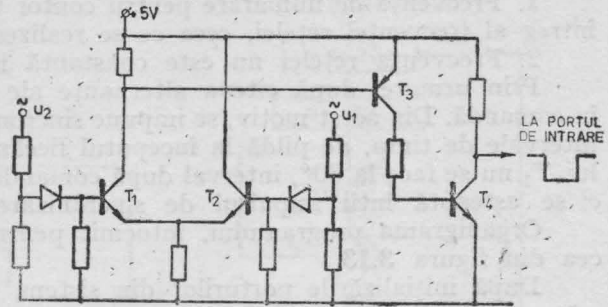
În același fel, unghiuri de comandă mari, apropiate de 180° , deși realizabile cu microprocesor (pînă la $179,46^\circ$), nu sînt utile practic, deoarece, chiar acceptînd că tiristoarele mai pot intra în conducție la sfîrșitul alternanței, motorul nu se mai poate roti, dacă la bornele sale tensiunea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

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

3.2.3. REDRESOR TRIFAZAT ÎN PUNTE COMPLET COMANDATA CU MICROPROCESOR

Structura circuitului de putere este prezentată în figura 3.6. În acest caz, impulsurile de comandă pentru tiristoare se obțin pe 6 linii ale unui port de ieșire din microsistem.

Fig. 3.12. 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 figura 3.12 se obține un impuls care este preluat de sistem pe o linie a 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 tensiunea alternativă trifazată, este suficient un singur impuls de sincronizare [69]. Și 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 α_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 α , impus pentru stabilirea unei anumite valori medii a tensiunii de ieșire. La sfârșitul intervalului α , sistemul generează impulsul de comandă pe grilă, pentru tiristoare. Primul impuls de comandă se aplică tiristorului T_1 [20].

Desigur, și la această aplicație, sistemul trebuie izolat galvanic de circuitul de forță, prin optocuplare 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° sexagesimale între ele $\left(\frac{10}{3} \text{ ms}\right)$ [69]. Acest lucru se realizează relativ simplu, încărcînd în numărătorul, hardware (timer) sau software, utilizat pentru contorizarea intervalului de 60° , un număr N ce reprezintă echivalentul celor 3,33 ms. Valoarea lui N depinde de durata tactului sistemului.

Un ciclu de comandă se încheie după ce s-a comandat tiristorul T_6 . Ciclul următor începe după încă un interval de 60° , cînd trebuie, din nou, comandat T_1 și, apoi, pe rînd, celelalte tiristoare (tot după cîte 60°). 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° , interval după comanda lui T_6 din ciclul precedent, ci se așteaptă întâi impulsul de sincronizare cu rețeaua.

Organigrama programului, întocmit pentru microprocesorul Z80 este cea din figura 3.13.

După inițializările porturilor din sistem, programul detaliat este următorul :

LD HL,ADR α	ET7: DEC B
ET1: LD D,0FFH	JR NZ,ET7
LD A,D	LD A,0FFH
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 [20], 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 α . După primul impuls de comandă la T_1 , același timer (timer 1) devine contorul pentru intervalul de 60° 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 (timer 2) la comanda precedentă și numărul N utilizat la contorizarea celor 60° . Dacă cele două valori nu sînt egale, se corectează N . Se observă că N , corectat după un impuls de comandă, se utilizează de fapt, abia la comanda următoare.

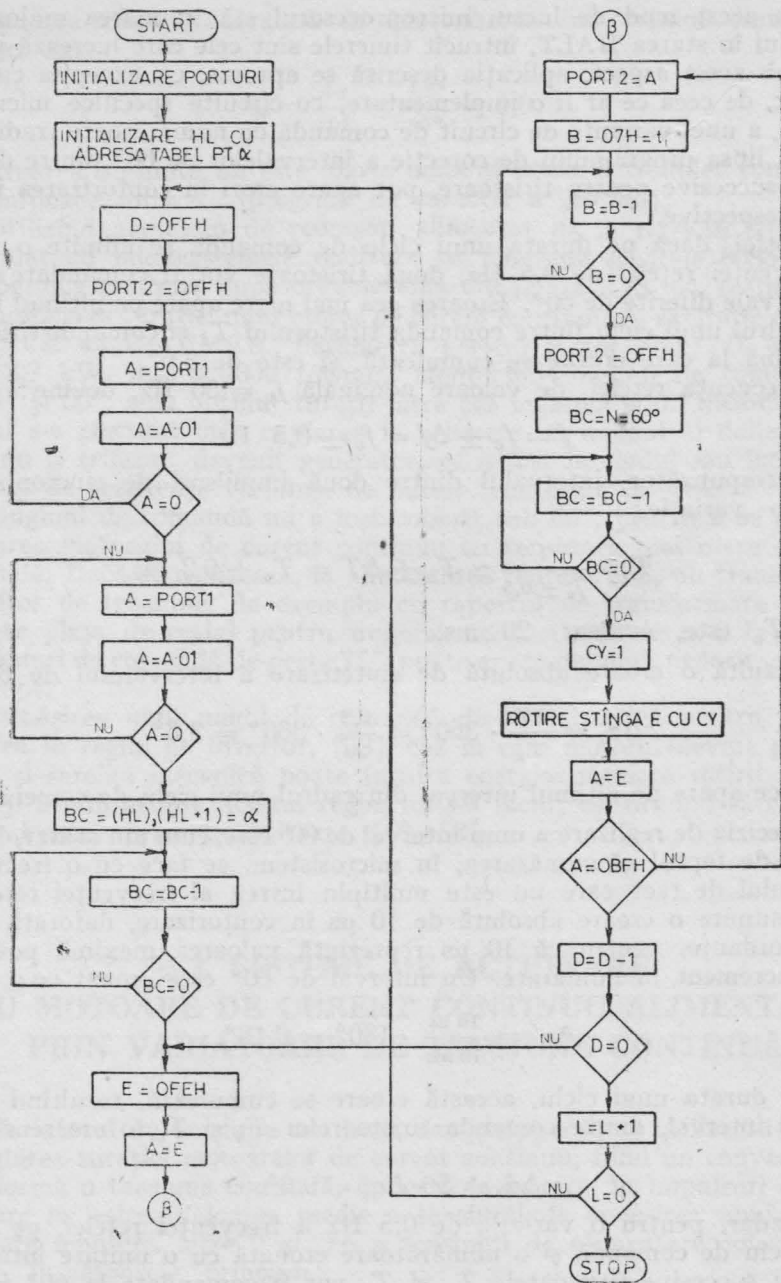


Fig. 3.13. Organigrama programului de comandă a redresorului trifazat.

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° 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° . 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:

$$f = f_0 \pm \Delta f = f_0 \pm 0,5 \text{ Hz.} \quad (3.16)$$

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

$$T = \frac{1}{f_0 \pm 0,5} \simeq T_0 \pm \Delta T = T_0 \pm 0,2 \text{ ms,} \quad (3.17)$$

unde T_0 este, evident, 20 ms.

Rezultă o eroare absolută de sintetizare a intervalului de 60° de:

$$\Delta\alpha_1 = \frac{\Delta T}{T_0} \cdot 360^\circ = \frac{0,2}{20} \cdot 360^\circ = 3,6^\circ. \quad (3.18)$$

eroare ce apare pe ultimul interval din cadrul unui ciclu de comenzi.

Precizia de realizare a unui interval de 60° este, cum am arătat, influențată și de faptul că numărarea, în microsistem, 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 μ s în contorizare, datorată acestei neconcordanțe, pentru că 10 μ s reprezintă valoarea maximă posibilă a unui increment în numărare. Un interval de 60° este eronat cu:

$$\Delta\alpha'_2 = \frac{10 \mu s}{10 \text{ ms}} \cdot 180^\circ = 0,18^\circ. \quad (3.19)$$

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

$$\Delta\alpha_2 = 5 \cdot \Delta\alpha'_2 = 0,9^\circ. \quad (3.20)$$

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° interval cu o eroare absolută maximă totală de:

$$\Delta\alpha = \Delta\alpha_1 + \Delta\alpha_2 = 4,5^\circ. \quad (3.21)$$

Eroarea relativă maximă în contorizarea intervalului dintre două comenzi este:

$$\varepsilon_r = \frac{\Delta\alpha}{60^\circ} = \frac{4,5^\circ}{60^\circ} = 0,075. \quad (3.22)$$

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 micro sistemele realizate cu microprocesoarele 8085 și Z80, cu programe realizate după organigrama din figura 3.13, s-a realizat practic modificarea turației unui motor de curent continuu cu excitație separată, de tip CE 42 E—IMEB. Parametrii acestui motor sînt: $U_a = 220$ V, $P_n = 2,7$ kW, $n_{max} = 2300$ rot/min. Modificînd unghiul de comandă, între 60° și 90° , s-au obținut turații între cea maximă și 0. Încărcarea 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 coborît sub 60° , pentru a se evita alimentarea motorului de curent continuu cu tensiunea mai mare decît 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° , dar la unghiuri de comandă de peste 75° , poate apare regimul, nedorit, de curent întrerupt.

Depășirea unui unghi de comandă de 90° , implică pentru redresor, intrarea în regim de invertor, [28], caz în care motorul devine generator (dacă și sarcina mecanică poate furniza energia necesară rotirii arborelui motor). Studii asupra acestui regim nu s-a făcut, ele urmînd să constituie obiectul unor cercetări viitoare [49, 52].

3.3. 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 0 și cea a tensiunii de alimentare, folosind unul dintre următoarele principii:

- modificarea frecvenței de repetiție a unor impulsuri de durată constantă;
- modularea în durată a unor impulsuri de frecvență constantă.

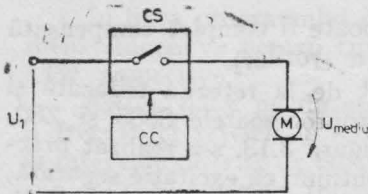


Fig. 3.14. Schema de principiu a variatorului de tensiune continuă.

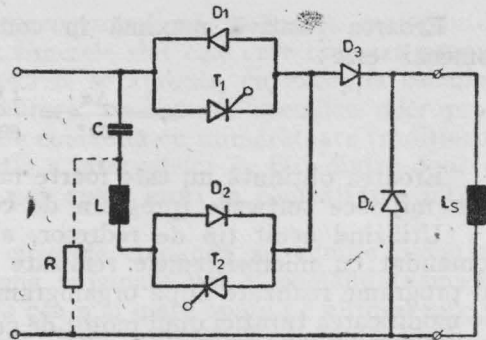


Fig. 3.15. Chopper cu stingere forțată.

În acest fel este posibil să se efectueze reglajul prin tensiune al turației motoarelor de curent continuu [36, 22, 31].

Schema de principiu a unui astfel de sistem de acționare este prezentată în figura 3.14. În această figură, CS reprezintă un contactor static (chopper), CC circuitul său de comandă și M indusul motorului de curent continuu.

Închizând și deschizând contactorul CS, după o secvență repetată periodic, motorul va fi alimentat cu tensiunea medie:

$$U_m = \frac{T_c}{T} U_1, \quad (3.23)$$

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 decât constanta de timp electromagnetice a motorului, [36].

O variantă concretă pentru contactorul static din variator este chopperul cu stingere forțată, prezentat în figura 3.15. 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 câte o schimbare de stare (conducție sau blocare) pentru dispozitivele din circuit [65].

În figura 3.16 sînt reprezentate schemele echivalente ale chopperului pe fiecare dintre cele șase intervale de timp, iar în figura 3.17 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 presupune 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 figura 3.16, b (condensatorul

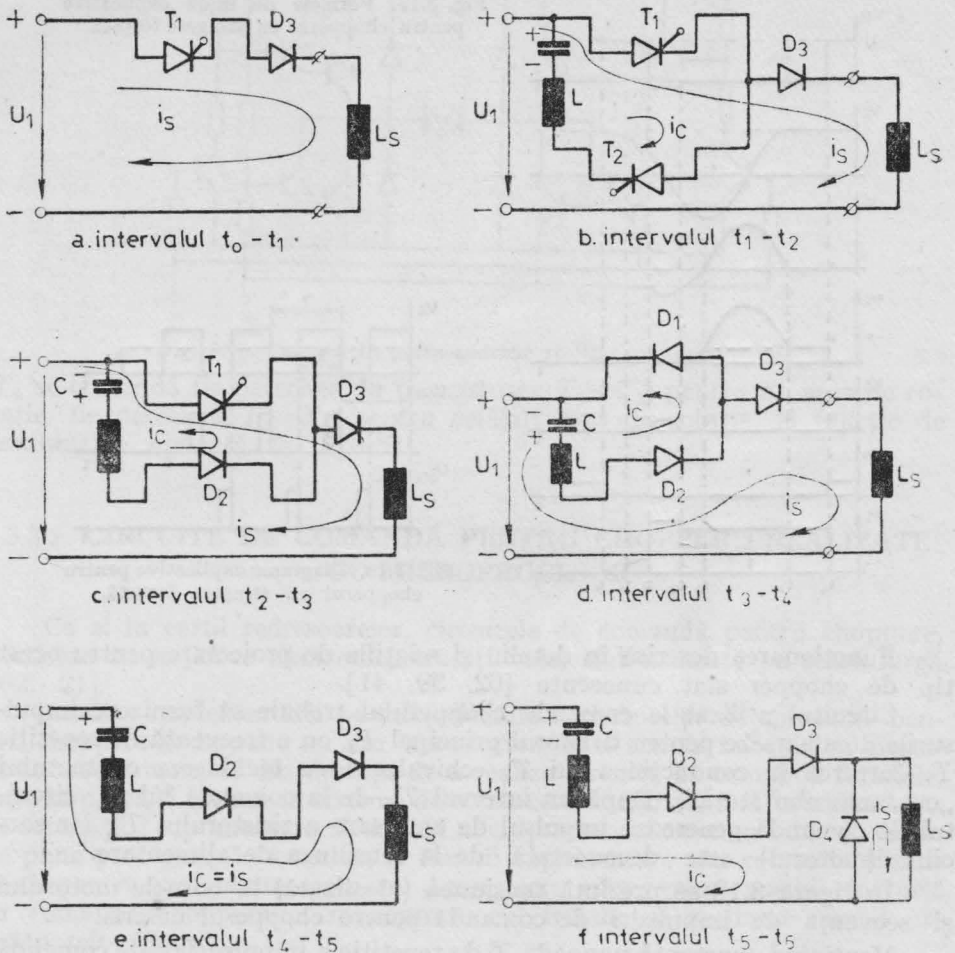


Fig. 3.16. Circuitul echivalent al chopperului în diferite momente ale funcționării.

s-a încărcat în momentul conectării sursei de alimentare). Curentul rezonant de descărcare a lui C se anulează 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 curenților 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 C este mai mare decât 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 momentul 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 .

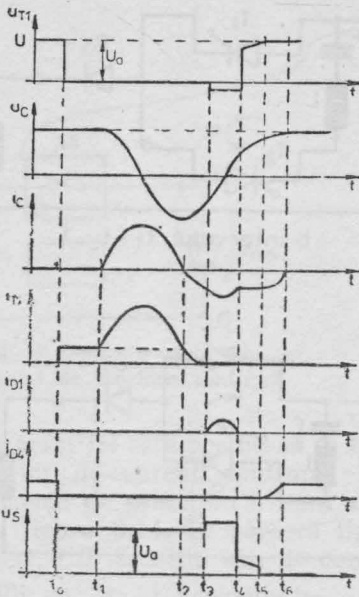


Fig. 3.17. Formele de undă explicative pentru chopperul cu stingere forțată.

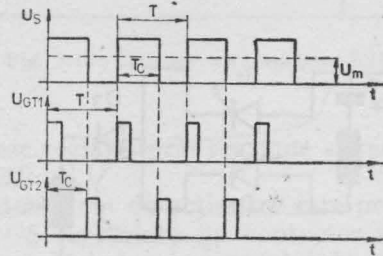


Fig. 3.18. Diagrame explicative pentru chopperul cu stingere forțată.

Funcționarea descrisă în detaliu și relațiile de proiectare pentru acest tip de chopper sînt cunoscute [62, 39, 41].

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.

În figura 3.18 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, implicit, modificarea turației motorului.

Chopperul prezentat are dezavatajul 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 răspîndire în practică [61].

În figura 3.19 se prezintă o variantă de chopper realizată cu tranzistoare care funcționează în patru cadrane, [66].

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

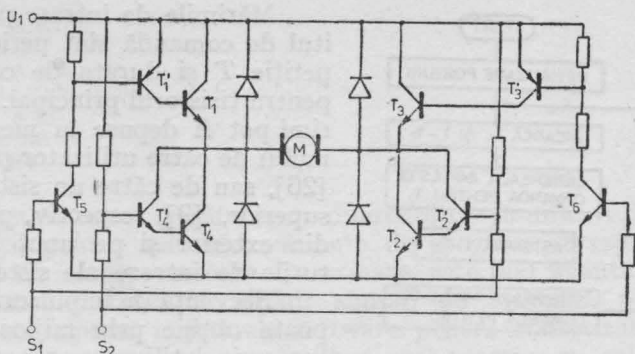


Fig. 3.19. Chopper în patru cadrane realizat cu tranzistoare.

T_6 se comandă fie perechea de tranzistoare T_1-T_2 , pentru un sens de rotație, fie perechea T_3-T_4 , pentru celălalt sens de rotație, în funcție de combinația aplicată la S_1-S_2 .

3.3.1. CIRCUITE DE COMANDĂ PENTRU CHOPPERE REALIZATE CU MICROPROCESOR

Ca și în cazul redresoarelor, circuitele de comandă pentru chopper, realizate cu mijloace electronice tradiționale, sînt utilizate pe scară largă [62, 21].

Analiza implementării pe microsisteme a circuitelor de comandă pentru chopper se impune nu din necesitatea unei simple înlocuiri a circuitelor tradiționale, ci 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 .

3.3.2. CHOPPER CU STINGERE FORȚATĂ COMANDAT CU MICROPROCESOR

Structura circuitului de putere utilizat în aplicația descrisă în acest paragraf este cea din figura 3.15. 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 2.1, figura 2.3).

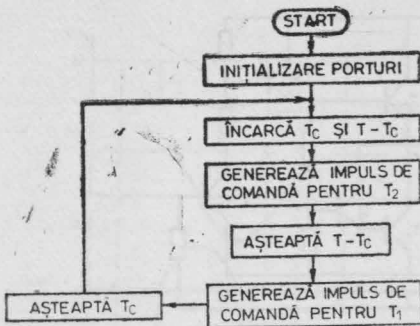


Fig. 3.20. Organigrama programului de comandă pentru chopper.

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, [26], sau de către un sistem ierarhic superior, [9], 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 figura 3.20 este reprezentat un exemplu de program care realizează comanda chopperului.

Programele detaliate pentru comanda chopperului utilizînd microprocesoarele 8035, 8085 și Z80 sînt cele de mai jos [11, 9].

