

INSTITUTUL POLITEHNIC "MILIAN VOIA" TIMIȘOARA
FACULTATEA DE ELECTROTEHNICĂ

Ing. RADU IOAN STOINELCU

CERCETARI PRIVIND CONDUCEREA INTELECTUALA A
ROBOTILOR MUNCII CU MANIPULATOR SI VEDERE
ARTIFICIALA

TEZA DE DOCTORAT

BIBLIOTECA CENTRALĂ
UNIVERSITATEA "POLITEHNICĂ"
TIMIȘOARA

Conducător științific
Prof.dr.ing. TIBERIU MURDĂȘ

- 1986 -

517.397
359 E

C U P R I N S

Prefață	
Capitolul 1. Introducere	1
Capitolul 2. Analiza evoluției sistemelor robotice inteligente	5
2.1. Introducere	5
2.2. Sisteme de roboți integrali	6
2.2.1. Proiectul Institutului Stanford-SHAKY	6
2.2.2. Proiectul Universității Stanford	8
2.2.3. Proiectul "ochi-mână" al Universității Stanford	10
2.2.4. Proiectul de sistem "ochi-mână" al firmei Hitachi	12
2.2.5. Proiectul de robot cu inteligența artificială NTL-1	13
2.2.6. Proiectele Universității Edinburgh	15
2.2.7. Proiectul Universității Tottigham	16
2.3. Cercetări de robotică avansată	17
2.3.1. Asamblarea automată	17
2.3.2. Laboratorul de microautomatizare MIT	17
2.3.3. Laboratorul de cercetare FBI	18
2.3.4. MIT - sistem robotic mic	20
2.3.5. Proiectul de automatizare industrială Stanford	21
2.3.6. Cercetări "ochi-mână" ale Universității Stanford după 1973	22
2.3.7. Laboratorul de propulsie reactivă - Pasadena	23
2.3.8. Laboratorul LAAS - Toulouse	25
2.3.9. Laboratorul firmei Hitachi	27
2.4. Concluzii	
Capitolul 3. Proiectul Experimental Finisirea de Robot Inteligent pentru Cercetare Avansată - P.E.T.E.L.O.A.	
3.1. Concepția generală a proiectului	39
3.2. Robotul mobil cu manipulator - P.E.T.E.L.O.A	40
3.2.1. Subsistemul mecanic efectoz	422
3.2.1.1. Manipulatorul	42
3.2.1.2. Căruciorul	45

3.2.2. Microcalculatorul de comandă imbarcat. . .	46
3.2.2.1. Structura microcalculatorului. . . .	47
3.2.2.2. Modulul de achiziții de date	49
3.2.2.3. Acționarea cu motoare electrice. . .	50
pas cu pas	
3.3. Arhitectura sistemelor de calcul pentru conducerea inteligentă a robotului P.E.T.R.I.C.A.	56
3.3.1. Structura distribuită de conducere și luare a deciziilor.	56
3.3.2. Canalul de comunicație PETRICA-MCFN. . .	60
3.3.3. Canalul de comunicație ROBOTRON-MCFN . .	63
Capitolul 4. Modelele matematice ale robotului P.E.T.R.I.C.A.	
4.1. Introducere în problematica modelelor cinematice	67
4.2. Algoritm de stațare a coordonatelor Hartenberg-Denavit	68
4.3. Modelul cinematic direct	73
4.4. Modelul cinematic invers	77
4.5. Algoritm și program de transformare de coordonate	81
Capitolul 5. Sistemele de vedere artificială ale robotului	
5.1. Introducere în problematica vederii artificiale pentru roboți.	83
5.2. Interfață de conversie analog-numerică . . .	85
cu camera TV	
5.3. Cercetări asupra prelucrării imaginilor și recunoașterii obiectelor	90
5.3.1. Metodologia de lucru	90
5.3.2. Prelucrarea imaginilor numerice . . .	91
5.3.2.1. Operatori de convoluție.	91
5.3.2.2. Netizarea spațială	92
5.3.3. Diferențierea spațială și extragerea contururilor	95
5.3.3.1. Operatorul Roberts	96
5.3.3.2. Operatorul Sobel	96
5.3.3.3. Operatorul Marr-Hildreth	97
5.3.3.4. Operatorul Enechel	98
5.3.3.5. Operatorul Rosenfeld	100
5.3.3.6. Operatorul MacLeod	101

5.3.3.7.	Concluzii rezultate din compararea performanțelor diferiților operatori prezentați.	101
5.3.4.	Sistem de vedere pentru identificarea unei piese mecanice de dimensiuni mici.	103
5.3.4.1.	Obținerea imaginilor binare și izolarea componentelor.	104
5.3.4.2.	Algoritmul de urmărire a conturului	105
5.3.4.3.	Relațiile de calcul ale caracteristicilor	112
5.3.4.4.	Recunoașterea obiectelor și învățarea	118
5.4.	Procedură de prelucrare a informației vizuale pentru o lume a cuburilor marcate cu litere	121
5.4.1.	Calculul mediilor pentru aflarea obiectelor.	121
5.4.2.	Extregerea ferestrei de studiu și calculul histogramei	123
5.4.3.	Izolarea obiectului în rețeaua de studiu	124
5.4.4.	Calculul caracteristicilor	126
5.4.5.	Învățarea obiectelor din imagine	127
5.4.6.	Recunoașterea obiectelor după regula celui mai apropiat k-vecin	130
Capitolul 6.	Experimente "ochi-mână-vehicul" realizate cu robotul P.M.T.H.I.C.A.	
6.1.	Integrarea resurselor robotice	133
6.2.	Organizarea ierarhică a căii efectoare	136
6.2.1.	Nivelul tactic.	138
6.2.2.	Expertul de mișcare	139
6.3.	Integrarea vederii cu manipularea și locomoția	140
6.3.1.	Structurarea mediului de lucru.	140
6.3.2.	Relații matematice pentru coordonarea vederii cu robotul mobil.	142
6.3.3.	Problema calibrării	145
6.4.	Nivelul strategic de comandă.	147
6.4.1.	Nivelul de comandă al acțiunilor elementare	148
6.4.2.	Nivelul sarcinilor robotice simple.	150
6.4.3.	Nivelul sarcinilor robotice complexe.	152
6.5.	Experimente realizate.	152

6.5.1. Experimente de programare prin învățare.	152
6.5.2. Experimente "ochi-mind-vehicul"	153
Capitolul 7. Concluzii finale.	159
Bibliografie	164

P R E F A T A

Preocupările mele în domeniu au ca punct de plecare anul 1973 în care am avut ocazia unui prim contact cu o carte robotică, (intitulată "Roboți integrali"), într-o librărie din Leningrad, URSS.

La cinci ani după acest moment, în 1978 am elaborat primele lucrări cu subiect de robotică care după publicarea în editura "Dacia", Cluj-Napoca mi-au întărit convingerea că domeniul roboticii poate fi abordat chiar dacă era foarte nou și pretențios pe toate planurile - științific, tehnologic și material.

Această convingere m-a determinat să trec la conceperea, proiectarea și realizarea în 1979 a primului robot de laborator (P.E.T.R.I.C.A.) comandat cu microprocesor având posibilități de manipulare și locomoție.

După anul 1981 am beneficiat în perioada elaborării tezei de îndrumarea atentă, competentă și exigentă a prof.dr.ing. Tiberiu Mureșan, cărui pentru sfaturile și observațiile primite și pentru sprijinul moral și încurajările profesionale îi aduc cele mai respectuoase și calde mulțumiri împreună cu întreaga mea stimă și considerație.

Vi mulțumiri aduc conf.Crișan Strugaru care mi-a permis să lucrez în laboratorul condus de doamna-sa și mi-a pus la dispoziție unele echipamente și aparate de care am avut nevoie.

Profesorilor dr.ing.Francișc Kovacs și Nicolae Gheorghiu le dătesc multă recunoștință pentru sfaturile și încrederea pe care mi-au arătat-o constant pe parcursul elaborării tezei.

Autorul mulțumește deasemenea colegilor din catedrele Automatică și calculatoare, Electronică și măsurii pentru înțelegerea și sprijinul acordat, colaboratorilor studenți și tuturilor celor care în diverse moduri au sprijinit realizarea acestei lucrări.

Mulțumesc Luciei Albu și Dorinei Maghețiu care au asigurat calitatea redactării acestei lucrări.

În fine dar nu în ultimul rând aduc prinoseul meu de recunoștință și mulțumire celor care în cursul anului 1984 și după aceea au fost alături de mine susținându-mă moral și material, fără de care finalizarea acestei lucrări nu ar fi fost posibilă.

CAPITOLUL 1

I N T R O D U C E R E

Una din pătrunderile științifice și tehnice majore ale secolului nostru o constituie neîndoiește apariția și dezvoltarea impetuoașă a unei noi clase de mașini cunoscută sub termenul generic robot. Indiferent că ei operează pe sau sub pământ, în apă sau în aer, în spațiul extraterestru sau pe suprafața Lunii ori a lui Marte, având înfățișări extrem de variate, roboții - ca mașini inteligente - au dovedit că sînt indispensabili omenirii și civiliizației.

În România, problema creșterii nivelului tehnico și calitativ al producției prin accelerarea procesului de automatizare, electronizare și robotizare este enunțată în raportul C.C. al P.C.R.

Directivele Congresului al XII-lea al P.C.R. cu privire la dezvoltarea economică-socială a României subliniază că "se va pune accentul pe dezvoltarea industriei de mecanică fină, a producției de roboți și manipolatoare care vor încorpora echipamente și aparatură electronică de înaltă performanță".

Analizînd atent textul citat reiese clar că robotica se va sprijini pe două ramuri industriale: pe industria de mecanică fină care va construi corpul roboților și pe industria electronică care va elabora aparatura de conducere.

De fapt robotica, ca disciplină inginerescă, își are obârșia în sonele de specializare a trei discipline mai vechi: mecanica aplicată pentru părțile mecanice, electronica și fizica aplicată pentru interfețele electromecanice și senzori și în fine știința și tehnica de calcul pentru conducerea în timp real și obținerea de comportamente inteligente.

În special rolul jucat de ultimul domeniu permite formularea unui adevăr fundamental despre robotică: un proiect ingineresc de robotică care nu are în vedere posibilitățile tehnicii de calcul actuale este sortit eșecului. Aproape totul în robotică este întins legat de utilizarea calculatoarelor - atît ca mijloace de proiectare, cît și ca elemente structurale ale proiectelor.

Orice sistem robotic apare mai întîi sub forma unui model abstract ale cărui părți trebuie convertite sub formă de

hardware și software. Modelul abstract poate fi reprezentat sub forma unei diagrame piramidale ca în fig.1.1.

Înainte de implementare întregul sistem robotic este o abstracțiune. Spre vârful piramidei, gradul de abstractizare crește iar spre bază apar sistemele robotice reale sub forma unor realizări particulare fixe. Sus apar conceptele generale ale roboticii care nu sînt legate de vreo construcție particularizată. Obiectivul mijloacelor software în cursul procesului de proiectare este să extindă elementele abstracte de pe nivelul superior cît mai aproape de bază. În stadiul premergător implementării elementele hardware ale sistemului robotic final pot fi reprezentate prin simulări pe calculator. În stadiul final anumite părți sînt convertite în hardware și software, iar testul suprem este confruntarea cu realitatea.

Lucrarea de față își propune să abordeze problematica conducerii inteligente a sistemelor robotice "ochi-mînă-vehi-

cul", prin parcurgerea tuturor nivelelor modelului piramidal anterior prezentat, în dublul scop de a găsi unele soluții teoretice noi dar și a le experimenta apoi într-un mediu de cercetare robotică complet.

În capitolul doi se prezintă o analiză a evoluției sistemelor robotice inteligente din categoriile "ochi-mână", "ochi-vehicul" și "ochi-mână-vehicul", de la începuturi (1973) și până în prezent, prin descrierea rezumativă a 20 de proiecte de sisteme robotice inteligente în majoritate în universități. Formularea a 30 de puncte de concluzii și privirea comparativă asupra structurii și performanțelor acestor sisteme constituie o perspectivă amplă asupra celor 20 de ani de cercetare în robotică avansată și permite încadrarea preocupărilor la joncțiunea manipulării cu locomoția având ca sensor principal vederea robotică.

În capitolul trei se descrie mai întâi concepția generală a proiectului de sistem robotic inteligent "ochi-mână-vehicul", prescurtat P.E.T.K.I.C.A., după care sînt prezentate soluțiile constructive ale subsistemului mecanic efector, partea de manipulare și cea de locomoție împreună cu calcule și algoritmi destinați acționării cu motoare pas cu pas. În calitate de unitate de comandă a robotului este elaborat un microsistem de calcul îmbarcat avînd rolul de expert de mișcare. Sistemul de conducere este completat de alte două sisteme de calcul formînd un ansamblu pe trei nivele ierarhice.

Capitolul patru se ocupă de modelele matematice ale robotului, modelul cinematic direct și soluția cinematică inversă. Rezolvarea face uz de coordonatele Hartenberg-Denavit dar extinde utilizarea lor din domeniul manipulatorilor robotice asupra locomoției. Se elaborează astfel modele matematice cinematice globale care tratează împreună gradele de libertate ale manipulării cu cele ale locomoției. Soluția cinematică inversă este apoi implementată ca algoritm și program.

Capitolul cinci face obiectul descrierii mai întîi a părților de hardware și software pentru interfața cu camera TV. Se prezintă apoi cei mai cunoscuți operatori de filtrare prin convoluție și extragere a contururilor. Algoritmii, programele și interpretarea rezultatelor permit analiza critică sub raportul performanțelor, complexității și timpului de calcul. În continuare se elaborează un sistem de vedere robotică destinat recunoaș-

terii și localizării de piese industriale mici bazat pe algoritmul urmării de contur cu conectivitate 6 și calculul a lo caracteristici. Soluțiile adoptate fac uz de cele mai bune rezultate la nivelul anului 1983 și desigur de unele contribuții personale.

Un al doilea sistem de vedere elaborat, în întregime original, este destinat recunoașterii și localizării cuburilor marcate cu litere ca obiecte de manipulat în cursul experimentelor. Cu această ocazie se demonstrează posibilitatea adaptării și simplificării procedurii la domeniul de aplicație care se soldează cu importante câștiguri sub raporturile performanță/timp și performanță/cost.

În capitolul șase sînt expuse problemele creării unui mediu de dezvoltare robotic în care resursele software sînt chemate să joace un rol primordial și modul cum au fost ele rezolvate într-un caz concret. Tot la cazul concret al robotului mobil se referă tratarea integrării vederii robotice cu manipulara și loconomiă. Deasemenea se prezintă nivelele tactic și strategic de conducere a robotului.

Ultimul capitol reprezintă o sinteză asupra obiectivelor urmărite, concluziile desprinse și evidențiază unele contribuții originale ale autorului împreună cu direcțiile de aplicare posibile în viitor ale sistemelor robotice inteligente "ochi-mînă-vehicul".

CAPITOLUL 2

ANALIZA EVOLUTIEI SISTEMELOR ROBOTICE INTELIGENTE

2.1. Introducere

Sub aspect teoretic și istoric sistemele robotice pot fi considerate ca aparținând unor discipline având literatură, fundamente și dezvoltare proprie ca, inteligența artificială (I.A.) și automatizările industriale (A.I.) amândouă beneficiind de rezultatele și suportul oferit de știința și tehnica de calcul.

Între anii 1960-1973, sistemele robotice s-au dezvoltat separat în cadrul celor două discipline ale inteligenței artificiale și automatizărilor industriale sub denumiri ca sisteme de roboți inteligenți sau integrali și respectiv roboți industriali. În acea vreme distincțiile majore între sistemele aparținând uneia sau alteia din discipline erau în primul rând absența comenzii cu calculator a robotului industrial și insuficiența însestrării senzoriale a aceluiași.

Corectarea acestei stări de lucruri a avut loc în anii 1974-1980 și s-a accentuat odată cu apariția microprocesorului înregistrându-se o dezvoltare rapidă a cercetării și dezvoltării de sisteme robotice cu două aspecte majore :

- (1) elaborarea de noi sisteme de comandă ale roboților bazate pe mini și microcalculatoare, modele matematice ameliorate și tehnica de programare prin limbaj de calculator.
- (2) elaborarea de sisteme senzoriale complexe și noi modalități de integrare a acestora în roboții industriali.

În jur de 1980 faza de dezvoltare (1) se consideră terminată deși nu definitiv tranșată de către cercetători în sensul că s-a atins un nivel bun de aplicabilitate industrială, iar direcția de dezvoltare (2) a continuat și primește și în momentul de față o atenție deosebită. A mai apărut o tendință (3) aceea a introducerii de rezultate și tehnici obținute de inteligența artificială.

În cuprinsul capitolului de față sînt prezentate și comentate cele mai importante lucrări de cercetare în domeniul sistemelor robotice inteligente corespunzînd celor două mari etape mai sus amintite iar la sfîrșit se formulează concluzii personale

asupra evoluției domeniului.

2.2. Sisteme de roboți integrali

Primele lucrări legate de studiul experimental al sistemelor robotice inteligente s-au desfășurat în principalele centre de cercetare a inteligenței artificiale ca Institutul de cercetări și Universitatea Stanford, Institutul de Tehnologie Massachusetts, Laboratorul de propulsie reactivă Pasadena, în S.U.A., Universitatea Edinburgh și Universitatea Nottingham din Marea Britanie precum și Laboratorul de electrotehnică din Japonia (ETL). Lucrările au fost întreprinse ca "proiecte de cercetare avansată" cu finanțarea din fonduri de cercetare științifică ale ministerului apărării al S.U.A. Principalul obiectiv era studiul problemei coordonării "ochi-mână" prin care se avea în vedere crearea de sisteme tehnice capabile de un comportament "sensoriomotor" complex în spațiul tridimensional, sisteme "ochi-mână", sisteme "ochi-mână-urechi", sisteme "ochi-cărucior", și altele.

Pentru aceste sisteme, reputatul specialist în I.A. Nilson H.J. a introdus termenul de robot integral. Prin roboți integrali se înțelegea o structură mecanică comandată cu ajutorul calculatorului, aptă să desfășoare o activitate autonomă de manipulare și/sau deplasare cu ajutorul subsistemelor senzoriale vizual, tactil și de alte tipuri și capabilă de adaptare activă la mediu.

Din punct de vedere al aplicabilității s-au urmărit obiective militare, ale cercetării spațiului cosmic și în cele din urmă și aplicații cu caracter industrial.

2.2.1. Proiectul Institutului Stanford-SHAKY

Shakey /M1 69/ a fost un robot telecomandat prin radio, alimentat cu baterie. Se prezintă (fig.2.1) ca un cărucior autonom pe care se află imbarcată comanda logică, camere de televiziune și un telemetru optic având posibilități de orientare. Locomoția este asigurată de două roți antrenate independent de câte un motor pas cu pas.

Pentru orientarea camerei TV și telemetrului sînt prevăzute alte motoare între care și cele pentru reglarea diafragmei de localizării.

În afară de camera TV și telemetrul există și senzori pentru detectarea eventualelor coliziuni. Dispozitivul de comandă imbarcat pe robot se află conectat prin canal radio numeric la un sistem de calcul SDS-940.

Scopul urmărit de cercetători a fost studierea proceselor de comandă a robotului și interacțiunea cu mediul extern complex în timp real. În vederea efectuării unei categorii largi de sarcini, robotul trebuie dotat cu metode de rezolvare cu caracter general. Acest lucru a însemnat realizarea

într-un sistem unic a acelor caracteristici care în I.A. se abordează în mod obișnuit separat. Funcțiile ce trebuiau îndeplinite de robot au fost grupate în trei clase mari: 1) rezolvarea problemelor, 2) modelare, 3) percepție.

Rezolvarea problemelor pentru Shakey a însemnat determinarea unor acțiuni elementare ca de exemplu rotirea căruciorului sau schimbarea spațiului văzut de cameră. Procesul de obținere a secvenței de acțiuni elementare a fost numit planificare. Cunoașterea de către robot a efectului acțiunilor sale este indispensabilă acțiunii de planificare.

Modelarea înseamnă reprezentarea în calculator a cunoștințelor de bază asupra efectelor produse de îndeplinirea unor acțiuni, cunoștințe cuprinse în modelul lumii exterioare.

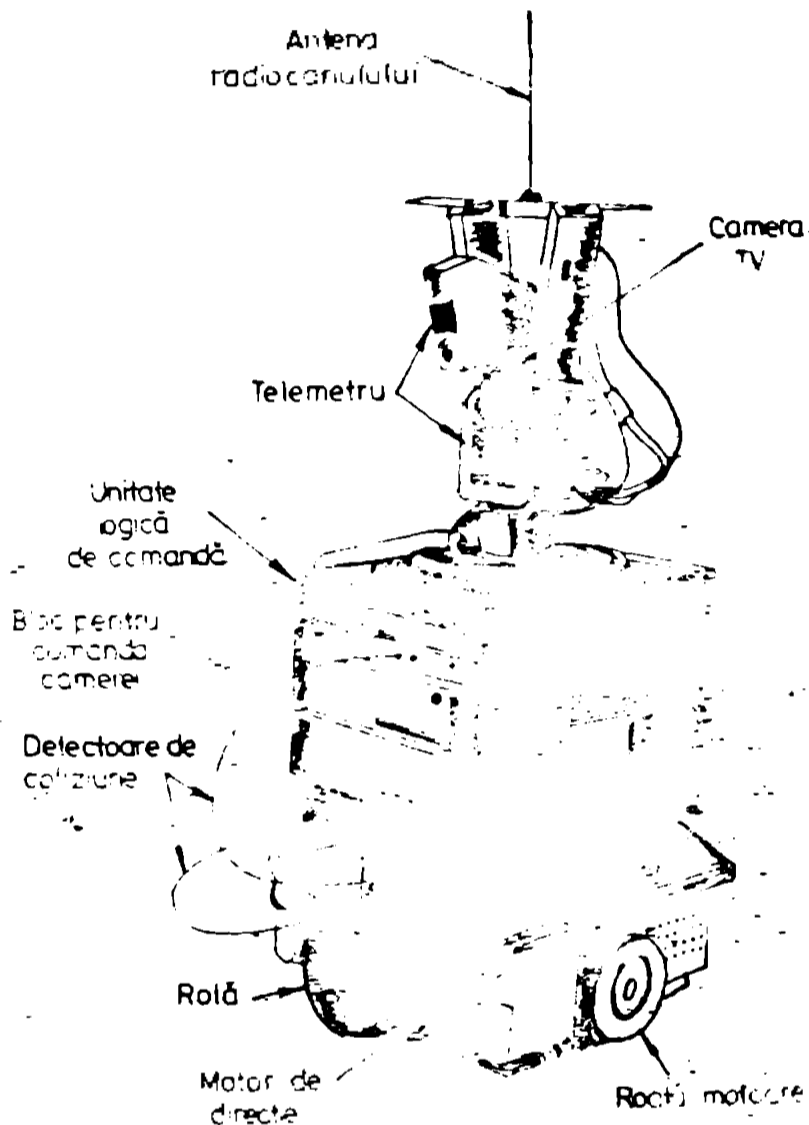


Fig.2.1.

Acest model al lumii reflectă modificările asupra mediului aduse de robot sau percepute prin sistemul senzorial.

Percepția, este considerată vitală servind la obținerea de informații asupra mediului. Desigur cel mai important a fost considerat senzorul vizual dar acesta neputând furniza toate informațiile complete a fost suplimentat cu cel de coliziune.

Shakey posedă câteva nivele de programant pentru a efectua sarcini ca cea de a împinge blocuri de lemn dintr-o cameră în alta. Programul STIPS a fost folosit ca model al lumii împreună cu metodele de demonstrare a teoremelor pentru a determina dacă o secvență de subprograme din bibliotecă (denumite noțiuni de nivel intermediar -ANI) rezolvă corect sarcina.

Experiența dobândită cu ANI a fost preluată de Politehnica din Milano care în cadrul Proiectului de Inteligență Artificială încearcă același lucru pe un calculator UNIVAC 1108 pentru un robot industrial SIGMA în vederea rezolvării procedurilor de corecție a erorilor (error recovery).

2.2.2. Proiectul de robot integral cu "mână, ochi și urechi" al Universității Stanford

În /MC 68/ se descrie un robot integral conceput după autori ca un "calculator având mână, ochi și urechi". Termenii antropomorfici "mână, ochi și urechi" trebuiesc înlocuiți în realitate de manipulator, cameră de televiziune și microfoane.

În intenția autorilor a stat obiectivul efectuării de cercetări fundamentale privind realizarea unui robot-laborator de explorare automată a planetei Marte.

Configurația de blocuri a sistemului de inteligență artificială este dată în fig.2.2. Blocurile au următoarele destinații : 1 - interfață cu teleimprimtoarele, 2- interfață cu display-ul, 3 - sistemul audio, 4- preprelucrarea sunetului, 5 - convertor A/E, 6 - multiplexor de canale analogice, 7-c canale de comandă I/E , 8 - acționarea hidraulică a brațului mecanic; 9 - acționarea electrică a brațului mecanic, 10 - receptor TV, 11 - emițător, 12 - braț mecanic, 13 - cameră TV, 14- disector, 10^5 bit/s, 15 - convertor rapid A/E 10^7 bit/s, 16-canal rapid de date, 17 - memorie externă, 18 - memorie operativă, 19-unitate centrală PDP - 6, 20 - imprimantă, 21- benzii magnetice,

22 - canal
de I/E,
23 - magne-
tofoane,
24 - disc
magnetic.

Sis-
temul de
calcul PLP-
6 a fost
ales pe con-
siderentul
că are posi-
bilitatea
de lucru în
timp dis-

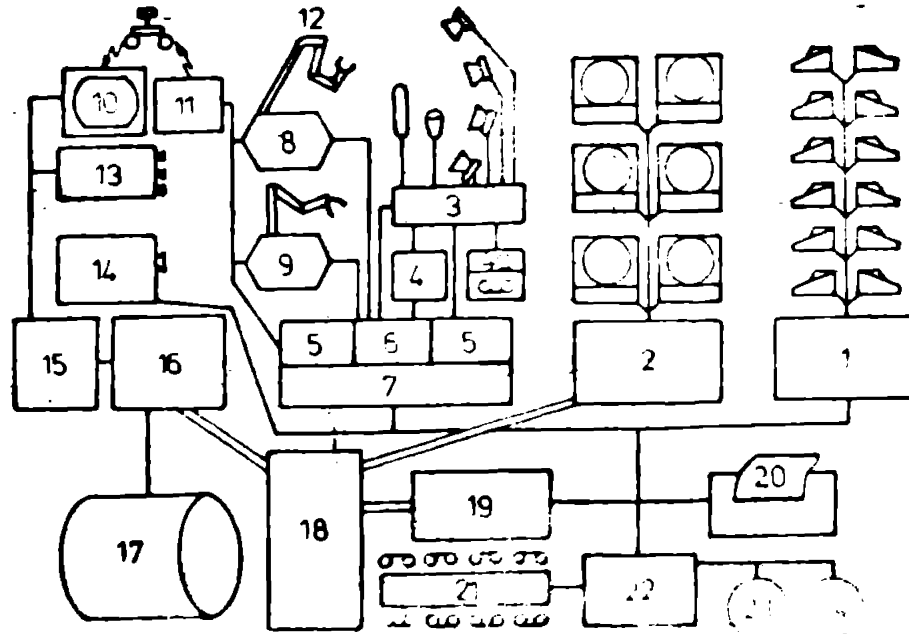


Fig.2.2.

tribuit precum și pentru capacitatea de a realiza transferuri de date pe canalele I/E cu viteze mari de 30 milioane de bit/secundă.

Intrarea vizuală se compune dintr-o cameră TV de tip vidicon și un convertor analog - numeric rapid pe 4 bit (16 gra-
dații de intensitate) cu un debit de 6,5 milioane semne pe se-
cundă. Rezoluția este de 666x 500 de puncte.

Intrarea de sunet este constituită din microfoane și
convertor analog-numeric conectat la calculator. Șantionarea
se face cu viteza de 20000 valori pe secundă și se codifică pe
9 bit.

Brațul mecanic electric a fost inițial proiectat ca pro-
teză pentru paralitici și avea 6 grade de libertate (DDL). Vite-
zele de execuție erau de 4-6 radiani pe minut. Traductorii de
poziție sînt de tip potențiometrici câte unul pentru fiecare ar-
ticulație. Deschiderea maximă a dispozitivului de prebensiune era
de 6,4 cm iar sarcina maximă purtată era de aproximativ 68 N. Ra-
sa maximă de acțiune avea valoarea de circa 70 cm.

Experimentele efectuate s-au purtat în domeniul analizei
și descrierii scenelor vizuale bazate pe lucrările anterioare
ale lui Roberts /No 63/ și Narasimhan /No 66/, Gussman /Cu 67/,
Pingle și alții /PS 68/. Lumea văzută de robot era compusă din cu-
buri.

Mai interesante sînt rezultatele privind comanda mișcări-

lor brațului mecanic. În vederea realizării mișcărilor de manipulare a obiectivelor autorii subliniază importanța activităților de planificare și comandă efectivă a mișcării pe mai multe nivele. Nivelul cel mai înalt îl constituie sarcina (obiectivul) de realizat în cadrul manipulării. Stabilirea sarcinii (task-ului) se face cu concursul dispozitivelor senzoriale în urma cărora se construiește un model al lumii.

De asemenea autorii s-au străduit să elaboreze comenzi date prin limbaj și voce în total 20 de comenzi cu sintaxă corectă. Cu aceste comenzi se puteau aborda aranjamente de cuburi mai mici, mijlocii și mari unul deasupra celuilalt sau unul lângă altul într-o ordine prestabilită.

2.2.3. Proiectul "ochi-mână" al Universității Stanford

În /Pe 69/ se subliniază că principiile elaborării unor roboți integrali nu sînt deplin lămurite de specialiștii în I.A. În acest sens scopul realizării unui proiect "ochi-mână" este construirea unui sistem cu comportament sensoriomotor complex și cu rezolvări în procesul construirii sistemului și componentelor sale suficient de generale pentru a stîrni interes științific.

Sistemul "ochi-mână" construit a avut ca prim obiectiv minimizarea resurselor de aparate și programe necesare. Sistemul capabil de construcții simple în lumea cuburilor sub coordonare vizuală a fost terminat în 1967. (fig.2.3)

Sarcina construirii de "căsuțe" din cuburi a fost formulată pentru prima dată în tehnică. Latura cuburilor era de 6,5 mm maximum. Se menționează că pentru prima oară s-au formulat problemele legate de reacția vizuală în raport cu cea tactilă în cursul fazelor activității robotului. În cadrul proiectului, Pieper a rezolvat pentru prima oară un model matematic al mîinii mecanice.

Pentru percepția vizuală s-a ajuns la conceperea corectă a nivelurilor de prelucrare necesare: prelucrări primare ale intensității punctelor din imagine, identificarea obiectelor din scenă, analiza scenei și mișcării obiectelor. Aceste înseamnă o succesiune de percepții de sensul: puncte-linii-zone-corpuri-obiecte- scenă generală.

Lucrările lui Guzman /Gu 69/ au fost folosite pentru a identifica cuburile ca și ale lui Sobel /So 69/, Roberts /Ro 69/ și Mueckel /Hu 69/. Ele au constituit lucrări fundamentale pentru dezvoltarea vederii artificiale.

Partea centrală a sistemului de calcul era constituită de un minisistem PDP-6 cu o memorie de 131 cuvinte pe 32 de biți având unități de disc cu capete fixe. Calculatorul a fost ales pentru capacitatea de lucru în regim de multiprogramare și debitul ridicat de date între memorie și periferice ($30 \cdot 10^6$ bit/secundă). Intrarea vizuală în sistem cuprindea o cameră TV de tip vidicon. Codificarea nivelelor de gri se realizează pe 4 bit cu o viteză de $6,5 \cdot 10^6$ semne/sec. Rezoluția era de 666×500 de puncte captate din cele două cadre în $1/15$ s.

Mina mecanică cu acționare electrică a fost de tipul Franco Los Amigos, inițial destinată ca proteză, având 6 grade de libertate cu 2 degete. Se puteau obține viteze de 4-6 radiani pe minut. Traducătorii de poziție erau potenționetri. Raza maximă de acțiune era de 70 cm și sarcina maximă 68 N.

În faza finală sistemul ochi-mină a mai fost dotat cu un calculator PDP-10 cu 128 cuvinte conectat la calculatorul PDP-6 inițial.

A fost elaborat limbajul SAIL de către Spruill /SS70/. Limbajul avea la bază ALGOL-ul și era destinat programării majorității proceselor în sistemele "ochi-

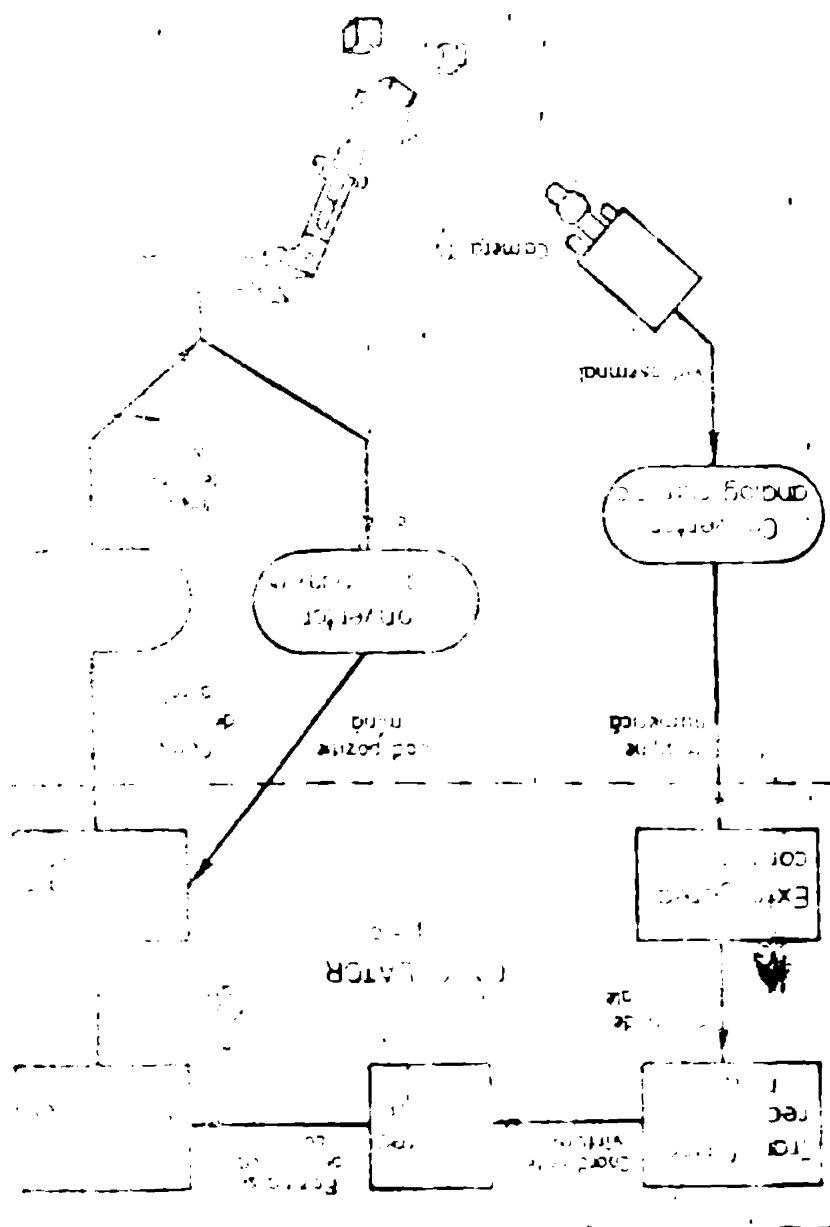


Fig.2.3

mină".

2.2.4. Proiectul de sistem "ochi-mină" al firmei Hitachi

A fost primul proiect de sistem "ochi-mină" având ca obiectiv cercetarea posibilității de automatizare cu roboți a industriei /MT 71/. Proiectul a beneficiat de toate cercetările anterioare adică /Ni 69/, /MC 68/ /Pe 69/ și /Pe 71/ și a adus o importantă contribuție, aceea a precizării sarcinii date robotului prin desen, (fig.2.4)

Robotul primește macroinstrucții sub formă de imagini grafice care indică obiectivul de realizat. Astfel s-a realizat prototipul unui sistem inteligent care pornind de la desenele obiectelor tridimensionale este capabil să

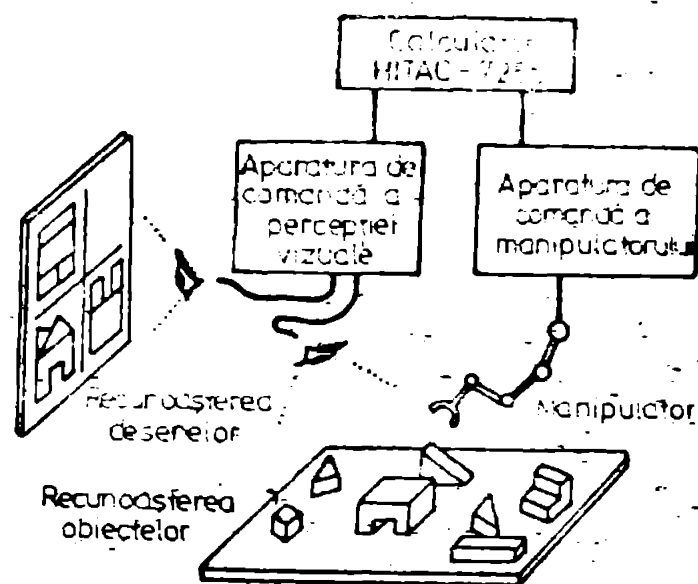


Fig.2.4.

rezolve problemele ridicate de manipulara obiectelor. Robotul are trei subsisteme : ochi, mină, creier. Privind desenul obiectului de asamblat cu un ochi, robotul înțelege configurația spațială, formele, părțile componente, numărul și ordinea acestor părți pentru asamblare. Cu celălalt ochi robotul privește obiectele reale aflate pe masă , le identifică apoi ia decizia de manipulare și trece la lucru. Obiectele erau limitate la poliedre. Configurația sistemului folosit în experimente a fost cea din fig.2.5. Două camere de televiziune cu vidicon operind la 60 cadre/s produc o rezoluție spațială de 240 x 320 pixeli. Conversia numerică se face pe 5 bit (32 nivele) cu un convertor rapid A/E cu ciclul de conversie de 667 ns.

Mină mecanică a robotului era un mecanism articulată cu 7 CDL fiecare comandat independent de șapte servomotoare. Mină și ochii erau conectați prin canale de date la un calculator HITAC

7250 cu 32 cuvinte de 16 bit/cuvint și 512 kcu pe tambur magnetic.

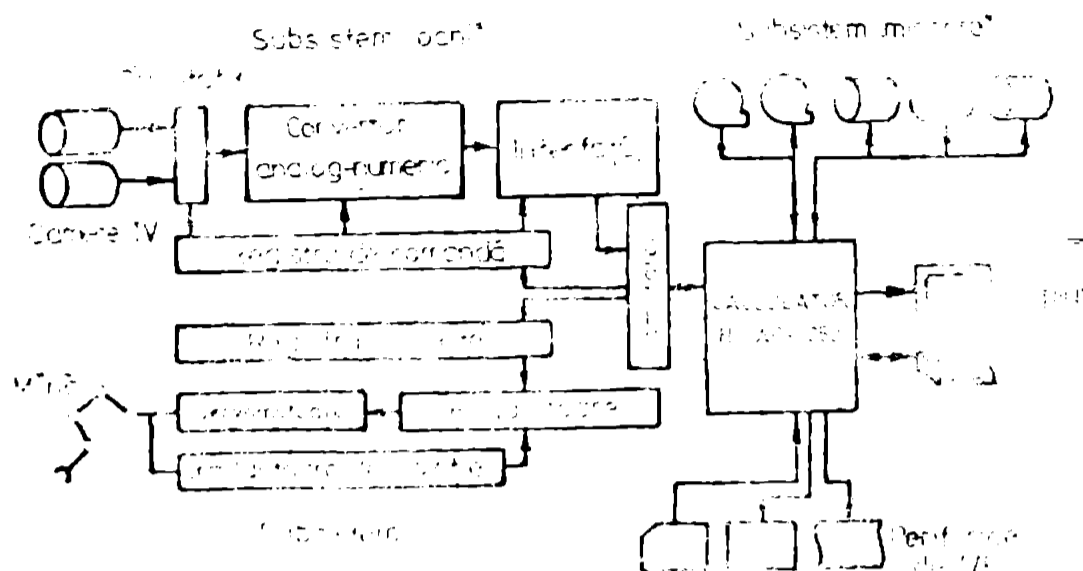


Fig.2.5.

Toate funcțiile de recunoaștere și luarea deciziilor se realizează în continuare prin program (software). Cu excepția unor subprograme scrise în asamblor toate programele erau în FORTRAN și împărțite în trei părți mari: algoritmul de recunoaștere a desenelor, programul de recunoaștere al obiectelor și programul de luarea deciziilor și manipulare. Intregul sistem de programe ocupă 400 cuvinte, incluzând sistemul de operare și zona de memorare a imaginii. Mai bine de 90% din programe erau scrise în FORTRAN. Pentru o prelucrare de imagine erau necesare 240 s, 20 de secunde pentru recunoașterea unui desen, 50 s pentru recunoașterea unui obiect, 10 s pentru luarea unei decizii.

Autorii arată în încheiere că reducerea cu 1/10 sau 1/20 a timpului de calcul se va putea face în viitor pe seama unor structuri de calcul noi bazate pe concepte noi și prelucrare paralelă.

2.2.5. Proiectul de robot cu inteligență artificială ETL-I

Obiectivul avut în vedere de cercetători /ME 71/ a fost elaborarea de roboți cu nivel mai ridicat de inteligență pentru dezvoltarea automatizării industriale. S-a pus accentul pe 1) creșterea mobilității sistemului mecanic, 2) capacității inte-

lectuale adecvate pentru perceperea stării mediului, a găsi-
rii mijloacelor de schimbare a acestuia, a adaptării la con-
diții noi de lucru. În acest scop se indică necesitatea uti-
lizării ultimelor rezultate din teoria comezii, știința și
tehnica de calcul, bionică și mecanică aplicată. Apar și alte
domenii ca informatica, comanda optimă a mecanismelor multi-
legături, sisteme de comandă cu învățare, recunoașterea obiec-
telor tridimensionale, teoria rezolvării problemelor și teoria
limbajelor. Cercetările au început în 1968 și primul raport a
fost publicat în 1971 de către cel mai modern laborator de
electronică din Japonia, Laboratorul de Electrotehnică (ETL).

Dr. Tzuji a elaborat sistemul de vedere iar Dr. Sato și co-
laboratorii sistemul mecanic. Sistemul a fost conceput pentru
dezvoltări ulterioare. Configurația bloc a sistemului "ochi-
mână" apare în fig. 2.6. Sistemul de calcul folosit a fost un
NEAC 3100 cu memorie pe ferită de 32 k cuvinte și pe disc de

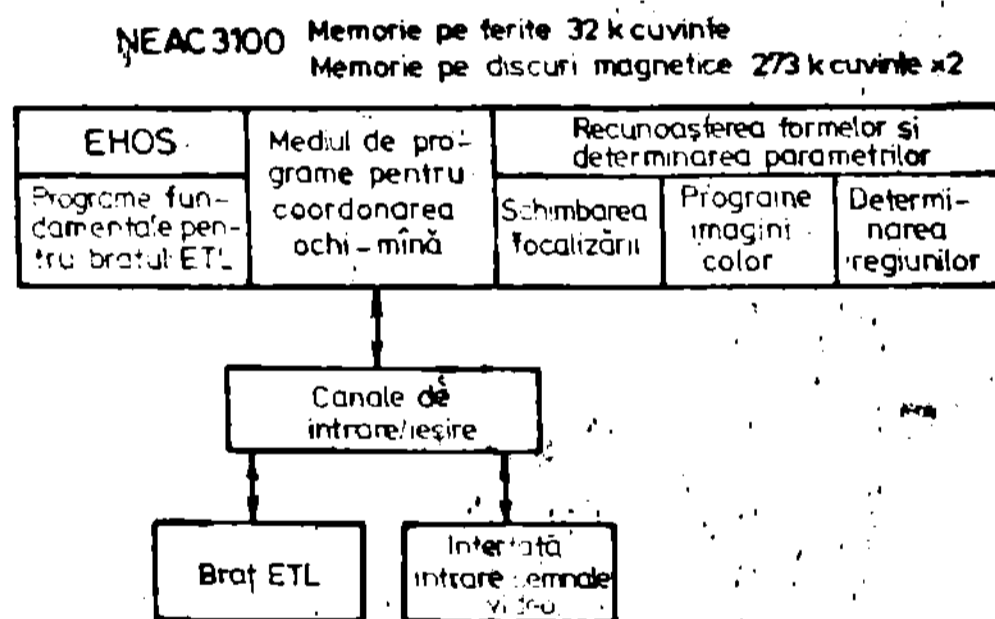


Fig. 2.6.

273 k cuvinte x2. Brațul mecanic având 6 GDL, toate de tip ro-
tație este acționat cu motoare pas cu pas în buclă închisă.

A fost elaborat un sistem de comandă a mișcării manipu-
latorului robotic "ETL" cuprinzând mai multe programe de bază
a căror coordonare se realizează de către programul E OS. Pro-
gramele de bază cuprind programe de transformare de coordonate
generalizate $\theta_1, \theta_2, \dots, \theta_6$ în coordonate operaționale (CONV),

programe de acționare independentă pe fiecare cuplă (MOVE 14, MOVE 15), programe de poziționare și orientare (MOVE 20, MOVE 30), programe de comandă a dispozitivului de prehensiune a robotului (OPLM, CLOS) ș.a. Un program de mișcare în cadrul unei sarcini date robotului reprezintă o secvență de apeluri a subprogramelor amintite.

Inzestrarea senzorială a robotului a permis experimentarea unor procedee de clasificare tactilă. Deasemeni au fost întreprinse experimente ochi-mână folosind obiecte ca paralelipipede, prisme sau cilindri. După achiziția unei imagini, identificarea și localizarea obiectelor s-a putut trece la o activitate robotică de genul celei prezentate mai jos :

```
==== POT ÎNCEPE EXECUTAREA SARCINII?
DA
==== OBIECTUL ESTE UN CUB
==== EXECUT SARCINA
==== POT ÎNCEPE EXECUTAREA SARCINII?
DA
==== PRIVIȚO DIN NOU
==== OBIECTUL ESTE UN CILINDRU
==== SĂTE PE LA PARE PENTRU A-L APUCA
==== VA ROG DATI-MI ALT OBIECT
LA
==== OBIECTUL ESTE O PRISMA
==== EXECUT SARCINA
==== POT ÎNCEPE EXECUTAREA SARCINII ?
NU SPERȘIT
===== SPERȘIT DEPARTEMENT =====
```

Au mai fost realizate experimente cu folosirea reacției vizuale în scopul compensării unor erori în poziționarea și orientarea unui obiect. S-a demonstrat acest lucru în rezolvarea unei sarcini de asamblare a unor cuburi de jucărie cu lățimea de 4,3-4,5 cm. Schema de lucru a fost o strategie bazată pe ciclul "recunoaștere - manevrare" executată repetat.

2.2.6. Proiectele Universității Edinburgh

La universitatea Edinburgh au fost construite două sisteme robotice de laborator de tip ochi-mână, MARK 1 (1971) și MARK 2 (1972) cu o fază intermediară MARK 1,5 /BC 72/. Sistemul

MARK 1 avea o mină cu 3 GDL și 2 GDL pentru platforma de lucru (după concepția lui D. Hilly). Percepția vizuală era asigurată de o cameră TV mobilă cu rezoluția 64x64 pixeli pe 16 nivele. Sistemul MARK 2 (fig.2.7) a constituit dezvoltarea primului avînd ca scop mărirea spațiului de lucru și perfecționarea sistemului vizual. Robotul MARK 2 are 7 GDL și 2 camere TV precum și rezoluția tactilă. Autorii arată că s-au abținut în mod intenționat de la variantele oferite de un robot staționar sau robotul mobil Shakey de la Stanford. Platforma de lucru a lui MARK 1 a rămas în continuare ajun - gîndu-se la o suprafață de lucru de 2 m². Puterea de calcul este asigurată

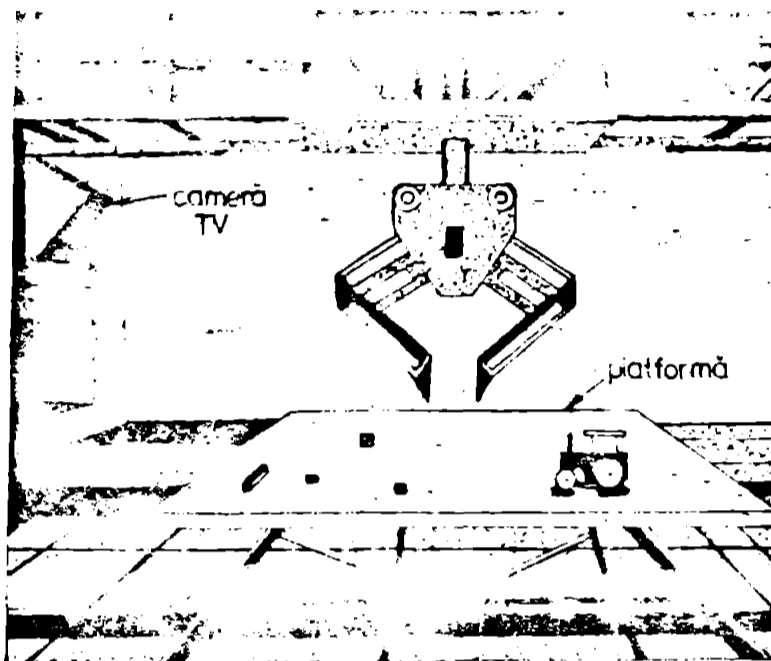


Fig.2.7.

de 2 calculatoare, Honeywell - 316 cu 8Kx16 bit memorie operativă și ICL-4130 cu 24 biți pe cuvânt. În cadrul configurației alese toate așa numitele „periferice” adică platformă (mediu), manipulator, camera TV și procesor video au fost conectate la calculatorul satelit Honeywell. Mediul de programare a fost dezvoltat sub forma unei biblioteci de programe utilizate în regim de multiprogramare. Limbajul folosit a fost POP-2 avînd circa 50 de comenzi pentru asigurarea coordonării cu calculatorul satelit (11), mișcările robotului (10), prelucrarea imaginilor (4), transformări de perspectivă (6). Sistemul vizual folosește o rezoluție de 400 x 300 de puncte cu un interval de eșantionare de 90 ms. Codificarea se realizează pe o scară logaritmică spre deosebire de MARK 1 la care era liniară.

2.2.7. Proiectul Universității Nottingham

Cercetătorii de la Universitatea Nottingham au abordat manipularea pieselor mici, plate folosind un manipulator comandat cu calculator și reacție vizuală, sistem denumit SIRCH /HP 75/. SIRCH poate sorta piese aduse pe un transportor cu bandă. Utilizează imagini binare. Este capabil să determine care e

cea mai bună cale de a apuca o piesă văzută și ca urmare necesi-
tă un calculator mai mic. Accentul s-a pus pe viteza de preluc-
rare vizuală - o sarcină de recunoaștere durează mai puțin de
1 s. Prelucrarea semnalului video ca și comanda se realizează cu
ajutorul unui calculator cu memorie de 8 cuvinte.

2.3. Cercetări de robotică avansată

2.3.1. Asamblarea automată

Abordarea sistemică a problemei asamblării a primit o deo-
sebită atenție începând cu anul 1974 din partea a numeroase grupe
de cercetători în special din Japonia, S.U.A. și unele țări euro-
pene tehnologice avansate. Aria de curățare și obiectivele cer-
cetării întreprinse de japonezi apar clar definite în lucrările
 prezentate la simpozioanele ISIT 1970, 1972 și 1973. La nivelul
anului 1975 progresele înregistrate au permis producerea de robo-
ți cu 30 de firme japoneze în raport cu 15 în S.U.A.

Rezultatele tehnice remarcabile sînt cel mai evident re-
 prezentate de lucrarea lui Sijiri ș.a. prezentată anterior la pa-
ragraful 2.2.4 în care un program efectuează analiza unui desen
tehnic și a unei scene de blocuri dezordonate, dezvoltă apoi o
strategie de asamblare a blocurilor pentru a forma obiectul din
desen și trece cu succes la construcția acestuia. Această lucrare
a rezolvat în principiu toate problemele ce au devenit clasice
pentru cercetarea aplicațiilor inteligenței artificiale (I.A.)
pînă în acel moment. După acest moment efortul s-a concentrat asu-
pra dezvoltării cercetărilor pentru a fi aplicate în mediul lumii
reale cuprînzînd : piese fără formă bloc, obiecte reale și suban-
semble, efectul manipulatorilor imperfecti, îmbunătățirea reac-
ției vizuale, tactile sau de efort - moment, perfecționarea struc-
turilor de calcul.

2.3.2. Laboratorul de microautomatizare-Institutul Tehno- logic Massachusetts

U.C. principală: PDP - 10
Limbeje: LISP, LISP-11
senzori: TV, tactili

U.C. pentru comanda
manipulatorului: PDP -11
manipuloare: 2 VICAR-MIT

Laboratorul de microautomatizare - MIT a utilizat două mi-
nicalculatoare pentru cîteva sarcini de manipulare ochi-mînă, sau
inspecții vizuale /Sp 76/. Au fost dezvoltate modele dinamice in-

517-397
359 E

terecante ale manipuletoarelor. Unul dintre ele realizează par-
tiționarea spațiului de lucru într-o manieră structurată deter-
minist /Fa 76/ și /Ho 76/ în comparație cu metoda aleatoare
CAMAC a lui J. Albus.

Au mai fost elaborate alte tehnici de comandă asemănătoe-
re celor de la Laboratorul de propulsie reactivă din Pasadena
/Bl 76/ precum și experimente de urmărire a unei mingi de ping-
pong iluminată normal ambiental.