8035	8085	Z80
DIS TCNT I	MVI A,03	LD C,61H
JMP E0	OUT 20H	LD D,02
IN A,P2	E1: LXI H,83A0H	LD HL,T-T _c
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 _c
E1: IN A,P1	OUT 21H	E1: OUT (C),D
MOV R7,A	CALL IMP	LD B,t _i
E2: JFO E3	INR L	IM: DJNZ IM
RETR	MOV B,M	OUT (C),B
E0: CPL F0	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
EN I	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	
EN I	JMP E1	
E8: MOV R4,#t _i	AST: DCX B	
E5: DJNZ R4,E5	MOV A,C	
ANL P2,#80	ORA B	
MOV T,A	JNZ AST	
STR T	RET	
E7: JTF E6	IMP: LXI B,00t _i	
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

```

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.

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 [9] 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. Atît valorile turațiilor, cît și durata de menținere a unei anumite trepte pot fi modificate după necesități. Organigrama programului este dată în figurile 3.21 și 3.22. Ulterior acest program a fost extins la mai multe trepte de turație realizabile [48, 73].

O variantă de program care realizează 20 de trepte de viteză pentru motor este cea de mai jos:

ORG 8000 H

0001	E1 EQU 9190H	0230	CPI 9FH
0002	E2 EQU 9194H	0240	JZ A3
0003	E3 EQU 919AH	0250	CPI 0A5H
0004	E4 EQU 91A0H	0260	JZ A4
0010	START MVI A,01H	0270	CPI 0ABH
0020	OUT 28H	0280	JZ A5
0030	IAR LXI H,E1	2090	CPI 0B1H
0040	REF MOV D,M	0300	JZ A6
0050	INR L	0310	CPI 0B7H
0060	MOV E,M	0320	JZ A7
0070	MVI A,01H	0330	CPI 0BDH
0080	OUT 29H	0340	JZ A8
0090	CALL IMP	0350	CPI 0C3H
0092	INR L	0360	JZ A9
0095	MOV B,M	0370	CPI 0C9H
0097	INR L	0380	JZ A10
0100	MOV C,M	0390	CPI 0CFH
0110	CALL AST	0400	JZ A11
0120	MVI A,02H	0410	CPI 0D5H
0130	OUT 29H	0420	JZ A12
0140	CALL IMP	0430	CPI 0DBH
0150	MOV C,E	0440	JZ A13
0160	MOV B,D	0450	CPI 0E1H
0170	CALL AST	0460	JZ A14
0180	MOV A,L	0470	CPI 0E7H
0190	CPI 93H	0480	JZ A15
0200	JZ A1	0490	CPI 0EDH
0210	CPI 99H	0500	JZ A16
0220	JZ A2	0510	CPI 0F3H

0520 JZ A17
0530 CPI 0F9H
0540 JZ A18
0550 CPI 0FFH
0560 JZ A19
0570 A1 LXI H,E2
0580 CALL TR
0590 A2 LXI H,E3
0600 CALL TR
0610 A3 LXI H,E4
0620 CALL TR
0630 A4 LXI H,E5
0640 CALL TR
0650 A5 LXI H,E6
0660 CALL TR
0670 A6 LXI H,E7
0680 CALL TR
0690 A7 LXI H,E8
0700 CALL TR
0710 A8 LXI H,E9
0720 CALL TR
0730 A9 LXI H,E10
0740 CALL TR
0750 A10 LXI H,E11
0760 CALL TR
0770 A11 LXI H,E12
0780 CALL TR
0790 A12 LXI H,E13
0795 CALL TR
0800 A13 LXI H,E14
0810 CALL TR
0820 A14 LXI H,E15
0830 CALL TR
0840 A15 LXI H,E16
0850 CALL TR
0860 A16 LXI H,E17
0870 CALL TR
0880 A17 LXI H,E18
0890 CALL TR
0900 A18 LXI H,E19
0910 CALL TR
0920 A19 LXI H,E20
0930 CALL TR
0940 LXI H,E20
0950 CALL INC
0960 LXI H,E3
0970 CALL INC
0980 LXI H,E4
0990 CALL INC
1000 LXI H,E5
1010 CALL INC
1020 LXI H,E6
1030 CALL INC
1040 LXI H,E7
1050 CALL INC
1060 LXI H,E8
1070 CALL INC
1080 LXI H,E9
1090 CALL INC
1100 LXI H,E10

1110 CALL INC
1120 LXI H,E11
1130 CALL INC
1140 LXI H,E12
1150 CALL INC
1160 LXI H,E13
1170 CALL INC
1180 LXI H,E14
1190 CALL INC
1200 LXI H,E15
1210 CALL INC
1220 LXI H,E16
1230 CALL INC
1240 LXI H,E17
1250 CALL INC
1260 LXI H,E18
1270 CALL INC
1280 LXI H,E19
1290 CALL INC
1300 LXI H,E20
1310 CALL INC
1320 JMP IAR
1330 AST DCX B
1340 DCX D
1350 MOV A,C
1360 ORA B
1370 JNZ AST
1380 RET
1390 IMP MVI C,07H
1400 MVI B,0H
1410 CALL AST
1420 OUT 29H
1430 RET
1440 TR MOV B,M
1450 INR L
1460 MOV C,M
1470 DCX B
1480 MOV M,C
1490 DCR L
1500 MOV M,B
1510 DCR L
1520 DCR L
1530 DCR L
1540 DCR L
1550 MOV A,C
1560 ORA B
1570 RZ
1580 POP D
1590 JMP REF
1600 INC MVI M,0H
1610 INR L
1620 MVI M,10H
1630 RET
1640 E5 EQU 91A6H
1650 E6 EQU 91ACH
1660 E7 EQU 91B2H
1670 E8 EQU 91B8H
1680 E9 EQU 91BEH
1690 E10 EQU 91C4H
1700 E11 EQU 91CAH

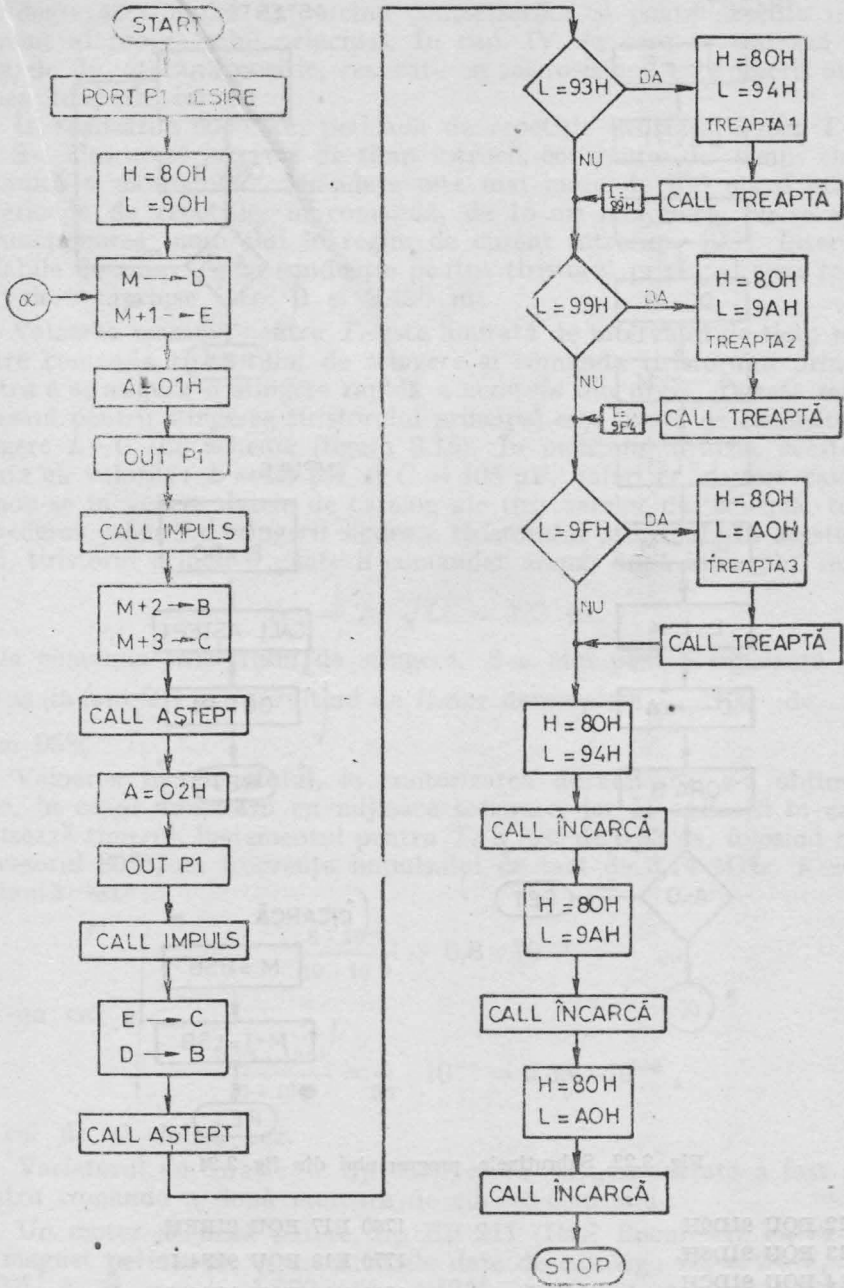
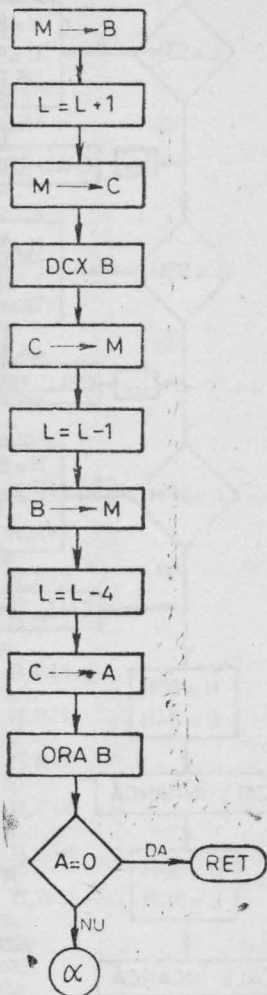
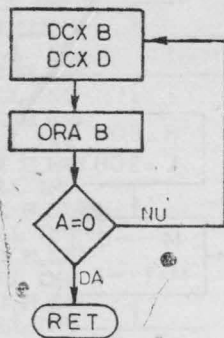


Fig. 3.21. Organigrama programului de comandă prin chopper a 3 trepte de viteză pentru un motor de c.c.

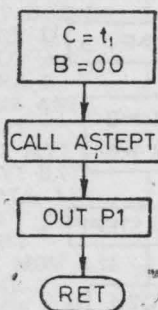
TREAPTĂ:



ASTEPT:



IMPULS:



ÎNCARCĂ

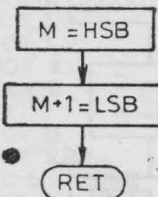


Fig. 3.22. Subrutinele programului din fig. 3.21.

1710 E12 EQU 91D0H
 1720 E13 EQU 91D6H
 1730 E14 EQU 91DCH
 1740 E15 EQU 91E2H
 1750 E16 EQU 91E8H

1760 E17 EQU 91EEH
 1770 E18 EQU 91F4H
 1780 E19 EQU 91FAH
 1790 E20 EQU 9200H

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

În realizările concrete, perioada de repetiție utilizată a fost $T = 10$ ms. S-a ales acest interval de timp întrucât constanta de timp electro-mecanică a motoarelor comandate este mai mare de 150 ms și utilizând o perioadă de repetiție, în comandă, de 15 ori mai mică, nu se ajunge la funcționarea motorului în regim de curent întrerupt [28]. 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 rapidă a acestuia din urmă. Durata minimă necesară pentru stingerea tiristorului principal este fixată de elementele de stingere $L-C$ din schemă (figura 3.15). În montajul utilizat aceste elemente au valorile: $L = 25 \mu\text{H}$ și $C = 105 \mu\text{F}$, valori ce au fost calculate avându-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;

$$\Delta T = 2\pi \sqrt{LC} = 322 \mu\text{s}. \quad (3.24)$$

de la comanda tiristorului de stingere. S-a ales pentru siguranță $\Delta T \approx 500 \mu\text{s}$ (în fapt 504 μs), rezultând un factor de umplere $\frac{T}{T_C} \cdot 100$ de maximum 95%.

Valoarea incrementului, la contorizarea duratei T_C , s-a obținut de 8 μs , în cazul numărării cu mijloace software, iar în aplicații în care se utilizează timerul, incrementul pentru T_C a fost de 0,33 μs , folosind microprocesorul 8085, cu frecvența impulsului de tact de 3,14 MHz. Rezoluția obținută este:

$$\frac{8 \cdot 10^{-6}}{10 \cdot 10^{-3}} = 0,8 \cdot 10^{-3}, \quad (3.25)$$

într-un caz și

$$\frac{0,33 \cdot 10^{-6}}{10 \cdot 10^{-3}} = \frac{1}{30} \cdot 10^{-3} = 3,33 \cdot 10^{-6}, \quad (3.26)$$

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 excitație 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 parametri: $U_n = 220$ 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.

3.3.3. CHOPPER CU TRANZISTOARE COMANDAT CU MICROPROCESOR [66]

Structura circuitului de putere este prezentată în figura 3.19. 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 [66].

În funcție de sensul de rotație al rotorului, necesar la un moment dat, una dintre cele două linii, dintr-un port de ieșire, utilizate la comanda chopperului, este poziționată pe nivel 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 $\frac{1}{T}$ constantă, dar cu factor de umplere variabil, funcție de viteza cu care trebuie să se rotească rotorul. Pentru celălalt sens de rotație, impulsurile 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 figura 3.23. Programul utilizează o listă de date din memoria sistemului. 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

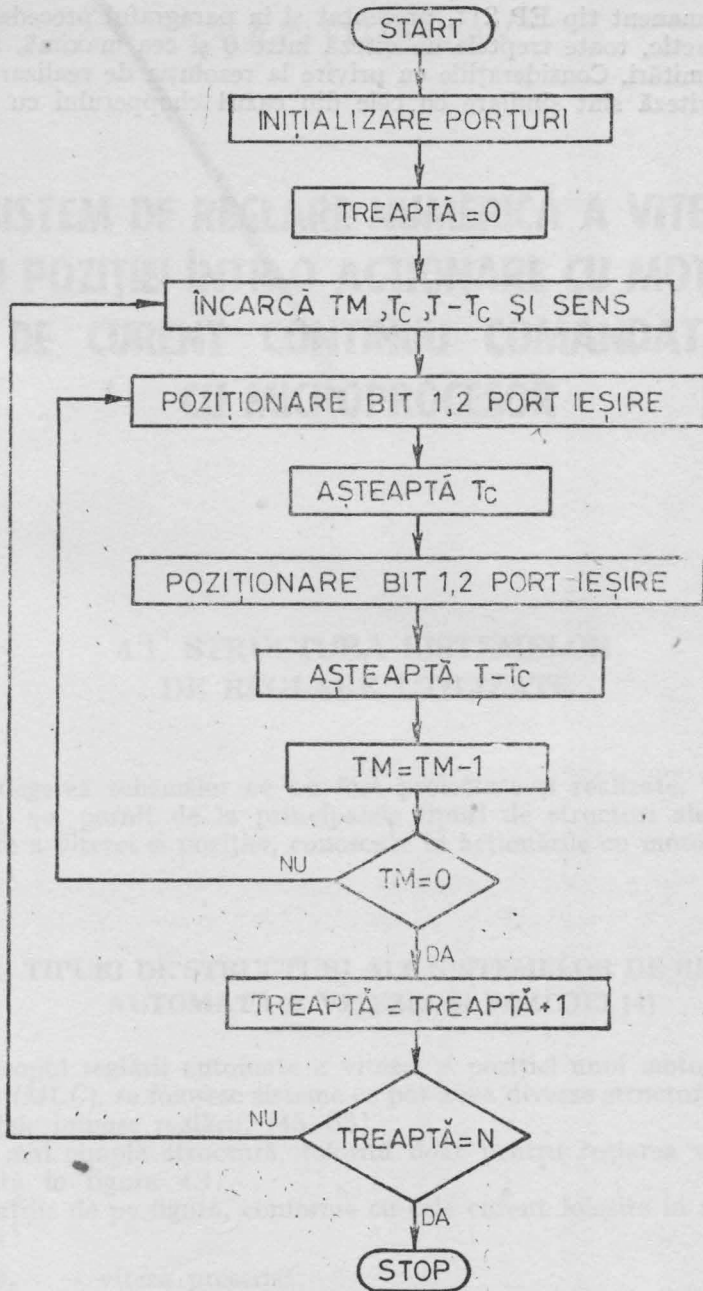


Fig. 3.23. Organigrama programului de comandă a chopperului cu tranzistoare.

magnet permanent tip EP 211, prezentat și în paragraful precedent. S-au obținut, practic, toate treptele de viteză între 0 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.

SISTEM DE REGLARE NUMERICĂ A VITEZEI ȘI POZIȚIEI ÎNTR-O ACȚIONARE CU MOTOR DE CURENT CONTINUU COMANDATĂ CU MICROPROCESOR

4.1. STRUCTURA SISTEMELOR DE REGLARE UTILIZATE

În alegerea schemelor ce au fost proiectate și realizate, în variantă 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.

4.1.1. TIPURI DE STRUCTURI ALE SISTEMELOR DE REGLARE AUTOMATĂ A VITEZEI ȘI POZIȚIEI [4]

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, [45, 33].

Cea mai simplă structură, folosită doar pentru reglarea vitezei, este prezentată în figura 4.1.

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

- w_v — viteza prescrisă,
- r_v — viteza măsurată (reacția de viteză),
- $RG-V$ — regulatorul de viteză,
- y — mărimea de comandă,

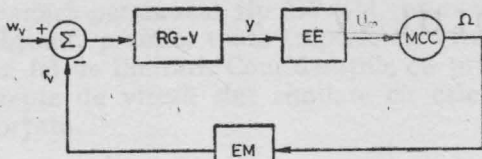


Fig. 4.1. Structura unui sistem de reglare a vitezei unui motor de c.c.

- EE — elementul de execuție,
- U_m — mărimea de execuție (tensiunea medie aplicată la bornele motorului),
- Ω — viteza unghiulară,
- EM — elementul de măsură.

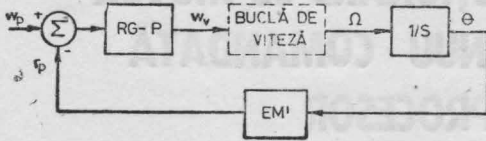


Fig. 4.2. Structura unui sistem de reglare a vitezei și poziției cu două bucle de reglare legate în cascadă.

Pentru a se asigura atât reglarea vitezei cât și a poziției, sînt necesare două bucle de reglare. Acestea se pot lega în cascadă (figura 4.2) sau în paralel (figura 4.3). Notățiile de pe figuri au semnificații similare cu cele prezentate pentru figura 4.1.

Regulatele folosite uzual sînt de următoarele tipuri :

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

Structura lor poate fi analogică sau numerică. O variantă analogică de realizare a schemei din figura 4.2, care include și o buclă internă de curent, este prezentată în figura 4.4. Proiectarea concretă a regulatelelor schemei este prezentată în [44].

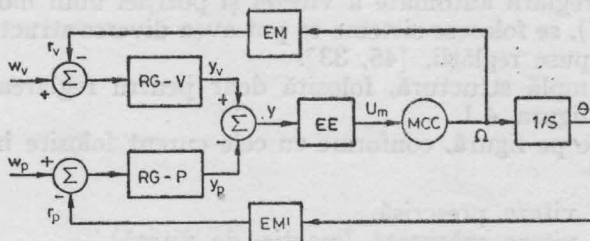


Fig. 4.3. Structura unui sistem de reglare a vitezei și poziției cu două bucle de reglare legate în paralel.

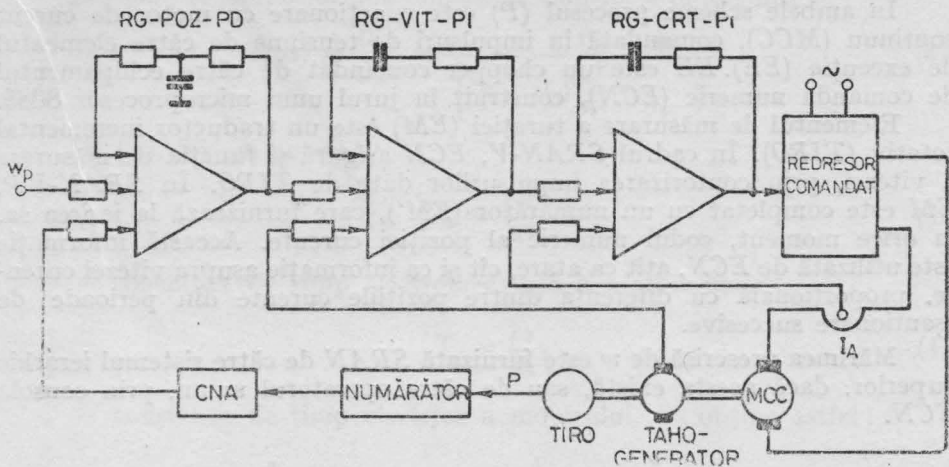


Fig. 4.4. Schemă de reglare a vitezei și poziției cu reguloare analogice.

4.1.2. 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 4.5 și 4.6.

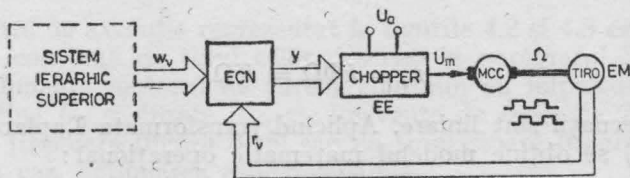


Fig. 4.5. Structura sistemului de reglare automată numerică a vitezei (SRA-V) realizată.

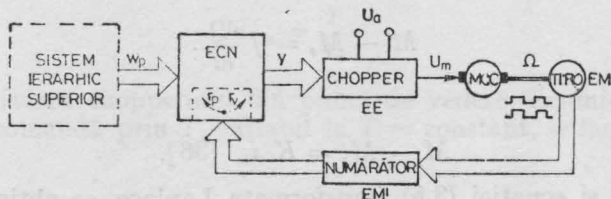


Fig. 4.6. Structura sistemului de reglare automată numerică a vitezei și poziției (SRAN-VP) realizată.

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.

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

4.1.3. PROCESUL REGLAT

4.1.3.1. MODELUL MATEMATIC OPERAȚIONAL AL MCC

În capitolul III am arătat ecuațiile care descriu funcționarea unui motor de curent continuu. Ecuația (3.6), dacă se ține seama de faptul că motorul se alimentează în impulsuri de tensiune și deci intervine și inductivitatea înfășurării devine:

$$u = u_e + R_i i_i + L_i \frac{di_i}{dt}, \quad (4.1)$$

cu

$$u_e = k\Phi\Omega = K\Omega. \quad (3.7)$$

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

$$U(s) = (R_i + sL_i)I_i(s) + K\Omega(s). \quad (4.2)$$

Ecuația de mișcare a motorului este:

$$M - M_s = J \frac{d\Omega}{dt}. \quad (3.5)$$

Dar

$$M - M_s = K i_i, \quad [36]. \quad (4.3)$$

Aplicînd și ecuației (3.5) transformata Laplace, se obține:

$$K I_i(s) = sJ\Omega(s). \quad (4.4)$$

Din (4.2) și (4.4) se obține funcție de transfer a procesului :

$$H_p(s) = \frac{\Omega(s)}{U(s)} = \frac{1}{\left[(R_i + sL_i) \frac{sJ}{K^2} + 1 \right] \cdot K} \quad (4.5)$$

În [28, 36] se notează :

$$T_m = \frac{JR_i}{K^2}, \quad (4.6)$$

— constanta de timp electromecanică a motorului și

$$T_e = \frac{L_i}{R_i}, \quad (4.7)$$

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

$$H_p(s) = \frac{\frac{1}{K}}{s^2 T_m T_e + s T_m + 1} \quad (4.8)$$

Pentru un tip 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 :

$$H_p(s) = \frac{\frac{1}{K}}{(1 + s T_m)(1 + s T_e)} \quad (4.9)$$

4.1.3.2. MODELUL MATEMATIC AL ELEMENTULUI DE EXECUȚIE

Elementul de execuție reprezentat în figurile 4.2 și 4.3 este un variator de tensiune continuă de tipul celor descrise în paragraful 3.3. 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 (figura 3.15) sau cu tranzistoare (figura 3.19). În cel de al doilea caz, acționarea este reversibilă.

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

$$U_m = \frac{T}{T_c} U_1 \quad (3.23)$$

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 :

$$H_c(s) = \frac{U_m(s)}{T_c(s)} = \frac{U_1}{T} \quad (4.10)$$

4.1.3.3. 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 [72].

Traductorul utilizat în cazul de față este un traductor incremental rotativ optic TIRO 1000, produs de Întreprinderea de 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 $\pm \frac{T}{4}$, 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 rezoluția de $0,18^\circ$ sexagesimale.

Principiul de funcționare al traductorului se explică pe baza schițelor și diagramelor din figura 4.7, [50].

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

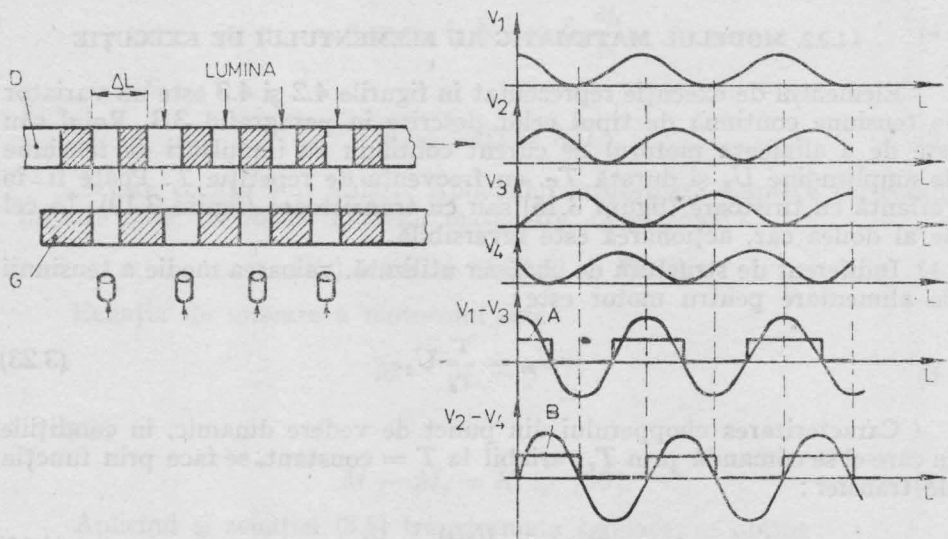
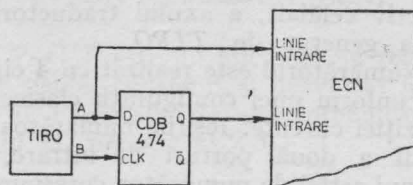


Fig. 4.7. Principiul de funcționare al traductorului TIRO.

Fig. 4.8. Determinarea sensului de rotație al *TIRO* cu bistabil *D*.



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 figura 4.8. 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 figura 4.9 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 cîte o linie de port de intrare în sistem, se obțin impulsurile în concordanță cu sensul de rotație.

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 incrementat sau decrementat cu cîte o unitate, la fiecare deplasare unghiulară de $0,36^\circ$, într-un sens,

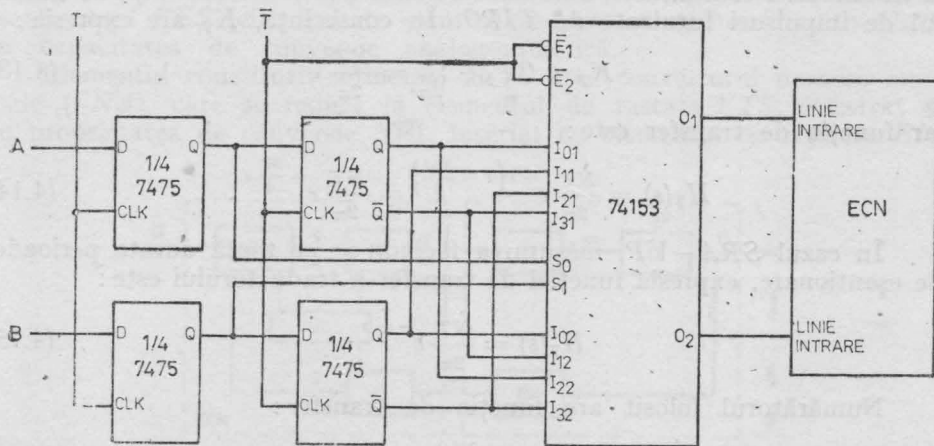


Fig. 4.9. Schemă pentru determinarea sensului de rotație al *TIRO*.

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 a pasul în 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 atinsă la mijlocul perioadei de măsură. Ca urmare, măsurarea este afectată de un timp mort $-\tau_M$:

$$\tau_M = T - \frac{T_M}{2}. \quad (4.11)$$

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:

$$H_T(s) = \frac{N(s)}{\Omega(s)} = K_T e^{-s \tau_M}, \quad (4.12)$$

în care K_T (coeficientul de transfer) și τ_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:

$$K_T = 2T_M \cdot \frac{N}{2\pi} = \frac{NT}{2\pi}, \quad (4.13)$$

iar funcția de transfer este:

$$H_T(s) = \frac{NT}{2\pi} e^{-s \left(T - \frac{T_M}{2} \right)} = \frac{NT}{2\pi} e^{-\frac{3T}{4}s}. \quad (4.14)$$

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

$$H_T(s) = \frac{NT}{2\pi} e^{-\frac{T}{2}s}. \quad (4.15)$$

Numărătorul folosit are funcția de transfer:

$$H_N(s) = \frac{N(s)}{\Omega(s)} = \frac{N}{2\pi}. \quad (4.16)$$

4.1.4. ECHIPAMENTUL DE COMANDĂ NUMERICĂ (ECN)

ECN utilizat în $SRA-V$ și $SRAN-VP$ este un microsistem realizat în jurul microprocesorului 8085. Acest sistem construit și dezvoltat în laboratorul de Electronică industrială al Facultății de electrotehnică din cadrul IPTVT, este descris în capitolul II, paragraful 2.3.

4.2. PROIECTAREA ALGORITMILOR DE REGLARE NUMERICĂ

4.2.1. PROIECTAREA ALGORITMICĂ A UNUI SISTEM CONVENȚIONAL DE REGLARE AUTOMATĂ NUMERICĂ [4, 24]

Un sistem de reglare automată numerică (SRAN) convențional are structura din figura 4.10, [24, 77]. Pe figură, semnificațiile pentru ECN, E , EM , y , Ω , r au fost prezentate în paragraful 4.1, 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 — perturbația,
- \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 ,

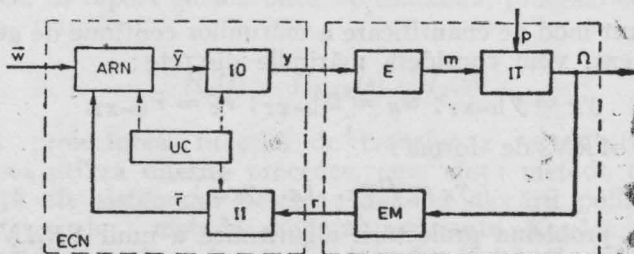


Fig. 4.10. Structura unui SRAN convențional.

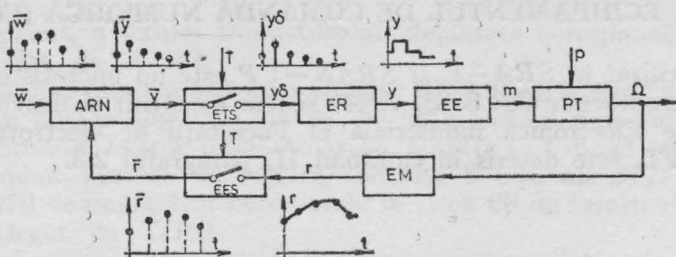


Fig. 4.11. Schema bloc a SRAN convențional.

destinat refacerii semnalului de comandă continuu. În general, eșantionorul și elementul de tastare lucrează sincron, cu aceeași perioadă, numită perioadă de eșantionare (T).

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 figura 4.11. 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, avînd funcția de transfer:

$$H_{ER}(s) = \frac{1 - e^{-sT}}{s}. \quad (4.17)$$

În $SRAN$, rolul regulatorului îl joacă ARN , ECN elaborînd^{*} 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:

$$t_K = mT, \quad m = \text{întreg}, \quad (4.18)$$

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

$$y_K = \bar{y}|_{t=KT}; \quad w_K = \bar{w}|_{t=KT}; \quad r_K = \bar{r}|_{t=KT}, \quad (4.19)$$

respectiv ARN de forma:

$$y_K = f(w_K, r_K, y_{K-1}). \quad (4.20)$$

Principial, problema proiectării algoritmice a unui $SRAN$ convențional este similară cu problematica proiectării algoritmice a unui SRA continuu convențional [24].

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 algoritmi de reglare numerică cvasicontinuă (*ARNC*).

Ipoteza 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*, cît și *EES*, să aibă funcția de transfer $H_{EER}(s) \simeq 1$. În acest scop, grupul *ETS* + *ER* se echivalează cu un singur element de transfer, *EER*, numit element de tastare și reținere, avînd funcția de transfer:

$$H_{EER}(s) = \frac{1 - e^{-sT}}{sT} \quad (4.21)$$

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

$$H_{EER}(s) = e^{-\frac{sT}{2}} \quad (4.22)$$

Proiectarea *ARNC* decurge conform organigramei din figura 4.12 [24].

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 (4.21) sau o expresie de aproximare, de exemplu (4.22). Se proiectează apoi regulatorul continuu, în ipoteza că, în raport cu mărimea de comandă, procesul condus prezintă funcția de transfer:

$$H_p(s) = H_{EER}(s) \cdot H_p(s) \quad (4.23)$$

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 [14] 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.

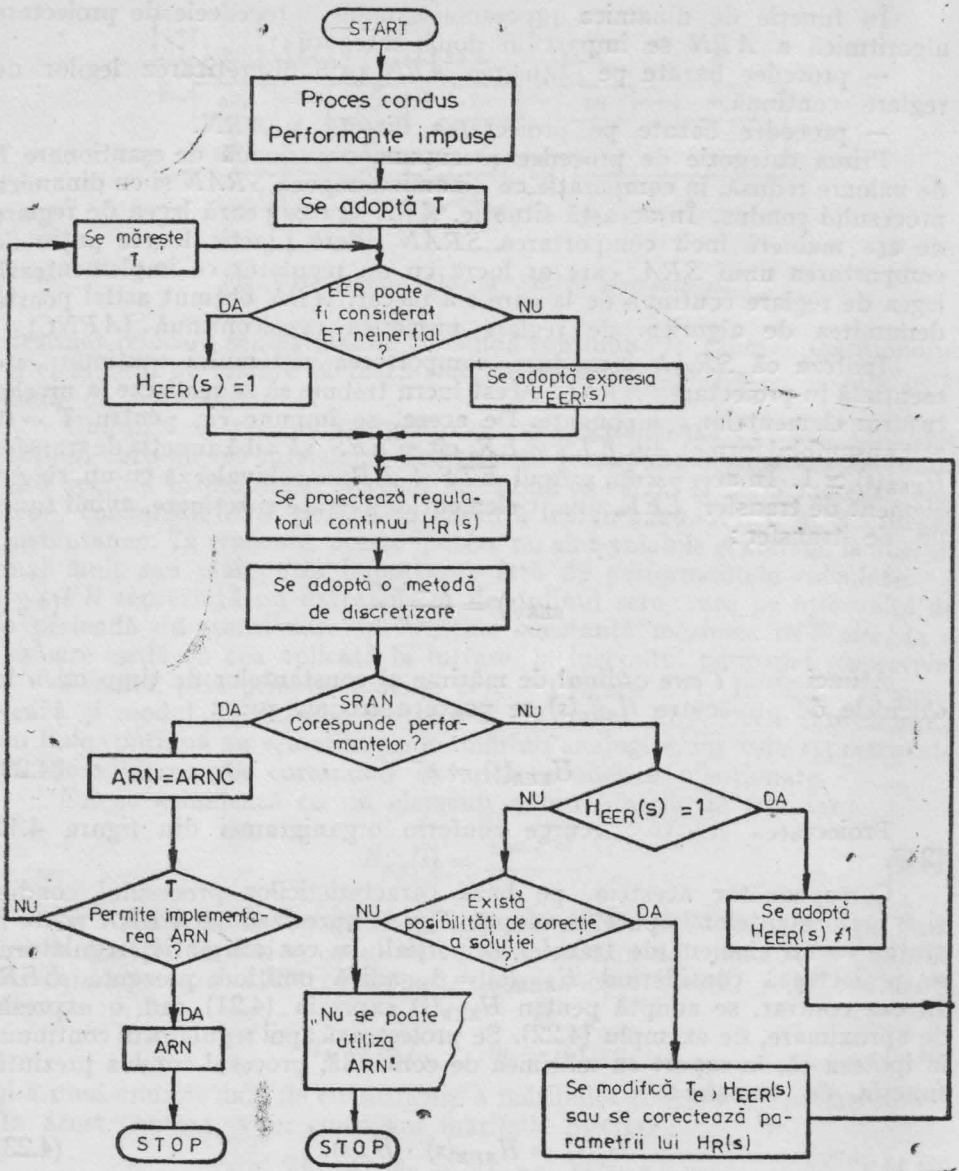


Fig. 4.12. Procedura de proiectare a ARNC.

SRAN, astfel proiectat, este supus, mai departe, validării. În funcție de rezultatul ultimei operații, se consideră ca soluție posibilă ARNC obținut sau se reia proiectarea. În final, soluția posibilă este definitiv validată, numai dacă ARN poate fi implementat pe ECN, în intervalul de timp T .

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

$$T \leq 0,1 \sum_j T_j, \quad (4.24)$$

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

4.2.2. DISCRETIZAREA MODELELOR MATEMATICE ALE ELEMENTELOR DE TRANSFER CONTINUE

Dintre procedeele de discretizare prezentate în [24, 14], 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:

$$\int_{t-T}^t x(\tau) d\tau = \frac{T}{2} [x(t) + x(t - T)]. \quad (4.25)$$

Se particulează 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:

$$H(s) = \frac{\beta_0 + \beta_1 s}{\alpha_0 + \alpha_1 s}, \quad (4.26)$$

căreia îi corespunde ecuația diferențială:

$$\alpha_0 y'(t) + \alpha_1 y(t) = \beta_0 a(t) + \beta_1 \dot{a}(t). \quad (4.27)$$

Dacă se integrează ecuația (4.27), utilizînd metoda trapezului de discretizare se obține:

$$\begin{aligned} \alpha_0 \frac{T}{2} [y(t_K) + y(t_{K-1})] + \alpha_1 [y(t_K) - y(t_{K-1})] = \\ = \beta_0 \frac{T}{2} [a(t_K) + a(t_{K-1})] + \beta_1 [a(t_K) - a(t_{K-1})]. \end{aligned} \quad (4.28)$$

În acest fel, pentru funcția de transfer (4.26), ARN corespunzător obținut prin discretizare, este:

$$y_K = d_0 a_K + d_1 a_{K-1} + c_1 y_{K-1} = d_0 (w_K - r_K) + d_1 (w_{K-1} - r_{K-1}) + c_1 y_{K-1}, \quad (4.29)$$

cu

$$d_0 = \frac{\beta_0 \frac{T}{2} + \beta_1}{\alpha_0 \frac{T}{2} + \alpha_1}; \quad d_1 = \frac{\beta_0 \frac{T}{2} - \beta_1}{\alpha_0 \frac{T}{2} + \alpha_1}; \quad c_1 = \frac{\alpha_1 - \alpha_0 \frac{T}{2}}{\alpha_0 \frac{T}{2} + \alpha_1}, \quad (4.30)$$

unde :

a este mărimea de acționare (eroarea; abaterea),

y — mărimea de comandă.

4.2.3. PROIECTAREA ALGORITMICĂ A SRA ÎN CASCADĂ [4]

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 figura 4.6 se încadrează în acest tip de structură [24]. 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 bucele de reglare.

3. Se proiectează succesiv bucele 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.

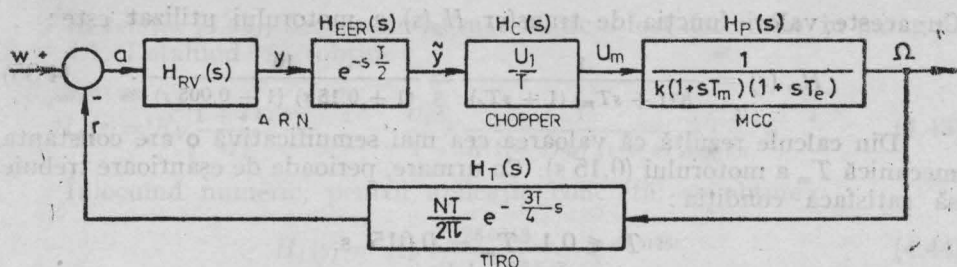


Fig. 4.13. Schema bloc a SRA-V cu f.d.t. ale elementelor componente reprezentate.

4.2.4. PROIECTAREA CONCRETĂ A SRA-V [4, 73, 26]

SRA-V are schema bloc din figura 4.13, în care blocurile au funcțiile de transfer stabilite în paragraful 4.1.

Pentru stabilirea valorii perioadei de eșantionare T , se pornește de la determinarea constantelor T_m și T_e din $H_p(s)$ (modelul matematic al motorului de curent continuu). MCC folosit este de tip EP 211 (IME Pitești), avînd excitația cu magnet permanent și următoarele date de catalog:

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

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

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

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

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

Momentul de inerție al motorului este:

$$J_m = \frac{mD^2}{8} = \frac{\rho \frac{D^2}{4} \cdot l \cdot D^2}{8}, \quad (4.31)$$

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

Constanta motorului este:

$$K = \frac{U_n - R_i I_n}{2\pi n_n} \cdot 60 \approx 0,1 \frac{\text{V}}{\text{rad/s}}. \quad (4.32)$$

Constantele de timp au următoarele valori:

$$T_e = \frac{L_i}{R_i} \approx 5 \text{ ms}. \quad (4.33)$$

$$T_m = \frac{J R_i}{K^2} \approx 150 \text{ ms}. \quad (4.34)$$

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

$$H_p(s) = \frac{1}{K(1 + sT_m)(1 + sT_e)} = \frac{10}{(1 + 0,15s)(1 + 0,005s)}. \quad (4.35)$$

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

$$T \ll 0,1 \quad T_m = 0,015 \text{ s.}$$

Se adoptă $T = 0,01 \text{ s} = 10 \text{ ms.}$ (4.36)

Timpul de măsură T_M se adoptă $\frac{T}{2} = 5 \text{ ms.}$ (4.37)

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 [24]:

a) SRA este asimptotic stabil, dacă rezerva de fază $\varphi_M \in (50^\circ \text{ } 70^\circ)$. 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:

$$t_{r, 0,05} \simeq 4 \quad T_t = 4/\omega_t, \quad (4.38)$$

$$t_{r, 0,02} \simeq 5 \quad T_t = 5/\omega_t. \quad (4.29)$$

c) Dacă sistemul deschis are un caracter integrator, atunci în regim staționar SRA satisface, în general, condiția de anulare asimptotică a ieșirilor de apreciere [14, 4].

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:

$$H_{RV}(s) = K_V \left(1 + \frac{1}{sT_i} \right). \quad (4.40)$$

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

$$T_i = T_m = 0,15 \text{ s.} \quad (4.41)$$

Funcția de transfer a sistemului deschis (fig. 4.13) este:

$$H_L(s) = H_{RV}(s) \cdot H_{EER}(s) \cdot H_C(s) \cdot H_P(s) \cdot H_T(s). \quad (4.42)$$

În relația (4.42), fiecare model matematic a fost determinat în paragraful 4.1. Detaliind se obține:

$$H_L(s) = K_V \frac{1 + sT_i}{sT_i} \cdot e^{-\frac{T}{2}s} \cdot \frac{U_1}{T} \cdot \frac{1}{K(1 + sT_m)(1 + sT_e)} \cdot \frac{N}{2\pi} e^{-\frac{3T}{4}s} \quad (4.43)$$

Înlocuind numeric, pentru aplicația concretă, se obține:

$$H_L(s) = K_V \frac{254648}{s(1 + 0,005s)} \cdot e^{-0,0125s} \quad (4.44)$$

Coefficientul de reglare al regulatorului, K_V , se determină grafo-analitic. Aceasta constă în reprezentarea caracteristicilor de frecvență ale sistemului deschis, $|H_L|$ și φ_L , pentru o valoare arbitrară K_V , a coeficientului de amplificare și determinarea lui K_V cu relația:

$$K_V = K_{V_0} \cdot 10^{-0,05|H_L|^* \text{ dB}}, \quad (4.45)$$

în care $|H_L|^* \text{ dB}$ este ordonata caracteristicii amplitudine-pulsație, la $\omega = \omega_i^*$ (figura 4.14), corespunzătoare unei rezerve de fază de 60° .

Reprezentarea caracteristicilor de frecvență s-a făcut alegînd:

$$K_V = 1/254648 = 3,93 \cdot 10^{-6}, \quad (4.46)$$

obținîndu-se pentru K_V valoarea:

$$\begin{aligned} K_V &= K_{V_0} \cdot 10^{-0,05|H_L|^*} = \omega_i^* \cdot K_{V_0} = 30,0 \cdot 3,93 \cdot 10^{-6} = \\ &= 1,18 \cdot 10^{-4} \text{ s}. \end{aligned} \quad (4.47)$$

ARNC se determină prin discretizarea, cu ajutorul metodei trapezului, a legii de reglare de tip *PI* (4.40), identificată cu ecuația (4.26), în care:

$$\beta_0 = K_V = 1,18 \cdot 10^{-4} \text{ s};$$

$$\beta_1 = K_V T_i = 1,77 \cdot 10^{-5} \text{ s};$$

$$\alpha_0 = 0; \alpha_1 =$$

$$= T_i = 0,15 \text{ s}.$$

Folosind formulele de calcul (4.30), se obțin pentru coeficienți, valorile:

$$d_0 = 1,22 \cdot 10^{-4} \text{ s};$$

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

$$c_1 = 1. \quad (4.48)$$

ARNC este deci de forma:

$$y_K = d_0 a_K + d_1 a_K + y_{K-1}. \quad (4.49)$$

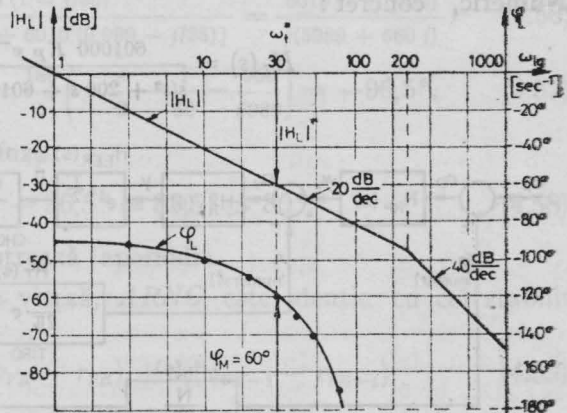


Fig. 4.14. Caracteristica de fază și frecvență ale sistemului deschis, pentru o valoare arbitrară a lui K_{V_0} .

4.2.5. PROIECTAREA SRA—VP

SRA—VP are schema bloc din figura 4.15, în care blocurile au funcțiile de transfer stabilite în paragraful 4.1.3.

După cum am arătat în paragraful 4.2.3, proiectarea *SRA* în cascadă decurge similar cu proiectarea *SRA* convențională.

Ca urmare, se pot utiliza coeficienții calculați în paragraful 4.2.4 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:

$$\varphi_{MV} = 64,3^\circ. \quad (4.50)$$

Acest lucru este de dorit, pentru că prin această creștere a marginii de fază, crește și amortizarea *SRA* [26].

Bucloa internă de viteză are funcția de transfer:

$$H_V(s) = \frac{H_{RV}(s) \cdot H_{EER}(s) \cdot H_C(s) \cdot H_P(s)}{1 + H_{RV}(s) \cdot H_{EER}(s) \cdot H_C(s) \cdot H_P(s) \cdot H_T(s)}. \quad (4.51)$$

Înlocuind valorile numerice calculate în (4.24) rezultă:

$$H_V(s) = \frac{3776 e^{-0,005s}}{s^2 + 200s + 6010}. \quad (4.52)$$

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:

$$H_L(s) = H_{RP}(s) \cdot H_V(s) \cdot \frac{H_N(s)}{s}. \quad (4.53)$$

Numeric, concret:

$$H_L(s) = \frac{601000 K_P e^{-0,005s}}{s(s^2 + 200s + 6010 e^{-0,01s})}. \quad (4.54)$$

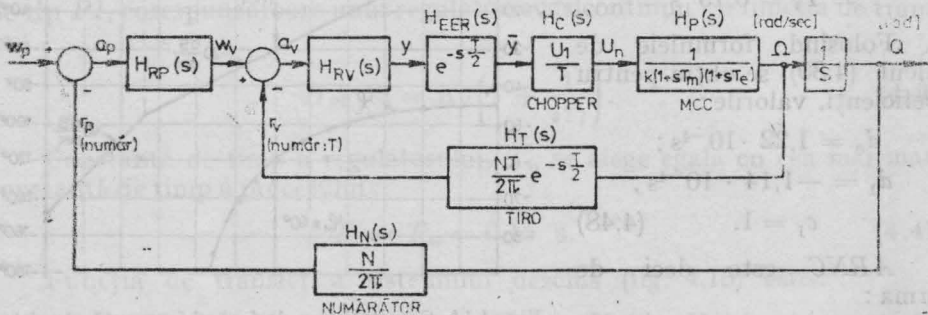


Fig. 4.15. Structura bloc a *SRA—VP* cu f.d.t. ale elementelor detaliate.

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 neglijați.

Pulsațiile de frîngere ale funcției de transfer (4.54) sînt:

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

$$\omega_1 = 36,8 \text{ s}^{-1},$$

$$\Rightarrow \omega_2 = 163,2 \text{ s}^{-1}.$$

Caracteristicile de frecvență ale sistemului deschis

sînt reprezentate sînt în figura 4.16; s-a ales pentru K_P valoarea inițială $K_{P_0} = 0,01$.

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 $\varphi_M > 80^\circ$ [24,14].

Din figura 4.16 se observă că pentru $\omega_i^* = 4 \text{ rad/s}$, $\varphi_M = 82,4^\circ$. La această pulsație rezultă:

$$K_P = K_{P_0} \cdot 10^{-0,05 \cdot |H_L|^*} = \omega_i^* \cdot K_{P_0} = 0,04. \quad (4.55)$$

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

$$H_L(4) = \frac{24040 (1 - j/50)}{4j[-16 + 800 + 6010(0,999 - j/25)]} = \frac{6010 (1 - j/50)}{j(5989 + 560j)}, \quad (4.56)$$

$$\varphi_L(4) = \arg H_L(4) = \frac{180}{\pi} \left(-\frac{\pi}{2} - \frac{1}{50} - \frac{560}{5989} \right) = -96,5^\circ. \quad (4.57)$$

Rezultă o margine de fază:

$$\varphi_M = 180^\circ - 96,5^\circ = 83,5^\circ > 80^\circ. \quad (4.58)$$

Deci, răspunsul SRA se păstrează aperiodic.

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

$$y_K = y_{K-1} + d_0(w_{VK} - r_{VK}) + d_1(w_{VK-1} - r_{VK-1}). \quad (4.59)$$

Pentru regulatorul de poziție se poate scrie:

$$w_{VK} = K_P(w_{PK} - r_{PK}). \quad (4.60)$$

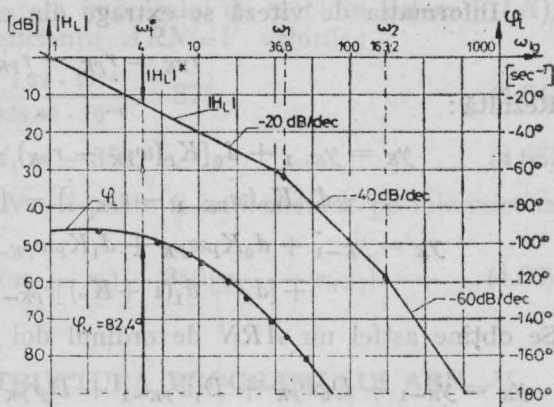


Fig. 4.16. Caracteristicile de frecvență ale sistemului deschis pentru SRA-P.

Informația de viteză se extrage din cea de poziție :

$$r_{VK} = r_{PK} - r_{PK-1}. \quad (4.61)$$

Rezultă :

$$\begin{aligned} y_K &= y_{K-1} + d_0[K_P(w_{PK} - r_{PK}) - (r_{PK} - r_{PK-1})] + \\ &\quad + d_1[K_P(w_{PK-1} - r_{PK-1}) - (r_{PK-1} - r_{PK-2})], \\ y_K &= y_{K-1} + d_0K_Pw_{PK} + d_1K_Pw_{PK-1} - d_0(1 + K_P)r_{PK} + \\ &\quad + [d_0 - d_1(1 + K_P)]r_{PK-1} + d_1r_{PK-2}. \end{aligned}$$

Se obține astfel un ARN de ordinul doi de forma :

$$y_K = y_{K-1} + D_0w_{PK} + D_1w_{PK-1} + D_2r_{PK} + D_3r_{PK-1} + D_4r_{PK-2}. \quad (4.62)$$

În acest ARN, coeficienții au valorile :

$$\begin{aligned} D_0 &= d_0K_P = 4,88 \cdot 10^{-6} \text{ s.} \\ D_1 &= d_1K_P = -4,56 \cdot 10^{-6} \text{ s,} \\ D_2 &= -d_0(1 + K_P) = -1,27 \cdot 10^{-4} \text{ s,} \\ D_3 &= d_0 - d_1(1 + K_P) = 2,41 \cdot 10^{-4} \text{ s.} \\ D_4 &= d_1 = -1,14 \cdot 10^{-4} \text{ s.} \end{aligned} \quad (4.63)$$

4.3. IMPLEMENTAREA ALGORITMULUI DE REGLARE NUMERICĂ A VITEZEI [26]

4.3.1. FORMA DISCRETĂ FINALĂ PENTRU ARN—V

ARN—V a rezultat sub forma ecuației discrete (4.49) :

$$y_K = d_0a_K + d_1a_{K-1} + y_{K-1},$$

cu coeficienții de valori calculate în (4.48).

În relația (4.49) 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 (4.49), 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:

$$d_0 = \frac{1,22 \cdot 10^{-4}}{325,52 \cdot 10^{-9}} = 375, \quad (4.64)$$

$$d_1 = -350. \quad (4.65)$$

Forma finală sub care $ARN-V$ poate fi implementat pe microsistem este:

$$y_K = y_{K-1} + 375(w_K - r_K) - 350(w_{K-1} - r_{K-1}). \quad (4.66)$$

4.3.2. SARCINILE ECN. STRUCTURA PROGRAMULUI $ARN-V$

ECN este realizat cu microsistemul SDK-85, construit în jurul microprocesorului 8085, prezentat în paragraful 2.3. Pentru realizarea $SRAN-V$ microsistemul trebuie să îndeplinească următoarele funcții:

(F1) — preluarea mărimii de prescriere, w , de la operator, prin consola *DAF*, utilizând nivelul de întrerupere *RST 6.5*;

(F2) — măsurarea vitezei curente r_K , de la *TIRO*, utilizând o subrutină de frecvențmetru;

(F3) — calculul mărimii de comandă y_K , conform ecuației (4.66) pentru $ARN-V$;

(F4) — conversia mărimii numerice y_K într-un interval de timp real T_{CK} , timp de conducție pentru chopper, utilizând 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 figura 4.17.

În figura 4.17 sînt reprezentate intervalele de timp în care se efectuează de către microsistem operațiile corespunzătoare celor cinci funcții care trebuie să le îndeplinească. Funcția F4, 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 de prescriere, w , poate fi preluată din exterior în orice moment prin întreruperea, pe nivelul *RST 6.5*, a ori-

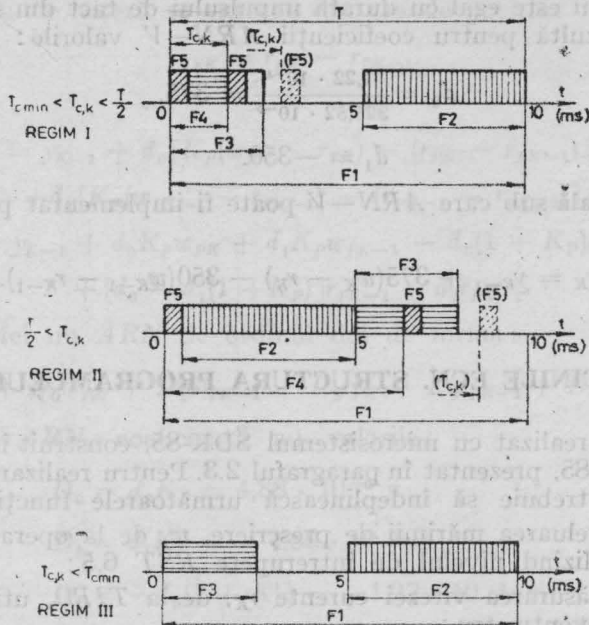


Fig. 4.17. Momentele de execuție a celor cinci funcții de către sistemul de comandă pe parcursul perioadei de eșanționare T .

cărei operații aflate în curs de desfășurare cu excepția (F2). În regimul III, chopperul nu mai este comandat, adică motorul rămâne nealimentat.

Schema logică-bloc a programului este prezentată în figura 4.18.

Programele detaliate, atât pentru varianta cu chopper cu tiristoare, cât și pentru varianta cu chopper cu tranzistoare sînt date în anexa A1 [67].

4.3.3. PRINCIPALELE SUBPROGRAME CONSTITUENTE ALE ARN—V

4.3.3.1. SUBRUTINA DE TRATARE A INTRERUPERILOR GENERATE DE TIMER

Timerul utilizat, este parte constitutivă a circuitului 8155 și este folosit în modul de lucru 2, [63, 56]. 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 semnal de comandă.

În momentul activării întrării RST 7.5 se execută subrutina reprezentată în organigrama din figura 4.19.

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

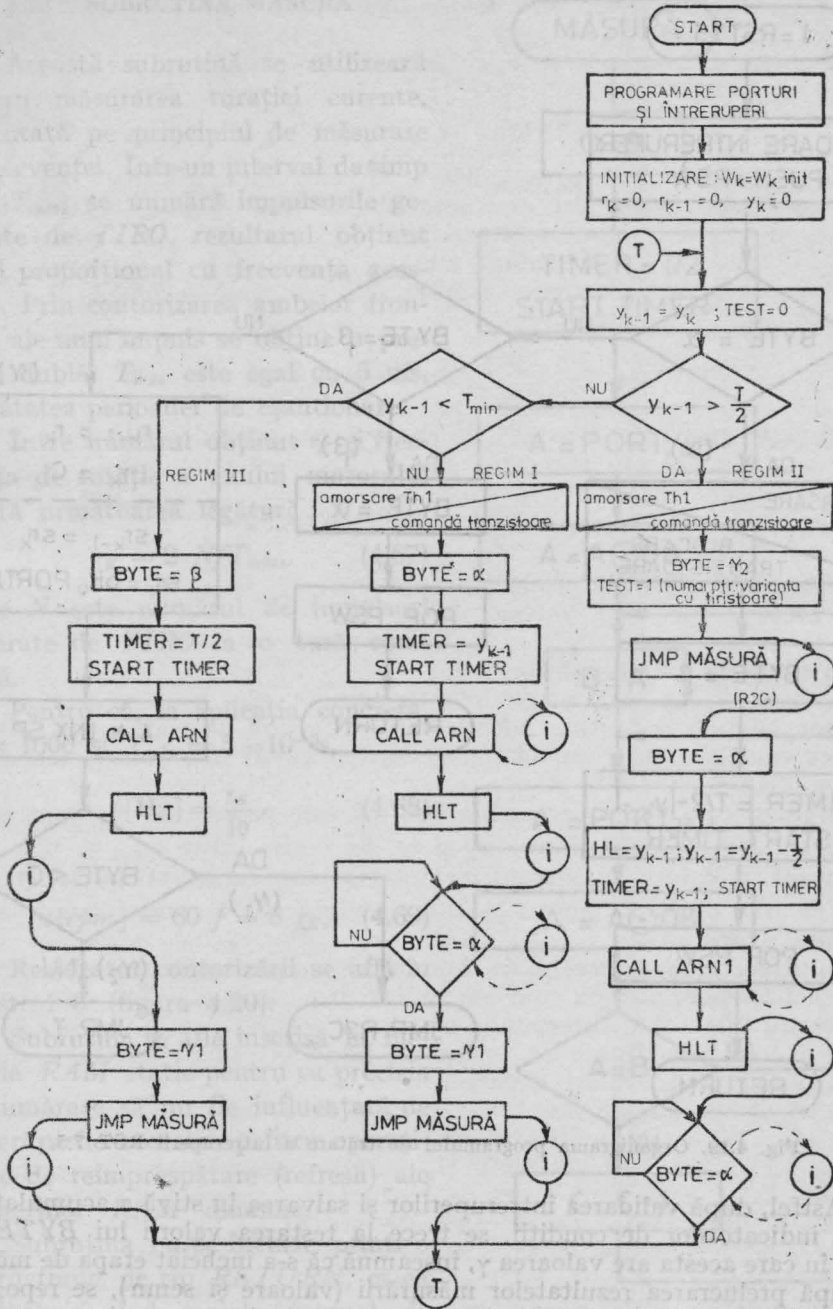


Fig. 4.18. Organigrama programului pentru ARN-V.

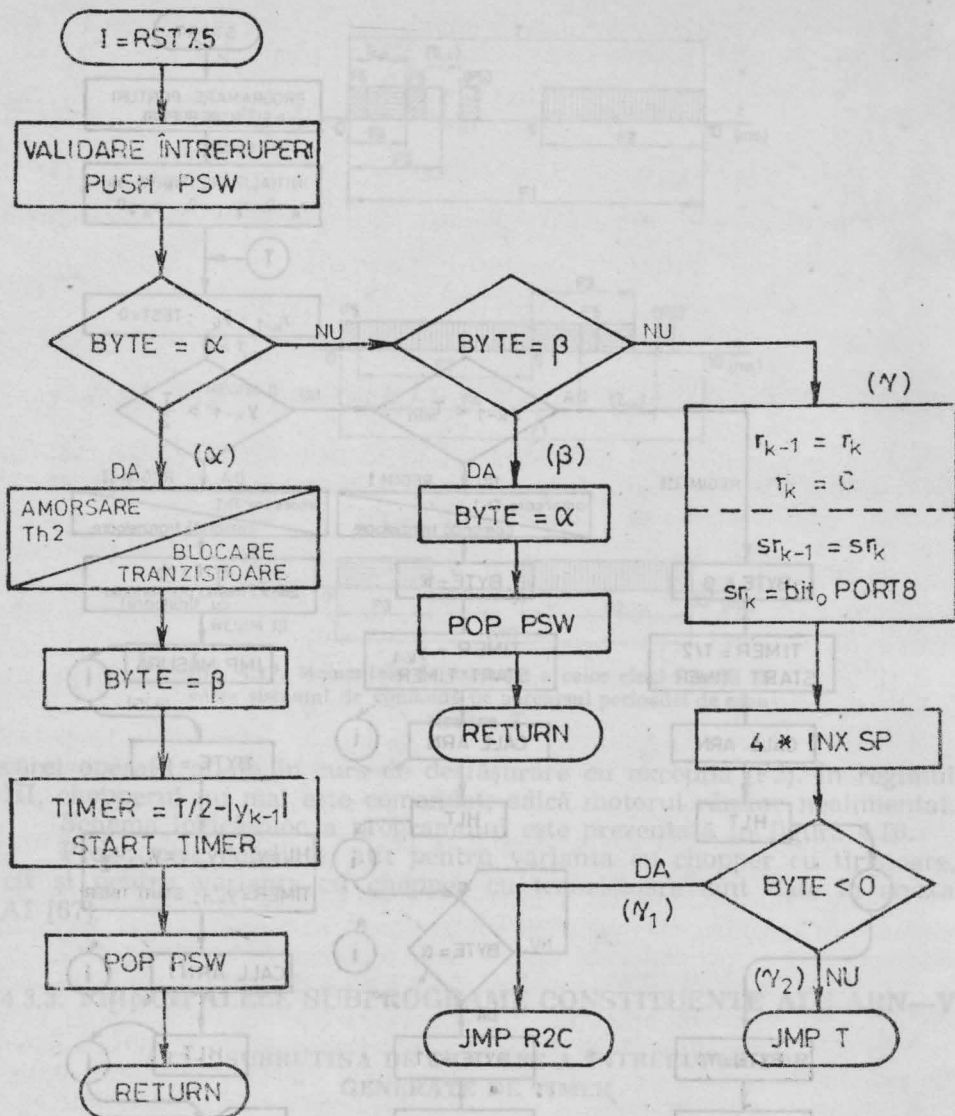


Fig. 4.19. Organigrama programului de tratare a întreruperii RST 7.5.

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 γ , 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 tip *JMP*.

4.3.3.2. SUBRUTINA MĂSURĂ

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.

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

$$r_K = 2 N f T_{m\grave{a}s}, \quad (4.67)$$

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

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

$$f[\text{Hz}] = \frac{r_K}{10}, \quad (4.68)$$

iar

$$n[rpm] = 60 f = 6 r_K, \quad (4.69)$$

Rezultatul contorizării se află în registrul C (figura 4.20).

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 cînd sosește semnal de întrerupere generat de *TIMER*.

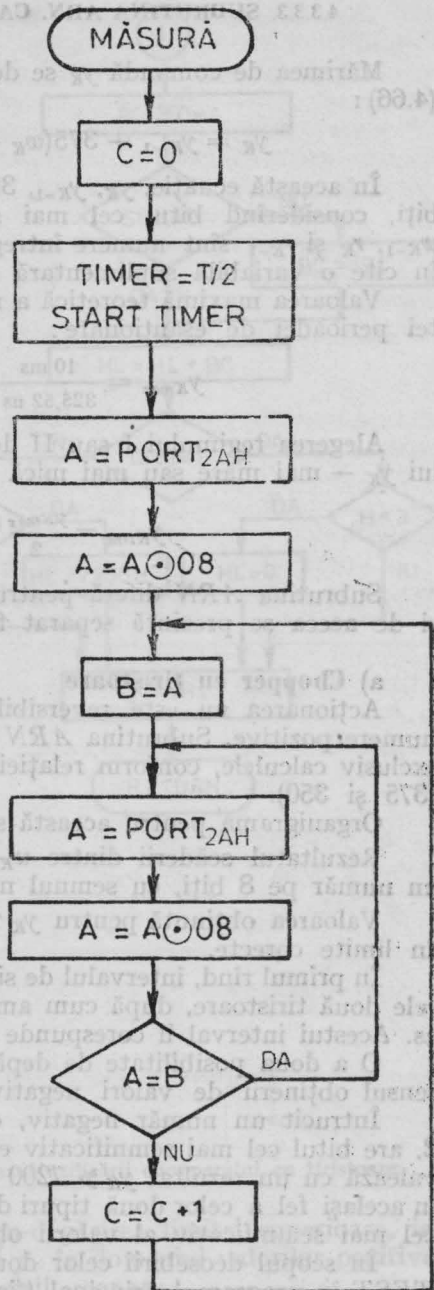


Fig. 4.20. Organigrama subrutinei măsură.

4.3.3.3. SUBRUTINA ARN. CALCULUL MĂRIMII DE COMANDA

Mărimea de comandă y_K se determină, după cum am arătat, cu relația (4.66):

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

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

$$y_{Kmax} = \frac{10 \text{ ms}}{325,52 \text{ ns}} = 30720 = 7800 \text{ H.} \quad (4.70)$$

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:

$$y_{Klim} = \frac{y_{Kmax}}{2} = 15360 = 3000 \text{ H.} \quad (4.71)$$