2.3.3. Laboratorul de cercetare - IBM /WG 75/

U.C. principală: IBM 370/145

U.C. manipulator: IBM System 7

Sist. de operare: VMOS

Sist. de operare: MOS sub MPS 7

Manipulator: Hidraulic 7 GDL

autodotare

Senzori: Tactili, forță, ultrasunori

Limbaje: ML, AUTOPASS, MAPLS, PL/1, BCPL,

EMILY

Configurația fizică a sistemului este arătată în fig.2.8.
Folul central este îndeplinit de un calculator IBM System/7. Uti-
lizatorul dispune de un terminal cu tastatură, efișaj cu memorie
și o manșă de tip joystick. Un alt terminal oferă conexiunea la
un sistem IBM 370 Model 145. Sistemul/7 este interfațat cu motoa-
re, senzori și instalații înconjurătoare. Manipulatorul a fost
proiectat inițial pentru 12 GDL dar a avut în cele din urmă 7 GDL
plus mobilitatea degetelor. Acționarea este complet hidraulică și
poziționarea se face în buclă închisă de reglare cu o perioadă
de timp de 10 ms. Aceasta consumă circa 80% din timpul de calcul
al lui System/7. Un aspect interesant al organizării programentu-
lui de comandă este folosirea tabelor de numere aflate sub for-
ma unui modul unic încărcabil produs de un program tip asamblor
EMILY rulet în IBM 370/145. În calculatorul System/7 există un
program denumit MOS (sistem de operare manipulator) care interpre-
tează conținutul acestor tabele și mișcă manipulatorul. Întrerup-
torul MOS rulează sub comanda în timp real a sistemului de opera-
re al System/7. Programele pot fi corectate folosind limbajul ML
cuprinzând comenzi pentru : control, secvență, mișcare, senzori,
date, aritmetice, editare, test, calculator-gază. Un exemplu de
program se dă mai jos :

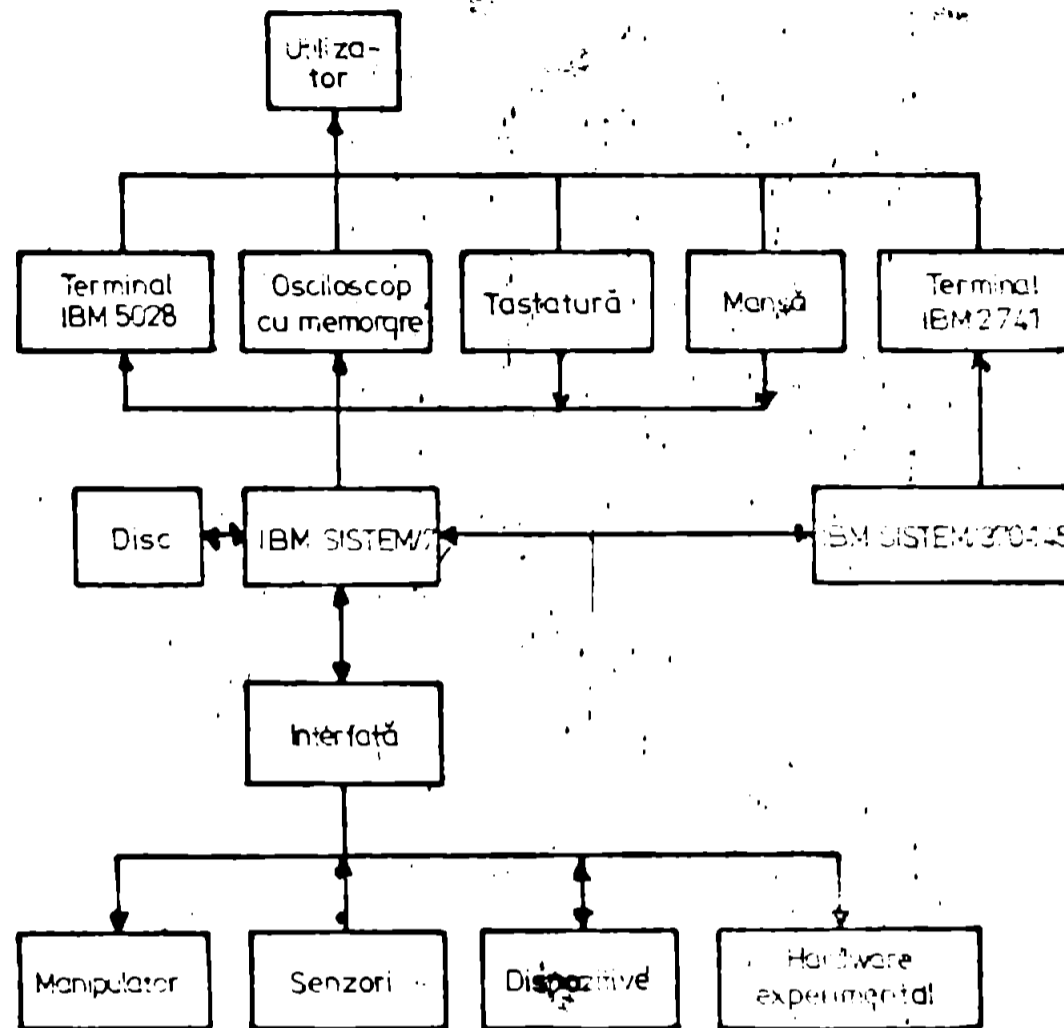


Fig. 2.8.

200 SENSOR 5 - 178 450
210 SENSOR 8 0 1000
215 SENSOR 14 - 800 - 600
280 GOPOINT 5
285 MOVs 0 0 2000 0 0 0 0 4000 - 9000 650 1900
288 MOTOR 3 4000
295 MOTOR 2 - 100
400 BRANCH 500 410 5 14
410 BRANCH 550 600 8
500 etc. ... , etc.

Cu această secvență manipulatorul execută o mișcare în care sensorul 5 e programat pentru limitele (-178, 450) iar ceilalți senzori 8 și 14 au valori mai ridicate. Instrucțiunile 280 la 295 indică deplasarea la punctul 5, urmează apoi o mișcare precizată prin vectorul dat la 285, apoi o mișcare cu motorul 3 numai și în fine o mișcare diferențială pentru motorul 2. Orice poziționare a funcțiilor pentru senzorii 5, 8 sau 14 poate

bloca mișcarea și se continuă la 400 testând fiecare sensor din punct de vedere logic cu saltul necondiționat la instrucția dată.

Limbaajul ML poate funcționa independent de calculatorul gazdă. Aplicațiile pot include piese de la 1 g la 500 g într-un volum de 30 cm³. Cea mai spectaculoasă performanță a fost montarea unui ansamblu suport de car pentru o mașină de scris având 20 de piese inclusiv inserția și înșurubarea unor șuruburi cu o șurubelniță electrică.

Unul din limbajele cele mai interesante elaborat la acest laborator a fost AUTOPASS bazat pe modelarea mediului și planificare. AUTOPASS, MAPLE produc programe preluate de EMILY (ML) și executate de interpretorul MOS.

2.3.4. Institutul de Tehnologie Massachusetts-sistem robotic mic /S1 73/

U.C. PDP - 10

Limbaaj: LISP

Manipulator: 4 GDL, electric
autodotare

Senzori: efort-moment pe 6 GLL
în prehensiune

Laboratorul de I.A. al MIT a construit un manipulator mic și lent denumit "robotul mic" capabil de inserții cu toleranțe de 12 μm și reacție în forță /In 74/. Pe vremea construcției lui exista deja un robot Hitachi aflat în producție care execută inserții cu toleranțe de 20 μm, mai rapid decât operatorul uman.

Tot la acest laborator a fost dezvoltat un limbaj de modelare a asamblării mecanice LAMA. Utilizatorul precizează doar ordinea asamblării părților, apoi LAMA generează secvența de mișcări IA și PUNA și alege punctele de prehensiune pe suprafața obiectelor.

LAMA are nevoie de un sistem de proiectare geometrică asistată de calculator și un limbaj în engleză similar celui folosit de SHEDDU (robotul logic al MIT simulat de Winograd). Strategia de inserție elaborată de Inoue nu atinge performanțele de la Laboratorul Ch. Stark Draper dar punctul forte al lui LAMA îl constituie capacitatea de a executa mișcări constrinse, adică să ocolească obiectele și să-și utilizeze avantajos articulațiile.

Un exemplu de program scris în LAMA reprezentând planul de asamblare al unui piston :

(GRASP OBJ : /PISTON-PIN/)
(PLACE-IN-VISL OBJ : /PISTON-PIN/
SUCH THAT:
(PAR ALLSL /PISTON-PIN//TABLE/))
(UNGRASP OBJ : /PISTON-PIN/)
(GRASP OBJ : /PISTON/)
SUCH THAT:(FACING + (/PISTON/TOP)DOWN))
(INSERT OBJ1: /PISTON-PIN/
OBJ2:/ PISTON-PIN-HOLE/
SUCH THAT:(PARTLY (FITS-IN OBJ1 OBJ2)o,25))
etc. ...

2.3.5. Proiectul de automatizare industrială-Stanford
/Rs 74/

U.C. principală: PLP-11/40, 28Kx16 bit U.C. manipulator:
Manipulatoare: 6 GDL UNIMATE 2000B LSI-11
VICAM - Stanford
Senzori: TV, efort - moment 6 GDL
U.C. pentru dezvoltare programe: DEC KL-10
Sisteme de operare:
LSI -11 : autodotare SII
PLP -11 : KSI - 11/M
PDP -10 : TOPS - 10

Proiectul de automatizare industrială al Centrului de cercetări în I.A. al Institutului Stanford a dezvoltat și demonstrat tehnici de aplicare a manipulării programabile dirijate prin senzori, inspecție și asamblare pentru producția de bunuri industriale de serie :

Aplicațiile includ spre exemplu sortarea, clasificarea și orientarea unor piese metalice turnate aflate într-o cutie , găsirea găurilor într-un bloc motor și inserția guruburilor precum și vopsirea uniformă a obiectelor pe o bandă rulantă ce se mișcă cu viteze variabile.

Lucrările lui Agin /Ag 77/ pentru vederea cu calculatorul descriu tehnici de prelucrare a datelor vizuale și le compară cu sistemele de vedere industrială existente.

S-au mai întreprins cercetări în legătură cu un interpretor de genul FONTAN-ului similar celui elaborat la laboratorul Ch. Stark Draper, Olivetti, IBM și laboratorul de I.A. al Uni-

versității Stanford. El este destinat să ruleze într-un mediu multiprocesor și se pune accentul pe facilitățile de depanare interactivă în cursul executării programului. Deasemeni s-a mai încercat comanda vocală prin intermediul procesorului VIP-100.

2.3.6. Cercetări "ochi-mână" ale Universității Stanford după 1973

U.C. principală: PDP - 10	Limbaje: SAIL, asamblor
U.C. manipulator: PDP -6	Manipulatoare: două VICAR
Senzori: TV, poziția cuple și moment	Stanford

Continuarea cercetărilor începute în anii 60 a condus la elaborarea de modele dinamice ale manipulatorului VICAR de către Paul cu precalcularea forțelor generalizate nominale pentru o traiectorie dată. Folosind algoritmi de reglare numerică de tip convențional, calculatorul PDP-6 corectează deviațiile mici ale brațului mecanic în timpul mișcării. Accidentele și erorile necesită multe secunde pentru replanificarea mișcărilor.

Totodată s-a pus la punct limbajul WAVE /Pa 76/ pentru dezvoltarea de algoritmi complecși de manipulare încercându-se demonstrarea faptului că diferite sarcini se pot atribui aceluiași robot. S-au putut demonstra asamblări de pompă de apă, de ascuțitor de creioane și alte sarcini printre care unele cu folosirea a două manipulatoare și a unor unelte simple. Facilitățile de depanare interactivă au permis dezvoltarea rapidă de noi programe pentru activități noi deși timpii de execuție sînt de 2 la 4 ori mai lungi decît cei necesari unei persoane.

Pentru a scrie un program de comandă a mișcării manipulatorului de definesc macroinstrucțiuni care sînt expandate în secvențe de acțiuni simple ca înșurubarea unui șurub sau apăsarea unei șurubelnițe electrice.

Planificarea obligă mai mult apelul de macroinstrucțiuni decît la subprograme deoarece fiecare apel generează multe informații ce depind de poziția brațului. Suprapunerea de apeluri este permisă și fiecare macroinstrucțiune poate fi testată, extinsă și revăzută individual. Odată macroinstrucțiunile de-

finite de un program de activitate se poate scrie sub forma unei alte macroinstrucțiuni care e o secvență de macroinstrucțiuni definite anterior. MAVO prezintă deficiențe ca executarea pas cu pas și depanarea în cursul execuției (hot editing).

Alte cercetări au avut o înzestrare materială diferită ca mai jos:

U.C. principală: DDC KL-10

U.C. manipulator: PDP-11/45

Manipulatoare: 2 VICARY-S

Senzori: TV, forță-moment 6 GLL

Limbajul AL /si 74/ a fost primul limbaj destinat programării unui robot. Structura limbajului e de tip ALGOL bazată pe blocuri și cu posibilitățile clasice (if... then ... else, case, for, while ... do, do ... until).

Se pot realiza procese paralele prin declarațiile "cobegin" și "coend". Este permisă o aritmetică completă și declarații ale unor tipuri de date precum: SCALAR, VECTOR, ROTATION, PLANE, TRANS. Un exemplu de program apare mai jos:

```
MOVE ARM TO box ;  
CLOSE FINGERS, ON GRASP SENSOR DO STOP ;  
IF FINGERS < 25 = INCHES THEN BEGIN error ... END ;  
APPROX box TO ARM  
MOVE box TO pallet VIA (point 1, point 2)  
ON FORCE = 3 = OUNCES DO STOP ;
```

AL seamănă cu AUTOPASS oferind în plus facilități de depanare interactivă și editare în execuție. Există câteva aspecte investigate la Laboratorul I.A. Stanford (SAIL) care nu au primit suficientă atenție: creșterea vitezei procesului de asamblare, efectul toleranțelor dimensionale asupra probabilității asamblării, metode de calibrare automată pentru senzorii forță și moment.

2.3.7. Laboratorul de propulsie reactivă-Pasadena

U.C. principală: DDC PDP-10

U.C. robot: General Automation SPC-16

Limbaje: FORTRAN, assembler

sistem de operare: REX

manipulator, două TV-uri,
telemetru laser, proximitate
II.

La acest laborator /Le 77/ s-a lucrat la un vehicul fără oameni destinat explorărilor planetare, construcțiilor orbite-

le și misiunilor de reparații a sateliților. Accentul s-a pus pe obținerea de performanțe maxime cu ajutorul sistemului de comandă aflat la bordul robotului, pentru analiza vizuală a scenei, comanda manipulatorului și loconomie intrucit legătura radio cu calculatorul principal (de pe Pământ) poate avea întârzieri mari și limite de bandă. Un număr ridicat de sarcini sînt rezolvate de comanda proprie: navigația vehicolului, căutarea unor roci cu telecamera și telemetrul laser, planificarea mișcărilor brațului pentru a fi colectate și controlul în timp real al motoarelor de acționare. Se mai rezolvă mișcările brațului care sînt planificate pentru a evita coliziunile cu 12 obstacole permanente și cîteva variabile, detectarea unor obstacole noi, transformări de coordonate, calculele de reglare a cuplelor cinematice și coordonarea acestora. Se pot efectua mișcări pe traiectorii segmente de dreaptă, curbilini și aproximativ 20 de puncte de-a lungul fiecărei linii sînt testate pentru evitarea coliziunilor. Precizia de poziționare este mai bună decît 2 mm. Subprogramele de vedere sînt cele mai complexe.

O structură bazată total pe microprocesoare LSI-11 și AN/UYK-30 avînd un caracter evident de comandă ierarhizată decentralizată pentru același robot este arătată în fig.7.9.

Ierarhia operațională este inerentă în cadrul roboților ca în multe alte sisteme complexe. Pe nivelul cel mai coborît calculatoarele primesc programe relativ simple avînd însă un debit de intrare-ieșire foarte ridicat ca de pildă cele pentru reglarea motoarelor manipulatorului. Pe nivelul doi sînt definite subsistemele funcționale ca manipulatorul, navigația și comanda centrală, vederea. Un subsistem integrează funcții expert de genul manipulator, senzori de proximitate, senzori de forță. Pe nivelul superior al comensii și planificării există însă numeroase necunoscute care solicită resurse de calcul mari în vederea definirii strategiei globale, interpretării imaginii, etc.

EXECUTIVUL SISTEMULUI

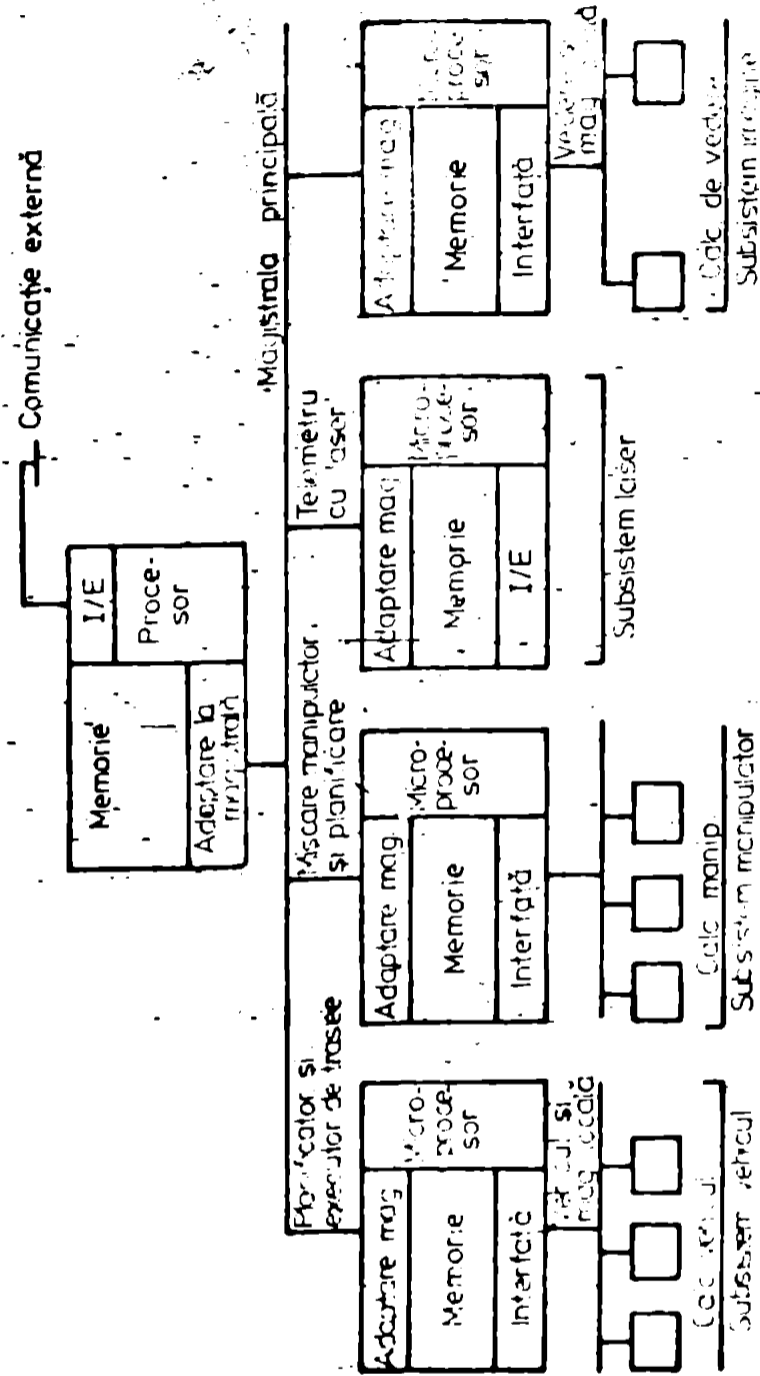


Fig. 2.9.

2.3.8. Laboratorul LAAS-Toulouse

U.C. principală: IBM 370/168 U.C. manipulator: micropro-
 U.C. secundară: MIRA 15 procesoare 8085
 senzori: TV, telemetru laser, detectori ultrasonici, radar

Lucrările conduse la Laboratorul de Automatică și Analize Sistemelor (LAAS) din Franța prezentate în /Ba 79/, /G1 79/, /Pr 80/ a întreprins variate cercetări de robotică avansată, unele cu aplicabilitate imediată iar altele pe termen lung. Ale

au început în 1976 și se desfășoară în continuare. Robotul mobil HILARE (Heuristici integrate în logică și automatismă într-un robot evolutiv) a fost început în 1977 ca suport experimental puternic pentru cercetări în robotică. Concepția lui HILARE, din punct de vedere constructiv preia unele idei de la MARS-ROVER prezentat în paragraful precedent 2.3.7. dar introduce unele aspecte originale. Vehiculul pe 3 roți (fig.2.10) cu acționare independentă cu două motoare pas cu pas este organizat pe trei nivele constructive: locomotie, comandă, senzori. Percepția dezvoltată a lui HILARE a permis abordarea unor probleme de navigație automată cu și fără vedere, organizarea este modulară și nivelul superior de comandă este orientat ca sistem de producții în vederea rezolvării problemelor ridicate de navigație.

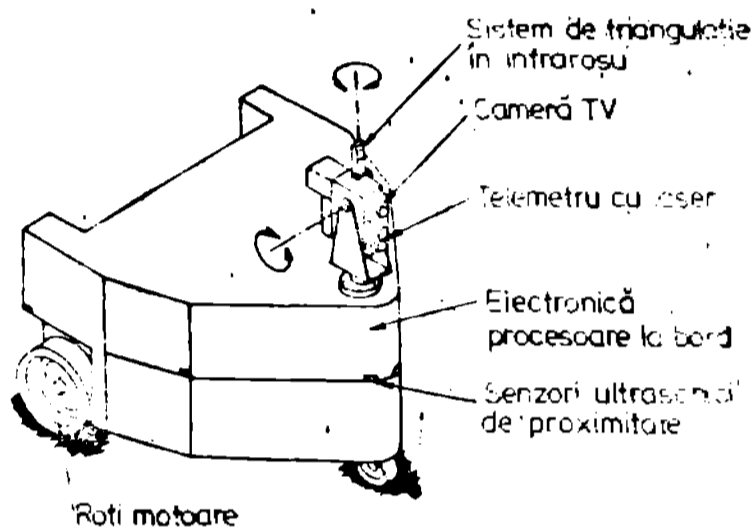


Fig.2.10.

O altă categorie de probleme a fost aceea a rezolvării unor probleme de asamblare (fig.2.11). Au fost folosite în plus față de resursele principale IBM 370/168 și MITRA 15, un microcalculator 8080-A INTEL, există o masă în coordonate XYZ și o cameră TV. Experimentele s-au desfășurat cu cuburi marcate cu figuri geometrice simple. S-au obținut rezultate bune în planificarea traiectoriei, validarea acesteia și facilități de corectare în execuție. O problemă de pla-

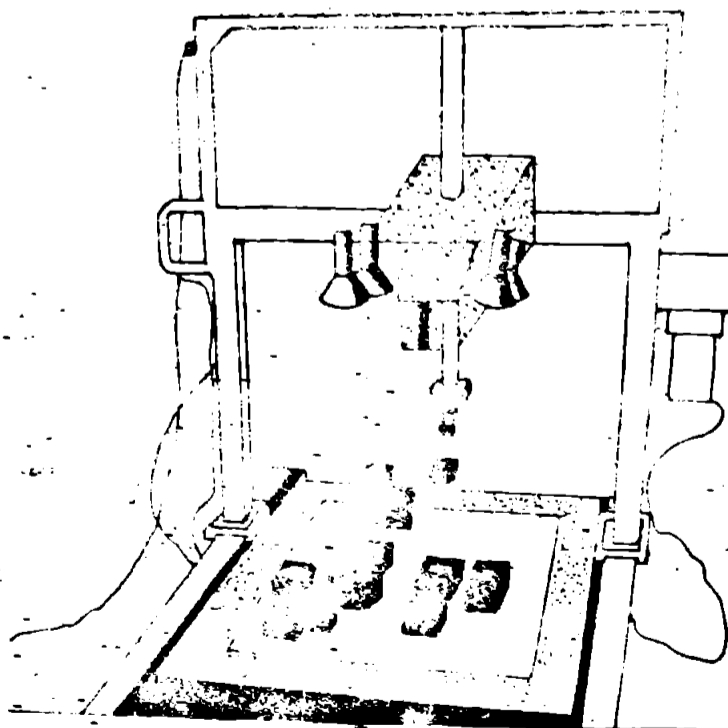


Fig.2.11.

nificare pentru 90 de locații și 26 de cuburi s-a efectuat în 25 de mișcări, cu numai 10 secunde timp de calcul pe IBM 370/168.

2.3.9. Laboratorul firmei HITACHI

U.C. principală: MIDIC-500

U.C. manipulator HI. 10-150

Senzori: 8 camere TV, tactili

manipulatoare: 2 roboți articulați HITACHI-8 GDL

În /Ta 77/ și /Ka 77/ se prezintă o aplicație foarte interesantă realizată în laboratorul firmei Hitachi destinată montajului robotizat al aspiratoarelor de praf electrice. Aspiratorul se compune din trei părți principale: motorul, filtrul și capacul. Fiecare braț robotic dispune de 8 GDL și 3 degete ale dispozitivului de prehensiune prevăzut cu 30 de senzori tactili. Kashioka utilizează 8 camere TV. Trei camere servesc la determinarea poziției pieselor aspiratorului și a posibilităților de prehensiune a acestora. Alte cinci camere TV sînt destinate reacției vizuale în momentul asamblării.

Discretizarea spațială a imaginilor se realizează pe 256x256 de puncte și semnalul video este binarizat cu un prag variabil dependent de luminositatea anumitor părți ale pieselor, iluminarea globală fiind normalizată. Se extrag contururile prin diferențiere spațială și netezire.

Schema bloc a sistemului robotic este dată în fig.2.12.

2.4. Concluzii

În prima perioadă desfășurată între anii 1960-1973 cercetările pentru realizarea de sisteme robotice inteligente s-au desfășurat pe două planuri: primul reprezentat de sistemele robotice cu titlu experimental avînd un pronunțat caracter de cercetare fundamentală, iar al doilea vizînd crearea de sisteme robotice industriale nu neapărat inteligente.

Cursul urmat de aceste cercetări a fost complet independent. Totuși spre sfîrșitul perioadei s-a ajuns la o convergență a obiectivelor care s-a datorat progreselor înregistrate de cercetătorii japonezi. Astfel obiectivele principale urmărite de sistemele robotice inteligente au devenit aplicațiile industriale a rezultatelor obținute în laborator. Asamblarea autome-

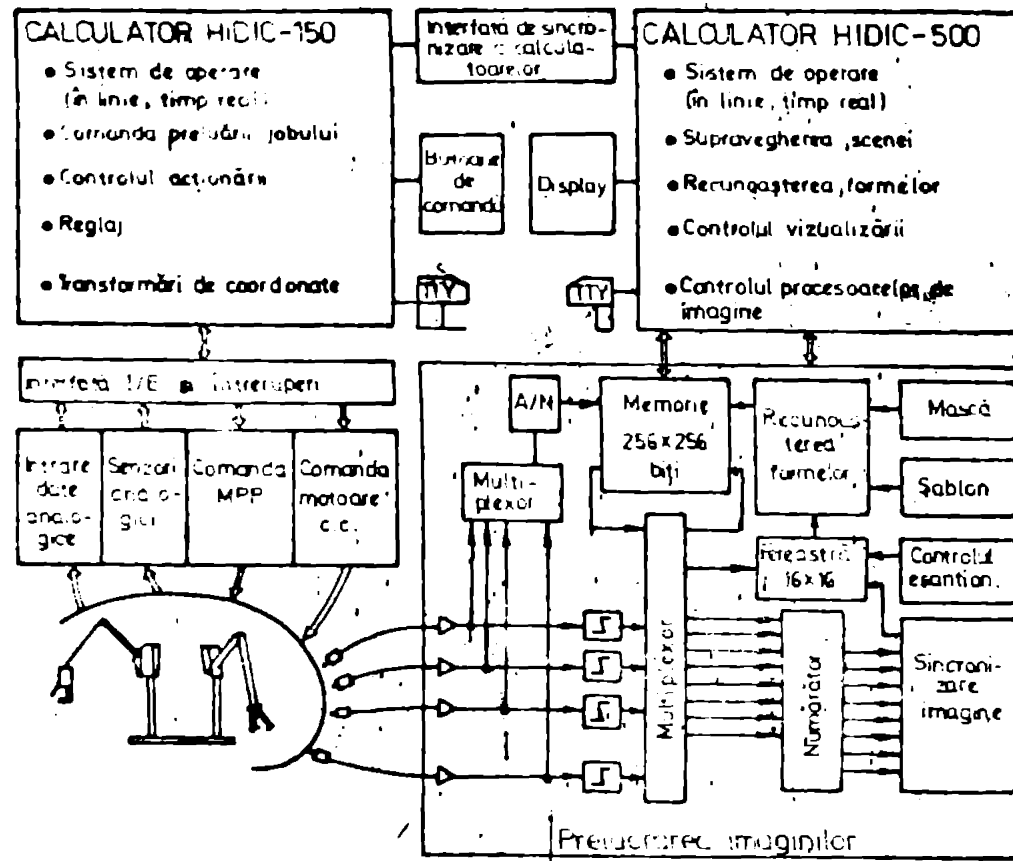


Fig. 2.12

tă cu roboți industriali a primit o atenție deosebită și a ajuns una din categoriile majore de aplicații industriale ale roboticii.

Concepția inițială în cercetarea sistemelor robotice inteligente a purtat amprenta specialiștilor în inteligența artificială și calculatoare. Aceștia "vedeau" lucrurile ca pe o problemă de dotare a calculatorului cu "ochi", "mâini" și "urechi". Într-adevăr, sub raport practic experimental, implementarea sistemelor robotice inteligente s-a realizat prin construirea și conectarea la calculator a unor sisteme mecanice efectoare, senzori vizuali, tactili și/sau de forță-moment, eventual auditivi.

Absolut toate cercetările acestei perioade au avut ca suport sisteme experimentale-roboți de laborator construiți prin autodotare. Sistemele experimentale au fost concepute ca aplicații ale teoriei și tehnicilor inteligenței artificiale. S-a urmărit elaborarea unor noi metode de comandă automată verificate experimental și având suficient caracter de generali-

tate pentru a fi introduse în practică pe o scară cât mai largă. S-a avut în vedere identificarea și delimitarea principalelor domenii de investigație printre care: subsistemul efector, vizual și de comandă inteligentă bazată pe rezolvarea problemelor, planificare și luarea deciziilor.

Rezultatele acestei perioade de cercetări pot fi prezentate pe scurt după cum urmează:

1. Au fost abordate primele sisteme robotice inteligente de tip "ochi-mână", "ochi-cărucior", "ochi-mână-cărucior" sau "ochi-mână-urechi", atât sub raport teoretic cât și practic.
2. Cercetările s-au desfășurat cu precădere în universități, pe sisteme robotice de laborator.
3. S-au făcut primele evaluări ale resurselor de calcul necesare atât sub raport hardware cât și software.
4. Proiectul "ochi-cărucior" al Institutului Stanford a introdus pentru prima dată conceptul organizării ierarhice a comenzii.
5. În tabelul Tab.21 se prezintă principalele caracteristici ale suporturilor materiale destinate experimentelor din care reiese:
 - 5.1. Cu excepția unui singur sistem care folosește două manipulatoare toate celelalte folosesc un singur sistem de calcul. Tendința introducerii unui al doilea sistem de calcul este impusă de necesitățile de calcul sporite ale conducerii manipulatorului robotic.
 - 5.2. Manipulatoarele robotice sînt în majoritate concepute și construite cu mijloace de autodotare ale laboratoarelor respective. Structura cinematică se bazează preponderent pe cuple cinematice de rotație. Unele brațe robotice utilizate au fost adaptări ale unor brațe protetice (brațul mecanic Mancho Los Amigos).
 - 5.3. Sistemele de percepție vizuală au folosit una sau două camere de televiziune de tip vidicon cu posibilități de orientare și unele cu telecomandă. Rezoluția realizată a fost între minimum 120 x 120 pixeli și maximum 400 x 300 pixeli cu 16-32 nivele de cuantizare.
 - 5.4. Acționările manipuletoarelor au cuprins întreaga gamă de la motoare electrice de c.c., motoare pas cu pas și ajungînd la acționare hidraulică.

PROIECTIA DE PROIECTI INTEGRALI ÎN PERIOADA 1960-1973.

Tab.2.1.

TITLUL PROIECTULUI	SISTEMUL MECANIC ACTIONAL LA SI COMANDA	SISTEMUL VIZUAL TACTIL ALFABETIC	SISTEMUL DE CALCUL	SUPORTUL DE PROGRAMAMENT
Institutia de cercetari Stanford- "CHAKAY" (S.U.A.) ochi-crefucior 1964-1969	robot mobil auto- nom, actionare electrica cu MPP independenta pe roti	camera TV imbercuta 120x120 4 bit/pixel	telemetrice	bloc de comanda imbercuta . radiotelemetric . calculator SDS-940
Univ. Stanford (S.U.A.) "ochi-mina" si "ochi-mina-ureche" 1963-1968	manipulator manual Los Angeles (6 QDL) manipulator Stanford (6 QDL) actionare electrica reglare	camera TV fixa, 666x500 (333x500) 4 bit/pixel	microfoane piezo, filtre conv. A/N 9 bit, 20.10 ³ agent./sec.	minicalculator LAMP-limba asociativ PDP-6+periferice dezvoltate specializate pentru robotica conectare cu PDP-10
Laborator central de cercetari HITACHI "ochi-mina" JAPONIA	manipulator 7 QDL (rotatie si pivotare) electric cu bucla inchisa	camera TV 240x320 5 bit/pixel		minicalculator FORTAN, asamblor manual HITAC-7250+periferice, 32 cuvinte

Tab.2.1.(continuare)

1	2	3	4	5	6	7
Laboratorul "TL JAPONIA	manipulator "TL, 6 GDL actionat cu MPP in bucla inchisa	camera TV 6 bit/pixel ferestre programabi- le	pe toate laturile degete- lor	-	minicalculator NoAC-3100, 32 kc.+peri- ferie	POSTMAN, asen- blor monitor EMOS
Univ. din burgh "ochi- mala"	manipulator +platforma mobila MARK 1, MARK 2, (7 GDL)	doua cam- ere TV 400x300	da	-	minicalculator satelit Honeywell-316 + sistem de calcul ICL-4130	PCR-2, biblio- teca de progra- me pentru robot, vedere, transf. de perspective
M. BRITANIA "FRUDLY"	masa XYZ	camera TV	-	-	minicalculator Honeywell DDP 516	necunoscut

- 5.5. Mediul de programare a avut în preponderență compilatoarele FORTRAN și limbaje de asamblare. Au apărut primele sisteme de operare a manipuletoarelor de roboți (GHOS, VALST).
- 5.6. Au apărut primele limbaje specializate de programare a manipuletoarelor robotice ca de exemplu SAIL și POP-2.
6. În mod absolut rapoartele de cercetare publicate se referă mai mult la structura sistemică a roboților inteligenți cu prezentarea sumară a soluțiilor teoretice și practice folosite care sînt doar sugerate.
7. Cadrul cercetărilor a fost de obicei stabilit prin tematici ale unor proiecte subvenționate prin contract. Durata cercetărilor a fost în medie 3-4 ani.
8. Concluzia generală în urma cercetărilor întreprinse a arătat că sînt necesare încă investigații îndelungate pentru înregistrarea unor progrese notabile.

La sfîrșitul acestei perioade un prim transfer de rezultate ale cercetării s-a făcut spre industria constructoare de roboți prin corpul de cunoștințe dobîndite constituit sub numele de comanda cu calculator a manipuletoarelor roboților (computer controlled robot manipulator).

Asupra perioadei imediat următoare desfășurată între 1974-1980, o caracteristică principală o constituie continuarea liniei de cercetare începută în perioada premergătoare pe de o parte și accentuarea cercetărilor destinate aplicațiilor pe termen scurt sau imediat în industrie. În plus eforturile de cercetare întreprinse de firmele constructoare de roboți industriali au condus la apariția de roboți cu structuri mecanice și acționări perfecționate.

Primul robot industrial comandat de calculator a fost realizat de Hahn /Ho 76/. Răspîndirea cercetărilor s-a datorat în primul rînd apariției microprocesorului care a permis abordarea problemelor roboticii cu investiții financiare mai mici în domenii cum sînt comanda roboților, vederea artificială, ș.a.

Principalele trăsături ale cercetărilor acestei perioade pot fi desprinse ca mai jos:

9. Continuarea cercetărilor anterior întreprinse cu precizarea unor obiective cu caracter practic mai pronunțat.
10. Preluarea unor domenii ale cercetării de către industrie

și realizarea primilor roboți industriali susceptibili de perfecționare.

11. O atenție sporită a primit problema coordonării "ochi-mână".
12. Percepția tactilă și reacția de forță-moment a început să fie larg folosită.
13. Au fost elaborate noi structuri de comandă bazate pe micro-procesoare.
14. În tabelul Tab.2.2 se prezintă sintetic cele mai importante experimente cu sisteme robotice inteligente.
 - 14.1. Majoritatea sistemelor robotice inteligente sînt implementate cu un echipament din cel puțin două sisteme de calcul și apoi suplimentar microcalculatoare.
 - 14.2. Se remarcă tendința elaborării de structuri de comandă ierarhizate pe verticală sau pe eșaloane.
 - 14.3. Apare tendința construcției de sisteme "expert" adică realizarea în jurul unuia sau mai multor microprocesoare a unui subsistem funcțional fie el de navigație, manipulare, vedere, telemetrie, etc.
 - 14.4. Problematika "navigației" este clar despărțită de problematica "manipulării" nu neapărat avantajos decît pentru moment.
 - 14.5. Unele sisteme robotice au început să utilizeze roboți industriali pentru operațiile de manipulare cum sînt: UKIMATE 2000 B, MITSUBISHI, VICAM în general avînd 4-6 grade de libertate.
 - 14.6. Se dezvoltă cercetările legate de aplicațiile vederii artificiale și apar primele utilizări industriale sub numele de vederea mașinilor ca ramură a vederii cu calculator.
 - 14.7. Din 14 experimente de robotică avansată, majoritatea 9, sînt de tip "ochi-mână" restul folosesc reacția tactilă și efort-moment.
 - 14.8. Se realizează primele experimente cu coordonarea a două brațe robotice (HITACHI).
 - 14.9. Se realizează primele sisteme pentru inserție-asamblare.
 - 14.10. Mediul de programare continuă să folosească FORTRAN-ul și apare o preocupare sporită pentru implementarea comenzii prin LLSP.
 - 14.11. Continuă eforturile de elaborare a limbajelor destinate algoritimizării proceselor de manipulare și modelării cinemate a manipulării (WAVE, EMILY, AUTOPASS, AL, SAIL, etc.).

PROIECTE DE ROBOTICA AVANSATA INTER 1974-1979 Tab.2.2.

INSTITUTIA SI TIPUL PROIECTULUI	SISTEMUL ELCA-NIC ACTIONARE SI COMANDA	SISTEMUL SENZORIAL VIZUAL	PACTIL	ALTELE	SISTEMUL DE CALCUL	SUPPORT DE PROGRAMANT
1	2	3	4	5	6	7
<p>Laboratorul de cercetare IBM S.U.A. "asamblare automată cu robot"</p>	<p>manipulator acționat hidr.+ electric, 9 GDL în buclă închisă (R,3T, 3K+2F)</p>	-	da	-	<p>minicalculator de proces IBM-7+ sisteme de calcul IBM 370/145</p>	<p>limbaj specializat ML, AUTOPASS, EMILY</p>
<p>Laboratorul LAAS-FRANTA "HILARE" "ochi-cărucior"</p>	<p>complet autonom, radiocombatate în dependent cu MPP</p>	<p>cameră TV Sany VCM 2000, 60x400 pte 2 bit/pixel cu 2 GDL pt. orientare</p>	-	-	<p>ultrasonic, microcalculator telemetru re cu microprocesor 8085+ minicalculator MITA 15+ sisteme de calcul IBM 370/168</p>	<p>asamblor, FORTIAN, LISP</p>
<p>Laboratorul LAAS "ochi-mână"</p>	<p>masă XYZ acționare cu MPP, în buclă deschisă</p>	<p>cameră TV cu 2 GDL pentru orientare</p>	-	-	<p>idem</p>	<p>idem</p>
<p>Laboratorul JPL-S.U.A. "MARS-ROVER" "coordonare ochi-manipulare-locomotivă"</p>	<p>cărucior autonom, plus manipulator</p>	<p>cameră TV cu 2 GDL pentru orientare</p>	-	<p>telemetru laser</p>	<p>microprocesoare interconectate în rețea de tip VDS pt.nave spațiale-13 calculatoare</p>	<p>necunoscut</p>

Tab.2.2. (continuare)

1	2	3	4	5	6	7
Proiectul Stanford de Automatizări Industriale	Unimate 2000 B (6 GDL) Vicarm-Stanford reclare	camera matrice de diode	TV efort moment 6 GDL	-	microcalculator LSI-11 pt.comanda de manipulator, minicalculator PDP 11/40	asemblor, BLISS-11
Proiectul Univ. Stanford 1976	două manipulatoare Vicarm-Stanford reglere	camera TV	TV efort moment	-	PDP-10, PDP-6 Dec KL-10, PDP-11/45	WAVE AL(SAIL)
Institutul de Tehnologie Massachusetts robot-mic	manipulator unidirectional electric, 4 GDL, unidirectional	-	efort, moment 6 GDL	-	minicalculator PDP-10	LISP
Laboratorul de microcomputere bizare Inst. Tehn. Mass.	două manipulatoare Vicarm - MIT	camera TV	tactil	-	minicalculator PDP-10	LISP, LISP-11
Laboratorul Charles S. Draper	manipulator electric, 4 GDL	-	efort moment 6 GDL	-	minicalculator NOVA-2, 32 kc	asemblor, POLTRAN
Laboratorul OLIVETTI ITALIA	două manipulatatoare portel XYZ, actionate cu MPP	-	efort moment 3 GDL	-	minicalculator SABAN, 16 bit/cuv.	limbaj special SIGLA

Teb.2.2. (continuare)

1	2	3	4	5	6	7
<p>Laboratorul HITACHI "coordonare 2 brațe"</p>	<p>manipulator re 7 GDL, Mitsubishi actionare cu MPP și în de c.c.</p>	<p>8 camere TV 256x256 ferestre 16x16</p>	<p>16 senz. tactili</p>	<p>-</p>	<p>minicalculatoare HIDIC-500, HIDIC-150 (16 bit/cuvint)</p>	-
<p>Biroul Națio- nal de Stan- darde S.U.A. "CAMAC"</p>	<p>manipulator Vicarm-Stanford</p>	-	-	<p>proxi- mirati cu in- fraroqu</p>	<p>PDP 11/45</p>	-
<p>Universitatea Rhode-Island</p>	<p>industrial manipulator 6 GDL(3T+3R)</p>	<p>camera cu matrice de diode</p>	-	-	-	-

O privire generală asupra celor peste 20 de ani de cercetare în robotică pentru elaborarea de sisteme robotice inteligente ne permite să desprindem principalele subdomenii în care se desfășoară lucrările de bază. Acestea sînt: Reprezentare și modelare, Senzori, Manipulare, Locomoție, Suprastructura inteligentă, Integrare și aplicații.

Un al doilea aspect care se poate desprinde este numărul roboților "compleți" care există în momentul de față în laboratoare. Este surprinzător faptul că roboții de acest tip capabili și de manipulare și de locomoție avînd o înzestrare senzorială suficientă și dotați cu algoritmi și programe de I.A. sînt foarte puțini. Aceștia sînt în număr de patru: Shakey, MARS-ROVER, Jason și HILARE.

Shakey, MARS-ROVER și HILARE au fost prezentați la paragrafele 2.2.1, 2.3.7 și 2.3.8. Jason a fost construit la Universitatea Berkeley și primea ordine ca și Shakey prin radiocanal. Era prevăzut cu limbaje ca FORTRAN și LISP. Simțurile sale includeau o cameră TV, captatori de contact, traductoare de proximitate în infraroșu și cu ultrasunete. Jason a fost dotat cu un braț mecanic de tip proteză, niciodată conectat și cu un sintetizor de cuvinte. El a permis testarea a două generatoare de plan funcționînd într-un univers sigur și unul nesigur.

Unul din motivele principale care explică raritatea roboților compleți este acela că relația decizie-acțiune este greu de stăpînit și că în plus ea este foarte interactivă. Interactivitatea nu apare în roboții simulați prin software care se mulțumesc cu rezolvarea problemelor deconectați de lumea materială. De menționat că un robot complet costă mai mult decît unul simulat ultimul consumînd doar timp de calculator.

Un al treilea aspect îl constituie căile de ajungere spre sisteme robotice cît mai inteligente. Există două astfel de căi: a) dotarea robotului cu simțuri artificiale ale vederii și tactilului, eventual alte tipuri de senzori; b) introducerea tehnicilor I.A. ultima fiind controversată de prima cale. Mai clar, aceasta înseamnă că întîi să se dezvolte capacitățile senzoriale ale roboților și abia după atingerea unui stadiu considerat suficient să se treacă la introducerea tehnicilor I.A.

Sintetizînd lucrurile, subliniem în cele ce urmează că sîntem interesați într-un tip de robot-complet, avînd posibilități multiple atât de manipulare cît și de locomoție precum și

e înregistrare senzorială corespunzătoare.

Cel mai puternic tip de senzor robotic este vederea artificială sau vederea robotică (V.R.). Vederea este necesară când se manipulează materiale, se efectuează operații de inspecție, când se recunosc piese sau când se efectuează operații de asamblare sau inserție pentru a corecta poziția brațului. Deocamdată senzorii tactili nu au atins perfecțiunea celor vizuali.

Pe de altă parte problemele manipulării robotice și locomoției au fost și sînt în continuare tratate separat în cadrul amintitelor subdomenii Manipulare și Locomoție. Există însă o resursă potențială pentru reunirea gradelor de libertate puse la dispoziție de partea de manipulare împreună cu partea de locomoție pentru a spori capacitățile de manipulare.

Un astfel de robot ar putea servi ca robot de întreținere sau eventual o grupă de astfel de roboți aflată la dispoziția unui dispecer automat ar putea fi trimisă pentru a desfășura activități diferite în diferite locuri. Tipul acesta de robot s-ar putea deplasa prin navigație autonomă pînă la locul de muncă unde s-ar transforma prin docare la sol într-un robot semistaționar sau ar rămîne mobil.

Tot în continuare, cu unul sau două brațe el ar urma să efectueze operații de întreținere ca înlocuire de siguranțe, schimbare de plăci cu circuit imprimat, strîngeri de șuruburi, alte înlocuiri de piese.

Capitolul 3

PROIECTUL EXPERIMENTAL TIMISOAREAN DE ROBOT INTELI- GENT PENTRU CERCETAREA AVANSATA - P.E.T.R.I.C.A.

3.1. Concepția generală a proiectului

În concordanță cu titlul capitolului și paragrafului se va explica cadrul general care a stat la baza lucrării de față. S-au avut în vedere mai multe considerente între care: 1) desfășurarea cercetării sub forma unui proiect amplu, care după modelul altor proiecte de același gen prezentate în capitolul 2, să se desfășoare treptat cu acumulări succesive pe parcursul câtorva ani (1979-1985); 2) de la bun început s-a avut în intenție imprimarea unui puternic caracter experimental, considerându-se că efectuarea de experimente în robotică este de primă importanță, ceea ce presupune neapărat existența unui robot în laborator; 3) termenul de cercetare avansată a fost introdus din dorința de a se sublinia faptul că pe lângă studiul unor principii de bază ale roboticii s-au avut permanent în vedere cercetări fundamentale cu luarea în considerare a aplicațiilor industriale pe termen scurt sau mediu.

În fine, a mai rămas de lămurit spre ce categorie de sisteme robotice-considerate inteligente - se va îndrepta lucrarea de față: 4) aceste sisteme robotice vor trebuie să fie capabile de percepția și recunoașterea mediului înconjurător, având posibilitatea să reacționeze la schimbări sau perturbații în acest mediu și prin folosirea anumitor cunoștințe despre siluete și forme să fie capabile de îndeplinirea unor acțiuni îndreptate spre un anumit scop, acțiuni de genul mișcărilor și manipulărilor antropomorfe.

Se pot aduce unele completări după cum urmează: a) robotul acționează într-o lume reală fapt care trebuie accentuat; b) cuvântul antropomorfic nu are o pondere fundamentală, obiectivul sistemului sau consecințele acțiunilor sale pot avea un caracter umanoid dar este nerelevant să se considere dacă modul în care acestea sînt efectuate este antropomorfic sau nu; c) roboților inteligenți li se pot atribui sarcini diferite, ei înșiși

iau decizii și comunică cu operatorul prin limbaj (mai mult sau mai puțin apropiat de cel natural).

Printre cele patru funcțiuni principale necesare unui robot inteligent găsim: a) percepția, b) luarea deciziilor și planificarea folosind cunoștințele despre lume, c) comunicația, d) capacitatea de a învăța din experiența anterioară. Se subliniază că primelor două li se va acorda atenție necesară în cuprinsul lucrării de față.

După cum s-a arătat în capitolul 2, roboții compleți (cu existență fizică), capabili și de locomoție și de manipulare sînt neașteptat de puțini. Un alt aspect important este acela că problemele de navigație automată au fost studiate separat de cele de manipulare, că există insuficiente cunoștințe teoretice și experimentale la joncțiunea celor două direcții de bază ale roboticii actuale, de aceea se va acorda și acestui aspect atenția necesară în lucrarea de față.

3.2. Robotul mobil cu manipulator-P.E.T.h.I.C.A.

Pornind de la ideea că o cercetare solidă în robotică nu se poate întreprinde fără să se dispună de roboți în laborator concepuți anume pentru cercetarea fundamentală s-a proiectat și construit robotul mobil cu manipulator P.E.T.h.I.C.A. (Foto 3.1) spre a servi ca principal suport experimental în cadrul proiectului cu același nume.

Robotul a fost proiectat în vederea abordării unei game largi de cercetări în robotica avînd următoarele trăsături principale:

- 1) fără specific (nespecializat pentru o sarcină sau un anumit mediu de lucru)
- 2) mobil (cu autonomie limitată ombilical)
- 3) înzestrat cu manipulator
- 4) avînd un microcalculator imbarcat
- 5) echipat cu senzor complex de vedere

Accentul după cum se vede a fost pus pe capacitatea de manipulare în legătură cu cea de navigație și cu precădere pe cea de creștere a capacității de adaptare bazată pe senzor vizual. În cuprinsul capitolului de față vor fi prezentate toate

caracteristicile conceptuale și tehnico-constructive ale robotului referitoare la resursele robotice ale acestuia adică

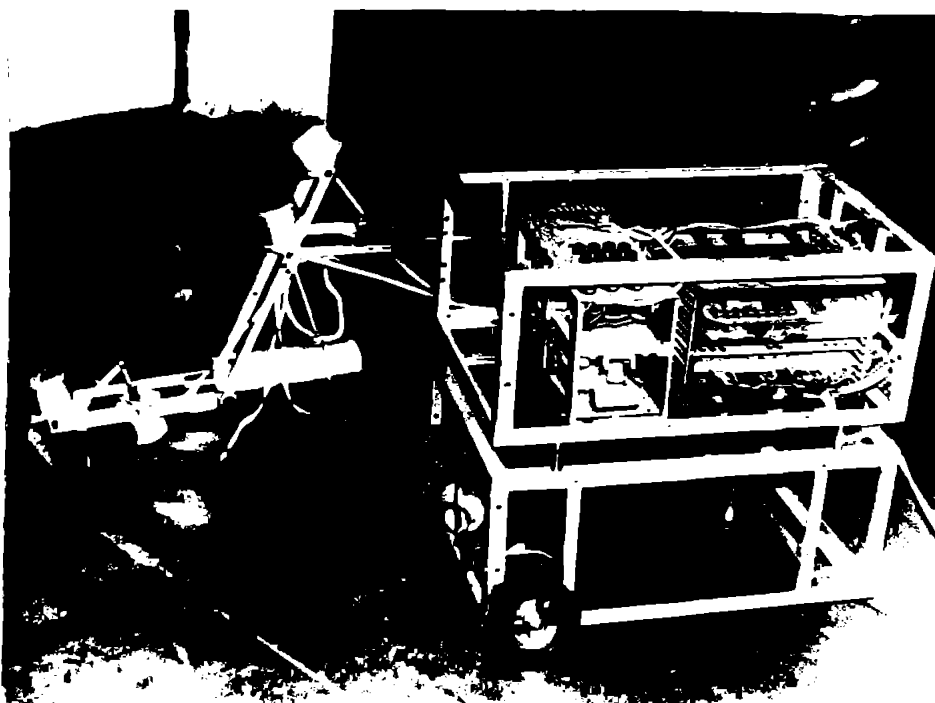


Foto 3.1.

subsistemul mecanic executor de locomotie și manipulare, acționarea cu motoare electrice pas cu pas, apăsura de conducere cu resursele și echipament și programant de calcul.

Arhitectura robotului (fig.3.1) reține concepția modulară foarte eficientă utilizată și de alți constructori de roboți mobili. Aceasta înseamnă

că au fost prevăzute asemănător cu Shakey, Jason și Milare, trei nivele structurale distincte. Pe primul, nivelul de bază, se află dispuse componentele mecanice de locomotie și sursele de alimentare autonomă. Pe nivelul intermediar (torul) se află amplasat echipamentul de calcul și comandă îmbucător și sursa de alimentare pentru evoluția semiautomă când există cordon ombilical.

Pe ambele părți laterale ale torului există posibilități de amplasare a două brațe robotice având o dispunere unireală față de tors, comparativ cu Jason și Mars-over care au o dispunere frontală. Robotul a fost înzestrat doar cu un singur braț robotic.

Pe nivelul superior s-a văzut o platformă pentru echipament senzorial de tipul cameră de televiziune, eventual de telemetrie cu laser și detectori cu ultrasunete.