Subrutina ARN diferă pentru cele două tipuri de chopper 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 (4.66), cu valori fixe pentru coeficienți (375 și 350).

Organigrama pentru această subrutină este prezentată în figura 4.21.

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 3.3.2, este de 500 μ 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.

Întrucît un număr negativ, exprimat în codul complement față de 2, are bitul cel mai semnificativ egal cu 1, iar o depășire superioară echivalează cu un rezultat $y_K \gg 7200$ 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. 4.18). Prin această variabilă se indică dacă se lucrează în regimul II ($y_K > \frac{T}{2} = 3000$ H) sau nu. În cazul în care se

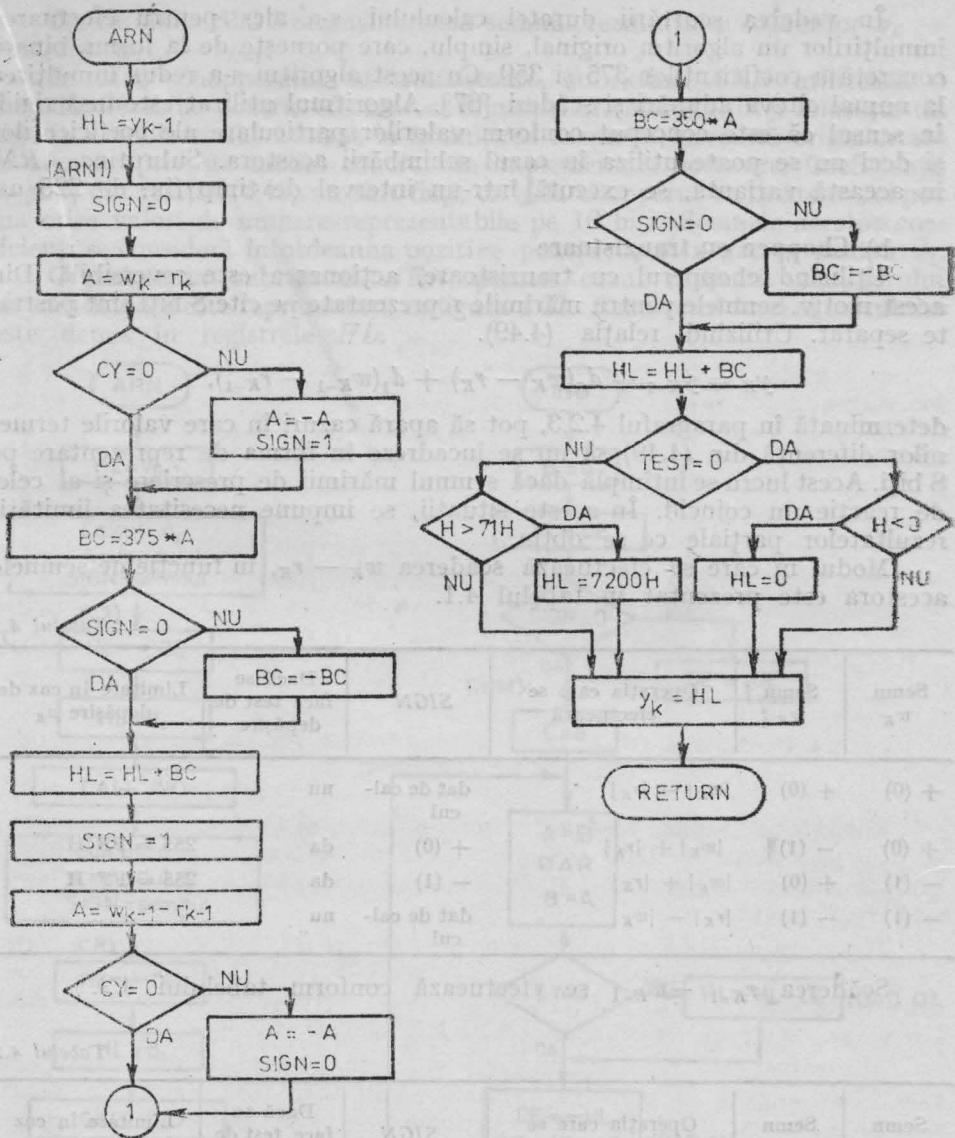


Fig. 4.21. Organigrama subrutinei ARN în cazul utilizării chopperului cu tiristoare.

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 (4.66), acesta 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 [67]. 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 μ s.

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 (4.49).

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

determinată în paragraful 4.2.3, pot să apară cazuri în care valorile termenilor diferență din (4.49) 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 4.1.

Tabelul 4.1

Semn w_K	Semn r_K	Operația care se efectuează	SIGN	Dacă se face test de depășire	Limitare în caz de depășire a_K
+ (0)	+ (0)	$ w_K - r_K $	dat de calcul	nu	—
+ (0)	— (1)	$ w_K + r_K $	+ (0)	da	255 = FF H
— (1)	+ (0)	$ 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 4.2.

Tabelul 4.2

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	—

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. Metoda, 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 (4.49) nu sînt fișși, 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 .

Organigrama subrutinei ARN pentru cazul utilizării chopperului cu tranzistoare este reprezentată în figura 4.22. Rezultatul final al calculului este depus în registrele HL.

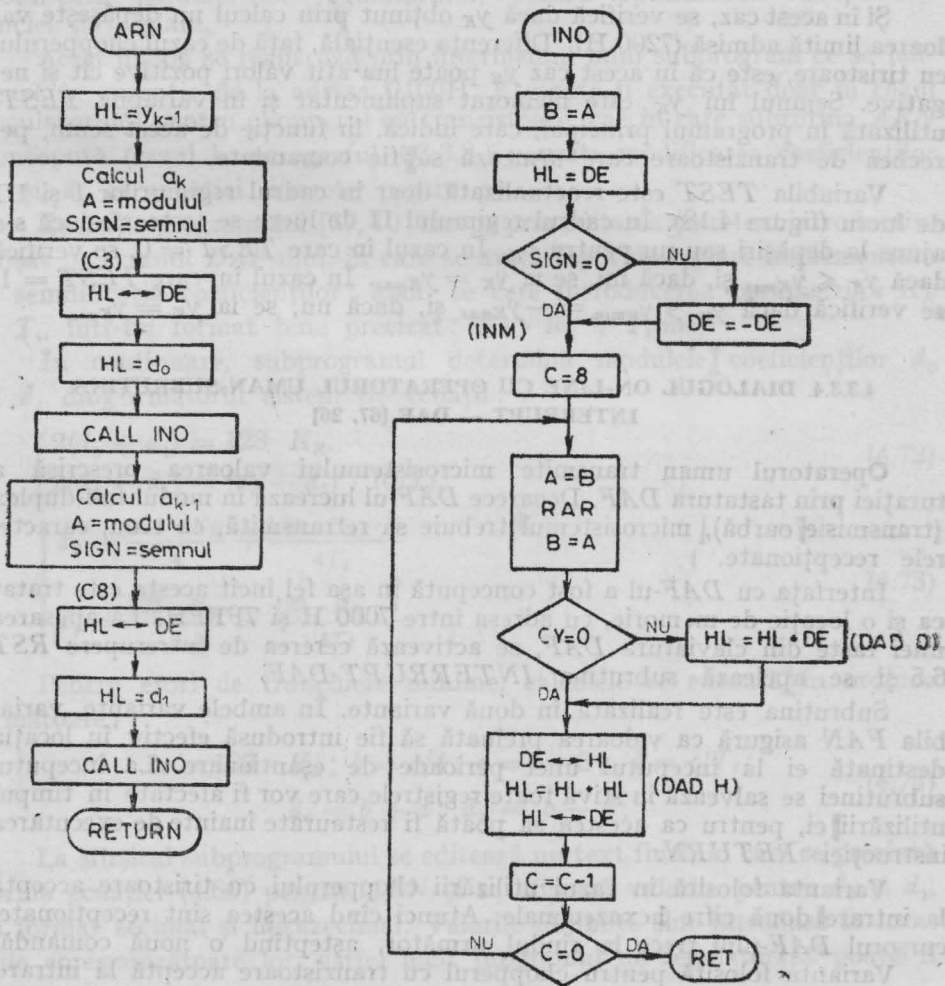


Fig. 4.22. Organigrama subrutinei ARN în cazul chopperului cu tranzistoare.

Operația $HL = HL + B \cdot DE$ este executată într-o altă subrutină ajutătoare *INM* (figura 4.22). *INM* operează după un algoritm similar cu cel folosit la înmulțirea manuală. După cum se știe, [19], 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 rφ*, 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 μs.

Și 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 (figura 4.18). 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 $y_K = y_{Kmin}$.

4.3.3.4. DIALOGUL ON-LINE CU OPERATORUL UMAN-SUBROUTINA INTERRUPT — DAF [67, 26]

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, caracterile recepționate.

Interfața cu *DAF*-ul a fost concepută în așa fel încît 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 cînd 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 sexazecimal, trimite către DAF: „ = rezultat conversie H” și trece la rîndul următor, așteptînd o nouă comandă. Preluarea mărimii prescrie 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.

4.3.3.5. DIALOGUL OFF-LINE CU OPERATORUL UMAN

Dialogul off-line cu operatorul uman asigură acestuia posibilitatea 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 C100H. El poate fi executat doar în cazul regulatorului pentru chopperul cu tranzistoare, caz în care subrutina ARN concepută (cazul b, paragraful 4.3.3.3.) permite modificarea coeficienților d_0 și d_1 ai ecuației discrete pentru y_K .

Execuția programului [67, 10] 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:

$$\begin{cases} 2(d_0 + d_1) = 123 K_R \\ 2(d_0 - d_1) = (d_0 + d_1) \cdot 10/T_i \end{cases} \quad (4.72)$$

$$\begin{cases} d_0 = \frac{123 \cdot K_R}{4} + \frac{123 \cdot K_R \cdot 5}{4T_i} \\ d_1 = \frac{123 \cdot K_R}{4} - \frac{123 \cdot K_R \cdot 5}{4T_i} \end{cases} \quad (4.73)$$

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

$$\begin{aligned} a &= 123 \cdot K_R; \quad b = a/4; \quad c = b \cdot 5; \quad d = c/T_i; \\ d_0 &= b + d; \quad d_1 = b - d; \end{aligned} \quad (4.74)$$

La sfîrșitul subprogramului se editează un text final, în care se prezintă forma ecuației (4.66) 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.

4.3.4. 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 [14].

După cum s-a arătat în paragraful 4.2.4, 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 în 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 figura 4.23 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.

4.3.5. 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 [24].

Configurația utilizată este prezentată în figura 4.24. În locul traductorului incremental (*TIRO*) se folosește un generator de semnale dreptunghiulare, cu frecvența reglabilă și de nivel *TTL* (în cazul concret aparatul *VER-SATESTER*, 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), măbind mărimea de comandă y_K . Se obține o accelerare a motorului.

Cu aceeași valoare a mărimii de prescriere se mărește frecvența semnalului furnizat de generator și rezultă o decelerare a motorului.

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.

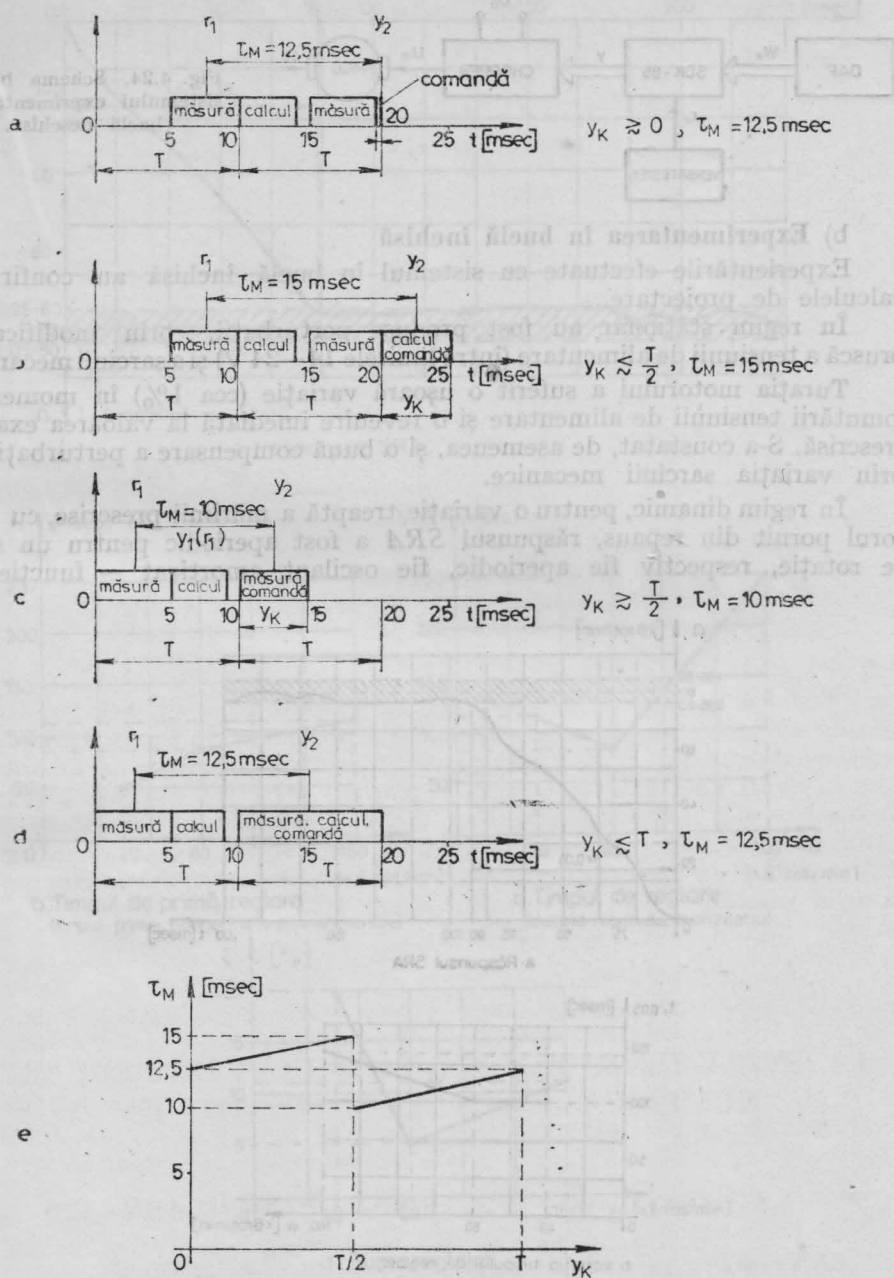


Fig. 4.23. Structura timpului mort pentru diferite valori ale mărimii de comandă. Graficul dependenței $\tau_M = f(y_K)$.

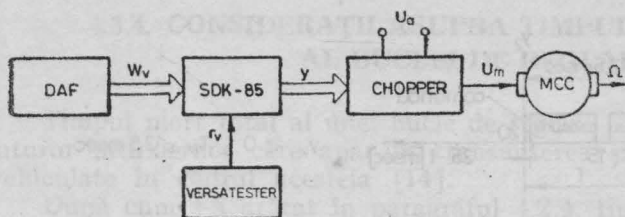


Fig. 4.24. Schema bloc a sistemului experimental în buclă deschisă.

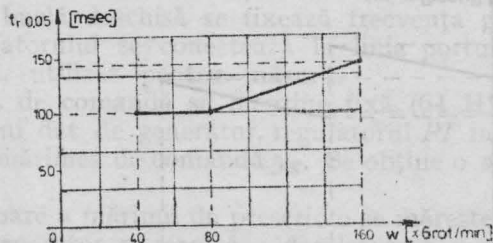
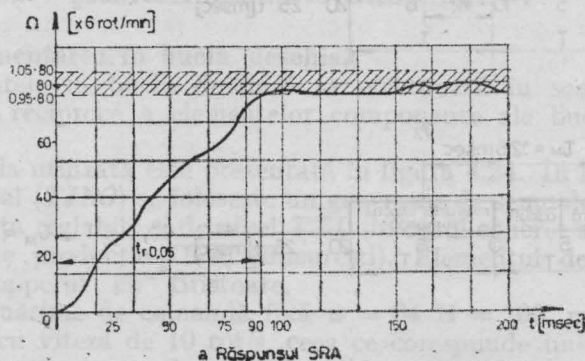
b) Experimentarea în buclă închisă

Experiențele 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.

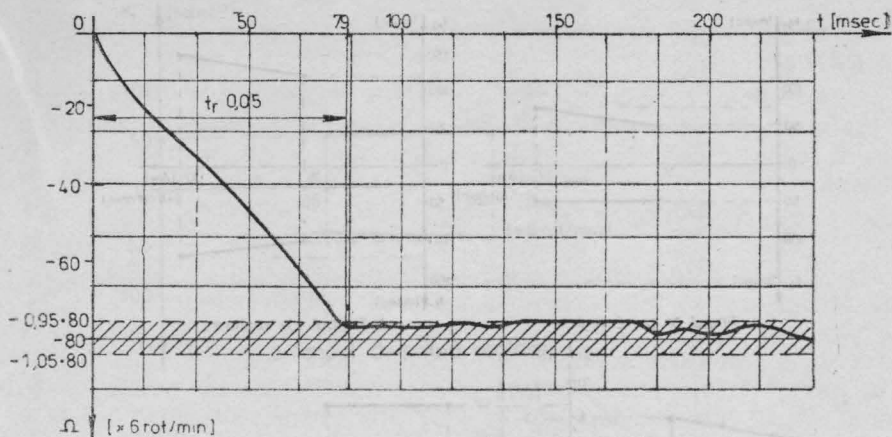
Turația motorului a suferit o ușoară variație (cca 1%) în momentul comutării tensiunii de alimentare și o revenire imediată la valoarea exactă prescrisă. S-a constatat, de asemenea, și o bună compensare a perturbațiilor prin variația sarcinii mecanice.

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

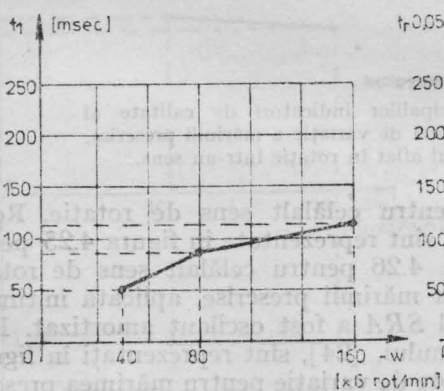


b Variația timpului de reglare

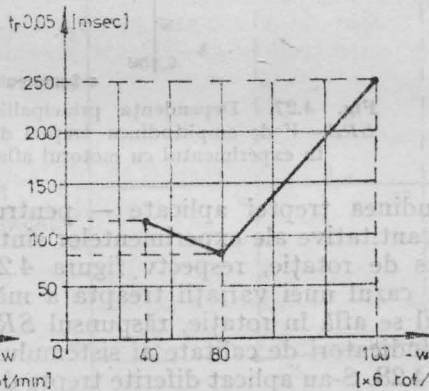
Fig. 4.25. Răspunsul SRA la o variație treaptă a mărimii prescrise și dependența timpului de reglare de amplitudinea treptei de variație a mărimii prescrise în experimentul cu motorul pornit din repaus într-un sens de rotație.



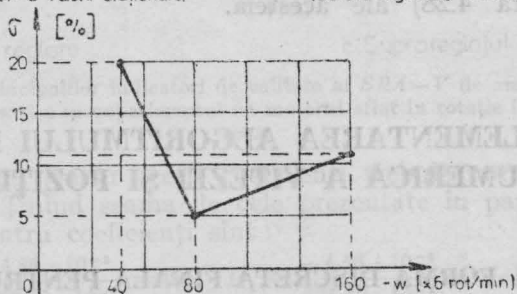
a Răspunsul SRA



b. Timpul de primă reglare
(timpul primei atingeri a valorii staționare)



c. Timpul de reglare
(durata regimului tranzitoriu)



d Suprareglajul

Fig. 4.26. Răspunsul SRA la o variație treaptă a mărimii prescrise și dependența principalelor indicatori de calitate ai SRA-V de amplitudinea treptei de variație a mărimii prescrise în experimentul cu motorul pornind din repaus în cel de al doilea sens de rotație.

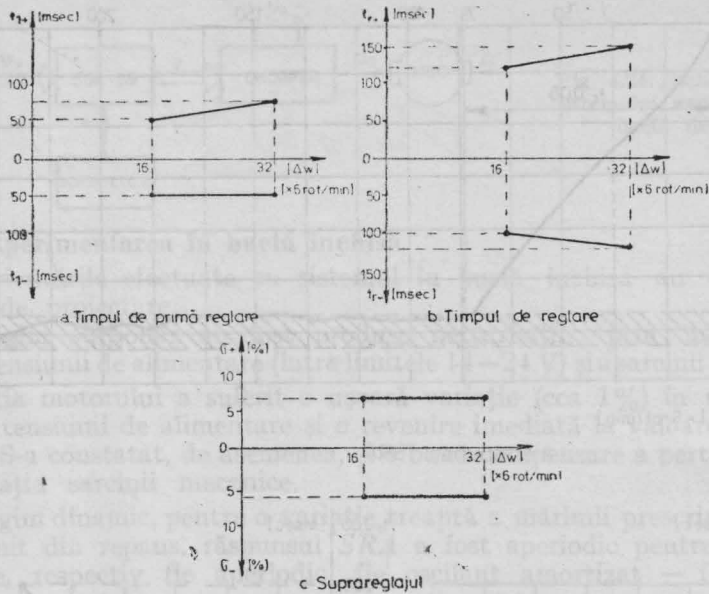


Fig. 4.27. Dependența principalilor indicatori de calitate ai $SRA-V$ de amplitudinea treptei de variație a mărimii prescrise, în experimentul cu motorul aflat în rotație într-un sens.

amplitudinea treptei aplicate — pentru celălalt sens de rotație. Rezultatele cantitative ale experimentelor sînt reprezentate în figura 4.25 pentru un sens de rotație, respectiv figura 4.26 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, [24], sînt reprezentați în figurile 4.27 și 4.28. S-au aplicat diferite trepte de variație pentru mărimea prescrisă în jurul valorilor $w_0 = 480 \text{ rot/min}$ (figura 4.27) și, respectiv, $w_0 = -480 \text{ rot/min}$ (figura 4.28) ale acesteia.

4.4. IMPLEMENTAREA ALGORITMULUI DE REGLARE NUMERICĂ A VITEZEI ȘI POZIȚIEI [25]

4.4.1. FORMA DISCRETĂ FINALĂ PENTRU ARN—VP

$ARN-VP$ a rezultat de forma ecuației discrete (4.62):

$$y_K = y_{K-1} + D_0 w_{PK} + D_1 w_{PK-1} + D_2 r_{PK} + D_3 r_{PK-1} + D_4 r_{PK-2},$$

unde coeficienții D_0 , D_1 , D_2 , D_3 și D_4 au fost calculați în paragraful 4.2.5.

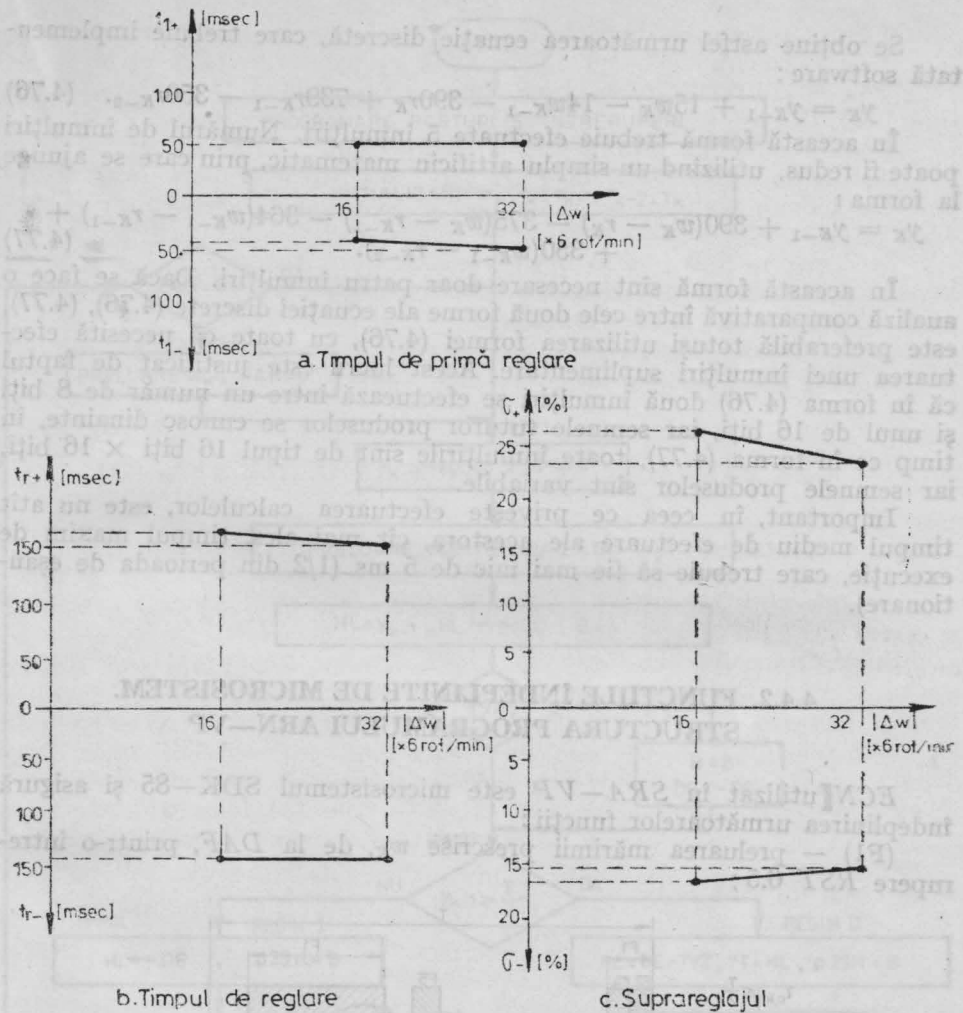


Fig. 4.28. Dependența principalilor indicatori de calitate ai SRA-V de amplitudinea treptei de variație a mărimii prescrise în experimentul cu motorul aflat în rotație în cel de al doilea sens.

Valorile obținute pentru acești coeficienți trebuie normati, la fel ca în cazul ARN-V. Ținând seama de cele prezentate în paragraful 4.3.1, valorile normate pentru coeficienți sînt:

$$\begin{aligned}
 D_0 &= \frac{4,88 \cdot 10^{-6}}{325,52 \cdot 10^{-9}} = 15, & D_1 &= \frac{-4,56 \cdot 10^{-6}}{325,52 \cdot 10^{-9}} = -14, \\
 D_2 &= \frac{-1,27 \cdot 10^{-4}}{325,52 \cdot 10^{-9}} = -390, & D_3 &= \frac{2,41 \cdot 10^{-4}}{325,52 \cdot 10^{-9}} = 739, \\
 D_4 &= \frac{-1,14 \cdot 10^{-4}}{325,52 \cdot 10^{-9}} = -350.
 \end{aligned}
 \tag{4.75}$$

Se obține astfel următoarea ecuație discretă, care trebuie implementată software :

$$y_K = y_{K-1} + 15w_K - 14w_{K-1} - 390r_K + 739r_{K-1} - 350r_{K-2}. \quad (4.76)$$

În această formă trebuie efectuate 5 înmulțiri. Numărul de înmulțiri poate fi redus, utilizând un simplu artificiu matematic, prin care se ajunge la forma :

$$y_K = y_{K-1} + 390(w_K - r_K) - 375(w_K - r_{K-1}) - 364(w_{K-1} - r_{K-1}) + 350(w_{K-1} - r_{K-2}). \quad (4.77)$$

În această formă sînt necesare doar patru înmulțiri. Dacă se face o analiză comparativă între cele două forme ale ecuației discrete (4.76), (4.77), este preferabilă totuși utilizarea formei (4.76), cu toate că necesită efectuarea unei înmulțiri suplimentare. Acest lucru este justificat de faptul că în forma (4.76) 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 (4.77), toate înmulțirile sînt de tipul 16 biți \times 16 biți, iar semnele produselor sînt variabile.

Important, în ceea ce privește efectuarea calculului, este nu atît timpul mediu de efectuare ale acestora, cît mai ales, timpul maxim de execuție, care trebuie să fie mai mic de 5 ms (1/2 din perioada de eșantionare).

4.4.2. FUNCȚIILE ÎNDEPLINITE DE MICROSISTEM. STRUCTURA PROGRAMULUI ARN—VP

ECN utilizat în SRA—VP este microsistemul SDK—85 și asigură îndeplinirea următoarelor funcții :

(F1) — preluarea mărimii prescrise w_K , de la DAF, printr-o întreprupere RST 6.5;

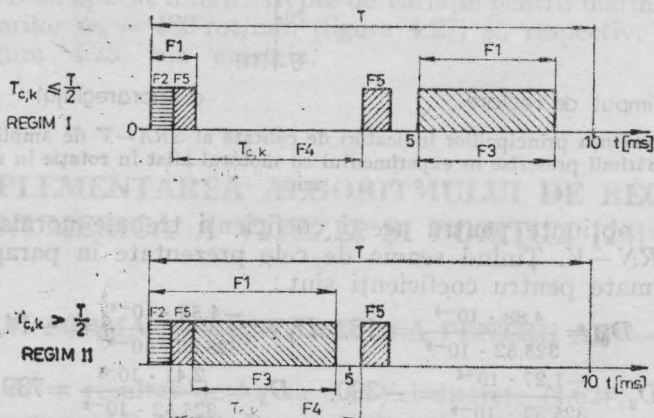


Fig. 4.29. Modul de desfășurare în timp (de-a lungul perioadei de eșantioane) al momentelor de execuție a celor cinci funcții.

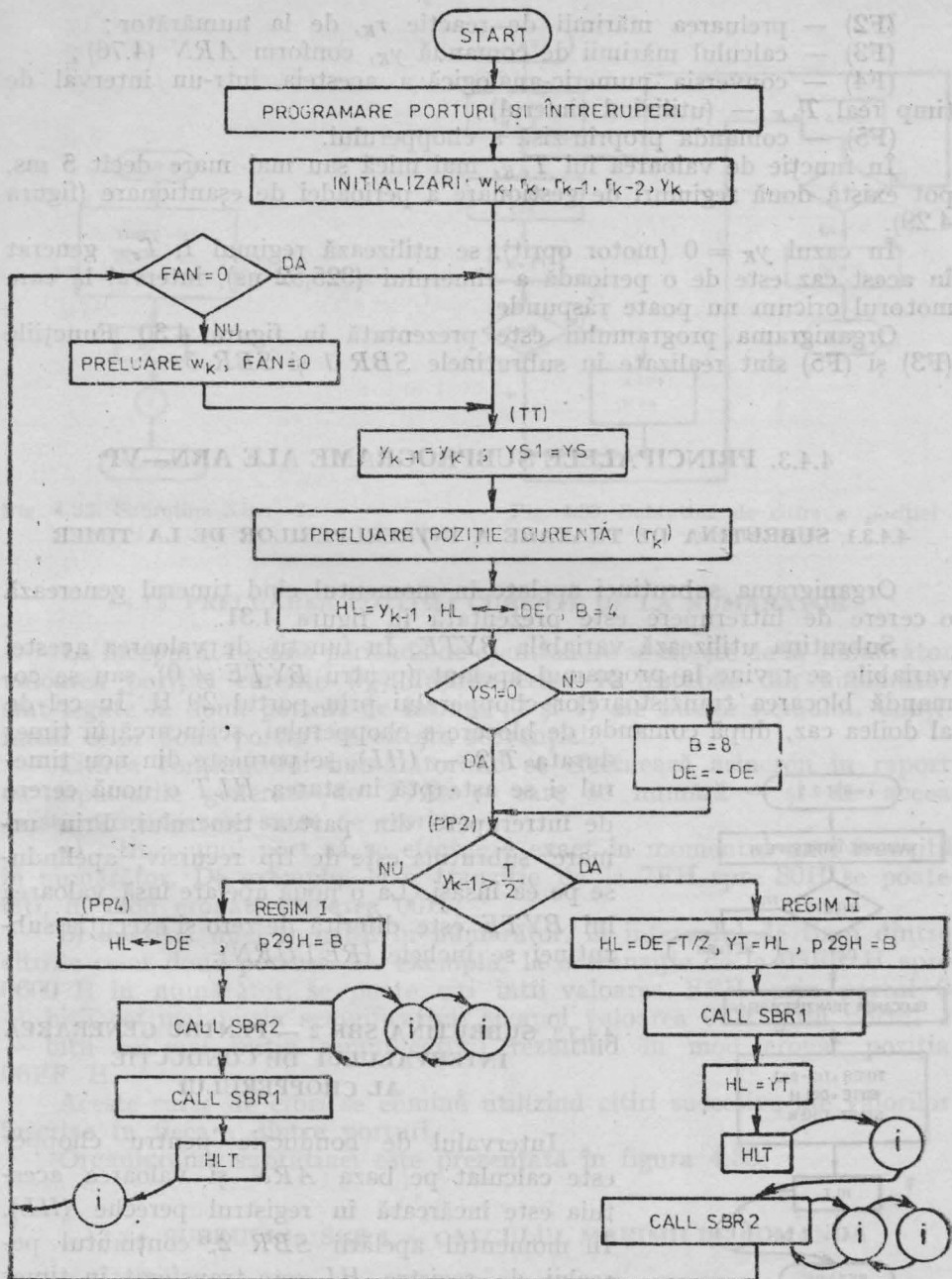


Fig. 4.30. Organigrama ARN-VP.

(F2) — preluarea mărimii de reacție r_K , de la numărător;
 (F3) — calculul mărimii de comandă y_K , conform *ARN* (4.76);
 (F4) — conversia numeric-analogică a acesteia, într-un interval de timp real $T_{c,K}$ — (utilizînd timerul);

(F5) — comanda propriu-zisă a chopperului.

În funcție de valoarea lui $T_{c,K}$, mai mică sau mai mare decît 5 ms, pot exista două regimuri de gestionare a perioadei de eșantionare (figura 4.29).

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.

Organigrama programului este prezentată în figura 4.30. Funcțiile (F3) și (F5) sînt realizate în subrutinele *SBR 1* și *SBR 3*.

4.4.3. PRINCIPALELE SUBPROGRAME ALE *ARN*—*VP*

4.4.3.1. SUBRUTINA DE TRATARE A ÎNTRERUPERILOR DE LA TIMER

Organigrama subrutinei apelate în momentul cînd timerul generează o cerere de întrerupere este prezentată în figura 4.31.

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 ea însăși. La o nouă apelare însă, valoarea lui *BYTE* este diferită de zero și execuția subrutinei se încheie (*RETURN*).

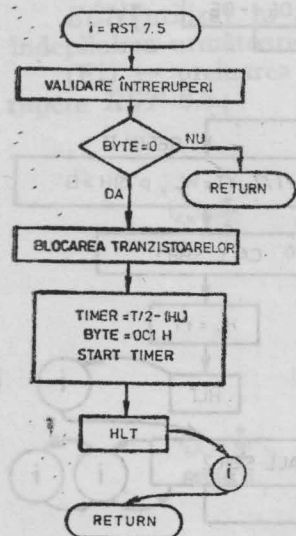


Fig. 4.31. Subrutina de tratare a întreruperilor.

4.4.3.2. SUBRUTINA *SBR 2* — PENTRU GENERAREA INTERVALULUI DE CONDUȘIE 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 (figura 4.32).

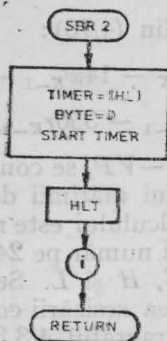


Fig. 4.32. Subrutina *SBR-2*.

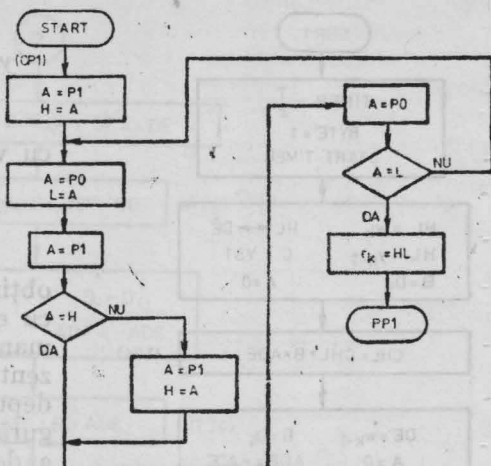


Fig. 4.33. Subrutina de citire a poziției curente.

4.4.3.3. 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:

a) citirea unui port să se efectueze exact în momentul unei tranziții în numărător. De exemplu, la o tranziție de la 7FH spre 80H se poate citi, în mod eronat, valoarea 00H;

b) apariția unei tranziții în numărător, în intervalul de timp dintre citirile celor două porturi. De exemplu, la o tranziție de la 05FF H spre 0600 H în numărător, se poate citi întâi valoarea FFH (prin portul 0 — biții cei mai puțin semnificativi) și apoi valoarea 06H (prin portul 1 — biții cei mai puțin semnificativi), rezultînd în mod eronat poziția 06FF H.

Aceste surse de erori se elimină utilizînd citiri succesive ale valorilor înscrise în fiecare dintre porturi.

Organigrama subrutinei este prezentată în figura 4.33.

4.4.3.4. SUBRUTINA *SBR 1* — CALCULUL MĂRIMII DE COMANDA

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 (figura 4.34, a, b).

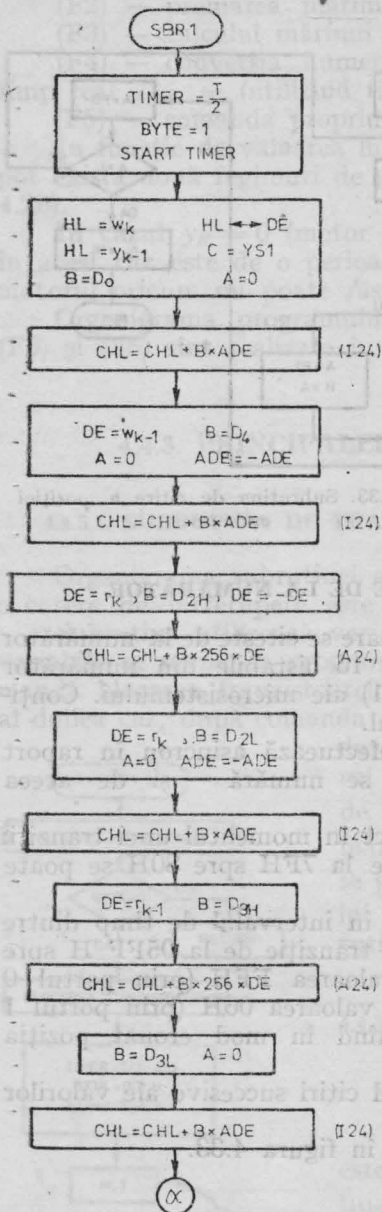


Fig. 4.34. Organigrama subrutinei de calcul a mărimii de comandă.

În I24, înmulțirile se efectuează după un algoritm similar cu cel utilizat în înmulțirea manuală. Însurarea produselor parțiale și rotirile

Ținând cont de (4.62):

$$y_k = y_{k-1} + D_0 w_{PK} + D_1 w_{PK-1} + D_2 r_{PK} + D_3 r_{PK-1} + D_4 r_{PK-2}$$

cu valorile concrete din (4.76):

$$y_k = y_{k-1} + 15w_k - 14w_{k-1} - 390r_k + 739r_{k-1} - 350r_{k-2},$$

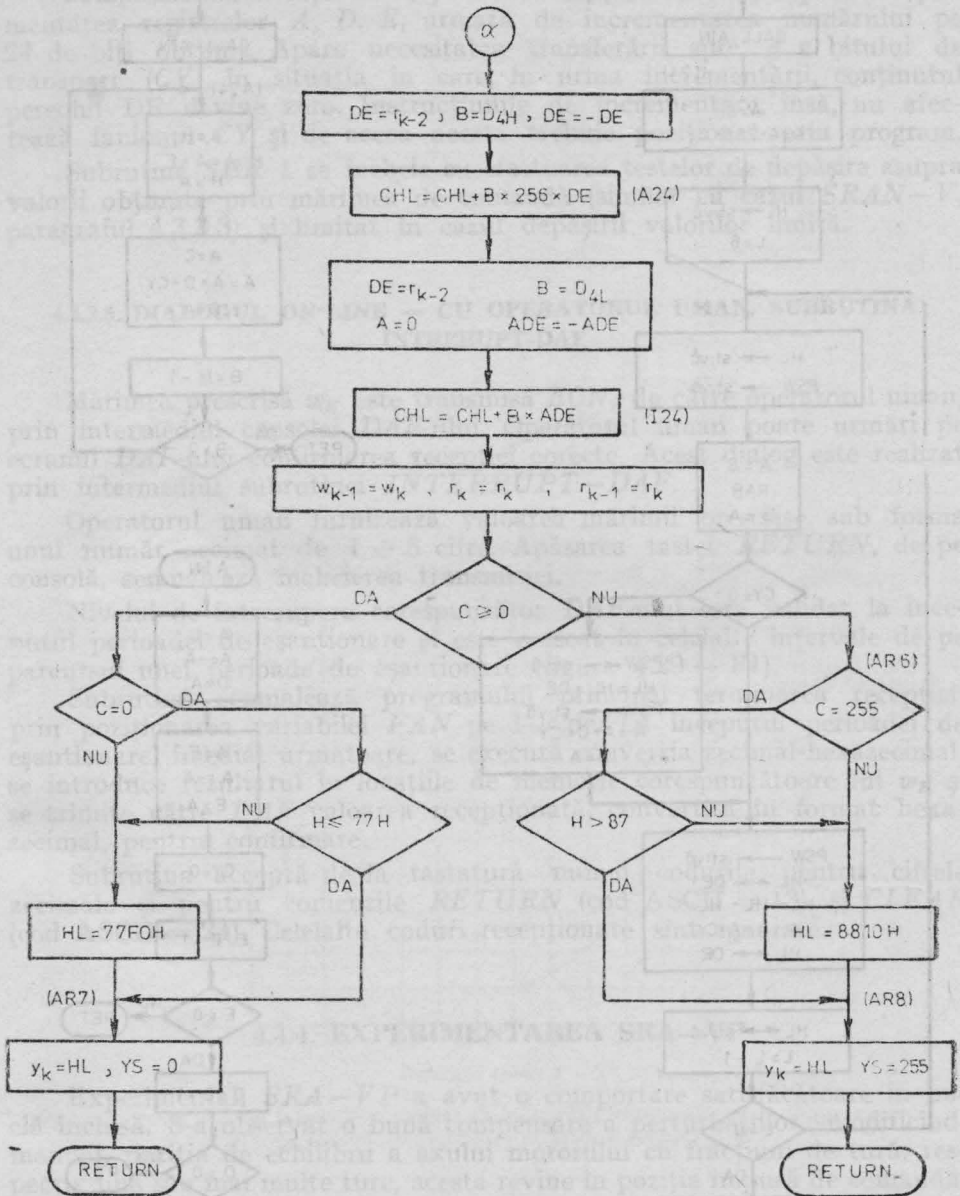
obținute pentru ARN-VP, se continuă cu efectuarea calculului mărimii de comandă. Rezultatul calculului este reprezentat sub forma unui număr pe 24 biți, deșus în registrele C, H și L. Se asigură astfel posibilitatea sesizării corecte a depășirilor (vezi paragraful 4.3.3.3).

Înmulțirile de tipul 8 biți × 24 biți = 24 biți se efectuează cu subrutina I 24, care execută operația $CHL = CHL + B \times ADE$. 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 cu subrutina A 24, în care operația $CHL = CHL + DE \times 256$ se execută de B ori, este rapidă, întrucît 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 I24, A24 și AIN sînt prezentate în organigramele din figura 4.35 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.

Fig. 4.34 (continuare)



spre stînga sînt în acest caz mai dificil de efectuat decît în cazul operațiilor pe 16 biți, prezentate în paragraful 4.3.3.3, dar principiul utilizat este similar. Se folosește și aici instrucția *DAD rp*, iar timpul de lucru cu stîva, absolut necesar, se scurtează cu ajutorul instrucției *XTHL*.

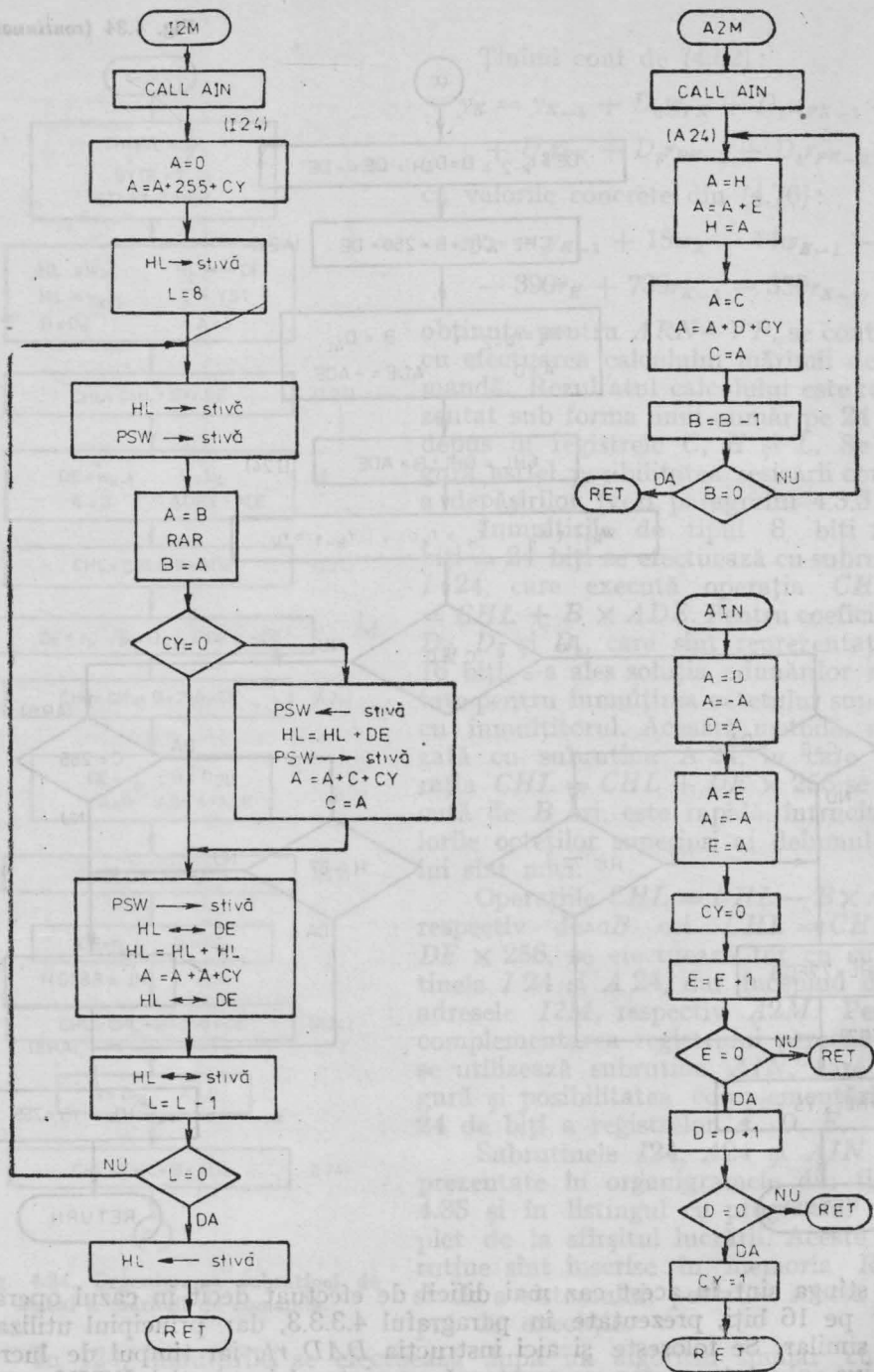


Fig. 4.35. Subrutinele I24, A24, și AIN.

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. 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 mărirea de comandă (similar cu cazul *SRAN-V*, paragraful 4.3.3.3) și limitat în cazul depășirii valorilor limită.

4.4.3.5. DIALOGUL ON-LINE — CU OPERATORUL UMAN, SUBRUTINA INTERRUPT-DAF

Mărirea 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, semnaleză încheierea transmisiei.

Nivelul de întrerupere corespunzător *DAF*-ului este validat la începutul perioadei de eșantionare și este mascat în celelalte intervale de pe parcursul unei perioade de eșantionare (figura 4.29 — F1).

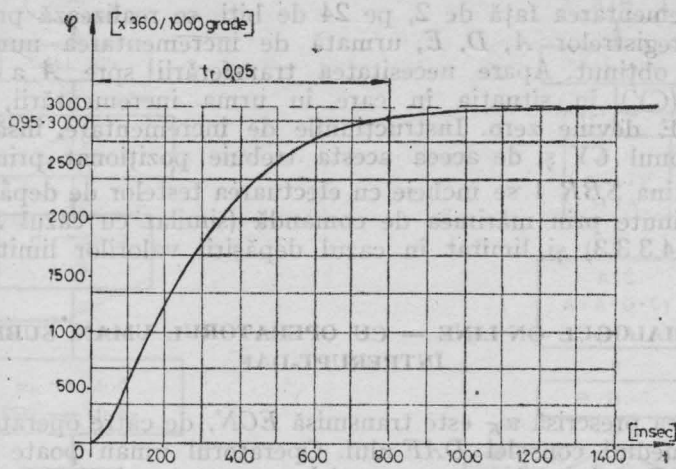
Subrutina semnaleză programului principal terminarea recepției, prin poziționarea variabilei *FAN* pe 1 logic. La începutul perioadei de eșantionare, imediat următoare, se execută conversia zecimal-hexazecimal, se introduce rezultatul în locațiile de memorie corespunzătoare lui w_K și se trimite către *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.

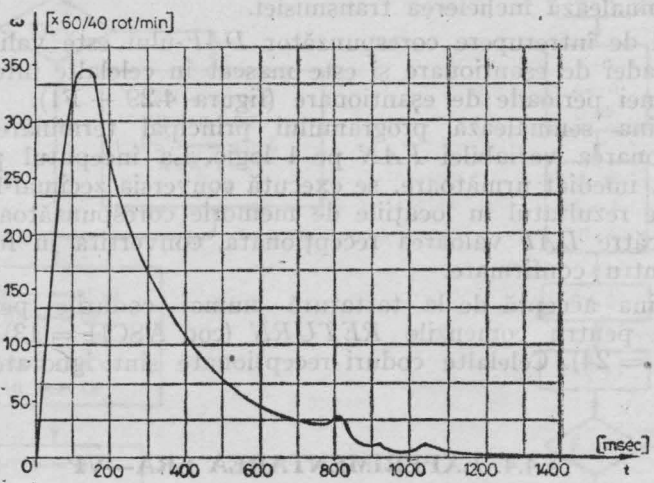
4.4.4. EXPERIMENTAREA SRA—VP

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. 4.36, σ). Măsurătorile s-au efectuat ca și în cazul sistemului de reglare a vitezei, prin mijloace software, tabelînd în memoria microcalculatorului, răspunsul sistemului în fiecare perioadă de eșantionare.



a. Poziția (unghiul)

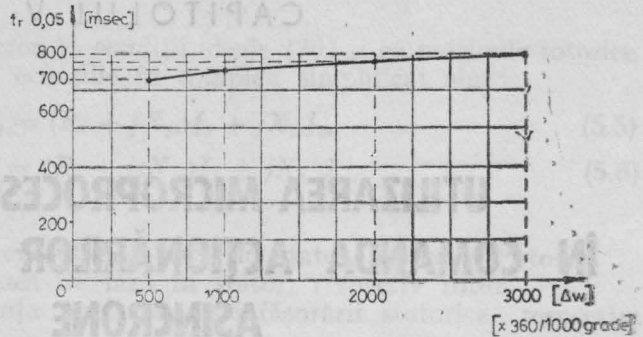


b. Viteza unghiulară

Fig. 4.36. Răspunsul sistemului de reglare a vitezei și poziției la o variație treaptă a mărimii prescrise: a) Poziția b) Viteza;

În figura 4.37 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.

Fig. 4.37. Dependența timpului de reglare obținut experimental de valoarea treptei de variație a mărimii prescrise.



4.5. CONTROLUL CURENTULUI DIN ÎNFĂȘURAREA ROTORICĂ A MOTORULUI

În sistemele de reglare a vitezei, respectiv a vitezei și poziției, realizate, se utilizează comanda în tensiune a motorului. Se impune însă și un control al curentului din înfășurare [71, 14]. Acest control asigură și o protecție a sursei de alimentare împotriva eventualelor șocuri de curent.

În ideea de a exclude complet utilizarea traductoarelor de curent și 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.

UTILIZAREA MICROPROCESOARELOR ÎN COMANDA ACȚIONĂRILOR CU MOTOARE ASINCRONE

5.1. PROBLEME SPECIFICE CU PRIVIRE LA ACȚIONĂRILE CU MAȘINA ASINCRONĂ [36]

Funcționarea motorului asincron trifazat se bazează pe existența unui câmp învîrtitor în întrefierul mașinii, creat de înfășurările statorului alimentat cu un sistem de tensiuni trifazate. Viteza de rotație a câmpului învîrtitor se numește viteză sincronă și este determinată de frecvența curentului din stator:

$$n_0 = \frac{60f_s}{p} \text{ rot/min,} \quad (5.1)$$

sau

$$\Omega_0 = \frac{2\pi f_s}{p} \text{ rad/s,} \quad (5.2)$$

unde: n_0 este viteza de rotație a câmpului învîrtitor, f_s frecvența curenților din stator, p numărul de perechi de poli.

Se definește alunecarea corespunzătoare turației n :

$$s = \frac{n - n_0}{n_0}. \quad (5.3)$$

Frecvența curenților din rotor este:

$$f_r = sf_s. \quad (5.4)$$

Pentru motorul asincron în condiții ideale [36] și cu mărimile rotorice raportate la stator [36] ecuațiile în complex simplificat sînt:

$$\underline{U}_s = (R_s + jX_{\sigma s})\underline{I}_s + jX_m \underline{I}_m, \quad (5.5)$$

$$\underline{U}_r = (R_r + jsX_{\sigma r})\underline{I}_r + jX_m s \underline{I}_m, \quad (5.6)$$

unde:

- $\underline{I}_s, \underline{I}_r$ reprezintă curentul de fază în stator, respectiv rotor,
- $\underline{U}_s, \underline{U}_r$ — tensiunea de fază în stator, respectiv rotor,
- R_s, R_r — rezistența unei faze a înfășurării statorice, respectiv rotorice,
- $X_{\sigma s}, X_{\sigma r}$ — reactanța de scăpări a unei faze statorice, respectiv rotorice,
- X_m — reactanța mutuală dintre stator și rotor,
- \underline{I}_m — curentul de magnetizare (componenta reactivă a curentului de mers în gol),

Relația dintre curentul statoric și cel rotoric este:

$$\underline{I}_s + \underline{I}_r = \underline{I}_0, \quad \underline{I}_0 \text{ fiind curentul de mers în gol.} \quad (5.7)$$

Datorită faptului că puterea transmisă rotorului este egală cu puterea absorbită, din care se scad pierderile, puterea electromagnetă este puterea pe rezistența R_r/s , deci pentru cele m_s faze statorice:

$$P_{EM} = m_s \underline{I}_r^2 \cdot \frac{R_r}{s}, \quad (5.8)$$

unde \underline{I}_r se obține prin rezolvarea sistemului (5.5), (5.6), (5.7).

Făcînd înlocuirile în relația (3.4) se obține:

$$M = \frac{m_s \cdot \dot{p}}{2\pi f_s} \left(\frac{U_s}{1 + \frac{X_{\sigma s}}{X_m}} \right)^2 \frac{\frac{R_r}{s}}{\left(R_s + \frac{R_r}{s} \right)^2 + \left(\frac{X_s}{1 + \frac{X_{\sigma s}}{X_m}} + X_m + X_{\sigma r} \right)^2}, \quad (5.9)$$

sau

$$M = \frac{m_s \cdot \dot{p}}{2\pi f_s} \left(\frac{U_s}{1 + \sigma_s} \right)^2 \frac{\frac{R_r}{s}}{\left(R_s + \frac{R_r}{s} \right)^2 + X^2} \quad (5.10)$$

cu notațiile:

$$\sigma_s = \frac{X_{\sigma s}}{X_m}$$

$$X = \frac{X_s}{1 + \sigma_s} + X_m + X_{\sigma r} \text{ — reactanța totală de scăpări a motorului.}$$

Se determină valoarea maximă a cuplului la o valoare $s = s_K$ — alunecarea critică :

$$\frac{dM}{ds} = 0,$$

$$M_K = \frac{m_s \cdot \phi}{2\pi f_s} \left(\frac{U_s}{1 + \sigma_s} \right)^2 \cdot \frac{1}{2(R_s \pm \sqrt{R_s^2 + X^2})}, \quad (5.11)$$

$$s_K = \pm \frac{R_r}{\sqrt{R_s^2 + (X_{\sigma s} + X_{\sigma r})^2}}. \quad (5.12)$$

Caracteristica mecanică a unui motor asincron se reprezintă, de obicei, pentru cazuri particulare practice, cînd $R_s \ll R_r$, și $\sigma_s \approx 1$. Se obișnuiește să se reprezinte în valori raportate la cuplul maxim și s_K :

$$\frac{M_K}{M} = \frac{2}{\frac{s}{s_K} + \frac{s_K}{s}}, \quad (5.13)$$

unde :

$$s_K \approx \frac{R_r}{X_{\sigma s} + X_{\sigma r}}. \quad (5.14)$$

Caracteristica naturală a motorului asincron este de forma din figura 5.1.

Din examinarea formei caracteristicii naturale rezultă că aceasta prezintă o oarecare rigiditate în porțiunea cuprinsă între $s = 0$ și $s = \pm s_K$. Posibilitatea supraîncărcării motorului este limitată de valoarea cuplului critic, mărime de care trebuie să se țină seama și care este de obicei indicată sub forma coeficientului de supraîncărcare $\lambda = M_K/M_n$.

Din relațiile (5.9) și (5.10) rezultă și metodele de reglare a turației motorului asincron : variația rezistenței din circuitul rotoric, variația reactanței de scăpări a rotorului (prin introducerea unei reactanțe suplimentare X_d variabile), variația alunecării, variația tensiunii de alimentare și variația frecvenței tensiunii de alimentare.

Metoda cea mai comodă și mai ieftină de reglare a turației ar fi cea prin reglarea tensiunii de alimentare, ca și la motorul de curent continuu. Domeniul de reglare obținabil are, însă, limite restrînse (teoretic pentru o alunecare cuprinsă între $s = 0$ și $s = s_K$) și deci, prin utilizarea acestei metode, motorul asincron nu este competitiv cu cel de curent continuu. În plus, prin reglarea tensiunii de alimentare este afectată și capacitatea de

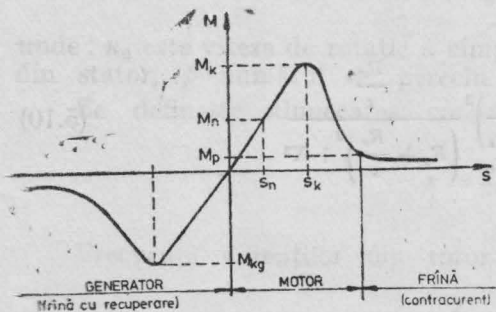


Fig. 5.1. Alura caracteristicii naturale a motorului asincron.

supraîncărcare a motorului, fenomen ce rezultă direct din relațiile (5.10) și (5.11) :

$$\frac{M_{Ki}}{M_{Kj}} = \left(\frac{U_{si}}{U_{sj}}\right)^2 \text{ și } \frac{M_i}{M_n} = \left(\frac{U_{si}}{U_{sn}}\right)^2 \quad (5.15)$$

și din caracteristicile mecanice artificiale reprezentate în forma din figura 5.2 [36, 70]. Pentru un cuplu dat, variația de turație obținabilă Δn este mică.

Din relația (5.1) se observă că o dată cu modificarea frecvenței tensiunii de alimentare, proporțional, se modifică și turația sincronă. Variația cuplului și alunecării critice cu frecvența poate fi studiată cu relațiile (5.11) și (5.12), simplificate la forma din relațiile (5.13) și (5.14), prin scoaterea în evidență a tensiunii de alimentare :

$$M_K \simeq k' \frac{U_s^2}{f_s^2} \quad (5.16)$$

și

$$s_K \simeq k'' \frac{1}{f_s}, \quad (5.17)$$

unde k' și k'' sînt constante de motor.

Pentru a păstra capacitatea de supraîncărcare constantă pentru diferite cupluri rezistente, din relația (5.16) se poate deduce condiția :

$$\frac{M_j}{M_i} = \frac{M_{Kj}}{M_{Ki}} = \left(\frac{U_{sj}}{U_{si}}\right)^2 \cdot \left(\frac{f_i}{f_j}\right)^2, \quad (5.18)$$

din care se poate determina variația tensiunii de alimentare la modificarea turației, prin schimbarea frecvenței de alimentare :

$$\left(\frac{U_{sj}}{U_{si}}\right) = \frac{f_{sj}}{f_{si}} \sqrt{\frac{M_j}{M_i}}. \quad (5.19)$$

Din (5.19) rezultă dependența impusă între tensiunea de alimentare și frecvența acesteia, dependență influențată și de variația cuplului rezistent cu turația [36].

Pentru un cuplu rezistent constant se obține :

$$\frac{U_{si}}{f_{si}} = \frac{U_{sj}}{f_{sj}}. \quad (5.20)$$

Pe de altă parte, neglijînd căderile de tensiune datorită rezistenței și reactanței de dispersie a statorului, se poate considera că tensiunea de alimentare este proporțională cu frecvența și cu fluxul de magnetizare din motor.

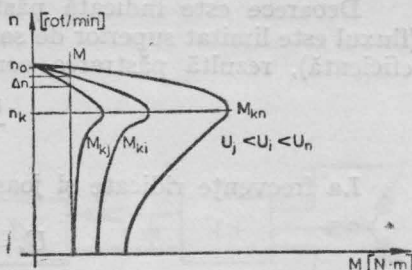


Fig. 5.2. Modul de determinare a variației de turație prin modificarea tensiunii de alimentare cu ajutorul caracteristicilor mecanice.

Deoarece este indicată păstrarea fluxului în jurul valorii nominale (fluxul este limitat superior de saturația fierului, iar inferior de exploatarea eficientă), rezultă păstrarea constantă a raportului:

$$\frac{U_s}{f_s} = \frac{U_{sn}}{f_{sn}} \quad (5.21)$$

La frecvențe ridicate și joase, relația (5.21) trebuie corectată [36]:

$$U_s = U_{sn} \frac{f}{f_n} \cdot k_R, \quad (5.22)$$

unde factorul de corecție k_R se calculează cu relația:

$$k_R = \sqrt{\frac{\lambda^2 + \left(\frac{f}{f_n}\right)^2}{\lambda^2 + 1}} \quad (5.23)$$

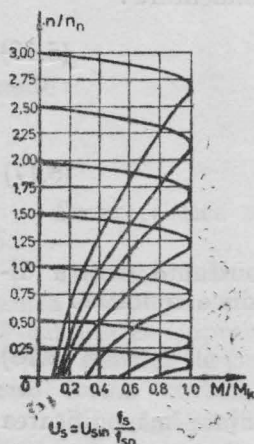


Fig. 5.3. Familia de caracteristici mecanice teoretice ale motorului asincron obținută prin $\frac{U}{f} = \text{const.}$

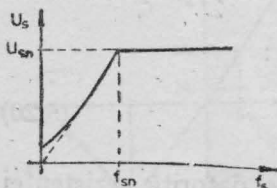


Fig. 5.4. Dependența tensiunii de frecvență la mașina asincronă.

Pentru cuplu critic constant se obțin caracteristicile artificiale mecanice de forma din figura 5.3. Se observă asemănarea caracteristicilor cu cele de la motorul de curent continuu.

Din aceste considerente rezultă că metoda variației frecvenței tensiunii de alimentare, corelată cu variația amplitudinii acesteia, cunoscută sub denumirea „metoda de reglare prin $U/f = \text{constant}$ ” [62], este o metodă foarte bună de reglare a turației pentru un motor asincron.

Principial, se obține o gamă de reglare foarte largă, competitivă, din acest punct de vedere, cu motorul de curent continuu. De asemenea, se conservă rigiditatea caracteristicilor și nu este afectată capacitatea de supraîncărcare a motorului.

Practic, însă, metoda implică limitări în aplicare.

Tensiuni de alimentare de valori peste cea nominală nu se pot folosi, pentru a nu periclita izolația mașinii.

La frecvențe joase rezistența statorică are valoarea mare în raport cu reactanțele, relațiile (5.15), (5.16), (5.17) nu sînt suficient de exacte și cuplul critic scade dacă raportul U/f este constant [36, 28, 71].

În figura 5.4 se artă cum trebuie corelate, practic, tensiunea de alimentare și frecvența. La frecvențe joase, $U/f > U_{sn}/f_n$ pentru obținerea aceluiași cuplu, iar la frecvențe peste cea nominală, tensiunea se menține constantă.

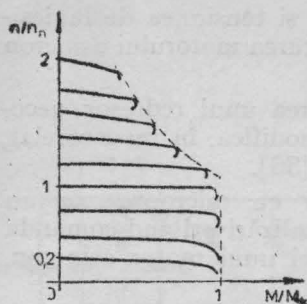


Fig. 5.5. Familia de caracteristici mecanice realizabilă la schimbarea frecvenței.

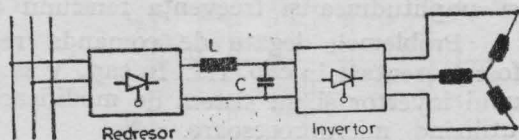


Fig. 5.6. Schema bloc a convertorului cu circuit intermediar de curent continuu.

Menținând constantă tensiunea, la frecvențe peste cea nominală, fluxul din mașină scade, ceea ce determină micșorarea cuplului dezvoltat [71].

Prin urmare, din punctul de vedere al aplicației practice, metoda $U/f = \text{const.}$ este utilă, mai ales la frecvențe sub cea nominală. Respectând dependența $U_s = U_s(f_s)$ din figura 5.4, se obține familia de caracteristici din figura 5.5.

5.2. SISTEMUL DE REGLARE A VITEZEI MOTOARELOR ASINCRONE PRIN CONVERTOARE DE FRECVENȚĂ INDIRECTE — CU CIRCUIT INTERMEDIAR DE CURENT CONTINUU

În paragraful 5.1 am arătat posibilitățile de reglare a turației la mașinile asincrone cu rotor în scurtcircuit, subliniind avantajele metodei variației frecvenței tensiunii de alimentare corelate cu variația amplitudinii acesteia.

În [36, 70, 28] se consideră această metodă ca fiind și cea mai economică.

Soluția cea mai cunoscută, care valorifică metoda amintită, este prin utilizarea convertoarelor cu circuit intermediar de curent continuu (figura 5.6), [36].

Aceste convertoare se caracterizează printr-o dublă conversie a energiei electrice și anume:

- transformarea tensiunii alternative, de frecvență constantă, a rețelei într-o tensiune continuă, prin intermediul unui redresor;

- transformarea tensiunii continue din circuitul intermediar într-o tensiune alternativă mono sau trifazată de frecvență reglabilă, folosind un invertor static.

Modificînd concomitent frecvența inverterului și tensiunea de la ieșirea redresorului comandat, se poate realiza alimentarea motorului asincron cu un raport $U/f = \text{const.}$

Metoda poate fi implementată și prin utilizarea unui redresor necomandat, urmat de un inverter, la care se pot modifica, în mod corelat, și amplitudinea și frecvența tensiunii de ieșire [36].

Problemele legate de comanda redresoarelor cu microprocesor au fost prezentate în cap. III. În cap. V se prezintă realizări privind comanda unui inverter și un sistem de modificare a turației unui motor asincron, utilizînd microprocesoare.

5.3. INVERTOARE

Invertoarele sînt circuite electronice care transformă energia de curent continuu în energie de curent alternativ. Cu ajutorul lor este posibilă transformarea unei tensiuni continue într-o tensiune alternativă, de o anumită formă, amplitudine și frecvență [62, 58].

În ultimii ani, utilizarea invertoarelor a cunoscut o largă răspîndire. În [28, 39] se fac clasificări pe tipuri de invertoare, în funcție de modul concret constructiv al acestora, de modul în care se realizează comutația sau de dispunerea dispozitivelor semiconductoare din structura acestora.

În preocupările privind utilizarea microprocesoarelor în comanda acționărilor electrice, am utilizat două tipuri de invertoare. Experiența dobîndită în comanda acestora permite extinderea aplicațiilor și la alte tipuri de invertoare.

5.3.1. INVERTOR TRIFAZAT CU TENSIUNE DE IEȘIRE SINTETICĂ, CU CIRCUIT DE STINGERE COMUN PENTRU TOATE TIRISTOARELE

Acest tip de inverter face parte din familia invertoarelor cu tiristoare în punte trifazată, fiind unul dintre circuitele folosite în acționarea unor motoare de mare putere. La aceste invertoare se impun măsuri de siguranță suplimentare, în vederea stingerii sigure a tiristoarelor care urmează să iasă din conducție, măsuri ce constau în utilizarea unor circuite de comutație, în care se introduc tiristoare auxiliare [61].

Schema circuitului este prezentată în figura 5.7. Forma tensiunilor U_R , U_S , U_T , în trepte, se sintetizează comutînd corespunzător tiristoarele principale $T1 \div T5$. De exemplu, dacă se comandă tiristoarele $T1$, $T4$, $T5$, amplitudinea tensiunii pe Z_S este dublă față de cea a tensiunilor pe Z_R și Z_T , întrucît curentul prin $T4$ și Z_R este suma curenților prin $T1$ și Z_R , respectiv prin $T5$ și Z_T . Segmentele componente ale tensiunii de

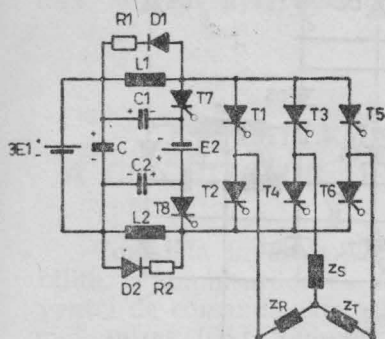


Fig. 5.7. Invertor trifazat cu tensiunea de ieșire sintetică.

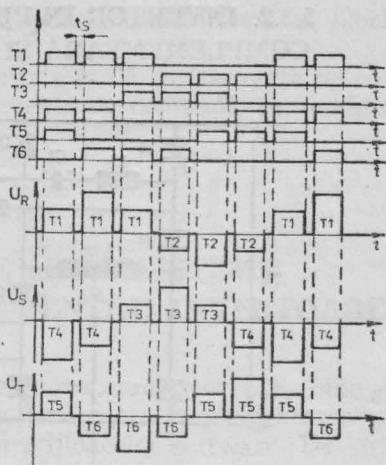


Fig. 5.8. Succesiunea comenzilor la tiristoare și forma tensiunii de ieșire.

ieșire sînt departajate cu intervale t_s , de durate egale, în care are loc stingerea tiristoarelor care au condus.

Sucesiunea comenzilor aplicate tiristoarelor constituie așa-numita secvență de comenzi a invertorului [65] și trebuie respectată întocmai, pentru a se obține forma corespunzătoare a tensiunii de ieșire. Pentru invertorul descris la acest paragraf, secvența de comenzi și forma tensiunii de ieșire sînt prezentate în figura 5.8

Întrucît după fiecare interval de conducție se blochează toate tiristoarele, la începutul fiecărui nou interval de conducție trebuie comandate două dintre tiristoarele care au condus și anterior și unul care a fost blocat anterior. În decurs de o perioadă a tensiunii de ieșire, fiecare tiristor trebuie comandat de trei ori consecutiv.

Stingerea tiristoarelor principale este realizată cu circuitul de stingere compus din tiristoarele $T7$ și $T8$, grupul $L1 - C1 - C - C2 - L2$ și sursa auxiliară $E2$. Funcționarea corectă a circuitului presupune amorsarea, la începutul unui ciclu de comenzi, a unuia dintre tiristoarele de stingere.

Circuitul este prezentat în detaliu în [61, 62, 63]. Frecvența tensiunii de ieșire este direct dependentă de frecvența (f_0) de comandă a tiristoarelor, după relația :

$$f = f_0/6. \quad (5.24)$$

Frecvența de comandă a tiristoarelor de stingere este :

$$f_s = 3f = \frac{f_0}{2}. \quad (5.25)$$

5.3.2. ÎNVERTOR ÎN PUNTE TRIFAZATĂ CU COMUTAȚIE COMPLEMENTARĂ ȘI TENSIUNE DE IEȘIRE SINTETICĂ

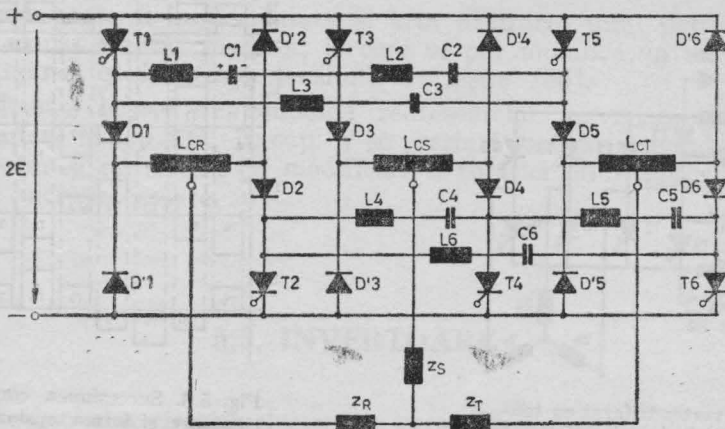
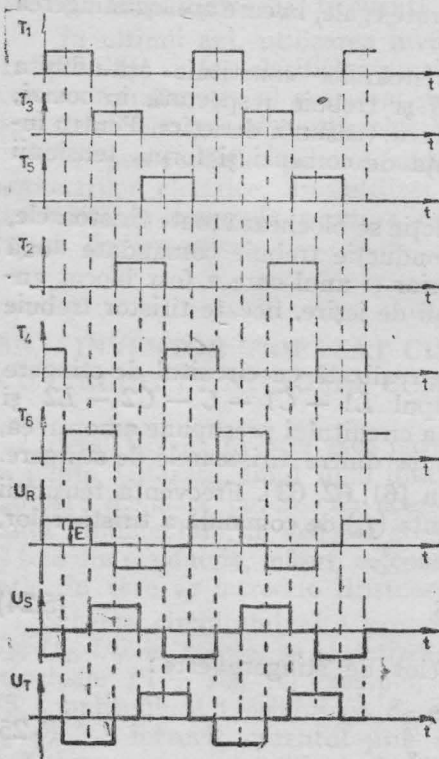


Fig. 5.9. Schema electrică a inverterului.



Schema circuitului este prezentată în figura 5.9.

Principiul de funcționare al circuitului este asemănător cu cel al schemei anterioare, cu deosebire că tiristoarele $T1 \div T6$ joacă un rol mixt: și de tiristor principal și de tiristor de stingere. Comanda tiristoarelor se face după o secvență impusă, reprezentată în diagramele din figura 5.10. Se observă că la acest inverter se află simultan în conducție numai câte două tiristoare.

Tiristoarele $T1, T3$ și $T5$, respectiv $T2, T4$ și $T6$, se sting reciproc prin comenzi adecvate. De exemplu, dacă $T1$ este, la un moment dat, în conducție, $C1$ este încărcat la valoarea tensiunii dintre faze. Stingerea lui $T1$ are loc în momentul comandării lui $T3$. Acesta intră în conducție și tensiunea de pe $C1$ se aplică cu

Fig. 5.10. Secvența de comandă a tiristoarelor și forma tensiunii de ieșire.

(+) pe catodul lui T1 și cu (-) pe anod, polarizându-l invers și blocându-l [68, 62].

Frecvența tensiunii sintetizate la ieșire este egală cu frecvența de comandă a fiecărui tiristor. Modificând frecvența de comandă se modifică, deci, și frecvența tensiunii de ieșire.

5.4. IMPLEMENTAREA PE MICROSISTEME A CIRCUITELOR DE COMANDĂ PENTRU INVERTOARE

Comanda invertoarelor presupune realizarea unei secvențe, bine stabilite, a impulsurilor de comandă pentru tiristoare. Implementarea secvenței de comandă este adecvată utilizării mijloacelor software. De altfel, unii autori [65] folosesc termenul de „program de comandă” pentru succesiunea impulsurilor de comandă a tiristoarelor.

Desigur că programul de comandă a tiristoarelor se poate sintetiza și cu mijloacele electronicii clasice. Utilizarea microprocesorului, la comanda invertoarelor, aduce, însă, avantaje indiscutabile [13, 31]:

- permite sintetizarea secvenței de comandă extrem de simplu, cu câteva instrucții;

- permite schimbarea frecvenței de comandă prin simpla schimbare a valorii unui număr;

- permite implementarea de noi funcții pentru circuitul de comandă, de exemplu: cu un singur circuit de comandă se pot comanda mai multe invertoare sau se poate realiza o buclă de reglare a frecvenței tensiunii de ieșire cu același circuit care asigură și comanda propriu-zisă a invertorului.

În ceea ce privește comanda propriu-zisă a invertorului, rolul micro-sistemului este de a furniza secvența de comandă impusă (F_1). El generează printr-unul din porturile sale de ieșire, impulsuri a căror durată, t_i , este în funcție de tipul tiristoarelor din invertor și care se repetă conform unei diagrame dinainte stabilite, dar cu frecvență de repetiție variabilă.

5.4.1. INVERTOR TRIFAZAT CU TENSIUNE DE IEȘIRE SINTETICĂ ȘI CIRCUIT DE STINGERE COMUN, COMANDAT CU MICROPROCESOR

Structura circuitului de putere este prezentată în figura 5.7. Secvența de comandă ce trebuie realizată și forma tensiunii trifazate sintetizate la ieșire sînt reprezentate în figura 5.8.

Impulsurile de comandă pentru tiristoare se obțin pe cele opt linii ale unuia dintre porturile de ieșire ale microsistemului utilizat ca circuit

de comandă. Izolarea galvanică între circuitul de putere și microsistem este realizată cu optocuploare.

Între tiristoarele inverterului și liniile portului de ieșire se stabilește o corespondență biunivocă, de exemplu astfel: $T1$ — linia 7, $T2$ — linia 6, pînă la $T8$ — linia 0.

Prima combinație de comenzi (figura 5.8) impune amorsarea tiristoarelor $T1$, $T4$ și $T5$. Ca urmare, portul de ieșire trebuie poziționat în starea $10011000 = 98H$. Se impun, apoi, comenzi pentru $T1$, $T4$ și $T6$ și deci portul trebuie poziționat în starea $10010100 = 94 H$ ș.a.m.d.

Pentru comanda circuitului de stingere (tiristoarele $T7$ sau $T8$) se programează portul de ieșire în starea 00000010 sau 00000001 .

Cele 8 numere binare, ce reprezintă stările portului de ieșire și, implicit, secvența de comenzi necesare pentru realizarea unei perioade a tensiunii trifazate, constituie o listă de date amplasată în memoria sistemului.

Lista de date poate fi completată cu încă un număr, care va fi convertit în frecvența tensiunii de ieșire a inverterului. Acest număr poate fi însă adus, din exterior, și la un port de intrare al sistemului. În ambele cazuri, numărul ce reprezintă frecvența poate proveni fie de la utilizator (prin consolă, panou de comandă etc.), fie de la un calculator mai puternic care dirijează procesul la nivel superior.

Secvența de impulsuri necesară în comandă se poate obține prin mijloace software sau prin utilizarea timerului și sistemului de întreruperi.

În organigrama din figura 5.11 este prezentat un exemplu de program care realizează comanda inverterului. Prin acest program se sintetizează K trepte de frecvență pentru tensiunea de ieșire a inverterului, iar fiecare treaptă de frecvență se menține câte un interval de timp t_K (diferit pentru fiecare treaptă).

În cele ce urmează sînt redate în întregime variantele de program întocmite pentru sintetizarea comenzii inverterului cu microprocesoarele 8035, 8085 și Z80, [11, 2].

Elementul central din program este subprogramul de generare a secvenței de comenzi. În versiunile scrise pentru 8085 și Z80, acest subprogram este realizat prin mijloace exclusiv software, temporizările implicate fiind realizate prin decrementări de registre. În varianta de program întocmită pentru microprocesorul 8035 se utilizează timerul intern al microprocesorului, rezultînd o scurtare a programului, în ciuda utilizării unui microprocesor mai puțin performant.

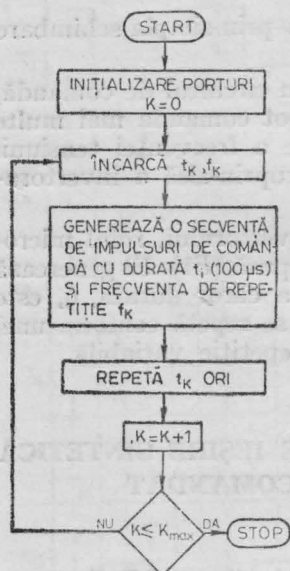


Fig. 5.11. Organigrama programului de comandă pentru generarea a K trepte de frecvență a tensiunii de la ieșirea inverterului.

8085

LXI SP, 20C2H	E2: DCR B	MOV A,L	ORA A
EE: MVI A,03	JNZ E2	CPI 06	JNZ E10
OUT 20H	LXI H,8100H	JZ E6	ORA D
PUSH PSW	E3: MOV A,M	RAR	JZ E11
E0: LXI H,8106H	OUT 21H	JNC E7	E10: MVI L,0
E1: MOV, B,M	MVI B,07	MVI A,02	JMP E7
INR L	E4: DCR B	OUT 21H	E11: POP PSW
MOV C,M	JNZ E4	JMP E8	DCR A
INR L	XRA A	E7: MVI A,01	PUSH PSW
MOV D,M	OUT 21H	OUT 21H	CPI 02
INR L	POP B	E8: MVI B,07	JZ E12
MOV E,M	PUSH B	E9: DCR B	DCR A
INR L	E5: DCX B	JNZ E9	JZ EE
MVI A,01	MOV A,C	POP B	MVI L,0EH
OUT 21H	ORA B	JMP E3	JMP E1
PUSH B	JNZ E5	E6: DCX D	E12: MVI L,0AH
MVI B,07	INR L	MOV A,E	JMP E1

8035

DIS TCNT I	DJNZ R0,E5
JMP E1	JMP E2
E1: MOV R5, #16	IMP: OUTL P1,A
E2: INC R5	MOV R3, #t ₁
MOV A,R5	E4: DJNZ R3,E4
JB5 E1	ANL P1, #00
MOVP3 A, (A) A	RETR
MOV R2,A	
INC R5	
MOV A,R5	
MOVP3 A, (A) A	
MOV R1,A	
INC R5	
MOV A,R5	
MOVP3 A, (A) A	
MOV R0,A	
MOV A, #01	
CALL IMP	
E5: MOV R7, #11	
MOV R6, #06	
E11: MOV A,R7	
MOVP3 A, (A) A	
CALL IMP	
MOV A,R2	
MOV T,A	
STRT T	
E7: JTF E8	
JMP E7	
E8: INC R7	
CPL F0	
MOV A, #02	
JF0 E9	
DEC A	
E9: CALL IMP	
DJNZ R6,E11	
DJNZ R1,E5	
	E0: LD HL,3106H
	E1: LD B,(HL)
	INC L
	LD C,(HL)
	INC L
	LD D,(HL)
	INC L
	LD E,(HL)
	INC L
	EXX
	LD DE,0102H
	LD BC,t ₁ 61H
	OUT (C),D
	E2: DJNZ E2
	LD HL,3100H
	E9: LD A,(HL)
	OUT (C),A
	LD B,t ₁
	E3: DJNZ E3
	OUT (C),B
	EXX
	PUSH BC
	E4: DEC BC
	LD A,C
	OR A,B

Z80

JR NZ,E4
 POP BC
 EXX
 INC L
 LD A,L
 CP A,06
 JR Z,E5
 RRA
 JR NC,E6
 OUT (C),E
 JR E7
 E6: OUT (C),D
 E7: LD B,t₁

E8: DJNZ E8
 JR E9
 E5: LD L,0
 EXX
 DEC DE
 LD A,D
 OR A,E
 EXX
 JR NZ,E6
 EXX
 BIT 4,L
 JR NZ,E1
 JR E0

În figura 5.12 se prezintă organigrama subprogramului de generare a secvenței de comenzi pentru inverter, în varianta implementată pe microprocesorul 8085.

Între programele întocmite pentru cele trei tipuri de micro sisteme nu există deosebiri semnificative. Se poate concluziona că oricare dintre aceste microprocesoare este utilizabil cu rezultate foarte bune în comanda inverterului, chiar în aplicații în care microprocesorul execută un program ceva mai complex decât generarea propriu-zisă a secvenței de comenzi,

cum este în cazul descris, de generarea mai multor trepte de viteză.

Efectuînd temporizările necesare în program cu ajutorul generatorului de timp real și al sistemului de întreruperi, microprocesoarele 8085 și Z80 pot executa și programe complexe, cum ar fi o buclă de reglare a turației unui motor asincron.

Frecvența maximă a tensiunii de ieșire ce se poate sintetiza cu inverterul prezentat la acest paragraf este limitată numai de timpul asigurat revenirii tiristoarelor principale, înaintea unei noi comenzi, după amorsarea tiristoarelor de stingere (t_s — figura 5.8) și nu depinde de vitezele cu care operează microprocesorul utilizat în comandă.

Astfel [61], la comanda unuia dintre tiristoarele de stingere, blocarea tiristoare-

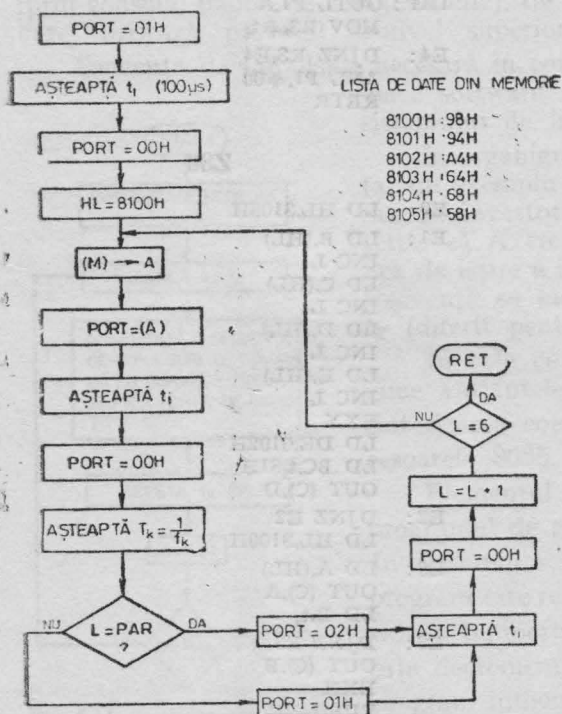


Fig. 5.12. Subprogram de generare a secvenței de comenzi, pentru tiristoarele inverterului, cu microprocesorul 8085.

lor principale se petrece pe o durată egală cu un sfert din constanta de timp de descărcare rezonantă a condensatorului C_1 (C_2) ce menține tensiunea anod-catod necesară stingerii. Pentru $L_1 = L_2 = 100 \mu\text{H}$ și $C_1 = C_2 = 330 \mu\text{F}$, valori ce au fost calculate avându-se în vedere datele de catalog ale tiristoarelor din schemă, rezultă intervalul de timp necesar pentru blocare:

$$\tau = \frac{2\pi}{4} \sqrt{L_1 C_1} = 285,35 \mu\text{s}. \quad (5.26)$$

Ținând cont de faptul că impulsul de comandă al tiristoarelor este, tipic, $t_i = 100 \mu\text{s}$ și că în acest interval de timp tiristorul amorsează singur, rezultă că din momentul comenzii pe grilă a unui tiristor și pînă la blocarea acestuia, împreună cu tiristoarele principale, este necesar un interval de timp:

$$\Delta T = t_i + \tau = 385,35 \mu\text{s}. \quad (5.27)$$

După intervalul ΔT , de la comanda unui tiristor de stingere, tiristoarele principale se blochează. Un plus de siguranță în funcționare se obține alegînd $\Delta T = 500 \mu\text{s}$ (de fapt, în realizarea concretă, $504 \mu\text{s}$) intervalul dintre comanda tiristoarelor de stingere și a celor de putere.

Pentru un factor de umplere $1/2$ a tensiunii sintetizate la ieșire rezultă o frecvență de comandă a tiristoarelor principale:

$$f_{0\max} = \frac{1}{2\Delta T} = \frac{1}{2 \cdot 500 \mu\text{s}} = 1 \text{ kHz}. \quad (5.28)$$

Valoarea (5.28) reprezintă frecvența maximă de comandă a tiristoarelor principale, deoarece durata de conducție a acestora nu se micșorează sub valoarea duratei lor de blocare, pentru a se obține o valoare medie acceptabilă a tensiunii sintetizate la ieșire [61].

Rezultă deci frecvența maximă obținabilă cu acest tip de invertor:

$$f_{\max} = \frac{f_{0\max}}{6} = 166,6 \text{ [Hz]}, \quad (5.29)$$

adică durata minimă:

$$T_{\min} = \frac{1}{f_{\max}} = 6 \text{ ms}. \quad (5.30)$$

În aplicații obișnuite însă, frecvența tensiunii de ieșire se reglează de la 0 la 150 Hz (de la 0 pînă la triplul frecvenței rețelei).

Folosind microprocesorul 8085, cu frecvența impulsurilor de tact 3,14 MHz, valoarea incrementului, la contorizarea intervalelor de timp, este de $8 \mu\text{s}$. Prin urmare, rezoluția sintetizării duratei minime a tensiunii de ieșire este:

$$\frac{8 \cdot 10^{-6}}{6 \cdot 10^{-3}} = 1,33 \cdot 10^{-3}. \quad (5.31)$$

La o frecvență de 50 Hz a tensiunii sintetizate la ieșirea inverterului, rezoluția este :

$$\frac{8 \cdot 10^{-6}}{20 \cdot 10^{-3}} = 0,4 \cdot 10^{-3}. \quad (5.32)$$

Invertorul trifazat cu tensiune de ieșire sintetică și de frecvență variabilă, comandat cu microprocesor, poate avea ca sarcină un motor asincron trifazat. Dacă se modifică frecvența tensiunii sintetizate, se modifică și turația motorului. Reglajul frecvenței tensiunii trebuie corelat și cu reglajul amplitudinii. Menținând constantă amplitudinea tensiunii de alimentare și modificând numai frecvența, se ajunge la următoarele situații :

a) la creșterea frecvenței, fluxul din mașină scade, ceea ce determină scăderea cuplului dezvoltat cu pătratul frecvenței [28].

b) la scăderea frecvenței, scad valorile reactanțelor și crește curentul prin indus și se poate ajunge la saturația miezului.

De aceea, numai reglajul frecvenței, necorelat cu reglajul amplitudinii tensiunii care alimentează motorul asincron, nu se utilizează ca metodă de reglare a turației (vezi paragraful 5.1, figurile 5.4 și 5.5).

5.5. MODIFICAREA TURĂȚIEI UNUI MOTOR ASINCRON PRIN METODA $U/f = \text{CONSTANT}$, CU COMANDĂ REALIZATĂ DE MICROPROCESOR

Principiul și avantajele metodei au fost prezentate în paragraful 5.1. Schema bloc a sistemului realizat este cea din figura 5.13. [29].

Structura prezentată nu reprezintă un sistem de reglare a vitezei în sensul riguros al termenului [14], întrucât nu conține bucla de reacție negativă. Sistemul asigură, însă, posibilitatea de a modifica simultan frecvența și amplitudinea tensiunii de alimentare a motorului, și, în condiții de cuplu rezistent cunoscut, permite modificarea turației unui motor asincron.

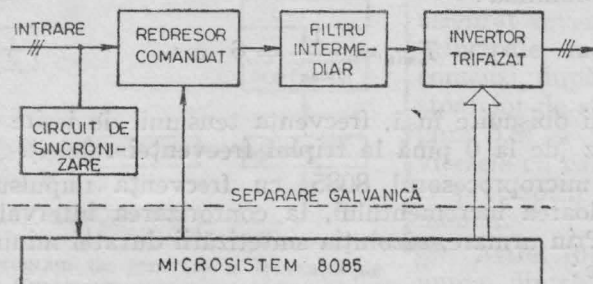


fig. 5.13. Schema bloc a sistemului pentru modificarea turației unui motor asincron.

Redresorul utilizat este un redresor trifazat în punte complet comandată. Schema și modul de comandă pentru acest tip de redresor au fost prezentate în cap. III (figurile 3.6, 3.7 și 3.13).

Invertorul din sistem este de tipul cu comutație complementară și tensiune de ieșire sintetică, prezentat în paragraful 5.3.2. (figura 5.9).

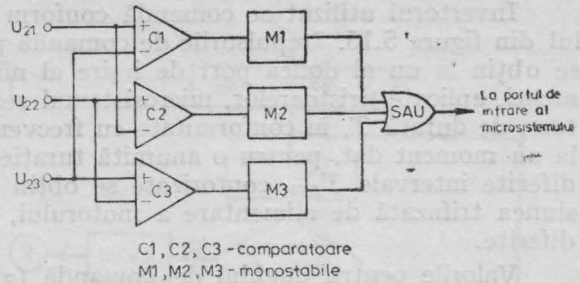


Fig. 5.14. Circuit de sincronizare cu momentele egalității a cite două faze (pozitive) ale rețelei de alimentare trifazate.

Cele două circuite de putere sînt comandate de un microsistem, care trebuie să îndeplinească, în acest tip de aplicație, următoarele funcții:

(F1) — Sesizarea momentului egalității, în domeniul valorilor pozitive, a două faze ale rețelei trifazate de alimentare. În acest scop se utilizează un circuit exterior de sincronizare, realizat conform schemei din figura 5.14 și care generează un impuls în momentul egalității a două faze ale rețelei. Impulsul de sincronizare este preluat de sistem pe o linie a unuia dintre porturile de intrare.

(F2) — Comanda redresorului la diferite unghiuri (α) de comandă, în raport cu impulsurile de sincronizare, în vederea realizării de valori medii variabile pentru tensiunea de la ieșirea redresorului (vezi paragraful 3.2).

Această funcție presupune realizarea unei temporizări corespunzătoare valorii unghiului α impusă la un moment dat și generarea impulsurilor de comandă pe grilă pentru tiristoarele redresorului, pe 6 linii ale unuia dintre porturile de ieșire ale sistemului.

(F3) — Comanda invertorului, după secvența necesară funcționării acestuia și la intervale de timp variabile între două comenzi succesive, pentru sintetizarea tensiunii trifazate cu frecvența variabilă, la ieșire.

Adresa din memorie	Conținutul adresei								HEXA	Tiristoare din invertor comandate
	Binar									
	7	6	5	4	3	2	1	0		
A5	0	0	0	0	1	0	0	1	09	T_1, T_4
A4	0	0	1	0	0	0	0	1	21	T_1, T_6
A3	0	0	1	0	0	1	0	0	24	T_3, T_6
A2	0	0	0	0	0	1	1	0	06	T_2, T_3
A1	0	0	0	1	0	0	1	0	12	T_2, T_5
A0	0	0	0	1	1	0	0	0	18	T_4, T_5

Fig. 5.15. Secvența de comandă a invertorului și modul de tabelare a acesteia în memoria sistemului.

Invertorul utilizat se comandă conform secvenței prezentate în tabelul din figura 5.15. Impulsurile de comandă pentru tiristoarele invertorului se obțin la un al doilea port de ieșire al microsistemului. Între două comenzi, aplicate tiristoarelor, microsistemul realizează câte o buclă de așteptare, de durată T , în conformitate cu frecvența tensiunii trifazate impusă, la un moment dat, pentru o anumită turație a motorului asincron. Pentru diferite intervale T_{INV} contorizate se obțin frecvențe diferite pentru tensiunea trifazată de alimentare a motorului, și, implicit, trepte de turație diferite.

Valorile pentru unghiul de comandă (α) al redresorului și pentru durata T_{INV} , ce corespunde frecvenței tensiunii de la ieșirea invertorului sînt corelate. Ele sînt calculate anterior, de către utilizator, și tabelate în memoria sistemului, pe perechi de valori astfel încît, la diferite turații ale motorului asincron, să fie îndeplinită condiția $U/f = \text{const.}$ pentru tensiunea de alimentare a motorului, sintetizată la ieșirea sistemului de comandă.

5.5.1. IMPLEMENTAREA SOFTWARE A COMENZILOR ÎN SISTEMUL PENTRU MODIFICAREA TURĂȚIEI UNUI MOTOR ASINCRON

Organigrama programului este prezentată în figura 5.16. În aceeași figură este reprezentată și organigrama subrutinei, *SECV*, utilizată la comanda invertorului. Impulsul de comandă pentru tiristoare ($t_i = 100 \mu\text{s}$) este generat prin secvența de program *IMP*. Temporizările necesare (α , T_{INV} , t_i) sînt realizate prin mijloace software.

După inițializările de la începutul programului se încarcă în registrele interne valoarea T_{INV} , necesară realizării unei anumite trepte de turație și o valoare inițială α_1 pentru unghiul de comandă al redresorului, se așteaptă impulsul de sincronizare și se comandă, după α_1 , redresorul ($\alpha_1 \approx 85^\circ$).

Se încarcă apoi valoarea α , propriu-zis necesară pentru obținerea treptei de turație dorite.

Programul este parcurs, în continuare, după o structură concepută modular, pe două bucle, în care se realizează contorizarea duratelor α și T_{INV} . Numerele binare, ce reprezintă cele două durate, sînt înscrise în registrele pereche *DE* și, respectiv, *BC* și sînt succesiv decrementate. În funcție de anularea conținutului unuia sau altuia dintre registrele pereche programul decide dacă trebuie comandat redresorul sau invertorul și trecerea, într-una sau alta dintre cele două bucle, asigurînd totodată evitarea suprapunerilor de comenzi.

În vederea egalizării pașilor de decrementare a conținutului registrelor, la trecerea prin cele două bucle, a fost necesară introducerea unor instrucțiuni suplimentare pentru compensarea duratelor de parcurgere.

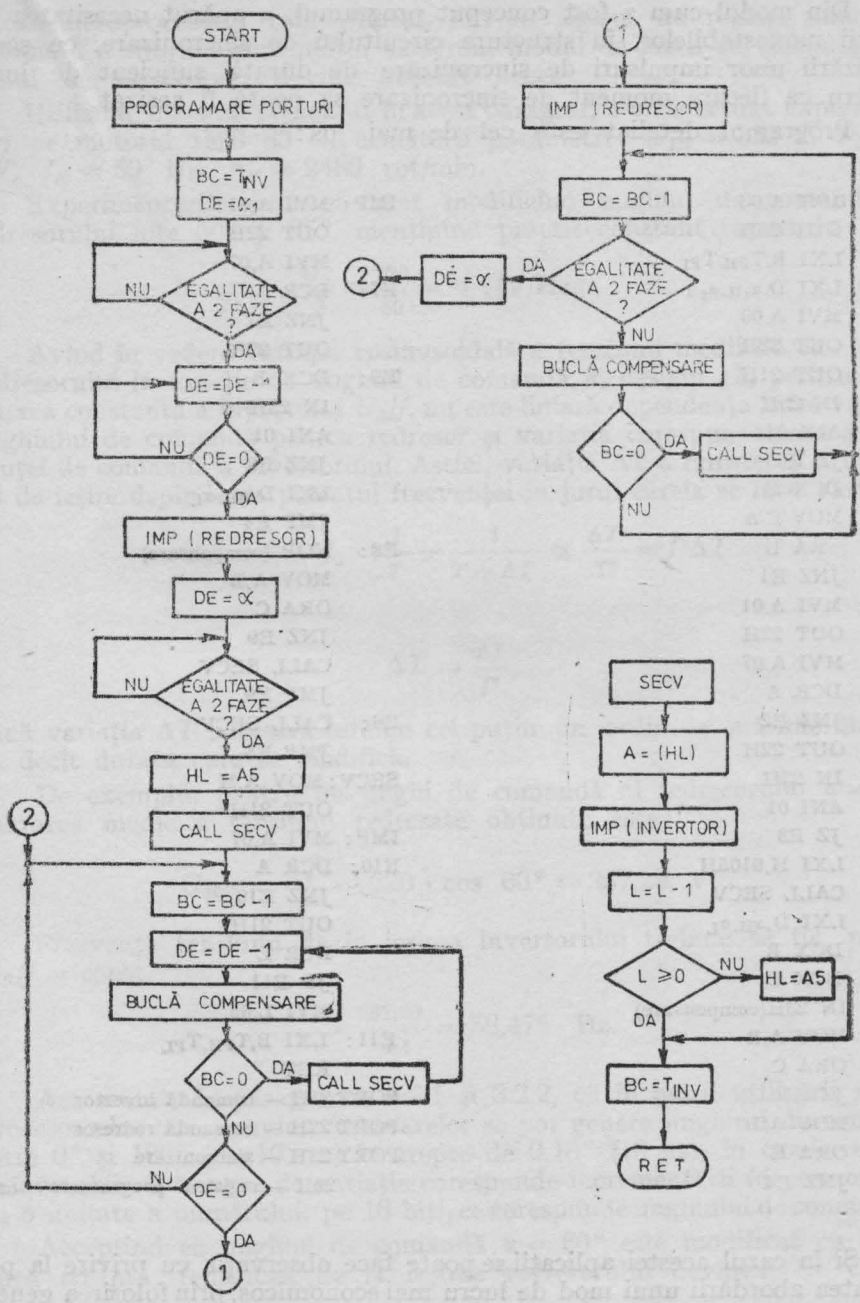


Fig. 5.16. Organigrama programului de comandă al sistemului de modificare a turației unui motor asincron.

Din modul cum a fost conceput programul, a apărut necesitatea utilizării monostabilelor, în structura circuitului de sincronizare, cu scopul realizării unor impulsuri de sincronizare de durată suficient de lungă, pentru ca fiecare moment de sincronizare să poată fi sesizat.

Programul detaliat este cel de mai jos :

MVI A,03	IMP: MVI A,01
OUT 20H	OUT 22H
LXI B,T _{PH} T _{PL}	MVI A,07
LXI D, α_1H,α_2H	E7: DCR A
MVI A,00	JNZ E7
OUT 22H	OUT 22H
OUT 21H	E9: DCX B
E0: IN 23H	IN 23H
ANI 01	ANI 01
JZ E0	JNZ E8
E1: DCX D	LXI D, α_H,α_L
MOV C,A	JMP E4
RA D	E8: NOP (compensare)
JNZ E1	MOV A,B
IMP: MVI A,01	ORA C
OUT 22H	JNZ E9
MVI A,07	CALL, SECV
E2: DCR A	JMP E9
JNZ E2	E6: CALL, SECV
OUT 22H	JMP E5
E3: IN 23H	SECV: MOV A,M
ANI 01	OUT 21H
JZ E3	IMP: MVI A,07
LXI H,9105H	E10: DCR A
CALL, SECV	JNZ E10
LXI D, α_H,α_L	OUT 21H
E4: DCX B	DCR L
E5: DCX D	JP E11
IN 23H(compensare)	MVI L,05
MOV A,B	E11: LXI B,T _{PH} T _{PL}
ORA C	RET
JZ E6	PORT 21H – comandă inverter
MOV A,D	PORT 22H – comandă redresor
ORA E	PORT 23H – sincronizare
JNZ E4	20H – registru programare orturi

Și în cazul acestei aplicații se poate face observația cu privire la posibilitatea abordării unui mod de lucru mai economic, prin folosirea generatoarelor de timp real și a sistemului de întreruperi. În acest fel, microprocesorul este degrevat de sarcina contorizărilor și poate efectua alte operații.

Dacă microprocesorul este utilizat într-un sistem de reglare numerică automată a turației unui motor asincron, modul de lucru cu timere și sistemul de întreruperi devine absolut necesar.

Utilizând sistemul prezentat în acest paragraf, s-au efectuat experimentări pe motorul IMB 83 cu următorii parametri: $U_{1N} = 380$ V, $P_N = 4$ kW, $f_N = 50$ Hz, $n_N = 2480$ rot/min.

Experimentările s-au efectuat modificând unghiul de comandă al redresorului între 60° și 85° , menținând practic constant raportul:

$$\frac{U_s}{f} = \frac{220}{50} = 4,4 \text{ [V/Hz]}. \quad (5.33)$$

Avînd în vedere variația cosinusoidală a tensiunii medii de la ieșirea redresorului în funcție de unghiul de comandă α , desigur că, pentru menținerea constantă a raportului U_s/f , nu este liniară dependența între variația unghiului de comandă pentru redresor și variația corespunzătoare a frecvenței de comandă a inverterului. Astfel, variația ΔT a duratei T a tensiunii de ieșire depinde de pătratul frecvenței în jurul căreia se face variația:

$$\Delta f = f_1 - f_2 = \frac{1}{T} - \frac{1}{T + \Delta T} \cong \frac{\Delta T}{T^2} = f^2 \Delta T \quad (5.34)$$

și

$$\Delta T = \frac{\Delta f}{f^2}, \quad (5.35)$$

dacă variația ΔT necesară este cu cel puțin un ordin de mărime mai mică decît durata care se modifică.

De exemplu, pentru un unghi de comandă al redresorului $\alpha = 60^\circ$, valoarea medie a tensiunii redresate obținute este:

$$U_{d60^\circ} = \frac{3\sqrt{6}}{\pi} \cdot 220 \cdot \cos 60^\circ = 257,29 \text{ V}. \quad (5.36)$$

Frecvența tensiunii de la ieșirea inverterului trebuie să fie, pentru $U_s/f = \text{const.}$

$$f_{60^\circ} = \frac{257,29}{4,4} = 58,475 \text{ Hz}. \quad (5.37)$$

Am arătat, în paragrafele 3.2.1 și 3.2.2, că în cazul utilizării microprocesoarelor la comanda redresoarelor se pot genera unghiuri de comandă între 0° și 180° (0–10 ms) în trepte de $0,18^\circ$ (10 μ s), în cazul cel mai defavorabil. O treaptă de variație corespunde incrementării (decrementării) cu o unitate a numărului, pe 16 biți, ce corespunde unghiului de comandă α .

Acceptînd că unghiul de comandă $\alpha = 60^\circ$ este modificat cu o singură treaptă, tensiunea de la ieșirea redresorului devine:

$$U_{d60,18^\circ} = \frac{3\sqrt{6}}{\pi} \cdot 220 \cdot \cos 60,18^\circ = 255,89 \text{ V}. \quad (5.38)$$

Frecvența tensiunii sintetizate de inverter trebuie și ea modificată la valoarea :

$$f_{60,18^\circ} = \frac{255,89}{4,4} = 58,156 \text{ Hz.} \quad (5.39)$$

Deci, modificând conținutul contorului pentru α (registrul pereche DE, în acest caz) cu o unitate, se obține o variație a tensiunii de ieșire cu :

$$\Delta U = U_{d60^\circ} - U_{d60,18^\circ} = 1,4 \text{ V.} \quad (5.40)$$

și este necesară o variație a frecvenței :

$$\Delta f = f_{60^\circ} - f_{60,18^\circ} = 0,319 \text{ Hz.} \quad (5.41)$$

Ținând seama de relația (5.35), variația frecvenței tensiunii la ieșire este :

$$\Delta T = \frac{0,319}{(58,475)^2} = 93,293 \cdot 10^{-6} \text{ s.} \quad (5.42)$$

În paragraful 5.4.1 am arătat că intervalul de timp între două impulsuri de comandă aplicate tiristoarelor inverterului este contorizat în trepte de $8 \mu\text{s}$ și că perioada tensiunii sintetizate la ieșire se modifică în trepte de minimum $6 \cdot 8 \mu\text{s} = 48 \mu\text{s}$ (6 tiristoare comandate).

În aceste condiții, pentru exemplul detaliat mai sus, o variație ΔT de valoare (5.42) se poate scrie :

$$\Delta T = 93,293 \cdot 10^{-6} = 11,66 \cdot 8 \cdot 10^{-6} \simeq 6 \cdot 2 \cdot 8 \cdot 10^{-6} \text{ s.} \quad (5.43)$$

Calculul exact pentru valoarea lui ΔT ne conduce la un rezultat similar :

$$\begin{aligned} \Delta T &= \frac{1}{f_{60,18^\circ}} - \frac{1}{f_{60^\circ}} = \frac{1}{58,156} - \frac{1}{58,475} = 93,8 \cdot 10^{-6} = \\ &= 11,7 \cdot 8 \cdot 10^{-6} \simeq 6 \cdot 2 \cdot 8 \cdot 10^{-6} \text{ s} \end{aligned} \quad (5.44)$$

În concluzie, în jurul unghiului de comandă $\alpha = 60^\circ$ pentru redresor, o modificare cu o treaptă ($\Delta\alpha = 0,18^\circ$) a acestui unghi implică modificarea cu 2 unități (2 trepte) a numărului ce reprezintă T_{INV} — durata dintre două impulsuri de comandă pentru inverter.

Efectuând calcule similare și pentru alte valori ale unghiului de comandă se obțin, spre exemplu, următoarele rezultate :

Pentru $\alpha = 75^\circ$, o modificare a unghiului de comandă cu $\Delta\alpha = 0,18^\circ$ implică o variație cu $\Delta T \simeq 8 \cdot 6 \cdot 8 \cdot 10^{-6}$ s. Deci, în jurul unghiului de comandă de 75° , variația cu un increment a numărului ce reprezintă α , implică variația cu 8 incremenți a numărului ce reprezintă T_{INV} .

Pentru $\alpha = 85^\circ$, o variație $\Delta\alpha = 0,18^\circ$ implică $\Delta T = 77 \cdot 6 \cdot 8 \cdot 10^{-6}$ s. Deci, în jurul unghiului de comandă de 85° , variația cu o unitate a numărului ce reprezintă α implică variația cu 77 de incremenți de numărare pentru durata dintre două impulsuri de comandă pentru inverter.

Avînd în vedere această interdependență neuniformă între variația unghiului de comandă al redresorului și frecvența de comandă a inverterului, cel mai comod este să se lucreze tabelar. Se stabilesc, de exemplu, în plaja de reglare pentru unghiuri între 60° și 85° , 70 de trepte de variație și se calculează duratele T_{INV} corespunzătoare. Perechile de valori obținute se depun tabelar în memoria nevolatilă a sistemului.

Stabilirea de 70 de trepte între 60° și 85° (exact $85,2^\circ$) presupune acceptarea unei variații pentru α în trepte de $0,36^\circ$, adică modificarea numărului care îl reprezintă în program, în trepte de cîte două unități. O asemenea rezoluție este foarte bună pentru toate aplicațiile practice, nefiind necesar lucrul, principial posibil, cu o rezoluție mai bună.

5.6. REGLAREA TURAȚIEI MOTOARELOR ASINCRONE PE BAZA PRINCIPIULUI ORIENTĂRII DUPĂ CÎMP

5.6.1. PRINCIPIUL REGLAJULUI „ORIENTARE DUPĂ CÎMP”

Prin analogie cu posibilitățile de reglare a turației la motoarele de curent continuu, s-a dezvoltat în ultimii ani o metodă, considerată foarte eficientă, de reglare a turației și la motoarele asincrone [36, 59, 40, 7].

La mașinile de curent continuu, cuplul este determinat fără influența reciprocă a curenților din indus și excitație, iar direcțiile curenților și fluxului inductor sînt fixe în spațiu. Modificînd unul dintre cei doi curenți, se realizează modificarea turației.

Mașina asincronă este alimentată cu un sistem trifazat simetric de tensiuni, rezultînd corespunzător cîte un sistem de trei curenți și trei fluxuri. Descrierea funcționării implică, deci, operarea cu trei mărimi de fază, care, datorită conexiunii electrice și magnetice dintre faze, nu sînt independente una de alta.

Cele trei mărimi de fază pot fi reprezentate, însă printr-un singur fazor, cunoscut în literatura de specialitate sub denumirea de fazor spațial, vector Park sau vector spațial.

În compunerea fazorului spațial intervin valorile instantanee ale mărimilor de fază. Prin urmare, acest fazor indică variația în timp și defazajul în timp pentru cele trei mărimi, sub formă vectorială. În plus, vectorul Park indică și defazajul în spațiu dintre cele trei mărimi de fază, defazaj determinat de dispunerea constructivă a înfășurărilor de fază.

Fazorul spațial este, deci, un vector „trifazat” care indică atît variația în timp a mărimilor de fază, cît și variația în spațiu a rezultantei acestora.

În această reprezentare, cei trei curenți de fază, de exemplu, se caracterizează printr-o singură mărime, fazorul spațial de curent. Cînd cei trei curenți de fază formează un sistem sinusoidal trifazat, cîmpul magnetic resultant este un cîmp învîrtitor în spațiu, iar fazorul spațial va fi învîrtitor cu viteza unghiulară constantă și egală cu pulsația curenților.

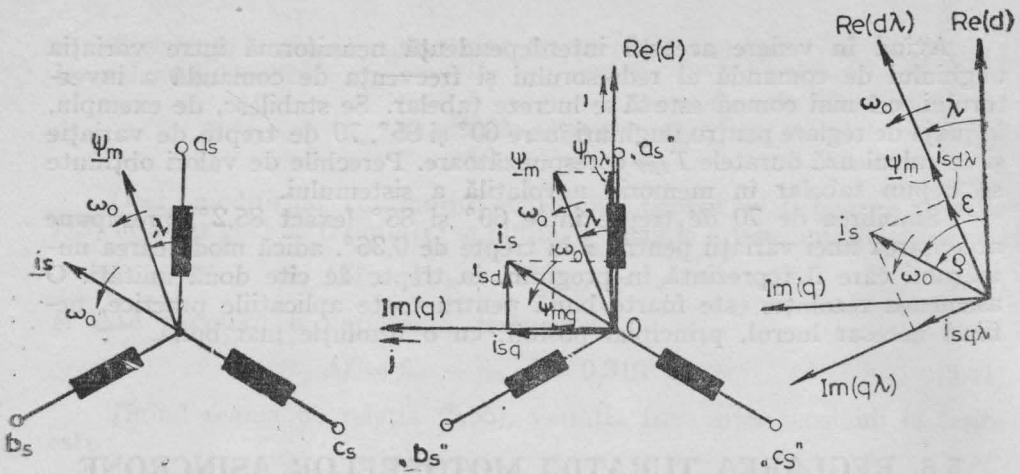


Fig. 5.17. Reprezentarea fazorilor spațiali ai fluxului învîrtitor și curentului la mașina asincronă trifazată.

Fig. 5.18. Modul de alegere al sistemului de referință pentru fazorii spațiali:

- a) după axe fixe;
- b) după axe roti toare.

La o mașină asincronă, alimentată cu un sistem trifazat simetric de tensiuni, toți cei trei fazori spațiali — al cîmpului învîrtitor, al curentilor din înfășurările statorice, respectiv al tensiunilor aplicate — se rotesc cu viteza sincronă (pulsăția tensiunii de alimentare). În figura 5.17 sînt reprezentați fazorii spațiali flux și curent pentru un motor trifazat.

La mașina asincronă de construcție obișnuită, avînd simetria cilindrică, toți vectorii spațiali sînt în planul perpendicular pe axa arborelui mașinii [36].

Se știe că, într-un plan, un vector este determinat de două mărimi, de obicei cele două componente după axe ortogonale (sau de modulul și argumentul său).

Rezultă, deci, un prim avantaj al studiului mașinii asincrone cu ajutorul fazorilor spațiali: reducerea celor trei mărimi de fază la două mărimi și deci utilizarea a numai două ecuații în loc de trei. Mașina trifazată se reduce la una bifazată cu înfășurări ortogonale.

În cazul în care sistemul sinusoidal trifazat de alimentare al mașinii nu este echilibrat, la schimbarea de variabile corespunzătoare reducerii mașinii trifazate la una bifazată intervine, matematic, un al treilea element — componenta homopolară, care nu este o proiecție după o a treia direcție în spațiu, fiind numai o mărime de calcul. Deci, fazorul spațial își păstrează caracterul plan și cînd există componentă homopolară. În [59, 36] se arată că în majoritatea aplicațiilor sau nu apar componente homopolare sau influența lor este neglijabilă.

Caracterul plan al fazorului spațial permite alegerea unui plan (complex) de referință. În figura 5.18, a și b se prezintă două moduri de alegere ale sistemului de referință.

În figura 5.18, a axa reală a planului de referință s-a ales după axa uneia dintre cele trei înfășurări statorice ale mașinii. Ca urmare, cele două componente ortogonale ale fazorilor spațiali sînt variabile, sinusoidal, în timp.

Dacă planul de referință este ales rotitor (figura 5.18, *b*), cu axa reală „orientată” după fazorul spațial al fluxului învîrtitor din întrefier, toți fazorii spațiali au o poziție relativă staționară. Ei pot fi considerați, deci, mărimi de curent continuu, deși toate mărimile de fază sînt variabile.

Alegînd un sistem de referință rotitor, cu viteză unghiulară egală cu pulsația tensiunii de alimentare, apare posibilitatea de a realiza un reglaj, cu ajutorul unor mărimi de curent continuu. Pentru separarea celor două bucle de reglaj, de flux și de turație, fazorul curentului statoric — staționar față de fazorul spațial al fluxului învîrtitor — se descompune după două direcții determinate de acesta din urmă, ca în figura 5.18, *b*. Cele două componente sînt, deci, ortogonale și independente una de alta, rolul lor fiind similar cu cel al curenților din indus și din excitație de la motorul de curent continuu. Efectuînd un reglaj asupra uneia sau alteia dintre componentele $i_{sq\lambda}$ sau $i_{sd\lambda}$ (figura 5.18, *b*) ale fazorului spațial al curentului, se poate realiza un reglaj de turație pentru mașina asincronă cu mărimi de curent continuu, similar cu metodele de reglaj de la motorul de curent continuu.

În ceea ce privește cele două componente ale fazorului spațial curent: $i_{sd\lambda}$, fiind orientată după direcția fluxului de magnetizare, reprezintă componenta reactivă — curentul de flux, iar $i_{sq\lambda}$, fiind perpendiculară pe această direcție, reprezintă componenta activă — curentul de cuplu.

Mărimile de curent continuu, utilizate în reglajul pe principiul orientării după cîmp, nu sînt măsurabile direct. Ele trebuie calculate din mărimile reale (naturale) ale motorului, adică din mărimile variabile și măsurabile. Acest calcul presupune, în esență, un șir de schimbări de variabile, ce implică și rezolvarea unor sisteme de ecuații algebrice și trigonometrice. Tratarea matriceală a calculului aduce simplificări în modul de lucru, dar timpul necesar efectuării acestora rămîne suficient de lung, în unele cazuri implicînd și utilizarea mai multor calculatoare [36].

5.6.2. SISTEME DE REGLARE A TURAȚIEI MOTOARELOR ASINCRONE, PE BAZA PRINCIPIULUI ORIENTĂRII DUPĂ CÎMP

În figura 5.19 se prezintă o schemă bloc detaliată a unui sistem de reglare a unui motor asincron, cu rotorul în scurt circuit [35]. Motorul este alimentat printr-un convertor de frecvență cu circuit intermediar cu caracter de sursă de curent — învertor de curent [36]. Etajele de putere sînt realizate după scheme cunoscute.