Autonomia de evoluție a robotului este limitată de un cordon ombilical în lungime de 3 m care la dimensiunile de gabarit

ce le are și asigură suficient spațiu de manevră. Cordonul om-

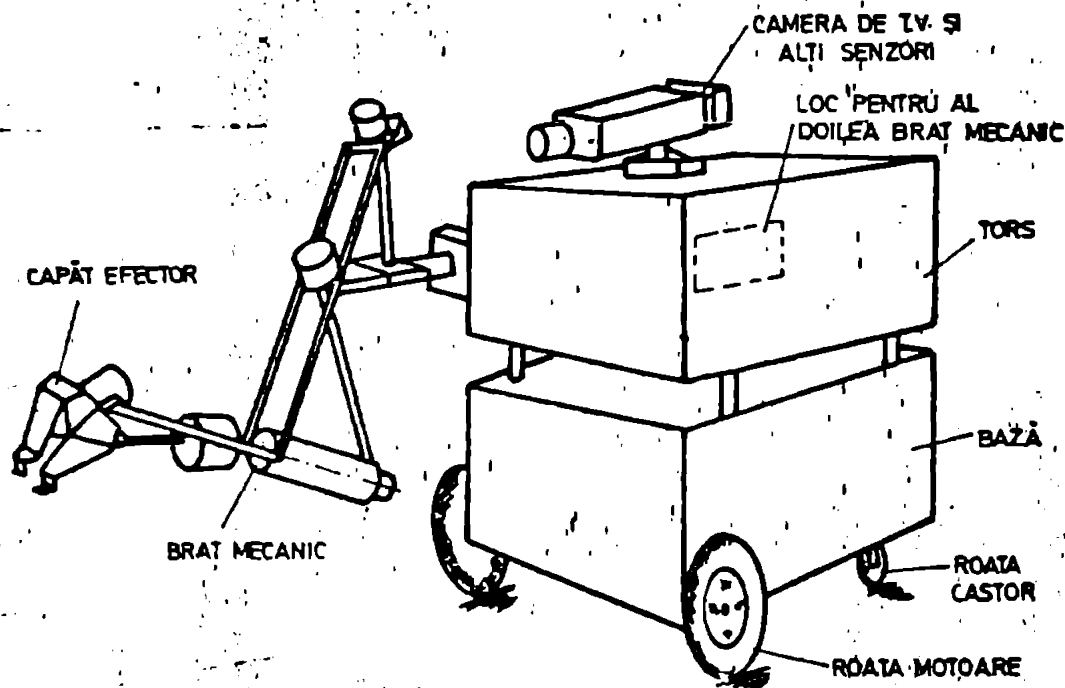


Fig.3.1.

bilical este fixat la unul din capete prin cuple la robot iar la celălalt printr-o cutie fixă la sol având dispuse cuple electrice destinate conectării robotului cu alte aparate și

echipamente.

3.2.1. Subsistemul mecanic efector

3.2.1.1. Manipulatorul

Cea mai mare parte a roboților industriali de manipulare actualmente în funcțiune în aplicații de transfer de materiale, sudură cu arc sau în puncte și asamblare pot fi împărțiți după geometrie și caracteristici de mișcare în patru categorii :

- coordonate carteziene
- coordonate cilindrice
- coordonate sferice sau polare
- coordonate sferic-articulate.

Ținând cont de faptul că cei mai avansați roboți de asamblare au manipuloare sferic-articulate s-a ales tipul de manipulator bazat exclusiv pe cuple cinematice de rotație. (Foto 3.2)

Alegerea schemei cinematice și de acționare a manipulatorului robotului P.S.T.F.I.C.A. a fost influențată de câteva considerații printre care :

- 1) realizarea unui braț robotic la un preț scăzut folosind componentele și materialele existente.

2) elaborarea unui suport de testare și evaluare real pentru algoritmi și programul de calculator destinat comenzii.

3) producerea unui manipulator de dimensiuni suficiente pentru a furniza o paletă de probleme de comandă similare cu cele întâlnite

pentru manipulatorile de sarcini ușoare din industrie.

4) posibilitatea de a efectua manipulări luând în considerare și gradele de mobilitate oferite de locomotie.

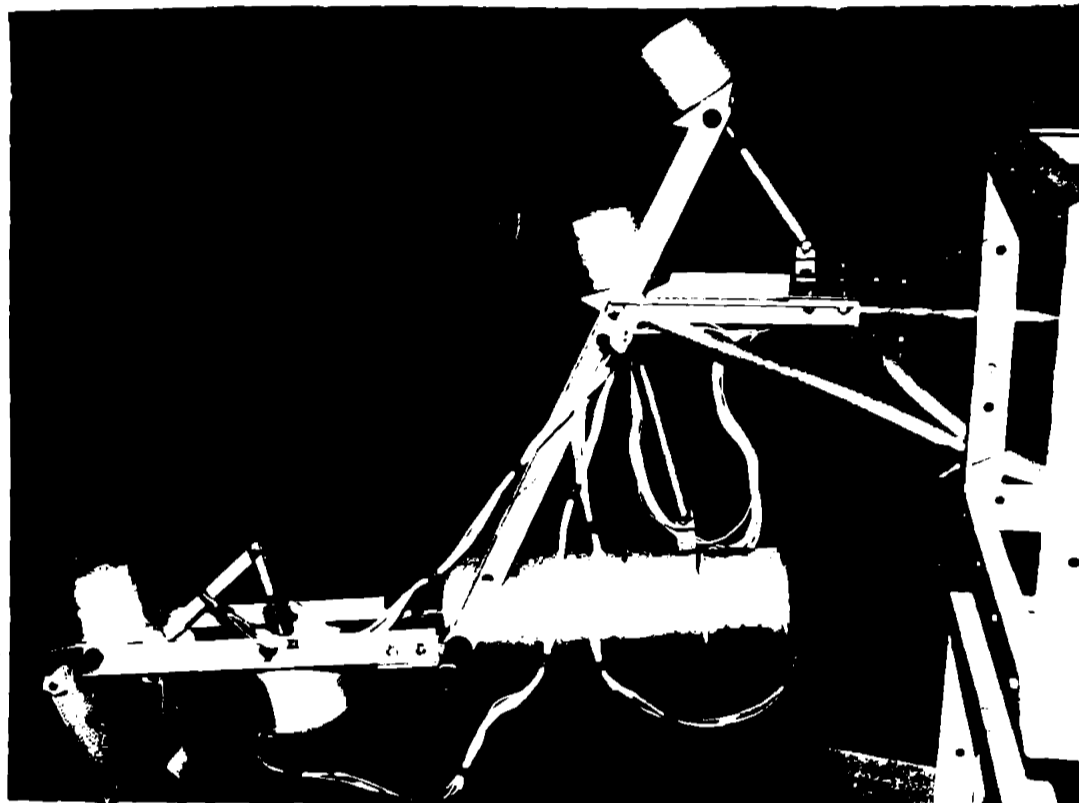


Foto 3.2.

Lună cum se vede din fotografie una din caracteristicile centrale constructive este folosirea mecanismului cu gurub conducător și piuliță pentru trei din cele cinci grade de libertate ale manipulatorului, cât și pentru dispozitivul de prehensiune. Celelalte două grade de libertate au fost rezolvate prin mecanisme de tip reductor cu roți dințate.

Manipulatorul lui P.a.T.I.I.C.A. poate fi considerat ca având dimensiuni antropomorfe (Fig.3.2). Această rămâne valabilă în privința tuturor dimensiunilor. Lungimea legăturii umăr-cot este de 248 mm, a legăturii cot-mână este de 260 mm și mână-degete de 20 mm. Faza maximă de lucru este de 300 mm iar mîna se poate deschide cu cele două degete la 85 mm. Suruburile conductoare au câteva avantaje asupra altor soluții. Ele pot fi proiectate să dezvolte forțe de acționare ridicate, să asigure autoblocarea în poziții de repaus și să producă mișcări line și precise. Unele șuruburi conductoare sînt comandate altele tensionate. De pildă șurubul conducător aparținînd umărului este

în permanență tensionat la fel cel de la încheietura minii. Surubul de cot este comprimat. Atît antebrațul, brațul și prehensiunea au fost alungite pentru a se realiza o echilibrare a greutăților acestor legături în jurul axelor de rotație. Suruburile realizează un moment de rotire bazat pe acțiunea unei forțe aplicate elementului de manipulator care se rotește. La unghiuri mici raza la care se aplică forța este aproape constantă dar odată cu creșterea domeniului de variație începe să scadă rapid. De aceea unghiurile de rotație nu sînt mai mari de 45° .

Principalul tip de material folosit în construcția manipulatorului au fost cornier de aluminiu eloxat de 1,6 mm grosime și 13 mm lățime. Cîteva piese s-au realizat din tablă de aluminiu de 1,5 mm grosime. Intregul manipulator a fost asamblat prin strîngeri cu șuruburi și piulițe.

Lagărele, atît cele pentru realizarea mobilității articulațiilor cit și cele pentru aplicarea forțelor de acționare sînt realizate din teflon grafitat.

Rezultatul acestei construcții simple este că se poate obține un grad de precizie considerabil cu un minim de dificultăți și operații de prelucrare de precizie.

Cursurile de imprecizie ale unui manipulator robotic sînt trei în principal: jocul în lagăre, jocul între flancurile filetului șurubului și piulița aflată în cilindrii de cursă, jocul reductorului cu roți dințate.

Ou toată echilibrarea, chiar dacă nu există o sarcină utilă greutatea elementelor însăși produce o încărcare. Astfel nu apare nici un efect tranzitoriu la încărcarea cu o sarcină utilă care ar deplasa sus-jos sau înainte-înapoi jocul din lagăre. După cum s-a amintit mai înainte anumite șuruburi rămîn

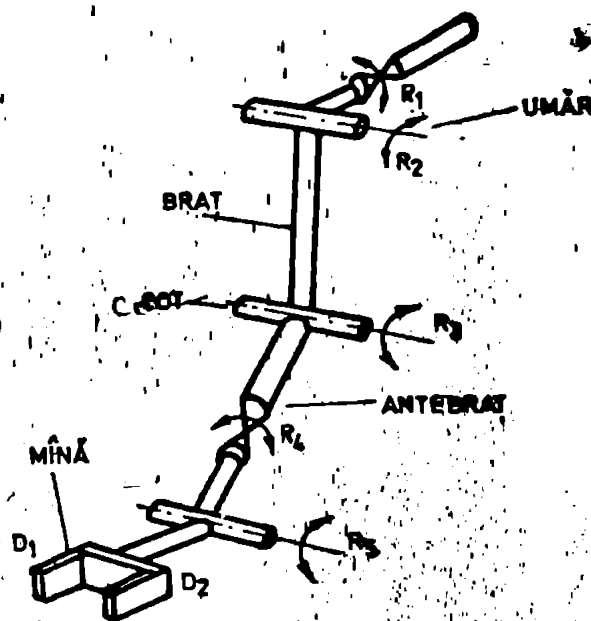


Fig.3.2

măre sau comprimate sau tensionate lucrând tot timpul pe aceeași suprafață a filetului indiferent că se ridică sau se coboară o sarcină utilă.

În tabelul 3.1 sînt date principalele caracteristici tehnice și constructive ale manipulatorului.

Tab.3.1.

CONFIGURAȚIE	→ cinci grade de libertate	
	• Pivotare umerală	- 180° (R ₁)
	• rotație umerală	- 45° (R ₂)
	• rotație de cot	- 45° (R ₃)
	• Suspiniție	- 180° (R ₄)
	• Balansare	- 35° (R ₅)
ACTIONARE	- motoare electrice pas cu pas tip ID-08 cu 15° unghiul de pas și 4 faze, 12 V, 1,2 A	
	- șuruburi conducătoare pentru R ₂ , R ₃ și R ₅	
	- reductor cu roți dințate pentru R ₄	
	- liber sub influența gravitațională R ₁	
PERFORMANȚE	- viteze de rotații medii	- rezoluție
	• Rotație umerală	-1 rad/16,8s 0,027°/pas
	• Rotație de cot	-1 rad/24 s 0,016°/pas
	• Suspiniție	-1 rad/1s 0,1°/pas
	• Balansare	-1 rad/10,8s 0,04°/pas
SARCINA MAXIMA	- 1,5 kg	GRUPATEA BRATULUI - 12,4 kg
LAZA MAXIMA	- 300 mm	

3.2.1.2. Căruciorul

Partea cu funcție de vehicul a robotului PSTRICA se înfățișează ca un cărucior pe 3 roți (inițial se fost 4). Roțile din față sînt antrenate independent de două motoare pas cu pas prin intermediul unui reductor cu roți dințate avînd raportul de transmisie 100. Prin combinații între sensurile celor două motoare de antrenare sau prin acționarea unei roți și menținerea celeilalte frinate se obțin opt mișcări diferite (tab.3.2)

Tab.3.2

MISCAREA	MOTOR 7	MOTOR 8	SENSURI
ROȚILE STINGA (7)	DA	NU	-
ROȚILE DREAPTA (8)	NU	DA	-
ROȚILE PIVOTA (9)	DA	DA	DIFERITE
DEPLASAREA LINIARA (A)	DA	DA	ACELAȘI

Astfel prin acționarea simultană în același sens a sube-
lor motoare se realizează mișcarea de DEPLASAREA LINIARA, înain-
te sau înapoi indicată cu codul hexazecimal (A) în fig.3.3. În
timpul acestei deplasări căruciorul exe-
cută o mișcare de translație pe o tra-
ectorie liniară. Mișcarea cu codul (9)
din figură numită ROȚILE PIVOTA, se de-
sebește de cea dinainte prin sensul dife-
rit în care sînt acționate roțile. Aceas-
tă mișcare face ca robotul să se rotească
în jurul unei axe virtuale care trece
prin mijlocul distanței între roțile din
față, punct notat cu R. În fine prin ac-
ționarea separată a uneia din roți și men-
ținerea celeilalte în stare blocată se exe-
cută mișcările notate cu (7) și (8) denumi-
te ROȚILE STINGA și ROȚILE DREAPTA.

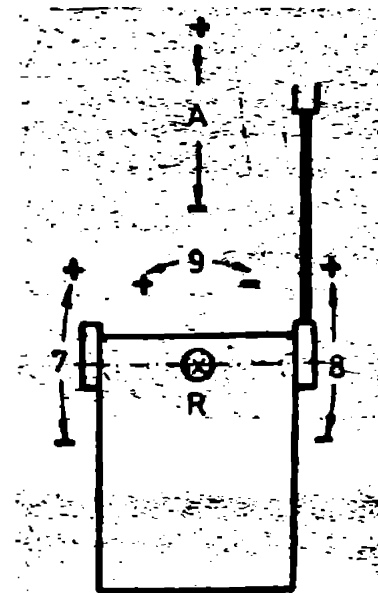


Fig.3.3.

Pentru apropiere treptată a căru-
ciorului de un obiect sînt avantajoase
ROȚILE STINGA și DREAPTA. Mișcarea ROȚILE PIVOTA este deose-
bit de utilă pentru deplasarea rapidă a obiectelor.

3.2.2. Microcalculatorul de comandă imbarcat

Robotica are nevoie de aplicații ale tehnicii de calcul
avînd un înalt grad de complexitate. Multă vreme performanțele
roboților au fost limitate de viteza și capacitatea de memora-
re a unei singure unități de calcul fie aceasta minicalculator
sau sistem mare de calcul, fiind în plus și foarte costisitoare.

Cel dintîi avantaj pentru robotică oferit de microproce-
soare este că, odată cu reducerea volumului util ocupat și pre-
țul scăzut, a devenit economică utilizarea unui număr mare de

calculatoare în cadrul unui singur robot. Un număr de microprocesoare mici dar puternice pot fi folosite pentru a crea dispozitive senzoriale și de comandă inteligente alcătuind împreună subsisteme cu o coordonare unică la nivel de sistem /Do 77/.

Pentru realizarea mișcărilor de manipulare și locomoție robotul PETRICA dispune de un microcalculator de comandă imbarcat la bord. El a fost conceput ca microcalculator de dezvoltare și este pe larg prezentat în /MS 81/. Configurația sistemului de comandă cu microcalculator este redată în fig.3.4.

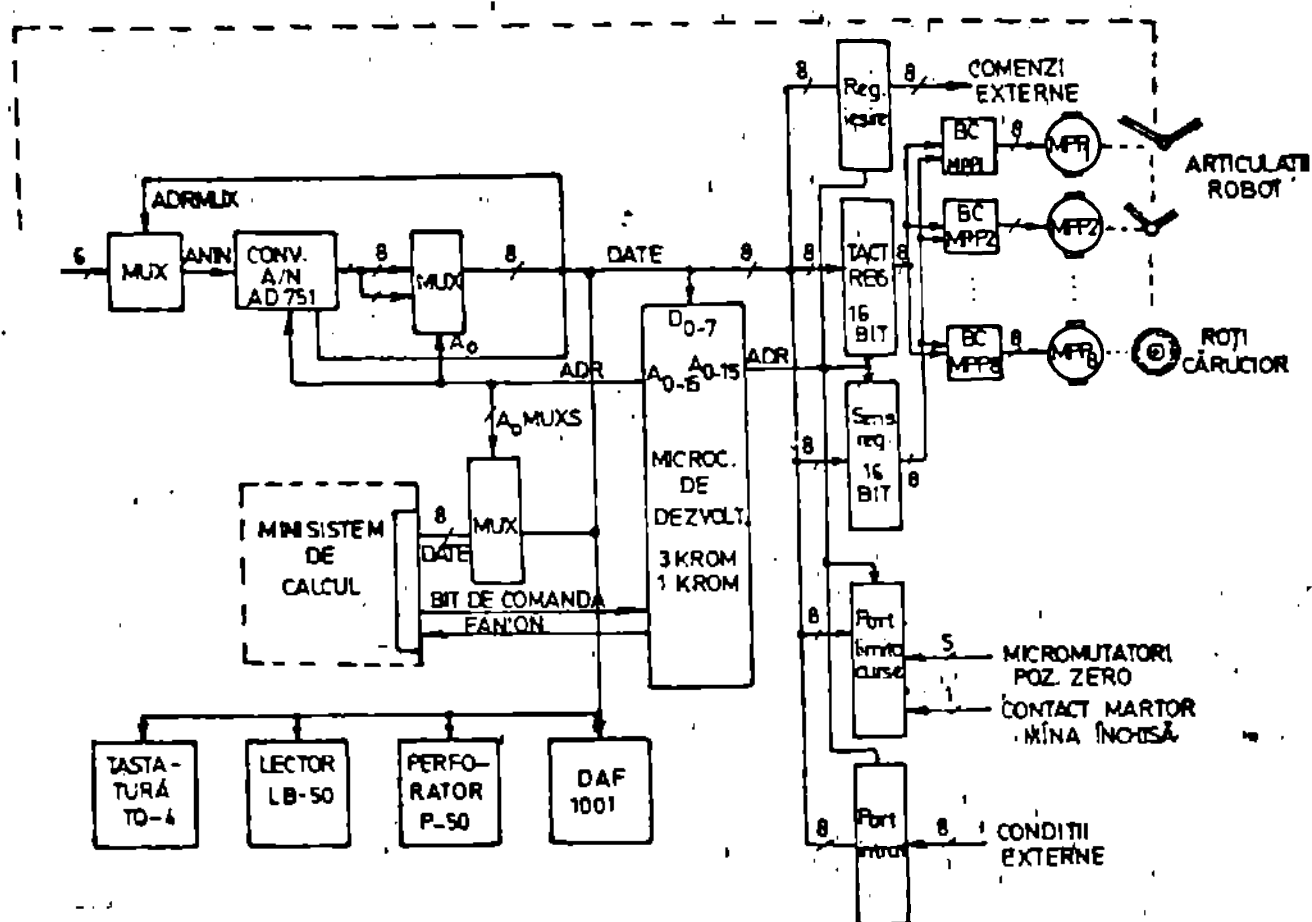


Fig.3.4.

3.2.2.1. Structura microcalculatorului

Circuitul integrat pe scară largă microprocesor poate fi considerat o componentă electronică având un repertoriu de funcțiuni logice care se realizează pe baza proprietăților combinatorii și secvențiale ale acestuia la care se adaugă caracterul programabil.

Aplicat la structura microcalculatorului robotului PETRICA, conceptul proiectării logice programate se traduce prin existența unor memorii fixe de programe care permit pe de o par-

te implementarea funcțiilor de bază ale microcalculatorului iar pe de altă parte funcțiunile de expert de mișcare a manipulatorului respectiv a căruciorului.

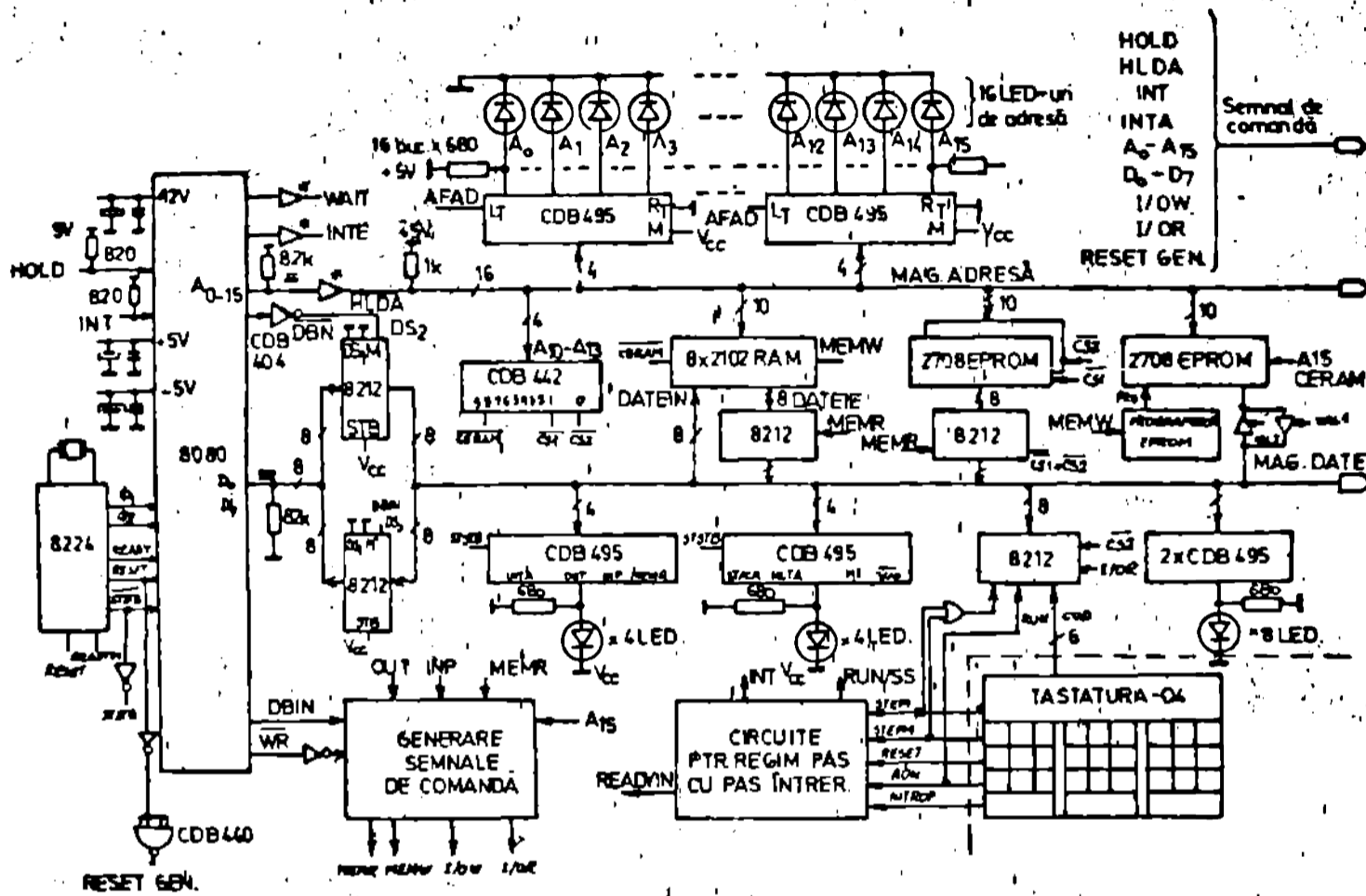


Fig.3.5.

Microcalculatorul imbarcat este realizat pe patru plăci de circuit imprimat dublu strat în tehnologia cea mai modernă pentru tehnica de calcul existentă în țară. Prima placă conține microprocesorul, memoriile RAM, EPROM și logica aferentă. A doua placă cuprinde interfețele cu echipamentele periferice colector, perforator, DAF-1001 și convertor A/N pentru 16 căi. În fine ultimele două plăci conțin electronica de acționare a motoarelor pas cu pas.

În schema din fig.3.5 se remarcă în primul rând grupul circuitelor unitate centrală cu microprocesor 8080 și generatorul de tact B224, caracteristic calculatoarelor bazate pe acest microprocesor. Capacitatea memoriei este de 1 koctet RAM și maximum 3 kocteți memorie EPROM. Generarea semnalelor de comandă ale microcalculatorului se realizează în blocul cu același nume, implementat cu circuite integrate pe scară mică TTL. Pentru ur-

mărirea activității microcalculatorului în regim de rulare pas cu pas, octetul de stare, datele și adresa sînt afișate cu diode luminescente. Comenzile operator se dau de la o tastatură tip TO-4 avînd afectate 12 taste pentru comenzi din cele 28, restul reprezentînd cele 16 cifre hexazecimale. Intrările și ieșirile sînt obținute prin combinarea celor două metode a I/E izolate și I/E în spațiul memoriei.

Microcalculatorul dispune de un program MONITOR care ocupă 1 koctet de memorie EPROM și care implementează funcțiunile de panou obișnuite ale oricărui calculator. Pentru realizarea funcțiunilor de expert de mișcare există un program de comandă numit EX.MIS care ocupă și el un alt koctet de memorie EPROM.

3.2.2.2. Modulul de achiziții de date

Schema convertorului A/N și multiplexorului analogic este arătată în fig.3.6. Se folosește un circuit integrat AD751 care conține în aceeași capsulă circuitul pentru tensiunea de referință, generatorul de tact, comparatorul, registrul de aproximare succesivă (SAR) și registrul tampon. Performanțele de bază sînt 10 bit și 25 de microsecunde timpul pentru o conversie. Circuitul AD571 întrebuintează cea mai avansată tehnologie de fabricație a circuitelor I²L. Compensarea cu temperatura se face cu o diodă Zener integrată.

Funcționarea lui AD571 în regim de conversie comandată prin impuls se petrece astfel : cînd $\overline{B \overline{C}}$ devine „0” logic, ieșirile trei stări de date sînt flotante și începe conversia. Terminarea ei este semnalată de \overline{DR} care ajunge la „0” și tot acum sînt validate ieșirile de date.

Semnalul pentru intrarea $\overline{B \overline{C}}$ este format de bistabilul RS implementat cu două porți CDB400. În urma executării unei instrucții de ieșire se generează semnalele I/OW și ADRCAN semnalul de selecție al convertorului. Aceste două semnale provoacă aducerea la intrarea $\overline{B \overline{C}}$ a lui „1” logic. După 1,5 μ s semnalul \overline{DR} devine „1” logic aduce bistabilul pe zero. Acum abia începe conversia care durează aproximativ 25 μ s. Evenirea la „0” a lui \overline{DR} marchează sfîrșitul conversiei și poate fi sesizată prin program. Rezultatul conversiei apare pe 10 bit care pot fi citiți printr-o singură instrucție INLD. Amplificatorul operațional μ A715 funcționează ca repetor și realizează adaptarea de impedanță între intrarea convertorului ANLN și ieșirea multiplexorului analogic.

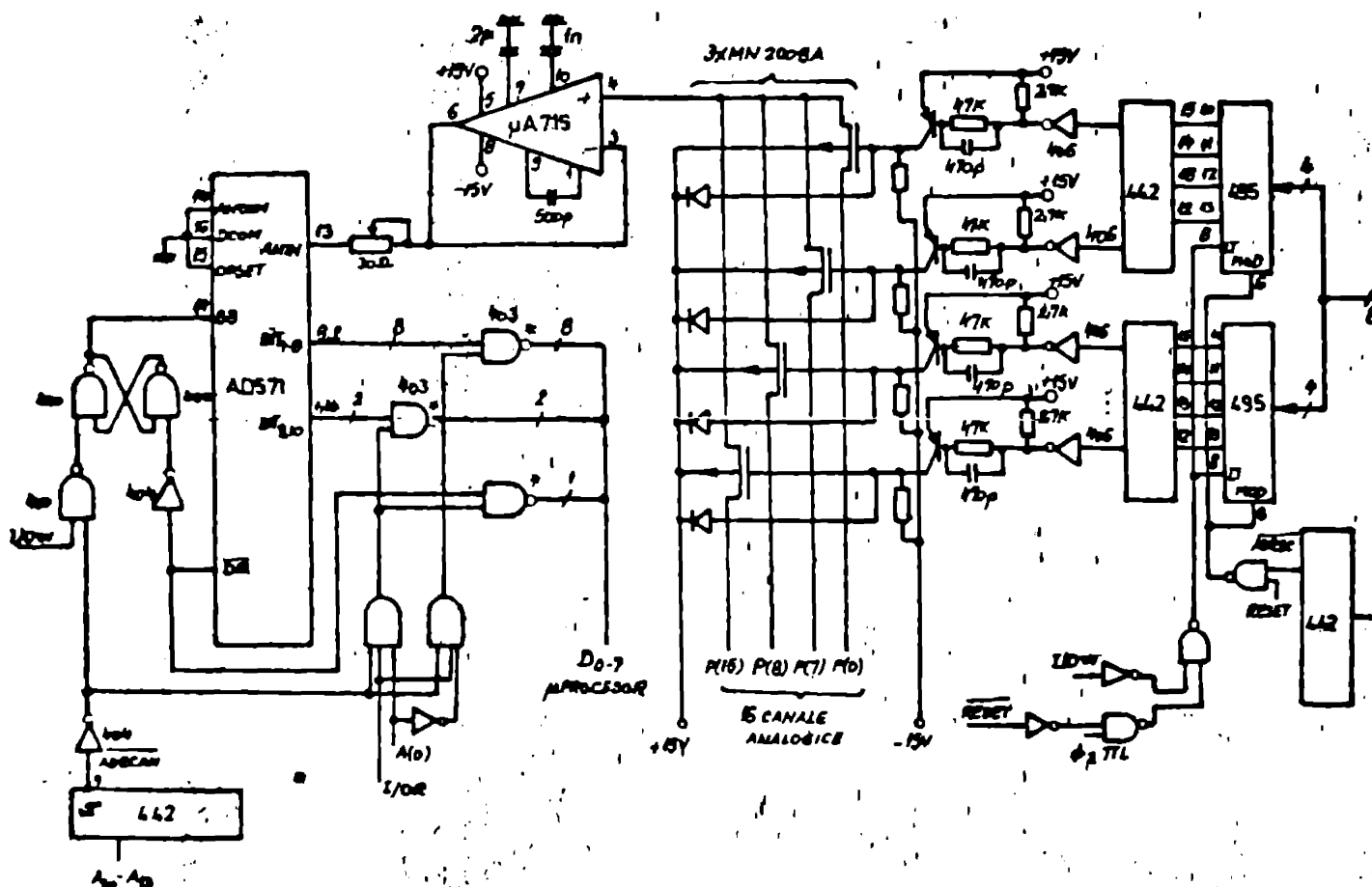


Fig.3.6.

3.2.2.3. Acționarea cu motoare electrice pas cu pas

Motorul electric pas cu pas este printre cel mai larg folosite dispozitive pentru conversia informațiilor numerice în lucru mecanic util. Comandat direct cu trenuri de impulsuri ele constituie un mod economic și simplu de poziționare numerică precisă. Prin înlocuirea servomotoarelor de c.c. funcționând în buclă închisă cu motoare pas cu pas (MPP) în buclă deschisă se elimină necesitatea unor convertoare N/A, amplificatoare de putere cu coeficient ridicat de linieritate, traductoare de poziție și viteză, rețele de compensare a buclei, etc.

Pentru mișcările robotului PETRICA se folosesc în total șapte MPP repartizate astfel: două pentru cărucior, patru pentru brațul mecanic și unul pentru mină. În caracteristicile MPP (fig. 3.6) se remarcă existența curbei START-STOP care indică pentru un anumit cuplu motor frecvența de comandă la care rotorul rămâne în sincronism la pornirea din repaus sau la oprire. Curba a doua, de FUNCTIONARE, indică frecvențele maxime cu care poate fi comandat motorul fără pierdere de pași. Cele două curbe și zonele delimitate constituie suportul alegerii punctului sau

punctelor succesive de funcționare ale MPP. Pentru obținerea unor mișcări line ale articulațiilor brațului, la pornire și oprire este nevoie de realizarea unei accelerații treptate respectiv decelerări înainte de oprire /Pa 77/. Din punctul de vedere al motorului aceasta înseamnă o lege de variație pentru frecvența de comandă a MPP astfel încât să se permită deplasarea punctului de funcționare din zona

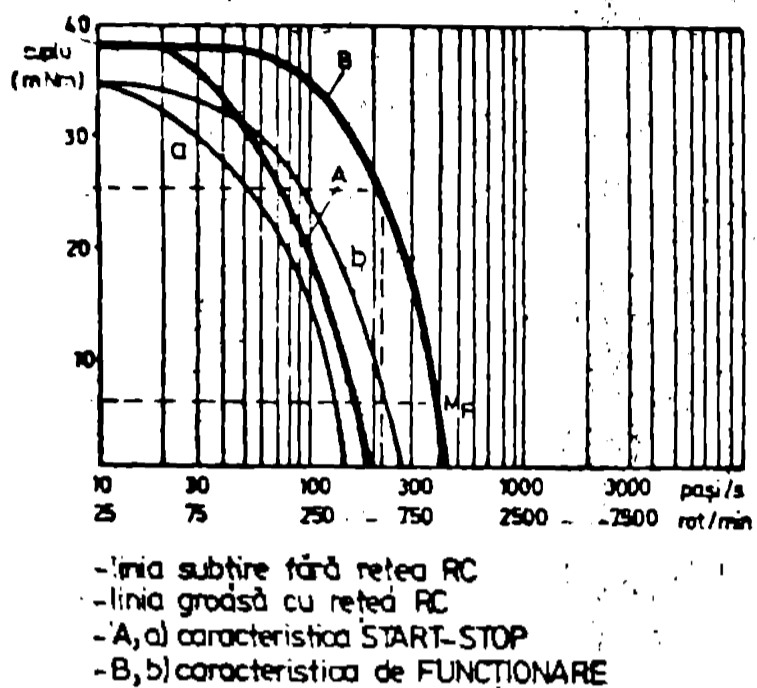


Fig.3.7.

START-STOP în cea de FUNCȚIONARE și înapoi în mod treptat. Forma de variație a frecvenței de comandă în funcție de timp și numărul de pași care să permită accelerarea, funcționarea la viteză constantă și frînare, este arătată în fig.3.8.

În continuare se dă un exemplu de calcul pentru modul în care s-a făcut alegerea MPP pentru acționarea robotului PETRICA. La una din articulațiile robotului (umăr), MPP antrenează un șurub având pasul 1 mm astfel încât pentru 100 de pași piulița se va deplasa cu circa 4,16 mm. Servomecanismul prezintă un moment

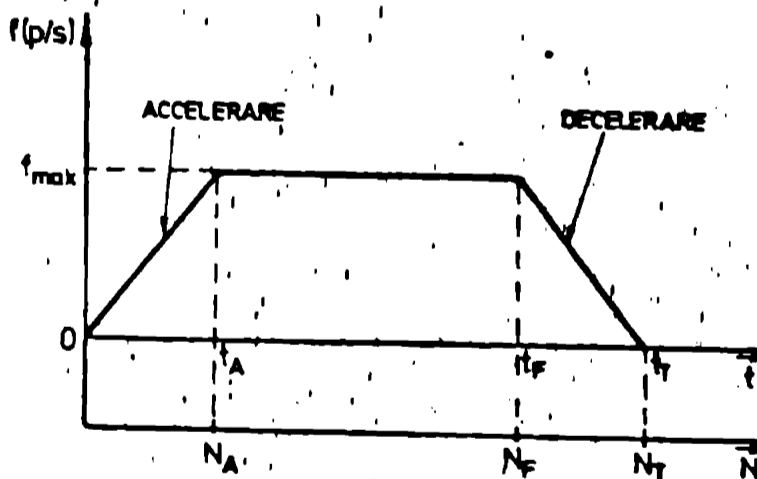


Fig.3.8.

de frecare $M_f = 6 \text{ mNm}$ și un moment inertial total (rotor+sarcină + șurub); $M_{jT} = 125 \times 10^{-4} \text{ gm}^2$. Dacă viteza impusă este de 100 pași în 0,5 secunde rezultă o frecvență de comandă de $v=200 \text{ pași/s}$. Motorul care se alege trebuie să aibă un cuplu mai mare de 6 mNm la o frecvență de pas mai mare decât 200 pași/s.

Pe caracteristica de FUNCTIONARE a MPP s-a determinat o viteză maximă considerind numai cuplul de frecare. O a doua determinare, mai exact estimare a frecvenței de lucru (mai mică decât cea maximă), 230 pași/s conduce la diferența:

$$M_1 - M_f = 28 - 6 = 22 \text{ mNm}$$

Cu un coeficient de siguranță de 50% se obține :

$$22 \text{ mNm} \times 0,5 = 11 \text{ mNm}$$

Folosind relația :

$$M = J_M \times \frac{V}{t} \times K, \text{ unde } K = \frac{2}{nrpași/rev} = \frac{2}{24} = 0,26$$

Se obține valoarea $t = t_A$, adică timpul de accelerare :

$$t_A = \frac{125 \times 10^{-4} \times 230 \times 0,26}{11} = 0,068 \text{ s}$$

Deoarece timpul de accelerare t_A este egal cu cel de decelerare t_D se poate determina numărul de pași pentru accelerare (N_A) și decelerare (N_D).

$$N_A + N_D = \frac{V}{2} \times t \times 2 \quad \text{sau}$$

$$N_A + N_D = V \times t_A = 230 \times 0,068 = 16 \text{ pași}$$

Timpul de mișcare la frecvența maximă este în acest caz:

$$T_{\text{FUNCT}} = \frac{N_T - (N_A + N_D)}{V} = \frac{100 - 16}{230} = \frac{84}{230} = 0,360 \text{ s}$$

în care $N_T = 100$ este numărul total de pași. Verificarea timpului total pentru mișcare se face însumând :

$$T_{\text{FUNCT}} + t_A + t_D \quad \text{sau}$$

$$0,360 + 0,068 + 0,068 = 0,5 \text{ s}$$

Exemplul de calcul de mai înainte este o primă estimare. Se poate face mișcarea mai lentă dacă se dorește o siguranță mai mare, sau mai rapidă - dacă se urmărește optimizarea ei.

Algoritmul 3.1.

Algoritmul pentru comanda motorului pas cu pas pentru cazul discutat, 8 pași pentru accelerare, 8 pentru frinare este următorul :

. dacă $N_T > 16$ se aplică comanda de accelerare (primii

8 pași), apoi viteza maximă și în cele din urmă comanda de frinare (ultimii 8 pași).

. dacă $N_T \leq 16$ se comandă motorul cu o viteză fixă la o frecvență de sincronism mult mai mică decât cea maximă.

Ordinograma subprogramului de comandă care implementează acest algoritm este dată în fig.3.9.

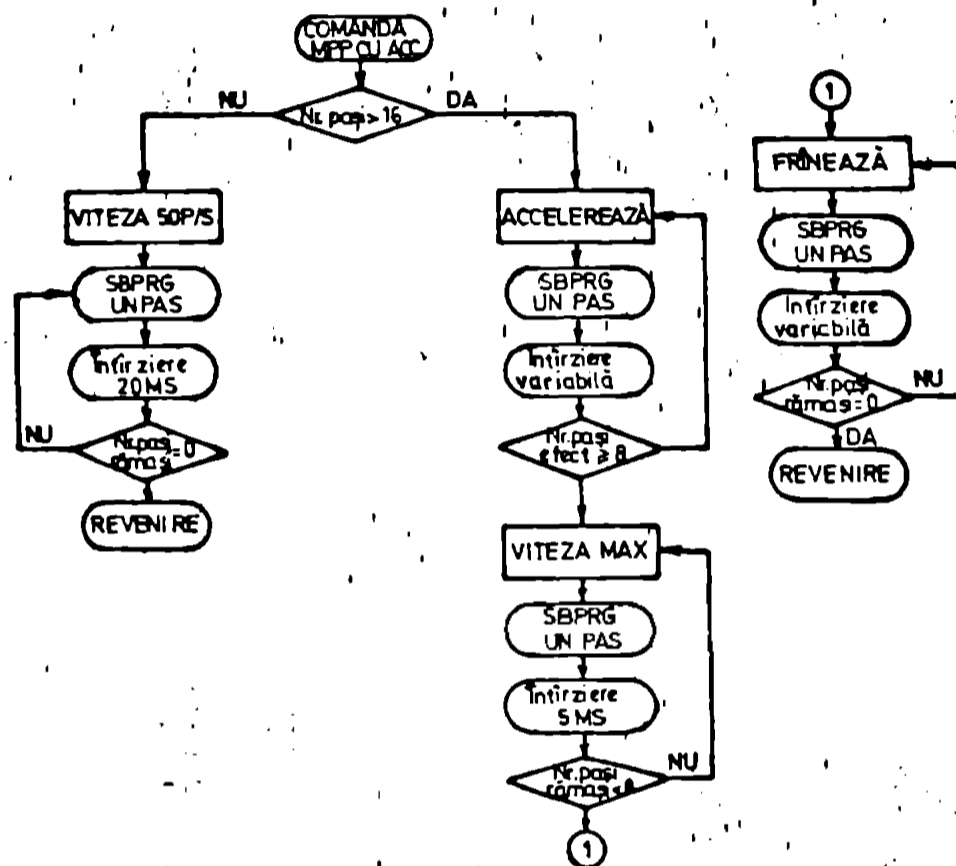


Fig.3.9.

Pe de altă parte se va examina mai îndeaproape problema de terminării valorilor prescrise gradelor de libertate furnizate de sistemul de comandă pentru ca să aibă loc o poziționare anumită a brațului sau căruciorului.

Putem distinge următoarele tipuri de poziționări :

- 1) poziționare simplă printr-un punct în altul fără constrângeri asupra tipului de parcurs de urmărit între punctul inițial al mișcării și cel terminal;
- 2) mișcare dintr-un punct în altul cu constrângerea de a trebui să se treacă prin puncte intermediare specifice.
- 3) mișcare de-a lungul unei traiectorii rectilinii.
- 4) mișcare de-a lungul unei traiectorii înregistrate anterior, adică descriasă de o mulțime de puncte înregistrate cu o frecvență anumită.

Să luăm în discuție poziționarea de primul tip, adică acele poziționări în care trebuie să se realizeze mișcarea dintr-o no-

ziție inițială într-una finală. Traectoria urmărită este indiferentă, dar evoluțiile diverselor GDL trebuie să fie sincronizate, adică pornirea și oprirea să aibă loc simultan pentru toate articulațiile.

In /CG 83/ se prezintă un algoritm pentru realizarea mișcărilor coordonate ale cuplelor minirobotului MINIMOVER /HJ 80/. Algoritmul este următorul.

Algoritm 3.2.

- . se determină numărul maxim de pași MAX, care trebuie executat, dintre cele șapte numere pentru fiecare motor.
- . se inițializează șapte contoare la valoarea MAX/2.
- . se repetă de MAX ori :
 - scade valoarea absolută a numărului de pași ai fiecărui motor din contorul său
 - dacă conținutul vreunui contor este mai mic ca zero se dă comanda de mișcare pentru acel motor în direcția dorită și se adună MAX la contorul său
 - așteaptă un interval de timp cu o durată indicată.

Consecința aplicării acestui algoritm este că motorul cu numărul maxim de pași tinde să domine: aceasta înseamnă că se va mișca cu viteza determinată de timpul de așteptare. Algoritmul este dat în fig.3.10.

Deși acest Algoritm 3.2 rezolvă problema mișcărilor sincrone el nu rezolvă problema mișcării cu accelerare și frinare care aparține de Algoritmul 3.1. Se pune astfel problema realizării unui nou algoritm care să rezolve ambele aspecte adică mișcarea coordonată cu accelerare și decelerare. La elaborarea noului algoritm ne bazăm pe faptul că motorul cu număr maxim de pași are tendința să conducă întreaga distribuție a impulsurilor de comandă pentru toate celelalte MPP. Rezultă de aici că o accelerare sau decelerare a motorului cu număr maxim de pași se va răsfrînge în mod inevitabil asupra distribuției impulsurilor tuturor celorlalte motoare care vor fi și ele accelerate sau decelerate. Astfel se ajunge la Algoritmul 3.3 care este redat în fig.3.11. S-au făcut următoarele notații: PMM -pointer motor care se mișcă, MAX - numărul maxim de pași de efectuat pentru unul din cele 7 motoare, NEFT - cel mai mare număr de pași care se execută (contor), 00_{16} - viteză constantă (50 ms) 01_{16} - accelerare; 02_{16} - viteză constantă (5 ms), 03_{16} - frinare, NN-con-

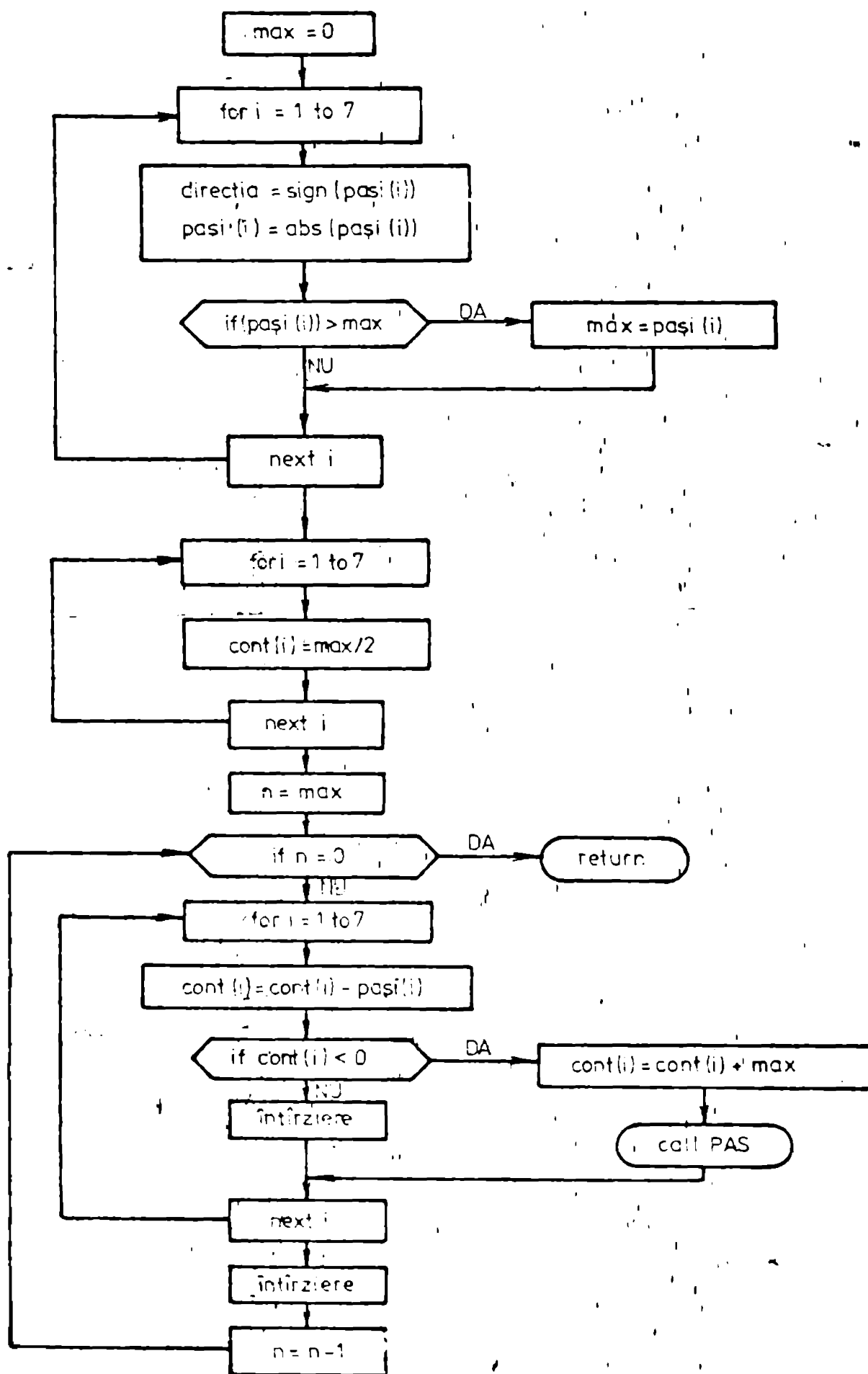


Fig.3.10.

tor cu decrementare pentru primii opt pași.

Algoritmul 3.3 se prezintă astfel:

- . se determină numărul maxim de pași MAX, care trebuie executat de către unul din cele șapte motoare.
- . se inițializează șapte contoare la valoarea MAX/2.
- . se testează dacă NEFT este 16 și se ia decizia dacă mișcarea se face cu accelerare și decelerare sau cu viteză constantă mică (50 pași/s)
- . se repetă de MAX ori :
 - scade valoarea absolută a numărului de pași al fiecărui motor din contorul său
 - dacă conținutul vreunui contor este mai mic ca zero se dă comanda de mișcare pentru acel motor în direcția dorită și se adună MAX/2 la contorul său.
 - se testează conținutul contorului NN pentru primii opt pași și de asemenea conținutul contorului NEFT pentru ultimii opt pași. În funcție de rezultatele testelor se ia decizia pentru mișcare cu viteză constantă (200 pași/s) sau pentru frînare
 - așteaptă un interval de timp cu o durată indicată de fanionul de viteză care poate lua patru valori: 00, 01, 02 și 03.

Ca schemă electronică de acționare a MPP s-a prevăzut pentru fiecare motor un distribuitor de impulsuri pentru cele 4 faze ale unui motor și câte 4 chei de curent. Distribuitorul de impulsuri funcționează corespunzător regimului de lucru al MPP numit de putere întregă. Conform acestui regim se alimentează în permanență câte două înfășurări adiacente adică înfășurarea 1 cu înfășurarea 1 cu înfășurarea 2 - 2 cu 3 - 3 cu 4 - 4 cu 1 ș.a.m.d. Bineînțeles că schimbarea sensului de rotație înseamnă parcurgerea secvenței în ordine inversă adică 1 cu 2 - 4 cu 1 - 3 cu 4 - 2 cu 3 - 1 cu 2 - 4 cu 1 - ... Schema este dată în fig.3.12.

3.3. Arhitectura sistemelor de calcul pentru conducerea inteligentă a robotului PETRICA

3.3.1. Structura distribuită de conducere și luare a deciziilor

Sistemele robotice sofisticate care eu de îndeplinit sar-

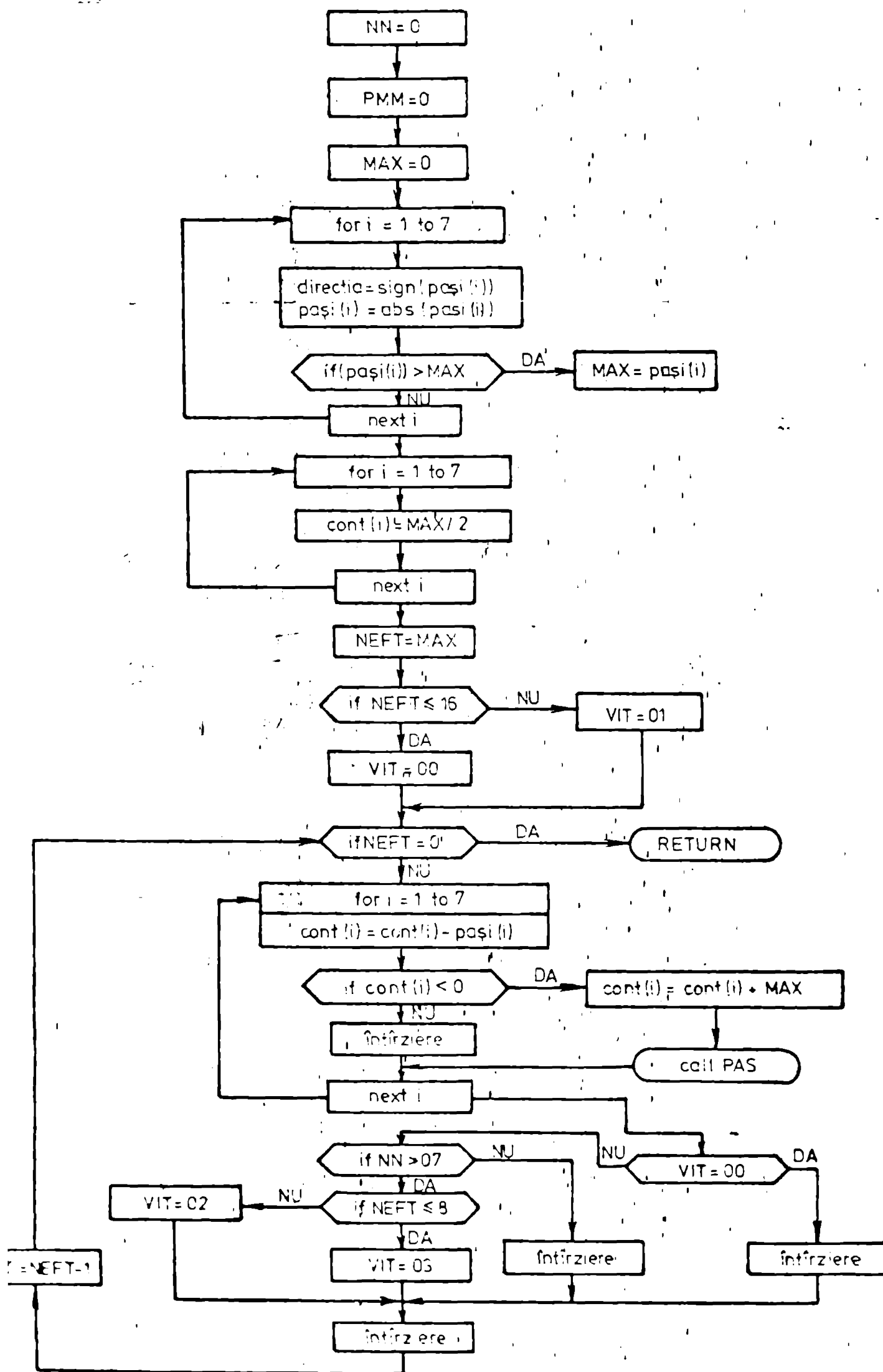


Fig. 3.11

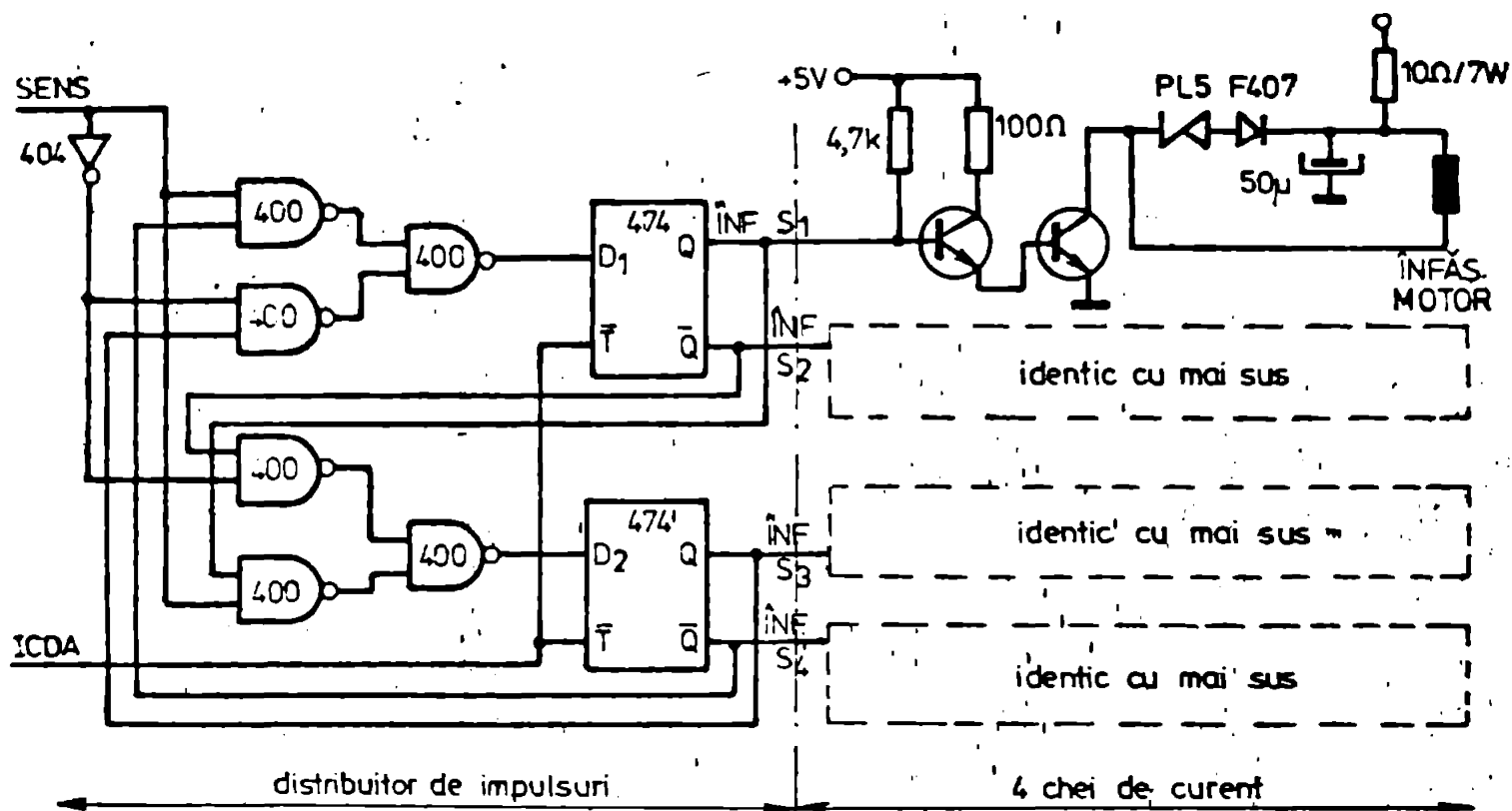


Fig.3.12.

cini complexe trebuie să dispună de resurse puternice și eficiente de prelucrare a datelor. Pentru organizarea acestor resurse pot fi luate în considerare câteva idei distincte:

- 1) Modularitatea este foarte importantă. Ca o consecință senzorii și efectorii trebuie să dispună de propriile lor microprocese de comandă. Astfel toate problemele legate de un senzor sau efector dat sînt rezolvate odată și pentru totdeauna. Combinația între un senzor sau efector și un microprocesor devine o entitate autonomă capabilă de a îndeplini o sarcină. Această sarcină poate fi activată de către alte părți ale sistemului folosind un limbaj de comandă de un nivel semnificativ mai înalt decît o comandă directă a senzorului sau efectorului.
- 2) Pentru comanda sistemului este necesar un minicalculator local pentru a efectua calcule de complexitate medie și pentru a interacționa cu utilizatorul sistemului.
- 3) Anumite sarcini de prelucrare a datelor sînt foarte complexe (de exemplu calcule științifice sau planificare de nivel înalt) și necesită astfel un sistem de calcul din categoria mini sau macrosistemelor de calcul.

S-a definit prin urmare o arhitectură de calculatoare cuprinzînd trei nivele de resurse de prelucrare. Totuși modul în care această arhitectură a fost folosită în experimentele

P.E.T.H.I.C.A. descrise în cap.6. este mai neobișnuită: nivelul care joacă rolul de monitor nu este cel mai înalt pe scara puterii de calcul. Cel mai puternic nivel de calcul este văzut ca o resursă la care accesul se face doar ocazional, în timp ce minisistemul local este în mod constant responsabil de sistemul robotic. În caz de deranjament al minisistemului aflat în vârful ierarhiei de calcul, minicalculatorul local poate prelua sarcina conducerii sistemului într-un mod oarecare degradat oferind astfel o soluție problemei unui robot care se poate degrada într-un chip satisfăcător.

Arhitectura definită mai sus nu provine din oarecare considerații elaborate, Mai degrabă ea trebuie considerată ca o arhitectură validă de facto.

Se ia acum în discuție programamentul de comandă și planificare. S-a ales decizia să se folosească un program de comandă și planificare distribuit. Motivele pentru acest lucru sînt următoarele :

- . O structură de program (software) similară celei hardware furnizează o mai bună eficiență și robustețe. De exemplu, un răspuns mai eficient al sistemului poate fi obținut prin tratarea simultană a diferitelor decizii.
- . Un sistem de luarea deciziilor centralizate, deci general, poate fi foarte inefficient pentru a rezolva anumite probleme ce apar în cursul sarcinii date unui robot. De aceea se susține cu tărie importanța unor sisteme de program și planificare specifice sarcinii (dependența de domeniu). Ceea ce se pierde în generalitate se câștigă în eficiență.
- . Pe lângă eficiență, un sistem de program și planificare specific sarcinii prezintă deasemenea avantajul că mărimea programului este dependentă direct de complexitatea problemei, în timp ce un program de comandă și planificare cu caracter general poate fi greu menținut la dimensiuni mici.

Pe de altă parte, cînd vine în discuție comunicația om-mașină sau nevoia de a schimba ceva în sarcina robotului, sau anumite capacități de învățare, flexibilitatea unui program de comandă și planificare cu caracter general este de dorit.

Structura de comandă definită mai înainte este ierarhică

și modulară. Ea constă din module expert independente și un coordonator de nivel înalt care funcționează într-o structură multi-nivel. Fiecare expert este specializat în domeniul său. Fiecare expert are baza sa de date și modelul propriu al lumii. În fine fiecare expert poate fi folosit ca o primitivă sau un operator de către un expert aflat pe un nivel superior. Structura distribuită de comandă și luare a deciziilor este redată în fig.3.13.

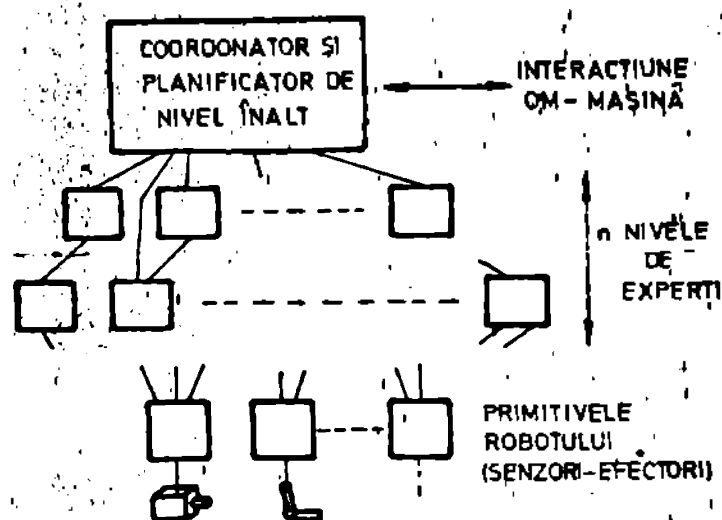


Fig.3.13

Structura de calcul

pentru comanda inteligentă a robotului PETRICA constă din 3 nivele de prelucrare: 1) un minicalculator ROBOTRON 4201 cu memorie de 32 cuvinte și lungimea cuvintului de 16 bit 2) un mini calculator construit prin autodotare /MS 81/ denumit MCFN bazat pe microprocesor INTEL 8080 cu 32 cuvinte memorie operativă asemănător lui MC-18) și 3) un microcalculator INTEL-8080 cu 1 koctet RAM și 3 kocteți EPROM (prezentat în acest capitol la 3.2.2).

Sistemul de calcul MCFN (fig.3.14) este coordonatorul întregului sistem robotic: el asigură funcțiunile unui executiv, comunicația cu utilizatorul și dialogurile cu ROBOTRON 4201 și cu robotul PETRICA precum și achiziția imaginilor. Minicalculatorul ROBOTRON 4201 este folosit ca resursă pentru calcule științifice iar microcalculatorul înmărcat pe robotul PETRICA este expertul de mișcare.

3.3.2. Canalul de comunicație PETRICA - MCFN

Interfața dintre robotul PETRICA și minicalculatorul MCFN este reprezentată în fig.3.15. Ea cuprinde două registre tampon aparținând microcalculatorului care comandă robotul PETRICA, un registru tampon al minisistemului MCFN și fanioane.

a) Comunicația MCFN-PETRICA

În această situație MCFN își activează linia de cerere de dialog care poziționează fanionul din tamponul de recepție aflat în microcalculatorul lui P.E.T.R.I.C.A. Acest fanion este memorat și într-un bistabil în MCFN. În continuare fanionul este citit

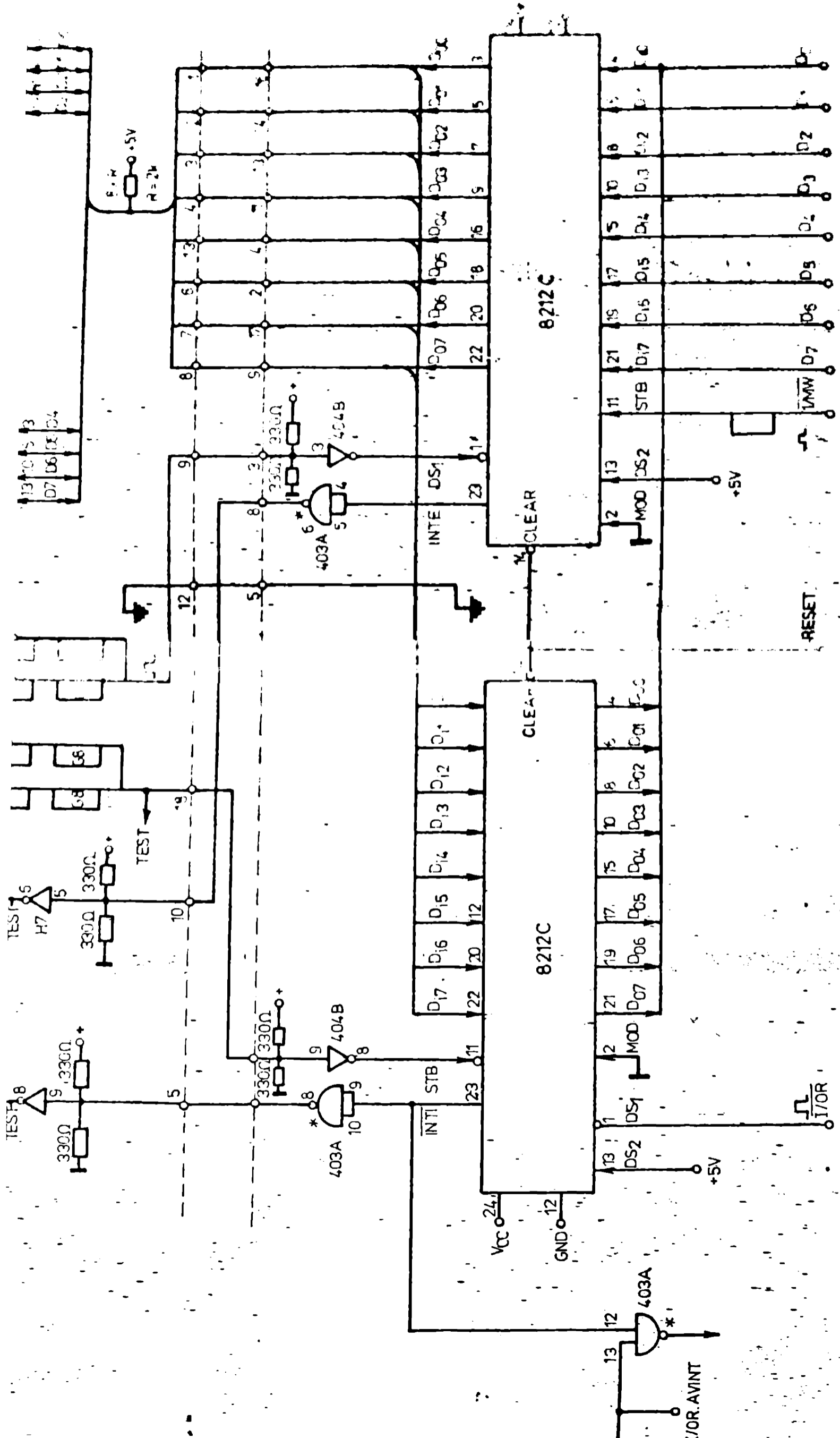


Fig. 3.15

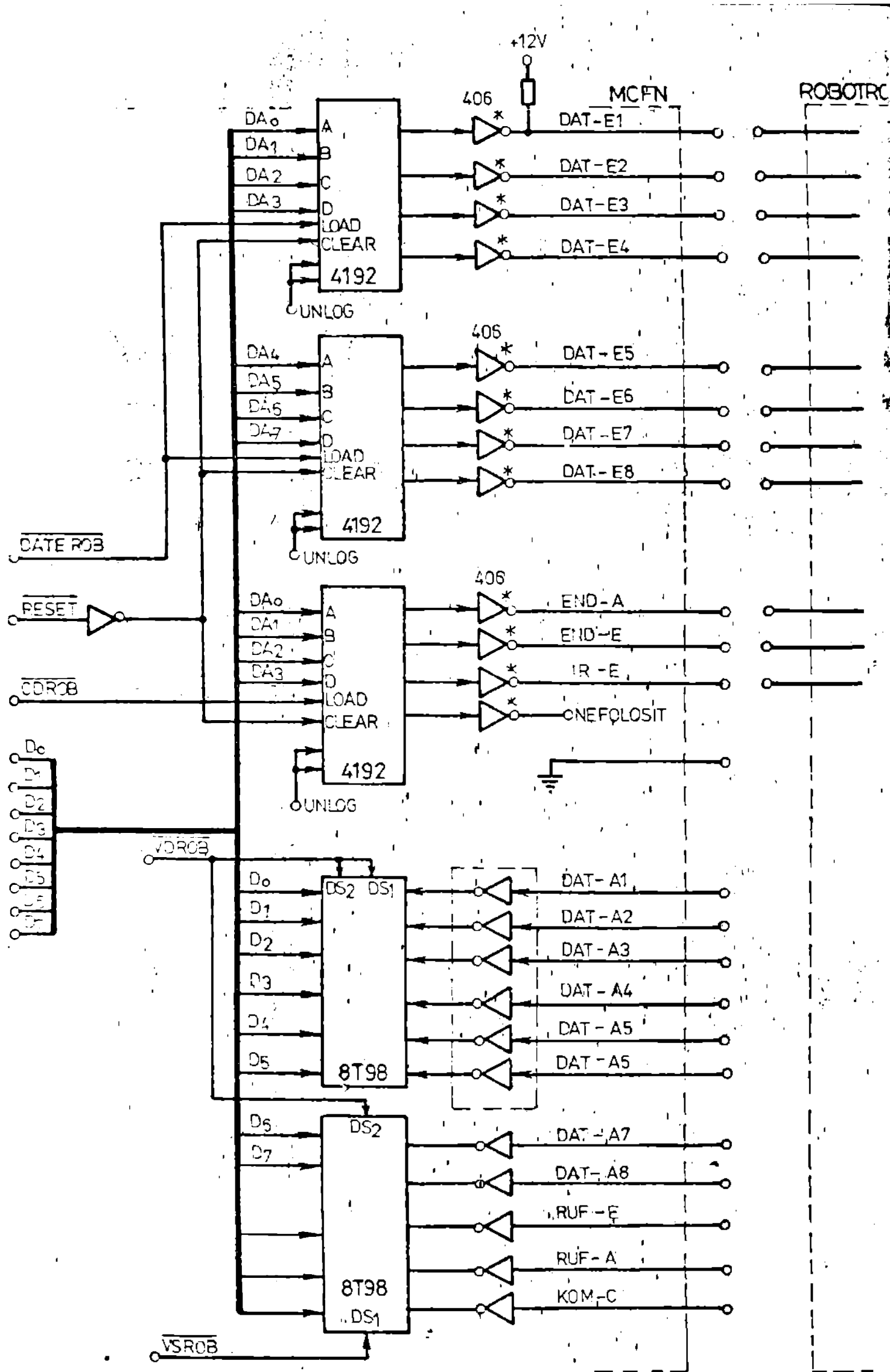


Fig.3.16

de către programul de comunicație care validează transferul de informație din registrul tampon MCFN în registrul tampon de recepție date PETRICA. Totodată se dezactivează fanionul și bistabilul corespunzător din MCFN. Programul de dialog din MCFN testează starea bistabilului aferent fanionului și încarcă tamponul când se sesizează dezactivarea bistabilului.

b) Comunicația PETRICA - MCFN.

În acest caz din partea robotului se lansează o cerere de dialog simultan cu încărcarea datelor în tamponul de emisie. Fanionul activat se memorează într-un bistabil aflat în MCFN. Programul de dialog din MCFN testează bistabilul pe care, dacă îl vede activ, citește date iar în caz contrar intră într-o buclă de așteptare. La fiecare comunicație de date fanionul și bistabilul aferent sunt dezactivați fapt sesizat de către programul de dialog din microcalculatorul lui P.S.T.R.I.C.A., program ce încarcă următorul caracter în tamponul de emisie. În ambele cazuri a și b) transferul are loc în paralel pe 8 bit fără control de paritate.

Programele de dialog cuprind câte două subprograme în microcalculatorului lui P.S.T.R.I.C.A. respectiv alte două de partea minisistemului MCFN și sunt utilizate astfel:

emisie MCFN - recepție PETRICA și
recepție MCFN - emisie PETRICA

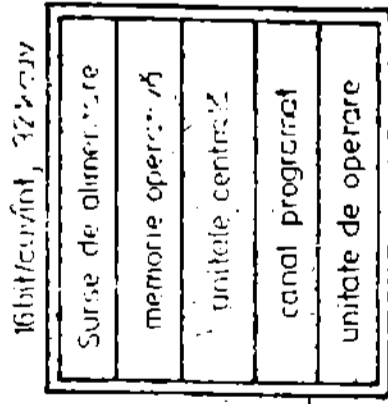
Tipul transferului este cunoscut în urma inspectării fanioanelor corespunzătoare emisie și recepției aflate la următoarele adrese: A062 în MCFN și 9000 în P.S.T.R.I.C.A. Datele se încarcă în porturi la adresele A066 în MCFN și 9400 în PETRICA. Citirea informației se face corespunzător la adresele: A163 în MCFN și 8C00 în P.S.T.R.I.C.A.

3.3.3. Canalul de comunicație ROBOTRON- MCFN

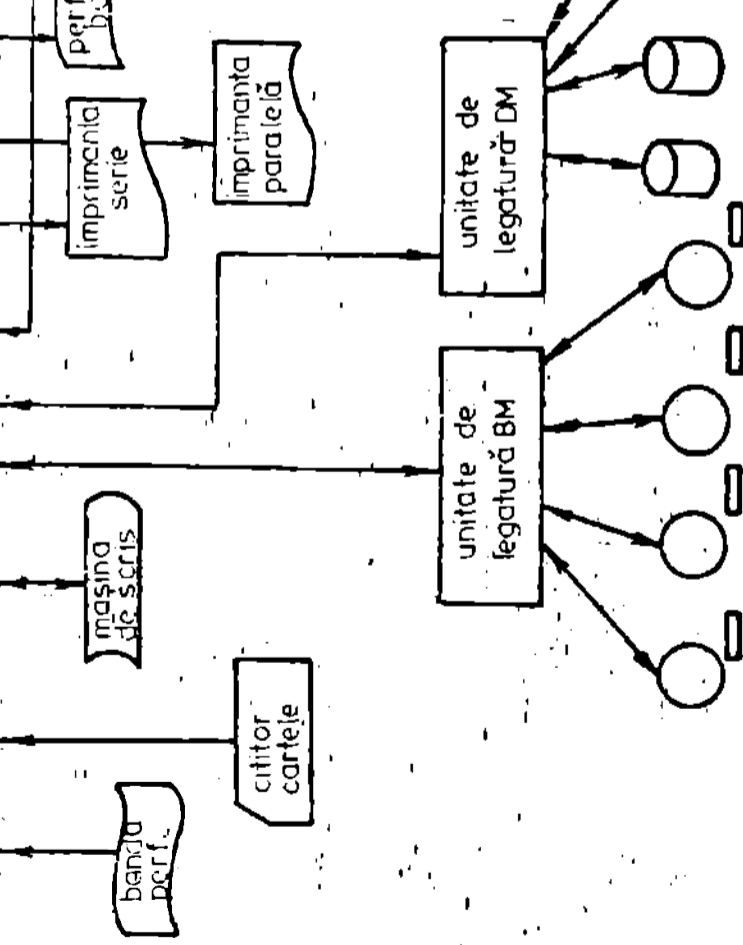
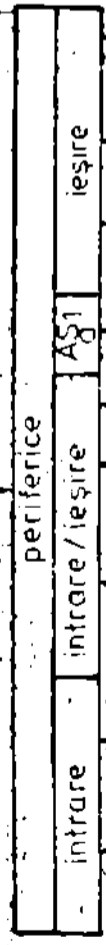
Legătura MCFN - ROBOTRON a fost realizată printr-o interfață reprezentată în fig.3.16 care ține cont de protocolul de semnale ROBOTRON - periferic standard precum și de necesitatea transmiterii sau recepției cu bit de paritate, care face ca din cei opt biți ai canalului de comunicație doar șapte să fie utili.

S-a ținut cont că în sistemul său de operare ROBOTRONUL "vede" minisistemul MCFN ca pe un periferic standard și anume lector de bandă (pentru intrare) sau perforator de bandă (pentru

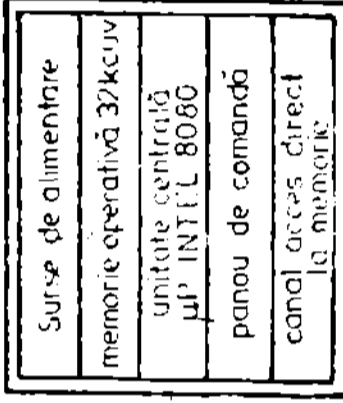
calculatoare vintilice



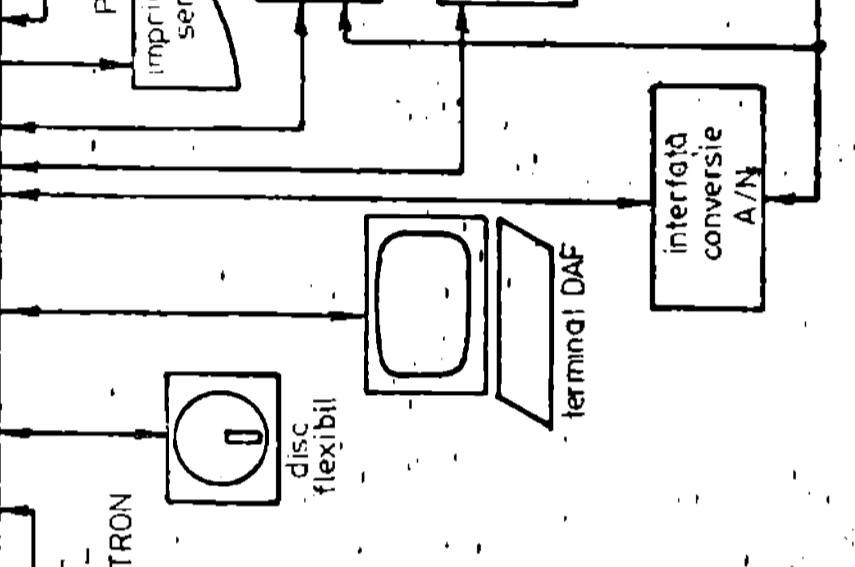
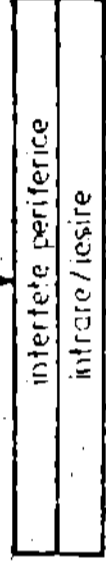
minicalculator
robotron 7201



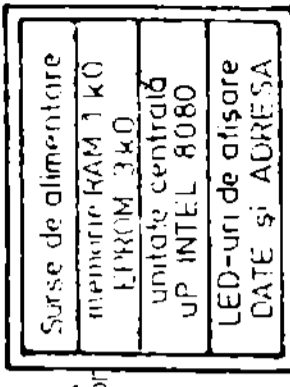
calculatoare cu arhitectura minicomput



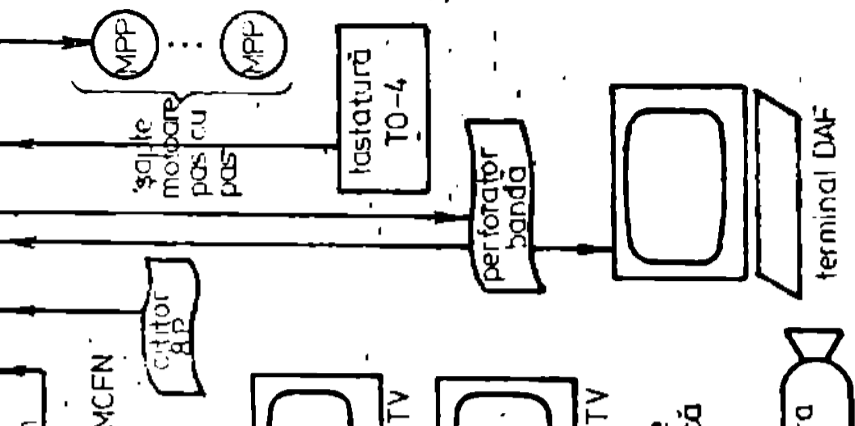
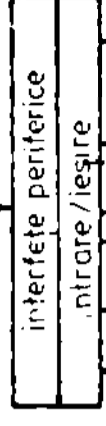
minicalculator
MCFN



robot PLURICA



microcalculator
inbarcat
INTEL 8080



unități de bandă magnetică discuri magnetice cartridge

ieşire), iar MCFN-ul "vede" sistemul de calcul ROBOTRON ca pe un port de intrare/ieşire cu o adresă cunoscută.

Pe partea ROBOTION-ului dialogul este rezolvat de către programele de gestiune a perifericilor specificând sistemului de operare perifericul respectiv printr-o deceleraţie de configuraţie de aparatură periferică.

La declanşarea programului de transmisie a imaginii se specifică adresa de început a zonei de memorie ce va fi transmisă şi lungimea ei. Informaţia pe un octet este divizată în doi semiocteţi înregistraţi cu partea cea mai puţin semnificativă a octetului care se transmite. Pentru a deveni cod ASCII, semiocetelul cel mai semnificativ este "forţat" la valoarea 4H (olăo binar), ocazie cu care caracterul ce este recepţionat în ROBOTRON este considerat cu aceeaşi valoare în cod SYF.

Programul de dialog este structurat în două părţi distincte indicate de primul caracter la utilizare :

a) transfer de date MCFN-ROBOTION (primul caracter este "Z").

În acest caz se calculează bitul de paritate şi se marchează în bitul cel mai semnificativ al octetului ce se va transmite; se apelează subprogramul IUP-S. Acesta testează fanionul IUP-E (adresa A054H) care indică dacă ROBOTION-ul este în aşteptare de informaţie. În caz afirmativ octetul se încarcă la adresa A062H şi se poziţionează fanionul END-E (adresa A063H). În continuare se testează dezactivarea semnalului IUP-E de către ROBOTION în momentul preluării octetului transferat. Semnalul END-E este dezactivat de către MCFN după ce IUP-S a fost dezactivat. Procesul se reia pînă la epuizarea numărului de octeţi indicaţi sau pînă ce ROBOTION-ul încetează recepţia de informaţii.

Trebuie remarcat că în cadrul informaţiei utile sînt intercalaţi octeţi cu valoarea 15H care delimitează blocurile de scriere în general conform specificaţiilor din programul apelant aflat în ROBOTION (FORMAT pentru FOFTRAN, lungime bloc de date în SYPS).

b) transfer de date ROBOTION-MCFN (primul caracter este 1). În acest caz, la recepţia unui octet se maschează semiocetelul mai semnificativ şi din doi semiocteţi separaţi, prin asamblare se formează un octet conţinînd doi pixeli. Codurile speciale 15H şi 00H sînt neglijate, ele corespunzînd sfîrşitului de

bloc util, respectiv interspațiul între blocuri de date. În această situație este apelat subprogramul RUF-A care în urma testării fanionului cu același nume (adresa A054H) încarcă octetul transmis (la adresa A056H) după care se poziționează fanionul END-A analog cu situația de la punctul a), procesul fiind în continuare similar.

CAPITOLUL 4

MODELELE MATEMATICE ALE ROBOTULUI P.A.T.R.I.C.A

4.1. Introducere în problematica modelelor cinematice

Problema cinematică inversă pentru o manipulare robotică este să se afle vectorul unghiurilor de articulație dându-se pentru un robot cu n - grade de libertate (GDL) poziția și orientarea (atitudinea) dispozitivului de prehensiune. Vectorul de atitudine - sau, mai general, matricea de atitudine - descrie poziția originii și orientarea de tip rotațional a unui sistem de coordonate atașat dispozitivului de prehensiune relativ la un sistem de coordonate fix aflat la baza brațului mecanic.

Nu există model cinematic invers exact cu caracter general aplicabil oricărui robot cu n - grade de libertate, nici măsură pentru brațele mecanice cu 6 GDL utilizate frecvent în industrie. Este uneori posibil să se determine modelul geometric invers pentru anumite clase speciale de structuri cu n - GDL, chiar dacă numărul operațiilor algebrice și trigonometrice devine enorm. Pentru un robot dat problema matematică se compune dintr-un set de 12 ecuații cu n necunoscute și complexitatea ei constă în aceea că nu există un set de soluții unic ba uneori nu există soluții deloc.

Din punct de vedere istoric Pieper /Pi 68/ și Pieper și Foth /Pf 69/ au fost primii care au elaborat modele cinematice inverse pentru un număr de brațe robotice speciale. De asemenea ei s-au orientat în lucrările lor spre algoritmi de rezolvare numerică iterativă a modelelor cinematice inverse. Paul /Pa 81/ a elaborat modele cinematice inverse pentru robotul Stanford (cilindric polar) și un manipulator sferic articulată cu 6 GDL. Hollerbach și Gideon /HG 83/ au obținut un număr de modele cinematice inverse pentru o clasă specială de manipulatori robotici. Practica în industrie este să se considere un manipulator robotic și să se încerce elaborarea modelelor cinematice inverse pentru a fi mai tirziu introduse în programant.

Uneori este aproape imposibil să se găsească soluții pentru problema cinematică inversă a unor roboți industriali cu 6 GDL. S-au făcut încercări pentru a trata această problemă prin mijlocirea algoritmilor iterativi. De exemplu, Lumelsky /Lu 83/ a prezentat soluții iterative pentru modele cinematice inverse destinate unui tip de manipulator robotic iar Campbell

/Ca 84/ și Shahinpoor și Campbell /Sh 84/ au utilizat tehnica de optimizare a lui Powell /Po 84/ pentru a obține un algoritm iterativ al modelului cinematic invers în cazul general al unui manipulator robotic cu n -GDL. Cu toate aceste progrese, timpul de calcul tipic pentru un robot cu 6 GDL pe un calculator VAX-11/780 a fost de 1 la 25 secunde ceea ce este inacceptabil în condiții de conducere în timp real. Trebuie de asemenea menționate soluțiile exacte recente ale lui Tsai și Morgan /TM 84/ și Shahinpoor, Jemshidi și Kim /Sh 85/.

În capitolul de față se prezintă modelele cinematice directe și inverse obținute pentru robotul de laborator P.E.T.R.I.C.A. Se subliniază de la bun început că modelele obținute au luat în considerare atât gradele de libertate oferite de partea de vehicul cât și cele ale brațului mecanic obținându-se în final modele cinematice complete (globale) ale robotului.

4.2. Algoritm de atașare a coordonatelor Hartenberg-Denavit

Brațul mecanic al robotului de laborator P.E.T.R.I.C.A. este un lanț cinematic deschis. Un capăt al acestuia este fixat de cărucior care constituie sistemul de referință de bază. La celălalt capăt se găsește dispozitivul de prehensiune acționat de un motor pas cu pas. Lanțul cinematic este format dintr-o serie de legături și cuple cinematice. Poziția capătului efector este descrisă de un vector de poziție care își are originea în sistemul de referință de bază și capătul la originea sistemului de coordonate atașat dispozitivului de prehensiune. Orientarea acestuia din urmă este exprimată de o matrice de rotații care indică rotațiile relative ale sistemului de coordonate ale dispozitivului de prehensiune față de sistemul de referință de bază. Împreună, poziția și orientarea unui sistem de coordonate față de altul pot fi descrise de o matrice de 4 ori 4 elemente care a fost denumită matricea de atitudine /Sh 84/. Aceste matrici au fost concepute și elaborate într-o lucrare clasică deja de către Hartenberg și Denavit /De 55/ și sînt cunoscute sub denumirea de matrici Denavit-Hartenberg (D-H) sau matrici A. Matricile D-H reprezintă de fapt transformări omogene de 4 ori 4 suficiente pentru a descrie proprietățile cinematice ale mecanismelor cu legături.

Coordonatele fiecărei perechi de legături sînt reprezentate prin 4 parametri: a , α , θ și d . Primul parametru a este cunoscut ca lungimea legăturii, α este torsiunea legăturii, θ este unghiul legăturii iar d este distanța legăturii. Coordonatele pentru notația D-H sînt definite în fig.4.1. În general, două legături sînt ategate printr-o cuplă cinematică care are două suprafețe în contact. Așa cum s-a menționat, o cuplă cinematică (a i-a) este stabilită de conexiunea a două legături.

Coordonatele sînt alese după cum urmează:

1. Axa de coordonate Z_i se va alinia cu axa de translație sau rotație a cuplei cinemate a (i+1)-a.
2. Axa x_i este direcționată normal la amîndouă axele z_i și z_{i-1} orientată dinspre axa z_{i-1} .
3. Axa y_i este aleasă astfel încît să formeze un sistem de coordonate $x_i y_i z_i$ cu sensul determinat de regulă mîinii drepte.
4. Distanța între cele două normale a_i și a_{i-1} este numită distanța legăturii d_i .
5. Lungimea normalei a_i este distanța minimă între axele z_i și z_{i-1} . Dacă ea este zero, direcția lui a_i devine arbitrară. Parametrul a_i este cunoscut ca lungimea legăturii i .
6. Orientarea relativă a axei x_i față de x_{i-1} este denumită unghiul $\theta = (x_i, x_{i-1})$ sau vectorul unghiurilor cuplei cinemate
7. Orientarea relativă a axei z_i față de axa z_{i-1} este denumită $\alpha = (z_i, z_{i-1})$ sau vectorul unghiului de torsiune.

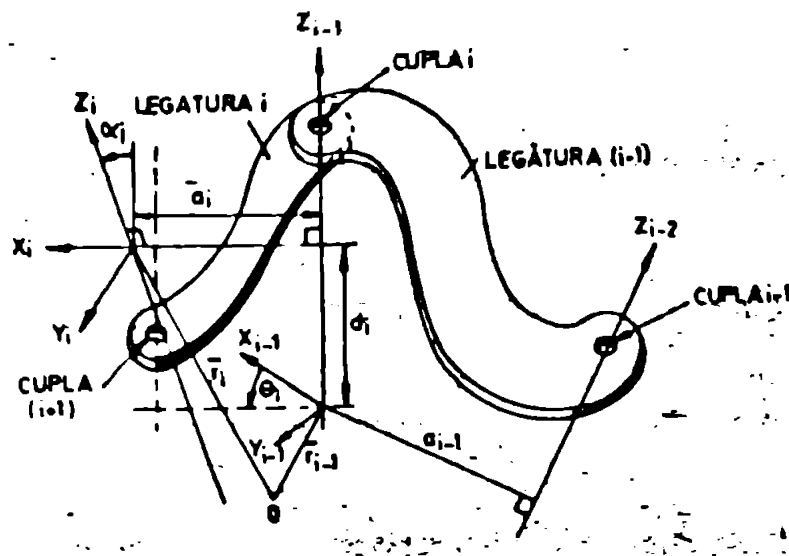


Fig.4.1

Cei patru parametri d_i , θ_i , a_i și α_i constituie un set minimal și suficient pentru a determina configurația cinematică a fiecărei legături a lanțului cinematic. Denumirile pentru cei patru parametri sînt :

d_1 = distanța legăturii, θ_1 = unghiul legăturii, a_1 = lungimea legăturii, α_1 = unghiul de torsiune al legăturii se menționează că pentru o cuplă plană de rotație, d_1 , a_1 și α_1 sînt toate constante, în timp ce θ_1 variază pe măsură ce legătura i se rotește în jurul axei cuplei i . Pe de altă parte, pentru o cuplă prismatică θ_1 , a_1 și α_1 sînt constante în timp ce d_1 variază cînd legătura i alunecă în lungul axei cuplei i .

Odată ce sistemul de coordonate D-H a fost stabilit pentru fiecare legătură, o matrice de transformare omogenă poate fi ușor exprimată referind sistemul de coordonate al i -lea la sistemul de coordonate $(i-1)$. Referindu-ne la fig.4.1, este clar că un punct \bar{r}_i exprimat în sistemul de coordonate i poate fi exprimat în sistemul de coordonate $(i-1)$ ca \bar{r}_{i-1} prin efectuarea următoarelor transformări succesive:

1. Rotație în jurul axei z_{i-1} cu un unghi θ_1 pentru a alinia axele x_{i-1} și x_i notată, $R(z_{i-1}, \theta_1)$.
2. Translație în lungul axei z_{i-1} pe o distanță d_1 pentru a aduce axele x_{i-1} și x_i în coincidență, notată $T(z_{i-1}, d_1)$.
3. Translația de-a lungul axei x_i pe o distanță a_1 pentru a aduce originile în coincidență, notată $T(x_i, a_1)$.
4. Rotație în jurul axei x_i cu un unghi α_1 pentru a suprapune complet cele două sisteme de coordonate, notată $R(x_i, \alpha_1)$.

Astfel transformarea compusă și completă a legăturii i relativă la legătura $i-1$ sau a cuplei i relativ la cupla $i-1$ este :

$$A_{i-1}^i = R(z_{i-1}, \theta_1) T(z_{i-1}, d_1) T(x_i, a_1) R(x_i, \alpha_1) \quad (4.1)$$

sau matricial

$$A_{i-1}^i = \begin{vmatrix} C\theta_1 & -S\theta_1 & 0 & 0 \\ S\theta_1 & C\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & a_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_1 & -S\alpha_1 & 0 \\ 0 & S\alpha_1 & C\alpha_1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.2)$$

$$= \begin{vmatrix} C\theta_1 & -C\alpha_1 S\theta_1 & S\alpha_1 S\theta_1 & a_1 C\theta_1 \\ S\theta_1 & C\alpha_1 C\theta_1 & -S\alpha_1 C\theta_1 & a_1 S\theta_1 \\ 0 & S\alpha_1 & C\alpha_1 & d_1 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

unde $S\theta_1 = \sin\theta_1$, $C\theta_1 = \cos\theta_1$, etc sînt folosite pentru o notație mai compactă. Similar, $S_{ij} = S\theta_{ij} = \sin(\theta_i + \theta_j)$, $C_{ijk} = \cos(\theta_i + \theta_j + \theta_k)$.

În mod frecvent se poate omite subîndiciera cu $i-1$ și în acest caz A_{i-1} devine A_i . În general A_i are forma următoare:

$$A_{i-1}^i = A_i = \begin{vmatrix} n_{xi} & a_{xi} & p_{xi} & 0 \\ n_{yi} & a_{yi} & p_{yi} & 0 \\ n_{zi} & a_{zi} & p_{zi} & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.3)$$

Inversa acestei matrici se notează A_i^{-1} și are forma:

$$A_i^{-1} = A_{i-1}^{-1} = \begin{vmatrix} n_{xi} & n_{yi} & n_{zi} & -(p.n) \\ o_{xi} & o_{yi} & o_{zi} & -(p.o) \\ a_{xi} & a_{yi} & a_{zi} & -(p.a) \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.4)$$

Mai departe dacă -1 este coordonata omogenă a cuplei i și $i-1$ este coordonata omogenă a cuplei $(i-1)$ este clar că:

$$\bar{r}_i = A_{i-1}^i \cdot \bar{r}_{i-1} \quad (4.5)$$

$$\text{unde } \bar{r}_i = (x_i, y_i, z_i, 1)^T \quad (4.6)$$

$$\bar{r}_{i-1} = (x_{i-1}, y_{i-1}, z_{i-1}, 1)^T \quad (4.7)$$

Pentru atașarea coordonatelor fiecărei legături se poate folosi o procedură algoritmică. Algoritmul atribuie sisteme de coordonate fiecărei legături în conformitate cu transformările D-H pentru un braț robotic cu n GDL. Legăturile adiacente pot fi puse în corespondență una cu alta din punct de vedere cinematic printr-un set de matrici de transformare 4 ori 4. Primul sistem de coordonate atașat sistemului de referință de bază este notat $(x_0, y_0, z_0, 1)^T$. Originea acestui sistem este denumită cuplă cinematică zero.

Algoritmul 4.1

pas 1. Se stabilește sistemul de coordonate de bază (x_0, y_0, z_0) astfel ca axa z_0 să fie orientată paralel cu axa de mișcare a cuplei 1.

pas.2. Se inițializează și se buclează pentru fiecare $i=1, 2, \dots, n$.

- pas.3. Se aliniază toate axele z_i cu axele de translație sau rotație ale cuplei $i+1$.
- pas.4. Se stabilește originea sistemului de coordonate i , sau la intersecția axelor z_i și z_{i-1} , sau la intersecția normalei comune între axele z_i și z_{i-1} și axa z_i .
- pas.5. Se stabilește axa x_i a fiecărei cuple, sau după direcția $x_i = \pm (\bar{z}_{i-1} \times \bar{z}_i)$ sau de-a lungul normalei comune între z_{i-1} și z_i dacă acestea sînt paralele.
- pas.6. Se stabilește axa y_i după direcția $\bar{y}_i = (\bar{z}_i \times \bar{x}_i)$ pentru a forma sistemul de coordonate rectangular.
- pas.7. Se află d_i ca distanța de la origină sistemului de coordonate $(i-1)$ la intersecția axelor z_{i-1} și x_i .
- pas.8. Se află a_i ca distanța de la intersecția axelor z_{i-1} cu axa x_i la originea sistemului de coordonate i .
- pas.9. Se determină θ_i ca unghiul de rotație între axa x_{i-1} și axa x_i ; $\theta_i = (x_{i-1}, x_i)$.
- pas.10. Se determină α_i ca unghiul de rotație între axele z_{i-1} și z_i de-a lungul axei x_i ; $\alpha_i = (z_{i-1}, z_i)$

Față de schema cinematică obișnuită a unui manipulator robotic care se prezintă sub forma unui lanț deschis, pentru robotul de laborator P.A.L.K.I.C.A. au fost luate în considerare și gradele de libertate oferite de mișcările pe roți ale vehiculului. S-au ales, dintre cele patru cuple cinematice instantanee posibile de realizat de către roțile vehiculului, două și anume: rotație (R_1) în jurul roții acționată de motorul 8 prin acționarea doar a motorului 7 și o translație (T_2) realizată prin rotirea în același sens cu același număr de pași a ambelor motoare 7 și 8. (fig.4.2).

Referitor la brațul mecanic s-au rigidizat cuplele din număr R_3 și cea de susținere R_5 a antebrățului. Rezultă în noua variantă o schemă cinematică care are formula R(rotație)-T(translație)-R-R.

De observat faptul că nu pot coexista deodată cuplele R_1 și T_2 realizate de vehicul.

Folosind algoritmul de atașare a coordonatelor D-H se obține figura 4.3. Trebuie făcute următoarele remarci în legătură cu această operație. Sistemul de referință de bază (x_0, y_0, z_0) are axa z_0 aliniată după cupla instantanee de rotație R_1 . Un sistem de coordonate, atașat vehiculului în dreptul roții în jurul căreia se pivotează, notat (x_v, y_v, z_v) indică poziția vehiculului.

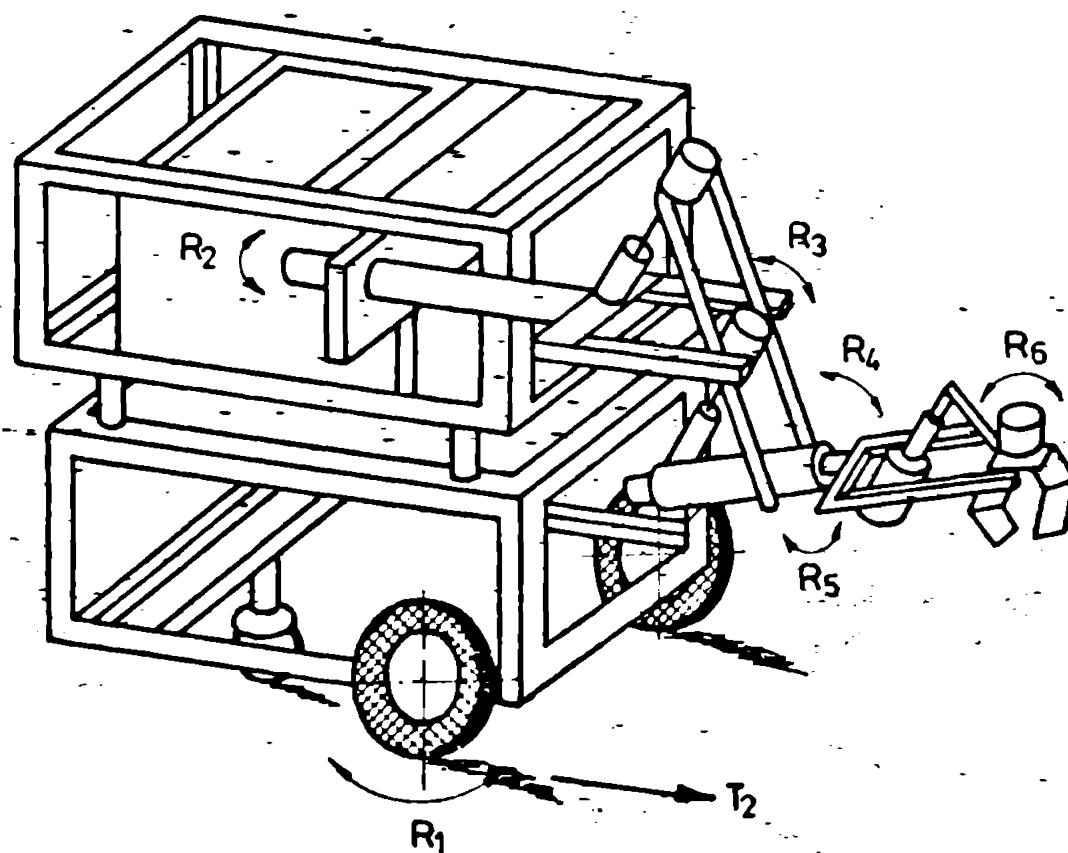


Fig.4.2.

lui față de sistemul de coordonate de bază. Bazat pe fig.4.3 se poate construi tabelul general 4.1 pentru parametrii cuplilor cinematice. **4.3. Modelul cinematic direct**

Descrierea capătului efector al brațului mecanic termenii transformărilor omogene este dată de

$${}^{n-1}T_6 = A_n A_{n+1} \dots A_6 \quad (4.8)$$

relativ la sistemul de coordonate \$(n-1)\$. Descrierea capătului oricărei legături \$i\$ relativ la sistemul de coordonate \$j\$ este :

$${}^jT_i = A_{j+1} A_{j+2} \dots A_i, \quad j < i \quad (4.9)$$

În general capătul manipulatorului robotic, adică legătura 6 pentru cazul în care sînt 6 GDL, față de sistemul de referință de bază este exprimat astfel:

$$T_6 = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6 \quad (4.1c)$$

Subliniem că pentru o cuplă de rotație, toți parametri sînt constanți cu excepția unghiului \$\theta\$. Pentru o cuplă prismatică singura variabilă este distanța \$d\$ în timp ce ceilalți parametri sînt menținuți constanți. Cei patru parametri furnizează sufi-

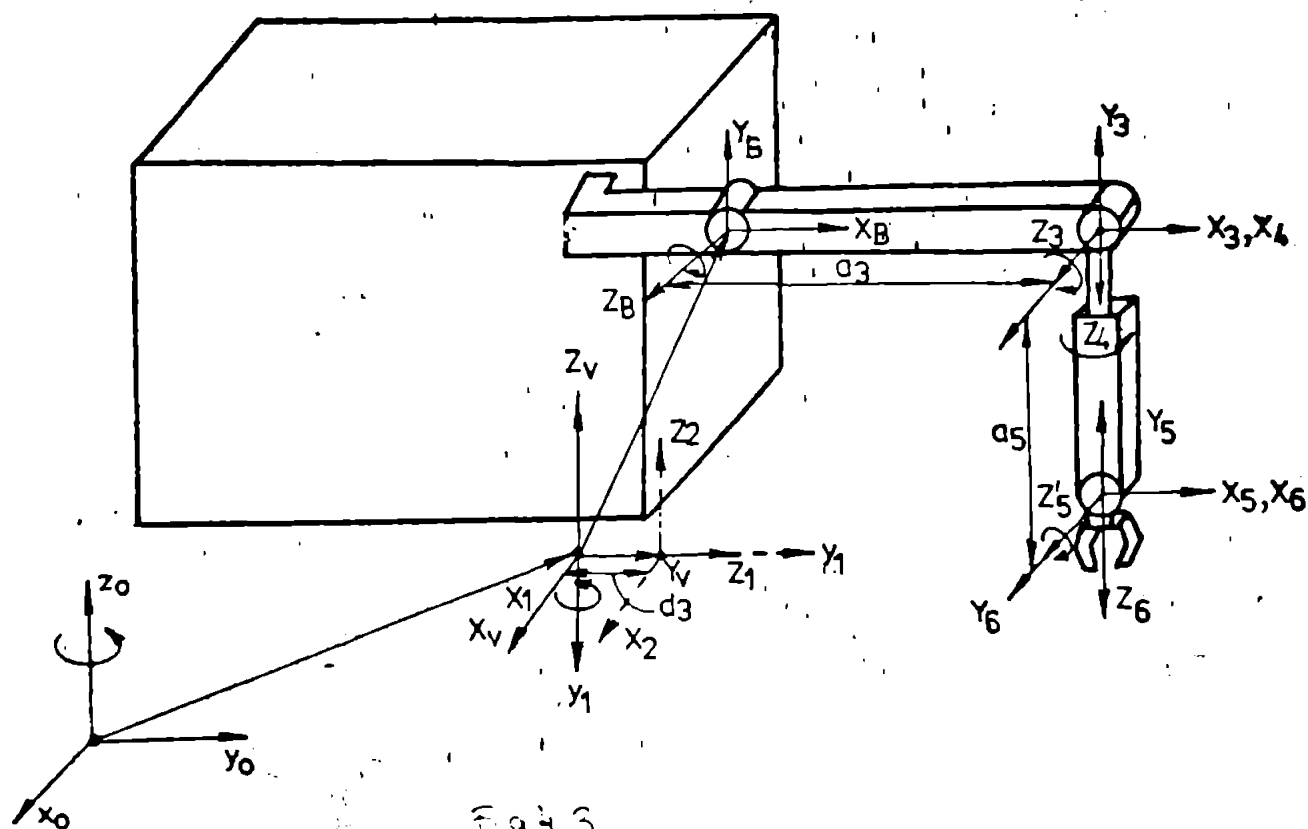


Fig. 4.3

Tab 4.1

NR. CUPLEI	TIPUL CUPLEI	α	a	d	VARIABILA	$\cos \alpha$	$\sin \alpha$
1	R	-90°	0	0	θ_1	0	-1
2	T	90°	0	d_2	d_2	0	1
- BRAT transformare constantă ce exprimă poziția fixă a brațului pe cărucior							
3	R	0°	a_3	0	θ_3	0	1
4	R	90°	0	0	θ_4	0	1
5	R	-90°	a_5	0	θ_5	0	-1
6	R	90°	0	0	θ_6	0	1

cientă informatică mentală a reprezenta matematic un lanț cinematic. O dată ce au fost obținute parametrii Denavit-Hartenberg se pot exprima matrici de transformare omogenă ce indică trecerea

de la un sistem de coordonate la altul. Astfel fiecare matrice A furnizează mijlocul de a trece de la un capăt al unei legături la altul. Prin înmulțiri succesive ale matricilor A devine posibilă transformarea ce conduce de la un vector din sistemul de coordonate de bază la un vector al capătului efector. Pentru un braț robotic cu n GDL aceasta se exprimă sub forma operației matriciale:

$$T_n = A_1 A_2 A_3 A_4 A_5 A_6 \dots A_n$$

Aceasta înseamnă că dacă parametrii Denavit-Hartenberg sînt cunoscuți atunci modelul cinematic direct poate fi obținut. Produsul matricial al celor n matrici A este notat cu T_n . Aceasta este transformarea de la bază pentru determinarea capătului efector și exprimă atitudinea capătului efector în raport cu baza.