Reglajul în buclă închisă implică utilizarea următoarelor traductoare: de turație (TG), de curenți de fază și tensiuni de linie la motor, respectiv de curent la alimentarea redresorului la rețea.

Sistemul realizează reglajul fluxului de magnetizare ψ_m , prin componenta reactivă $i_{sd\lambda}$ a fazorului de curent și reglajul turației (n) prin componenta activă $i_{sq\lambda}$ a acestui fazor (figura 5.18, *b*), deci reglajul se efectuează în curent continuu.

Mărimile de prescriere sînt ψ_u^* și n^* .

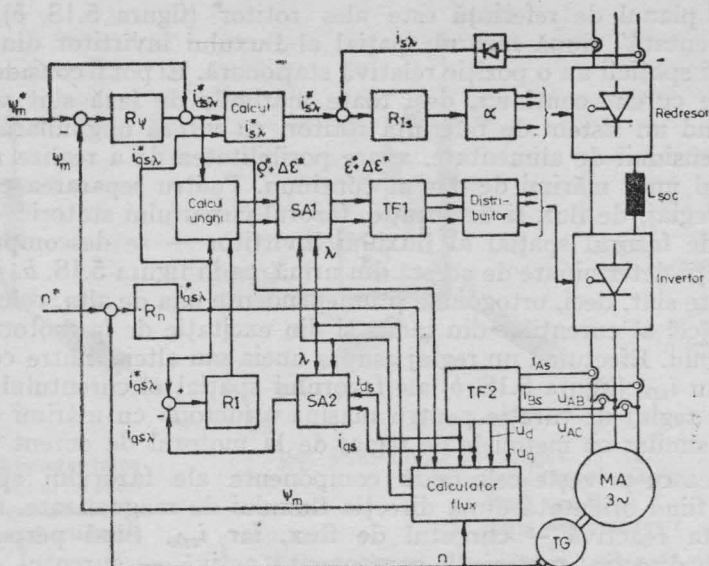


Fig. 5.19. Schema bloc a sistemului de reglare pentru motor asincron pe baza principiului „orientării după cîmp”, varianta indirectă.

Realizarea reglajului pe principiul orientării după cîmp necesită calculul componentelor fazorului spațial al fluxului, din mărimile măsurate la bornele motorului. De aceea, metoda utilizată este o variantă indirectă de reglare [59].

Transformatorul de fază $TF2$ conține detectoare pentru curenții și tensiunile statorice, cu ajutorul cărora se determină componentele fazorilor spațiali, i_s și u_s în sistemul de coordonate fixe d, q (figura 5.18), adică ale sistemului de axe bifazat.

Din mărimile bifazate $u_{sd}, u_{sq}, i_{sd}, i_{sq}$ și turația măsurată se determină, pe baza ecuațiilor de funcționare ale motorului, componentele ψ_{md} și ψ_{mq} ale fluxului, în raport cu sistemul de axe fixe (figura 5.18), respectiv fluxul de magnetizare ψ_m și faza acestuia λ . Operațiile sînt efectuate de calculatorul de flux.

Determinarea unghiului λ înseamnă, de fapt, determinarea poziției fazorului spațial flux, adică a sistemului de referință rotitor.

Reglatoarele de flux și de turație R_ψ și R_n operează cu mărimi staționare. Mărimile de ieșire ale acestor reglatoare sînt: pentru R_ψ , $i_{s\lambda}^*$ — curentul de flux, iar pentru R_n , $i_{sq\lambda}^*$ — curentul de cuplu, calculate pe seama mărimilor de prescriere și de reacție. Valorile $i_{s\lambda}^*$ și $i_{sq\lambda}^*$ reprezintă pentru circuitele următoare mărimile de referință.

În primul rînd, din $i_{s\lambda}^*$ și $i_{sq\lambda}^*$ se calculează amplitudinea necesară $i_{s\lambda}^*$ a fazorului spațial curent, referința pentru regulatorul proporțional integrator R_{i_s} . Cu acest regulator se asigură reglarea curentului statoric la valoarea necesară, prin comanda redresorului cu un unghi α corespunzător.

În al doilea rând, principiul orientării după câmp impune și implementarea unei bucle de reglare a unghiului de defazaj ρ (figura 5.18, b) dintre fazorul curentului statoric i_s și fazorul fluxului de magnetizare ψ_m . Cu această buclă de reglare se menține unghiul ρ la valoarea necesară.

În acest scop, calculatorul de unghi determină unghiul ρ^* , de referință, între fazorii curent și flux, efectuînd operațiile:

$$i_{s\lambda}^* = \sqrt{i_{sd\lambda}^{*2} + i_{sq\lambda}^{*2}} \quad (5.45)$$

$$\cos \rho^* = \frac{i_{sd\lambda}^*}{i_{s\lambda}^*}, \quad \sin \rho^* = \frac{i_{sq\lambda}^*}{i_{s\lambda}^*} \quad (5.46)$$

Blocul SA1 calculează unghiul ε^* , defazajul necesar al fazorului de curent în raport cu sistemul de referință fix (figura 5.18):

$$\cos \varepsilon^* = \cos(\rho^* + \lambda) = \cos \rho^* \cos \lambda - \sin \rho^* \sin \lambda,$$

$$\sin \varepsilon^* = \sin(\rho^* + \lambda) = \sin \rho^* \cos \lambda + \cos \rho^* \sin \lambda.$$

Valoarea calculată pentru ε^* determină, în final, momentul declanșării impulsurilor de comandă pentru tiristoarele inverterului, adică al momentului comutării curentului de alimentare al motorului de pe o înfășurare, pe alta. În acest fel se obține o poziționare corectată a fazorului spațial curent, conform unghiului de defazaj (ρ), dorit, între fazorii flux și curent.

În figura 5.20 [27] se prezintă o schemă bloc în care principiul orientării după câmp este implementat într-o variantă directă de reglare [59] în sensul că poziția fluxului învîrtitor nu se calculează din măsurile de fază, ci se utilizează două sonde Hall, amplasate în întrefierul mașinii, pentru măsurarea directă a unghiului fluxului.

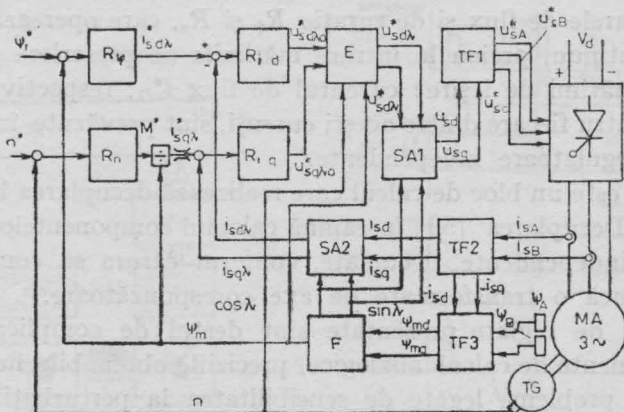


Fig. 5.20. Schema bloc a sistemului de reglare pentru motor asincron pe baza principiului „orientării după câmp”, varianta directă.

Metoda implică dificultăți tehnologice, legate de amplasarea cu precizie extrem de ridicată a traductoarelor Hall în întrefier dar conferă avantaje în ceea ce privește exactitatea cu care se determină poziția fluxului.

Etajul de putere utilizat pentru alimentarea motorului este un invertor, care permite reglarea simultană a amplitudinii și frecvenței tensiunii sintetizate la ieșirea sa [36].

Scopul urmărit de schema din figura 5.20 este o reglare independentă a fluxului, respectiv a curentului de magnetizare $i_{sd\lambda}$ și a turației, respectiv curentului de cuplu $i_{sq\lambda}$. Pentru a se obține valorile reale ale acestor mărimi, în mașină se utilizează mărimi măsurate cu traductoare (i_{sA} , i_{sB} , ψ_A și ψ_B) și blocuri de calcul.

Blocurile pentru transformările de fază T_F fac o schimbare a mărimilor trifazate în mărimi bifazate, raportate la sistemul de axe fixe d, q (figura 5.18), conform relațiilor:

$$\begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} u_{sA} \\ u_{sB} \end{bmatrix} \quad (5.47)$$

Trecerea de la mărimile bifazate u_{sd} , u_{sq} la cele raportate la sistemul de axe rotitor se face cu o schimbare de axe (bloc SA), conform relațiilor:

$$\begin{bmatrix} u_{sd\lambda} \\ u_{sq\lambda} \end{bmatrix} = \begin{bmatrix} \cos \lambda & \sin \lambda \\ -\sin \lambda & \cos \lambda \end{bmatrix} \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}. \quad (5.48)$$

Valorile funcțiilor $\sin \lambda$, $\cos \lambda$, necesare, se calculează din componentele $\psi_{m\lambda}$, $\psi_{m\lambda}$ ale fluxului de magnetizare în blocul F . Componentele $\psi_{m\lambda}$, $\psi_{m\lambda}$ ale fluxului sînt calculate în blocul $TF3$ din valorile măsurate ψ_A , ψ_B .

Reglatoarele de flux și de turație R_ψ și R_n , care operează cu mărimi de curent continuu, preiau la intrare mărimile de prescriere ψ^* , respectiv n^* și au ca mărimi de ieșire: curentul de flux $i_{sd\lambda}^*$, respectiv curentul de cuplu $i_{sq\lambda}^*$. Pentru fiecare dintre acești curenți, sînt prevăzute în continuare, în schemă, reglatoare independente.

Blocul E este un bloc de calcul care realizează decuplarea buclei de flux și de cuplu. Decuplarea [59] înseamnă calculul componentelor $u_{sd\lambda}^*$ și $u_{sq\lambda}^*$, componente independente, decuplate, conform cărora se comandă invertorul, după încă o transformare de axe corespunzătoare.

Sistemele de reglare prezentate sînt destul de complicate. Dacă se utilizează elemente de calcul analogice, preciziile obținabile nu sînt foarte mari și apar probleme legate de sensibilitatea la perturbații și de fiabilitate scăzută.]

Utilizarea microprocesoarelor deschide noi perspective în extinderea acestei metode în acționările cu motoare asincrone.

5.6.3. SISTEM DE REGLARE PENTRU MOTOR ASINCRON COMANDAT CU MICROPROCESORUL 8080 [36]

Sistemul este realizat în Laboratorul de acționări electrice al Facultății de electrotehnică din Cluj-Napoca și reprezintă una din primele încercări din țara noastră cu privire la comanda motorului asincron, utilizând microprocesoare. Metoda de reglare utilizată derivă din principiul orientării după câmp și folosește sistemul din figura 5.21.

Circuitul de putere constă dintr-un redresor în punte trifazată complet comandată și un invertor de curent, construite după scheme tradiționale. Microsistemul, o structură clasică dezvoltată în jurul microprocesorului 8080, generează mărimile de comandă I^* (amplitudinea fazorului spațial curent), care determină unghiul de comandă α al redresorului și un tren de impulsuri, cu frecvența variabilă, cu care se sintetizează secvența de impulsuri de comandă pentru invertor.

Redresorul este direct comandat de către microsistem. Se utilizează un convertor numeric analogic și un circuit analogic de sintetizare, la unghiuri de comandă variabile, a impulsurilor de amorsare pentru tiristoare.

Invertorul de curent necesită o secvență de 6 impulsuri de comandă pentru sintetizarea unei perioade complete a curentului trifazat de ieșire. Secvența de impulsuri de comandă este realizată, în afara microsistemului, cu un circuit distribuitor de impulsuri, din trenul de impulsuri cu frecvență variabilă, generat de microsistem (fig. 5.22).

La fiecare impuls de comandă, curentul constant furnizat de invertor este comutat de pe o fază pe alta a motorului. Ca urmare, frecvența de comutare este $1/6$ din cea a impulsurilor de comandă, iar fazorul spațial al curenților din stator ocupă în spațiu 6 poziții fixe între care se comută (vezi figura 5.23). Fazorul spațial curent execută rotația discretă, dintr-o

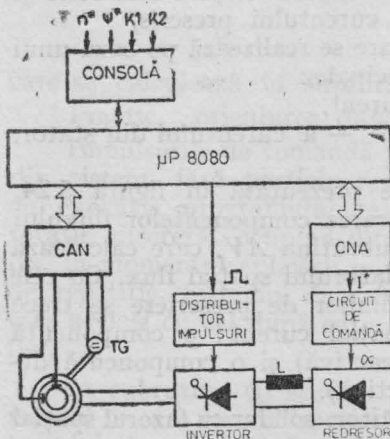


Fig. 5.21. Sistem de reglare pentru motor asincron realizat pe baza principiului „orientării după câmp” cu microprocesor de comandă.

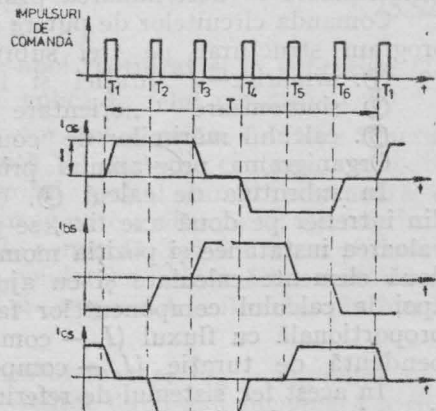


Fig. 5.22. Secvența de impulsuri de comandă pentru tiristoare și variația curenților de fază la alimentarea motorului asincron cu invertor de curent.

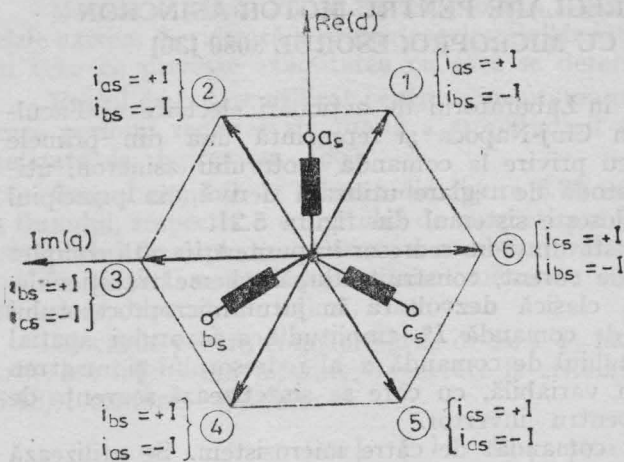


Fig. 5.23. Diagrama trajectoriei fazorului curent din stator.

tă a fluxului din întrefier. Se măsoară două componente ale fluxului, după două axe fixe; acestea sînt preluate de microprocesor, împreună cu valoarea curentă a turației, măsurată cu tahogenerator. O subrutină dedicată AV (analizor de vector) calculează amplitudinea și poziția curentă a fazorului spațial flux. Ecuațiile trigonometrice care intervin se rezolvă tabelar. Precizia de determinare a poziției fluxului, la $f = 50$ Hz, este $4^{\circ}17'$ grade electrice, pentru o durată de $246 \mu s$ de execuție a subrutinei.

Mărimile de prescriere se introduc în microsistem de la consola unui DAF. Aceste mărimi sînt: turația prescrisă n^* , valoarea prescrisă a fluxului Φ_m^* și două constante $K1$ și $K2$, ce țin de parametrii mașinii asincrone și sînt utilizate la determinarea prin calcul a curentului prescris.

Comanda circuitelor de putere din acționare se realizează pe baza unui program structurat pe trei subrutine principale:

- ①. urmărire—a turației și fluxului curent,
- ②. sincronizare — „orientare după cîmp” — a curentului din stator,
- ③. calculul mărimilor de comandă.

Organigrama programului principal este prezentată în figura 5.24.

În subrutina de calcul ③, după măsurarea componentelor fluxului din întrefier pe două axe fixe, se parcurge subrutina AV , care calculează valoarea instantanee și poziția momentană a fazorului spațial flux. Cu cele două elemente calculate și cu ajutorul mărimilor de prescriere se trece apoi la calculul componentelor fazorului spațial curent: o componentă proporțională cu fluxul (I_r — componenta reactivă) și o componentă dependentă de turație (I_a — componenta activă).

În acest fel, sistemul de referință este rotitor, solidar cu fazorul spațial flux, în conformitate cu principiul orientării după cîmp.

Valoarea instantanee obținută pentru fazorul curent este folosită pentru sintetizarea unghiului de comandă pentru redresor, dar nu se utilizează în comanda invertorului și deci a curentului propriu-zis din motor. S-a

poziție fixă în alta, exact în momentul comenzii.

Trenul de impulsuri, din care se sintetizează secvența de comandă pentru invertor, este generat de microsistem, utilizînd timerul din componența sa. Timerul, realizat cu circuitul 8253, este încărcat, pornit și oprit prin programul principal de comandă.

În sistemul prezentat, principiul orientării după cîmp este aplicat într-o variantă care folosește măsurarea direc-

utilizat acest mod de lucru, deoarece proiecțiile fazorului spațial curent după două axe perpendiculare, solidare cu fazorul flux sînt variabile în timp, în acest caz. Într-adevăr, după cum s-a arătat anterior și după cum rezultă din figura 5.22 și figura 5.23, fazorul spațial curent are 6 poziții fixe în spațiu, între care comută discret, în timp, iar fluxul rezultat din întrefier se rotește continuu.

Din acest motiv, principiul orientării după cîmp se aplică, în exemplul descris, într-un mod mai deosebit.

În vederea obținerii cuplului necesar la turația impusă motorului, inverterul se comandă în așa fel încît să se asigure permanent un defazaj minim între fazorul spațial flux și fazorul spațial curent, sincronizarea fiind efectuată, prin program, după poziția fazorului flux.

Argumentul fazorului spațial curent :

$$\nu = \arctg \frac{I_a}{I_r}, \quad (5.52)$$

care se calculează în subrutina ③ este apoi reutilizat și în subrutina ②.

Practic, „orientarea după cîmp” se aplică în felul următor :

Impulsurile de comandă pentru inverter sînt generate de către timerul din sistem, fără participarea permanentă a unității centrale, ocupată cu execuția programului principal. Fazorul spațial curent efectuează o rotație discretă, ocupînd direcția axei imaginare a sistemului de referință fix, în momentul aplicării impulsurilor de comandă 3 și 6 dintr-o perioadă de repetiție. Sistemul de referință fix, solidar cu înfășurările motorului, este același cu cel după care se măsoară și componentele fluxului din întrefier.

În subrutina ① se determină momentul în care $\Phi_m = \Phi_q$, adică momentul în care fazorul spațial flux este orientat după axa imaginară a sistemului de referință fix, și se oprește generarea impulsurilor de comandă pentru inverter. Această oprire are loc întodeauna după ce impulsul 2 (sau 5) dintr-o perioadă a fost deja generat.

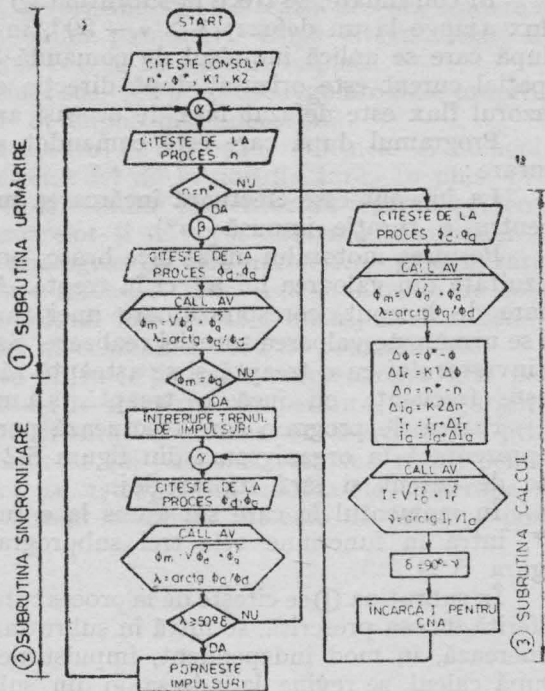


Fig. 5.24. Organigrama programului principal.

În continuare, se trece în subrutina ② și se așteaptă pînă cînd fazorul flux ajunge la un defazaj $\lambda \geq \nu - 30^\circ$, în raport cu axa imaginară fixă, după care se aplică impulsul de comandă 3 (sau 6). În acest fel, fazorul spațial curent este orientat după direcția axei imaginare, numai după ce fazorul flux este defazat față de aceeași axă, cu cel puțin $\nu - 30^\circ$.

Programul după care este comandat sistemul are următoarea desfășurare:

La început este efectuată încărcarea mărimilor de prescriere necesare pentru o turație impusă (n^*).

Pornirea motorului nu se face brusc, prin comandă cu frecvența direct rezultată din valoarea lui n^* , ci în trepte. Astfel, se încarcă în timer o valoare de început, corespunzătoare unei turații mici, se pornește timerul și se urmărește valoarea turației realizate. Se mărește frecvența de comandă a inverterului cu o treaptă și se așteaptă răspunsul motorului, apoi se mărește frecvența cu încă o treaptă ș.a.m.d.

Partea de program care efectuează pornirea în trepte (egale) nu este reprezentată în organigrama din figura 5.24. Se realizează o pornire fără șoc de curent și fără suprareglaj.

În momentul în care s-a ajuns la o turație apropiată de cea impusă n^* , intră în funcțiune cele trei subprograme principale reprezentate în figura 5.24.

În subrutina ① se citește de la proces turația curentă n . Dacă aceasta este diferită de cea prescrisă, se intră în subrutina de calcul. Între timp, timerul generează, în mod independent, impulsurile de comandă pentru inverter. După calcul, se revine la adresa ③ din subrutina de urmărire. Se verifică dacă fazorul flux are orientarea după axa fixă, imaginară. Dacă nu, se citește din nou turația curentă (ori de cîte ori turația curentă diferă de cea prescrisă, se trece prin subrutina ③ pentru calculul noilor mărimi de comandă — dintre care, pentru inverter, este important ν). Dacă fazorul flux este orientat după axa imaginară, se oprește generarea impulsurilor către inverter și se așteaptă pînă cînd fazorul flux ajunge la un defazaj de $\nu - 30^\circ$ față de această axă. Apoi se comandă, din nou, pornirea impulsurilor. Este momentul în care se aplică la inverter impulsul 3 (sau 6) dintr-o perioadă de repetiție.

În figura 5.25 se înfățișează modul în care sînt generate impulsurile de comandă pentru inverter. Intervalele notate $\Delta T' > \Delta T$, includ întîrzierile introduse de subrutina ② (sincronizare) și reprezintă intervale de așteptare, pentru a se asigura defazajul minim ($\nu - 30^\circ$) între fazorul flux și fazorul curent.

Dacă, la un moment dat, apare o creștere în treaptă a cuplului de sarcină, după un interval de timp determinat de caracteristicile inerțiale ale siste-

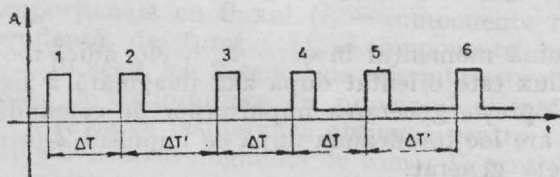


Fig. 5.25. Secvența de impulsuri sincronizate pentru comanda inverterului.

mului, turația începe să scadă. Subrutina ① sesizează această scădere și apelează subrutina de calcul a noilor parametri. După câteva cicluri de urmărire și sincronizare, sistemul revine la turația impusă.

Cu toate că în aplicația de mai sus, [36], lipsesc regulatoarele propriuzise, sistemul descris prezintă o importanță deosebită.

În primul rînd, aplicația are meritul de a fi pus problemele în domeniu și de a fi o primă încercare de acest fel de la noi din țară. În plus, prin aplicarea procedului și utilizarea sistemului prezentat au ieșit în evidență avantajele utilizării microprocesoarelor și în acest domeniu al acționărilor electrice: înlocuirea sistemelor analogice cu cele numerice, generalizarea sistemelor de reglare prin mijloace software, reducerea numărului de componente de sistem și preluarea rolului lor de către microprocesor, scăderea volumului și prețului de cost al echipamentelor de comandă etc.

Cercetările viitoare, corelate cu apariția microprocesoarelor mai performante și cu dezvoltarea pe linie de software vor permite, cu siguranță, răspîndirea pe scară largă a reglării turației motoarelor asincrone pe principiul orientării după cîmp, considerat [59] cel mai complex reglaj pentru motoare asincrone. În acest fel, se așteaptă ca motoarele asincrone să devină competitive cu cele de curent continuu, în ceea ce privește posibilitățile de reglare.

UTILIZAREA MICROPROCESOARELOR ÎN COMANDA ACȚIONĂRILOR CU MOTOARE PAS CU PAS

6.1. PROBLEME SPECIFICE CU PRIVIRE LA MOTORUL PAS CU PAS [43]

Motorul electric pas cu pas (*MPP*) este un convertor electromagnetice, care realizează conversia unui tren de impulsuri de comandă, aplicate înfășurărilor de fază ale motorului, într-o mișcare de rotație ce constă din deplasări unghiulare discrete, de mărime egală. Unghiul de rotație pe care îl execută rotorul, la aplicarea unui impuls de comandă, reprezintă unghiul de pas al motorului. Numărul pașilor efectuați trebuie să corespundă, în cazul unei funcționări corecte, cu numărul impulsurilor de comandă.

Majoritatea *MPP* sînt bidirecționale și permit o accelerare, oprire și reversare rapidă, fără pierderi de pași, dacă sînt comandate cu impulsuri a căror frecvență de repetiție este inferioară frecvenței limită, corespunzătoare unui cuplu rezistent și unui moment de inerție dat. Pentru extinderea funcționării motoarelor pas cu pas la viteze mai mari decît viteza corespunzătoare frecvenței limită, este necesară o accelerare prin creșterea treptată a frecvenței impulsurilor de comandă.

MPP sînt utilizate, în special, în aplicațiile unde se dorește realizarea unei mișcări incrementale, folosind sisteme de comandă numerică.

Utilizarea *MPP* conferă, în principiu, următoarele avantaje:

- a. asigură univocitatea conversiei număr de impulsuri — deplasare și pot fi utilizate în circuit deschis;
- b. gamă largă de frecvență de comandă;
- c. precizie de poziționare și rezoluție mare;
- d. permit porniri, opriri, reversări, fără pierderi de pași;

- e. memorează poziția;
- f. sînt compatibile cu comanda numerică.

Dezavantajele utilizării *MPP* sînt:

- a. unghi de pas, deci increment de rotație, de valoare fixă pentru un motor dat;
- b. viteză de rotație relativ scăzută;
- c. putere dezvoltată la arbore de valoare redusă;
- d. randament energetic scăzut;
- e. necesită o schemă de comandă adaptabilă la tipul constructiv;
- f. necesită o schemă de comandă complexă pentru asigurarea unei funcționări la viteze mari.

Avînd în vedere aceste dezavantaje, cu deosebire cele de la punctele c, d, și f, la care se adaugă și dificultățile tehnologice care survin în fabricarea *MPP*, se constată, după o perioadă de extindere a utilizării lor în anii '70, o scădere a interesului față de utilizarea acestui tip de motoare, în ultimul timp.

6.1.1. MODALITĂȚI DE COMANDĂ A FAZELOR *MPP*

Principial, fazele *MPP* se alimentează cu impulsuri de curent de amplitudine constantă, care se comută de pe o fază pe alta, în ritmul unui tact de comandă.

În [43] se prezintă următoarele modalități de comandă (alimentare) a fazelor *MPP*:

1. comandă potențială sau prin impulsuri;
2. comandă monopolară sau bipolară;
3. comandă simetrică (simplă sau dublă) sau nesimetrică.

Un circuit concret de alimentare a fazelor unui *MPP* asigură realizarea simultană a 3. — câte una de tip 1, una de tip 2 și una de tip 3 — dintre cele 6 modalități de comandă prezentate mai sus.

Comanda potențială se referă la durata alimentării unei faze, în raport cu durata între două tacturi de comandă $1/f$, f fiind frecvența de comandă a *MPP*. Dacă durata alimentării unei faze este cel puțin egală cu $1/f$, atunci comanda este potențială. În acest caz, durata aplicării tensiunii pe o fază variază invers proporțional cu frecvența. Dacă durata alimentării este constantă și întotdeauna mai mică decît $1/f$, atunci este vorba de o comandă prin impulsuri.

Deși majoritatea schemelor de comutație a fazelor *MPP* sînt cu comandă potențială, s-a consacrat denumirea de comandă prin impulsuri, probabil datorită formei de variație în timp a tensiunilor de alimentare.

Comanda monopolară asigură sens unic al curentului prin fiecare înfășurare a motorului, în tot timpul funcționării. Comanda bipolară determină ca, pe parcursul unui ciclu complet de comenzi aplicate înfășurărilor, sensul curentului prin fiecare înfășurare să se schimbe succesiv.

Comanda este simetrică, dacă, la un moment dat, sînt alimentate un număr egal de faze și nesimetrică, dacă numărul fazelor alimentate simultan se schimbă alternativ la fiecare tact de comandă.

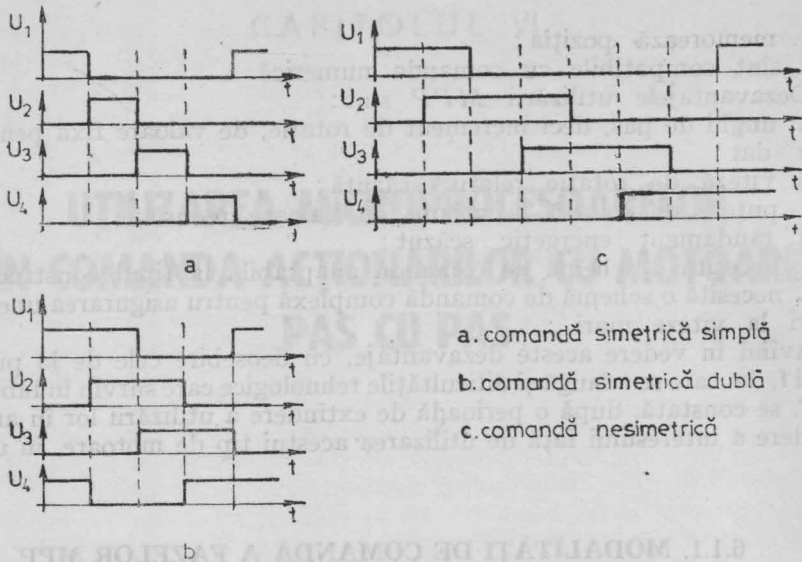


Fig. 6.1. Comanda monopolară a *MPP* cu patru faze.

Comanda simetrică este numită simplă sau dublă, după cum sînt alimentate cîte una sau cîte două înfășurări simultan. Prin utilizarea comenzii duble, pentru un *MPP* cu patru faze se mărește atît cuplul dezvoltat, cît și amortizarea, dar cresc consumul de putere și temperatura motorului.

În figura 6.1 sînt reprezentate formele de variație în timp a tensiunilor de alimentare pe faze, pentru comanda monopolară a unui *MPP* cu patru faze, pentru a exemplifica diferite moduri de comandă.

Pentru inversarea sensului de rotație, impulsurile reprezentate în secvențele din figura 6.1. se aplică în ordine inversă: $U_4-U_3-U_2-U_1$ etc.

În figura 6.2 este reprezentat modul de comandă bipolară simetrică dublă a unui *MPP* cu patru faze. Comanda bipolară necesită o schemă de comandă complexă, care să asigure circulația curenților din înfășurări, în ambele sensuri.

6.1.2. MODELUL MATEMATIC ÎN COORDONATE REALE AL *MPP* [43]

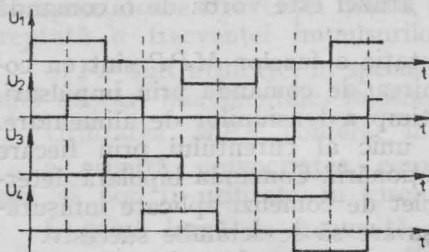


Fig. 6.2. Comanda bipolară dublă a unui *MPP* cu patru faze.

Pentru descrierea matematică a unui *MPP*, se va considera un model generalizat. În caz general, un *MPP*, fiind un motor sincron modificat, poate fi considerat ca o mașină electrică cu întrefier dințat și cu m înfășurări de comandă (faze) distribuite pe stator sau rotor, înfășurări ce sînt alimentate prin impulsuri.

În afară de acestea, circuitul magnetic mai poate conține magneti permanenți sau înfășurări de excitație alimentate cu curent constant.

Deoarece orice magnet permanent poate fi înlocuit cu o înfășurare de excitație fictivă, în caz general, un motor pas cu pas cu m faze poate fi considerat ca avînd $m + 1$ circuite electrice.

Prin analogie cu principiile din mecanica analitică, *MPP* poate fi considerat ca un sistem unitar de conversie electromecanic, caracterizat prin $m + 2$ coordonate generalizate (sau grade de libertate) m curenți în fazele *MPP*, curentul din înfășurarea de excitație și unghiul de rotație. Pentru descrierea funcționării *MPP* sînt necesare tot atîtea ecuații de echilibru electric și ecuația de mișcare (echilibrul dinamic al cuplurilor). Acest sistem de ecuații formează modelul polifazat în coordonate reale al *MPP*.

Sistemul de ecuații poate fi scris sub forma matriceală astfel :

$$[i_K][R] + \frac{d}{dt} [\psi_K] = [u_K],$$

$$J \frac{d^2\theta_m}{dt^2} + B \frac{d\theta_m}{dt} + M_r = M_e, \quad (6.1)$$

în care :

$[i_K]$ este matricea coloană a curenților cu $m + 1$ linii, ultima fiind curentul de excitație real sau echivalent ;

$[R]$ — matricea diagonală a rezistențelor de dimensiuni $(m + 1) \times (m + 1)$;

$[\psi_K]$ — matricea coloană a fluxurilor totale, care se consideră aici funcții de curenții din circuitele respective ; are $m + 1$ linii, ultima corespunzînd fluxului de excitație.

J — momentul de inerție total redus al arborele *MPP*, considerat constant ;

B — coeficientul frecării vîscoase, constant ;

M_r — cuplul rezistent redus la arbore ;

M_e — cuplul electromecanic dezvoltat de motor ;

θ_m — unghiul mecanic instantaneu al rotorului.

Ultima linie din ecuația matriceală corespunde circuitului de excitație. Dacă excitația este un magnet permanent, ea se reduce la $I_M = \text{const.}$ și sistemul va conține numai $m + 1$ ecuații diferențiale. Același lucru se poate spune și pentru cazul excitației separate, alimentată de la o sursă de curent constant.

Sistemul ecuații (6.1), scris pentru cazul general, poate fi particularizat pentru sistemele de poziționare cu diferite tipuri constructive de *MPP*. Această particularizare se referă și la deducerea expresiei cuplului electromagnetic, pentru fiecare tip constructiv de motor, precum și la posibilitatea de a lua sau nu în considerare saturația circuitului magnetic. Se știe că *MPP*, ca orice dispozitiv de conversie electromagnetice, avînd în componența sa fizică un circuit magnetic, prezintă o caracteristică neliniară, dato-

rită saturației circuitului magnetic, curenților turbionari, precum și efectului histeresis. Această caracteristică neliniară are un efect important asupra performanțelor *MPP*. În stabilirea modelului matematic al *MPP* sînt foarte dificil de modelat aceste neliniarități; chiar și în cazurile cînd se obțin modele matematice foarte exacte pentru *MPP*, ele sînt dificil de utilizat practic.

6.1.3. SCHEMA BLOC DE COMANDĂ A MPP

Performanțele unui *MPP* sînt strîns legate de tipul schemei sale de comandă. Astfel, amortizarea, frecvența maximă de mers, cuplul dinamic maxim etc, precum și randamentul sau puterea disipată depind în mare măsură de schema de alimentare utilizată.

În general, comanda unui *MPP* se face printr-un circuit electronic de putere, care alimentează secvențial înfășurările motorului. Sensul de distribuție al alimentării, ca și tipul secvenței (simetrică, nesimetrică, număr de ieșiri), precum și frecvența de comutare a înfășurărilor sînt realizate prin prelucrare logică secvențială cu circuite integrate, în timp ce curentul în fazele motorului este asigurat printr-un etaj de comutație statică forțată.

Rolul circuitului de comandă este de a prelua semnale standard *TTL* (impulsuri de comandă și nivel de sens), a le distribui secvențial într-un număr de combinații egal cu numărul fazelor *MPP*, a le amplifica și apoi de a le aplica fazelor motorului. În acest fel, schema de comandă a unui *MPP* îmbină semnale discrete standard cu mărimi electrice de putere.

Schema bloc generală este prezentată în figura 6.3 și reprezintă comanda în circuit deschis (fără reacție) a unui *MPP*.

Impulsurile de comandă pot proveni de la un generator propriu, reglabil manual, sau de la un calculator numeric de proces.

6.1.3.1. BLOCUL CONTACTOARELOR STATICE

Semnalele de la distribuitor trebuie amplificate pentru a transmite puterea adecvată motorului. În mod uzual, sînt folosite etaje cu tranzistoare de putere cuplate direct sau tiristoare, pentru curenți mai mari. Se deosebesc contactoare statice pentru alimentare unipolară și contactoare statice pentru alimentare bipolară, de obicei patru tranzistoare de putere în punte.

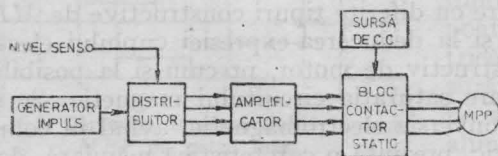
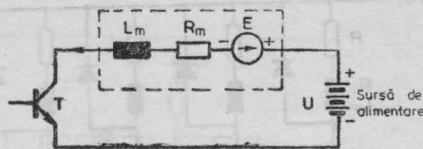


Fig. 6.3. Schema bloc de comandă și alimentare a *MPP*.

Fig. 6.4. Schema echivalentă a înfășurării unei faze.



Înfășurarea unui *MPP* este o sarcină rezistiv-inductivă, cu inductivitate constantă sau periodic variabilă cu unghiul de rotație. În plus, mișcarea rotorului generează o tensiune electromotoare E , cu sens invers față de tensiunea de alimentare. O schemă echivalentă a înfășurării unei faze a *MPP* este reprezentată în figura 6.4.

Rezistența înfășurării variază, datorită toleranțelor de fabricație, cu $\pm 10\%$. În plus, rezistența înfășurării crește în timpul funcționării motorului, pentru că cele mai multe *MPP* sînt destinate a funcționa la o temperatură de ordinul a 100°C , deci practic fierbinți. Se acceptă o variație a rezistenței înfășurării de 25% cu temperatura. Totodată, cele mai multe *MPP* sînt destinate a lucra în condiții de saturație a circuitului magnetic, mai ales cînd viteza unghiulară este scăzută, deoarece, în cazurile practice, amplitudinea impulsului de curent prin înfășurare se modifică (crește cu scăderea vitezei de rotație). Rezultă, implicit, o variație a inductanței înfășurării, funcție de curentul prin înfășurare.

Circuitele de alimentare sînt de curenți mari, care utilizează tranzistoare de putere capabile să suporte vîrfuri de putere. Tranzistoarele se aleg luînd în considerare cele mai grele condiții de lucru. De asemenea, înfășurările reprezintă sarcini inductive care sînt conectate și deconectate și, deci, tranzistoarele de putere trebuie protejate contra supratensiunilor inductive tranzitorii.

Curentul prin faza motorului are expresia :

$$i(t) = \frac{U}{R_m} \left(1 - e^{-\frac{t}{T_m}} \right), \quad (6.2)$$

unde: $T_m = \frac{L_m}{R_m}$ este constanta de timp a fazei, L_m , R_m — inductivitatea proprie medie, respectiv rezistența fazei, iar U — tensiunea continuă de alimentare. Pentru ca *MPP* să răspundă la frecvențe cît mai mari, este necesar ca timpul de stabilire al curentului la valoarea $\frac{U}{R_m}$ să fie cît mai mic. Blocul contactoarelor statice trebuie să asigure și forțarea pantei curentului prin fazele motorului.

Tehnicile cunoscute de micșorare a timpului de creștere a curentului prin faze [43] sînt :

Ă. Forțarea prin rezistență serie — se bazează pe micșorarea constantei de timp T_m , prin inserierea de rezistențe cu fazele motorului; constanta de timp se reduce de la $\frac{L_m}{R_m}$ la $\frac{L_m}{R_m + R}$, iar tensiunea de alimentare crește de la $U = R_m I$ la $U_1 = (R_m + R)I$, I fiind curentul nominal

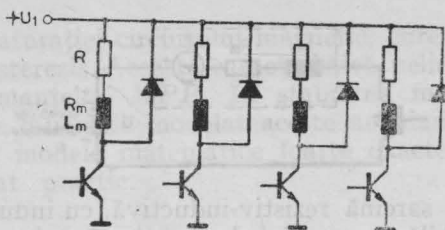


Fig. 6.5. Forțarea cu rezistență serie în cazul *MPP* cu patru faze.

al fazei (figura 6.5). Dezavantajele metodei constau în randamentul slab, datorat surplusului de putere disipată pe rezistența exterioră și accentuarea oscilațiilor în răspunsul *MPP*, precum și a fenomenului de rezonanță mecanică a motorului.

B. Forțarea prin tensiune — în care forțarea pantei curentului se face prin folosirea unei tensiuni înalte pe durata creșterii curentului în înfășurare și comutarea, apoi, la o tensiune scăzută, corespunzătoare curentului nominal. În figura 6.6 se prezintă o variantă de schemă, realizată după acest principiu, utilizată la comanda *MPP* 1,8/0,2 — Electromotor Timișoara. Caracteristicile unui astfel de motor sint: unghi de pas 1,8°, cuplu maxim de mers 0,2 N.m, curent nominal pe fază 1 A, patru faze. Cu schema de tipul din figura 6.6 s-a realizat frecvența maximă de avans 1,2 kHz, cu 0,12 N · m cuplu de mers [12].

În momentul aplicării impulsului de comandă, tranzistorul T_3 se deschide, iar circuitul monostabil *CDB* 4121 basculează în starea nestabilă ($Q = 1$), Monostabilul comandă deschiderea tranzistorului T_2 , pe durata $\tau = R_1 C_1$ și, ca urmare, înfășurarea *MPP* este alimentată cu tensiunea înaltă U_1 . După durata τ , monostabilul revine în starea stabilă ($Q = 0$) și T_2 se blochează, iar înfășurarea rămâne alimentată de la tensiunea scăzută U_2 , pînă la sfîrșitul duratei impulsului de comandă.

Pentru un motor de putere mai mare, tip *MPP* 1,8/3 cu 4 faze și caracteristicile: unghi de pas 1,8°, cuplu maxim de mers 3N · m, curent nominal pe fază 5,3 A, s-a conceput o schemă de forțare în tensiune cu tiristoare [12]. Pentru fiecare dintre cele patru faze, schema conține cîte un tiristor T_P , care comută tensiunea înaltă un timp scurt. Tiristoarele principale se sting cu ajutorul unui singur tiristor de stingere și cîte un condensator C_S aferent (figura 6.7).

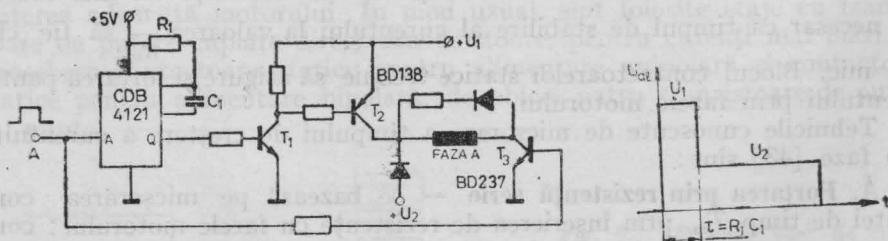


Fig. 6.6. Forțarea prin tensiune înaltă.

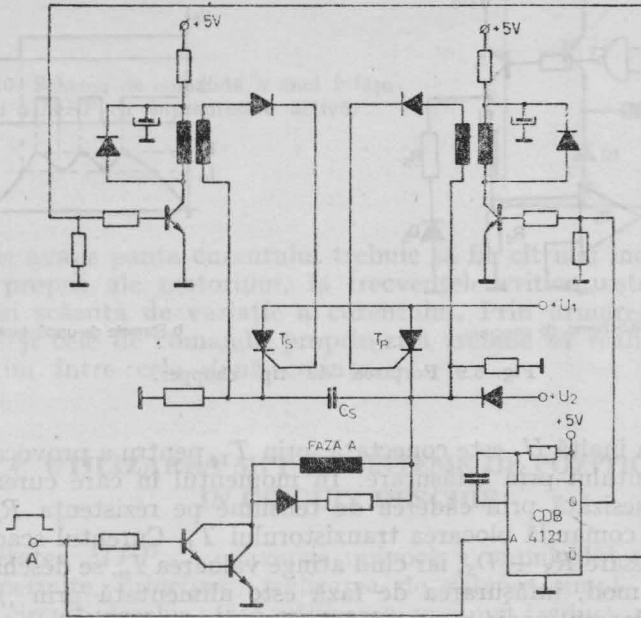


Fig. 6.7. Schema de comandă pentru o fază a unui MPP de curent mare, cu forțare prin tensiune.

Din exemplele prezentate se observă complexitatea schemelor de comandă implicate de utilizarea acestei metode. În plus, apare dezavantajul utilizării de elemente active (tranzistoare, tiristoare, diode) costisitoare, de curenți și tensiuni mari. Solicitarea acestor componente implică și o fiabilitate mai scăzută a schemelor.

C. Forțarea cu rezistență și condensator. — prin care panta de creștere a curentului prin fază este mărită, datorită unui curent suplimentar, de descărcare a unui condensator, prin înfășurarea motorului (figura 6.8).

Deși performanțele și randamentul sînt mai bune decît în cazul forțării cu rezistență, metoda prezintă dezavantajul unei forme oscilante a curentului prin fază și accentuarea fenomenelor oscilatorii ce apar în comportarea MPP.

D. Forțarea tip chopper — reprezintă o metodă care folosește o reacție de curent, prin care curentul de fază este silit să oscileze, cu frecvență ridicată, în jurul valorii nominale. În figura 6.9 se prezintă schema de principiu și formele de undă explicative ale metodei. O schemă de acest tip a fost utilizată pentru comanda unui motor MPP 1,8/3 — Electromotor Timișoara și a permis realizarea unui cuplu de $2,4 \text{ N} \cdot \text{m}$ la o frecvență de avans de 600 pași/s [1].

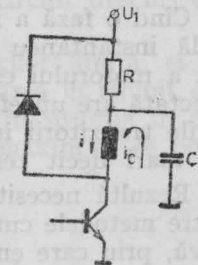


Fig. 6.8. Schema de principiu pentru forțarea prin rezistență și condensator pentru o fază a MPP.

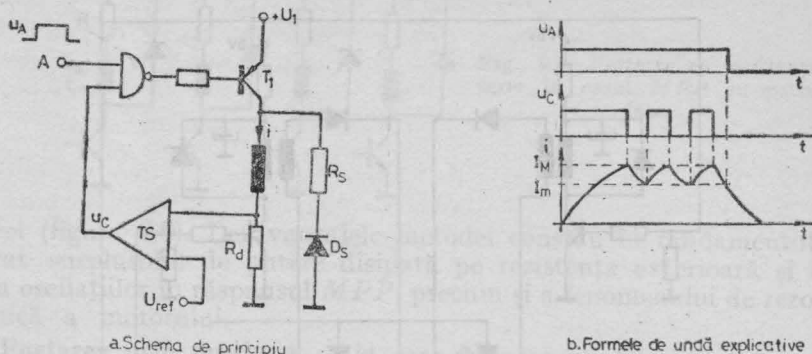


Fig. 6.9. Forțarea de tip chopper.

Tensiunea înaltă U_1 este conectată, prin T_1 , pentru a provoca o creștere rapidă a curentului prin înfășurare. În momentul în care curentul atinge valoarea I_M , sesizată prin căderea de tensiune pe rezistența R_d , triggerul Schmitt (TS) comandă blocarea tranzistorului T_1 . Curentul scade prin circuitul de supresare $R_s - D_s$, iar când atinge valoarea I_m , se deschide din nou T_1 . În acest mod, înfășurarea de fază este alimentată prin „choppare”, astfel încât valoarea medie a curentului să se situeze la nivel constant. Triggerul Schmitt este realizat cu histereza $U_{pMax} - U_{pmin}$ constantă, dar valoarea tensiunilor de prag U_{pMax} și U_{pmin} se poate modifica simultan prin U_{ref} . Dacă U_{ref} se stabilește cu ajutorul unui potențiometr semireglabil, aceeași schemă de comandă se poate utiliza la mai multe tipuri de motoare, cu diferite valori ale curentului nominal prin înfășurare.

Metoda forțării tip chopper se dovedește a fi cea mai eficientă metodă de alimentare în curent constant, dar schema devine foarte complexă pentru cazul alimentării unui motor cu patru faze în secvență dublă și bipolară.

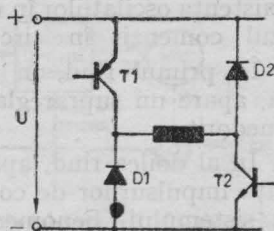
6.1.3.2. METODE DE SUPRESARE

Când o fază a motorului este deconectată, curentul de fază nu poate să scadă instantaneu din cauza inductivității înfășurării. Când următoarea fază a motorului este comandată, scăderea curentului în faza tocmai deconectată are un efect de încetinire a răspunsului motorului. În plus, tensiunile tranzitorii inductive apărute, pot atinge valori de 50—100 de ori mai mari decât tensiunea de alimentare.

Rezultă necesitatea de supresare rapidă a acestor tensiuni tranzitorii. Dintre metodele cunoscute [43], cea mai eficientă se dovedește supresarea activă, prin care energia înmagazinată în înfășurare este transmisă înapoi sursei de alimentare (figura 6.10).

Practica a dovedit că panta de variație a curentului are o influență importantă asupra vitezei maxime de avans a motorului, dar și asupra intensității de manifestare a fenomenului de rezonanță. Dacă pentru vi-

Fig. 6.10. Schema de comandă a unei înfășurări a *MPP* cu suprapresare activă.



teze mari de avans panta curentului trebuie să fie cât mai mare, atenuarea oscilațiilor proprii ale motorului, la frecvențele critice, este corelată cu o pantă mai scăzută de variație a curentului. Prin urmare, schemele de suprapresare și cele de comandă propriu-zisă trebuie să realizeze un compromis optim între cele două cerințe.

6.1.4. UTILIZAREA *MPP* ÎN SISTEME DE POZIȚIONARE ÎN CIRCUIT DESCHIS

Proprietatea *MPP* de conversie univocă a impulsului electric în pas unghiular permite, principial, realizarea de sisteme simple de reglare a poziției în circuit deschis, fără utilizarea vreunui traductor în structura lor. Pentru aceasta este însă necesară păstrarea proprietății de conversie într-un domeniu larg de condiții de lucru (viteză, cuplu).

În general, pentru un *MPP* dat, performanțele care se pot obține (viteză maximă, cuplu maxim, stabilitatea mișcării) depinde în mare măsură de schema de comandă. În cazul unui *MPP* comandat în circuit deschis, din lipsa unei bucle de reacție, nu există nici un mijloc de a sesiza dacă motorul a pierdut un impuls sau dacă viteza are un caracter oscilant. Foarte adesea, dacă frecvența impulsurilor de comandă este prea ridicată, *MPP* pierde total sincronismul cu impulsurile, oprindu-se pur și simplu. Frecvența impulsurilor de comandă pînă la care *MPP* funcționează sincron (fără pierderi de pași) scade cu creșterea sarcinii dinamice și statice.

O altă problemă legată de funcționarea *MPP* în circuit deschis este cea a fenomenelor oscilatorii care apar.

Dacă motorul este bine ales, pe baza caracteristicilor limită, și are asigurată o bună amortizare, el poate satisface poziționarea în circuit deschis în multe aplicații. Rezultate foarte bune se obțin în situația în care cuplul rezistent este mult mai mic (sub $1/3$) decît cuplul maxim ce se poate atinge cu un motor dat.

6.1.4.1. AMORTIZAREA *MPP*

Este cunoscut [43] că, în cazul comenzii *MPP*, răspunsul la un impuls singular prezintă oscilații importante. Frecvența acestor oscilații depinde de parametrii întregului sistem electromecanic comandat.

Existența oscilațiilor în răspunsul *MPP* are cel puțin două inconveniente în cazul comenzi în circuit deschis:

a. În primul rând, în procesul de poziționare, la oprirea în poziția impusă, apare un suprareglaj de valoare mare (până la 80%), ceea ce este total nedorit;

b. În al doilea rând, apare fenomenul de rezonanță în comutație, dacă frecvența impulsurilor de comandă este egală cu frecvența proprie de oscilație a sistemului. Fenomenul de rezonanță se manifestă prin pierderea sincronismului, mișcarea rotorului devenind dezordonată, complet scăpată de sub acțiunea impulsurilor de comandă. În aplicații practice se întâlnesc situații în care rezonanța se manifestă nu numai la o singură frecvență de comandă. Valoarea frecvenței la care se manifestă rezonanța depinde de mulți factori, dintre care, nu în ultimul rând, de valoarea curentului din înfășurare și a pantei de variație a acestuia.

Din motivele enunțate, apare necesitatea utilizării unor tehnici de amortizare a motoarelor pas cu pas, în afara îndeplinirii condiției elementare de a asigura permanent o sarcină statică la arborele motorului. Mijloacele de amortizare utilizate se împart în [43]:

1. Amortizoare mecanice externe.
2. Amortizoare electrice.
3. Amortizoare electronice.
4. Amortizare prin proiectarea *MPP*.

Fiecare dintre familiile de mijloace, indicate mai sus, aduc îmbunătățiri, dar reprezintă acțiuni asupra efectului și nu asupra cauzei fenomenului în sine.

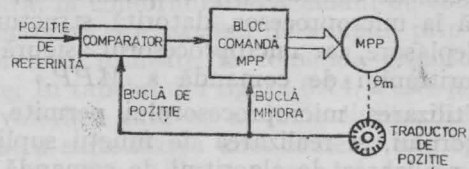
O metodă prin care se ocolește fenomenul de rezonanță este aceea în care circuitul de comandă evită generarea de impulsuri cu frecvențe de repetiție apropiate de cele de rezonanță.

6.1.5. UTILIZAREA *MPP* ÎN SISTEME DE POZIȚIONARE ÎN CIRCUIT ÎNCHIS

Caracteristica sistemelor cu *MPP* în circuit închis este că *MPP* este cuplat cu un traductor de poziție sau de deplasare, de care este legat și funcțional. Traductorul constituie, de obicei, elementul de măsurare a unghiului de rotație, de la care pleacă reacția negativă de poziție. Mărima de referință sau poziția impusă poate fi analogică (tensiune) sau numerică (număr de impulsuri).

La acționările cu *MPP* este posibil ca de la traductorul de poziție să se închidă și o altă buclă, subordonată celei de reacție negativă, care are rolul de a furniza impulsurile de comandă pentru *MPP* în sincronism cu unghiul de rotație. Această buclă nu constituie o reacție negativă și nici nu este inclusă într-o buclă de reglare propriu-zisă, ci are rolul, particular, de a aplica un nou impuls de comandă, exact în momentul când pasul precedent a fost complet efectuat. Bucla este numită în literatura de specia-

Fig. 6.11. Schema bloc de utilizare a *MPP* în sistem cu circuit închis.



litate [43] buclă minoră, iar prin utilizarea ei, traductorul de poziție este, de fapt, generatorul impulsurilor de comandă pentru *MPP*.

Schema bloc pentru un sistem de poziționare cu *MPP* în circuit închis este prezentată în figura 6.11.

Avantajele ce se obțin prin includerea unui *MPP* într-un circuit de reglare închis sînt:

- viteze superioare de mers,
- stabilitate mai bună în raport cu variațiile parametrilor poziționării (în special sarcina).
- un mers mai liniștit, fără oscilații.

Toate aceste avantaje nu pot fi obținute simultan cu orice schemă de poziționare în circuit închis. În plus, complexitatea și, implicit, prețul de cost al unor astfel de sisteme este din ce în ce mai ridicat, pe măsura perfecționărilor, necesare, aduse.

Și în cazul utilizării comenzii în circuit închis, pot apare erori în poziționare, deoarece mișcarea rotorului este controlabilă numai în unități întregi de pași, pe cînd în interiorul pasului, între două poziții de echilibru vecine, mișcarea scapă de sub controlul electronic, fiind la discreția sarcinii mecanice. Cauza principală a apariției de astfel de erori este așa-numita deviație a *MPP*, [43] — faptul că rotorul ocupă poziții elastice în raport cu statorul. Pentru eliminarea în cît mai mare măsură a erorilor de deviație, prin controlul poziției la nivelul fracțiunii de pas, se utilizează comanda în regim de micropășire [43] a *MPP*, care implică însă complicarea suplimentară a schemei de comandă.

În concluzie, la vitezele de lucru și cuplurile obținabile, *MPP* își găsește destul de greu justificările utilizării sale practice.

6.2. CIRCUITE DE COMANDĂ PENTRU *MPP* CU MICROPROCESOR

Din punct de vedere istoric, apariția și răspîndirea microprocesoarelor a coincis cu extinderea cercetărilor cu privire la utilizarea motoarelor pas cu pas. Creșterea complexității și varietății echipamentelor cu *MPP* a condus la probleme dificil de rezolvat, în variante electronice tradiționale, și la prețuri de cost din ce în ce mai ridicate.

La sfîrșitul anilor '70 a apărut, ca de la sine înțelesă, utilizarea microprocesoarelor în comanda acțiunilor cu *MPP*. Acest lucru a fost argumentat

de o serie de considerente fundamentale, ca: *MPP* permite interfațarea directă la microprocesor, datorită structurii sale de convertor discret impuls-deplasare, iar microprocesorul asigură realizarea simplă, prin program a algoritmului de comandă a *MPP*.

Utilizarea microprocesorului permite, în afara comenzii propriu-zise a motorului, și realizarea de funcții suplimentare: reglarea numerică directă, realizarea de algoritmi de comandă (comenzi de sens, funcții logice etc.), elaborarea numerică a mărimilor de referință ș.a.

În aplicațiile de poziționare, *MPP* pot fi comandate cu microprocesor în circuit închis sau deschis.

6.2.1. SISTEME DE POZIȚIONARE ÎN CIRCUIT DESCHIS CU MPP COMANDATE DE MICROPROCESOR

În preocupările noastre cu privire la utilizarea microprocesoarelor în acționările electrice, s-au realizat câteva variante de circuite de comandă a *MPP* în circuit deschis [8, 11, 12].

Schema bloc a unui astfel de sistem este cea din figura 6.12.

Aplicat în comanda *MPP* în circuit deschis, microsistemul trebuie să realizeze următoarele funcții de bază:

(F1) — Preluarea mărimilor prescrise — viteză de avans, număr de pași — din exterior (sau din memorie).

(F2) — Sintetizarea duratei necesare a impulsurilor de comandă.

(F3) — Distribuirea impulsurilor de comandă pe fazele *MPP*, printr-unul din porturile de ieșire ale sistemului.

(F4) — Generarea unui număr, prescris, de impulsuri în vederea realizării poziției dorite în deplasare.

Aceste patru funcții pot fi realizate cu un sistem în configurație minimă și cu mijloace exclusiv software. Organigrama bloc a programului ce trebuie executat este prezentată în figura 6.13.

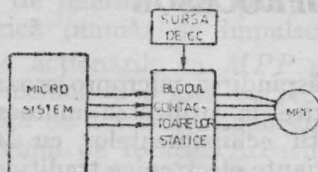


Fig. 6.12. Schema bloc pentru comanda *MPP* în circuit deschis cu microprocesor.

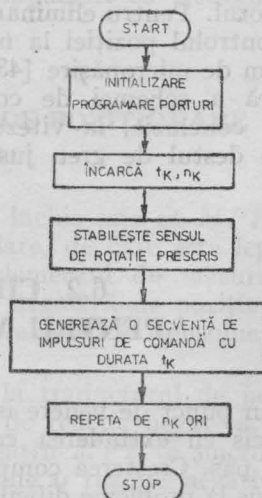


Fig. 6.13. Organigrama programului de comandă pentru *MPP*.

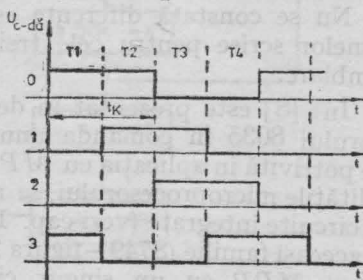
Secvența de impulsuri necesară, în conformitate cu modul de comandă impus *MPP*, se realizează cu microprocesor extrem de simplu, prin poziționarea pe 1 sau 0 logic a unor linii, utilizate la comanda propriu-zisă, ale unuia dintre porturile de ieșire. În tabelul din figura 6.14, *a* este prezentată succesiunea de variație a valorilor care se atribuie liniilor de port 0 ÷ 3, dintr-un port de ieșire, pentru sintetizarea unei comenzi monopolare simetrice duble a unui *MPP* cu patru faze, adică a secvenței de impulsuri necesară, prezentată în figura 6.14, *b*. Cu $T_1 \div T_4$ au fost notate cele 4 intervale de timp dintr-un ciclu de comandă a motorului.

Din tabelul 6.14, *a* se observă că distribuirea impulsurilor de comandă, pentru care în variante tradiționale se folosesc circuite electronice de construcție dedicată-distribuitoare, în cazul utilizării microprocesorului se face prin simpla înscrisere a unui număr în registrul de programare al unui port de ieșire. Numerele (codurile) corespunzătoare secvenței de comandă se păstrează în memoria sistemului și sînt apelate de programul principal, la fiecare comutare a fazelor motorului sau se sintetizează prin rotirea conținutului acumulatorului, poziționat inițial pentru starea T_1 (rotirea se efectuează fără bitul *CY*, cu instrucții de tip *RRC* sau *RLC*).

FAZA	T1	T2	T3	T4	Nr. linie PORT
1	1	1	0	0	0
2	0	1	1	0	1
3	0	0	1	1	2
4	1	0	0	1	3

VALOARE
(HEXA)
PENTRU UN
PORT CU
OPT LINII

09 03 06 0C



a

b

Fig. 6.14. Secvența de stări pentru cele patru linii ale portului de ieșire prin care microsistemul comandă *MPP*.

Programele detaliate pentru comanda *MPP* realizată cu microprocesoarele 8035, 8085 și Z80 sînt prezentate mai jos [11, 8]:

8085	8035	Z80
LHLD 9000H	JMP E0	LD BC, (3000H)
XCHG	E0: ORL P1, #FF	LD DE, (3002H)
LXI H, 9002H	MOV R2, #FA	LD HL, (3004H)
MOV B, M	IN A, P1	LD A, E
INR L	MOV R6, A	RRA
MOV C, M	MOV R0, #00	JR C, E1
INR L	JBO E4	LD L, 07
MOV A, C	MOV R0, #03	E1: PUSH BC
RAR	E4: DIS TCNT I	E2: PUSH HL
JC E1	E3: MOV R1, #FF	LD B, 04
MVI L, 07	E2: DJNZ R1, E1	E4: LD A, (HL)
E1: PUSH D	DJNZ R2, E3	OUT (61H), A

E2: PUSH H	HLT: JMP HLT	PUSH DE
MVI D,04	E1: MOV R4,R0	E3: DEC DE
E4: MOV A,M	MOV R5,#04	LD A,E
OUT 21H	E9: MOV A,R4	OR A,D
PUSH B	MOVP3 A,@A	JR NZ,E3
E3: DCX B	OUTI P2,A	POP DE
MOV A,C	MOV A,R6	INC L
ORA B	STRT T	DJNZ E4
JNZ E3	E5: JTF E6	POP HL
POP B	JMP E5	POP BC
INR L	E6: STOP TCNT	DEC BC
DCR D	INC R4	LD A,C
JNZ E4	DJNZ R5,E9	OR A,B
E5: POP H	JMP E2	JR NZ,E1
POP D		RST 08
DCX D		
MOV A,D		
ORA E		
JNZ E1		
RST 1		

Nu se constată diferențe esențiale în ceea ce privește lungimile programelor scrise pentru cele trei familii de microprocesoare, în limbaj de asamblare.

În [8] este prezentat în detaliu un exemplu de utilizare al microprocesorului 8035 în comanda unui *MPP*. Utilizarea acestui microprocesor este potrivită în aplicația cu *MPP*, pentru că, fără a utiliza la maximum posibilitățile microprocesorului, se realizează o schemă de comandă, cu numai trei circuite integrate (vezi cap. II-figura 2.5). Utilizând un alt microprocesor din aceeași familie (8749—figura 2.7, cap. II) se obține un circuit de comandă pentru *MPP* cu un singur circuit integrat.

În figura 6.15 este reprezentată organigrama detaliată a unui program pentru comanda *MPP* cu microprocesorul 8035. Sistemul₁ utilizat are structura din figura 2.5. Programul generează de un număr n_K (ce poate fi schimbat) de ori secvența de impulsuri necesară pentru comanda monopolară dublă a unui *MPP*. Cazul concret din organigramă folosește $n_K = 256000 = \text{FAFFH}$. Impulsurile de comandă se obțin pe cele patru linii $P_{24} - P_{27}$ ale portului P_2 de ieșire al microprocesorului. Viteza de avans a *MPP*, adică durata t_K a impulsurilor de comandă este preluată de microprocesor, din exterior, prin portul de intrare P_1 (pinii $P_{10} - P_{17}$). Bitul P_{10} este utilizat și ca bit de sens (b_6) al rotației impuse. Contorizarea duratei t_K se face utilizând timerul intern (T).

Cele patru funcții de bază realizate în modul de lucru prezentat pot fi executate și cu mijloace hardware. De exemplu, funcția (F_2) poate fi preluată integral de timerul sistemului, cu utilizarea sistemului de întrepreri. Funcția (F_3) poate fi preluată de un distribuitor de impulsuri extern dedicat și comandat de microsistem. Funcția (F_4) poate fi preluată de un al doilea timer din sistem dacă există, sau de un contor extern atașat.

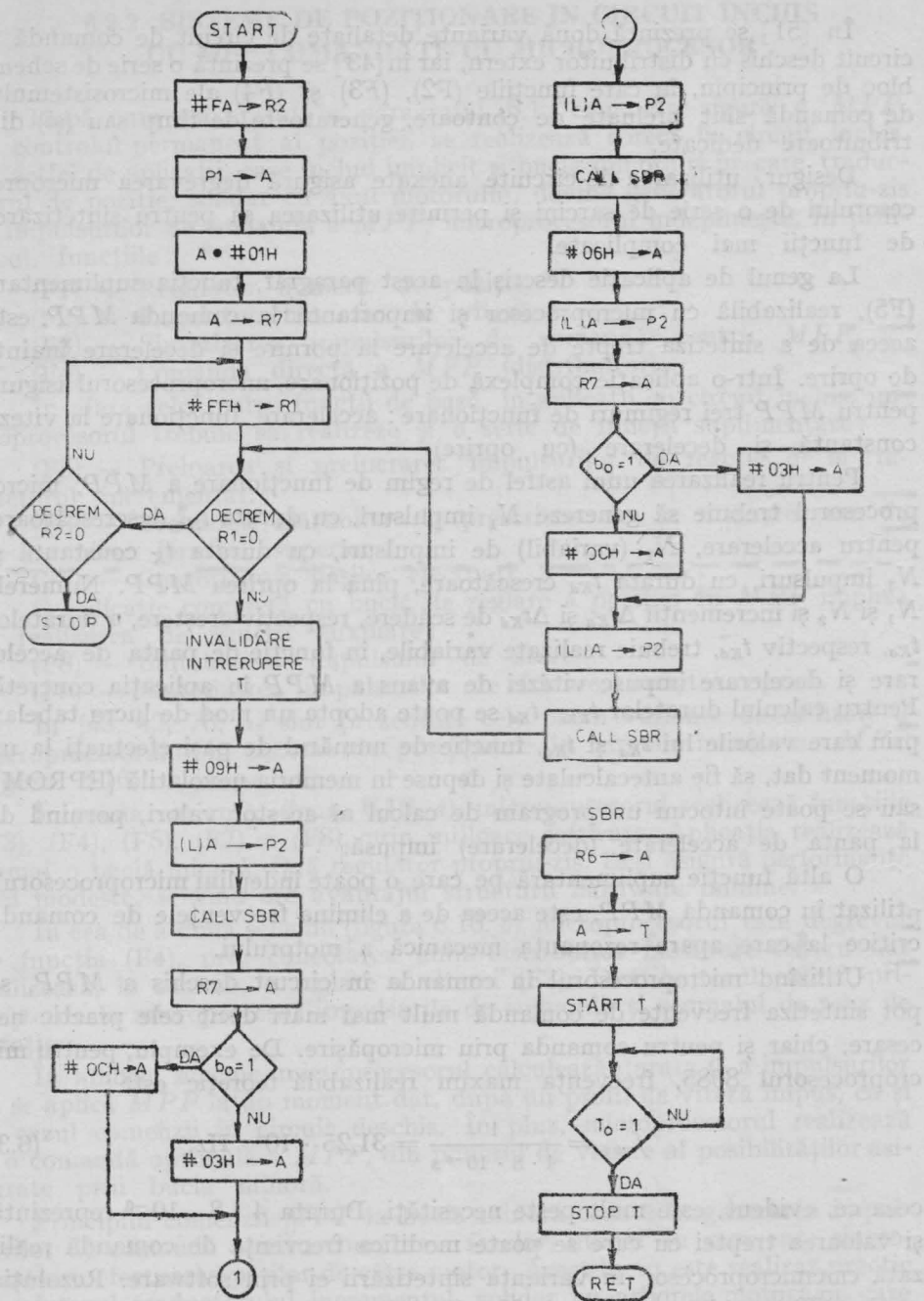


Fig. 6.15. Organigrama programului de comandă a MPP în secvența dublă monopolară.

În [51] se prezintă două variante detaliate de circuit de comandă în circuit deschis cu distribuitor extern, iar în [43] se prezintă o serie de scheme bloc de principiu, în care funcțiile (F2), (F3) și (F4) ale microsistemului de comandă sînt preluate de contoare, generatoare de timp sau (și) distribuitoare dedicate.

Desigur, utilizarea de circuite anexate asigură degrevarea microprocesorului de o serie de sarcini și permite utilizarea sa pentru sintetizarea de funcții mai complicate.

În genul de aplicație descris în acest paragraf, funcția suplimentară (F5), realizabilă cu microprocesor și importantă la comanda MPP, este aceea de a sintetiza trepte de accelerare la pornire și decelerare înainte de oprire. Într-o aplicație complexă de poziționare, microprocesorul asigură pentru MPP trei regimuri de funcționare: accelerare, funcționare la viteză constantă și decelerare (cu oprire).

Pentru realizarea unui astfel de regim de funcționare a MPP, microprocesorul trebuie să genereze N_1 impulsuri, cu durată t_{Ka} descrescătoare, pentru accelerare, N_2 (variabil) de impulsuri, cu durată t_K constantă și N_3 impulsuri, cu durată t_{Kd} crescătoare, pînă la oprirea MPP. Numerele N_1 și N_2 și incrementii Δt_{Ka} și Δt_{Kd} de scădere, respectiv creștere, a duratelor t_{Ka} , respectiv t_{Kd} , trebuie realizate variabile, în funcție de panta de accelerare și decelerare impuse vitezei de avans a MPP în aplicația concretă. Pentru calculul duratelor t_{Ka} , t_{Kd} se poate adopta un mod de lucru tabelar, prin care valorile lui t_{Ka} și t_{Kd} , funcție de numărul de pași efectuați la un moment dat, să fie antecalculate și depuse în memoria nevolatilă (EPROM), sau se poate întocmi un program de calcul al acestor valori, pornind de la panta de accelerare (decelerație) impusă.

O altă funcție suplimentară pe care o poate îndeplini microprocesorul, utilizat în comanda MPP, este aceea de a elimina frecvențele de comandă critice la care apare rezonanța mecanică a motorului.

Utilizînd microprocesorul în comanda în circuit deschis a MPP, se pot sintetiza frecvențe de comandă mult mai mari decît cele practic necesare, chiar și pentru comanda prin micropășire. De exemplu, pentru microprocesorul 8085, frecvența maxim realizabilă teoretic este:

$$f_{max} = \frac{1}{4 \cdot 8 \cdot 10^{-6} \text{ s}} = 31,25 \cdot 10^3 \text{ Hz.} \quad (6.3)$$

ceea ce, evident, este mult peste necesități. Durata $4 \cdot 8 \cdot 10^{-6}$ reprezintă și valoarea treptei cu care se poate modifica frecvența de comandă realizată cu microprocesor în varianta sintetizării ei prin software. Rezoluția minimă obținabilă este, deci, $32 \cdot 10^{-6}$ (la 1 Hz frecvență de comandă) sau mai bună.

6.2.2. SISTEME DE POZIȚIONARE ÎN CIRCUIT ÎNCHIS CU MPP COMANDATE CU MICROPROCESOR

După cum s-a arătat în paragraful 6.1, comanda sigură a *MPP*, cu controlul permanent al poziției, se realizează corect în circuit închis. În astfel de aplicații, care includ implicit și bucla minoră și în care traductorul de poziție, solidar cu axul motorului, devine generatorul propriu-zis al impulsurilor de comandă a *MPP*, microprocesorul îndeplinește, în principal, funcțiile:

- (F1) — Regulator numeric de poziție.
- (F2) — Regulator numeric de viteză.
- (F3) — Sintetizarea impulsurilor de comandă pentru *MPP*.
- (F4) — Comanda directă a *MPP* (distribuitor).

În afara celor patru funcții de bază, în aplicații cu circuit închis, microprocesorul trebuie să realizeze și o serie de funcții suplimentare:

- (F5) — Preluarea și prelucrarea impulsurilor de reacție de la traductorul incremental.
- (F6) — Preluarea mărimilor de prescriere de la utilizator sau de la un nivel ierarhic superior.
- (F7) — Contorizarea pașilor efectuați.

O aplicație completă cu buclă de reglare a poziției cu *MPP* implică și realizarea de funcții auxiliare:

- (F8) — Sintetizarea impulsului de start.
- (F9) — Sintetizarea impulsurilor de frînare, înainte de opriri.

În [43, 42, 76, 5] sînt prezentate cele patru variante de utilizare a microprocesorului în sisteme de poziționare în circuit închis cu *MPP* (figura 6.16, *a*, *b*, *c*, *d*).

În prima variantă (figura 6.16, *a*), microprocesorul realizează funcțiile (F3), (F4), (F5), (F7) și (F8), prin mijloace software. Aplicația realizează numai o buclă minoră, fără regulator propriu-zis. Deși asigură performanțe mai modeste, schema are avantajul structurii hardware minime.

În cea de a doua schemă (figura 6.16, *b*) microprocesorul este degrevat de funcția (F4), prin conectarea unui distribuitor hardware extern dar realizează, în schimb, o reglare a vitezei (F2). Distribuitorul extern primește de la microprocesor impulsurile de comandă și semnalul de sens de rotație.

În ambele scheme, microprocesorul calculează durata t_K a impulsurilor ce se aplică *MPP* la un moment dat, după un profil de viteză impus, ca și în cazul comenzii în circuit deschis. În plus, microprocesorul realizează și o comandă optimală a *MPP*, din punctul de vedere al posibilităților asigurate prin bucla minoră.

Principiul comenzii *MPP* în bucla minoră constă în generarea impulsurilor de comandă pentru comutarea fazelor, numai în momente sincronizate cu efectuarea pașilor de către motor. Acest lucru este realizat practic cu ajutorul traductorului incremental, solidar cu arborele motorului, care furnizează câte un impuls la fiecare pas efectuat.

6.1.1 SISTEM DE POZITIONARE IN CIRCUIT INCHIS
 CU MFT CONTACTOR MICROPROCESOR

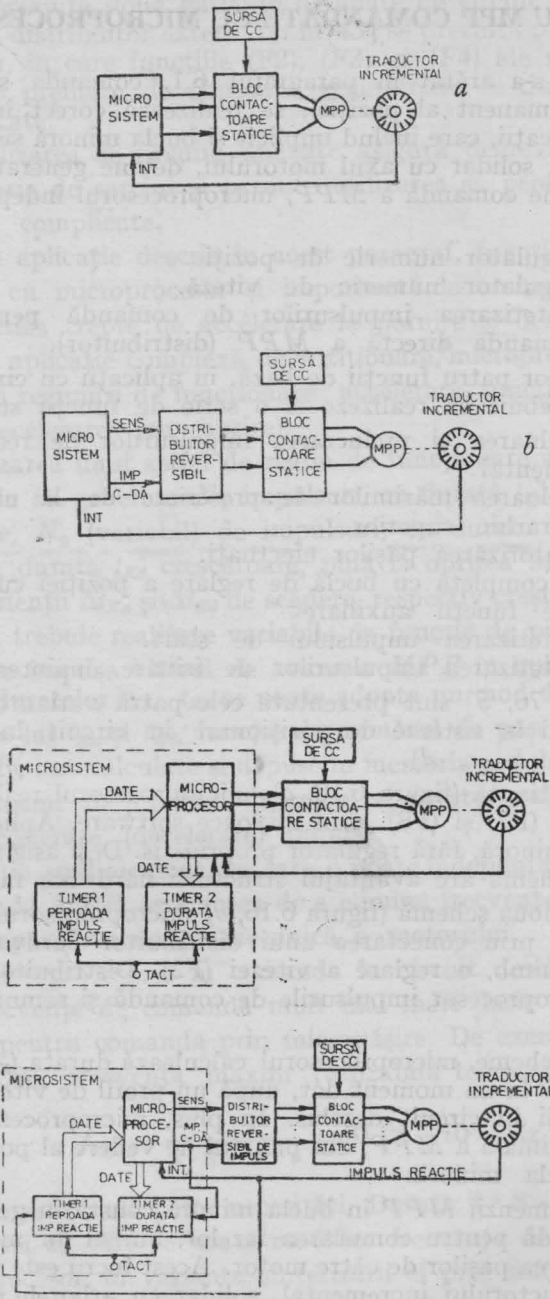


Fig. 6.16. Scheme de comandă a MPP în circuit închis cu microprocesor.

Din caracteristicile *MPP* se cunoaște [43] că accelerația maximă în deplasare se obține dacă impulsurile de comandă sînt aplicate în momente optim decalate, în raport cu momentele trecerilor prin pozițiile de echilibru ale rotorului. Decalajul este optim dacă impulsul de comandă este aplicat pe timpul efectuării unui pas, de către *MPP*, în momentul în care cuplul este maxim.

Pentru realizarea practică a decalajului necesar, discul traductorului incremental se montează mecanic, astfel încît fantele sale să fie aliniată cu pozițiile de echilibru ale rotorului, iar senzorii fotoelectrici se montează în poziții deplasate cu un unghi α_K^{ref} , numit unghi de comutație de referință [43], întrucît comutarea fazelor *MPP* în regim static — la viteze joase — este realizată sincron cu impulsurile furnizate de traductor.

În acest fel, impulsurile de reacție de la traductor sînt decalate cu α_K^{ref} , în raport cu trecerile rotorului prin pozițiile de echilibru. Aceste impulsuri reprezintă tactul pentru impulsurile de comandă a *MPP* și, deci, și comutarea fazelor acestuia se produce la momente decalate în raport cu trecerile prin pozițiile de echilibru.

Pentru un *MPP* cu patru faze, unghiul optim de comutație, corespunzător cuplului maxim, în regim static — la viteze joase, este :

$$\alpha_c = \alpha_K^{opt. st.} = \alpha_K^{ref} = \frac{\pi}{4} \text{ rad.el.} \quad (6.4)$$

iar $\frac{\pi}{2}$ rad. el. este pasul electric al motorului [43].

La creșterea vitezei *MPP* apare necesitatea compensării timpului de creștere și scădere a curentului din faze, ceea ce impune modificarea unghiului de comutație propriu-zisă, pentru a se obține cuplu maxim :

$$\alpha_c = \alpha_K^{opt. din.} = \alpha_K^{opt. st.} - \varphi_K^{max} = \frac{\pi}{4} - \Omega_K^{med} T \cdot Z, \text{ rad.el.} \quad (6.5)$$

unde T — reprezintă o constantă de timp, avînd ordinul de mărime al celei corespunzătoare fazelor *MPP*, Ω_K^{med} — este viteza medie a *MPP* după K pași, iar Z , — este numărul de dinți rotorici [43].

În timpul funcționării pot apare variații de cuplu accidentale. Ca urmare, defazajul φ_K din expresia lui α_c , notat cu φ_K^{real} , poate să se schimbe. Limitele admise pentru φ_K^{real} sînt 0 (cînd $\alpha_c = \alpha_K^{ref} = \frac{\pi}{4}$) și φ_K^{max} , pentru o turație dată a motorului.

$\varphi_K^{real} = 0$ înseamnă că impulsul de comandă este aplicat după momentul în care cuplul este maxim, pe timpul efectuării unui pas, adică se realizează un cuplu de frînare, dar în apropierea cuplului maxim, iar $\varphi_K^{real} = \varphi_K^{max}$ corespunde realizării cuplului maxim (figura 6.17).

Calculul mărimilor φ_K^{real} și φ_K^{max} este efectuat de microprocesor (F5). Înaintea sintetizării impulsului de comandă pentru *MPP*, microprocesorul face o comparație a lui φ_K^{real} cu valorile 0 și φ_K^{max} .

Dacă $\varphi_K^{real} > \varphi_K^{max}$, ceea ce este cauzat de un cuplu, accidental, superior celui maxim, și, deci, survine o accelerare prea mare a motorului, impul-

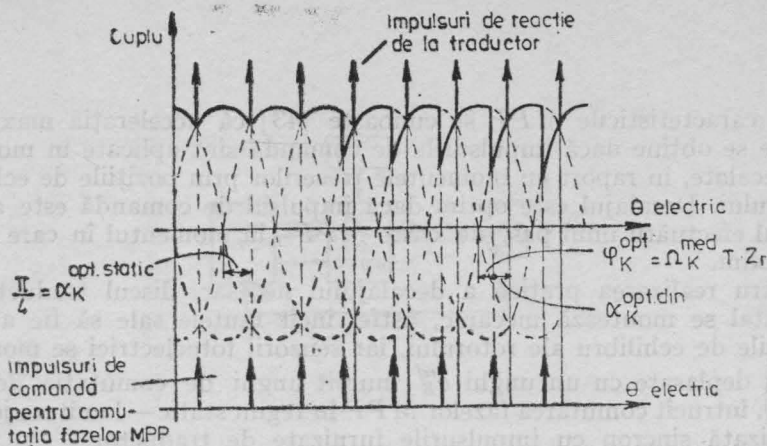


Fig. 6.17. Comanda MPP la unghiuri de comutație optimă.

sul de comandă imediat următor este suprimat, pentru a nu se ajunge la instabilitate dinamică.

$\varphi_K^{real} \leq 0$ înseamnă o frinare nedorită a motorului. Această situație poate surveni la apariția unei puternice perturbații exterioare, pe timpul accelerării MPP sau la tendințe de pierdere a pașilor. În această situație, se generează instantaneu un impuls comandă suplimentar (se comandă efectuarea unui pas în plus).

Pe seama impulsului de comandă generat se face apoi distribuția, externă sau nu microprocesorului, a impulsurilor pe fazele MPP.

Unghiurile φ_K^{max} și φ_K^{real} sînt evaluate de către microprocesorul de comandă în unități de număr de impulsuri.

Astfel, un timer intern al microsistemului marchează durata constantă T . În acest interval se contorizează impulsurile de reacție de la traductor. φ_K^{max} se evaluează prin numărul de impulsuri de reacție sosite în intervalul T . Această evaluare este corectă, întrucît la comanda optimă, numărul de impulsuri de comandă trebuie să fie egal cu cel al impulsurilor de reacție în intervalul de timp T . Avansul real la comutație φ_K^{real} se determină în aceleași unități de măsură, prin evaluarea diferenței dintre suma impulsurilor de comandă a MPP și cea a impulsurilor de reacție [43].

Avînd în vedere considerentele de mai sus, funcția (F5) a microprocesorului pentru circuit de comandă a MPP în buclă minoră, se detaliază după cum urmează:

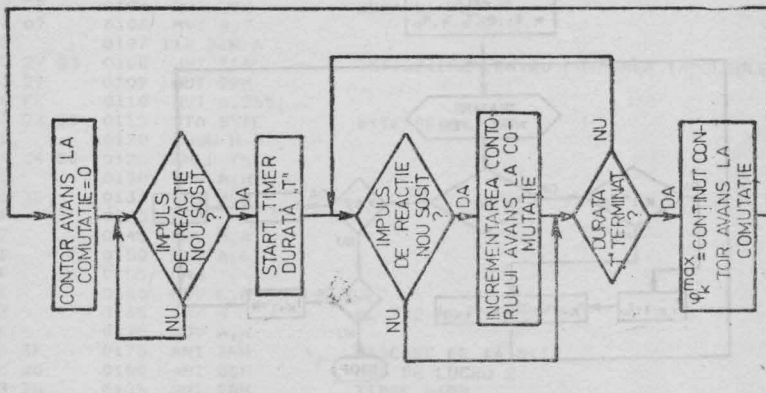
(F5A) — Calculul unghiului φ_K^{max} .

(F5B) — Limitarea superioară a lui φ_K^{real} .

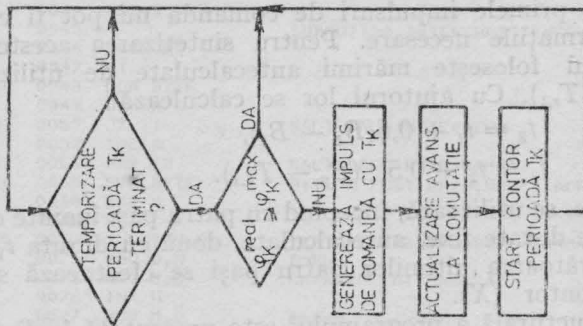
(F5C) — Limitarea inferioară a lui φ_K^{real} .

În figura 6.18 se prezintă organigramele de implementare a funcțiilor (F5A), (F5B) și (F5C) și a comenzilor care se dau în cazul neîncadrării lui φ_K^{real} în limitele admise.

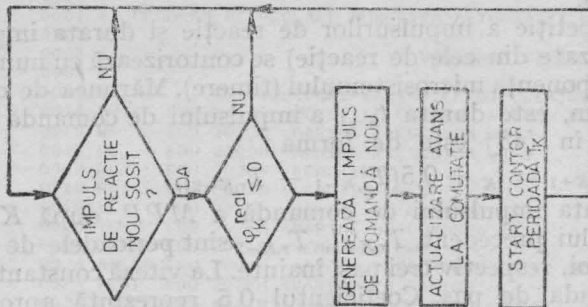
În figurile 6.16, c, d, microprocesorul este utilizat într-un sistem de reglare automată a vitezei (SRA—V). Profilul de viteză din ciclul de poziționare nu este predeterminat, ci se sintetizează, în timp real, durata impulsurilor de comandă necesare la un moment dat. În acest scop, se folosește



a Determinarea lui φ_k^{\max} (F5A)



b Limitarea superioară pentru φ_k^{real} (F5B)



c Limitarea inferioară pentru φ_k^{real} (F5C)

Fig. 6.18. Organigramele subprogramelor pentru funcția (F5) implicate la comanda optimă în buclă minoră a MPP.

un algoritm de reglare numeric (ARN), care asigură durata minimă pentru regimul tranzitoriu și eroare nulă în regim staționar — tip PI cu viteză optimă de răspuns.

Perioada de repetiție a impulsurilor de reacție și durata impulsurilor de comandă (sintetizate din cele de reacție) se contorizează cu numărătoare în timp real, din componența microsistemului (timere). Mărimea de comandă, calculată de algoritm, este durata t_{K+1} a impulsului de comandă următor. Algoritmul utilizat în [43] este de forma:

$$t_{K+1} = t_K + 0,5(T_{r,K-1} - T_{r,K-2}), \quad (6.6)$$

unde t_{K+1} este durata impulsului de comandă a MPP, după K pași, t_K este durata impulsului precedent, $T_{r,K-1}$, $T_{r,K-2}$ sînt perioadele de repetiție de la traductor cu doi, respectiv trei pași înainte. La viteză constantă, $T_{r,K-1}$ devine T_0 — perioada de pas. Coeficientul 0,5 reprezintă aproximarea, pentru implementare mai ușoară, a unui coeficient subunitar pozitiv.

Valoarea t_{K+1} obținută prin calcul este introdusă de microprocesor în timerul pentru durata impulsului, care realizează apoi temporizarea propriu-zisă.

Duratele pentru primele impulsuri de comandă nu pot fi calculate, întrucît lipsesc informațiile necesare. Pentru sintetizarea acestor prime impulsuri, programul folosește mărimi antecalulate de utilizator, t_1 , B_1 , B_2 ($B_1 < B_2 < T_{r,1}$). Cu ajutorul lor se calculează:

$$t_2 = t_1 + 0,5(B_1 - B_2), \quad (6.7)$$

$$t_3 = t_2 + 0,5 (B_2 - T_{r,1}). \quad (6.8)$$

Pentru decelerare, se utilizează, începînd cu patru pași înainte de oprire ($N_p - 4$), impulsuri de durate fixe, antecalulate, două cu durata t_1 și două cu durata t_f . Numărătoarea ultimilor patru pași se efectuează software, cu ajutorul unui contor (X).

Organigrama structurală a programului este prezentată în figura 6.19

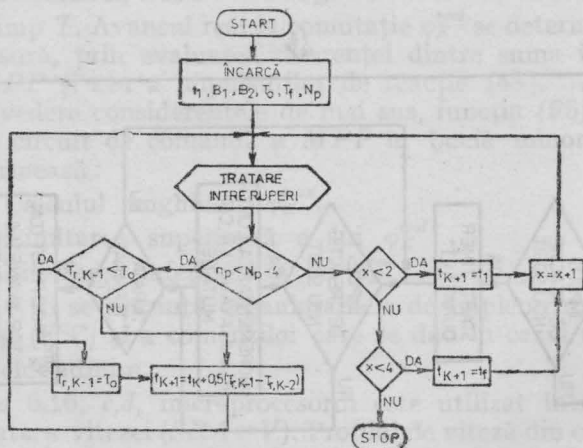


Fig. 6.19. Organigrama simplificată a programului de reglare a vitezei MPP în circuit închis cu microprocesor.

ANEXA 1

ASB e