Pentru robotul de laborator P.S.T.F.I.C.A. s-a obținut bazați pe tabela 4.1 și ecuația (4.2) următoarele matrici:

$$A_0^1 = \begin{vmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.11)$$

$$A_1^2 = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.12)$$

$$A_2^{BRAT} = \begin{vmatrix} 0 & 0 & 1 & -10 \\ 1 & 0 & 0 & 280 \\ 0 & 1 & 0 & 470 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.13)$$

$$A_3^{BRAT} = \begin{vmatrix} C_3 & -S_3 & 0 & a_3 C_3 \\ S_3 & C_3 & 0 & a_3 S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.14)$$

$$A_3^4 = \begin{vmatrix} C_4 & 0 & S_4 & 0 \\ S_4 & 0 & -C_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.15)$$

$$A_4^5 = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & a_5 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.16)$$

$$A_5^6 = \begin{vmatrix} C_6 & 0 & S_6 & 0 \\ S_6 & 0 & -C_6 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.17)$$

Rezultă că produsul matricial va fi :

$$T_0^6 = \underbrace{A_0^1 \cdot A_1^2}_{\text{vehicul}} \cdot \underbrace{A_2^{BkAT}}_{\text{matrice constantă}} \cdot \underbrace{A_3^{BkAT} \cdot A_4^5 \cdot A_5^6}_{\text{braț mecanic}} \quad (4.18)$$

Dar T_0^6 are elementele din fig.(4.3) în consecință se obțin după înmulțire, următoarele relații :

$$n_x = -S_1 C_{346} \quad (4.19)$$

$$n_y = C_1 C_{346} \quad (4.20)$$

$$n_z = S_{346} \quad (4.21)$$

$$o_x = C_1 \quad (4.22)$$

$$o_y = S_1 \quad (4.23)$$

$$o_z = 0 \quad (4.24)$$

$$a_x = -S_1 S_{346} \quad (4.25)$$

$$a_y = C_1 S_{346} \quad (4.26)$$

$$a_z = -C_{346} \quad (4.27)$$

$$p_x = -a_5 S_1 S_{34} - a_3 S_1 C_3 - 10C_1 - 280S_1 + d_2 \cdot 1 \quad (4.28)$$

$$p_y = a_5 C_{34} + a_3 C_1 C_3 - 10S_1 + 280C_1 + d_2 C_1 \quad (4.29)$$

$$p_z = -a_5 C_{34} + a_3 S_3 + 470 \quad (4.30)$$

In cazul in care se urmărește obținerea soluției directe în timp real se folosesc relațiile finale (4.19-4.30) care furnizează poziția capătului efector în situația în care se cunoaște starea fiecărei articulații precum și poziția vehiculului. Calculul a trei coloane din dreapta lui T_6 solicită 6 funcții trigonometrice, 18 înmulțiri și 10 adunări. Prima coloană a lui T_6 se obține ca produs vectorial al celei de-a doua și a treia coloane.

4.4. Modelul cinematic invers

Înainte de a trece la determinarea modelului cinematic invers exact al robotului P.L.T.F.I.C.A., trebuie să menționăm că Paul /Pa 81/ a conturat o procedură care poate fi utilizată pentru a genera modele cinematice inverse pentru câteva clase de manipulatori robotici. Fundamental, procedura include operații trigonometrice și regrupări de matrici pentru a asigura egalitatea element cu element a matricii T_n cu vectorul de atitudine. Trebuie să deosebim de asemenea menționate lucrarea lui Pieper și Roth /PR69/ și cea a lui Duffy și Rooney /Du.a/ și Duffy /Du.b/ care au solicitat legile sinus și cosinus ale triunghiurilor spațiale mecanismului spațial echivalent al unui manipulator și au obținut soluții corecte. În cele ce urmează se vor utiliza rezultatele acestor cercetări.

În mod obișnuit se cunoaște poziția capătului efector prin matricea de atitudine și se cer determinările necunoscutele θ_1 și d_1 în termenii componentelor lui T_0^6 și altor parametri structurați.

Pentru manipulatori robotice T_0^6 este cunoscut și este egal cu produsul celor 6 matrici A (ecuația 4.18). Se pot obține șase ecuații matriciale prin înmulțirea succesivă cu inversele matricilor A în ecuația 4.18 după cum urmează

$$(A_0^1)^{-1} \cdot T_0^6 = T_1^6 \quad (4.31)$$

$$(A_1^2)^{-1} \cdot (A_0^1)^{-1} \cdot T_0^6 = T_2^6 \quad (4.32)$$

$$(A_{BFAT}^3)^{-1} (A_2^{BFAT})^{-1} (A_1^2)^{-1} \cdot (A_0^1)^{-1} \cdot T_0^6 = T_3^6 \quad (4.33)$$

$$(A_3^4)^{-1} (A_{BFAT}^3)^{-1} (A_2^{BFAT})^{-1} (A_1^2)^{-1} \cdot (A_0^1)^{-1} \cdot T_0^6 = T_4^6 \quad (4.34)$$

$$(A_4^5)^{-1} (A_3^4)^{-1} (A_2^3)^{-1} (A_{BFAT}^3)^{-1} (A_2^{BFAT})^{-1} (A_1^2)^{-1} \cdot (A_0^1)^{-1} \cdot T_0^6 = T_5^6 \quad (4.35)$$

Elementele matricii din membrii stângi ai acestor ecuații

sint funcții ale elementelor lui T_0^6 și primelor $n-1$ variabile ale articulațiilor. elementele matricii din partea dreaptă sint egale cu zero, constante sau funcții de cele n la 6 variabile. Deoarece egalitatea matricilor implică egalitatea element cu element se obțin cîte 12 ecuații din fiecare egalitate matricială, adică cîte p ecuație pentru fiecare din vectorii n, o, a și p .

Înmulțind ecuația 4.18 cu $(A_0^1)^{-1}$ se obține :

$$(A_0^1)^{-1} \cdot T_0^6 = A_1^2 \cdot A_2^{BRAT} \cdot A_3^{BRAT} \cdot A_4^5 \cdot A_5^6 \quad (4.36)$$

Membrul stîng al ecuației 4.36 este dat de

$$(A_0^1)^{-1} \cdot T_0^6 = \begin{vmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -S_1 & C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.37)$$

sau cu alte notații :

$$(A_0^1)^{-1} \cdot T_0^6 = \begin{vmatrix} f_{11}(n) & f_{11}(o) & f_{11}(a) & f_{11}(p) \\ f_{12}(n) & f_{12}(o) & f_{12}(a) & f_{12}(p) \\ f_{13}(n) & f_{13}(o) & f_{13}(a) & f_{13}(p) \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.38)$$

unde

$$f_{11} = C_1 x + S_1 y \quad (4.39)$$

$$f_{12} = -z \quad (4.40)$$

$$f_{13} = -S_1 x + C_1 y \quad (4.41)$$

în care x, y, z , reprezintă componentele vectorilor n, o, a, p .

Partea dreaptă a ecuației 4.36 se obține înmulțind matricile $A_1^2 \dots A_5^6$ ca mai jos

$$T_6^1 = \begin{vmatrix} 0 & 1 & 0 & -l_0 \\ -S_{346} & 0 & C_{346} & a_5 C_{34} - a_3 S_3 - 470 \\ C_{346} & 0 & S_{346} & a_5 S_{34} + a_3 C_3 + 280 + d_2 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.42)$$

Se observă că elementul de pe linia întii coloana a patra

nu depinde de vreo variabilă. Se obține egalitatea

$$f_{11}(p) = -l_0 \quad (4.43)$$

sau $C_1 p_x + S_1 p_y = -l_0 \quad (4.44)$

Pentru rezolvarea unei ecuații trigonometrice avind forma de mai sus se fac următoarele substituții

$$p_x = r \cdot \cos \theta \quad (4.45)$$

$$p_y = r \cdot \sin \theta \quad (4.46)$$

unde

$$r = +\sqrt{p_x^2 + p_y^2} \quad (4.47)$$

$$\theta = \tan^{-1} \left(\frac{p_y}{p_x} \right) \quad (4.48)$$

Substituind p_x și p_y în ecuația 4.44 se obține :

$$\cos \theta \cos \theta_1 + \sin \theta \sin \theta_1 = -l_0/r \quad (4.49)$$

cu restricția

$$0 < -l_0/r \leq 1$$

Ecuația 4.49 devine

$$\cos(\theta - \theta_1) = -l_0/r \quad (4.50)$$

cu

$$0 < \theta - \theta_1 < 180^\circ$$

Sinusul se poate obține ca

$$\sin(\theta - \theta_1) = \pm \sqrt{1 - (-l_0/r)^2} = \pm \frac{\sqrt{r^2 - l_0^2}}{r} \quad (4.51)$$

În final se obține prin împărțire

$$\theta_1 = \arctg(p_y/p_x) - \arctg(\sqrt{r^2 - l_0^2}/l_0) \quad (4.52)$$

Egalind, mai departe, elementele de pe linia a doua coloana a patra din membrul stîng al ecuației 4.42 se obține :

$$f_{12}(p) = -p_z \quad (4.53)$$

$$-p_z = a_5 C_{34} - a_3 S_3 - 470 \quad (4.54)$$

Deoarece cupla F_3 este o axă redundantă se adoptă $\theta_3 = -90^\circ$ caz în care $C_3 = 0$ și $S_3 = -1$. Astfel se obține :

$$-p_z = a_3 - a_5 S_4 - 470 \quad (4.55)$$

Din această relație se poate exprima S_4 după cum urmează :

$$S_4 = \frac{470 - a_3 - p_z}{a_5} \quad (4.56)$$

Rezultă pentru valoarea lui C_4 relația :

$$C_4 = \sqrt{1 - \frac{(470 - a_3 - p_z)^2}{a_5^2}} = \frac{\sqrt{a_5^2 - (470 - a_3 - p_z)^2}}{a_5} \quad (4.57)$$

Prin împărțirea ecuației 4.56 cu 4.57 se obține:

$$\operatorname{tg} \theta_4 = \frac{470 - a_3 - p_z}{\sqrt{a_5^2 - (470 - a_3 - p_z)^2}} \quad (4.58)$$

și evident :

$$\theta_4 = \operatorname{arctg} \frac{470 - a_3 - p_z}{\sqrt{a_5^2 - (470 - a_3 - p_z)^2}} \quad (4.59)$$

Mai departe egalând elementele de pe linia a treia coloana a patra din ecuația 4.42 avem :

$$a_5 S_{34} + a_3 C_3 + 280 + d_2 = -S_1 p_x + C_1 p_y \quad (4.60)$$

Deoarece s-a adoptat pentru θ_3 valoarea de -90° se obține:

$$a_5 C_4 + 280 + d_2 = -S_1 p_x + C_1 p_y \quad (4.61)$$

Se poate calcula d_2 astfel :

$$d_2 = -S_1 p_x + C_1 p_y - 280 + a_5 C_4 \quad (4.62)$$

În care relația d_2 este exprimat funcție de θ_1 și θ_4 deja cunoscute prin relațiile 4.52 și 4.59.

În sfârșit egalând elementele de pe linia a doua și a trei coloana a treia din ecuația 4.42 avem :

$$C_{346} = -a_z \quad \text{și} \quad (4.63)$$

$$S_{346} = -S_1 a_x + C_1 a_y \quad (4.64)$$

Prin împărțirea relației 4.64 și 4.63 rezultă:

$$\operatorname{tg}(\theta_3 + \theta_4 + \theta_6) = \frac{S_1 a_x - C_1 a_y}{a_z} \quad (4.65)$$

Dacă se cunoaște $\theta_3 = -90^\circ$ se poate determina θ_6

$$\theta_4 + \theta_6 - 90^\circ = \operatorname{arctg} \frac{S_1 a_x - C_1 a_y}{a_z} \quad (4.66)$$

sau mai departe :

$$\operatorname{tg}(\theta_4 + \theta_6 - 90^\circ) = -\operatorname{ctg}(\theta_4 + \theta_6) \quad (4.67)$$

$$\theta_6 = \operatorname{arctg} \frac{-a_z}{a_x \sin \theta_1 - a_y \cos \theta_1} - \theta_4 \quad (4.68)$$

În concluzie prin relațiile 4.52, 4.53, 4.62 și 4.68 au fost calculate variabilele cinematice θ_1 , d_2 , θ_3 (adoptat) θ_4 și θ_6 care permit trecerea din coordonatele carteziene în coordonatele generalizate ale robotului PATRICA.

4.5. Algoritm și program de transformare de coordonate

Pentru mulți roboți implementarea modelului cinematic invers sub formă de program pe calculator se mai numește și transformare de coordonate.

Înainte de prezentarea acestui subprogram se vor discuta aspectele legate de cazurile de degenerare ale modelului matematic.

Astfel în expresia 4.52 a lui θ_1 posibilitatea ca atât numărătorul p_y cât și numitorul p_x să aibă valoare zero este posibilă caz în care robotul pierde un grad de libertate.

În relația 4.59 apare din nou un caz de degenerare dacă atât numărătorul cât și numitorul devin zero. În plus se mai pune condiția ca sub radical să nu apară numere negative prin urmare.

$$a_5 \geq 470 - a_3 - p_z \quad (4.69)$$

În fine un ultim caz de degenerare poate apărea în relația 4.68 pentru situația când $a_z = 0$ și numitorul $a_x \sin \theta_1 - a_y \cos \theta_1 = 0$.

Prezentăm în continuare subprogramul CAL care calculează coordonatele generalizate notate T_1 , T_2 , T_4 și T_6 pornind de la elementele matricii de atitudine.

rei.

Algoritm 4.2, are următorii pași :

- pas 1. Se calculează termenul al doilea din expresia 4.52
- pas 2. Se calculează primul termen din expresia 4.52
- pas 3. Dacă $p_x = 0$ se forțează p_x la valoarea 0,01
- pas 4. În funcție de semnele lui p_x și p_y se calculează θ_1
- pas 5. Se calculează θ_4 cu expresia 4.59
- pas 6. Se calculează θ_6 cu expresia 4.68
- pas 7. Se calculează θ_2 cu expresia 4.62.

Capitolul 5

SISTEMELE DE VEDERE ARTIFICIALA ALE ROBOTULUI

5.1. Introducere in problematica vederii artificiale pentru roboti

Prelucrarea imaginilor vizuale cu calculatorul a constituit subiectul unui enorm efort de cercetare in ultimii 25 de ani. /Cl 71/ /DH 73/ /PB 75/ /Pe 77/ /RM 77/ /Re 78/ /TB 79/. Aproape imediat după invenția calculatorului au existat preocupări de conectare a camerelor de televiziune la calculatoare pentru a realiza mașini ce pot vedea. Această tehnologie a avut unele succese limitate în ce privește interpretarea fotografiilor seriene, ghidarea rachetelor, numărarea celulelor roșii sanguine sau a cromosomilor. În momentul de față, vederea artificială cu calculator sau mai pe scurt vederea cu calculatorul (VCC), și-a sporit flexibilitatea și s-a ieftinit. Folosirea acestei tehnologii senzoriale care dotează roboții cu un grad de „inteligentă” sporit în scopul creșterii capacității de interacțiune cu mediul /Ce 81/ primește o atenție deosebită. Un robot care poate „vedea” este mai ușor de aplicat în efectuarea de sarcini complexe în timp ce mecanismele de comandă devin mai puțin rigide ca la roboții preprogramați /GS 82/. Un robot cu capacități senzoriale dezvoltate este mult mai ușor de adaptat într-o varietate largă de sarcini, câștigând astfel în universalitate care se traduce imediat cu obținerea de producție și întreținere mai scăzute.

Există doi factori principali care întăresc dezvoltarea vederii cu calculator /RV 8a/. Primul este cantitatea enormă de date reprezentată de o imagine TV. O imagine TV alb-negru cere aproape 8 milioane de biți de informație pentru o reproducere fidelă și cu o cadență de 25 cadre pe secundă! Această cantitate de date depășește capabilitatea de memorie a celor mai multe minicalculatoare.

Al doilea motiv este lipsa unor algoritmi sau metode de prelucrare a acestei informații care să fie suficient de rapide

și „robuste” pentru uz general. Dificultatea construirii de algoritmi rapizi este ușor de înțeles în lumina uriașului volum de date ce trebuie prelucrat pentru o imagine.

Dificultatea construirii unui algoritm robust este poate mai greu de înțeles pentru cineva familiarizat cu succesele remarcabile ale calculatoarelor în alte domenii.

Un algoritm și program sînt robusți cînd pot face față cu succes într-o mare varietate de situații și nu doar cîtorva demonstrații de laborator.

Procesul vederii cu calculatorul poate fi împărțit în cinci domenii principale: achiziție, segmentare, descriere, recunoaștere și interpretare. /GS 82/ /DH 73/. Aceste categorii sînt sugerate într-o mare măsură de modul în care multe din sistemele de vedere cu calculator au fost implementate. Desigur aceasta nu înseamnă că vederea și înțelegerea umană a imaginilor poate fi atât de net subîmpărțită, nici că aceste procese sînt independente unul de altul. De exemplu, se poate considera logic că funcțiunile de recunoaștere și interpretare sînt puternic întrepătrunse la om. Din nefericire aceste relații nu sînt încă înțelese într-atît încît să poată fi modelate analitic.

Astfel împărțirea în domeniile de mai sus poate fi considerată ca o metodă practică de abordare a sistemelor actuale de vedere dîndu-se nivelul nostru de înțelegere precum și instrumentele analitice dispînibile curent în acest domeniu.

Să examinăm mai îndeaproape necesitățile de vedere ale roboticii, deoarece o convergență a acestora cu posibilitățile actuale este de dorit. Se poate da o definiție vederii roboților (VR) ca un mijloc de a interacționa cu mediul în cazuri neprevăzute. În contextul roboticii, neprevăzutul este incertitudinea asupra poziției obiectului sau robotului. La întrebarea de ce nu se poate conserva poziția și orientarea obiectelor răspunsul este că acest lucru ar fi prea costisitor din cauza necesității introducerii unor dispozitive speciale sau transportoare de piese. Identificarea pieselor este deasemenea o funcțiune importantă ca de altfel și inspecția acestora. Se pare că toate tehnicile VCC sînt potențial aplicabile în VR. Aceste tehnici pot fi ușor adoptate la aplicații industriale, ținînd

cont de mediul structurat de obicei existent în industrie.

Sisteme VR în două dimensiuni (2D) folosind în special imagini binare sînt deja disponibile comercial și unele dintre ele în folosință. Utilizările potențiale ale VR sînt aproape nelimitate. Se pare că domeniul VR este un domeniu cheie în dezvoltarea și utilizarea echipamentelor aducătoare (de livrare). O aplicație mai generală a VR în robotică ar fi posibilitatea de a reduce dimensiunile, greutatea și complexitatea roboților actuali. Roboții actuali din punct de vedere conceptual seamănă mai mult cu mașinile-unelte decît cu analogul lor antropomorfic. Cheia de boltă pentru manipulatori robotici mai ușori, mai puțin preciși și mai ieftini este creșterea capacității de interacțiune cu mediul bazată pe dezvoltarea reacției senzoriale și îmbunătățirea interpretării acestor date.

5.2. Interfața de conversie analog-numerică cu cameră TV (digitizer)

Camerele de televiziune au posibilitatea de a furniza pentru un robot ceea ce ochii furnizează pentru om: cea mai bogată sursă de informație. Înainte însă ca un calculator să poată furniza informație utilă de la o cameră TV, semnalul analog video trebuie convertit într-o formă de reprezentare numerică aflată în memoria calculatorului. Aceasta este funcția îndeplinită de interfața de conversie analog-numerică cu cameră TV.

Procesul de conversie începe prin partiționarea imaginii într-o rețea rectangulară de elemente de imagine, sau pixeli. Pentru fiecare pixel, interfața măsoară media strălucirii, atribuie o valoare întregă numită nivel de cenușiu care corespunde strălucirii și o depune în memoria calculatorului. Rezultatul este că imaginea numerică apare ca o matrice bidimensională de nivele de cenușiu. Fiecare locație de nivel de cenușiu din matrice corespunde prin linie și coloană la locația unui pixel din imagine.

Rezoluția spațială a unei imagini numerice este o funcție de numărul de pixeli dintr-o linie și coloană definite de interfața de conversie. Cu cît este mai mare numărul de pixeli cu care eșantionată imaginea, cu atît reprezentarea ei este mai fidelă. Filmele pe 35 mm au o rezoluție teoretică de 2000x2000 pixeli

timp a momentului de explorare a cadrului precum și a intervalului de timp între acest moment și pixelii p respectiv $p+1$

- stabilirea dimensiunilor ferestrei 80×285 sau 320×285
- stabilirea adresei de început în memoria IAM de la care începe depunerea imaginii numerice.
- declanșarea procesului de extragere a pixelilor (câte doi pe o linie) până la terminarea semicadrului, după care prin reprogramarea numărătoarelor CN1 și CN2 procesul se repetă până la atingerea limitelor ferestrei.

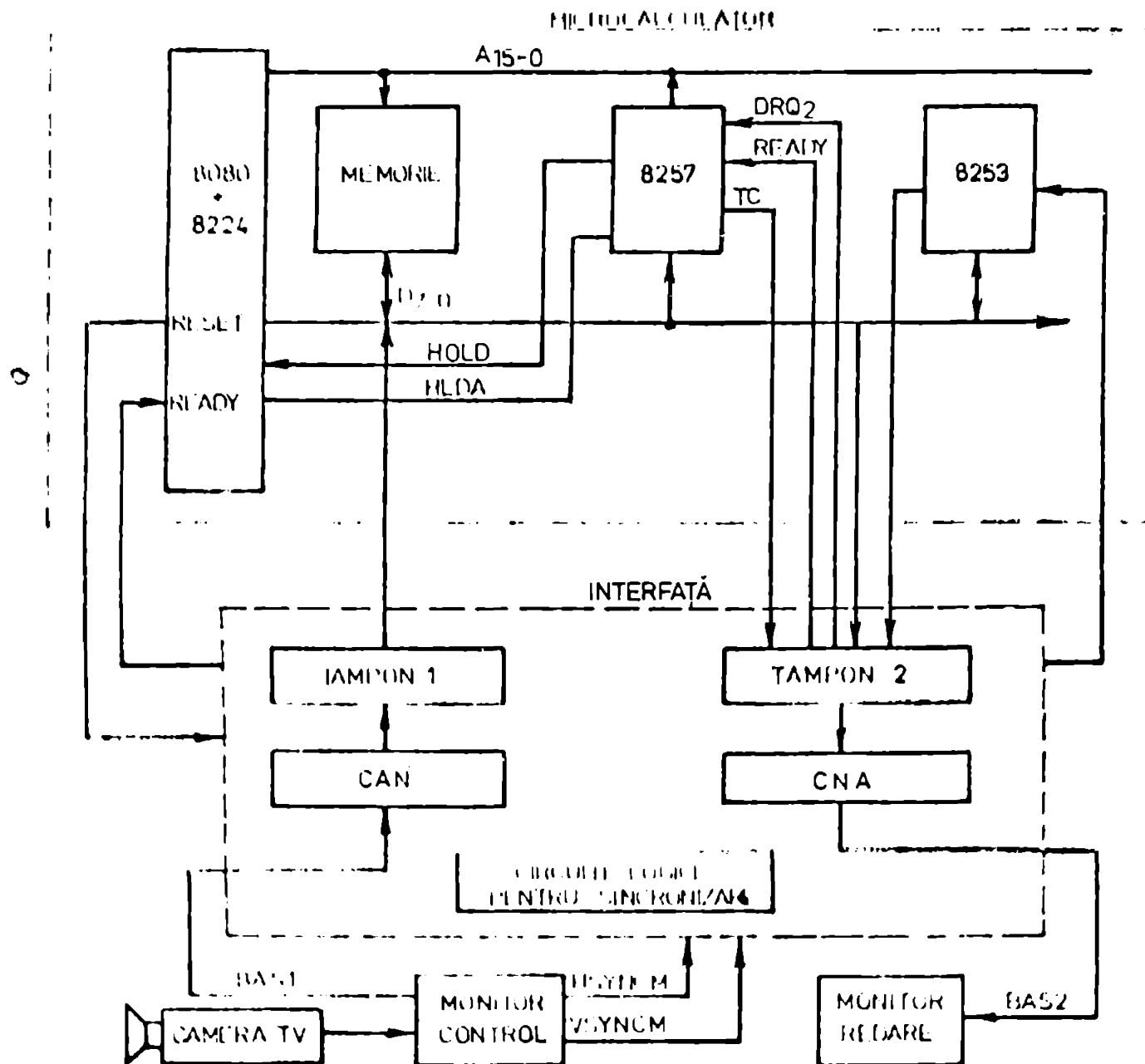


Fig.5.1

Ordinograma programului de conversie este redată în fig.5.2.

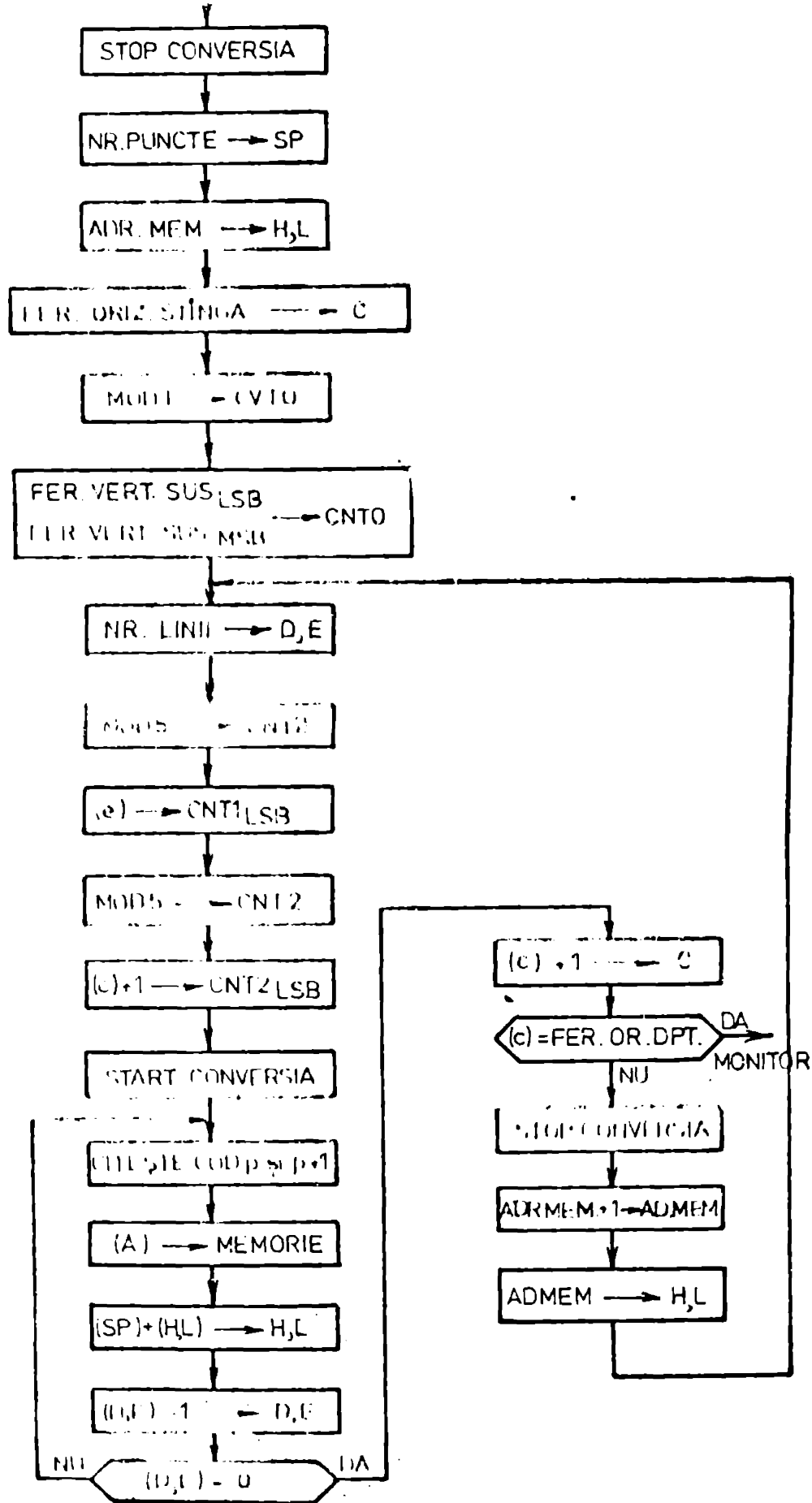
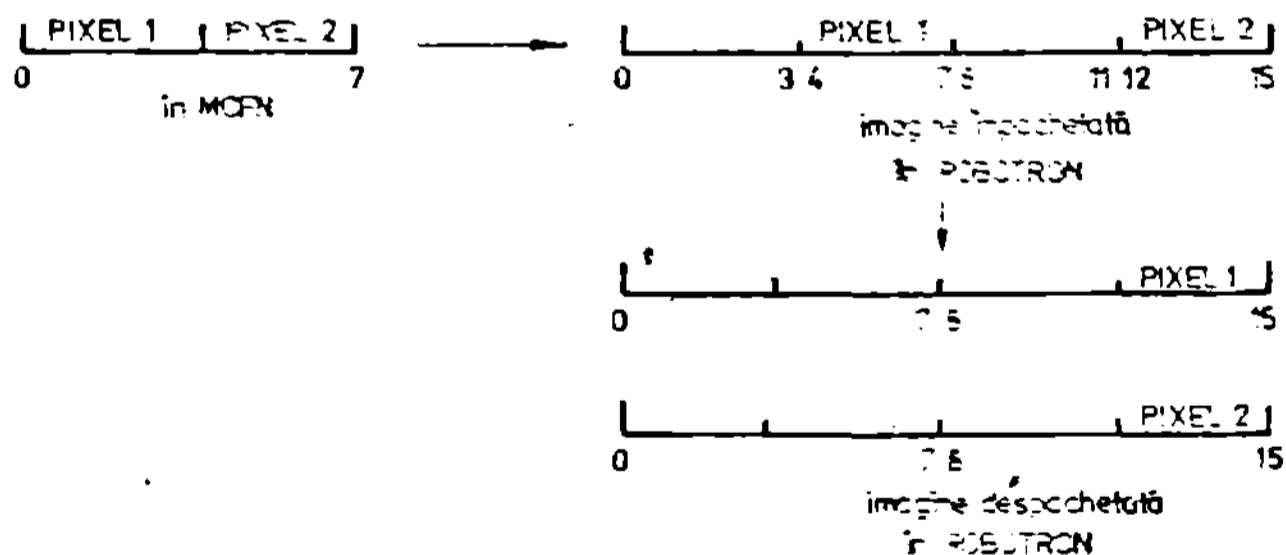


fig.5.2.

Timpu de achiziție al unei imagini este de 30 de secunde. Deoarece rezoluția este de 320 x 285 se obține un raport corect geometric de 4:3 cît este cel folosit în televiziune. Deoarece memorarea imaginii numerice necesită 45 octeți iar memoria calculatorului este de numai 32 octeți, conversia și transferul imaginii din calculatorul MCPN în minisistemul ROBOTRON se face în două etape. În prima etapă programul citește 25 octeți reprezentînd partea superioară a imaginii iar în etapa a doua 20 octeți (partea inferioară a imaginii). Dimensiunea de lucru fiind de 320 x 288 ultimele trei linii se egalează cu zero. Imaginea este transmisă spre ROBOTRON octet cu octet. În fiecare octet din MCPN sînt prezenți doi pixeli. Deoarece octetul transmis pe canalul spre ROBOTRON trebuie să conțină și bitul de paritate nu se poate transmite decît un pixel la un moment dat. În ROBOTRON recepția caracterelor se face pe 16 biți. Acest format este greu de folosit pentru o viitoare prelucrare, de aceea se preferă despachetarea adică reprezentarea unui pixel pe un cuvînt de 16 bit, cu FORMAT INTEGER în PORTAN. Subprogramul S1 scris în



PORTAN citește cite un bloc de 1 octet de date de imagine după care apelează subprogramul A6 pentru despachetarea imaginii.

Apoi porțiuni din imagine sînt scrise pe bandă magnetică sub formă de blocuri de dimensiuni fixe. Subprogramul AK este scris în limbaj de asamblare al calculatorului ROBOTECH denumit SYPS. Se primește din subprogramul apelant adresa blocului de imagine împachetate fiecare cuvînt de la această adresă fiind memorat în două locații de memorie astfel: neschimbat în prima locație de memorie și cu octeții inversați între ei în cea de-a doua locație. Ambele locații se intersectează apoi prin operația SI cu o valoare constantă ($000F+1$) pentru a păstra doar valoarea pixelului util.

5.3. Cercetări asupra prelucrării imaginilor și recunoașterii obiectelor

5.3.1. Metodologia de lucru

În prima fază nu s-a urmărit o aplicație concretă, specifică ci s-au studiat mai mulți algoritmi prezentați în literatură, dintre cei mai reprezentativi, avîndu-se în vedere pe de o parte familiarizarea cu tehnicile de lucru din prelucrarea și prelucrarea și recunoașterea imaginilor iar pe de altă parte studiul comparativ al performanțelor acestor algoritmi în special sub raportul timpului consumat.

În concepția de lucru abordată s-a considerat sistemul de calcul nu numai ca un element structural de lucru ci și ca un mijloc activ, suport, în timpul proiectării. Elaborarea unui sistem de tip VI care să lucreze în timp real este o problemă deosebit de complexă. Se citează cazul firmei Machine Intelligence Corp căruia i-au trebuit 3 ani pentru proiectarea și punerea la punct a sistemului VS-100 deși a beneficiat de toate rezultatele anterioare ale cercetărilor de la Stanford Research Institute. Se merge la început pe linia utilizării unui sistem de calcul de destinație generală și a experimentărilor bazate cit mai mult pe programe care se pot schimba și adapta foarte ușor. Abia după punerea la punct a acestora se poate trece la îmbunătățirea performanțelor încercînd transferul unor sarcini, mai ales cele critice din punct de vedere al timpului unui hardware dedicat. Transformarea sistemului experimental într-unul industrial trebuie să urmărească ceea din fig.5.3. Ceea

ce a fost o procedură sau un modul în sistemul simulat prin program poate fi transpusă într-o schemă logică specializată.

Au fost studiați un număr foarte mare de algoritmi și operatori care se încadrează în următoarele tipuri: preprelucrări - ce filtrare spațială prin convoluție, operatorul Sobel; diferențiere spațială, operatorii. Roberts, Marr-Hildreth, Husckel, Rosenfeld și McLeod, programe de achiziționare a datelor, de calcul a caracteristicilor, de învățare și recunoaștere.

5.3.2. Preprelucrarea imaginilor numerice (preprocesarea)

5.3.2.1. Operatori de convoluție

Preprelucrarea imaginilor numerice, în general, implică transformări matematice în vederea analizei imaginilor, codificări pentru a realiza compresia datelor imaginii, operații de mărire pentru detalieri (Zooming) restaurării unor imagini degradate.

O operație deosebit de importantă în procesul de filtrare este cea de convoluție. Definiția operatorului de convoluție notat ca $*$ ce se aplică asupra a două funcții g și t dintre care $g(x,y)$ este funcția imagine, se face astfel :

$$g * t(x,y) = \iint_{-\infty}^{\infty} g(\alpha,\beta) t(x-\alpha, y-\beta) d\alpha d\beta \quad (5.1)$$

Practic, convoluția este procesul de filtrare liniară prin care intensitatea luminoasă a unui pixel dintr-o imagine se modifică în funcție de valorile intensităților luminoase ale pixelilor vecini. În fig.5.4 se arată un nucleu de filtrare sub forma unei ferestre de 3×3 , centrat pe locația (i,j) în care intensitatea este $g(i,j)$. În imaginea filtrată noua intensitate a lui (i,j) va fi :

$$S(i,j) = \frac{1}{9} \sum_{u=-1}^1 \sum_{v=-1}^1 g(i-u, j-v) \quad (5.2)$$

Această sumă dublă poate fi rescrisă într-o formă ușor diferită. Mai întâi se ia pentru $t(i,j) = \frac{1}{9}$ unde atât i cât și j iau valorile $-1, 0, 1$. În acest caz $S(i,j)$ se poate re-

scrie :

$$S(i,j) = \sum_u \sum_v t(u,v)g(i-u,j-v) \quad (5.3)$$

Din punct de vedere matematic aceasta este o convoluție discretă în care imaginea $g(i,j)$ este convoluționată cu operatorul $t(i,j)$. Această dublă sumă nu este altceva decât forma discretă a integralei duble de convoluție. Rezultă că aplicarea operatorului de convoluție înseamnă decuarea unor ferestre și înlocuirea valorii intensității luminoase a pixelului central cu o sumă liniară formată cu ajutorul coeficienților pondere și nucleului înmulțiți cu valorile intensităților pixelilor din fereastră. În tabelul 5.1 sînt prezentate diferite nuclee de convoluție iar în Fig.5.5.a și 5.5.b imaginile corespunzătoare filtrate prin operator Laplace și gaussian. Timpul necesar pentru filtrare pentru o imagine 32 x 32 este de 3 minute.

$(i-1,j-1)$	(i,j)	$(i+1,j+1)$
$(i,j-1)$	(i,j)	$(i,j+1)$
$(i+1,j-1)$	$(i+1,j)$	$(i+1,j+1)$

Fig.5.4.

5.3.2.2. Netezirea spațială

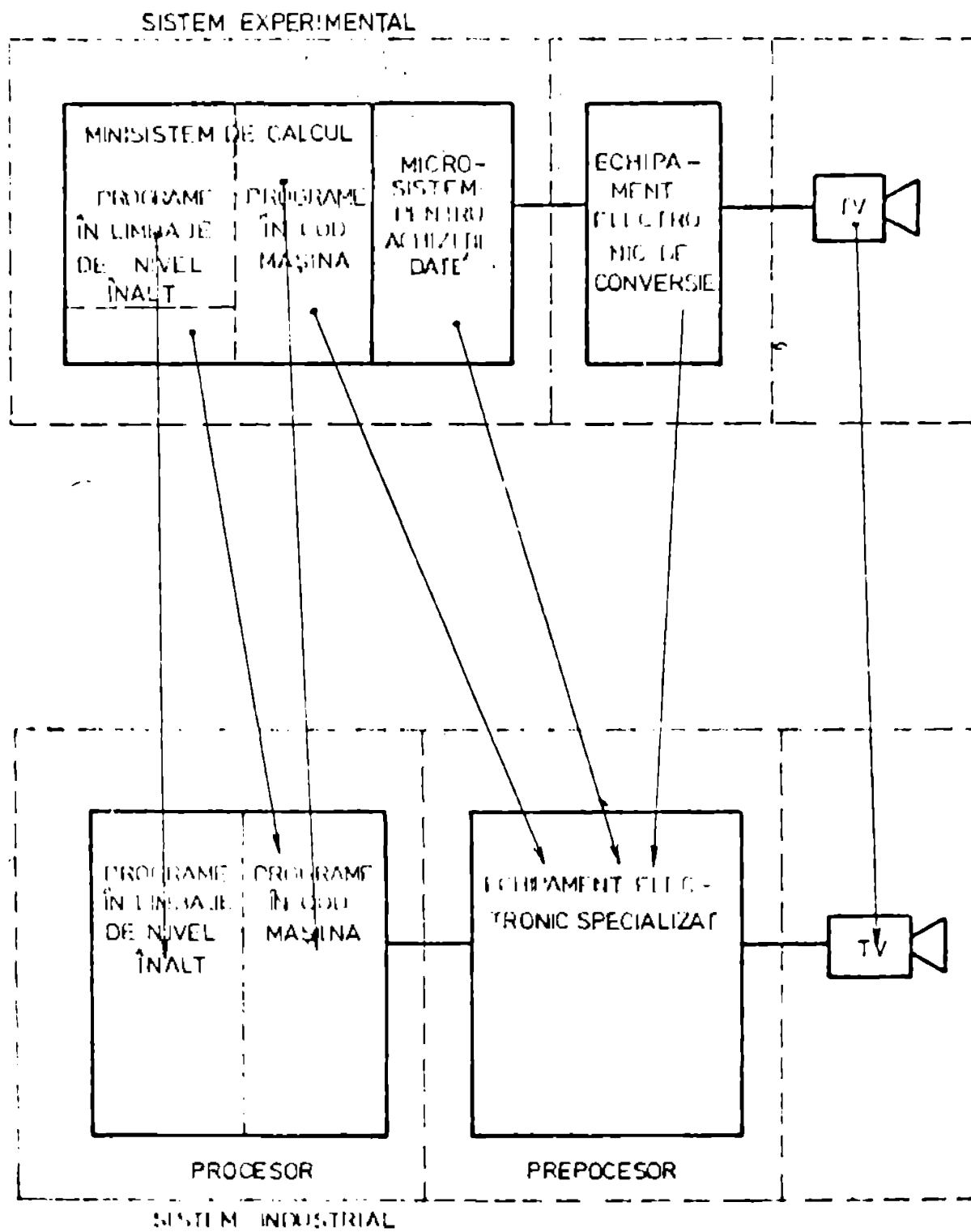
Una din metodele de eliminare a zgomotului din imagine este cea bazată pe operația de mediere sau netezire spațială (smoothing). Ideea de bază a metodei este de a înlocui valoarea intensității luminoase a unui pixel cu valoarea medie a intensităților luminoase a pixelilor din imediata lui vecinătate. Una din deficiențele metodei este și faptul că medierea elimină și detaliile. Pentru simplitate se examinează cazul funcției imagine unidimensionale $g(x)$.

$$g_w(x) = \frac{1}{w} \int_{x - \frac{w}{2}}^{x + \frac{w}{2}} g(u) du \quad (5.4)$$

Deci mărimea a ferestrei crește imaginea pierde din contrast, în caz contrar, $g_w(x)$ se apropie mult de $g(x)$. Extin-

NO	REMARKS	AMOUNT	CUMULATIVE
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

Total
 Grand Total
 ...



128.5.3

	Tab. 5.1		
Nici o filtrare	o	o	o
	o	o	o
	o	o	o
Convoluție arbitrară	a	b	c
	d	e	f
	g	h	i
Convoluție parametrică	a	b	a
	b	1	b
	a	b	a
Filtru trece-jos	1	1	1
	1	1	1
	1	1	1
Filtru trece-jos gaussian	1/4	1/2	1/4
	1/2	1	1/2
	1/4	1/2	1/4
Transformare Laplace A	o	-1	o
	-1	1	-1
	o	-1	o
Transformare Laplace B	1/4	-1/2	1/4
	-1/2	1	-1/2
	1/4	-1/2	1/4
Transformare Laplace C	-1/8	-1/8	-1/8
	-1/8	1	-1/8
	-1/8	-1/8	-1/8

Case No.	Age	Sex	Height	Weight	Temp	Pulse	Respiration	Blood Pressure	Heart	Lungs	Abdomen	Neurological	Diagnosis
1	20	M	170	65	37.5	72	18	120/80	Normal	Clear	Soft	Normal	Healthy
2	25	F	160	55	37.2	68	16	110/70	Normal	Clear	Soft	Normal	Healthy
3	30	M	180	75	37.8	78	20	130/90	Normal	Clear	Soft	Normal	Healthy
4	35	F	170	65	37.6	75	19	125/85	Normal	Clear	Soft	Normal	Healthy
5	40	M	185	80	37.9	80	21	135/95	Normal	Clear	Soft	Normal	Healthy
6	45	F	175	70	37.7	77	20	130/90	Normal	Clear	Soft	Normal	Healthy
7	50	M	190	85	38.0	82	22	140/100	Normal	Clear	Soft	Normal	Healthy
8	55	F	180	75	37.8	79	21	135/95	Normal	Clear	Soft	Normal	Healthy
9	60	M	195	90	38.1	85	23	145/105	Normal	Clear	Soft	Normal	Healthy
10	65	F	185	80	37.9	82	22	140/100	Normal	Clear	Soft	Normal	Healthy
11	70	M	200	95	38.2	88	24	150/110	Normal	Clear	Soft	Normal	Healthy
12	75	F	190	85	38.0	85	23	145/105	Normal	Clear	Soft	Normal	Healthy
13	80	M	205	100	38.3	90	25	155/115	Normal	Clear	Soft	Normal	Healthy
14	85	F	195	90	38.1	88	24	150/110	Normal	Clear	Soft	Normal	Healthy
15	90	M	210	105	38.4	95	26	160/120	Normal	Clear	Soft	Normal	Healthy
16	95	F	200	95	38.2	92	25	155/115	Normal	Clear	Soft	Normal	Healthy
17	100	M	215	110	38.5	100	27	165/125	Normal	Clear	Soft	Normal	Healthy
18	105	F	205	100	38.3	98	26	160/120	Normal	Clear	Soft	Normal	Healthy
19	110	M	220	115	38.6	105	28	170/130	Normal	Clear	Soft	Normal	Healthy
20	115	F	210	105	38.4	102	27	165/125	Normal	Clear	Soft	Normal	Healthy
21	120	M	225	120	38.7	110	29	175/135	Normal	Clear	Soft	Normal	Healthy
22	125	F	215	110	38.5	108	28	170/130	Normal	Clear	Soft	Normal	Healthy
23	130	M	230	125	38.8	115	30	180/140	Normal	Clear	Soft	Normal	Healthy
24	135	F	220	115	38.6	112	29	175/135	Normal	Clear	Soft	Normal	Healthy
25	140	M	235	130	38.9	120	31	185/145	Normal	Clear	Soft	Normal	Healthy
26	145	F	225	120	38.7	118	30	180/140	Normal	Clear	Soft	Normal	Healthy
27	150	M	240	135	39.0	125	32	190/150	Normal	Clear	Soft	Normal	Healthy
28	155	F	230	125	38.8	122	31	185/145	Normal	Clear	Soft	Normal	Healthy
29	160	M	245	140	39.1	130	33	195/155	Normal	Clear	Soft	Normal	Healthy
30	165	F	235	130	38.9	128	32	190/150	Normal	Clear	Soft	Normal	Healthy
31	170	M	250	145	39.2	135	34	200/160	Normal	Clear	Soft	Normal	Healthy
32	175	F	240	135	39.0	132	33	195/155	Normal	Clear	Soft	Normal	Healthy
33	180	M	255	150	39.3	140	35	205/165	Normal	Clear	Soft	Normal	Healthy
34	185	F	245	140	39.1	138	34	200/160	Normal	Clear	Soft	Normal	Healthy
35	190	M	260	155	39.4	145	36	210/170	Normal	Clear	Soft	Normal	Healthy
36	195	F	250	145	39.2	142	35	205/165	Normal	Clear	Soft	Normal	Healthy
37	200	M	265	160	39.5	150	37	215/175	Normal	Clear	Soft	Normal	Healthy
38	205	F	255	150	39.3	148	36	210/170	Normal	Clear	Soft	Normal	Healthy
39	210	M	270	165	39.6	155	38	220/180	Normal	Clear	Soft	Normal	Healthy
40	215	F	260	155	39.4	152	37	215/175	Normal	Clear	Soft	Normal	Healthy
41	220	M	275	170	39.7	160	39	225/185	Normal	Clear	Soft	Normal	Healthy
42	225	F	265	160	39.5	158	38	220/180	Normal	Clear	Soft	Normal	Healthy
43	230	M	280	175	39.8	165	40	230/190	Normal	Clear	Soft	Normal	Healthy
44	235	F	270	165	39.6	162	39	225/185	Normal	Clear	Soft	Normal	Healthy
45	240	M	285	180	39.9	170	41	235/195	Normal	Clear	Soft	Normal	Healthy
46	245	F	275	170	39.7	168	40	230/190	Normal	Clear	Soft	Normal	Healthy
47	250	M	290	185	40.0	175	42	240/200	Normal	Clear	Soft	Normal	Healthy
48	255	F	280	175	39.8	172	41	235/195	Normal	Clear	Soft	Normal	Healthy
49	260	M	295	190	40.1	180	43	245/205	Normal	Clear	Soft	Normal	Healthy
50	265	F	285	180	39.9	178	42	240/200	Normal	Clear	Soft	Normal	Healthy

fig 5.6

dece la cazul bidimensional se face definind o funcție $g_w(x,y)$ prin relația:

$$g_w(x,y) = 1/A_w \iint_{w(x,y)} g(u,v) du dv \quad (5.5)$$

unde $w(x,y)$ e o fereastră de arie A_w centrată în punctul (x,y) .
In cazul discret $g_w(i,j)$ este :

$$g_w(i,j) = 1/A_w \sum_{W(i,j)} g(m,n) \quad (5.6)$$

unde $w(i,j)$ e o fereastră de arie A_w centrată în pixelul (i,j) .
De obicei se utilizează ferestre de forme geometrice simple. Dacă presupunem că utilizăm o fereastră rectangulară cu baza $(2b+1)$ și înălțimea $(2h+1)$ atunci funcția de netezire ce corespunde lui $g_w(i,j)$ va fi :

$$g_w(i,j) = 1/(2b+1)(2h+1) \sum_{-h \leq n < h} \sum_{-b \leq m < b} g(i+m, j+n) \quad (5.7)$$

Operația de netezire se utilizează în diferite scopuri speciale. Deseori se utilizează la imagini binare. A netezi o imagine binară înseamnă a compara funcția $g_w(i,j)$ cu un prag. Efectul operatorului de netezire pentru o imagine nebinară și pentru una binară se poate vedea în figurile 5.6.a și 5.6.b.

5.3.3. Diferențierea spațială și extragerea conturilor

Unul din pașii primari este convertirea imaginii date într-o imagine cu contururi puse în evidență. Există baze psihologice pentru determinarea conturilor unei imagini. Experiențele au arătat că omul își concentrează atenția vizuală mai mult asupra conturilor (marginilor) dintre regiunile mai mult sau mai puțin omogene. Reducerea unei imagini la contururi introduce însă și un procent de hazard. Imaginea contur conține, în general, mai puține informații ca cea originală și nu există nici o garanție că informația pierdută e nesemnificativă. In continuare însă ne vom baza pe faptul că reducerea unei imagini la un contur

este utilă în circumstanțe date și ne vom îndrepta atenția asupra metodelor de extragere a conturilor. Existența unui contur într-o imagine poate fi provocată de tranziția bruscă din regiuni întunecate spre regiuni luminoase. Altfel spus conturile reprezentând margini ale obiectelor sînt prin definiție tranziții de lumină între două regiuni. În termenii funcției imagine $g(x,y)$, o margine este o regiune a planului X-Y unde $g(x,y)$ are un gradient de o amplitudine mare. Deci problema determinării conturului unui obiect dintr-o imagine este echivalentă cu estimarea amplitudinii gradientului unei funcții.

5.3.3.1. Operatorul Roberts

Gradientul poate fi calculat dacă se cunosc derivatele după direcție ale unei funcții luînd ca direcții cele două axe perpendiculare ale planului cartezian. Prin urmare este necesar să se aleagă o pereche de direcții ortogonale și o aproximație a derivatei în vederea scrierii algoritmului de extragere a conturului folosind gradientul. Formula este :

$$\|g(i,j)\| \approx F(i,j) = \sqrt{[g(i,j) - g(i+1,j+1)]^2 + [g(i,j+1) - g(i+1,j)]^2} \quad (5.8)$$

Operatorul $F(i,j)$ se numește operatorul cruce al lui Roberts. El poate fi scris simplificat pentru eficiența calculului utilizînd amplitudini în valoare absolută în loc de radical și pătrați:

$$F(i,j) = |g(i,j) - g(i+1,j+1)| + |g(i,j+1) - g(i+1,j)| \quad (5.9)$$

În fig. 5.7.b se arată rezultatul aplicării operatorului F asupra imaginii numerice din 5.7.a afișîndu-se doar acele puncte (i,j) pentru care $F(i,j) > p$, unde p este un prag.

5.3.3.2. Operatorul Sobel

Examinînd mai îndeaproape problema obținerii imaginii gradient se poate spune că cea mai mare dificultate întîmpinată este existența zgomotului. Funcția de imagine $g(x,y)$ poate fi scrisă ca suma a două funcții: o imagine "ideală" sau imaginea semnal $S(i,j)$ și una de zgomot $Z(i,j)$. Problema este determinarea gradientului funcției S și nu a funcției g .