```

0000          0000 *
0000          0001 * PROGRAM DE REGLARE NUMERICA A TURATIEI UNUI MCC
0000          0002 *
0000          0003 *
0000          0004 * VARIANTA TIRISTORIZATA
0000          0005 * INITIALIZARI VARIABILE SI COEFICIENTII
0000          0014 BYTE EQU 2023H =OCTET DE CONTROL PENTRU RST 7.5
0000          0015 WK EQU 2022H =MARIME PRESCRISA wk
0000          0016 W1 EQU 200BH =MARIME PRESCRISA WK-1
0000          0017 RK EQU 2026H =REACTIE rk
0000          0018 RK1 EQU 2027H =REACTIE rk-1
0000          0019 YK EQU 2028H =MARIME DE COMANDA yk
0000          0020 YK1 EQU 2024H =MARIME DE COMANDA yk-1
0000          0021 SIGN EQU 2021H =SEMN ABATERE
0000          0024 ADCOD EQU 2001H =ADRESA PRESCRIERE DE LA DAF
0000          0025 TEST EQU 202BH =VARIABILA FANION
0000          0026 CRLF EQU 4441H =SUBRUTINA PENTRU CR,LF
0000          0027 ADAD EQU 2012H =ADRESA CURENTA PENTRU RECEPTIE DAF
0000          0037 FAN EQU 2000H =FANION RECEPTIE COMPLETA DAF
0000          0040 *
0000          0041 DRG 20CEH AICI SE AJUNGE FRIN ACCEPTAREA RST 7.5
20CE C3 00 83 0042 JMP B300H SALT IN INTERRUPT TIMER
20D1          0043 *
20D1          0044 DRG B300H
B300          0045 * SUBRUTINA INTERRUPT TIMER
B300 F5          0046 PUSH M
B301 FB          0047 EI
B302 3A 23 20 0048 LDA BYTE
B305 B7          0049 ORA A
B306 CA 21 83 0050 JZ I1 SALT DACA BYTE=ALFA
B309 3C          0052 INR A
B30A C2 12 83 0054 JNZ I2 SALT DACA BYTE=GAMMA
B30D 32 23 20 0055 STA BYTE BYTE A FOST BETA,VA FI ALFA
B310 F1          0060 POP M
B311 C9          0065 RET
B312 21 26 20 0070 I2 LXI H,RK TRATAREA PENTRU BYTE=GAMMA
B315 56          0071 MOV D,M D=RK
B316 71          0075 MOV M,C RK=REZULTAT MASURA
B317 23          0076 INX H
B318 72          0077 MOV M,D RK1=D
B319 E1          0088 POP H IGNORARE PSW SALVAT INITIAL
B31A E1          0089 POP H IGNORARE ADRESA DE INTOARCERE IN PROGRAMUL APELANT
B31B FA 48 85 0090 JM R2C BYTE=GAMMA 2
B31E C3 00 85 0095 JMP I BYTE=GAMMA 1
B321 3E 08 0100 I1 MVI A,B TRATAREA PENTRU BYTE=ALFA
B323 D3 29 0105 OUT 29H COMANDA Th2
B325 3E 07 0106 MVI A,7
B327 3D 0107 I1A DCR A
B328 C2 27 83 0108 JNZ I1A INTIRZIERE PENTRU FORMAREA IMPULSULUI
B32B D3 29 0109 OUT 29H
B32D 3E FF 0110 MVI A,255
B32F 32 23 20 0115 STA BYTE BYTE=BETA
B332 E5 0120 PUSH H
B333 2A 24 20 0125 LHLD YK1
B336 7C 0130 MOV A,H
B337 D6 3C 0135 SUI 3CH
B339 2F 0140 CMA
B33A 67 0145 MOV H,A
B33B 7D 0150 MOV A,L
B33C 2F 0155 CMA
B33D 6F 0160 MOV L,A
B33E 23 0165 INX H HL=T/2-HL
B33F 7C 0170 MOV A,H
B340 E6 3F 0175 ANI 3FH MASCARE PE 14 BITI
B342 C6 80 0180 ADI 80H MOD DE LUCRU 2
B344 D3 2D 0185 OUT 2DH TIMER HIGH
B346 7D 0190 MOV A,L
B347 53 2C 0195 OUT 2CH TIMER LOW
B349 E1 0200 POP H

```

834A	3E	C1	0205	MVI A,0C1H	
834C	D3	2B	0210	OUT 2BH	START TIMER
834E	F1		0215	POP M	
834F	C9		0220	RET	
8350			0221	*	
8350			0223	*	SUBROUTINA MASURA
8350			0224	*	
8350			0225	ORG 2030H	IN STATIC RAM
2030	3E	BC	0230	MAS MVI A,0BCH	
2032	D3	2B	0235	OUT 2BH	
2034	3E	00	0240	MVI A,0	
2036	D3	2C	0245	OUT 2CH	TIMER=5msec
2038	3E	C1	0250	MVI A,0C1H	
203A	0E	00	0255	MVI C,0	CONTOR=0
203C	D3	2B	0260	OUT 2BH	START TIMER
203E	DB	2A	0265	EF1 IN 2AH	
2040	E6	0B	0270	ANI B	
2042	47		0275	MOV B,A	
2043	DB	2A	0280	EF2 IN 2AH	
2045	E6	0B	0285	ANI B	
2047	B8		0290	CMP R	
2048	CA	43 20	0295	JZ EF2	
204B	0C		0300	INR C	NUMARARE TRANZITIII
204C	C3	3E 20	0305	JMP EF1	
204F			0306	*	
204F			0308	*	SUBROUTINA DE CALCUL-ARN
204F			0309	*	
204F			0310	ORG 8380H	
8380	2A	24 20	0315	ARN LHLD YK1	
8383	AF		0318	ARN1 XRA A	
8384	32	21 20	0319	STA SIGN	SIGN=0
8387	3A	26 20	0330	LDA RK	
838A	47		0335	MOV B,A	
838B	3A	22 20	0340	LDA WK	
838E	90		0345	SUB B	A=W-R
838F	D2	9B 83	0350	JNC C3	SALT DACA W)=R
8392	2F		0355	CMA	W<R
8393	3C		0360	INR A	
8394	47		0365	MOV B,A	B=-A
8395	3E	01	0370	MVI A,1	
8397	32	21 20	0375	STA SIGN	SIGN=1
839A	78		0380	MOV A,B	A=B
839B			0384	*	
839H	57		0385	C3 MOV D,A	
839C	47		0390	MOV B,A	
839D	B7		0395	ORA A	
839E	1F		0400	RAR	
839F	5F		0415	MOV E,A	
83A0	3E	00	0420	MVI A,0	
83A2	1F		0425	RAR	
83A3	4F		0430	MOV C,A	
83A4	78		0435	MOV A,B	
83A5	B3		0440	ADD E	
83A6	47		0445	MOV B,A	
83A7	7A		0450	MOV A,D	
83A8	B7		0455	ORA A	
83A9	17		0460	RAL	
83AA	5F		0465	MOV E,A	
83AB	3E	00	0470	MVI A,0	
83AD	17		0475	RAL	
83AE	57		0480	MOV D,A	
83AF	79		0485	MOV A,C	
83B0	92		0490	SUB E	
83B1	6F		0495	MOV C,A	
83B3	76		0500	MOV A,B	
83B4	7A		0505	SUB D	
83B4	47		0510	MOV B,A	
83B5	71		0515	MOV A,E	
83B6	E7		0520	ORA A	
83B7	17		0525	RAL	
83B8	4F		0530	MOV E,A	
83B9	7A		0535	MOV A,D	

83BA	17	0540	FAL
83BB	57	0545	MOV D,A
83BC	79	0550	MOV A,C
83BD	93	0555	SUB E
83BE	4F	0560	MOV C,A
83BF	78	0565	MOV A,B
83C0	9A	0570	SEB B
83C1	47	0575	MOV B,A
83C2		0576	*
83C2	3A 21 70	0580	LDA SIGN
83C5	B7	0585	ORA A
83C6	CA 00 83	0590	JZ CS
83C9	79	0595	MOV A,C
83CA	2F	0600	CMA
83CB	4F	0605	MOV C,A
83CC	78	0610	MOV A,B
83CD	2F	0615	CMA
83CF	47	0620	MOV B,A
83CF	03	0625	INX B
83D0	09	0630	CS DAD B
83D1	3E 01 *	0635	MVI A,1
83D3	32 21 20	0640	STA SIGN
83D6	3A 27 20	0645	LDA RM1
83D9	47	0650	MOV B,A
83DA	3A 08 20	0655	LDA W1
83DD	90	0660	SUB E
83DE	B2 E9 83	0665	JNC CB
83E1	2F	0670	CMA
83E2	3C	0675	INR A
83E3	47	0680	MOV B,A
83E4	AF	0685	XRA A
83E5	32 21 20	0690	STA SIGN
83E8	78	0695	MOV A,B
83E9		0699	*
83E9	47	0700	CB MOV B,A
83EA	4F	0705	MOV C,A
83EB	B7	0710	ORA A
83EC	1F	0715	RAR
83ED	5F	0720	MOV E,A
83EE	3E 00	0725	MVI A,0
83F0	1F	0730	RAR
83F1	57	0735	MOV D,A
83F2	7B	0740	MOV A,E
83F3	1F	0745	RAR
83F4	5F	0750	MOV E,A
83F5	7A	0755	MOV A,D
83F6	1F	0760	RAR
83F7	57	0765	MOV D,A
83F8	81	0770	ADD C
83F9	4F	0775	MOV C,A
83FA	78	0780	MOV A,B
83FB	8B	0785	ADC E
83FC	47	0790	MOV B,A
83FD	B7	0795	ORA A
83FE	7B	0800	MOV A,E
83FF	1F	0805	RAR
8400	5F	0810	MOV E,A
8401	7A	0815	MOV A,D
8402	1F	0820	RAR
8403	81	0825	ADD C
8404	4F	0830	MOV C,A
8405	78	0835	MOV A,B
8406	8B	0840	ADC E
8407	47	0845	MOV B,A
8408		0846	*
8408	3A 21 20	0850	LDA SIGN
840B	B7	0855	ORA A
840C	CA 16 84	0860	JZ C10
840F	79	0865	MOV A,C
8410	2F	0870	CMA
8411	4F	0875	MOV C,A
8412	78	0880	MOV A,B

S-A EXECUTAT OPERATIA BC=378*A

SALI DACA SIGN=0

SIGN=1 ; BC=-BC
HL=HL+DC

SIGN=1

A=W-R
SALI DACA W>R
W<R

B=-A

SIGN=0
A=-B

S-A EXECUTAT OPERATIA BC=353*A

SALI DACA SIGN=0

8413	2F	0885	CMA	
8414	47	0890	MOV B,A	
8415	03	0895	INX B	SIGN=1 ; RC=-BQ
8416	09	0900	C10 PAD B	HL=HL+RC
8417	3A 2B 20	0905	LDA TEST	
841A	B7	0910	ORA A	
841B	CA 2A 84	0915	JZ C14	SALT DAC3 TEST=0
841E	3E 71	0920	MVI A,71H	TEST=1
8420	BC	0925	CMP H	TEST DEPASIRE SUPERIOARA
8421	D2 32 84	0930	JNC C15	
8424	21 20 1C	0935	LXI H,7200	CORECTIE DEPASIRE
8427	C3 32 84	0940	JMP C15	
842A	7C	0945	C14 MOV A,H	
842B	B7	0950	ORA A	TEST DEPASIRE INFERIOARA
842C	F2 32 84	0955	JP C15	
842F	21 00 00	0960	LXI H,0	CORECTIE DEPASIRE
8432	22 2B 20	0965	C15 SHLD YK	YK=REZULTAT CORECTAT
8435	C9	0970	RET	
8436		0985	*	
8436		0990	*	PROGRAMUL PRINCIPAL
8436		0995	*	
8436		1000	ORG 8480H	
8480	31 FE BF	1001	LXI M,0BFFEH	INITIALIZARE STIVA
8483	AF	1005	XRA A	
8484	32 22 20	1007	STA WK	INITIALIZARI VARIABLE
8487	32 26 20	1020	STA RK	
848A	32 C9 20	1022	STA 20C9H	ADRESA LOW PENTRU RST 6.5 ESTE 00
848D	32 27 20	1025	STA RK1	
8490	32 00 20	1026	STA FAN	
8493	32 28 20	1030	STA YK	
8496	32 29 20	1031	STA YK+1	
8499	3E 01	1032	MVI A,1	
849B	D3 28	1033	OUT 2BH	PROGRAMARE PORTURI 29H SI 2AH
849D	3E 0	1034	MVI A,9	
849F	30	1035	DB 30H	ACEASTA ESTE INSTRUCIUNEA SIM
84A0	3E B7	1036	MVI A,87H	
84A2	32 CA 20	1037	STA 20CAH	ADRESA HIGH PENTRU RST 6.5 ESTE 87H
84A5	C3 00 85	1042	JMP 1	
84A8		1043	*	
84A8		1044	T ORG 8500H	
8500	3A 28 20	1045	LBA YK	
8503	32 24 20	1050	STA YK1	YK1-YK (LOW)
8506	3A 29 20	1055	LBA YK+1	
8509	32 25 20	1060	STA YK1+1	YK1-YK (HIGH)
850C	AF	1070	XRA A	
850D	32 2B 20	1080	STA TEST	
8510	3A 22 20	1090	LBA WK	W1-WK
8513	32 0B 20	1095	STA W1	
8516	3A 00 20	1210	LDA FAN	
8519	FE 02	1220	CFI 2	
851B	C2 28 85	1230	JNZ FP1	
851E	3A 01 20	1240	LDA ADCOD	
8521	32 22 20	1250	STA WK	
8524	AF	1251	XRA A	
8525	32 00 20	1252	STA FAN	
8528		1257	*	
8528	3A 25 20	1260	FP1 LDA YK1+1	
852B	FE 3C	1270	CFI 3CH	
852D	FA 70 85	1280	JM FP2	SALT IN REGIMUL 1 SAU 3
8530		1285	*	
8530	3E DD	1290	R2 MVI A,ODDH	REGIMUL 2
8532	32 23 20	1300	STA BYTE	BYTE=GAMMA 2
8535	3E 04	1301	MVI A,4	
8537	D3 29	1302	OUT 29H	COMANDA Th1
8539	3E 07	1303	MVI A,7	
853B	3D	1304	R2I BCR A	
853C	C2 3B 85	1305	JNZ R2T	INTRIZIERE PENTRU FORMAREA IMPULSULUI
853F	D3 29	1306	OUT 29H	
8541	3C	1307	INR A	
8542	32 2B 20	1308	STA TEST	TEST=1
8545	C3 30 20	1310	JMP MAS	
8548	AF	1320	R2C XRA A	

8549	32	23	20	1325	STA BYTE	BYTE=ALFA
854C	2A	24	20	1330	LHLD YK1	
854F	7C			1340	MOV A,H	
8550	D6	3C		1350	SUI 3CH	
8552	32	25	20	1360	STA YK1+1	YK1=YK1-T/2
8555	E6	3F		1376	ANI 3FH	
8557	C6	80		1380	ADI 80H	
8559	D3	2D		1390	OUT 2DH	
855B	7D			1400	MOV A,L	
855C	D3	2C		1410	OUT 2CH	TIMER=YK1
855E	3E	C1		1420	MVI A,0C1H	
8560	D3	28		1430	OUT 28H	START TIMER
8562	CD	83	83	1440	CALL ARN1	
8565	76			1455	HLT	
8566	3A	23	20	1460	R2A LDA BYTE	
8569	B7			1470	DRA A	
856A	C2	66	85	1480	JNZ R2A	SALT DACA BYTE()ALFA
856D	C3	00	85	1490	JMP T	SALT DACA BYTE=ALFA
8570	FE	01		1495	*	
8572	D2	91	85	1500	PF2 CPI 1	
8575				1510	JNC R1	SALT IN REGIMUL 1
8575				1519	*	REGIMUL 3
8575	3E	FF		1520	R3 MVI A,25H	
8577	32	23	20	1530	STA BYTE	BYTE=BETA
857A	3C			1540	INR A	
857B	D3	2C		1550	OUT 2CH	
857D	3E	BC		1560	MVI A,0BCH	
857F	D3	2D		1570	OUT 2DH	TIMER=T/2
8581	3E	C1		1580	MVI A,0C1H	
8583	D3	28		1590	OUT 28H	START TIMER
8585	CD	80	83	1600	CALL ARN	
8588	76			1610	HLT	
8589	3E	23		1620	MVI A,23H	
858B	32	23	20	1630	STA BYTE	BYTE=GAMMA 1
858E	C3	30	20	1640	JMP MAS	
8591				1645	*	REGIMUL 1
8591	3E	04		1650	R1 MVI A,4	
8593	D3	29		1660	OUT 29H	AMORSARE TR
8595	3E	07		1661	MVI A,7	
8597	3D			1662	R1T INR A	
8598	C2	97	85	1663	JNZ R1T	FORMARE IMPULS
859B	D3	29		1664	OUT 29H	
859D	AF			1670	XRA A	
859E	32	23	20	1680	STA BYTE	BYTE=ALFA
85A1	3A	24	20	1690	LDA YK1	
85A4	D3	2C		1700	OUT 2CH	
85A6	3A	25	20	1710	LDA YK1+1	
85A9	E6	3F		1720	ANI 3FH	
85AB	C6	80		1730	ADI 80H	
85AD	D3	2D		1740	OUT 2DH	TIMER=YK1
85AF	3E	C1		1750	MVI A,0C1H	
85B1	D3	28		1760	OUT 28H	START TIMER
85B3	CD	80	83	1770	CALL ARN	
85B6	76			1780	HLT	
85B7	3A	23	20	1790	R1A LDA BYTE	
85BA	B7			1800	DRA A	
85BB	C2	B7	85	1810	JNZ R1A	SALT DACA BYTE()ALFA
85BE	3E	23		1820	MVI A,23H	BYTE=ALFA
85C0	32	23	20	1830	STA BYTE	BYTE=GAMMA 1
85C3	C3	30	20	1840	JMP MAS	
85C6				1980	*	
85C6				1985	*	SUBROUTINA INTERRUPT DAF
85C6				1990	*	
85C6				2000	ORG 8700H	
8700	F5			2010	PUSH M	
8701	C5			2020	PUSH B	
8702	3A	00	70	2030	LDA 7000H	
8705	CD	36	43	2040	CALL 4336H	TRANSFORMA A (DAF) IN B (ASCII)
8708	E3			2041	XTHL	
8709	E3			2042	XTHL	
870A	E3			2043	XTHL	
870B	E3			2044	XTHL	

870C E3	2045	XTHL	
870D E3	2046	XTHL	
870E E3	2047	XTHL	
870F E3	2048	XTHL	INTIRZIERE CERUTA DE INTERFATA CU DAF-UL
8710 CD 36 44	2050	CALL 4436H	TRIMITE B (ASCII) CATRE DAF
8713 3A 00 20	2060	LDA FAN	
8716 B7	2070	ORA A	
8717 C2 2A 87	2080	JNZ ED1	SALT DACA FAN)=1
871A 78	2090	MOV A,B	FAN=0 ; A=CARACTER ASCII
871B CD 65 45	2100	CALL 4565H	CONVERSI A (ASCII) IN A (HEXA)
871E 32 01 20	2110	STA ADCOD	DEPOZITARE PROVIZORIE
8721 3E 01	2120	MVI A,1	
8723 32 00 20	2130	STA FAN	FAN=1
8726 C1	2140	POP B	
8727 F1	2150	POP M	
8728 FB	2160	EI	
8729 C9	2170	RET	
872A 78	2180	EDI MOV A,B	FAN=1 ; A=CARACTER ASCII
872B CD 65 45	2190	CALL 4565H	CONVERSI A (ASCII) IN A (HEXA)
872E 47	2200	MOV B,A	B=ULTIMUL CARACTER RECEPTIONAT (HEXA)
872F E5	2210	PUSH H	
8730 21 01 20	2220	LXI H,ADCOD	
8733 7E	2230	MOV A,M	A=PRIMUL CARACTER RECEPTIONAT (HEXA)
8734 07	2240	RLC	
8735 07	2250	RLC	
8736 07	2260	RLC	
8737 07	2270	RLC	
8738 B0	2280	ORA B	FORMARE OCTET DIN CELE 2 CARACTERE
8739 77	2290	MOV H,A	ADCOD=VALOARE RECEPTIONATA
873A E1	2300	POP H	
873B 3E 02	2310	MVI A,2	
873D 32 00 20	2320	STA FAN	FAN=2 (RECEPTIE COMPLETA)
8740 CD 41 44	2330	CALL CRLF	CAR RETURN , LINE FEED
8743 C1	2340	POP B	
8744 F1	2350	POP M	
8745 FB	2360	EI	
8746 C9	2370	RET	
8747	2375	*	

ASSM 0

```

0000 *
0000 0001 * PROGRAM DE REGLARE NUMERICA A TJRATIEI UNUI MCC
0000 0002 *
0000 0003 * VARIANTA TRANZISTORIZATA
0000 0004 *
0000 0005 * INITIALIZARI VARIABILE SI COEFICIENTI
0000 0006 SRK EQU 202CH =SEMN RK
0000 0007 SRK1 EQU SRK+1 =SEMN RK1
0000 0008 SW EQU SRK+2 =SEMN WK
0000 0009 SW1 EQU 200BH =SEMN W1
0000 0010 YT EQU 201EH =MARIME DE COMANDA PENTRU TIMER
0000 0011 D0 EQU 2015H =COEFICIENT d1
0000 0012 D1 EQU 2017H =COEFICIENT d0
0000 0014 BYTE EQU 2023H =OCTET DE CONTROL PENTRU RST 7.5
0000 0015 WK EQU 2022H =MARIME PRESCRISA wk
0000 0016 W1 EQU 2005H =MARIME PRESCRISA wk-1
0000 0017 RK EQU 2026H =REACTIE rk
0000 0018 RK1 EQU 2027H =REACTIE rk-1
0000 0019 YK EQU 2028H =MARIME DE COMANDA yk
0000 0020 YK1 EQU 2024H =MARIME DE COMANDA yk-1
0000 0021 SIGN EQU 2021H =SEMN ABATERE
0000 0022 KR EQU 200CH =COEFICIENT kr
0000 0023 TI EQU 200EH =COEFICIENT TI
0000 0024 ADCOD EQU 2001H =ADRESA PRESCRIERE DE LA DAF
0000 0025 TEST EQU 202BH =VARIABILA FANION
0000 0026 CRLF EQU 4441H =SUBROUTINA PENTRU CR,LF
0000 0027 ABAD EQU 2012H =ADRESA CURENTA PENTRU RECEPTIE DAF
0000 0028 ABIN EQU 2019H =ADRESA INITIALA PENTRU RECEPTIE DAF
0000 0029 ABSI EQU 2014H =SEMN ADCOD
0000 0030 OU EQU 4437H =SUBROUTINA OUT DAF (ASCII)
0000 0031 MES1 EQU 0D000H ADRESA MESAJ 1
0000 0032 MES2 EQU 0D1A0H ADRESA MESAJ 2
0000 0033 MES3 EQU 0D1B0H ADRESA MESAJ 3
0000 0034 MES4 EQU 0D200H ADRESA MESAJ 4
0000 0035 HOUT EQU 456DH =SUBROUTINA OUT DAF (HEXA)
0000 0036 SCRN EQU 45ADH =SUBROUTINA PENTRU EDITAREA DE TEXTE PE DAF
0000 0037 FAN EQU 2000H =FANION RECEPTIE COMPLETA DAF
0000 0040 *
0000 0041 ORG 20CEH AICI SE AJUNGE PRIN ACCEPTAREA RST 7.5
20CE C3 00 88 0042 JMP 8800H SALT IN INTERRUPT TIMER
20D1 0043 *
20D1 0044 ORG 8800H
8800 0045 * SUBROUTINA INTERRUPT TIMER
8800 F5 0046 PUSH M
8801 FB 0047 EI
8802 3A 23 20 0048 LDA BYTE
8805 B7 0049 ORA A
8806 CA 2D 88 0050 JZ I1 SALT DACA BYTE=ALFA
8809 3C 0052 INR A
880A C2 12 88 0054 JNZ I2 SALT DACA BYTE=GAMMA
880D 32 23 20 0055 STA BYTE BYTE A FOST BETA,VA FI ALFA
8810 F1 0060 POP M
8811 C9 0065 RET
8812 F5 0069 I2 PUSH M TRATAREA PENTRU BYTE=GAMMA
8813 21 26 20 0070 LXI H,RK
8816 56 0071 MOV D,M D=RK
8817 71 0075 MOV H,C RK=REZULTAT MASURA
8818 23 0076 INX H
8819 72 0077 MOV M,D RK1=D
881A 2E 2C 0078 MVI L,2CH HL=ADRESA SRK
881C 56 0079 MOV D,M D=SRK
881D DB 08 0080 IN B PRELUARE SEMN REACTIE
881F E6 01 0081 ANI I MASCARE
8821 77 0084 MOV M,A SRK=A
8822 F1 0085 POP M RESTAURARE FLAGURI (I2)
8823 23 0086 INX H HL=ADRESA SRK1
8824 72 0087 MOV M,D SRK1=D
8825 E1 0088 POP H IGNORARE PSU SALVAT INITIAL
8826 E1 0089 POP H IGNORARE ADRESA DE INTOARCERE IN PROGRAMUL APELANI

```

8827	FA	92	85	0090	JM R2C	BYTE=GAMMA 2
882A	C3	00	85	0095	JMF T	BYTE=GAMMA 1
882D	D3	29		0100	I1 OUT 29H	TRATAREA FENTRU BYTE=ALFA
882F	3E	FF		0110	MVI A,255	
8831	32	23	20	0115	STA BYTE	BYTE=BETA
8834	E5			0120	PUSH H	
8835	2A	1E	20	0125	LHLD YT	
8838	7C			0130	MOV A,H	
8839	D6	3C		0135	SUI 3CH	
883B	2F			0140	CMA	
883C	67			0145	MOV H,A	
883D	7D			0150	MOV A,I	
883E	2F			0155	CMA	
883F	6F			0160	MOV L,A	
8840	23			0165	INX H	HL=T/2-HL
8841	7C			0170	MOV A,H	
8842	E6	3F		0175	ANI 3FH	MASCARE PE 14 BITI
8844	C6	80		0180	ADI 80H	MOD DE LUCRU 2
8846	D3	2D		0185	OUT 2DH	TIMER HIGH
8848	7D			0190	MOV A,L	
8849	D3	2C		0195	OUT 2CH	TIMER LOW
884B	E1			0200	POF H	
884C	3E	C1		0205	MVI A,0C1H	
884E	D3	2B		0210	OUT 2BH	START TIMER
8850	F1			0215	POF M	
8851	C9			0220	RET	
8852				0221	*	
8852				0223	*	SUBROUTINA MASURA
8853				0224	*	
8852				0225	ORG 2030H	IN STATIC RAM
2030	3E	8C		0230	MAS MVI A,0BCH	
2032	D3	2D		0235	OUT 2DH	
2034	3E	00		0240	MVI A,0	
2036	D3	2C		0245	OUT 2CH	TIMER=5msec
2039	3E	C1		0250	MVI A,0C1H	
203A	0E	00		0255	MVI C,0	CONTOR=0
203C	D3	2B		0260	OUT 2BH	START TIMER
203E	DB	2A		0265	EF1 IN 2AH	
2040	E6	08		0270	ANI B	
2042	47			0275	MOV B,A	
2043	DB	2A		0280	EF2 IN 2AH	
2045	E6	08		0285	ANI B	
2047	BB			0290	CMF B	
2048	CA	43	20	0295	JZ EF2	
2048	0C			0300	INR C	NUMARARE TRANZITII
204C	E3	42	20	0305	JMP EF2-1	
204F				0306	*	
204F				0308	*	SUBROUTINA DE CALCUL-ARN
204F				0309	*	
204F				0310	ORG 8800H	
8880	2A	24	20	0315	ARN LHLD YK1	
8883	AF			0318	ARN1 XKA A	
8884	32	21	20	0319	STA SIGN	SIGN=0
8887	3A	2C	20	0320	LDA SRK	
888A	47			0321	MOV B,A	
888B	3A	2E	20	0322	LDA SW	
888E	A8			0323	XRA B	
888F	C2	E3	88	0324	JNZ A2	SALI FENTRU SEMNE DIFERITE
8892	B0			0325	XRA B	SEMNE IDENTICE
8893	3A	26	20	0330	LDA RK	
8896	47			0335	MOV B,A	
8897	3A	22	20	0340	LDA WK	
889A	C2	AD	88	0341	JNZ A3	SALI DACA W<=0 SI R<=0
889D	90			0345	SUB B	W>=0 SI R>=0 ; A=W-R
889E	D2	D4	88	0350	A4 JNC C3	SALI DACA W>=R
88A1	2F			0351	CMA	
88A2	3C			0352	INR A	
88A3	47			0353	MOV B,A	
88A4	3E	01		0354	MVI A,1	
88A6	32	21	20	0355	STA SIGN	SIGN=1
88A7	78			0356	MOV A,B	
88AA	C3	D4	88	0357	JMF C3	

88AD 4F	0358	A3	MOV C,A	
88AE 78	0359		MOV A,B	
88AF 91	0360		SUB C	A=R-W
88B0 C3 9E 88	0361		JMP A4	
88B3 A8	0362	A2	XRA B	
88B4 3A 26 20	0363		LDA RK	
88B7 47	0364		MOV B,A	
88B8 3A 22 20	0365		LDA WK	
88BB C2 C7 88	0366		JNZ A5	SALT DACA W<0 SI R>=0
88BE 80	0367		ADD B	W)=0 SI R<0 ; A=W+R
88BF D2 D4 88	0368		JNC C3	
88C2 3E FF	0369		MVI A,255	CORECTIE DEPASIRE
88C4 C3 D4 88	0370		JMP C3	
88C7 80	0371	A5	ADD B	A=W+R
88C8 47	0372		MOV B,A	
88C9 3E 01	0374		MVI A,1	
88CB 32 21 20	0375		STA SIGN	SIGN=1
88CE 78	0376		MOV A,B	
88CF D2 D4 88	0378		JNC C3	
88D2 3E FF	0380		MVI A,255	CORECTIE DEPASIRE
88D4 EB	0390	C3	XCHG	
88D5 2A 15 20	0400		LHLD DO	
88D8 CD 70 87	0410		CALL INO	INMULTIRE CU do
88DB AF	0635		XRA A	
88DC 32 20 20	0640		STA SIGN	SIGN=0
88DF 3A 2D 20	0641		LDA SRK1	
88E2 47	0642		MOV B,A	
88E3 3A 0B 20	0643		LDA SW1	
88E6 A8	0644		XRA B	
88E7 C2 0B 89	0645		JNZ A6	SALT PENTRU SEMNE DIFERITE
88EA B0	0646		ORA B	SEMNE IDENTICE
88EB 3A 27 20	0647		LDA RK1	
88EE 47	0648		MOV B,A	
88EF 3A 0B 20	0649		LDA W1	
88F2 CA 05 89	0650		JZ A7	SALT DACA W)=0 SI R)=0
88F5 90	0651		SUB B	W<0 SI R<0 ; A=W-R
88F6 D2 2C 89	0652	AB	JNC C8	SALT DACA W)=R
88F9 2F	0653		CMA	
88FA 3C	0654		INR A	
88FB 47	0655		MOV B,A	
88FC 3E 01	0660		MVI A,1	
88FE 32 21 20	0661		STA SIGN	N=1
8901 78	0662		MOV A,B	
8902 C3 2C 89	0663		JMP C8	
8905 4F	0664	A7	MOV C,A	
8906 78	0665		MOV A,B	
8907 91	0666		SUB C	A=R-W
8908 C3 F6 88	0667		JMP A8	
890B A8	0668	A6	XRA B	
890C 3A 27 20	0669		LDA RK1	
890F 47	0670		MOV B,A	
8910 3A 0B 20	0675		LDA W1	
8913 CA 1F 89	0676		JZ A9	SALT DACA W)=0 SI R<0
8916 80	0677		ADD B	A=W+R
8917 D2 2C 89	0678		JNC C8	
891A 3E FF	0679		MVI A,255	CORECTIE DEPASIRE
891C C3 2C 89	0680		JMP C8	
891F 80	0685	A9	ADD B	A=W+R
8920 47	0686		MOV B,A	
8921 3E 01	0687		MVI A,1	
8923 32 21 20	0688		STA SIGN	SIGN=1
8926 78	0689		MOV A,B	
8927 D2 2C 89	0690		JNC C8	
892A 3E FF	0693		MVI A,255	CORECTIE DEPASIRE
892C EB	0700	C8	XCHG	
892D 2A 17 20	0710		LHLD D1	
8930 CD 70 87	0720		CALL INO	INMULTIRE CU d1
8933 C9	0970		RET	REZULTATUL NECORECTAT SE AFLA IN
8934	0985	*		
8934	0990	*		PROGRAMUL PRINCIPAL
8934	0995	*		
8934	1000	ORG	84A0H	

8440	31	FF	FF	1001	LXI H,OFFFH	INITIALIZARE STIVA
8443	AF			1005	XRA A	
8444	03	0A		1006	OUT 10	PROGRAMARE PORT B
8446	32	22	20	1007	STA WK	INITIALIZARI VARIABLE
8449	32	1E	20	1011	STA YT	
844C	32	1F	20	1012	STA YT+1	
844F	32	2E	20	1013	STA SW	
848C	32	28	20	1014	STA TEST	
8485	32	2C	20	1016	STA SRK	
8488	32	20	20	1017	STA SRK1	
848B	32	26	20	1020	STA RK	
848E	32	14	20	1021	STA ADSI	
84C1	32	C9	20	1022	STA 20C9H	ADRESA LOW PENTRU RST 6.5 ESTE 00
84C4	32	27	20	1025	STA RK1	
84C7	32	00	20	1026	STA FAN	
84CA	32	28	20	1030	STA YK	
84CD	32	29	20	1031	STA YK+1	
84D0	3E	01		1032	MVI A,1	
84D2	D3	28		1033	OUT 29H	PROGRAMARE PORTURI 29H SI 2AH
84D4	3E	09		1034	MVI A,9	
84D6	30			1035	DB 30H	ACEASTA ESTE INSTRUCIUNEA "SIM"
84D7	3E	87		1036	MVI A,87H	
84D9	32	CA	20	1037	STA 20CAH	ADRESA HIGH PENTRU RST 6.5 ESTE 87H
84DC	21	19	20	1038	LXI H,ADIF	
84DE	22	12	20	1039	SHLD ADAD	
84E2	C3	00	85	1042	JMP I	
84E5				1043	*	
84E5				1044	I ORG 8500H	
8500	3A	28	20	1045	LDA YK	YK1=YK (LOW)
8503	32	24	20	1050	STA YK1	
8506	3A	29	20	1055	LDA YK+1	YK1-YK (HIGH)
8509	32	25	20	1060	STA YK1+1	
850C	3A	22	20	1090	LDA WK	C=WK
850F	4F			1092	MOV C,A	W1=WK
8510	32	08	20	1095	STA W1	
8513	3A	2E	20	1100	LDA SW	SW1=SW
8516	32	0B	20	1105	STA SW1	
8519	3A	00	20	1110	LDA FAN	
851C	B7			1111	ORA A	
851D	CA	70	85	1112	JZ PF1	SALT PENTRU FAN=0
8520	3A	01	20	1115	LDA ADCOD	D=WDAF
8523	57			1120	MOV D,A	
8524	3A	14	20	1125	LDA ADSI	J=SDAF
8527	47			1130	MOV B,A	A=SW
8528	3A	2F	20	1135	LDA SW	
852B	A8			1140	XRA B	A=WK
852C	79			1145	MOV A,C	SALT PENTRU SEMNE DIFERITE
852D	C2	5D	85	1150	JNZ T1	SEMNE IDENTICE
8530	BA			1155	CMP D	SALT PENTRU WDAF)WK
8531	DA	43	85	1160	JC T2	WDAF<=WK ; A=WK-20
8534	D6	14		1165	SUI 20	SALT PENTRU A<0 (DEPASIRE INFERIOARA)
8536	DA	4F	85	1170	JC T3	WK=A
8539	32	22	20	1175	STA WK	
853C	BA			1180	CMP D	
853D	D2	70	85	1182	JNC PF1	SALT DACA WDAF<=A (PRELUAREA NU S-A TERMINAT)
8540	C3	4F	85	1185	JMP T3	SALT DACA WDAF)A
8543	C6	14		1190	T2 ADI 20	WDAF)WK ; A=WK+20
8545	DA	4F	85	1195	JC T3	SALT IN CAZ DE DEPASIRE
8548	32	22	20	1210	STA WK	WK=A
854B	BA			1220	CMP D	
854C	DA	70	85	1225	JC PF1	SALT DACA WDAF)A (PRELUAREA NU S-A TERMINAT)
854F	7A			1230	T3 MOV A,D	FINAL PRELUARE
8550	32	22	20	1232	STA WK	WK=WDAF
8553	AF			1233	XRA A	
8554	32	00	20	1234	STA FAN	FAN=0 (RECEPTIE COMPLETA)
8557	32	14	20	1236	STA ADSI	ADSI=0
855A	C3	70	85	1238	JMP PF1	
855D	D6	14		1240	T1 SUI 20	SEMNE DIFERITE ; A=WK-20
855F	32	22	20	1244	STA WK	WK=A
8562	D2	70	85	1245	JNC PF1	SALT DACA A)0 (PRELUAREA NU S-A TERMINAT)
8565	2F			1246	CMA	
8566	4C			1247	INR A	

8567	4F	1248	MOV C,A	
8568	78	1249	MOV A,B	
8569	32 2E 20	1250	STA SW	SW=SDAF
856C	79	1252	MOV A,C	
856D	C3 4B 85	1255	JMP T3-4	SALT IN RAMURA CU SEMNE IDENTICE
8570		1257	*	
8570	3A 25 20	1260	PP1 LDA YK1+1	
8573	B7	1262	ORA A	
8574	F2 78 85	1264	JF *+1	
8577	2F	1266	CMA	
8578	FE 3C	1270	CPI 3CH	
857A	FA EB 85	1280	JM PP2	SALT IN REGIMUL 1 SAU 3
857D		1285	*	
857D	3E DD	1290	R2 MVI A,ODDH	REGIMUL 2
857F	32 23 20	1300	STA BYTE	BYTE=GAMMA 2
8582	3A 2B 20	1301	LDA TEST	
8585	B7	1302	ORA A	
8586	3E 04	1303	MVI A,4	
8588	CA 8D 85	1304	JZ R2T	SALT DACA TEST=0 ; A=4
858B	C6 04	1305	ADI 4	TEST=1 ; A=B
858D	D3 29	1306	R2T OUT 29H	COMANDA TRANZISTOARE
858F	C3 30 20	1310	JMP MAS	
8592	AF	1320	R2C XRA A	
8593	32 23 20	1325	STA BYTE	BYTE=ALFA
8596	2A 24 20	1330	LHLD YK1	A=YK1 (HIGH)
8599	84	1340	ADD H	
859A	E5	1342	PUSH H	
859B	F2 AA 85	1345	JP R2B	SALT DACA YK1}=0
859E	C6 3C	1350	ADI 3CH	YK1<0
85A0	2F	1360	CMA	
85A1	67	1362	MOV H,A	
85A2	7D	1363	MOV A,L	
85A3	2F	1364	CMA	
85A4	6F	1365	MOV L,A	
85A5	23	1366	INX H	HL=YK1+T/2
85A6	7C	1367	MOV A,H	
85A7	C3 AD 85	1368	JMP R2E	
85AA	D6 3C	1369	R2B SUI 3CH	
85AC	67	1370	MOV H,A	HL=YK1-T/2
85AD	22 1E 20	1372	R2E SHLD YT	YT=HL
85B0	E6 3F	1376	ANI 3FH	
85B2	C6 80	1380	ADI 80H	
85B4	D3 2D	1390	OUT 2DH	
85B6	7D	1400	MOV A,L	
85B7	D3 2C	1410	OUT 2CH	TIMER=HL
85B9	3E C1	1420	MVI A,0C1H	
85BB	D3 2B	1430	OUT 2BH	START TIMER
85BD	E1	1435	POP H	
85BE	CD 83 88	1440	CALL ARN1	
85C1	3A 2B 20	1441	LDA TEST	
85C4	B7	1442	ORA A	
85C5	C2 D4 85	1443	JNZ R2F	SALT DACA YK1<0
85C8	3E 77	1444	MVI A,77H	YK1>0
85CA	BC	1445	CMP H	TEST DEPASIRE SUPERIOARA
85CB	D2 DD 85	1446	JNC R2D	
85CE	21 F0 77	1447	LXI H,77FOH	CORECTIE DEPASIRE
85D1	C3 DD 85	1448	JMP R2D	
85D4	3E 87	1449	R2P MVI A,87H	
85D6	BC	1450	CMP H	TEST DEPASIRE INFERIOARA
85D7	DA DD 85	1451	JC R2D	
85DA	21 10 88	1452	LXI H,8810H	CORECTIE DEPASIRE
85DB	22 28 20	1453	R2D SHLD YK	YK=REZULTAT CALCUL (ARN) , CORECTAT
85E0	76	1455	HLT	
85E1	3A 23 20	1460	R2A LDA BYTE	
85E4	B7	1470	ORA A	
85E5	C2 E1 85	1480	JNZ R2A	SALT DACA BYTE<>ALFA
85E8	C3 00 85	1490	JMP T	SALT DACA BYTE=ALFA
85EB		1495	*	
85EB	B7	1500	PP2 ORA A	
85EC	C2 1C 86	1501	JNZ R1	
85EF	3A 24 20	1505	LDA YK1	
85F2	B7	1510	ORA A	

85F3 C2 1C 86	1515	JNZ R1	
85F6	1519	*	REGIMUL 3
85F6 3E FF	1520	R3 MVI A,255	
85F8 32 23 20	1530	STA BYTE	BYTE=BETA
85FB 3C	1540	INR A	
85FC D3 2C	1550	OUT 2CH	
85FE 3E KC	1560	MVI A,0BCH	
8600 D3 2D	1570	OUT 2DH	TIMER=T/2
8602 3E C1	1580	MVI A,0C1H	
8604 D3 28	1590	OUT 28H	START TIMER
8606 CD B0 88	1600	CALL ARN	
8609 7C	1601	MOV A,H	
860A 07	1602	RLC	
860B E6 01	1603	ANI 1	
860D 32 2F 20	1604	STA TEST	ACTUALIZARE VARIABILA "TEST"
8610 22 28 20	1605	SHLD YK	YK=REZULTAT CALCUL (ARN) , CORECTAT
8613 76	1610	HLT	
8614 3E 23	1620	MVI A,23H	
8616 32 23 20	1630	STA BYTE	BYTE=GAMMA 1
8619 C3 30 20	1640	JMP MAS	
861C	1645	*	REGIMUL 1
861C 3A 2B 20	1650	R1 LDA TEST	
861F B7	1652	ORA A	
8620 3E 04	1654	MVI A,4	
8622 CA 27 86	1656	JZ R1T	SALT DACA TEST=0 ; A=4
8625 C6 04	1658	ADI 4	TEST=1 ; A=8
8627 D3 29	1660	R1T OUT 29H	COMANDA TRANZISTOARE
8629 2A 24 20	1665	LHLD YK1	
862C AF	1670	XRA A	
862D 32 23 20	1680	STA BYTE	BYTE=ALFA
8630 84	1681	ADD H	A=YK1 (HIGH)
8631 E5	1682	FUSH H	
8632 F2 3B 86	1683	JP R1B	SALT DACA YK1)=0
8635 2F	1684	CMA	
8636 A7	1685	MOV H,A	
8637 7D	1686	MOV A,L	
8638 2F	1687	CMA	
8639 6F	1688	MOV L,A	
863A 23	1689	INX H	HL='YK1'
863B 22 1E 20	1690	R1B SHLD YT	
863E 7D	1695	MOV A,L	
863F D3 2C	1700	OUT 2CH	
8641 7C	1710	MOV A,H	
8642 E6 3F	1720	ANI 3FH	
8644 C6 80	1730	ADI 80H	
8646 D3 2D	1740	OUT 2DH	TIMER=HL
8648 3E C1	1750	MVI A,0C1H	
864A D3 28	1760	OUT 28H	START TIMER
864C E1	1765	POP H	
864D CD B3 88	1770	CALL ARN1	
8650 7C	1771	MOV A,H	
8651 07	1772	RLC	
8652 E6 01	1773	ANI 1	
8654 32 2B 20	1774	STA TEST	ACTUALIZARE VARIABILA "TEST"
8657 22 28 20	1775	SHLD YK	YK=REZULTAT CALCUL (ARN) , CORECTAT
865A 76	1780	HLT	
865B 3A 23 20	1790	R1A LDA BYTE	
865E B7	1800	ORA A	
865F C2 5B 86	1810	JNZ R1A	SALT DACA BYTE()ALFA
8662 3E 23	1820	MVI A,23H	BYTE=ALFA
8664 32 23 20	1830	STA BYTE	BYTE=GAMMA 1
8667 C3 30 20	1840	JMP MAS	
866A	1980	*	
866A	1985	*	SUBROUTINA INTERRUPT DAF
866A	1990	*	
866A	2000	ORG 8700H	
8700 F5	2010	PUSH M	
8701 C5	2020	PUSH B	
8702 E5	2022	PUSH H	
8703 2A 12 20	2025	LHLD ADAD	HL=ADRESA ADRESEI CURENTE
8706 3A 00 70	2030	LDA 7000H	
8709 CD 36 43	2040	CALL 4336H	TRANSFORMA A (DAF) IN B (ASCII)

870C	OE 19	2041	MVI C,25	
870E	OD	2042	BCR C	
870F	C2 OE 87	2043	JNZ *-4	INTIRZIERE CERUTA DE INTERFATA CU DAF-UL
8712	78	2050	MOV A,B	A=CARACTER ASCII RECEPTIONAT
8713	FE 2D	2060	CPI *-*	
8715	C2 23 87	2070	JNZ ED2	SALT DACA NU E "MINUS"
8718	CD 37 44	2080	CALL OU	TRIMITE ECOU CATRE DAF
8718	3E 01	2090	MVI A,1	
871D	32 14 20	2100	STA ADSI	ADSI=1
8720	C3 30 87	2110	JMP ED3	
8723	FE OD	2130	ED2 CPI 13	
8725	CA 35 87	2130	JZ ED1	SALT DACA E "RETURN"
8728	CD 37 44	2140	CALL OU	ECOU CATRE DAF
872B	70	2150	MOV H,B	INSCRIERE IN SIRUL DE CARACTERE
872C	23	2160	INX H	
872D	22 12 20	2170	SHLD ADAD	ADAD=ADAD+1
8730	E1	2180	ED3 POP H	RESTAURARE REGISTRE - FINAL SUBROUTINA
8731	C1	2190	POP B	
8732	F1	2200	POP M	
8733	CB	2210	EI	
8734	C9	2220	RET	
8735		2 25 *		
8735	3E 3D	2230	ED1 MVI A,"="	
8737	CD 37 44	2240	CALL OU	TRIMITE LA DAF "="
873A	36 00	2250	MVI M,0	SFIRIETE SIR DE CARACTERE RECEPTIONATE
873C	3A 14 20	2251	LDA ADSI	
873F	B7	2252	ORA A	
8740	CA 48 87	2253	JZ ED4	SALT DACA ADSI=0
8743	3E 2D	2254	MVI A,"-"	DACA NU
8745	CD 37 44	2255	CALL OU	TRIMITE LA DAF "-"
8748	01 19 20	2260	ED4 LXI B,ADIN	
874B	21 19 20	2270	LXI H,ADIN	
874E	22 12 20	2280	SHLD ADAD	ADAD=ADRESA INITIALA
8751	D5	2290	PUSH D	
8752	CD 34 45	2300	CALL 4534H	TRANSFORMA CIFRELE ZECIMALE (COD ASCII) INCEPIND DE LA
8755		2301 *		ADRESA BC IN HEXA (HL).REVENIREA DIN SUBROUTINA SE FACE
8755		2302 *		LA PRIMUL ZERO (00) INITIALNI.SUBROUTINA AFECTEAZA TOATE
8755		2303 *		REGISTRELE.
8755	E5	2310	PUSH H	
8756	7D	2320	MOV A,L	
8757	CB 6D 45	2330	CALL HOU1	TRIMITE A LA DAF (FORMAT HEXA)
875A	E1	2340	POP H	
875B	D1	2350	POP D	
875C	3E 48	2360	MVI A,"H"	
875E	CD 37 44	2370	CALL OU	TRIMITE LA DAF "H"
8761	CD 41 44	2380	CALL CRLF	CAR RETURN , LINE FEED
8764	7D	2390	MOV A,L	
8765	32 01 20	2400	STA ABCDB	ABCOB=VALOAREA RECEPTIONATA
8768	3E 01	2410	MVI A,1	
876A	32 00 20	2420	STA FAN	FAN=1
876D	C3 30 87	2430	JMP ED3	
8770		2450 *		SUBROUTINA DE INMULTIRE PE 16 BITI
8770	47	2500	IN0 MOV B,A	
8771	EB	2510	XCHG	
8772	3A 21 20	2520	LDA SIGN	
8775	B7	2530	ORA A	
8776	CA 80 87	2540	JZ INH	SALT DACA SIGN=0
8779	7A	2550	MOV A,D	SIGN=1
877A	2F	2560	CMA	
877B	57	2570	MOV D,A	
877C	7B	2580	MOV A,E	
877D	2F	2590	CMA	
877E	5F	2600	MOV E,A	
877F	13	2610	INX D	DE--DE
8780		2612 *		SUBROUTINA INM EXECUTA OPERATIA HL=HL+B*DE
8780	OE 08	2620	INM MVI C,8	
8782	78	2630	IN2 MOV A,B	
8783	1F	2640	RAR	
8784	47	2650	MOV B,A	
8785	D2 89 87	2660	JNC IN3	
8788	19	2670	DAD D	
8789	EB	2680	IN3 XCHG	

B78A 09	2690	DAD H	
B78B EB	2700	XCHG	
B78C 00	2710	DCR C	
B78D 02 82 87	2740	JNZ IN2	
B790 C9	2750	RET	
B791	2760	*	
B791	5980	*	IALOGUL OFF-LINE CU OPERATORUL UMAN
B791	5990	*	
B791	6000	ORG 0C100H	
C100 21 00 D0	6001	LXI H,MES1	ADRESA PRIMULUI MESAJ
C103 0E 0A	6010	MVI C,10	NUMAR DE RINDURI
C105 CD AD 45	6020	SI CALL SCRNI	TRIMITI LA DAF SIRUL DE CARACTERE INCEPIND DE LA ADRESA
C108	6021	*	HL,PINA LA PRIMUL RETURN INTLNIT (ACESTA NU E TRANSMIS)
C108 CD 41 44	6030	CALL CRLF	CAR RETURN , LINE FEED
C10B 23	6040	INX H	SALT PESTE CODUL RETURN
C10C 00	6050	DCR C	
C10D C2 05 C1	6060	JNZ SI	
C110	6065	*	
C110 21 A0 D1	6070	LXI H,MES2	
C113 CD A3 C1	6080	CALL S0	RECEPTIE KR
C116 21 B0 D1	6090	LXI H,MES3	
C119 CD A3 C1	6100	CALL S0	RECEPTIE TI
C11C 01 A0 D1	6110	LXI B,MES2	
C11F CD 34 45	6120	CALL 4534H	CONVERSIE HEXA
C122 22 0C 20	6130	SHLD KR	
C125 01 B0 D1	6140	LXI B,MES3	
C128 CD 34 45	6150	CALL 4534H	CONVERSIE HEXA
C12B 22 0E 20	6160	SHLD TI	
C12E	6190	*	
C12E 06 7B	6200	MVI B,123	
C130 2A 0C 20	6210	LHLD KR	
C133 EB	6215	XCHG	DE=KR
C134 21 00 00	6220	LXI H,0	HL=0
C137 CD 80 87	6230	CALL INM	HL=123*KR (a)
C13A 11 04 00	6240	LXI D,4	
C13D CD 93 C1	6250	CALL IMP	BC=HL/4 (b)
C140 60	6252	MOV H,B	
C141 69	6254	MOV L,C	HL=(b)
C142 22 0C 20	6260	SHLD KR	SALVARE PROVIZORIE
C145 EB	6270	XCHG	DE=(b)
C146 21 00 00	6280	LXI H,0	HL=0
C149 06 05	6290	MVI B,5	
C14B CD 80 87	6300	CALL INM	HL=(b)*5 (c)
C14E EB	6310	XCHG	
C14F 2A 0E 20	6320	LHLD TI	
C152 EB	6330	XCHG	DE=TI
C153 CD 93 C1	6340	CALL IMP	BC=(c)/TI (d)
C156 2A 0C 20	6350	LHLD KR	HL=(b)
C159 09	6360	DAD B	
C15A 22 15 20	6370	SHLD D0	d0=(b)+(d)
C15D 7B	6380	MOV A,B	
C15E 2F	6390	CMA	
C15F 47	6400	MOV B,A	
C160 79	6510	MOV A,C	
C161 2F	6520	CMA	
C162 4F	6530	MOV C,A	
C163 03	6540	INX B	
C164 09	6550	DAD B	
C165 09	6560	DAD B	
C166 22 17 20	6570	SHLD D1	d1=(B)-(d)
C169	6571	*	
C169 0E 04	6572	MVI C,4	
C168 21 00 D2	6575	LXI H,MES4	
C16E CD AD 45	6580	SJ CALL SCRNI	
C171 CD 41 44	6590	CALL CRLF	
C174 23	6600	INX H	
C175 00	6604	DCR C	
C176 C2 6E C1	6606	JNZ SJ	
C179 CD AD 45	6610	CALL SCRNI	
C17C EB	6620	XCHG	
C17D 2A 15 20	6630	LHLD D0	HL=d0
C180 CD B9 C1	6640	CALL S3	

C1B3	23	6650	INX H	
C1B4	CD AD 45	6660	CALL SCRNI	
C1B7	EB	6670	XCHG	
C1B8	2A 17 20	6680	LHLD DI	
C1B8	CD F9 C1	6690	CALL S3	
C1BE	23	6700	INX H	
C1BF	CD AD 45	6710	CALL SCRNI	
C192	C9	6730	RET	REVENIRE IN SISTEMUL DE OPERARE
C193		6740	*	
C193	01 00 00	8000	IMF LXI R,0	SUBROUTINA EXECUTA IMPARTIREA HC=HL/DE
C196	7E	8010	MOV A,E	
C197	2F	8020	LMA	
C198	5F	8030	MOV E,A	
C199	7A	8040	MOV A,H	
C19A	2F	8050	CMA	
C19E	57	8060	MOV D,A	
C19F	13	8070	INX D	
C19D	19	8080	IMI DAD D	
C19E	0D	8090	RNC	
C19F	03	8100	INX B	
C1A0	C3 9D C1	8110	JMP IMI	
L1A3		8120	*	
C1A3	CD 29 44	8170	SO CALL 4429H	ASTEAPTA SOSIREA UNUI CARACTER DE LA DAF, IL TRANSFORMA
C1A6		8171	*	IN R (ASCII)
C1A6	70	8180	MOV M,R	
C1A7	7B	8190	MOV A,R	
C1A8	CD 37 44	8200	CALL DU	ECOU CATRE DAF
C1A8	7B	8210	MOV A,R	
C1AC	FE 0D	8220	CPI 13	
C1AF	24	8230	INX H	
C1AF	62 03 C1	8240	JNZ SO	SALT DACA NU E "RETURN"
C1B2	2F	8250	DCX H	
C1B3	36 00	8260	MOV M,0	SFIRSII SIR DE CARACTERE RECEPTIONATE
C1B5	CD 41 44	8265	CALL CRLF	
C1B8	C9	8270	RET	
C1B9		8275	*	
C1B9	EB	8340	S3 XCHG	SUBROUTINA TRIMITE LA DAF HL, IN FORMAT HEXA SI ZECIMAL
C1BA	E7	8345	PUSH H	
C1BF	7A	8350	MOV A,H	
C1BC	CD 6D 45	8360	CALL HOUT	
C1BF	7E	8370	MOV A,E	
C1CD	CD 6D 45	8380	CALL HOUT	
C1C3	E1	8390	POP H	
C1C4	23	8400	INX H	
C1C5	CD AD 45	8410	CALL SCRNI	
C1C8	23	8415	INX H	
C1C9	E5	8420	PUSH H	
C1CA	21 00 00	8430	LXI H,0	
C1CD	7A	8435	S4 MOV A,D	
C1CE	83	8440	ORA E	
C1CF	CA DF C1	8450	JZ S2	
C1D2	1B	8460	DCX D	
C1D3	7D	8470	MOV A,L	
C1D4	3C	8480	INR A	
C1D5	27	8490	DAA	
C1D6	6F	8495	MOV L,A	
C1D7	7C	8500	MOV A,N	
C1D8	CE 00	8510	ACI 0	
C1DA	27	8520	DAA	
C1DB	67	8525	MOV H,A	
C1DC	C3 CD C1	8530	JMP S4	
C1DF	7C	8540	S2 MOV A,H	
C1E0	5D	8545	MOV E,L	
C1E1	CD 6D 45	8550	CALL HOUT	
C1E4	7B	8560	MOV A,E	
C1E5	CD 6D 45	8570	CALL HOUT	
C1E8	E1	8580	POP H	
C1E9	CD AD 45	8590	CALL SCRNI	
C1EC	CD 41 44	8600	CALL CRLF	
C1EF	C9	8610	RET	

ANEXA 2

```

0000      0001 *
0000      0002 *      Program de reglare numerica a vitezei si pozitie,
0000      0003 *      unui MCC
0000      0005 *
0000      0008 *      Definitie coeficienti si variabile
0000      0009 *
0000      0010 ABAD EQU 2016H adresa curenta pentru receptie DAF
0000      0020 D0 EQU 2016H coeficient D0
0000      0030 D1 EQU D0+1 coeficient D1
0000      0040 D2 EQU D0+2 coeficient D2
0000      0050 D3 EQU D0+4 coeficient D3
0000      0060 D4 EQU D0+6 coeficient D4
0000      0070 YK EQU D0+8 marime de comanda y(k)
0000      0080 YK1 EQU D0+10 marime de comanda y(k-1)
0000      0090 YS EQU D0+12 semn YK
0000      0100 YS1 EQU D0+13 semn YK1
0000      0110 YF EQU D0+14 marime de comanda pentru TIMER
0000      0120 WK EQU D0+16 marime prescrisa w(k)
0000      0130 W1 EQU D0+18 marime prescrisa w(k-1)
0000      0140 RK EQU D0+20 reactie r(k)
0000      0150 RK1 EQU D0+22 reactie r(k-1)
0000      0160 RK2 EQU D0+24 reactie r(k-2)
0000      0180 BYTE EQU D0+28 octet control pentru RST 7.5
0000      0190 FAN EQU D0+29 fanion receptie completa DAF
0000      0200 ADIN EQU D0+30 adresa initiala pentru receptie DAF
0000      0210 HOUT EQU 456DH subrutina OUT DAF (hexa)
0000      0220 OU EQU 4437H subrutina OUT DAF (ASCII)
0000      0230 *
0000      0290 *      Subrutina INTERRUPT TIMER
0000      0295 *
0000      0300 ORG 20CEH
20CE C3 B0 84 0310 JMP II      salt in RAM-ul dinamic
20D1      0315 *
20D1      0320 LI ORG 86B0H
86B0 FB      0330 EI
86B1 3A 34 20 0340 LBA BYTE
86B4 B7      0350 ORA A
86B5 C0      0360 RNZ      return daca byte(>0) , continua pentru byte=0
86B6 D3 29 0365 JUT 29H      blocare tranzistoare
86B8 7C      0380 MOV A,H
86B9 D6 3C 0390 SUI 3CH
86BB 2F      0400 CMA
86BC 67      0410 MOV H,A
86BD 7D      0420 MOV A,L
86BE 2F      0430 CMA
86BF 6F      0440 MOV L,A
86C0 23      0450 INX H      HL=T/2-HL
86C1 7C      0460 MOV A,H
86C2 E6 3F 0470 ANI 3FH      mascare pe 14 biti
86C4 C6 80 0480 ANI 80H      mod de lucru 2
86C6 D3 2D 0490 OUT 2DH      TIMER high
86C8 7D      0500 MOV A,L
86C9 D3 2C 0510 OUT 2CH      TIMER low
86CB 3E C1 0520 MVI A,0C1H
86CD 32 34 20 0530 STA BYTE      byte=0C1H
86D0 D3 28 0540 OUT 28H      start TIMER
86D2 76      0550 HLT      se asteapta in HLT o cerere de intrerupere de la TIMER
E D3 09      0560 RET
86D4      0570 *
86D4      0580 *      Programul principal
86D4      0590 *
86D4      0600 ORG 8600H
86D5 71 00 00 0610 LXI M,0      initializare stiva
86D7 AF      0620 XRA A
86D9 32 C9 20 0630 STA 20C9H      adresa low pentru RST 6.5 este 00
86DB 21 70 20 0640 LXI H,2020H
86DD 06 20 0650 MVI B,32
86DE 77      0660 MOV H,A
86E0 05      0670 DCR B

```

860E	23	0675	INX H	
860F	C2 0C 86	0680	JNZ *-6	initializari variabile
8612	D3 02	0690	OUT 2	programare port 0
8614	D3 03	0700	OUT 3	programare part 1
8616	3E 02	0710	MVI A,2	
8618	D3 0A	0720	OUT 10	programare port B
861A	D3 08	0725	OUT 8	CLEAR numarator
861C	AF	0730	XRA A	
861D	D3 08	0740	OUT 8	
861F	3E 87	0750	MVI A,87H	
8621	32 CA 20	0760	STA 20CAH	adresa high pentru RST 6.5 este 87H
8624	21 36 20	0770	LXI H,ADIN	
8627	22 16 20	0780	SHLD ADAD	
862A	3E 09	0790	MVI A,9	
862C	30	0800	DB 30H	aceasta este instructiunea SIM
862D	C3 26 88	0805	JMP T	
8630		0806	*	
8630		0810	TT ORG 2050H	in RAM-ul static
2050	3E 09	0811	MVI A,9	sint validate RST 6.5 , RST 7.5
2052	30	0812	DB 30H	aceasta este instructiunea SIM
2053	2A 20 20	0813	LHLD YK	
2056	22 22 20	0814	SHLD YK1	yk1=yk
2059	3A 24 20	0816	LDA YS	
205C	32 25 20	0818	STA YS1	ys1=ys
205F		0830	*	preluare pozitie curenta de la numarator
205F	DB 00	0840	CF1 IN 0	
2061	6F	0845	MOV L,A	
2062	DB 01	0850	CP2 IN 1	
2064	67	0855	MOV H,A	
2065	DB 01	0860	IN 1	
2067	BC	0863	CMP H	
2068	CA 6E 20	0866	JZ CP3	
206B	DB 01	0870	IN 1	
206D	67	0875	MOV H,A	
206E	DB 00	0880	CP3 IN 0	
2070	BD	0885	CMP L	
2071	C2 5F 20	0890	JNZ CP1	
2074	22 20 20	0892	SHLD YK	
2077	C3 00 89	0895	K JMP FF1	
207A		0899	*	
207A		0900	FF1 ORG 8900H	
8906	3A 25 20	0905	LDA YS1	
8903	B7	0910	ORA A	
8904	2A 22 20	0920	LHLD YK1	
8907	EB	0930	XLHB	DL=yk1
8908	06 04	0940	MVI B,4	
890A	CA 12 89	0950	JZ FF2	salt daca ys1=0 ; B=4
890D	04 08	0960	MVI B,8	ys1=255 ; B=8
890F	CB AB 20	0970	CALL AIN	DE=-DE
8912	7A	0980	FF2 MOV A,D	
8913	FE 3C	0990	CPI 3CH	
8915	FA 2F 89	1000	JM FF3	salt in registru 1
8918		1005	*	Registru 2
8918	D6 3C	1010	SUI 3CH	
891A	67	1020	MOV H,A	
891B	8B	1030	MOV L,E	
891C	22 24 20	1040	SHLD Y1	y1=yk1-1/2
891F	78	1050	MOV A,B	
8920	D3 29	1060	OUT 29H	comanda tranzistoare
8922	CD 43 87	1070	CALL SBRI	etapa de calcul
8925	2A 26 20	1080	LHLD Y1	
8928	76	1085	HI 1	
8929	CD 08 88	1090	CALL SBRI	etapa de comanda
892C	C3 26 88	1100	JMP I	
892F		1105	*	Registru 1
893F	FE 00	1110	FF3 CPI 0	
8931	C2 3A 89	1120	JNZ FF4	
8934	78	1130	MOV A,E	
8935	B7	1135	ORA A	
8936	C2 3A 89	1140	JNZ FF4	
8939	1C	1150	ENR FF4-JZ	daca yk1=0, atunci DL=1
893A	EB	1160	FF4 XCHG	

893B	78	1180	MOV A,B	
893C	D3 29	1190	OUT 29H	comanda tranzistoare
893E	CD 08 88	1200	CALL SBR2	etapa de comanda
8941	3E 09	1206	MVI A,9	sint validate RST 6.5 ; RST 7.5
8943	30	1208	DB 30H	aceasta este instructiunea SIM
8944	CD 43 87	1210	CALL SBR1	etapa de calcul
8947	76	1215	HLT	
8948	C3 26 88	1220	JMP T	
894B		1240	*	
894B		1242	*	Subrutina INTERRUPT DAF
894B		1244	*	
894B		1250	ORG 8700H	
8700	F5	1260	PUSH M	
8701	C5	1270	PUSH B	
8702	E5	1280	PUSH H	
8703	2A 16 20	1290	LHLD ADAD	HL=adresa curenta
8703	3A 00 70	1300	LDA 7000H	A=caracter emis de DAF
8709	CD 36 43	1310	CALL 4336H	conversie A (DAF) - B (ASCII)
870C	78	1320	MOV A,B	
870D	FE 0D	1330	CFI 13	
870F	CA 34 87	1340	JZ ED1	salt daca e "RETURN"
8712	FE 18	1350	CPI 24	
8714	C2 1D 87	1360	JNZ ED2	salt daca nu e "CLEAR"
8717	CD 37 44	1370	CALL OU	ecou "CLEAR"
871A	C3 2F 87	1380	JMP ED3	
871D	FE 30	1390	ED2 CPI 30H	
871F	DA 2F 87	1400	JC ED3	salt daca e < '0'
8722	FE 40	1410	CPI 40H	
8724	D2 2F 87	1420	JNC ED3	salt daca e > '9'
8727	CD 37 44	1430	CALL OU	ecou cifra zecimala
872A	70	1440	MOV M,B	inscriere in sir
872B	23	1450	INX H	
872C	22 16 20	1460	SHLD ADAD	adad=adad+1
872F	E1	1470	ED3 POP H	final subrutina
8730	C1	1480	POP B	
8731	F1	1490	POP M	
8732	FB	1500	EI	
8733	C9	1510	RET	
8734	3E 3D	1520	ED1 MVI A,'='	
8736	CD 37 44	1530	CALL OU	ecou '='
8739	36 00	1540	MVI M,0	
873B	3E 01	1690	MVI A,1	
873D	32 35 20	1700	STA FAN	fan=1
8740	C3 2F 87	1710	JMP ED3	
8743		1790	*	
8743		1795	*	Subrutina SBR1 (etapa de calcul)
8743		1798	*	
8743	3E BC	1800	SBR1 MVI A,0BCH	
8745	D3 2D	1810	OUT 2DH	
8747	AF	1820	XRA A	
8748	D3 2C	1830	OUT 2CH	TIMER=5msec
874A	3E C1	1840	MVI A,0C1H	
874C	D3 28	1850	OUT 28H	start TIMER
874E	3E 01	1860	MVI A,1	
8750	32 34 20	1870	STA BYTE	byte=1
8753		1875	*	
8753	2A 28 20	1880	ARN LHLD WK	
8756	EB	1890	XCHG	DE=wk
8757	2A 22 20	1900	LHLD YK1	HL=yk1
875A	3A 25 20	1910	LDA YS1	
875D	4F	1920	MOV C,A	C=ys1
875E	3A 18 20	1930	LDA D0	
8761	47	1940	MOV B,A	B=D0
8762	AF	1950	XRA A	A=0
8763	CD 81 20	1960	CALL I24	CHL=CHL+B*ADE
8766	EB	1970	XCHG	
8767	2A 2A 20	1980	LHLD W1	
876A	EB	1990	XCHG	DE=w1
876B	3A 19 20	2000	LDA D1	
876E	47	2010	MOV B,A	B=D1
876F	CD 7A 20	2020	CALL I2M	CHL=CHL-B*ADE
8772	EB	2030	XCHG	

8773	2A	20	2040	LHLD RK	
8774	ER		2050	XCHG	IE=rk
8777	3A	1B	2060	LDA D2+1	A=D2 (high)
877A	B7		2070	ORA A	
877B	CA	B2	2080	JZ AR2	salt data A=0
877E	47		2110	AR1 MOV B,A	
877F	CD	9D	2120	CALL A2M	CHL=CHL-256*B*DE
8782	3A	1A	2130	AR2 LDA D2	
8785	47		2140	MOV B,A	B=D2 (low)
8786	EB		2142	XCHG	
8787	2A	2C	2144	LHLD RK	
878A	ER		2146	XCHG	DE=rk
878B	CD	7A	2150	CALL 12M	CHL=CHL-B*ADE
878E	ER		2160	XCHG	
879F	2A	2E	2170	LHLD RK1	
8792	ER		2180	XCHG	DE=rk1
8793	3A	1D	2190	LDA D3+1	A=D3 (high)
8796	B7		2200	ORA A	
8797	CA	9E	2210	JZ AR3	salt data A=0
879A	47		2220	MOV B,A	
879F	CD	A0	2230	CALL A24	CHL=CHL+256*B*DE
879E	3A	1C	2240	AR3 LDA D3	
87A1	47		2250	MOV B,A	B=D3 (low)
87A2	AF		2260	XRA A	A=0
87A3	CD	B1	2270	CALL 124	CHL=CHL+B*ADE
87A6	EB		2280	XCHG	
87A7	2A	30	2290	LHLD RK2	
87AA	ER		2300	XCHG	DE=rk2
87AB	3A	1F	2310	LDA D4+1	A=D4 (high)
87AE	B7		2320	ORA A	
87AF	CA	B6	2330	JZ AR5	salt data A=0
87B2	47		2360	AR4 MOV B,A	
87B3	CD	9D	2370	CALL A2M	CHL=CHL-256*B*DE
87B6	3A	1E	2380	AR5 LDA D4	
87B9	47		2390	MOV B,A	B=D4 (low)
87BA	EB		2392	XCHG	
87BB	2A	30	2394	LHLD RK2	
87BE	ER		2396	XCHG	DE=rk2
87BF	CD	7A	2400	CALL 12M	CHL=CHL-B*ADE
87C2			2401	*	in CHL se afla rezultatul , corectat
87C2	E5		2405	FUSH H	
87C3	2A	2B	2410	LHLD WK	
87C6	22	2A	2420	SHLD W1	w1=wk
87C9	2A	2E	2470	LHLD RK1	
87D1	22	30	2480	SHLD RK2	rk2=rk1
87D2	2A	2C	2490	LHLD RK	
87D2	22	2E	2500	SHLD RK1	rk1=rk
87D5	E1		2505	POP H	
87D6	79		2510	MOV A,L	
87D7	B7		2520	ORA A	
87DB	FA	EF	2530	JM AR6	salt data C10
87DB	CD	E4	2540	JNZ AR7-3	salt data C10
87DE	3E	77	2550	MVI A,77H	C=0
87E0	BC		2560	CMP H	
87E1	D2	E7	2570	JNC AR7	salt data H(=77H)
87E4	21	F0	2580	LXI H,77FOH	corectie depasire superioara
87E7	22	30	2590	AR7 SHLD YK	yk=HL
87FA	AF		2600	XRA A	
87FB	37	24	2610	STA Y5	ys=0
87EE	C9		2620	RET	
87EF	3C		2630	AR6 INR A	
87F0	CD	F9	2640	JNZ AR8-3	salt data C1255
87F3	3E	B7	2650	MVI A,B7H	
87F5	BC		2660	CMP H	
87F6	DA	FE	2670	JR AR8	salt data H(=B7H)
87F9	21	10	2680	LXI H,B710H	corectie depasire inferioara
87FL	22	30	2690	AR8 SHLD YK	yk=HL
87FF	3E	FF	2700	MVI A,FFH	
8801	32	24	2710	STA Y5	ys=255
8804	3E	0F	2714	MVI A,11	este validata RST 7.5
8806	90		2715	RR X01	accesia este instructiunea SIM
8807	C9		2720	RET	

8808		2725	*		
8808		2790	*	Subrutina SBR2 (etapa de comanda)	
8808		2795	*		
8808	7C	2800	SBR2	MOV A,H	
8809	E6	2810	ANI	3FH	mascara pe 14 biti
880B	C6	2820	ADI	B0H	mod de lucru 2
880D	D3	2830	OUT	2DH	TIMER high
880F	7D	2840	MOV	A,L	
8810	D3	2850	OUT	2CH	TIMER low
8812	AF	2860	XRA	A	
8813	32	2870	STA	BYTE	byte=0
8816	3E	2874	MVI	A,11	este validata RST 7.5
8818	30	2875	DB	30H	aceasta este instructiunea SIM
8819	3E	2880	MVI	A,0C1H	
881B	D3	2890	OUT	28H	start TIMER
881D	76	2900	HLT		
881E	C9	2910	RET		
881F		2980	*		
881F		2990	*	Subrutina de inmultire pe 24 biti	
881F		2995	*		
881F		2999	ORG	N+3	in RAM-ul static
207A	CD	3000	I2M	CALL AIN	DE=-DE
207D	3E	3005	MVI	A,0	
207F	CE	3010	ACI	255	
2081		3015	*		
2081	E5	3020	I24	PUSH H	
2082	2F	3030	MVI	L,B	
2084	E3	3040	I22	XTHL	
2085	F5	3050	PUSH	H	
2086	78	3060	MOV	A,B	
2087	1F	3070	RAR		
2088	47	3080	MOV	B,A	
2089	D2	3090	JNC	I23	
208C	F1	3100	POP	M	
208D	19	3110	DAB	D	
208E	F5	3120	PUSH	M	
208F	B9	3130	ADC	C	
2090	4F	3140	MOV	C,A	
2091	F1	3150	I23	POP M	
2092	EB	3160	XCHG		
2093	29	3170	DAD	H	
2094	8F	3180	ADC	A	
2095	FB	3190	XCHG		
2096	E3	3200	XTHL		
2097	2D	3210	DCR	L	
2098	C2	3220	JNZ	I22	
2099	E1	3230	POP	H	
209C	C9	3240	RET		CHL=CHL+B*AIE
209D		3280	*		
209D		3290	*	Subrutina de inmultire,prin adunari repetate,pe 24 biti	
209D		3295	*		
209D	CD	3300	A2M	CALL AIN	DE=-DE
20A0		3305	*		
20A0	7C	3310	A24	MOV A,H	
20A1	83	3320	ADD	E	
20A2	67	3330	MOV	H,A	H=H+E
20A3	79	3340	MOV	A,C	
20A4	8A	3350	ADC	D	
20A5	4F	3360	MOV	C,A	C=C+B+CY
20A6		3365	*		
20A6	05	3370	DCR	B	
20A7	C2	3380	JNZ	A24	repete de B ori
20AA	C9	3390	RET		CHL=CHL+256*B*DE
20AB		3392	*		
20AB		3395	*	Subrutina pentru complementarea DE, cu posibilitate de	
20AB		3396	*	complementare pe 24 biti	
20AB		3399	*		
20AB	7A	3400	AIN	MOV A,D	
20AC	2F	3410	CMA		
20AD	57	3420	MOV	D,A	
20AE	7B	3430	MOV	A,E	
20AF	2F	3440	CMA		

20B0	5F	3450	MOV E,A	
20B1	B7	3452	ORA A	
20B2	1C	3455	INR E	
20B3	C0	3460	RNZ	CY=0
20B4	14	3465	INR D	
20B5	C0	3470	RNZ	CY=0
20B6	37	3475	STC	CY=1
20B7	C9	3480	RET	
20B8		3490	*	
20B8		3492	*	Preluarea prescrierii de la DAF
20B8		3495	*	
20B8		3496	T ORG SBR2+30	
8826	3A 35 20	3500	LDA FAN	
8829	B7	3510	ORA A	
882A	CA 50 20	3520	JZ TT	salt daca fan=0
882D	AF	3530	XRA A	
882E	32 35 20	3540	STA FAN	fan=0
8831	01 36 20	3550	LXI B,ADIN	
8834	21 36 20	3560	LXI H,ADIN	
8837	22 16 20	3570	SHLD ADAD	
883A	CD 34 45	3580	CALL 4534H	conversie (zecimal ASCII) - (hexa)
883D	E5	3590	PUSH H	
883E	22 28 20	3600	SHLD WK	wk=w(DAF)
8841	7C	3610	MOV A,H	
8842	CD 6D 45	3620	CALL HOUT	ecou wk (high)
8845	E1	3630	POP H	
8846	7D	3640	MOV M,A,L	
8847	CD 6D 45	3650	CALL HOUT	ecou wk (low)
884A	3E 48	3660	MVI A,'H'	
884C	CD 37 44	3670	CALL OU	ecou caracter 'H'
884F	CD 41 44	3680	CALL 4441H	CR,LF
8852	C3 50 20	3690	JMP TT	
8855		3695	*	

BIBLIOGRAFIE

1. Andreiciuc, D., Bogdanov, I., Popescu, V., *Circuite electronice pentru acționarea cu motare pas cu pas a unui robot industrial*, Simpozionul „Robotizarea în industrie”, Timișoara, 29 nov. — 2 dec. 1984, vol. IV, pp. 1—4.
2. Andreiciuc, D., Bogdanov, I., Popescu, V., *Invertor trifazat cu tensiune de ieșire sintetică comandat cu microprocesorul 8085*, Simpozionul „Aplicații ale electronicii industriale”, Craiova, 17—18 noiembrie, 1983, pp. 27—33.
3. 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 Instrumentation, vol. IECI—27, No. 4, nov. 1980, pp. 291—298.
4. Babuția, I., Dragomir, T.L., Mureșan, I., Proștean, O., *Conducerea automată a proceselor*, Editura Facla, Timișoara, 1985.
5. Boado, S., *Commande en boucle fermée d'un moteur pas à pas à l'aide d'un microprocesseur*, Comptes rendus des journées d'électronique et de microtechnique, 19—21 octobre, 1976, EPE Lausanne, pp. 165—176.
6. Bodea, M., Vătășescu, A., Tănase, G., Neagu, S., Năstase, A., Gheorghiu, V., Marinescu, N., *Circuite integrate lineare. Manual de utilizare*, Vol. IV, Editura Tehnică, București, 1985.
7. Boldea, I., *Conducerea după cîmp a mașinilor electrice*, Curs intensiv, Institutul Politehnic „Traian Vuia” Timișoara, Catedra de electrotehnică și mașini electrice, Timișoara, 1987.
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., Popescu, V., Andreiciuc, D., *Comanda prin microprocesor 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.
10. 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.
11. Bogdanov, I., Precup, A., *Three microprocessor families in several electrical drives control*, The 5th NCED, Iași, may 16—17, 1986, pp. C7—C15.
12. Bogdanov, I., Tănase, M.E., Andreiciuc, D., *Instalație electrică pentru acționarea cu motoare pas cu pas a unui robot industrial*, Simpozionul „Aplicații ale electronicii industriale”, Craiova, 17—18 nov. 1983, pp. 14—20.
13. Buja, G., Fiorini, P., *Microcomputer control of PWM Inverters*, IEEE Transactions on Industrial Electronics, vol. IE—29, No. 3, August 1982, pp. 212—216.
14. Călin, S., Dumitrache, I., *Regulate automate*, Editura Didactică și Pedagogică, București 1985.
15. Căpățină, O., Cornea-Hășegan, M., Pușca, M., *Proiectarea cu microprocesoare*, Editura Dacia, Cluj-Napoca, 1983.

16. Coita, V., *Aparat pentru testarea circuitelor integrate tip β AA 145*, Certificat de inovator nr. 937, 04.06. 1987, MIET-IAEM Timișoara.
17. Coita, V., *Comandă numerică pentru un inverter trifazat în punte*, Certificat de inovator nr. 938, 04. 06. 1987, MIET—IAEM Timișoara.
18. Crușu, G., *Microprogrammed, microprogrammable automation*, The 5th NCED, Iași, may 16—17, 1986, pp. C23—C28.
19. Dancea, I., *Microprocesoare. Arhitectură internă, aplicații*, Editura, Dacia, Cluj-Napoca, 1979.
20. 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, Ian./feb. 1983, pp. 113—119.
21. 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.
22. Dewan, S.B., Stroughen, A., *Power semiconductor circuits*, Wiley Interscience Publication, New York, Toronto, Brisbane, 1975.
23. Dordea, T., *Mașini electrice*, Editura Didactică și Pedagogică, București, 1977.
24. Dragomir, T.L., *Regulate automate*, Curs, Litografia IPTV, Timișoara, 1986.
25. 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.
26. 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.
27. Flöter, W., Ripperger, N., *Die TRANSVEKTOR—Regelung für den feld orientierten Betrieb einer Asynchronmaschine*, Siemens-Zeitschrift, vol. 45, No. 10, 1971, p. 761—764.
28. Fransua, A., Măgureanu, R., *Mașini și acționări electrice. Elemente de execuție*, Editura Tehnică, București, 1986.
29. Guțulescu, C., *Instalație electronică pentru modificarea turației unui motor asincron cu metoda $U_f = ct$* , Proiect de diplomă, IPTV Timișoara, 1985.
30. Houpis, C.H., Lamont, G.B., *Digital control systems, theory hardware, software*, McGraw-Hill, International Student Edition, 1985.
31. Ionescu, F., Păcuraru, E., *Mutatoare*, Institutul Politehnic București, Fac. Electrotehnică, Catedra de măsurii și aparate electronice, 1983.
32. Ionescu, T., *Sisteme și echipamente pentru conducerea proceselor*. Editura Didactică și Pedagogică, București, 1982.
33. Iosif, N., Luca, D. M., Udrea-Spenea, M., Popa, E., Boulescu, G., Primejdie, G., *Ti-tistoare și module de putere. Catalog*, Editura Tehnică, București, 1984.
34. Jurca, I., *Sisteme de operare*, Curs, partea I—II, Litografia IPTV, Timișoara, 1984.
35. 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.
36. Kelemen, A., *Acționări electrice* Editura Didactică și Pedagogică, București, 1979.
37. Kelemen, A., Imecs, M., *Electronică de putere*, Editura Didactică și Pedagogică, București, 1980.
38. 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.
39. Kelemen, A., Imecs, M., *Mutatoare*, Editura Didactică și Pedagogică, București, 1978.
40. Kelemen, A., Imecs, M., *Sisteme de reglare cu orientare după câmp ale mașinilor de curent alternativ*, Institutul Politehnic Cluj-Napoca, 1987.
41. Kelemen A., Imecs, M., Matlac, I., Titz, G., *Mutatoare. Aplicații*, Editura Didactică și Pedagogică, București, 1980.
42. Kelemen, A., Rădulescu, M. M., *Closed-loop and microprocessor stepping motor control*, 3-rd International Symposium on Control Systems and Computer Science, Bucharest, 14—16 June, 1979, vol. 2, pp. 593—608.
43. Kuo, B.C., Kelemen, A., Crivii, M., Trifa, V., *Sisteme de comandă și reglare incrementală a poziției*, Editura Tehnică, București, 1981.

44. Kümmel, F., *Elektrische Antriebstechnik*, Springer Verlag, Berlin, 1971.
45. Lungu, M., ș.a., *Echipamente modulare cu semiconductoare de putere*, Vol. 1, MICMUE-CIETA, București, 1970.
46. Lupu, C., Stăncescu, S., *Microprocesoare. Circuite-Proiectare*, Editura Militară, București, 1986.
47. Lupu, C., Țepelea, V., Purice, E., *Microprocesoare. Aplicații*, Editura Militară, București, 1982.
48. Mihăescu, M., *Regulator numeric de turație cu microprocesor*, Proiect de diplomă, IPTV, Timișoara, 1984.
49. Mirbod, A., El-Amawy, A., *A general-purpose microprocessor-based control circuit for a three-phase controlled rectifier bridge*, IEEE Transactions on Industrial Electronics, vol. IE-33, No. 3, aug. 1986, pp. 310–316.
50. Mureșan, T., *Conducerea roboților industriali*, Note de curs, IPTV, Timișoara, 1986.
51. Mureșan, T., Strugaru, C., Stoinescu, R., *Microprocesorul 8080 în aplicații*, Editura Facla, Timișoara, 1981.
52. Olivier, G., Stefanovic, 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.
53. Ohmae, T., Mașuda, T., Kamiyama, K., Tachikawa, M., *A microprocessor-controlled high-accuracy wide-range speed regulator for motor drives*, IEEE Transactions on Industrial Electronics, vol. IE-29, No. 3, aug. 1982, pp. 207–211.
54. Osborne, A., *An introduction to microcomputers*, Vol. 0–1, *Basic Concepts*, Berkeley, California, 1976.
55. Osborne, A., *An introduction to microcomputers*, Vol. 2, *Some real microprocessors*, Berkeley, California, 1979.
56. Osborne, A., *An introduction to microcomputers*, Vol. 3, *Support Devices*, Berkeley, California, 1979.
57. 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.
58. Pfeiffer, I., *Tehnici de modulare în lățime de plus a invertoarelor cu tranzistoare sau tiristoare pentru controlul după timp al motoarelor asincrone*, Referat nr. 2 pentru doctorat, IPTV, Timișoara, 1988.
59. Pfeiffer, I., *Sisteme de control după timp a mașinilor asincrone alimentate prin invertoare cu tranzistoare de putere sau tiristoare*, Referat nr. 1 pentru doctorat, IPTV, Timișoara, 1987.
60. 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.
61. Ponner, I., *Electronică industrială*, Editura Didactică și Pedagogică, București, 1972.
62. Popescu, V., *Electronică industrială*, Curs, vol. 1–2, Litografia IPTV, Timișoara, 1983–1984.
63. Popescu, V., Andreiciuc, D., Bogdanov, I., *Electronică industrială. Indrumător de lucrări de laborator*, Litografia IPTV, Timișoara 1985.
64. 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.
65. Popescu, V., Balaci, I., *Electronică industrială. Indrumător de lucrări de laborator*, Litografia IPTV, Timișoara, 1980.
66. Popescu, V., Bogdanov, I., Andreiciuc, D., *D.C. motor speed adjusting with microprocessor*, Buletinul științific și tehnic al IPTV, Timișoara, Tom 29, Electrotehnică, 1984, pp. 73–76.
67. Precup, A., *Sistem de reglare a vitezei și poziției într-o acționare cu motor de curent continuu comandată cu microprocesor*, Proiect de diplomă, IPTV, Timișoara, 1986.
68. Ramamoorthy, M., *An introduction to thyristors and their applications*, The Mac Millan Press Limited, 1978.
69. Sen, P. Ch., *Thyristor DC Drives*. Library of Congress Cataloguing in Publication Date, Wiley Interscience Publication, USA, Canada, 1981.
70. Seracin, E., *Acționări electrice*, Curs, partea I–II, Litografia IPTV, Timișoara, 1980.

71. Seracin, E., Popovici, D., *Tehnica acționărilor electrice*, Editura Tehnică, București, 1985.
72. Simon, W., *Conducerea numerică a mașinilor unelte* (Traducere din 1. germană), Editura Tehnică, București, 1967.
73. 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.
74. Toacșe, Gh., *Introducere în microprocesoare*, Editura Științifică și Enciclopedică, București, 1985.
75. Trif, N. I., *Contribuții la acționările electrice de curent continuu comandate cu microsis- teme — cu aplicații la roboți*, Teză de doctorat, 1987.
76. Wells, B.H., *Microprocessor control of stepping motor*. Proceedings, Fifth Annual Symposium on Incremental Motors Control Systems and Devices, University of Illinois, SUA, 1972, pp. 150—230.
77. Wehrich, G., *Drehzahlregelung von Gleichstromantrieben unter Verwendung eines Zustands- und Störgrößenbeobachter*, Teil, I, *Regelungstechnik*, 26 (1978), H11, pp. 349—354.
78. Zaks, R., *Introduction to microprocessors*. Sybex Inc., Berkeley, California, 1978.
79. Zaks, R., *Programming microprocessors*. Sybex Inc., Berkeley, California, 1978.
80. *** Conferința Națională de Electrotehnică și Electroenergetică, Craiova, 20—21 sept. 1984.
81. *** 4 th European Conference on Electrotehnics Eurocon'80, Stuttgart, Germany, 24—28 march, 1980.
82. *** Proceedings, International Conference on Electrical Machines, Budapest, Hungary, 5—9 sept. 1982.
83. *** Proceedings. The Fifth National Conference on Electrical Drives, Iași 16—17 may, 1986.
84. *** Proceedings. The Sixth National Conference on Electrical Drives, Timișoara, may 13—14, 1988.
85. *** SDK—85, System design kit. User's manual, Intel. Corporation, 1978.
86. *** Simpozionul „Aplicații ale electronicii industriale”, Craiova, 17—18, nov. 1983.
87. *** 3 rd I FAC Symposium, Lausanne, Switzerland, 12—14 sept. 1983.

CUPRINS

Cap. I. Introducere	5
Cap. II. Sisteme cu microprocesoare utilizate în comanda acțiunilor electrice	7
2.1. Arhitectura standard a unui sistem cu microprocesor utilizat în comanda circuitelor de acționare electrică	7
2.2. Sistem de comandă cu microprocesorul 8035	12
2.3. Sistem de dezvoltare cu microprocesorul 8085	15
2.4. Sistem de comandă cu microprocesorul Z80	17
Cap. III. Utilizarea microprocesoarelor în comanda acțiunilor cu motoare de curent continuu	19
3.1. Probleme specifice cu privire la acțiunile cu mașini de curent continuu	19
3.2. Sisteme de acționare cu motoare de curent continuu alimentate de la redresoare comandate	22
3.2.1. Implementarea pe microsisteme a circuitelor de comandă pentru redresoare comandate	26
3.2.2. Redresor monofazat în punte semicomandată cu microprocesor	26
3.2.3. Redresor trifazat în punte complet comandată cu microprocesor	32
3.3. Sisteme de acționare cu motoare de curent continuu alimentate prin variatoare de tensiune continuă	37
3.3.1. Circuite de comandă pentru chopper realizate cu microprocesor	41
3.3.2. Chopper cu stingere forțată comandat cu microprocesor	41
3.3.3. Chopper cu tranzistoare comandat cu microprocesor	48
Cap. IV. Sistem de reglare numerică a vitezei și poziției într-o acționare cu motor de curent continuu comandată cu microprocesor	
4.1. Structura sistemelor de reglare utilizate	51
4.1.1. Tipuri de structuri ale sistemelor de reglare automată a vitezei și poziției	51
4.1.2. Schemele bloc funcționale ale sistemelor de reglare automată proiectate	53
4.1.3. Procesul reglat	54
4.1.3.1. Modelul matematic operațional al MCC	54
4.1.3.2. Modelul matematic al elementului de execuție	55
4.1.3.3. Modelul matematic al traductorului TIRO și al numărătorului	56
4.1.4. Echipamentul de comandă numerică (ECN)	59

4.2.	Proiectarea algoritmilor de reglare numerică	59
4.2.1.	Proiectarea algoritmică a unui sistem convențional de reglare automată numerică	59
4.2.2.	Discretizarea modelelor matematice ale elementelor de transfer continue	63
4.2.3.	Proiectarea algoritmică a <i>SRA</i> în cascadă	64
4.2.4.	Proiectarea concretă a <i>SRA-V</i>	65
4.2.5.	Proiectarea <i>SRA-VP</i>	68
4.3.	Implementarea algoritmului de reglare numerică a vitezei	70
4.3.1.	Forma discretă finală pentru <i>ARN-V</i>	70
4.3.2.	Sarcinile <i>ECN</i> . Structura <i>ARN-V</i>	71
4.3.3.	Principalele subprograme constitutive ale <i>ARN-V</i>	72
4.3.3.1.	Subrutina de tratare a întreruperilor generate de timer	72
4.3.3.2.	Subrutina <i>MĂSURĂ</i>	75
4.3.3.3.	Subrutina <i>ARN</i> . Calculul mărimii de comandă	76
4.3.3.4.	Dialogul on-line cu operatorul uman — subrutina <i>INTERRUPT-DAF</i>	80
4.3.3.5.	Dialogul off-line cu operatorul uman	81
4.3.4.	Considerații asupra timpului mort total al buclei de reglare	82
4.3.5.	Experimentarea <i>SRA-V</i>	82
4.4.	Implementarea algoritmului de reglare numerică a vitezei și poziției	86
4.4.1.	Forma discretă finală pentru <i>ARN-VP</i>	86
4.4.2.	Funcțiile îndeplinite de microsistem. Structura programului <i>ARN-VP</i>	88
4.4.3.	Principalele subprograme ale <i>ARN-VP</i>	90
4.4.3.1.	Subrutina de tratare a întreruperilor de la timer	90
4.4.3.2.	Subrutina <i>SBR-2</i> pentru generarea intervalului de conducție al chopperului	90
4.4.3.3.	Preluarea poziției curente de la numărător	91
4.4.3.4.	Subrutina <i>SBR-1</i> — calculul mărimii de comandă	91
4.4.3.5.	Dialogul on-line cu operatorul uman. Subrutina <i>INTERRUPT-DAF</i>	95
4.4.4.	Experimentarea <i>SRA-VP</i>	5
4.5.	Controlul curentului din înfășurarea rotorică a motorului	97
Cap. V.	Utilizarea microprocesoarelor în comanda acționărilor cu motoare asincrone	98
5.1.	Probleme specifice cu privire la acționările cu mașina asincronă	98
5.2.	Sistemul de reglare a vitezei motoarelor asincrone prin con-	

vertoare de frecvență indirecte — cu circuit intermediar de curent continuu	103
5.3. Invertoare	104
5.3.1. Invertor trifazat cu tensiune de ieșire sintetică, cu circuit de stingere comun pentru toate tiristoarele	104
5.3.2. Invertor în punte trifazată cu comutație complementară și tensiune de ieșire sintetică	106
5.4. Implementarea pe micro sisteme a circuitelor de comandă pentru invertoare	107
5.4.1. Invertor trifazat cu tensiune de ieșire sintetică și circuit de stingere comun, comandat cu microprocesor	107
5.5. Modificarea turației unui motor asincron prin metoda $U/f = \text{constant}$, cu comanda realizată de microprocesor	112
5.5.1. Implementarea software a comenzilor în sistemul pentru modificarea turației unui motor asincron	114
5.6. Reglarea turației motoarelor asincrone pe baza principiului orientării după câmp	119
5.6.1. Principiul reglajului „orientare după câmp”	119
5.6.2. Sisteme de reglare a turației motoarelor asincrone, pe baza principiului orientării după câmp	121
5.6.3. Sistem de reglare pentru motor asincron comandat cu microprocesorul 8080	125
Cap. VI. Utilizarea microprocesoarelor în comanda acționărilor cu motoare pas cu pas	130
6.1. Probleme specifice cu privire la motorul pas cu pas	130
6.1.1. Modalități de comandă a fazelor <i>MPP</i>	131
6.1.2. Modelul matematic în coordonate reale al <i>MPP</i>	132
6.1.3. Schema bloc de comandă a <i>MPP</i>	134
6.1.3.1. Blocul contactoarelor statice	134
6.1.3.2. Metode de supresare	138
6.1.4. Utilizarea <i>MPP</i> în sisteme de poziționare în circuit deschis	139
6.1.4.1. Amortizarea <i>MPP</i>	139
6.1.5. Utilizarea <i>MPP</i> în sisteme de poziționare în circuit închis	140
6.2. Circuite de comandă pentru <i>MPP</i> cu microprocesor.	141
6.2.1. Sisteme de poziționare în circuit deschis cu <i>MPP</i> comandate cu microprocesor	142
6.2.2. Sisteme de poziționare în circuit închis cu <i>MPP</i> comandate cu microprocesor	147
Anexa 1. Program de reglare numerică a turației unui motor de curent continuu	153
Anexa 2. Program de reglare numerică a vitezei și poziției unui motor de curent continuu	168
Bibliografie	174

EDITURA FACLA

ISBN 973-36-0030-X

Lei 20,50