I. Sobel a propus un gradient care îmbină două tipuri de

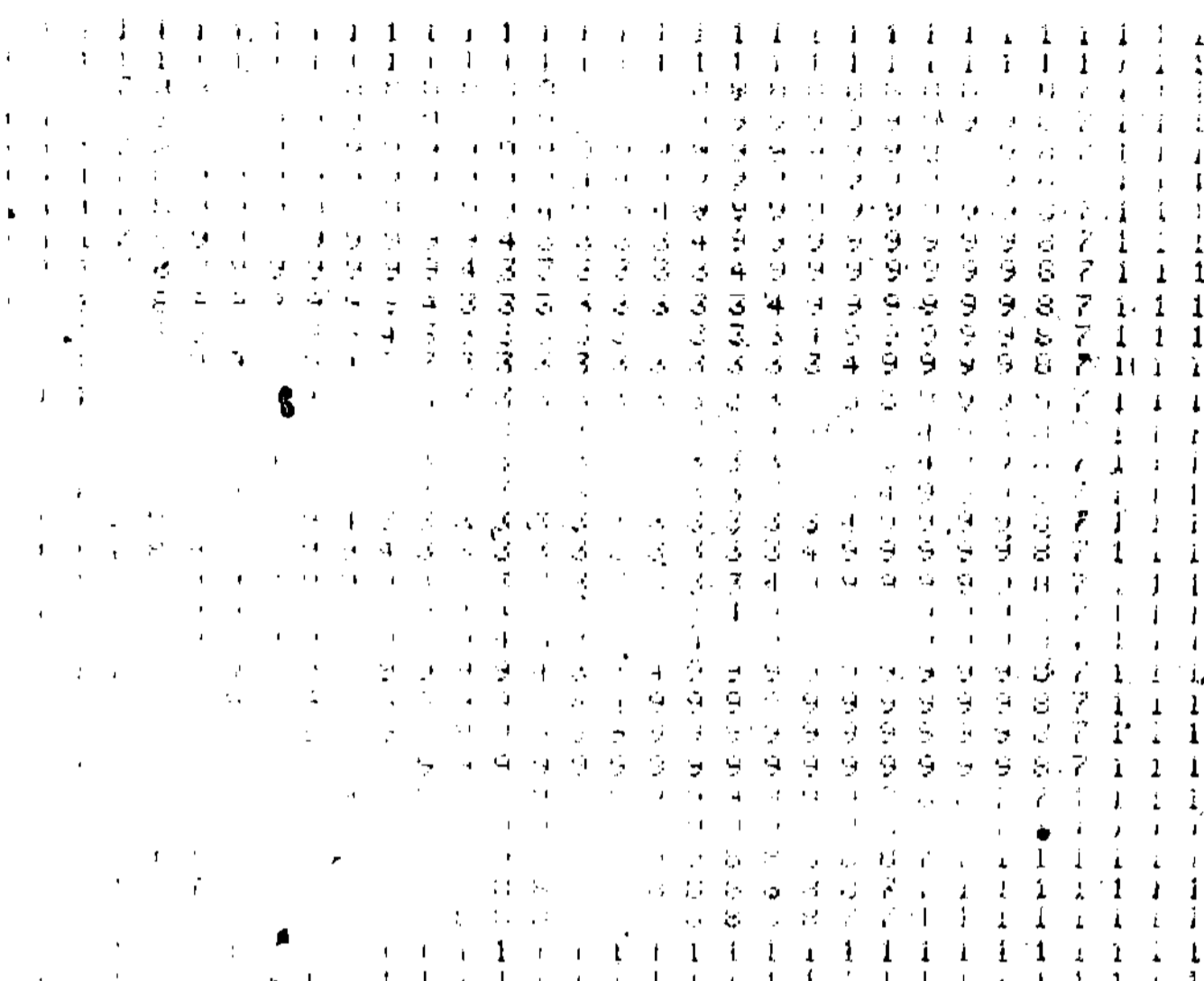


Fig 5.7a

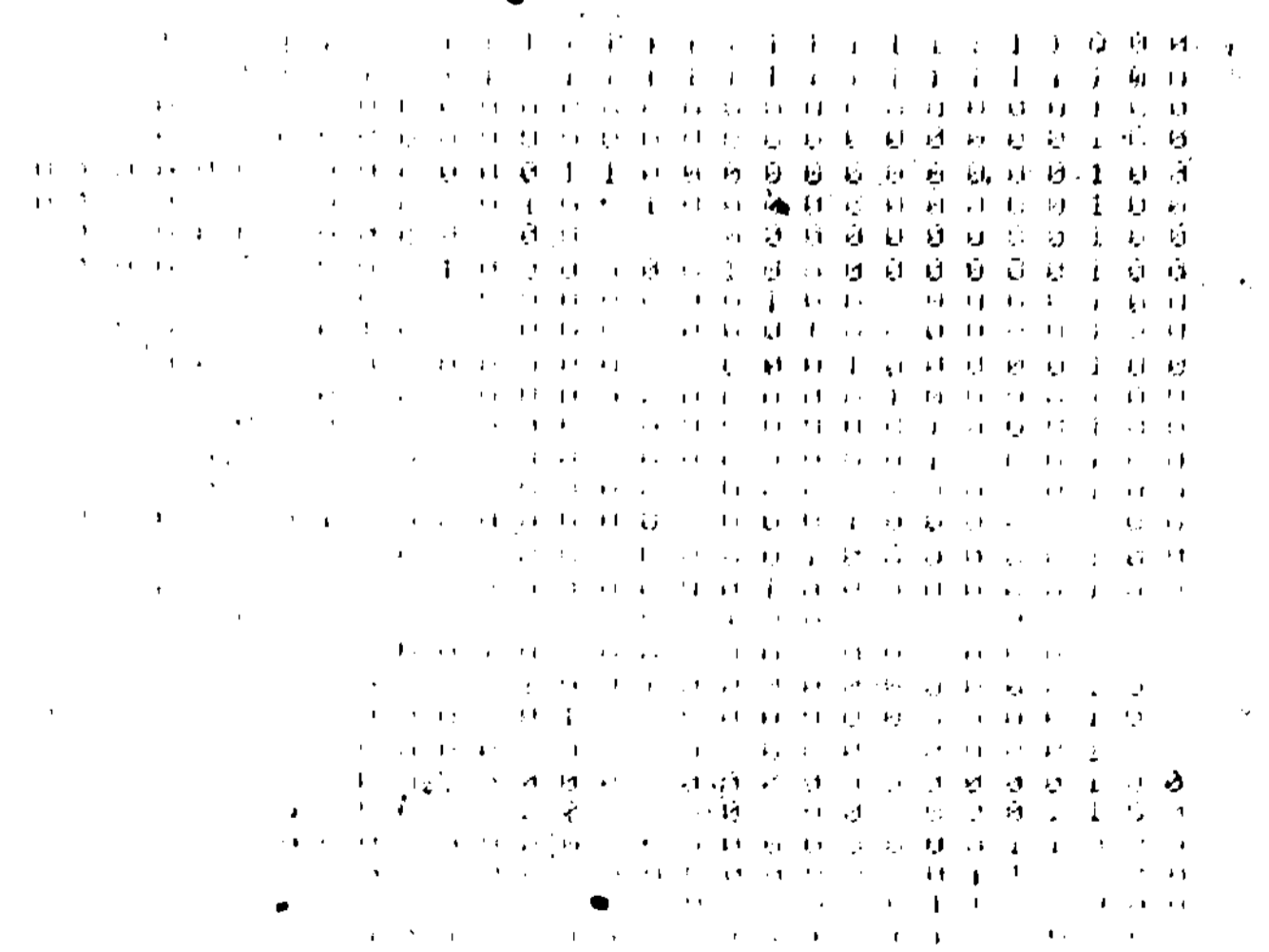


Fig 5.7.b

proceselor de diferențiere și de mediere. Un exemplu de operator de tip gradient ce lucrează cu ferestre de 3x3 (fig.5.8) conține două tipuri de diferențe:

$$S_x = (c + 2f + i) - (a + 2d + g) \quad (5.10)$$

și

$$S_y = (g + 2h + i) - (a + 2b + c) \quad (5.11)$$

Se definește gradientul în punctul 'e' prin formula:

$$S = \sqrt{S_x^2 + S_y^2} \text{ sau prin aproximare}$$

pentru eficiența calculului :

$$S = |S_x| + |S_y| \quad (5.11)$$

a	b	c
d	e	f
g	h	i

Modul în care este definit S_x este o aproximare a derivatei funcției imagine după direcția x. Mai întâi se formează două medii ponderate pentru a estima funcția imagine în punctele f și d. Cele două medii ponderate sînt apoi scăzute obținându-se aproximarea derivatei. (analog se obține și S_y). În fig. 5.9.a este prezentată imaginea inițială iar în fig.5.9.b contururile obținute ca urmare a aplicării operatorului Sobel. Pentru altă imagine contururile apar în fig.5.10.

Fig.5.8

5.3.3.3. Operatorul Marr-Hildreth /MH 81/

Bazați pe cercetările anterioare ale lui Marr /Ma 81/ privind fiziologia vederii, Marr și Hildreth propun următoarea secvență de pași pentru extragerea conturului: netezirea imaginii, derivarea ei și eșantionarea vîrfurilor (dacă se utilizează derivata de ordinul I) sau a treptărilor prin 0 (dacă se utilizează derivata de ordinul II).

Se presupune că s-a netezit imaginea printr-un proces de convoluție notînd operația cu $G \cdot I$, unde G este operatorul de convoluție iar I imaginea inițială. Derivăm imaginea obținută cu ajutorul operatorului lui Laplace :

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad (5.12)$$

Laplacianul furnizează o derivată de ordinul II neorientată. Noua imagine poate fi scrisă sub forma $\nabla^2(G \cdot I)$ și folosind o proprietate a convoluției :

$$\nabla^2(G \cdot I) = (\nabla^2 G) \cdot I \quad (5.13)$$

Se pot reduce cei doi operatori (' ∇^2 ' și ' \cdot ') la unul singur de forma :

$$\nabla^2 G = (2 - \frac{r^2}{\sigma^2}) e^{-\frac{r^2}{2\sigma^2}} \quad (5.14)$$

Procesul $\nabla^2 G \cdot I$ e un proces apropiat de cel ce se petrece în retină susțin autorii. Operatorul $\nabla^2 G$ constituie astfel un detector de contur justificat matematic și care are și o consistență bază biologică.

Felul cum acționează operatorul este clar. Partea G realizează netezirea imaginii la o anumită scară, iar partea ∇^2 realizează apoi o derivată de ordinul II. Pasul următor este determinarea trecerilor prin 0 care reprezintă margini în imagine. Avantajele cele mai importante ale operatorului $\nabla^2 G$ sînt: a) este localizat în spațiu și frecvență; b) este neorientat; c) reduce diferite efecte ale zgomotului în funcția de scară. Factorii de scară spațiali ai operatorului determină rezoluția capacității sale de netezire. (Cel mai mic operator utilizat a avut factorul de scară $\sigma=0,6$). Rezultatul aplicării operatorului Marr-Hildreth se poate urmări în fig.5.11 și 5.12.

5.3.3.4. Operatorul Neckel /Hu 69/

Acest operator descris de Neckel, găsește o margine într-o vecinătate circulară a unui pixel. În fig.5.13 se prezintă o astfel de vecinătate circulară împreună cu aproximarea ei prin pixeli. Se consideră că în această vecinătate circulară

[Faint, illegible text covering most of the page, possibly bleed-through from the reverse side or very low-contrast scanning. Some faint lines of text are visible but cannot be transcribed accurately.]

лр 59.6

The following text is a series of vertical bars of varying heights and widths, arranged in a regular grid pattern. The bars are organized into several distinct sections, each marked with a large letter:

- Section 1:** A grid of vertical bars at the top of the page.
- Section 2:** A grid of vertical bars, starting with a large letter **B** on the left side.
- Section 3:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 4:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 5:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 6:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 7:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 8:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 9:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 10:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 11:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 12:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 13:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 14:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 15:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 16:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 17:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 18:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 19:** A grid of vertical bars, starting with a large letter **A** on the left side.
- Section 20:** A grid of vertical bars, starting with a large letter **A** on the left side.

Fig 51a

The following text is a series of repeated characters and symbols, likely representing a corrupted or low-quality scan of a document. The characters are mostly small, repetitive marks and symbols, including:

• Repeated sequences of small, illegible characters and symbols, possibly representing a barcode or a specific data format that has been lost or corrupted.
 • Some recognizable symbols like asterisks (*), dots (.), and vertical bars (|) interspersed among the noise.
 • The text is organized into several distinct blocks, suggesting it might have been part of a structured document like a table or a list.

Due to the extreme low quality and repetition of the characters, the original content is completely unreadable.

Fig 5.11

The following table shows the results of the analysis of variance for the effect of the concentration of the solution on the rate of reaction. The data are as follows:

Concentration (M)	Rate of Reaction (mol/L.s)
0.1	0.001
0.2	0.002
0.3	0.003
0.4	0.004
0.5	0.005
0.6	0.006
0.7	0.007
0.8	0.008
0.9	0.009
1.0	0.010

From the above data, it can be seen that the rate of reaction increases linearly with the concentration of the solution. This indicates that the reaction is first order with respect to the concentration of the solution.

The following graph shows the relationship between the rate of reaction and the concentration of the solution. The data points are plotted on a graph and a straight line is drawn through them. The equation of the line is $y = 0.01x$, where y is the rate of reaction and x is the concentration of the solution.

The following table shows the results of the analysis of variance for the effect of the temperature on the rate of reaction. The data are as follows:

Temperature (°C)	Rate of Reaction (mol/L.s)
20	0.001
30	0.002
40	0.004
50	0.008
60	0.016
70	0.032
80	0.064
90	0.128
100	0.256

From the above data, it can be seen that the rate of reaction increases exponentially with the temperature. This indicates that the reaction is highly sensitive to temperature changes.

The following graph shows the relationship between the rate of reaction and the temperature. The data points are plotted on a graph and a curve is drawn through them. The equation of the curve is $y = 0.001 \cdot 2^{(x-20)/10}$, where y is the rate of reaction and x is the temperature.

Fig 5.12.

este prezentă o margine dată de funcția:

$$g(x,y) = e \text{ dacă } ax+by \leq e$$

$$g(x,y) = e+d \text{ dacă } ax+by > e$$

$$\text{unde } a^2 + b^2 = 1$$

Croaie medie pătratică a într-o funcția margine $g(x,y)$ și valoarea de cenugiu a pixelilor este minimizată pe întreaga suprafață circulară N_p :

$$S = \sum_{k,l \in N_p} [f(k,l) - g(k,l)]^2 \quad (5.16)$$

În vederea acestei minimizări valorile de cenugiu sînt proiectate pe o bază ortogonală aleasă pentru vecinătatea circulară. Funcțiile ortogonale H_i se împart în produse de tip funcții radiale sau unghiulare adică o analiză Fourier în coordonate polare. Pentru a reduce zgomotul de frecvență înaltă se utilizează în această dezvoltare doar șapte funcții de bază H_i astfel că:

$$\alpha_i = \sum_{k,l \in N_p} H_i(k,l) f(k,l) \quad i=1, \dots, 7 \quad (5.17)$$

Huackel oferă o soluție de minimizare a lui 5.16 bazată

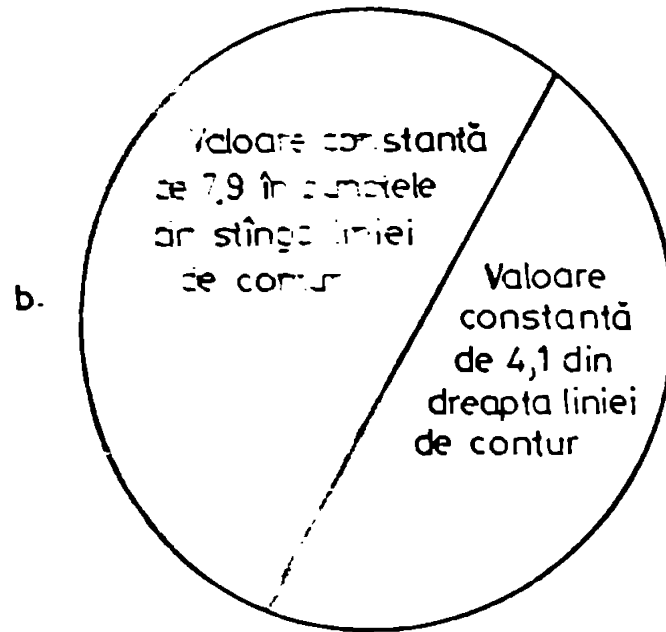
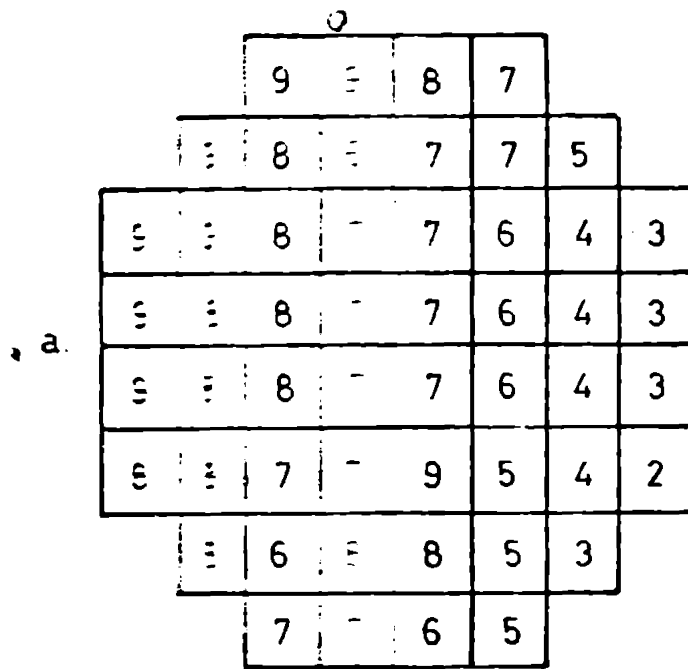


Fig.5.13

pe cei șapte coeficienți a_1 din 5.17. Calculul acestor parametri ai soluției: a, b, c, d, e este complex și nu va mai fi prezentat. Pentru exemplul dat de Hueckel din fig.5.13 s-au alcătuit algoritmul și programele corespunzătoare. Se observă că s-au obținut aceleași valori pentru e și d, adică $d=3,8$ și $e = 4,08$ deci $e+d = 7,88$ (fig.5.14). Rezultatele aplicării acestui operator pe imagini întregi este arătat în figurile 5.15.a și b și 5.16.a și b.

Programul durează 17-20 minute pe calculatorul ROBOTRON 4201 în funcție de complexitatea imaginii. Performanțele sale de detecție a conturului sînt bune dar timpul de calcul este mare pe acest tip de calculator. Pe alte calculatoare (IBM 370) s-a raportat un timp de execuție de 19-30 secunde.

În concluzie, programul are performanțe bune în ceea ce privește precizia de detecție a conturului exceptînd factorul timp ce poate fi înlăturat sau prin utilizarea unui calculator specializat pe calcule în timp real (unitate de virgulă flotantă) sau o structură hard dedicată.

5.3.3.5. Operatorul Rosenfeld

Acest tip de operator este bazat pe calculul diferențelor locale. Se calculează ponderi pentru fiecare pixel și se elimină ponderile care depășesc o valoare minimă de prag. (efect de subțiere). Programul elaborat, intitulat DIFF calculează mai întîi valoarea medie a nivelelor de cenușiu pentru fiecare pixel din centrul unei ferestre pătratice. Mărimile ferestrelor sînt de forma $2^k \times 2^k$, unde k poate lua valori întregi de la 1 la 6. Programul calculează apoi valoarea absolută a diferențelor valorilor medii pentru fiecare pereche de ferestre adiacente ce nu se suprapun una peste cealaltă și a căror orientare e perpendiculară pe direcția conturului, asociînd această valoare unui punct situat la mijlocul acestor ferestre. Se calculează ponderile de contur pentru 4 orientări: orizontal, vertical și două diagonale. Fiecare punct al imaginii va avea asociată o pondere în afară de punctele de pe contur. Apoi programul suprimă toate punctele pentru care există pondere mai mare într-o vecinătate de rază 2^{k-1} perpendiculară pe o direcție de detecție a conturului considerată. În testarea acestei metode au fost utilizate ferestre de $4 \times 4(k-2)$ și numai orientarea

111100

410 1047

e d

4400
3100

1111
1111

1111

494
300
400

1111

1111

1111

1111

Fig. 5.14

Handwritten text, possibly a list or notes, consisting of numerous lines of illegible characters and symbols. The text is densely packed and appears to be a collection of data points or a series of entries.

Fig. 5.15a

The text on this page is extremely faint and illegible. It appears to be a technical document or a list of items, but the specific content cannot be discerned.

Fig 5.15 b

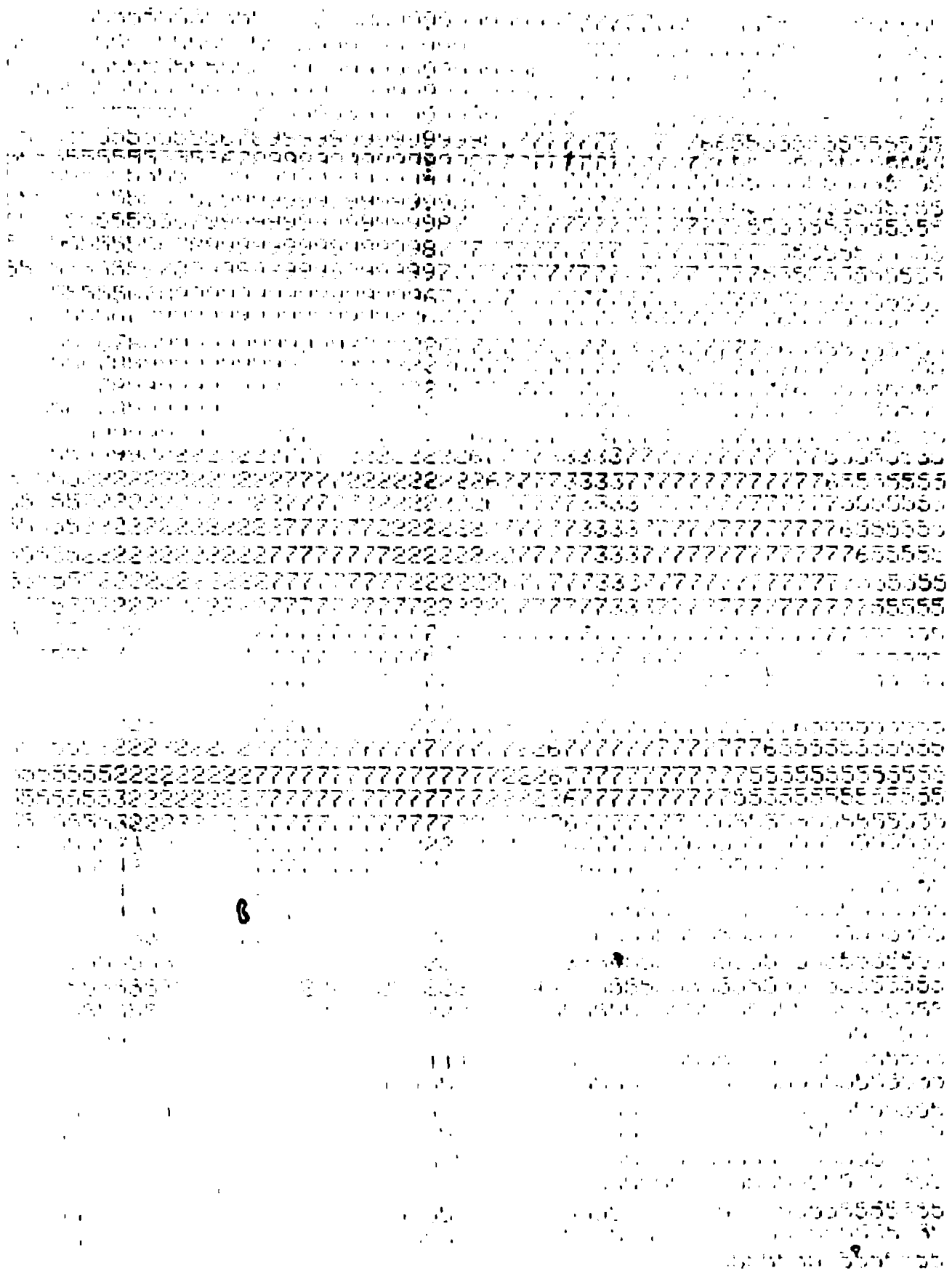


Fig 5.16a

The following table shows the results of the analysis of variance for the effect of the concentration of the solution on the rate of reaction. The data are presented in the form of a table with columns for the concentration of the solution and the corresponding rate of reaction. The table is as follows:

Concentration of the solution	Rate of reaction
0.1 M	0.01
0.2 M	0.02
0.3 M	0.03
0.4 M	0.04
0.5 M	0.05
0.6 M	0.06
0.7 M	0.07
0.8 M	0.08
0.9 M	0.09
1.0 M	0.10

The results of the analysis of variance show that there is a significant difference in the rate of reaction between the different concentrations of the solution. The rate of reaction increases as the concentration of the solution increases. This is expected because a higher concentration of the solution means there are more reactant molecules available to undergo the reaction, leading to a faster rate of reaction.

The following table shows the results of the analysis of variance for the effect of the temperature on the rate of reaction. The data are presented in the form of a table with columns for the temperature and the corresponding rate of reaction. The table is as follows:

Temperature	Rate of reaction
20°C	0.01
30°C	0.02
40°C	0.04
50°C	0.08
60°C	0.16
70°C	0.32
80°C	0.64
90°C	1.28
100°C	2.56

The results of the analysis of variance show that there is a significant difference in the rate of reaction between the different temperatures. The rate of reaction increases as the temperature increases. This is expected because a higher temperature means the reactant molecules have more kinetic energy, leading to a higher frequency of collisions and a faster rate of reaction.

The following table shows the results of the analysis of variance for the effect of the catalyst on the rate of reaction. The data are presented in the form of a table with columns for the presence of the catalyst and the corresponding rate of reaction. The table is as follows:

Presence of the catalyst	Rate of reaction
Without catalyst	0.01
With catalyst	0.02

The results of the analysis of variance show that there is a significant difference in the rate of reaction between the presence and absence of the catalyst. The rate of reaction is significantly higher when the catalyst is present. This is expected because a catalyst provides an alternative reaction pathway with a lower activation energy, leading to a faster rate of reaction.

Fig 5.16 b

verticală a fost investigată.

5.3.3.5. Operatorul Mac Leod

Această metodă folosește o mască gaussiană detectoare de contur. Ea constă în calcularea unor ponderi pentru fiecare punct al imaginii prin înmulțirea nivelului de cenușiu al fiecărui pixel dintr-o vecinătate considerată cu valoarea corespunzătoare a unui punct dintr-o mască și însumarea tuturor produselor de această formă obținute. Masca e dată de relația:

$$w(x,y) = \exp\left[-\left(\frac{y}{t}\right)^2\right] \left\{ \exp\left(-\left(\frac{x-p}{p}\right)^2\right) + \exp\left(-\left(\frac{x+p}{p}\right)^2\right) \right\} \quad (5.18)$$

unde x este componenta distanței de la un pixel din fereastră la punctul de contur pe direcție perpendiculară pe contur iar y este componenta paralelă a acestei distanțe. Acest operator are proprietatea că în cazul în care masca e centrată pe un element de imagine ce conține un contur, punctele de contur vor fi cel mai puternic ponderate. Valoarea lui x , pentru fiecare pixel, a fost dată prin componenta orizontală a distanței de la punctul din centrul măștii și valoarea lui y a fost dată prin componenta verticală a acestei distanțe. Punctele măștii au fost obținute cu valorile $p=t=4$ în cadrul unei ferestre de 7×7 .

5.3.3.7. Concluzii rezultate din compararea performanțelor diferiților operatori prezentați

Deoarece au fost studiați un număr considerabil de operatori de detecție a conturilor este necesară o comparare a acestora pentru a stabili limitele și performanțele acestora. Caracteristicile operatorilor de detecție a conturilor care au fost luate în considerare sînt următoarele :

- capacitatea de detecție a marginilor (linii de contur) de diferite orientări
- capacitatea de detecție a marginilor în prezența unghiului
- gradul de localizare a marginilor
- abilitatea de detecție a marginilor estompate (difuze)
- abilitatea de detecție a marginilor curbate
- abilitatea de detecție a marginilor în prezența altor margini
- viteza de calcul și necesarul de memorie.

În principal au fost supuși comparării operatorii de detecție a conturilor ai lui Hueckel, Mac Leod, Rosenfeld,

și Marr-Hildreth. S-au desprins următoarele constatări:

- operatorul Hueckel se comportă slab la zgomot din cauza caracterului său local. Totuși, pentru contraste puternice prin alegerea corespunzătoare a lui CONF și DIF (care decifrează dacă un domeniu circular analizat conține un element de contur sau nu) se pot obține rezultate satisfăcătoare.
- în prezența zgomotului algoritmi MacLeod și Rosenfeld se comportă destul de bine. În schimb ei nu pot scoate o mulțime de orientări ale conturului, așa cum poate operatorul Hueckel, datorită caracterului lor global. Performanțele lor sînt cele mai bune pentru orientare optimă după o direcție a conturului și nu pot extrage contururile perpendiculare pe această direcție. Performanțele lor pot fi îmbunătățite printr-un procedeu de afișare cu mărirea contrastului final pe baza translatații valorii nivelurilor de cenușiu. La operatorul Hueckel nu are sens o asemenea operație.
- cele mai bune performanțe le are operatorul Marr-Hildreth care are următoarele proprietăți: 1) este localizat în spațiu și frecvență; 2) posedă dimensiuni adecvate pentru a reduce efectele zgomotului și 3) este independent de direcția marginii. Factorii săi de scală determină atât regiunea în care se pun în evidență marginile cât și rezoluția sau "răspunsul" filtrului de netezire. Mărirea operatorului este importantă de asemenea în ce privește sensibilitatea la zgomot. Cel mai mic operator utilizat, avînd o scară spațială de $\sqrt{3}$ -8 pixeli, a fost capabil să răspundă la un semnal treaptă a intensității chiar pentru un raport semnal-zgomot egal cu una.

O prezentare calitativă a performanțelor celor șase operatori studiați este făcută în Tab.5.2.

În ce privește timpul de calcul Hueckel a sugerat o metodă de măsurare a acestuia: viteza operatorului este numărul de operații aritmetice necesare pe numărul de elemente de imagine. Prin operații aritmetice se înțeleg operațiile de tip +, -, \times , /, incrementările, decrementările și calculul adresei. Instrucțiunile rapide ca transferul de cuvînt în memorie sau complementări, etc. nu contează. O extragere de radical contează cît sece operații. Amploarea operatorului Hueckel este dată de cele 55 de operații pe pixel.

Ultima coloană din tab.5.2 prezintă numărul de operații

Tab.5.2

OPERA- TORI	CARACTE- RISTICI DOBITE	cap. detec- ție în prezența ogometului	capaci- tatea de de- tecție sub di- ferite orien- tări	capaci- tatea de de- tecție a margi- nilor curbate	capaci- tatea de de- tecție a margi- nilor în prezența altor marginii	Număr operații pe pixel
HUECKEL		slabă	foarte bună	slabă	bună	55
ROSENFELD		bună	slabă	slabă	bună	6
MAC LEOD		bună	slabă	slabă	bună	24
SOBEL		bună	slabă	slabă	bună	17
ROBERTS		slabă	slabă	slabă	slabă	5
MARR-		foarte	foarte	foarte	foarte	6
HILDRETH		bună	bună	bună	bună	

per pixel ale fiecărui operator. Se remarcă că cel mai lent este operatorul Hueckel, cel mai rapid fiind operatorii Rosenfeld, Roberts și Marr-Hildreth. Aceste din urmă are și performanțele calitative cele mai bune.

5.3.4. Sistem de vedere pentru identificarea unor piese mecanice de dimensiuni mici

În cadrul acestui paragraf se va descrie conceperea unui sistem de vedere tip 2D (două dimensiuni) pentru imagini binare destinat să funcționeze în condiții industriale în scopul identificării și localizării unor piese mecanice mici.

La elaborarea sistemului s-au avut în vedere următoarele condiții de operare :

- în câmpul vizual apar mai multe piese decodată,
- nu există suprapuneri de obiecte în modul normal de operare,
- dacă apar suprapuneri piesele nu vor fi clasificate.

Caracteristicile principale ale sistemului au fost fixate astfel:

- 1) identificare și localizare on-line a pieselor nesuprapuse
- 2) algoritmi de prelucrare rapizi, robusti și ieftini

- 3) o frecvență de erori de clasificare foarte scăzută
- 4) o procedură de învățare a obiectelor de clasificat.

La elaborarea sistemului s-au luat în considerare cele mai avansate metode și tehnici de prelucrare, extragere de caracteristici, recunoaștere și învățare existente la nivelul anului 1983.

În general sistemul are ca principale linii directoare, folosite pe larg în prezent de majoritatea cercetătorilor, cele elaborate la Institutul de Cercetări Stanford prezentate în lucrarea lui Agin și Luda /AD 75/. La aceste metode se adaugă unele rezultate mai noi obținute de alți cercetători între care se remarcă cele din cadrul Programului de Cercetare Robotică de la Laboratorul de propulsie reactivă N.A.S.A. din Pasadena /C. S1/ precum și câteva contribuții personale.

5.3.4.1. Obținerea imaginilor binare și izolarea componentelor

Imaginile binare pot fi obținute printr-un proces de comparare cu un prag a valorilor pixelilor unei imagini în urma căruia rezultă doar două "culori", alb și negru, adică niște siluete. Există două tipuri de procedee de realizare a acestei conversii : prin interfața de conversie analog-numerică pentru semnalul TV prevăzută cu prag de detecție (eventual două) sau prin program (software). O imagine binară are numai două tipuri de pixeli codificați 1 (negru) și 0 (alb).

Se poate folosi un prag global care trebuiește determinat în prealabil prin mijloace software folosind histograme ale imaginii. O alternativă este cea a folosirii de praguri locale obținute prin examinarea pixelilor dintr-o vecinătate conținută de o porțiune de imagine numită fereastră. În legătură cu tehnicile de alegere a pragurilor, Weska /We 76/ prezintă o perspectivă completă asupra acestora.

Soluția generală adoptată este aplicarea conectivității 4 la o culoare și a conectivității 8 la cealaltă culoare. Acest lucru complică algoritmi de izolare a componentelor într-o oarecare măsură.

O altă soluție este adaptarea convenției conectivității 6 care tratează pixelii albi și negri în mod egal. Din referi-

cire nici acest tip de conectivitate nu constituie o soluție perfectă. Pentru figurile 5.17a și b odată pătratul central este parte a fondului iar pentru cealaltă imagine pătratul nu este conectat de fond.

Alegerea tipului de conectivitate rămâne o decizie arbitrară. Pentru algoritmi ce s-au elaborat în prezentul sistem de vedere în interesul simplificării acestora s-a adoptat conectivitatea 6. Între sistemele de vedere elaborate până în prezent unul singur, CONSIGHT -/WF 79/ , folosește acest tip de conectivitate.

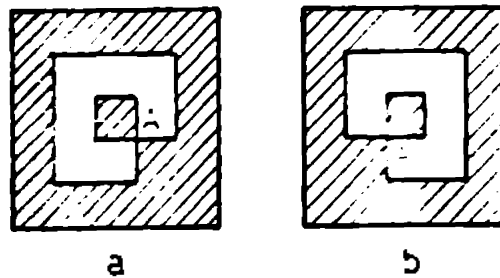


Fig.5.17.a.b

5.3.4.2. Algoritmul de urmărire a conturului (u.c)

Algoritmul UC începe explorarea imaginii din colțul stâng sus pe linii succesive. O nouă componentă a imaginii este detectată ori de câte ori apare o tranziție de culoare ($0 \rightarrow 1$ sau $1 \rightarrow 0$) între pixelii $(I-1, J)$ și (I, J) . Dacă se întâmplă acest lucru se suspendă explorarea și se trece la faza de urmărire a conturului în cursul căreia se elaborează descrierea completă a componentei prin intermediul calculului unor sume care în final dau valorile caracteristicilor metrice, de formă și structurale.

Evident, conturul fiecărei componente va fi întâlnit de mai multe ori în cursul explorării, dar urmărirea conturului se execută o singură dată pentru fiecare componentă deoarece se ține o socoteală a tuturor tranzițiilor în cadrul unei liste.

Considerând că avem deja la dispoziție o imagine binară sub forma unei matrici bidimensionale este important modul de izolare a componentelor din imagine. Din punct de vedere formal,

o componentă este constituită de un grup de pixeli conecși. Pixelii adiacenți se consideră conecși dacă au aceeași culoare.

Există în literatură două metode de izolare a componentelor unei imagini: analiza conectivității (AC) și urmărirea conturului (UC).

Așa cum sugerează și numele, prima metodă este un algoritm care examinează conectivitatea locală în jurul fiecărui pixel al imaginii construind o descriere a componentelor unei imagini pixel cu pixel. A doua metodă determină conectivitatea pixelilor urmărind conturul unei componente în vederea evaluării complete a extinderii acesteia (AC).

Analiza conectivității prezintă avantajul prelucrării imaginii printr-o singură explorare a rastrului în timp ce urmărirea conturului (UC) alternează între o explorare a rastrului pentru căutarea de noi componente și urmărirea conturului ceea ce conduce la un acces aleator la elementele imaginii.

Totuși, algoritmul UC are o trăsătură importantă care lipsește algoritmului AC - aceea de a produce o descriere de tip cod în lanț a fiecărei componente un rezultat esențial care pune în evidență forma componentei. Iată de ce s-a manifestat în cele din urmă o preferință pentru algoritmul UC.

Înainte de a prezenta algoritmul UC este necesar să se precizeze tipul de conectivitate care se folosește. În literatură predomină aproape exclusiv conectivitatea 4 și conectivitatea 8. Prima ia în considerare vecinii adiacenți de sus, jos, dreapta și stînga unui pixel în timp ce a doua include în plus și pixelii de pe diagonale.

Din păcate utilizarea unui singur tip de conectivitate fie 8 fie 4 conduce la paradoxuri în sensul că unei componente de tip gaură nu i se poate lămurii apartenența în mod clar la fondul imaginii (fig.5.17).

Există trei elemente distincte în algoritmul UC. Primul îl constituie însăși procedura de urmărire a conturului. Al doilea este procedura de calcul a sumelor statistice în cursul urmării conturului. Al treilea este procedura de actualizare a listei tranșițiilor. Aceste proceduri vor fi prezentate pe rînd.

Procedura de urmărire a conturului

Procedura UC tratează fiecare pixel ca pe un pătrat cu

latura o unitate, mărginit de patru laturi. Un pixel de contur este acela care are una sau mai multe laturi comune cu un pixel aparținând fondului. (Termenul fond este aici utilizat în sens general și puțină fi și o componentă). Laturile sînt numite în acest caz segmente de contur. Conturul unei componente nu este altceva decît o secvență de segmente de contur cu direcții determinate. Tehnica codului în lanț notează aceste direcții cu 0, 1, 2, 3 ca în fig.5.18.

Se poate lua ca alternativă și codificarea în lanț după cele 8 direcții 0 la 7 considerînd și pixelii adiacenți după diagonale. Se poate demonstra că deși sumele statistice au valori diferite pentru cele două codificări ele rîmîn constante în cadrul unei codificări

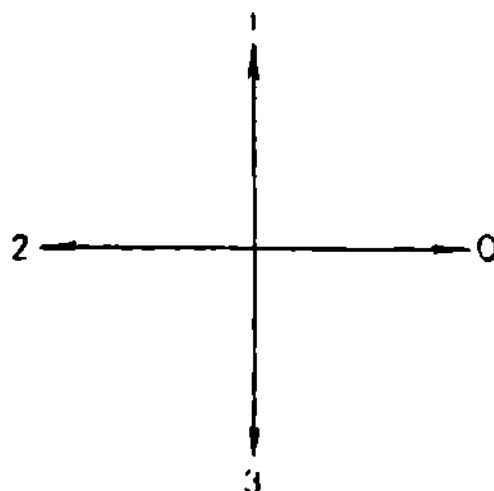


Fig.5.18

ceea ce nu împiedică în schimb codificarea după 2 direcții introduce în anumite cazuri particulare necesitatea revenirii pe un contur în sens contrar parcurgerii inițiale. Codificarea în lanț cu 4 direcții evită acest lucru.

Procedura UC funcționează după cum urmează. Se presupune că $0-0$ înțilnit pixelul unei noi componente fie acesta (IJ). El devine primul pixel al conturului și este memorat sub forma variabililor (I_0, J_0). Codul în lanț este inițializat spre "dreapta" (0), ceea ce corespunde segmentului de contur de sus al pixelului inițial. Sensul de parcurgere este cel orar față de interiorul componentei. Pentru fiecare pas procedura decide să se îndrepte la stînga, la dreapta sau înainte. Acest lucru se decide funcție de valorile pixelilor "a" și "b" din fig.5.19.

Aplicînd în acest moment convenția conectivității 6 regulile regulile de decizie pentru segmentele verticale sau orizontale. În anumite cazurile, dacă pixelul "a" este alb și "b" e negru, conturul continuă în aceeași direcție. Diferențele apar la in-

toarcerile spre stînga sau dreapta. Dacă segmentul curent este orizontal (spre stînga sau dreapta) pixelul "a" nu este conex cu pixelul curent. Prin urmare conturul întoarce spre stînga numai dacă amîndoi pixelii "a" și "b" sînt negri. Dacă pixelul "b" este alb, conturul întoarce spre dreapta indiferent de culoarea lui "a".

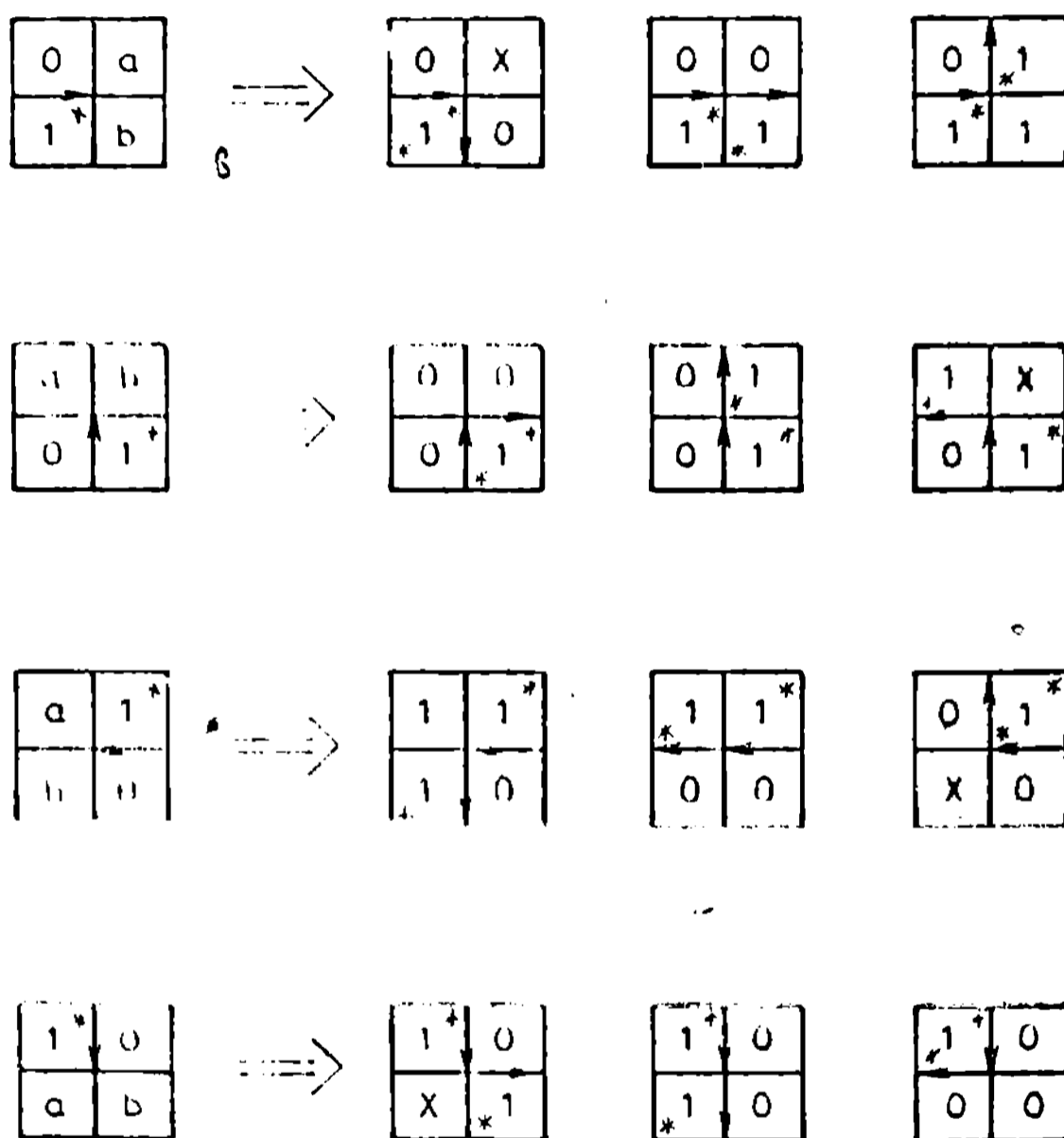


Fig.5.19

Dacă segmentul curent este vertical, pixelul "a" este conex cu pixelul curent prin urmare conturul întoarce la stînga numai dacă pixelul "a" este negru indiferent de culoarea pixelului "b". Conturul se întoarce spre dreapta numai dacă amîndoi pixelii "a" și "b" sînt albi.

Regulile de traversare a conturului unei componente albe sînt identice inversînd rolurile între alb și negru.

Procedura de calcul a sumelor statistice

Procedura pornește de la coordonatele pixelului curent și codul segmentului de centur curent. Analizând vecinătatea de 2×2 pixeli stabilește conform procedurii precedente poziția următorului punct de pe centur și următorul vector al codului în lanț. În același timp actualizează sumele statistice $P(1)$ la $P(7)$ care vor folosi la calculul momentelor I , I^2 , IJ , J , J^2 și arie.

Pentru explicarea cumularii sumelor $P(i)$ se dă exemplul din fig.5.20.

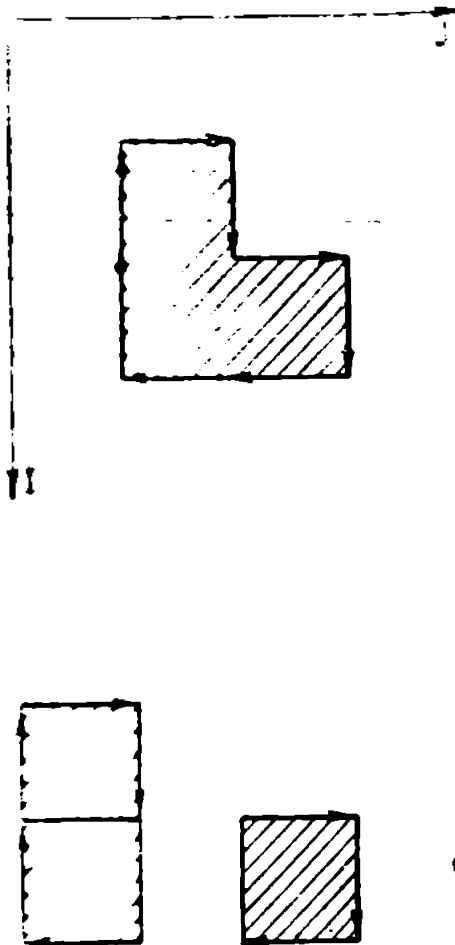


Fig.5.20

Avem codul în lanț 01032211. Codul e înconștient e margine de sus a unui centur. Sub el există o coloană care e marginită de un cod 2. Se consideră componenta împărțită în figuri verticale. Dacă m e coordonata I de sus a figurai și $k+m$ cea de jos atunci contribuția figurai la valoarea ariei este de $(k+m) \cdot m \cdot k$. Ținând cont de sensul de parcurgere rezultă că pentru calculul

axiei e suficient să se cumuleze cu - coordonata I a fiecărui cod e și cu + cea a fiecărui cod 2. Fiecare fișă contribuie la calculul sumei I cu

$$m+m+1+m+2+\dots+m+k-1 = m.k + (1+2+\dots+k-1) = m.k + k.k/2 \quad (5.1)$$

Dar ținând cont de egalitatea

$$(m+k)(m+k) - m.m = 2mk + k.k \quad (5.2)$$

rezultă că dacă se cumulează cu - pătratul coordonatei I a fiecărui cod 2 se obține o sumă de două ori mai mare decât I.

Contribuția fișei la calculul sumei I^2 este

$$\begin{aligned} m+(m+1)+ \dots +(m+k-1) &= \\ &= k.m + 2.m(1+2+\dots+k-1) + 1+2+\dots+(k-1) = \\ &= k.m + 2.m.k/2 + (k-1)k(2k-1)/6 = \\ &= m.k + k.m + k/3 - k(k/2 - 6) \end{aligned} \quad (5.3)$$

$$\text{Dar } (m+k)^3 - m^3 = 3mk + 3km + k \quad (5.4)$$

În relația 5.3 termenul $k(k/2-6)$ e neglijabil față de ceilalți. Pentru $k=10$ și $m=10$ eroarea ce se obține e de 10 la 2300. În cazurile practice eroarea aceasta e mult neglijabilă față de alte surse de erori. Rezultă că dacă se cumulează cu - cubul coordonatei I a fiecărui cod e și cu + cubul coordonatei I a fiecărui cod 2 se obține o sumă de 3 ori mai mare decât I^2 .

Analog prin împărțirea regiunii în fișe orizontale se deduc formulele pentru calculul sumelor J și J^2 .

În privința sumei IJ contribuția fișei orizontale de coordonată I e egală cu y și având începutul coordonatei J la m și sfârșitul la m+k se obține :

$$\begin{aligned} y(m+m+1+\dots+m+k-1) &= y(mk + k/2) = \\ &= y(m+k) - y/2 \end{aligned} \quad (5.5)$$

cumulând cu - termenii ym la fiecare cod 1 și cu + pe cei de la codul 2 se obține în final o sumă de două ori mai mare decât IJ . Mai apare în cazul de față încă o corecție la acest rezultat datorată inițializării coordonatelor primului segment de contur. Nu se pot folosi coordonatele pixelilor în cadrul acestor formule deoarece segmentele de contur nu trec prin centrul pixelilor. Aceasta conduce la calculul corect al coordonatelor unor colțuri de pixel cu formulele:

$$x = I \pm 0,5 \text{ și } y = J \pm 0,5.$$

Pentru primul pixel de contur de coordonate I, J se fac inițializările :

$$X = I - 0,5$$

$$\begin{aligned} X &= J - 0,5 \\ Y &= J - 0,5 \\ P(1) &= 0 \\ P(2) &= 0 \\ P(3) &= 0 \\ &\dots \\ P(7) &= 0 \end{aligned} \tag{5.6}$$

Apoi la fiecare pas pe contur, în funcție de noul cod în lanț se fac următoarele cumulări de termeni:

a) Pentru codul 0 (\rightarrow)

$$\begin{aligned} X &= X+1 \\ P(3) &= P(3) - Y^2 \\ P(7) &= P(7) - Y^3 \end{aligned} \tag{5.7}$$

b) Pentru codul 2 (\leftarrow)

$$\begin{aligned} X &= X-1 \\ P(3) &= P(3) + Y^2 \\ P(7) &= P(7) + Y^3 \end{aligned} \tag{5.8}$$

c) Pentru codul 1 (\uparrow)

$$\begin{aligned} Y &= Y-1 \\ P(1) &= P(1) - X \\ P(2) &= P(2) - X^2 \\ P(4) &= P(4) - X^3 \\ P(5) &= P(5) - X^2 \cdot Y \\ P(6) &= P(6) + X^2 \end{aligned} \tag{5.9}$$

d) Pentru codul 3 (\downarrow)

$$\begin{aligned} Y &= Y+1 \\ P(1) &= P(1) + X \\ P(2) &= P(2) + X^2 \\ P(4) &= P(4) + X^3 \\ P(5) &= P(5) + X^2 \cdot Y \\ P(6) &= P(6) + X^2 \end{aligned} \tag{5.10}$$

După parcurgerea întregului contur din termenii $P(i)$ se calculează:

$$\begin{aligned} AN1A &= P(1) \\ S1 &= P(2)/2 \\ S2 &= P(3)/2 \\ S12 &= P(4)/3 \\ S13 &= (2P(5) - P(6))/4 \\ S22 &= P(7)/3 \end{aligned}$$

Pe baza codului în lanț un conțor incrementat cu unu la fiecare pas determină perimetrul conturului.

Procedura de actualizare a listei tranzițiilor

S-a adoptat tehnica codificării liniilor matricii imaginii sub forma unei înregistrări cu patru cimpuri notate IS (coloană de început), ID (coloană de sfârșit) COMPID (identificator de componentă și un pointer numit LEG, care indică o altă înregistrare ce se referă la aceeași linie. Dacă înregistrarea e ultima LEG are valoarea -1. La început întreaga imagine are o singură înregistrare pentru fiecare linie inițializată cu valorile (0, NCOL, FOND, -1) unde NCOL e numărul de coloane. În timpul parcurgerii unei componente se creează noi înregistrări pentru pixelii conținuți în componentă. Considerând să pentru o anumită linie care începe cu o tranziție în coloana I_1 și se termină în I_2 listele de înregistrări vor fi actualizate după algoritmul următor :

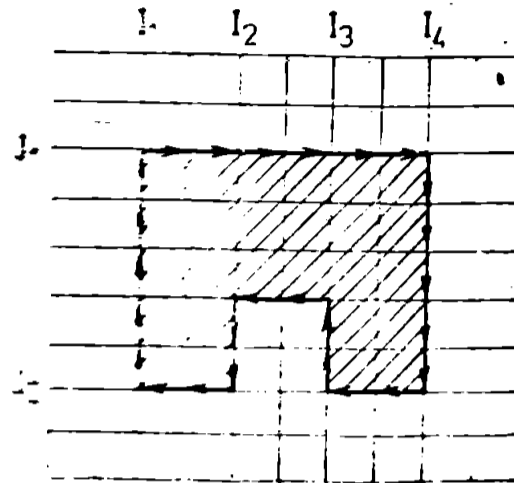
- pas 1. Se caută înregistrarea liniei J care conține codul liniei I_1 la I_2 și aparține fondului.
- pas 2. Se creează o nouă înregistrare de linie cu conținutul (I_1 , I_2 , NOU) unde NOU este numărul noii componente.
- pas 3. Se înlocuiește ID în înregistrarea fondului cu I_1-1 și se introduce în cimpul LEG al acestuia un pointer care indică noua înregistrare creată în pasul 2.
- pas 4. Se creează o nouă înregistrare (I_2+1 , ID, FOND, -1) a cărei cimp LEG va avea vechea valoare din același cimp al înregistrării de fond inițiale și se introduce în cimpul LEG al înregistrării de la punctul 2 valoarea ce indică această înregistrare.

Un exemplu de aplicare a acestui algoritmu este dat în fig.5.21.

5.3.4.4. Relațiile de calcul ale caracteristicilor

Alegerea unui set de caracteristici se va face așa fel încît acestea să descrie componenta din imagine și apoi obiectul cît mai complet. Se poate utiliza o gamă largă de caracteristici dar experimentele au dovedit că în funcție de situația unele sînt mai eficiente decît altele. Se pot utiliza următoarele tipuri de caracteristici :

- de poziție și orientare: coordonatele baricentrului, coordona-



Evoluția liniei J_1 :

a)

0	NCOL	FOND	-1
---	------	------	----

b)

0	?	FOND	
---	---	------	--

I_1	I_4	OBI	
-------	-------	-----	--

I_4+1	NCOL	FOND	-1
---------	------	------	----

c)

0	I_1-1	FOND	
---	---------	------	--

I_1	I_4	OBI	
-------	-------	-----	--

I_4+1	NCOL	FOND	-1
---------	------	------	----

Evoluția liniei J_2 :

a)

0	NCOL	FOND	-1
---	------	------	----

b)

0	?	FOND	
---	---	------	--

?	I_4	OBI	
---	-------	-----	--

I_4+1	NCOL	FOND	-1
---------	------	------	----

c)

0	I_3-1	FOND	
---	---------	------	--

I_3	I_4	OBI	
-------	-------	-----	--

I_4+1	NCOL	FOND	-1
---------	------	------	----

d)

0	?	FOND	
---	---	------	--

?	I_2	OBI	
---	-------	-----	--

I_2+1	I_3-1	FOND	X
---------	---------	------	---

e)

0	I_1-1	FOND	
---	---------	------	--

I_1	I_2	OBI	X
-------	-------	-----	---

Fig. 5.21

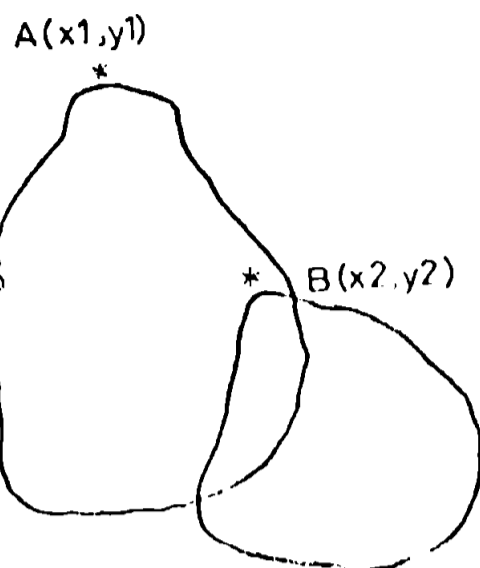


Fig. 5.22

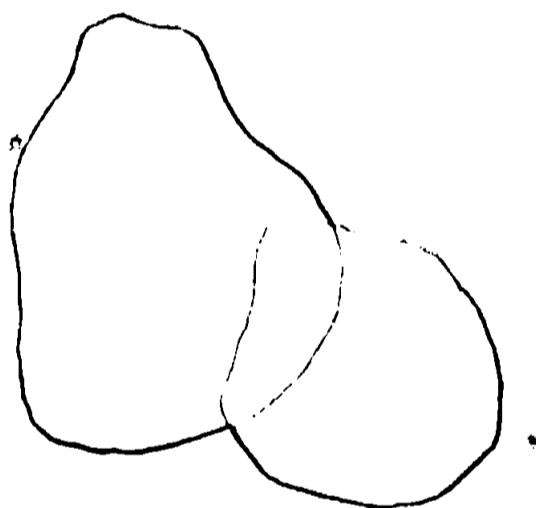


Fig. 5.23

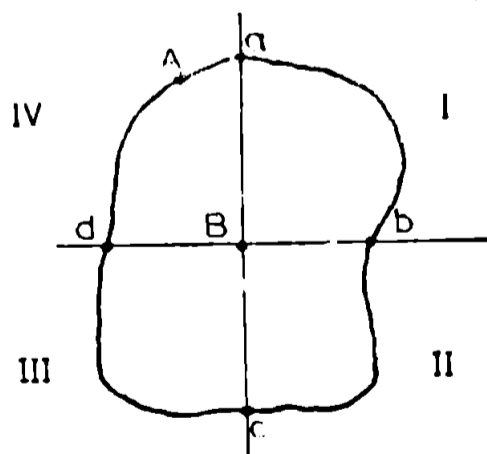


Fig. 5.24

tele centrului de greutate al găurii, direcția axei de inerție minime. Aceste date nu se folosesc la recunoașterea ci numai la cunoașterea locului pe care îl ocupă obiectul și orientării sale.

de formă structurale: numărul de găuri, numărul de unghiuri al curbei de contur exterior

de formă metrice: aria conturului exterior A , ariile găurilor A_i , raporturile adimensionale $\sum A_i/A_0$, perimetrul conturului exterior P , perimetrele găurilor P_i , rapoartele A_i/P_0^2 , P_i/P_0 .

Invarianti geometrice dați de momentele centrale și normalizate de ordinul I și II.

In sistemul de vedere robotică prezentat s-au ales 10 caracteristici: 7 sînt destinate recunoașterii obiectului, 3 pentru poziție și orientare :

1. Aria conturului exterior

$$CAR(1) = AHIA$$

2. Perimetrul conturului exterior

$$CAR(2) = PERIM$$

3. Raportul aria/perim.²

Acest raport e maxim în cazul cercului :

$$q = R^2/(2R)^2 = 1/4 = 0,25$$

Pentru o coloață cu baza 1 și înălțimea 10 raportul e:

$$q = 10/22 = 0,45$$

In scopul obținerii unei valori supraunitare s-a înmulțit raportul cu 100.

$$CAR(3) = AHIA/PERIM.^2 \cdot 100$$

4. Raportul $\sum A_i/A_0$. Pentru fiecare curbă inclusă se calculează aria, se face suma lor S_0 și se obține

$$CAR(4) = S_0/AHIA$$

5. Numărul de găuri.

Se determină tot în urma testelor de includere

$$CAR(5) = NG$$

6. Invariantul geometric β 1.

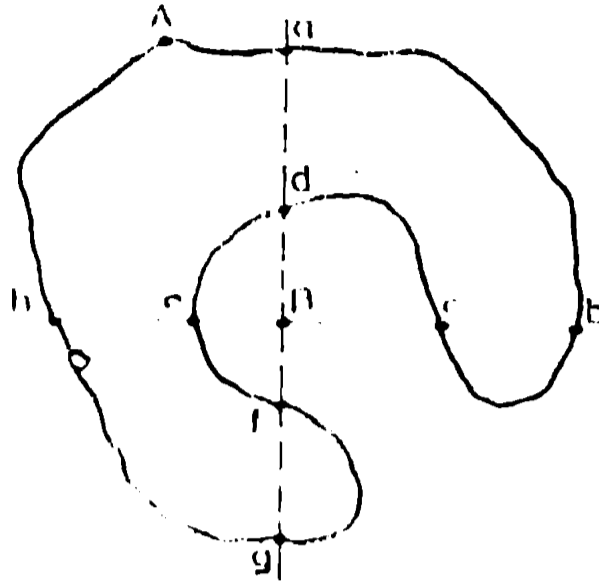


Fig. 5.25

$$\beta_1 = n_{20} - n_{02} = \frac{n_{20}}{n_{00}} - \frac{n_{02}}{n_{00}} =$$

$$= \frac{n_{20} - \bar{x} \cdot n_{10}}{n_{00}} - \frac{n_{02} - \bar{y} \cdot n_{01}}{n_{00}}$$

$$n_{00} = ARIA, \quad n_{10} = SJ, \quad n_{01} = SI$$

$$n_{20} = SJ^2, \quad n_{02} = SI^2, \quad \bar{x} = \frac{SJ}{ARIA}, \quad \bar{y} = \frac{SI}{ARIA}$$

$$\frac{n_{20} - \bar{x} \cdot n_{10}}{n_{00}} = \frac{SJ^2 - \bar{x} \cdot SJ}{ARIA}$$

$$\frac{n_{02} - \bar{y} \cdot n_{01}}{n_{00}} = \frac{SI^2 - \bar{y} \cdot SI}{ARIA}$$

$$CAR(6) = ABS((SJ^2 - SJ \cdot \bar{x}) / ARIA - (SI^2 - SI \cdot \bar{y}) / ARIA) / ARIA$$

7. Invariantul geometric β_2

$$\beta_2 = n_{11} = \left| \frac{u_{11}}{n_{00}} \right|, \quad u_{11} = \iint (x-x)(y-y) dx dy =$$

$$= \iint xy dx dy - \bar{x} \iint y dx dy - \bar{y} \iint x dx dy + \bar{x} \bar{y} \iint dx dy =$$

$$= SIJ + SI \cdot SJ / ARIA - \bar{x} \cdot SI \cdot SJ / ARIA + SI \cdot SJ \cdot ARIA / ARIA^2 - SIJ - SI \cdot SJ / ARIA$$

$$CAR(7) = ABS(SIJ - SI \cdot SJ / ARIA) / ARIA$$

8. Abacisa centrului de greutate

$$CAR(8) = SJ / ARIA$$

9. Ordonata centrului de greutate

$$CAR(9) = SI/ALIA$$

10. Direcția axei de inerție minimă

$$tg \alpha = \frac{2(m_{00} \cdot m_{11} - m_{01} \cdot m_{10})}{(m_{00} \cdot m_{02} - m_{01}^2) - (m_{00} \cdot m_{20} - m_{10}^2)}$$

$$CAR(10) = ATAN(2.(ALIA.SIJ-SI.SJ)/$$

$$((ALIA.SJ^2-SJ.SJ)-(ALIA.SI^2-SI.SI)).27,5$$

Constanta 27,5 a apărut pentru exprimarea unghiului în grade. Programul a fost scris în FORTRAN în varianta inițială pentru imagini la 64x64 pixeli având următoarele limitări:

- permite prelucrarea unor imagini cu maximum 10 curbe de contur.
- lungimea maximă a unui lanț e de 300 de pași.
- numărul maxim de tranziții 0→1 sau 1→0 de pe toate liniile este de 300.

Limitările se datorează memoriei utilizator disponibile pe sistemul de calcul ROBOTRON 4201.

În listingul din fig.5.26 este dat exemplul unui obiect pentru care au fost calculate caracteristicile cu două poziții diferite. Din experimentele realizate cu acest program au rezultat concluzii importante pentru această etapă aparținând procedurii generale a unui sistem de vedere robotică.

Aria calculată prin metoda numărării pixelilor de aceeași culoare manifestă o constanță foarte bună dacă iluminarea este corespunzător făcută încît să nu apară umbre. Perimetrul calculat după metoda numărării vectorilor din lanț dă erori de pînă la 0,4 din valoarea lungimii liniilor oblice cu înclinare de 45°.

Invariantii geometrici S_1 și S_2 sînt destul de constanți cu orientarea dar efectul agomolului asupra lor e mult mai mare decît asupra ariei și perimetrului. Acest lucru demonstrează că aproximarea integralelor prin sumele calculate a fost satisfăcătoare. S-au făcut experimente pe imagini mici pentru a putea fi urmărite ușor și uneori chiar recalculat manual pentru verificarea algoritmilor. Timpul de calcul a fost dependent de dimensiunea și complexitatea imaginii. Pentru imagini de 64x64 pixeli și cu obiecte cu cîteva sute de puncte pe contur intervalale de calcul nu a depășit 2 secunde. Această este o performanță foarte

bună dacă se ține seama că ROBOTRON 4201 are o structură veche de tip PDP-8. Cu o unitate de virgulă flotantă hardware ca la PDP-11/45 se pot obține performanțe sub 1 secundă.

5.3.4.5. Recunoașterea obiectelor și învățarea

Calcularea caracteristicilor în etape anterioară permite o reducere substanțială a capacității memoriei necesare pentru reprezentarea obiectelor. Pentru sistemul de vedere robotic prezentat s-au luat în considerare 7 caracteristici. Nu este nevoie de folosirea tuturor celor 7 caracteristici ci se poate face o alegere. Alegerea trebuie făcută astfel încât caracteristicile să aibă variații suficient de mari de la o clasă de obiecte la alta încât să permită diferențierea acestora. Alegerea este dificilă deoarece pentru 7 caracteristici cu câte 10 intervale de variație fiecare se pot distinge (7×10) clase.

Recunoașterea unui obiect ca aparținând unei clase precedată de învățare, care are ca scop în cazul de față, stabilirea intervalelor de variație a fiecărei caracteristici pentru clasa de obiecte învățată. Prin acest proces spațiul caracteristicilor a împărțit în zone corespunzătoare fiecărei clase. Distincția între două clase se poate face dacă intersecția zonelor corespunzătoare e vidă pentru cel puțin o caracteristică. Se știe că învățarea poate fi realizată cu instructor sau fără. Avantajul primeia este că permite stabilirea limitelor de variație a unei caracteristici prin arătarea unor seturi de date de valori extreme într-un mod foarte elastic.

- Algoritmul de învățare elaborat are următoarele etape:
- pas 1. primește ca date de intrare valorile maxime MAX_1 ale celor 7 caracteristici.
 - pas 2. împarte plaja de variație considerată $0-MAX_1$ în 10 intervale egale
 - pas 3. pentru fiecare set de date nou învățate pentru toate caracteristicile actualizează histogramele de tipul prezentat în fig.5.27.
 - pas 4. se împart cele 10 intervale de variație a caracteristicilor în 3 regiuni:
 - a) S_1 , care cuprinde intervalele în care probabilitatea este aproape 1 ca valoarea caracteristicii să aparțină unei clase învățate
 - b) A_1 , care cuprinde intervalele unde e puțin probabil să se gă-

sensul valori ale caracteristicii pentru clasa învățată.

e) U, restul intervalelor în care nu se poate lua o decizie de clasificare.

Programul de învățare

selecționează în regiunea S o,9 din numărul total de esuri de învățare. Dacă histo-

grama are un singur maxim (cazul obișnuit) atunci regiunea S e formată din intervale vecine. Pentru regiunea U se alege un interval la stânga și la dreapta regiunii S. Dacă intervalele regiunii S ating marginea de jos sau de sus a plajei caracteristicii pentru regiunea U se alege un singur interval.

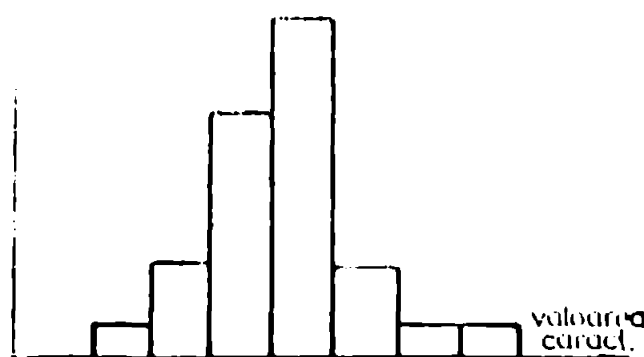


Fig.5.27

Intervalele ce rămân întră în regiunea N.

Ordinograma programului de învățare este prezentată în fig.5.28



Algoritm de recunoaștere primește



ca date de intrare caracteristicile aparținând unor obiecte neidentificate. Un obiect e declarat necunoscut dacă cel puțin o caracteristică a sa cade în regiunea N a tuturor claselor învățate, obiectul e declarat recunoscut și aparținând unei clase dacă fiecare caracteristică a sa are valoarea în regiunea S sau U a clasei. Dacă caracteristicile se situează în regiunile - obiectul nu va fi nici selectat nici respins.

Luarea deciziei de obiect necunoscut se poate face fără a fi nevoie de utilizarea tuturor caracteristicilor. În cazul deciziei de recunoaștere se face o căutare mai complexă doar pînă la caracteristica ce arată o singură clasă posibilă verificându-se cu celelalte caracteristici apartenența la clasa respectivă. Pentru un număr de obiecte învățate lungimea procedurii de recunoaștere depinde de ordinea în care se iau caracteristicile. Unele caracteristici permit discriminarea mult mai rapidă și alocarea obiectului la o clasă sau respingerea lui. Această etapă de optimizare nu a fost luată în considerare în cazul de față pentru că eventualele câștiguri de timp nu ar fi prea substanțiale.

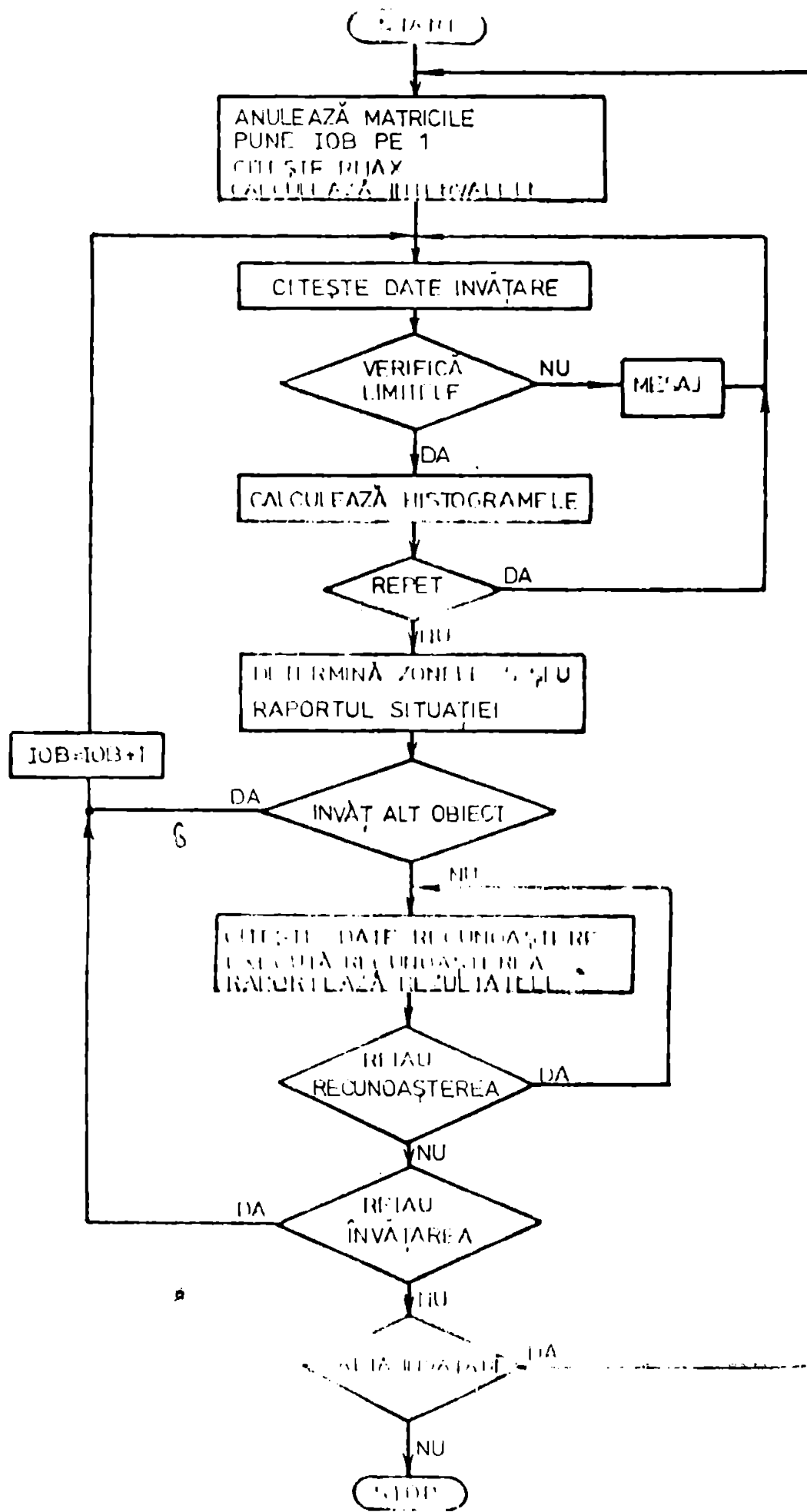


Fig. 5.28

```

  0 0 0 0 1 0 0 0 1 0 0 0 0
  0 0 1 1 1 1 2 0 1 0 0 0 0
  0 1 1 1 1 1 2 0 1 0 0 0 0
  0 1 0 0 0 0 1 0 1 0 2 0 0
  0 1 0 0 0 0 1 0 1 0 1 0 0
  0 1 1 1 1 1 0 1 2 1 0 0 0
  0 1 0 1 0 1 1 0 1 0 1 0 0
  0 1 1 1 1 1 0 1 0 1 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0
  CURBA ZINCEPUT I= 2 J= 8
  COD LAST
  0320332211101111
  CURBA ZINCEPUT I= 5 J= 4
  COD LAST
  030322221110
  CURBA ZINCEPUT I= 6 J= 4
  COD LAST
  03 1
  CURBA ZINCEPUT I= 8 J= 10
  COD LAST
  03 1311110
  CURBA ZINCEPUT I= 9 J= 5
  COD LAST
  03210
  
```

Fig. 5-29


```

DA
V... 10^5.2
101000001000000010100010000001000000100000010000001
V... 10^5.2
07000070000700007000070000700007000070000700007000
INVA...
DA
V... 10^5.2
0720007200072000720007200072000720007200072000720007200
INVA...
DA
V... 10^5.2
05100052000530005400055000560005700058000590005000
INVA...
DA
V... 10^5.2
0400004000041000042000043000044000045000046000047000048000
INVA...
DA
V... 10^5.2
0300003000031000032000033000034000035000036000037000038000
INVA...
DA
V... 10^5.2
0200002000021000022000023000024000025000026000027000028000
INVA...
DA
V... 10^5.2
0100001000011000012000013000014000015000016000017000018000
INVA...
DA
V... 10^5.2
00000000000000000000000000000000000000000000000000000000
INVA...
DA
V... 10^5.2
010000210000310000410000510000610000710000810000910001000
INVA...
DA
V... 10^5.2
0100001000002000003000004000005000006000007000008000090000
INVA...
DA
V... 10^5.2
00000000000000000000000000000000000000000000000000000000
INVA...
DA
V... 10^5.2
010000210000310000410000510000610000710000810000910001000
INVA...
DA
V... 10^5.2
0100001000002000003000004000005000006000007000008000090000
INVA...

```

Fig. 5.31.

In fig.5.29, 5.30 și 5.31 sînt arătate rezultate ale recunoașterii, învățării și dialogului cu programul.

5.4. Procedură de prelucrare a informației vizuale pentru o lina a cuburilor marcate cu litere

5.4.1. Calculul mediilor pentru aflarea obiectelor

Pentru a prelucra o imagine se încearcă cu un volum mare de date ceea ce constituie o problemă destul de serioasă pentru sistemele de calcul mici și chiar mini datorită insuficienței capacității memoriilor operative. Lucrînd cu o imagine de 320x 288 pixeli și memorînd fiecare pixel pe cîte un cuvînt ar fi necesar un spațiu de memorie de 100 k cuvinte în mini sistemul de calcul ROBOTRON numai pentru memorarea imaginii. Cum însă în memorie trebuie să mai existe înafară de datele de prelucrat și programele de prelucrare împreună cu zone de lucru destul de mari se aplică diferite tehnici de prelucrare pe porțiuni de imagine. Una din aceste tehnici este cea a ferestrelor care încearcă să încadreze un anumit obiect aflat în imagine reducînd drastic, numărul pixelilor de memorat și analizați.

În cadrul unei prime explorări procedura de prelucrare a imaginii condensează imaginea inițială în pătrate (mm) sau dreptunghiuri (mm) pentru care se face media aritmetică a pixelilor conținuți.

Imediat după terminarea achiziției imaginii aceasta este memorată pe bandă magnetică sub forma unui fișier conținînd blocuri cu număr fix de linii de imagine. Pentru condensarea imaginii trebuiesc alese dimensiunile pătratului sau dreptunghiului în care se cerește medierea. Acest lucru se face cunoscînd dimensiunile obiectelor din imagine astfel ca acestea să fie cuprinse în cel mai puțină valori medii (și cît mai apropiate). În cazul nostru obiectele sînt cuburi a căror față vizibilă apare ca un pătrat de 47x47 mm aflate într-o suprafață vizibilă de către camere TV de 340x280 mm.

Suprafața unui cub ocupă în acest caz aproximativ 40x40 de pixeli de ceea ce-a ales o rețea de pătrate cu latura de 1,68 cm care cuprinde un număr de 16x16 pixeli fiecare. Astfel rezultă o nouă matrice denumită a mediilor avînd o rezoluție mult mai mică de 20x18 elemente. Prin impunerea dimensiunilor menționate un obiect (cub) indiferent de orientarea și poziția

sa va fi cuprins in maximum patru valori medii. In fig.5.32 este prezentată o matrice a valorilor medii. In imaginea inițială au fost 4 suburi ceea ce se observă în concentrarea unor valori de la 1 la 5 in anumite zone.

Explorarea acestei matrici în vederea alegerii unui maximum de 4 valori medii pentru fiecare obiect însope de la stînga la dreapta pe linii, memorîndu-se indicii liniei și coloanei pentru fiecare element diferit de zero. Găsirea efectivă a unui obiect se face prin compararea elementelor matricii valorilor medii cu un prag stabilit în funcție de luminositatea la care au loc experimentele. Dacă testul indică că elementul este mai mare sau egal cu cea valoare de prag aleasă el este un candidat pentru a fi inclus într-un obiect mai mare.

Odată ce s-a identificat un element drept candidat pentru un obiect se declanșează un proces de căutare într-o fereastră în jurul elementului examinat. Dimensiunile ferestrei sînt variabile. In general, dimensiunile ferestrei de studiu se bazează pe dimensiunile maxime ale obiectelor care pot apărea în scenă.

Presupunînd un obiect în orice orientare trebuie să obținem un indice de prezență sigură a obiectului. Astfel, dimensiunea elementelor mediate oferă certitudinea prezenței obiectului în procent de 94% pe trei valori mediate vecine elementului de medie descoperit ca fiind peste sau egal cu pragul. Pentru a avea certitudinea "prinderii" obiectului în fereastra de studiu senza obligații se mai bazează cu o figură înconjurătoare de lățimea a 16 pixeli. In fig.5.33 de exemplu a fost găsit candidat pentru valoarea de prag egală cu 3 elementul de la inter-

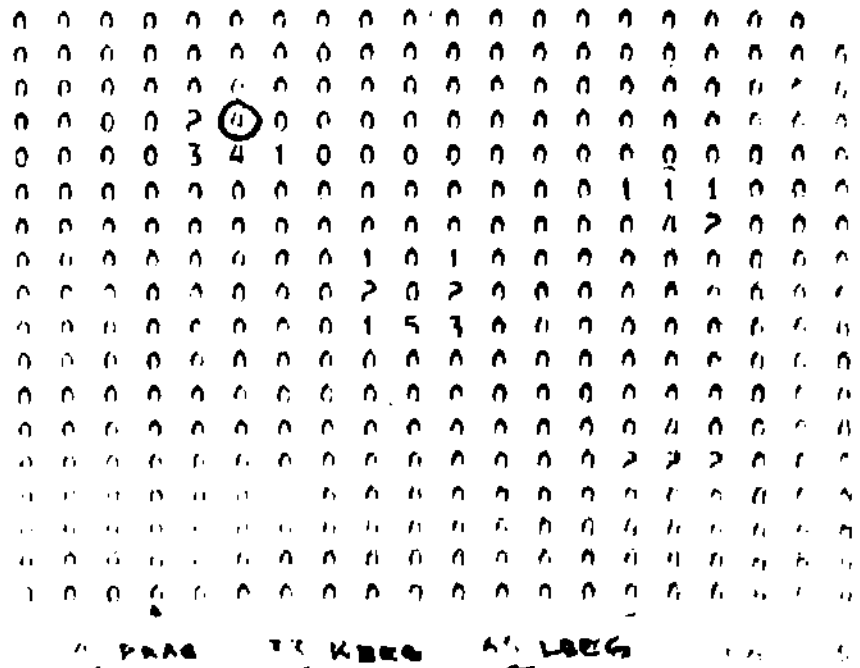


Fig.5.32

secția liniei I cu coloana J. În urma bordării se raportează ca început de obiect linia I-1 și coloana J-1. Prin aceasta sînt înclasați în fereastra de studiu pixelii de margine ai obiectului care au media mică (linia I-1, coloana J) de numai 1 și care altfel s-ar pierde. Rezultă

dimensiunea ferestrei de studiu pentru rețeaua de mediere cu 16x16 pixeli, de 72x72 pixeli suficientă pentru a cuprinde un sub în orice poziție și mai ales orientare.

Calculul matricii valorilor medii se face de către subprogramul S2 care calculează o linie a matricii la fiecare două citiri de pe bandă.

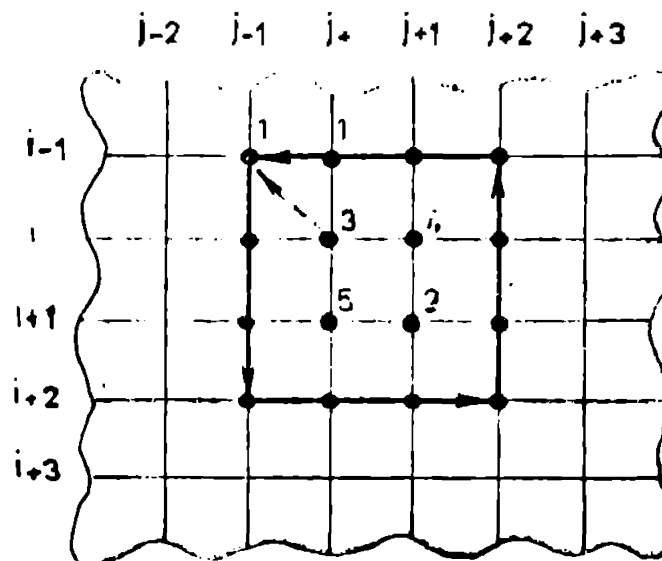


Fig.5.33

5.4.2. Extragerea ferestrei de studiu și calculul histogramei

Aceste operații sînt realizate de subrutina S3 care mai calculează și valoarea de prag. Aceasta din urmă are ca scop determinarea prezenței situației de literă marcată pe suprafața cubului și excluderea unor eventuale umbre sau zgomete rezultate din digitizarea imaginii.

Calculul histogramei înseamnă determinarea frecvenței de apariție a fiecărui tip de pixel de la 0 la 15 prin numărarea lor. În acest scop se extrage mai întâi fereastra de studiu prin citiri de pe bandă. Pentru început în subprogram sînt calculate coordonatele (linie și coloană) de început ale ferestrei de studiu. Pe baza coordonatelor primite în urma inspecției matricii valorilor medii (variabilele II, JJ) se obțin variabilele:

$$LREG = (II-1) \times 16 - 15 \text{ și}$$

$$KREG = (JJ-1) \times 16 - 15$$

Raportul $N=LREG/8$ dă numărul de blocuri de imagine citite în gol pentru ca la blocul $N+1$ să înceapă algoritmul de obținere a ferestrei prin identificarea fiecărui element din blocul de 8 linii citit, înscind cu coordonatele LREG, KREG și la coordonatele

LEEG+72 respectiv KEEG+72. In cazul in care in timpul construirei ferestrei de studiu se atinge valoarea 8 a liniei din tampon se cere o nouă citire de pe bandă și algoritmul se repetă. Spre exemplu in fig.5.32 pentru obiectul din soluțul stâng al matricii valorilor medii și valoarea de prag 4 a fost detectat elementul candidat cu valoarea 4 (înseruit) și au fost determinate valorile LEEG=33 și KEEG=65 care sînt tipărite sub matrice. In fig.5.34 este redată imaginea unui cub marcat cu litera B încadrat în fereastra de studiu de 72x72.

In ce privește calculul histogrammei din cei peste 5000 ai ferestrei de studiu, obiectul ocupă doar 1000 de pixeli ceea ce înseamnă că în histogramă există două maxime distincte și anume unul corespunzător fondului și celălalt corespunzător obiectului. Pragul se alege undeva între cele două valori de vîrf și anume acolo unde histograma prezintă un minim local. Această determinare a pragului are avantajul variației minime a ariilor obiectelor în condiții de iluminare diferite. In fig.5.34 histograma pentru valorile de cenzură de la 0-15 se află tipărită sub figură.

5.4.3. Izolarea obiectului în fereastra de studiu

Fereastra de studiu este constituită astfel încît obiectul din imagine asupra căruia ne concentram să fie cuprins de aceasta indiferent de poziție și orientare. Datorită acestui lucru nu mai este necesar un algoritma de căutare în matricea valorilor medii prin care să se testeze care din vecinii primei valori mai mari decît pragul păstrează această proprietate. Există certitudinea încadrării obiectului (literei) în fereastra de studiu dar nu este eliminată posibilitatea apariției unei părți a unui obiect învecinat. De asemenea, dacă fondul pe care sînt așezate obiectele prezintă pete vizibile.

acestea apar și ele în fereastra de studiu. Pentru o analiză corectă aceste prezențe nu trebuiesc luate în considerare și de aceea se aplică algoritmul de îndepărtare a lor. Experiența a arătat că valoarea minimului local (mărită cu o unitate) este mult mai mare decît cea pe care o au pixelii ce reprezintă petele sau alte sgonete nedorite. Astfel, acestea pot fi eliminate prin simplul test de comparare a valorilor respective cu pragul

local. Problema devine mai complicată atunci când apar părți din diferite alte obiecte învecinate. Pentru a putea rezolva cu succes această situație se impune o singură restricție anume ca obiectele să fie separate prin cel puțin un rând de pixeli între obiect și corpul străin ce apare în fereastră (1-1,5 mm). Pentru înțelegerea algoritmului trebuie făcută o precizare și anume: în fereastra de studiu nu pot să apară două sau mai multe obiecte în întregime. Așadar, obiectul străin (sau obiectele) va apare doar pe marginile ferestrei deci algoritmul constă în testarea din exterior spre interior a valorilor pixelilor de pe linie, respectiv coloane. Dimensiunile ferestrei de studiu se reduc pe măsură ce se detectează linii/coloane cu elemente având valori mai mari decât pragul obținut din histogramă. Dacă obiectul este prins defectuos în fereastră (unii dintre pixelii obiectului apărând pe margine se ratează procesul, fiind necesară repetarea algoritmului de găsiră a ferestrei.

Obținerea conturului obiectelor este o operație care urmează după eliminarea corpurilor străine (părți ale unor obiecte aflate la marginea ferestrei). S-au folosit două metode de obținere a conturului :

- a) operatorul lui Sobel aplicat după binarizarea imaginii (fig.5.35)
- b) determinarea conturului exterior bazat pe tranziții importante lumină-întuneric sau întuneric-lumină.

Deoarece operatorul Sobel a fost deja prezentat se va discuta doar metoda a doua. Algoritmul b) are ca rezultat obținerea marginii exterioare a obiectului fără a mai pune în evidență eventualele găuri pe care le poate prezenta silueta unui obiect, urmând ca în etapele următoare să se efectueze calcule pe fișii de obiect în care găurile intervin în calcule cu valoarea zero nemodificând rezultatele finale. Algoritmul este simplu și rapid din punct de vedere al timpului de calcul. Astfel se începe cu pixelul (1,1) în fereastra de studiu evoluind pe linie spre dreapta testând pixelii cu valoarea pragului rezultat din histogramă. În momentul detectării unei tranziții întuneric-lumină se memorează indicele pixelului după care se sare la capătul din dreapta al rândului căutându-se de această dată punctul de pe conturul exterior opus. Algoritmul continuă trecând pe linia următoare ș.a.m.d. Prin urmare cele două valori extreme cuprind o fișie a obiectului. În figura 5.34 este tipărită ma-

tricea cu două linii și 72 de coloane ce se obține în urma aplicării algoritmului. Subprogramul S4 realizează izolarea obiectului după cele două metode menționate mai sus prin apelarea subrutinei S5.

5.4.4. Calculul caracteristicilor

În acest moment, faza de calcul a caracteristicilor poate fi declanșată. Scopul primordial al calculului caracteristicilor este obținerea pe baza datelor a unui set de caracteristici care să fie reprezentativ pentru un anumit obiect. De exemplu litera I este "subțire" iar litera O este "rotundă". Atât subțirimea cât și rotunjimea pot fi exprimate matematic cantitativ folosind multe tehnici diferite. Una din tehnici este utilizarea momentelor de inerție de ordinul 2 normalizate și centrate.

Relațiile de calcul după referința /B4 80/ sînt date mai jos:

• pentru calculul ariei :

$$A = \sum_{i=B(1)}^{B(2)} \sum_{j=b(i,1)}^{b(i,2)} g(i,j)$$

formulă în care $g(i,j)$ este valoarea nivelului de cenușă al pixelului curent. Aceasta înseamnă că pixelul de coordonate (i,j) din fereastra de studiu intervine cu valoarea sa. Calculul se efectuează pe fișii cuprinse pe orizontală între valorile $b(1)$, $b(2)$ iar pe verticală între valorile $B(1)$ și $B(2)$.

• pentru calculul momentelor de inerție normalizate

$$M_{pq} = \frac{1}{A} \sum_{i=B(1)}^{B(2)} \sum_{j=b(i,1)}^{b(i,2)} i^p j^q g(i,j)$$

formulă care ne furnizează pentru $p=1$, $q=0$ și $p=0$, $q=1$, momentele de inerție de ordinul 1, M_{10} și M_{01} care sînt și coordonatele centrului de greutate (X_g, Y_g) .

• în fine calculul momentelor de ordinul 2 centrate și normalizate se face cu formulele :

$$\theta = \tan^{-1} \left[\frac{-2M_{11}}{M_{20} - M_{02}} \right]$$

$$M'_{20} = \left(\frac{M_{20} + M_{02}}{2} \right) + \left(\frac{M_{20} - M_{02}}{2} \right) \cos 2\theta - M_{11} \sin 2\theta$$

$$M'_{02} = (M_{20} - M_{02}) - M'_{20}$$

in care M'_{20} și M'_{02} sînt momentele secunde centrate maxim și minim, iar θ este unghiul de orientare. In tabelul 5.1 sînt prezentate rezultatele măsurătorilor efectuate pentru cinci tipuri de litere A, B, C, I și U pe baza a 12 imagini în care acestea ocupă diferite poziții și orientări. Sînt prezente 3 coloane de valori pentru fiecare literă: aria, M_{20F} și M_{02F} .

In figura 5.36 sînt ilustrate momentele de ordinul doi normalizate și centrate pentru diferite litere din alfabet care au fost folosite în lumea robotului. Momentele minime se află de-a lungul abscisei iar pe ordonată avem momentele maxime. Această reprezentare înfățișează un spațiu al caracteristicilor bidimensional și constituie baza pentru recunoașterea vizuală care urmează.

Fig.5.36 ne arată că literele alese (A, B, C, U, I) sînt relativ izolate în spațiul caracteristicilor indiferent de orientare. Alte litere precum H și E sînt mult prea apropiate în spațiul caracteristicilor și pot fi ușor confundate. In scopul recunoașterii lor pentru o separare completă a claselor sînt necesare alte caracteristici cum ar fi de pildă aria. După ce calculul caracteristicilor a fost efectuat localizarea și orientarea fiecărui cub sînt cunoscute. Direcția momentului maxim de ordinul doi normalizat și centrat este luată ca axă a cubului cu care sînt robotului se va alinia eventual.

5.4.5. Invățarea obiectelor din imagine

Extragerea caracteristicilor din etapa anterioară înseamnă o reducere substanțială a volumului de date necesar memorării, de la sute de koceteți la numai zeci de oceteți. Invățarea este o etapă premergătoare etapei de recunoaștere. Se cunosc două tipuri de învățare :

Tabellul 5.1

litera A		litera B		litera C		litera I		litera U						
A	OM ₂₀ ¹	A	OM ₂₀ ¹	A	OM ₂₀ ¹	A	OM ₂₀ ¹	A	OM ₂₀ ¹					
OM ₂₀ ¹	OM ₀₂ ¹	OM ₂₀ ¹	OM ₀₂ ¹	OM ₂₀ ¹	OM ₀₂ ¹	OM ₂₀ ¹	OM ₀₂ ¹	OM ₂₀ ¹	OM ₀₂ ¹					
261	82	15	330	130	10	288	187	40	204	108	7	526	120	91
234	61	10	393	93	22	342	110	34	162	59	3	523	128	79
267	52	11	399	84	50	351	79	52	189	99	6	525	142	71
270	80	18	471	119	20	327	163	20	174	80	2	568	147	66
279	81	12	408	111	27	318	159	20	171	98	11	539	140	73
291	84	13	339	111	18	282	139	31	178	101	10	627	115	98
258	72	19	396	88	30	351	84	52	171	52	2	514	142	72
303	54	24	384	82	34	339	74	60	198	70	3	520	130	81
201	52	20	435	97	45	282	77	56	190	74	4	524	125	87
282	64	17	408	91	37	354	101	37	187	73	3	513	123	90
261	48	16	411	103	40	327	88	43	172	71	5	517	126	94
294	50	18	405	112	45	330	97	40	180	72	6	518	128	91

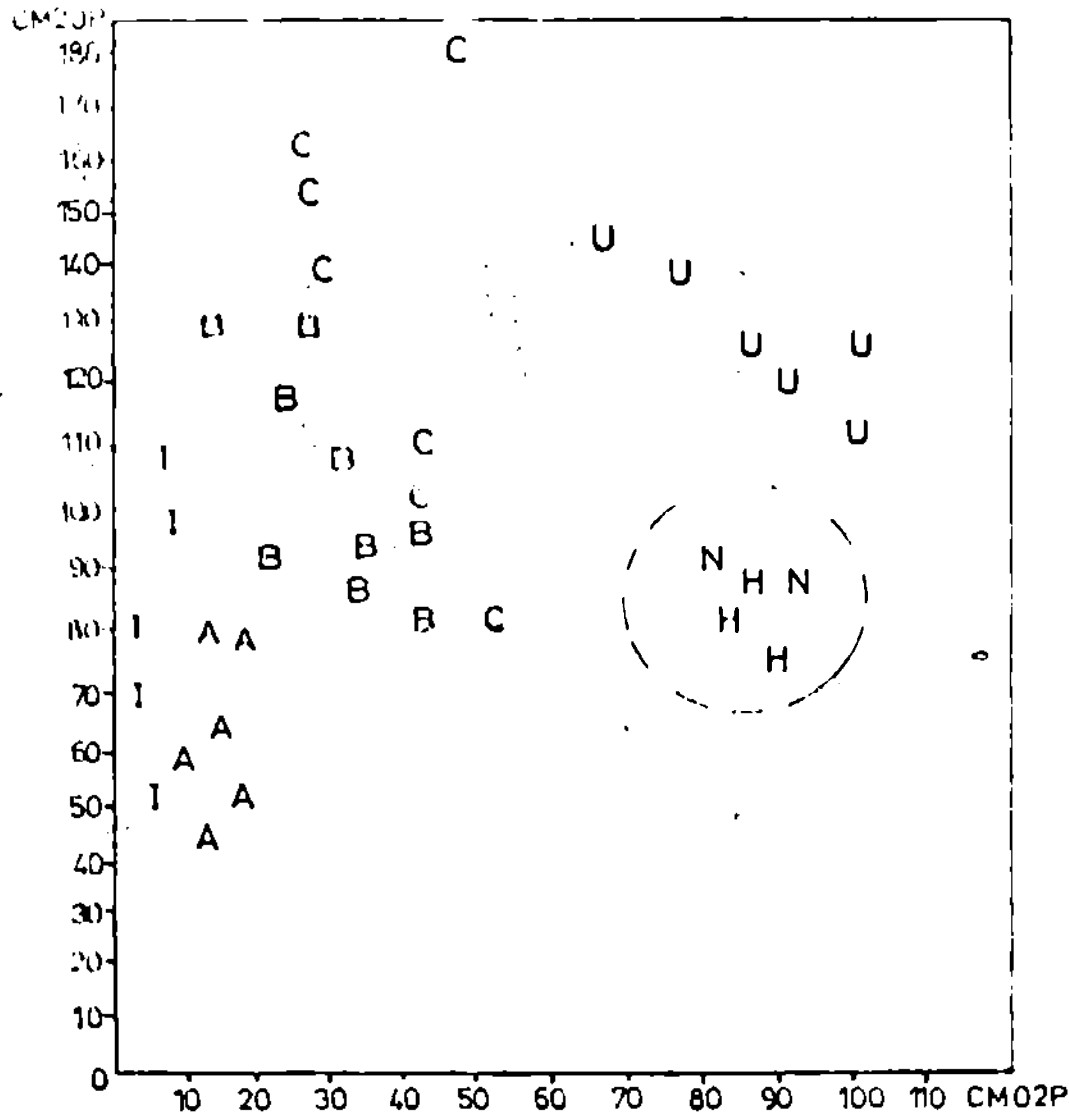


Fig.5.36

- a) învățare supervizată, la care edată cu arătarea setului de date se indică și clasa de apartenență;
- b) învățare nesupervizată, în care sistemul de calcul la arătarea primului set de date alege o clasă de obiecte sau include setul arătat într-una din clasele deja alese.

În general învățarea este un proces statistic în urma căruia se construiesc prototipurile pentru fiecare clasă de obiecte. O foarte mare importanță o are alegerea cazurilor de învățare (exemplul de învățare) astfel încât să se poată stabili pentru fiecare caracteristică întreg intervalul de variație posibilă a valorilor tipice clasei respective. Acest lucru s-a realizat înfățișând sub camera TV obiectele în câte 12 poziții și trei asien-

tări (0° , 45° , 90°) diferite, așa cum se indică în tabelul 5.1 și fig.5.36. Procesul de învățare este cu atât mai eficient cu cât numărul de cazuri prezentat este mai mare. Se impun limite însă în timpul de prelucrare și spațiul de memorie ocupat.

În cazul de față, așa cum se procedează în majoritatea sistemelor de V.C.C. aplicate în robotică s-a aplicat învățarea supervizată. Aceasta se caracterizează prin simplitate constând în memorarea caracteristicilor prototipului și a clasei de apartenență. Aceste date sînt memorate pe bandă magnetică în timpul execuției subprogramului 84. Sfîrșitul sesiunii de învățare este marcat printr-un semn special permițîndu-se completarea acestuia cu noi clase de obiecte fără a mai fi necesară reluarea întregii sesiuni.

5.4.6. Recunoașterea obiectelor după regula celui mai apropiat k vecin

Se prezintă în cele ce urmează metoda recunoașterii bazată pe tehnici nonparametrice descrisă de Duda și Hart [DH 73], care a fost aplicată în varianta celui mai apropiat k vecin.

Fie $X^n = (x_1, x_2, x_3, \dots, x_n)$ mulțimea eșantionelor etichetate și fie $x \in X^n$ eșantionul cel mai apropiat de x . În acest caz, regula de clasificare de tipul celui mai apropiat vecin pentru x constă în atribuirea acestuia categoriei asociate lui x . Regula celui mai apropiat vecin este o procedură suboptimală (nu ia în considerare parametrii teoretici). În lucrarea amintită, autorii demonstrează că, cu un număr ridicat de eșantioane, rata erorii nu depășește dublul erorii date de regula lui Bayes.

Pentru aplicația noastră s-a folosit regula celui mai apropiat k vecin. Această regulă clasifică elementul x asignîndu-i eticheta cea mai frecvent apărută în eșantionul k-apropiat. Cu alte cuvinte o decizie este luată în urma examinării etichetelor celor k-apropiați vecini. Se evită o analiză profundă a metodei, totuși dacă se ia în considerare un caz cu două clase și k impar se poate face o analiză suficient de precisă.

Principala motivare a metodei constă în observabilitatea mai ușoară asupra concordanței probabilității cu situația. Ob-

servăm mai întâi că dacă k este fix și numărul n de eșantioane se apropie de infinit, atunci toate vecinătățile k vor converge spre x . Ca și în cazul regulii vecinului cel mai apropiat, etichetele pentru fiecare vecin k - apropiat sînt distribuite variabil, ω_1 fiind independentă de probabilitatea $P(\omega_1/x)$. Dacă $P(\omega_m/x)$ este cea mai mare probabilitate a posteriori, atunci regula de decizie Bayes selectează întotdeauna ω_m . Regula celui mai apropiat vecin selectează ω_m cu o probabilitate $P(\omega_m/x)$. Regula celui mai apropiat k - vecin selectează ω_m dacă o majoritate a vecinilor k sînt etichetați ω_m , un eveniment cu probabilitatea :

$$\sum_{i=(K+1)/2}^K \binom{K}{i} P(\omega_m/x)^i [1 - P(\omega_m/x)]^{K-i}$$

unde i este numărul de clase.

În general, arată Duda și Hart, cu cît K este mai mare cu atît crește probabilitatea de selecție a lui ω_m . Se poate demonstra că pentru K impar cea mai mare rată a erorilor cu două clase pentru regula celui mai apropiat k -vecin este limitată de funcția $G_K(P^*)$, unde G_K este definită ca fiind cea mai mică funcție concavă a lui P^* mai mare ca:

$$\sum_{i=0}^{(K-1)/2} \binom{K}{i} [(P^*)^{i+1} (1-P^*)^{K-i} + (P^*)^{K-i} (1-P^*)^{i+1}]$$

Graficul prezentat în lucrarea amintită arată că, cu cît crește valoarea lui K , rata erorii se apropie de cea obținută cu regula lui Bayes.

Este indicat prin urmare să se aleagă o valoare cît mai mare K ca să obținem o estimare cît mai apropiată de cea reală dar, pe de altă parte, derin ca toți vecinii K ai lui X' să fie apropiați de x pentru a fi siguri că $P(\omega_1/X')$ să aproximeze $P(\omega_1/x)$. Acest lucru ne obligă să facem un compromis în ceea ce privește valoarea atribuită lui K , care este o fracție din numărul de eșantioane.

Programul de recunoaștere care aplică regula celui mai apropiat k -vecin se numește LITKA și calculează distanța euclidiană dintre punctul de coordonate (A_{1i} , Q_{20P} , Q_{20F}) și valorile analoge înregistrate în cursul învățării. Apoi se ordonează

să aceste valori într-un șir crescător și se aleg primele K distanțe. Între ecantioanele corespunzătoare este desemnată clasa care are frecvența de apariție cea mai mare.

După ordonarea primelor 15 distanțe euclidiene minime este apelat subprogramul LITERA (GAR, K) care stabilește K -vecini cei mai apropiați. Variabila K este prescriptibilă prin mesaj de la consolă. În fig.5.36 s-au prezentat rezultatele obținute în cursul experimentelor de recunoaștere. Se observă (valorile încercuite) că frecvența de apariție a lui λ_n în cadrul primelor 15 vecini este de 11-12, de aici rezultă un K de maximum 12 și minimum 5 așa cum reiese din experimentul al 3-lea.

01.0	01.0	35.6	37.0	37.8	38.6	39.4	40.2	41.0	41.8
02.0	02.0	36.5	38.0	38.9	39.8	40.7	41.6	42.5	43.4
03.0	03.0	37.4	38.9	39.8	40.7	41.6	42.5	43.4	44.3
04.0	04.0	38.3	39.8	40.7	41.6	42.5	43.4	44.3	45.2
05.0	05.0	39.2	40.7	41.6	42.5	43.4	44.3	45.2	46.1
06.0	06.0	40.1	41.6	42.5	43.4	44.3	45.2	46.1	47.0
07.0	07.0	41.0	42.5	43.4	44.3	45.2	46.1	47.0	47.9
08.0	08.0	41.9	43.4	44.3	45.2	46.1	47.0	47.9	48.8
09.0	09.0	42.8	44.3	45.2	46.1	47.0	47.9	48.8	49.7
10.0	10.0	43.7	45.2	46.1	47.0	47.9	48.8	49.7	50.6
11.0	11.0	44.6	46.1	47.0	47.9	48.8	49.7	50.6	51.5
12.0	12.0	45.5	47.0	47.9	48.8	49.7	50.6	51.5	52.4
13.0	13.0	46.4	47.9	48.8	49.7	50.6	51.5	52.4	53.3
14.0	14.0	47.3	48.8	49.7	50.6	51.5	52.4	53.3	54.2
15.0	15.0	48.2	49.7	50.6	51.5	52.4	53.3	54.2	55.1
16.0	16.0	49.1	50.6	51.5	52.4	53.3	54.2	55.1	56.0
17.0	17.0	50.0	51.5	52.4	53.3	54.2	55.1	56.0	56.9
18.0	18.0	50.9	52.4	53.3	54.2	55.1	56.0	56.9	57.8
19.0	19.0	51.8	53.3	54.2	55.1	56.0	56.9	57.8	58.7
20.0	20.0	52.7	54.2	55.1	56.0	56.9	57.8	58.7	59.6
21.0	21.0	53.6	55.1	56.0	56.9	57.8	58.7	59.6	60.5
22.0	22.0	54.5	56.0	56.9	57.8	58.7	59.6	60.5	61.4
23.0	23.0	55.4	56.9	57.8	58.7	59.6	60.5	61.4	62.3
24.0	24.0	56.3	57.8	58.7	59.6	60.5	61.4	62.3	63.2
25.0	25.0	57.2	58.7	59.6	60.5	61.4	62.3	63.2	64.1
26.0	26.0	58.1	59.6	60.5	61.4	62.3	63.2	64.1	65.0
27.0	27.0	59.0	60.5	61.4	62.3	63.2	64.1	65.0	65.9
28.0	28.0	59.9	61.4	62.3	63.2	64.1	65.0	65.9	66.8
29.0	29.0	60.8	62.3	63.2	64.1	65.0	65.9	66.8	67.7
30.0	30.0	61.7	63.2	64.1	65.0	65.9	66.8	67.7	68.6
31.0	31.0	62.6	64.1	65.0	65.9	66.8	67.7	68.6	69.5
32.0	32.0	63.5	65.0	65.9	66.8	67.7	68.6	69.5	70.4
33.0	33.0	64.4	65.9	66.8	67.7	68.6	69.5	70.4	71.3
34.0	34.0	65.3	66.8	67.7	68.6	69.5	70.4	71.3	72.2
35.0	35.0	66.2	67.7	68.6	69.5	70.4	71.3	72.2	73.1
36.0	36.0	67.1	68.6	69.5	70.4	71.3	72.2	73.1	74.0
37.0	37.0	68.0	69.5	70.4	71.3	72.2	73.1	74.0	74.9
38.0	38.0	68.9	70.4	71.3	72.2	73.1	74.0	74.9	75.8
39.0	39.0	69.8	71.3	72.2	73.1	74.0	74.9	75.8	76.7
40.0	40.0	70.7	72.2	73.1	74.0	74.9	75.8	76.7	77.6
41.0	41.0	71.6	73.1	74.0	74.9	75.8	76.7	77.6	78.5
42.0	42.0	72.5	74.0	74.9	75.8	76.7	77.6	78.5	79.4
43.0	43.0	73.4	74.9	75.8	76.7	77.6	78.5	79.4	80.3
44.0	44.0	74.3	75.8	76.7	77.6	78.5	79.4	80.3	81.2
45.0	45.0	75.2	76.7	77.6	78.5	79.4	80.3	81.2	82.1
46.0	46.0	76.1	77.6	78.5	79.4	80.3	81.2	82.1	83.0
47.0	47.0	77.0	78.5	79.4	80.3	81.2	82.1	83.0	83.9
48.0	48.0	77.9	79.4	80.3	81.2	82.1	83.0	83.9	84.8
49.0	49.0	78.8	80.3	81.2	82.1	83.0	83.9	84.8	85.7
50.0	50.0	79.7	81.2	82.1	83.0	83.9	84.8	85.7	86.6
51.0	51.0	80.6	82.1	83.0	83.9	84.8	85.7	86.6	87.5
52.0	52.0	81.5	83.0	83.9	84.8	85.7	86.6	87.5	88.4
53.0	53.0	82.4	83.9	84.8	85.7	86.6	87.5	88.4	89.3
54.0	54.0	83.3	84.8	85.7	86.6	87.5	88.4	89.3	90.2
55.0	55.0	84.2	85.7	86.6	87.5	88.4	89.3	90.2	91.1
56.0	56.0	85.1	86.6	87.5	88.4	89.3	90.2	91.1	92.0
57.0	57.0	86.0	87.5	88.4	89.3	90.2	91.1	92.0	92.9
58.0	58.0	86.9	88.4	89.3	90.2	91.1	92.0	92.9	93.8
59.0	59.0	87.8	89.3	90.2	91.1	92.0	92.9	93.8	94.7
60.0	60.0	88.7	90.2	91.1	92.0	92.9	93.8	94.7	95.6
61.0	61.0	89.6	91.1	92.0	92.9	93.8	94.7	95.6	96.5
62.0	62.0	90.5	92.0	92.9	93.8	94.7	95.6	96.5	97.4
63.0	63.0	91.4	92.9	93.8	94.7	95.6	96.5	97.4	98.3
64.0	64.0	92.3	93.8	94.7	95.6	96.5	97.4	98.3	99.2
65.0	65.0	93.2	94.7	95.6	96.5	97.4	98.3	99.2	100.1
66.0	66.0	94.1	95.6	96.5	97.4	98.3	99.2	100.1	101.0
67.0	67.0	95.0	96.5	97.4	98.3	99.2	100.1	101.0	101.9
68.0	68.0	95.9	97.4	98.3	99.2	100.1	101.0	101.9	102.8
69.0	69.0	96.8	98.3	99.2	100.1	101.0	101.9	102.8	103.7
70.0	70.0	97.7	99.2	100.1	101.0	101.9	102.8	103.7	104.6
71.0	71.0	98.6	100.1	101.0	101.9	102.8	103.7	104.6	105.5
72.0	72.0	99.5	101.0	101.9	102.8	103.7	104.6	105.5	106.4
73.0	73.0	100.4	101.9	102.8	103.7	104.6	105.5	106.4	107.3
74.0	74.0	101.3	102.8	103.7	104.6	105.5	106.4	107.3	108.2
75.0	75.0	102.2	103.7	104.6	105.5	106.4	107.3	108.2	109.1
76.0	76.0	103.1	104.6	105.5	106.4	107.3	108.2	109.1	110.0
77.0	77.0	104.0	105.5	106.4	107.3	108.2	109.1	110.0	110.9
78.0	78.0	104.9	106.4	107.3	108.2	109.1	110.0	110.9	111.8
79.0	79.0	105.8	107.3	108.2	109.1	110.0	110.9	111.8	112.7
80.0	80.0	106.7	108.2	109.1	110.0	110.9	111.8	112.7	113.6
81.0	81.0	107.6	109.1	110.0	110.9	111.8	112.7	113.6	114.5
82.0	82.0	108.5	110.0	110.9	111.8	112.7	113.6	114.5	115.4
83.0	83.0	109.4	110.9	111.8	112.7	113.6	114.5	115.4	116.3
84.0	84.0	110.3	111.8	112.7	113.6	114.5	115.4	116.3	117.2
85.0	85.0	111.2	112.7	113.6	114.5	115.4	116.3	117.2	118.1
86.0	86.0	112.1	113.6	114.5	115.4	116.3	117.2	118.1	119.0
87.0	87.0	113.0	114.5	115.4	116.3	117.2	118.1	119.0	119.9
88.0	88.0	113.9	115.4	116.3	117.2	118.1	119.0	119.9	120.8
89.0	89.0	114.8	116.3	117.2	118.1	119.0	119.9	120.8	121.7
90.0	90.0	115.7	117.2	118.1	119.0	119.9	120.8	121.7	122.6
91.0	91.0	116.6	118.1	119.0	119.9	120.8	121.7	122.6	123.5
92.0	92.0	117.5	119.0	119.9	120.8	121.7	122.6	123.5	124.4
93.0	93.0	118.4	119.9	120.8	121.7	122.6	123.5	124.4	125.3
94.0	94.0	119.3	120.8	121.7	122.6	123.5	124.4	125.3	126.2
95.0	95.0	120.2	121.7	122.6	123.5	124.4	125.3	126.2	127.1
96.0	96.0	121.1	122.6	123.5	124.4	125.3	126.2	127.1	128.0
97.0	97.0	122.0	123.5	124.4	125.3	126.2	127.1	128.0	128.9
98.0	98.0	122.9	124.4	125.3	126.2	127.1	128.0	128.9	129.8
99.0	99.0	123.8	125.3	126.2	127.1	128.0	128.9	129.8	130.7
100.0	100.0	124.7	126.2	127.1	128.0	128.9	129.8	130.7	131.6

01.0	01.0	31.1	32.5	33.4	34.3	35.2	36.1	37.0	37.9
02.0	02.0	32.0	33.4	34.3	35.2	36.1	37.0	37.9	38.8
03.0	03.0	32.9	34.3	35.2	36.1	37.0	37.9	38.8	39.7
04.0	04.0	33.8	35.2	36.1	37.0	37.9	38.8	39.7	40.6
05.0	05.0	34.7	36.1	37.0	37.9	38.8	39.7	40.6	41.5
06.0	06.0	35.6	37.0	37.9	38.8	39.7	40.6	41.5	42.4
07.0	07.0	36.5	37.9	38.8	39.7	40.6	41.5	42.4	43.3
08.0	08.0	37.4	38.8	39.7	40.6	41.5	42.4	43.3	44.2
09.0	09.0	38.3	39.7	40.6	41.5	42.4	43.3	44.2	45.1
10.0	10.0	39.2	40.6	41.5	42.4	43.3	44.2	45.1	46.0
11.0	11.0	40.1	41.5	42.4	43.3	44.2	45.1	46.0	46.9
12.0	12.0	41.0	42.4	43.3	44.2	45.1	46.0	46.9	47.8
13.0	13.0	41.9	43.3	44.2	45.1	46.0	46.9	47.8	48.7
14.0	14.0	42.8	44.2	45.1	46.0	46.9	47.8	48.7	49.6
15.0	15.0	43.7	45.1	46.0	46.9	47.8	48.7	49.6	50.5
16.0	16.0	44.6	46.0	46.9	47.8	48.7	49.6	50.5	51.4
17.0	17.0	45.5	46.9	47.8	48.7	49.6	50.5	51.4	52.3
18.0	18.0	46.4	47.8	48.7	49.6	50.5	51.4	52.3	53.2
19.0	19.0	47.3	48.7	49.6	50.5	51.4	52.3	53.2	54.1
20.0	20.0	48.2	49.6	50.5	51.4	52.3	53.2	54.1	55.0
21.0	21.0	49.1	50.5	51.4	52.3	53.2	54.1	55.0	55.9
22.0	22.0	50.0	51.4	52.3	53.2	54.1	55.0	55.9	56.8
23.0	23.0	50.9	52.3	53.2	54.1	55.0	55.9	56.8	57.7
24.0	24.0	51.8	53.2	54.1	55.0	55.9	56.8	57.7	58.6
25.0	25.0	52.7	54.1	55.0	55.9	56.8	57.7	58.6	59.5
26.0	26.0	53.6	55.0	55.9	56.8	57.7	58.6	59.5	60.4
27.0	27.0	54.5							

CAPITOLUL 6

EXPERIMENTE "OCHI-MINA-VEHICUL" REALIZATE CU ROBOTUL P.E.T.R.I.C.A.

6.1. Integrarea resurselor robotice

In capitolele anterioare, 3, 4 și 5 au fost prezentate principalele resurse robotice de care se dispune pentru desfășurarea în condiții de laborator a unor experimente de tip "ochi-mână-vehicul". Aceste resurse sînt în ordine : (1) robotul mobil P.E.T.R.I.C.A. ; (2) Sistemele de calcul interconectate formînd un ansamblu de trei calculatoare; (3) sistemul de vedere robotic în care se includ cele două camere TV, cele două monitoare TV-TEHNOTOM, digitizorul de imagini și programele procedurii de vedere a cuburilor marcate cu litere.

La aceste resurse de tip echipament (hardware) se adaugă cele de tip programant (software). În fig.6.1 sînt reprezentate ambele tipuri de resurse care împreună formează aşanumitul mediu al sistemului de dezvoltare robotic.

MEDIUL SISTEMULUI DE DEZVOLTARE ROBOTIC

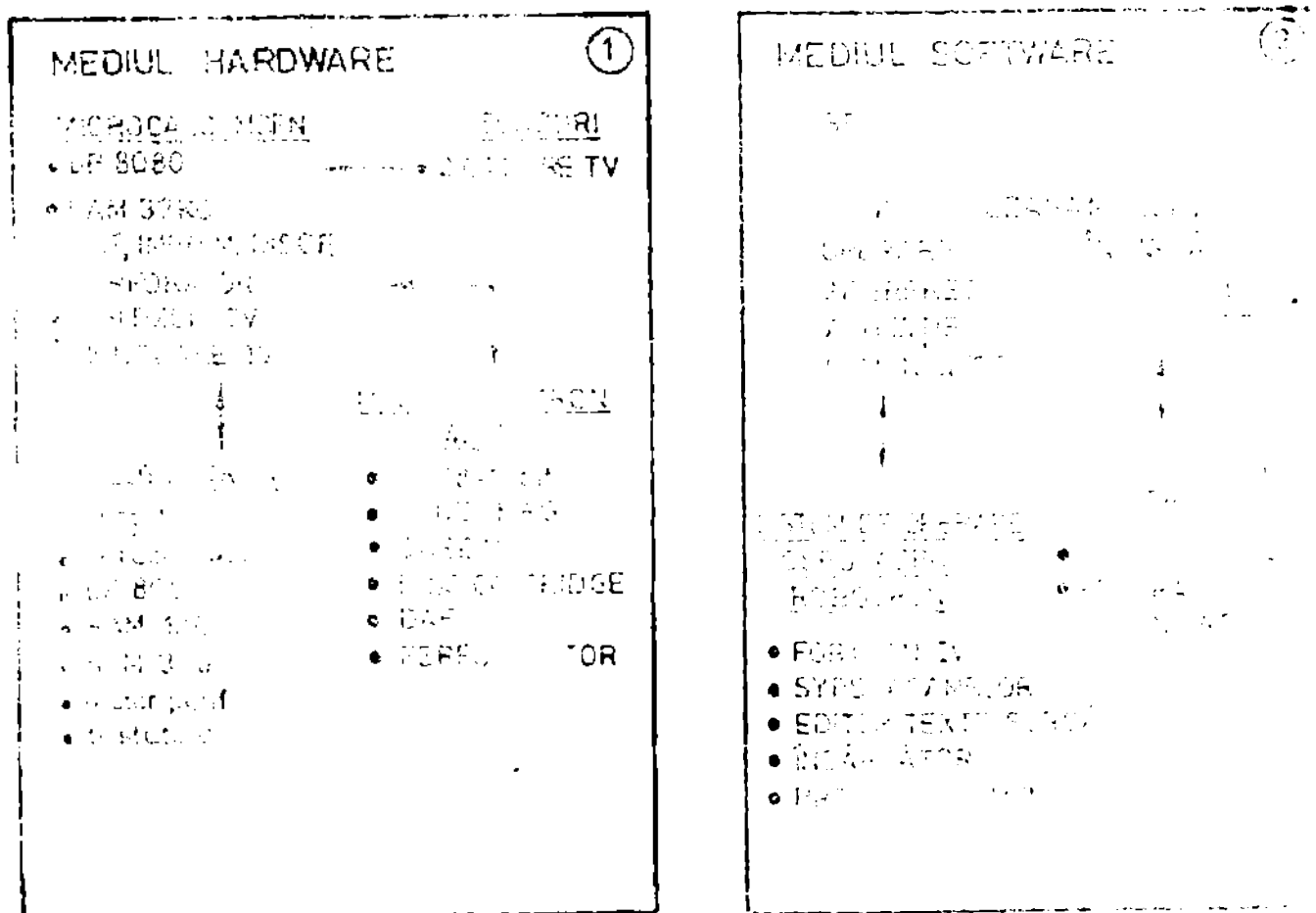


Fig.6.1

Așa cum s-a evidențiat în cap.1, un proiect robotic pornește de la o formulare abstractă a proiectului care necesită apoi o convertire în părți de hardware și software. În cadrul proiectului se elaborează aspectele principale ale modului în care vor interacționa algoritmi implementați în software cu electronica de acționare, cu senzori și apoi acestea din urmă cu lumea tridimensională alcătuită din sisteme efectoare, optice și fenomenele fizice însoțitoare.

S-a subliniat deja importanța sistemelor de calcul atât ca mijloc de proiectare cât și ca element de proiectare. Ele constituie principalul instrument de transpunere a unui proiect în practică. Suporturile software la rândul lor joacă un rol primordial în procesul de proiectare. Sistemul robotic final include automatizarea unor funcții diverse ca mișcările secvențiale ale robotului sau secvențe de semnale electrice. Dezvoltarea unui sistem robotic înseamnă automatizarea procesului de elaborare de programe, de taskuri de simulare, toate necesare creării de algoritmi destinați funcțiilor amintite. Iată deci în cadrul mediului de dezvoltare robotic mijloacele software au o influență covârșitoare. Aceasta înseamnă că utilizarea unor sisteme de calcul în dezvoltarea unui laborator de robotică - aplicație importantă a științei și tehnicii de calculare drept concept cheie sistemul de dezvoltare software ca set de instrumente software integrate.

Cea dintâi cerință a unui astfel de sistem software este cadrul pe care să-l ofere sistemul de operare. Sistemele de operare au diferite înfățișări : cele simple ale calculatoarelor pe o singură placă (SBC), cele independente de limbaj, tipice pentru calculatoare și microcalculatoare, cele totalmente dependente de limbaj (BASIC) ale calculatoarelor personale și în fine sistemele de operare complexe ale unor mașini LISP sau APL.

Pentru experimentele P.E.T.H.I.C.A. s-a luat decizia formării unui sistem de operare independent de limbaj ale cărui dimensiuni au fost păstrate la proporții rezonabile. El a fost elaborat pentru sistemul de calcul MCFN în limbaj de asamblare și cod-mașină. Tot pe calculatorul MCFN a fost elaborat un interpretor LISP pentru microprocesorul 8080.

Limbajul LISP oferă cele mai bune facilități pentru coman-

de roboților ce include concepte de inteligență artificială.

În fig.6.1 sînt reprezentate atât mediul hardware (1) cît și mediul software (2). Acesta din urmă este extins în continuare prin conceperea și elaborarea de algoritmi și programe care să realizeze funcționalități robotice.

Se pune în continuare problema organizării acestor resurse. Răspunsul la această problemă a fost dat de exemplele roboților Shakey, Mars-Rover și H'lare, de autori ca Nilsson, Evans /Ev 1977/, Albus /Al 81 /, Albus /Al 82 / și Dobrotin sub forma organizării ierarhice. Aspectele ridicate de organizarea ierarhică în sistemele de comandă ale roboților au fost studiate de autor în cadrul unui referat /St. 85/. În concluziile elaborate în acest referat se face o comparație între arhitecturile ierarhice organizate pe verticală propuse de Albus /Al 82/ și cele organizate pe eșaloane dintre care se remarcă cea propusă de Dobrotin /Do 77/. Se arată că organizarea pe eșaloane este preferată de cei mai mulți autori pentru că oferă posibilitatea de a dezvolta separat subsisteme inteligente și ulterior aceste să fie integrate cu un eșalon mai sus.

Corespunzător unui asemenea concept activitățile de nivel superior cum sînt percepția-recunoașterea și planificarea ar urma să aibă loc într-un sistem integrativ central avînd pe nivelul inferior un eșalon compus din două subsisteme inteligente, cel senzorial (în cazul de față vederea robotului) și cel efector.

În schema din fig.6.2 este ilustrată o astfel de organizare. Ceea ce caracterizează din capul locului această schemă este faptul că robotul primește un obiectiv de lucru (o sarcină robotică) și decide ce trebuie să facă pentru a-l atinge. Rezultatul acestei activități este un plan adică o secvență de acțiuni a cărei înlănțuire e rezultatul unor decizii luate de robot și nicidecum o preprogramare dată de utilizator.

Dacă robotul lucrează autonom el trebuie să aibă un sistem senzorial complex care să-i permită să verifice starea în care se găsește mediul de lucru. În timp ce elaborarea planului pînă la acțiunile elementare este o activitate descendentă, activitatea senzorială este de tip ascendent de la valori prelevate din lumea reală, la interpretare și confruntare cu modelul lumii.

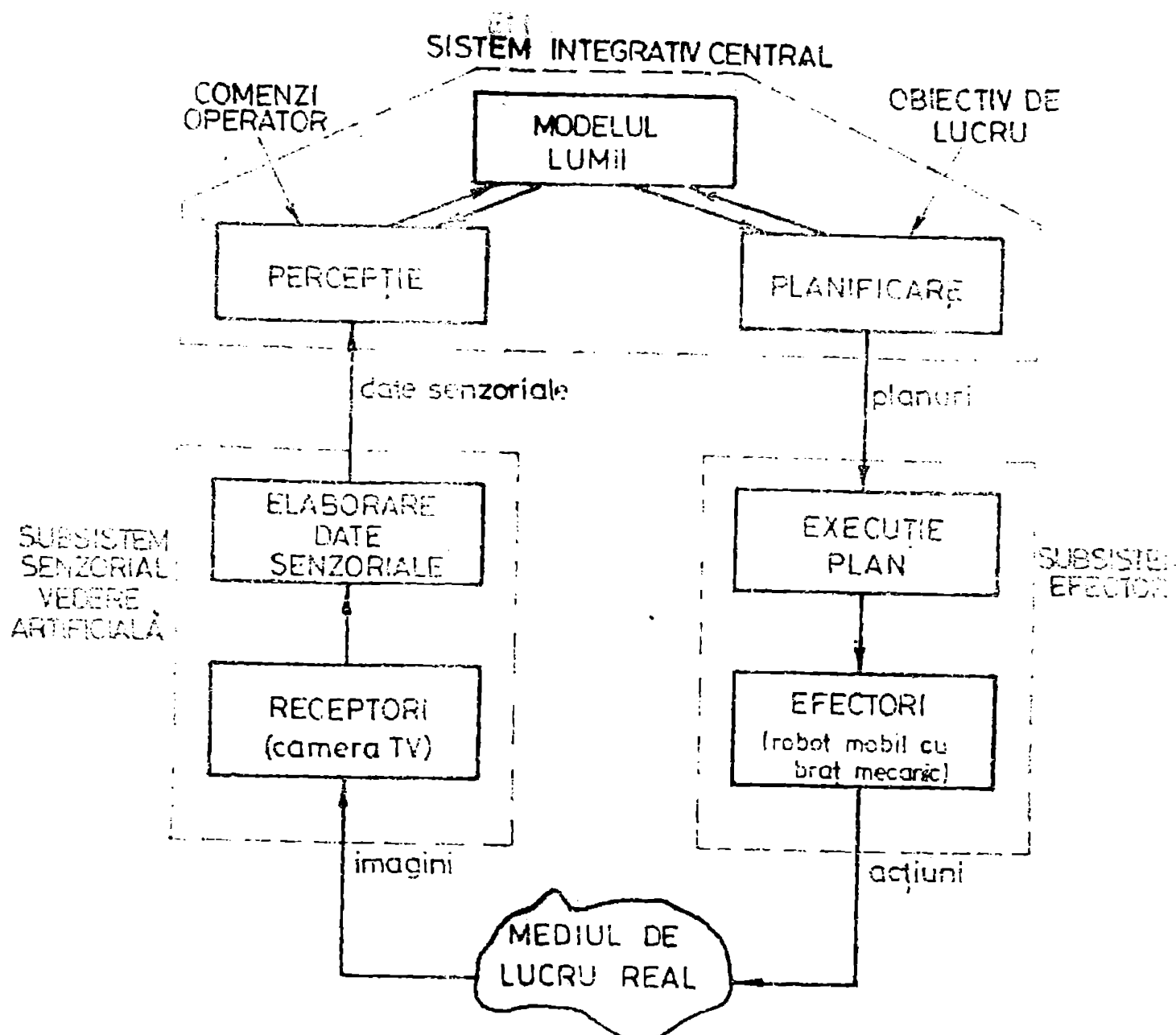


Fig.6.2

6.2. Organizarea ierarhică a căii efectoare

La rândul ei calea efectoare care posedă o ierarhie operațională încreștă poate fi organizată ierarhic folosind de data aceasta conceptual organizării pe verticală. Conceptul furnizează o metodă de partiționare a problemei senzorii în nivele (fig.6.3). Nivelul cel mai înalt formulează obiectivele sau le primește de la operator. Aceste obiective sunt decompuse printr-o succesiune de nivele de calcul în primitive de acțiune care sunt executate de o multitudine de sisteme de acționare aflate pe ultimul nivel.

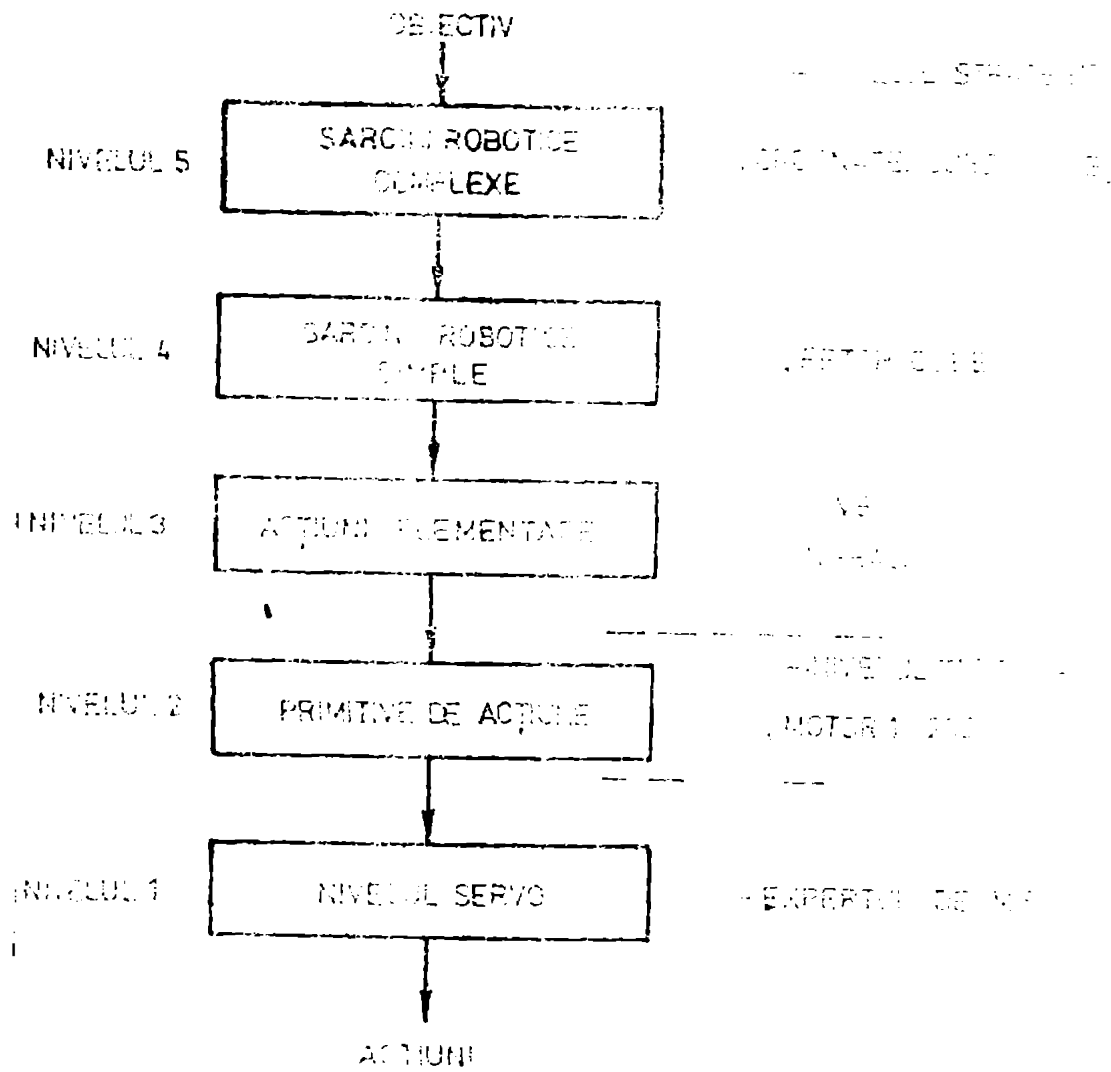


Fig.6.3

Este interesant de prezentat modul în care au fost repartizate aceste nivele în arhitectura de calcul destinată comenzi robotului P.E.T.R.I.C.A. Astfel primele trei nivele, 5, 4 și 3 sunt preluate de programe scrise în limbajul LISP. Ele furnează așa-zitul nivel strategic al comenzi robotului. Nivelul 2 constituie nivelul tactic în care este cuprins și modelul cinematic invers (transformarea de coordonate) și alte programe scrise în FORTRAN și rulate pe calculatorul ROBOTON 4201. Rezultatele obținute sunt transmise expertului de mișcare-EMIS, un program ce rulează pe microcalculatorul înăscut pe robot.

Nivelul strategic va fi descris într-un paragraf viitor, urmind să se prezinte nivelele 2 - tactic și apoi nivelul 1 - expertul de mișcare.

6.2.1. Nivelul tactic

Planul de mișcare formulat de nivelul strategic este elaborat ca o secvență de comenzi MOVE X, Y, Z ; UNGRASP și GRASP. Ele sînt codificate cu cifrele 3, 1 și 2.

În cazul comenzii 1 se apelează subprogramul SB2 care citește parametrul LIM pe care îl compară cu primul element al stării. Pe baza rezultatului comparării se determină numărul de pași și sensul de mișcare pentru motorul 1. al dispozitivului de prehensiune și se reactualizează starea.

În cazul comenzii 2 se apelează subprogramul SB1 care primește parametrii SIZE și FORCE cu ajutorul cărora calculează numărul de pași IDL, pe care îl compară cu primul element al stării INT (1). Pe baza acestui rezultat se calculează numărul de pași pentru motorul 1 iar în final se actualizează starea robotului.

În cazul comenzii 3 se apelează programul DEPL care citește coordonatele X, Y, Z. Se apelează apoi subprogramul de inversare și înmulțire de matrici INA care determină matricea T_4 , după care pe baza relațiilor 4.52, 4.59, 4.62, 4.68 ale modelului cinematic invers se calculează unghiurile și deplasările necesare pentru ca robotul să ajungă în punctul de coordonate carteziene date. Prin apelarea subprogramului NUA se calculează numărul de impulsuri și sensurile pe motoarele ce intervin în mișcare. În final programul DEPL recalculază poziția și starea robotului ca urmare a noii comenzi.

La sfîrșitul programului principal se transmite sistemului MCFM noua stare și noua poziție în coordonate carteziene precum și numărul de pași și sensul fiecărui motor implicat în mișcare.

La primirea rezultatelor se efectuează de către programele LISP conversia numerelor decimale în reprezentare binară pe dublu octet, iar apoi se formează cuvîntul motor pentru robot. Formatul cuvîntului motor este arătat în fig. 6.4. În fiecare pas de lucru robotul P.K.F.F.I.C.A. are nevoie de următoarele informații : deplasările relative ale articulațiilor și sensurile de mișcare. Pentru fiecare motor pas cu pas s-au atribuit doi octeți care exprimă în binar numărul de impulsuri de comandă cu care trebuie acționat motorul respectiv în cadrul

pasului de program (maximă 65536 impulsuri). Pentru cele opt motoare rezultă aşadar 16 octeţi.

Intrucît generarea impulsurilor de comandă pentru APP se face prin program înscriind "1" sau "0" în registrul TACT, în rangul corespunzător motorului dorit, este avantajos ca vectorul de comandă care grupează tactele celor 8 motoare să fie deja pregătit încă din timpul elaborării pasului de program. De asemenea pentru precizarea sensurilor de mişcare a fost prevăzut un octet de SENS. Convenţional s-a stabilit pentru sensul "+" reprezentarea prin "1" logic iar pentru "-" prin "0" logic.

Acest bloc de date completat de nivelul tactic este transmis spre expertul de mişcare.

6.2.2. Expertul de mişcare

Programul expert de mişcare, numit EXMIS, este destinat comenzii efective a robotului. El se află deus într-o memorie EPROM 2708 avînd o lungime de aproape un octet. Acest program testează în primul rînd modul de lucru cerut de operator care poate fi schimbat de la tastatura TO-4 conectată la microcalculatorul robotului. Modul de lucru pune la dispoziţia operatorului două regimuri: manual şi automat.

a) Regimul manual

În această situaţie comenzile sînt date de la tastatura TO-4 putînd fi specificate următoarele operaţii:

- iniţializarea cuvîntului motor prin anularea locaţiilor de memorie aferente
- specificarea motorului care se doreşte a fi acţionat.
- specificarea sensului de mişcare pentru motorul selectat.

În urma acţionării unuia dintre motoare, programul de comandă apelează subprogramul ACTUALIZARE prin care se controlează numărul de paşi executaţi de motoare în locaţiile de memorie corespunzătoare din cuvîntul motor.

Fig.6.4

b) Regimul automat

În cadrul acestui regim comenzile și datele provin de la micro sistemul MCFM prin comenzi desemnate cu câte o literă astfel:

- I (întrerupere) și conduce la întreruperea unei activități de mișcare
- C (continuă) care înseamnă reluarea activității întrerupte prin comanda I
- D (citește starea) prin care robotul P.E.T.F.I.C.A. raportează starea execuției cuvântului motor
- Z (zona) desemnează un transfer din memoria de lucru a robotului P.E.T.F.I.C.A. în memoria calculatorului MCFM
- B (bloc) prin care se transmite din calculatorul MCFM un bloc de date (pot fi unul sau mai multe cuvinte motor) în zona memoriei de lucru a robotului.
- E (execuție) cu această comandă se lansează execuția efectivă a cuvântului sau cuvintelor de motor care constituie programul de mișcare al robotului.

Robotul P.E.T.F.I.C.A. poate trece la efectuarea mișcărilor prevăzute de cuvintele motoare recepționate pe baza algoritmului 3.3 prezentat în cap.3 care realizează mișcări sincrone cu accelerare și decelerare.

În timpul execuției mișcărilor programele LISP așteaptă recepționarea caracterului "A" trimis de PETRICA care înseamnă sfârșitul execuției mișcărilor.

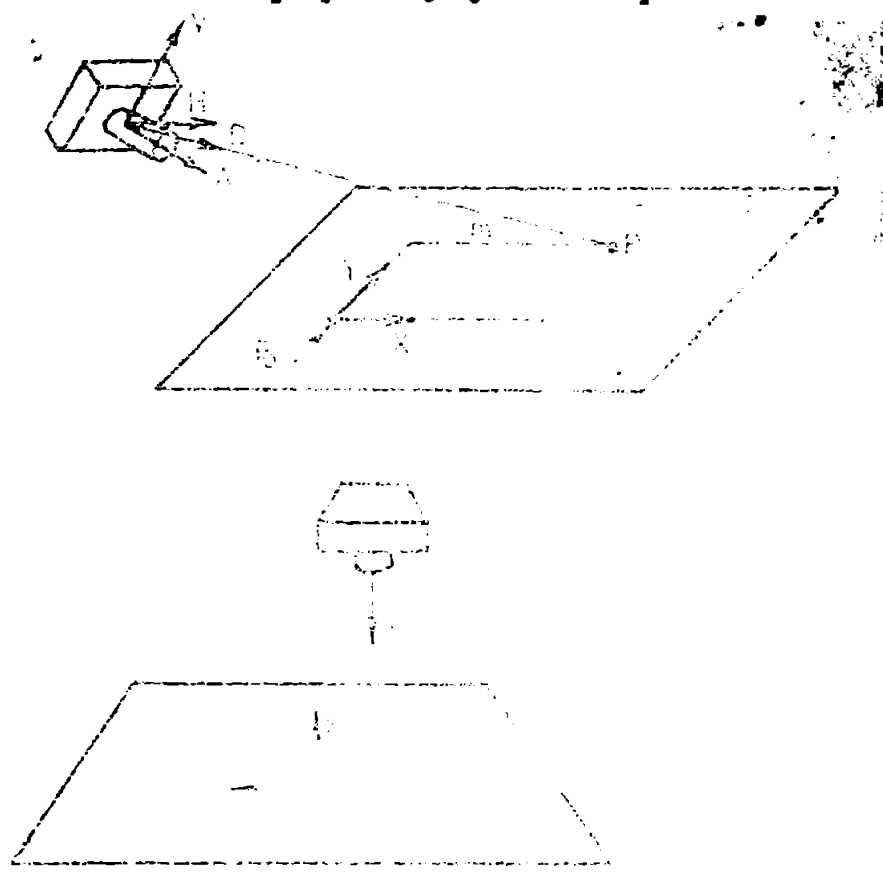
6.8. Integrarea vederii cu manipularea și locomoția

6.8.1. Structurarea mediului de lucru

Pentru un robot echipat cu vedere, relația între lumea reală și lumea bidimensională a imaginii prezintă probleme speciale datorate procesului de preluare a imaginii. Distorsiunile introduse de perspectivă complică problema recunoașterii și, mai grav, complică problema măsurării automate a distanțelor implicit cea a determinării exacte a relațiilor spațiale bazate pe informația vizuală.

Thompson /Th 81/ arată că este nevoie de un model al camerei TV și de transformări de perspectivă, adică relații matematice destul de complicate.

Pentru simplificarea calculului este posibil să se introducă anumite constrângeri asupra mediului de lucru al robotului. În fig.6.5.a se arată o cameră poziționată deasupra unui plan. Dacă se cunoaște poziția planului în spațiul 3D, orice punct P din plan poate fi descris ca $P = P_0 + mX + nY$, unde P_0 este un vector aparținând unui punct din plan, X și Y sunt axele de coordonate definite pe plan, iar m și n sunt parametri folosiți pentru definirea coordonatelor punctului în discuție.



Cel mai mare avantaj ce se poate obține este să se orienteze camera TV așa fel încât planul imaginii să fie paralel cu planul pe care se află

Fig.6.5..

obiectele, așa cum se arată în fig.6.5.b. În acest caz, intrucât vectorul de ațintire A al camerei este ortogonal pe plan, valoarea produsului $D.A.$, calculat pentru orice punct din plan este constantă și egală cu înălțimea la care se află camera TV deasupra planului. Astfel proiecția oricărui punct pe imagine este liniară și nu are apar distorsiuni datorate perspectivei. Actualele sisteme de vedere robotică industriale VS-100 (Machine Intelligence) și Autovision I (Automatic) se bazează pe o astfel de configurație.

Pentru experimentele "ochi-mână-vehicul" s-a ales soluția unei camere TV staționare, montată pe un suport fix și orientată perpendicular pe planul de lucru având în plus axele H și V ale camerei aliniate cu axele planului.

În fig.6.6 este ilustrat mediul de lucru structurat al robotului P.E.T.h.I.C.A. compus din suportul camerei TV, camera TV, zona de lucru de pe sol, corpurile de iluminare.

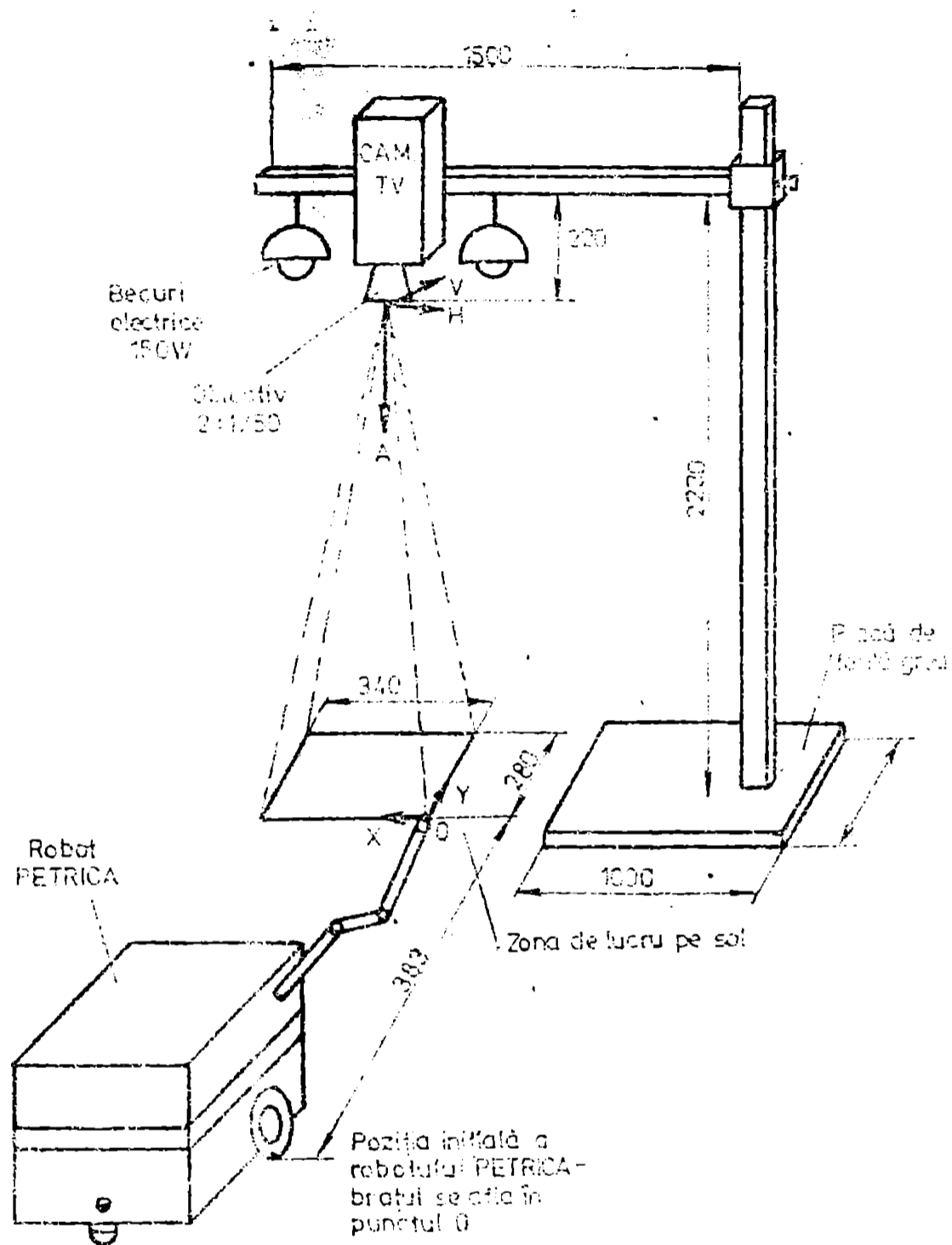


Fig.6.6.

6.3.2. Relații matematice pentru coordonarea vederii cu robotul mobil

Situația care se prezintă diferă de cele cunoscute în literatură, dare prezintă cazurile : 1) cameră și robot staționar 2) cameră montată pe o legătură a brațului și 3) robot mobil și cameră montată pe robot. În cazul de față camera este staționară iar robotul este mobil.

Maș întâi se va defini poziția robotului mobil și a brațului său printr-un produs de trei transformări, conside-

rînd o mișcare pînă într-un punct P_1 , astfel

$$\text{MOVE } P_1 = \text{MOVE } Z \cdot T_4 \cdot \text{TOOL} \quad (6.1)$$

în care

Z - reprezintă poziția robotului mobil relativ la sistemul de coordonate de bază ;

T_4 - reprezintă poziția capătului brațului robotic față de punctul de contact al roții drepte a robotului

TOOL -reprezintă un dispozitiv de prehensiune sau o unealtă atașată capătului brațului.

Printr-o asemenea descriere schimbarea poziției căruciorului va conduce la valori noi pentru matricea Z iar în caz că se schimbă dispozitivul de prehensiune, se va modifica doar TOOL .

Pe de altă parte pozițiile P_1 ale obiectelor sînt obținute pe cale vizuală. Deoarece camera TV se află într-o poziție fixă astfel încît ea poate furniza poziția și orientarea obiectelor sub forma unei transformări omogene față de sistemul de coordonate propriu, numită OB , se poate scrie

$$P = \text{CAM} \cdot OB \quad (6.2)$$

unde CAM este o transformare reprezentînd sistemul de coordonate al camerei TV față de sistemul de coordonate de bază.

Egalînd cele două relații (6.1) și (6.2) se obține :

$$Z \cdot T_4 \cdot \text{TOOL} = \text{CAM} \cdot OB \quad (6.3)$$

Utilizarea acestei relații impune calculul matricii T_4 care furnizează:

$$T_4 = Z^{-1} \cdot \text{CAM} \cdot OB \cdot \text{TOOL}^{-1} \quad (6.4)$$

În membrul drept al acestei ecuații apare Z - poziția căruciorului pe sol, CAM - poziția constantă a camerei TV, OB - poziția și orientarea obiectului furnizate de sistemul de vedere și TOOL - o matrice constantă.

Ecuația (6.4) a fost realizată prin subprogramul DEK - scris în PORTIAN IV . Subprogramul primește ca date inițiale poziția pe sol a robotului PBRICA care se transmite din calculatorul MCFN în calculatorul ROBOTON . Totodată se citesc și coordonatele carteziene X, Y, Z primite de asemenea de la MCFN . (În tabelul 6.1-linia 9). Aceste valori se atribuie elementelor $OB(13)$, $OB(14)$ și $OB(15)$ ale matricii OB (liniile 11, 12 și 13). Apoi se inversează matricile Z și TOOL prin apelarea sub-

programului AINV. Matricea TOOL are următoarea formă înainte de inversare.

$$\text{TOOL} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 125 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (6.5)$$

Pentru simplificarea calibrării s-a ales sistemul de referință de bază chiar în câmpul vizual al camerei. Prin aceasta s-a obținut ca valorile matricii CAM (notată C în program) să fie de forma :

$$C = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (6.6)$$

În liniile 16, 17, 18 se efectuează înmulțirile $Z^{-1} \cdot C$; apoi $Z^{-1} \cdot C \cdot OB$ și în sfârșit $Z^{-1} \cdot C \cdot OB \cdot \text{TOOL}^{-1}$. Rezultatul este transmis subrutinei CAL, care a fost deja prezentată în capitolul 4 ca model cinematic invers al robotului PETRICA. Un nou apel al subrutinei NUA, transmite valorile deplasărilor pentru a fi transformate în număr de pași pentru fiecare motor pas cu pas. În linia 24 se revine la valoarea inițială a matricii Z printr-o nouă inversare a lui Z^{-1} . Finalul programului, liniile 25-37 efectuează următoarele calcule : B(1) - reprezintă rotația pe sol a robotului iar B(2) - translația acestuia în direcția obiectului.

În termenii matricilor A_0^1 și A_1^2 din cap.4 se pot scrie:

$$A_0^1 = \begin{vmatrix} \cos(B(1)) & 0 & -\sin(B(1)) & 0 \\ \sin(B(1)) & 0 & \cos(B(1)) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad A_1^2 = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & B(2) \\ 0 & 0 & 0 & 0 \end{vmatrix} \quad (6.7)$$

Poziția nouă a căruciorului se obține în acest caz din relația :

$$Z_{\text{nou}} = Z_{\text{vechi}} \cdot A_0^1 \cdot A_1^2 \quad (6.8)$$

Calcululele se pot simplifica dacă se face calculul simbolic al produsului $A_0^1 \cdot A_1^2$ după cum urmează:

$$A = A_p^1 \cdot A_l^2 = \begin{pmatrix} \cos(B(1)) & -\sin(B(1)) & 0 & -B(2) \cdot \sin(B(1)) \\ \sin(B(1)) & \cos(B(1)) & 0 & B(2) \cdot \cos(B(1)) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \quad (6.9)$$

Liniile 26 la 34 corespund calculului elementelor matricii A cu formulele de mai sus iar în linia 37 se efectuează produsul matricial $Z_{\text{vechi}} \cdot A$.

6.3.3. Problema calibrării

O problemă, care se întâlnește atunci când se utilizează împreună un robot destinat manipulării și un sistem de vedere robotică, este aceea că cele două sisteme trebuie calibrate între ele într-un mod în care să existe un sistem comun de referință. Ea poate fi abordată după cum urmează Cini /GG 83/ pe diferite nivele de detaliu:

- 1) calibrarea separată a fiecărei componente a sistemului;
- 2) calibrarea conjugată a sistemului de vedere cu cel de manipulare așa fel ca amândouă să utilizeze aceleași unități de măsură;
- 3) determinarea relațiilor între sistemul de referință a camerei TV și cel al robotului;
- 4) utilizarea relației între cele două sisteme pentru determinarea poziției de prindere a obiectelor.

Punctele 3) și 4) au fost deja atinse în precedentul paragraf 6.3.2. Rămân de examinat punctele 1) și 2).

O caracteristică dorită în oricare sistem de vedere și/sau manipulare este posibilitatea sa de a fi calibrat într-o manieră automată ori de câte ori este pus în funcțiune. Dat fiind că acest lucru nu este totdeauna posibil e important să se pună la punct o procedură de tip semiautomat.

Considerând de exemplu problema calibrării sistemului de vedere se constată că a calibra un sistem de vedere înseamnă să se determine factorul de conversie între pixeli și unități reale. Acest lucru permite o corespondență între adresa pixelului și coordonatele fizice ale imaginii, corespondența depinzând de patru cantități: adresa pixelului (linie și coloană) cel mai apropiat de originea sistemului de coordonate $uv(i_0, j_0)$

și rezoluția de eșantionare, adică numărul de pixeli pe unitate de lungime în direcțiile orizontală și verticală, adică n și m . Se poate scrie :

$$u = n(j-j_0) \quad \text{și} \quad v = m(i-i_0) \quad (6.10)$$

Este de dorit ca n și m să aibă aceleași valori ceea ce se traduce imediat prin forma de pătrat a pixelilor în loc de dreptunghi. În acest caz, procesele de recunoaștere sînt simplificate deoarece o siluetă va apare cu aceleași proporții cînd este rotită într-o orientare diferită în planul imaginii.

În cazul de față $n=1,0625$ și $m=0,97$ adică un pixel foarte aproape de forma patratului.

O altă procedură care s-a folosit este introducerea în câmpul vizual al camerei TV a unui disc alb avînd raza de 25 mm. Aria calculată a discului fiind πR^2 adică $1963,49 \text{ mm}^2$.

Această arie a fost măsurată cu ajutorul sistemului de vedere rezultînd valoarea de 1905.

Raportul între cele două valori reprezintă factorul de conversie:

$$F_{\text{conv}} = \frac{1963,49}{1905} = 1,0306$$

Practic s-a lucrat cu o valoare a lui $F_{\text{conv}} = 1$.

Această calibrare poate fi făcută semiautomat în sensul că utilizatorul trebuie trebuie doar să arate obiectul de calibrare și să dea instrucțiunea de calibrare.

Calibrarea sistemului de vedere trebuie repetată doar în cazul schimbării poziției camerei TV în raport cu planul de eprijă al obiectelor de examinat.

În ce privește calibrarea sistemului format din vehiculul și brațul robotului P.E.F.E.I.C.A., aceasta a fost necesară pentru fiecare grad de libertate. Factorii de conversie pentru fiecare grad de libertate au fost determinați prin măsurători astfel: 9450 pentru rotația pe sol, 21 pentru translație, 24 pentru articulația de cot și 24 pentru balansul prehensiunii. Aceste valori reprezintă numărul de pași pentru ΔPP pentru 1 radian (rotație), respectiv pentru 1 mm pe translație, și în fine 1 mm pe șuruburile conducătoare.

Odată calibrat de către constructor, utilizatorul nu mai

trebuie să execute nici o calibrare a robotului cu excepția cazurilor de manipulare în care trebuie ocupată o poziție inițială atât pentru vehiculul cît și pentru brațe. În fig.5.6 este arătată această poziție inițială în care robotul se află cu amîndouă roțile la 383 mm de punctul de origine 0 al sistemului de referință de bază, rotația θ_1 este nulă și capătul efector este poziționat pe punctul 0, adică X, Y, Z sînt nule.

6.4. Nivelul strategic de comandă

Alegerea limbajului potrivit pentru un sistem de comandă robotic este foarte importantă. Orice lucru pe care robotul trebuie să-l execute, să-l știe sau să-l simtă (prin senzori) va avea o formă de exprimare în limbajul ales. Limbajul de asamblare este ieftin dar dificil de elaborat, înțeles sau modificat. BASIC-ul este larg accesibil, ușor disponibil, ușor de învățat și interactiv. Un dezavantaj serios îl constituie nemodularitatea întrucît toate variabilele sînt globale pe de o parte și de alta prin necesitatea de a ramifica (la adrese nemnemonic) pentru a controla secvența programului.

Programele de I.A. sînt prezentate în majoritate în LISP. Acest limbaj, inventat în anii '50 este în mod specific destinat manipulării de simboluri. LISP-ul este puternic și ușor de folosit. Este puternic deoarece permite exprimarea unor algoritmi de nivel înalt în mod concis și ușor de folosit și pentru că este un limbaj interpretat. LISP-ul permite utilizatorului să întrerupă programul în orice moment, să examineze valorile variabilelor, stiva și să modifice ceva.

Un alt avantaj esențial al LISP-ului este capacitatea de a defini alte limbaje.

Pentru experimentele cu P.E.T.K.I.C.A. s-a elaborat și testat un interpretor LISP care poate rula pe un microprocesor 8080, respectiv în calculatorul LCRN.

Strategia de comandă descrisă a fost proiectată să posede o generalitate cît mai mare și să permită realizarea de sarcini cît mai variate. Scopul urmărit este să se poate efectua un număr de acțiuni elementare ca mișcări, memorarea pozițiilor brațului, a pozițiilor obiectelor toate în cadrul unor sarcini precise fixate de utilizator.

6.4.1. Nivelul de comandă al acțiunilor elementare

Pe nivelul cel mai scăzut al strategiei de comandă a robotului se află rutinele de mișcare. Acestea se referă la trei tipuri de mișcări : o comandă de mișcare vehicul plus braț (în coordonate carteziene X, Y, Z), o comandă de apucare de lăsare a obiectului. Fiecare din comenzi este transmisă sub formă LISP spre calculatorul ROBOTON care prin programele FORTRAN calculează numărul de pași și sensurile de mișcare.

Se va lua ca exemplu comanda MOVE X, Y, Z. Ea se codifică prin cifra "3" urmată de coordonatele X, Y, Z.

```
3 x x x x x . y y y y y , z z z z z , CR
```

În sensul spre ROBOTON, trimiterea mesajului de mai sus se face prin "prin_1" (LISP) și recepția se face cu "READ" (FORTRAN) iar în sens invers cu "WRITE" (FORTRAN) și "read 1" (LISP).

Pentru transformarea de coordonate și calculul deplasărilor în număr de pași s-a definit o funcție numită "transf".

```
(defun transf (listă coord) ; transmite valorile coord.  
  (progn (prin_1 "3") ; carteziene  
  (prlist(listă coord))) ; trimite "3" fără CR  
(defun prlist (listă coord) ; transmite elementele listei  
  (cond ; cu "." între ele dacă  
  ((eql listăcoord nil)(terpri)); lista e vidă ieșire  
  (t(prin_1(car listăcoord)) ; <CR> ieșire element  
  (prin_1 ",") ; revenire pentru restul  
  (prlist (cdr listă coord))))); listei
```

Pentru comenzi GRASP și UNGRASP există două funcții asemănătoare deosebirile constând în numele funcțiilor, "apucă" și "desfă", în înlocuirea lui "3" cu "2" sau "1" iar "listă coord" cu "mărime-forță" sau "deschidere"

```
(defun apucă (mărime-forță) ( ) ...
```

și

```
(defun desfă (deschidere) ( ) ...
```

În privința recepționării rezultatelor furnizate de modelul cinematic implementat în ROBOTON, rutina LISP corespunzătoare se prezintă astfel:

```
(defun cit-depl ( ) ; returnează lista deplasărilor
(prog1 (listă-pași) ; pe articulații
  (list (carnum(read 1)) ; deplasare articulație 1
        (carnum(read 1)) ; deplasare articulație 2
        (carnum(read 1)) ; deplasare articulație 3
        (carnum(read 1)) ; deplasare articulație 4
        (carnum(read 1)))) ; deplasare mână
```

Funcția (carnum x) transformă o secvență de cifre într-un număr astfel:

```
(defun carnum (n) ; returnează întregul n,
  (prog (clista x) ; este deja un număr
    (cond((numberp n)(return n)) ; întreg
      (setq clistă explode n)) ; face lista de caractere
      (setq x 0) ; inițializează val.x
      (setq clistă (cdr clistă)) ; elimină
    loop :
      (setq x(* x 10) ; deplasare cu o cifră s
        (setq x(+ x(car clistă))) ; adună cifra
        (setq clistă (cdr clistă)) ; elimină primul elem.
        (cond((null clistă)(return x)) ; sfârșit returnează rez.
          (t(go loop)))) ; dacă nu ciclează
```

Funcția "fapași" transmite robotului P.B.T.F.I.C.A. în binar blocul de date cu structura din fig.6.4.

```
(defun fapași (lispași) ; transmite lista deplasărilor
  (setq lispași(citdepl)t) ; sub forma blocului din
  ; fig.6.4
```

En urmează să fie utilizată ori de câte ori se dau comenzi de mișcare spre robot, indiferent dacă e vorba de "Transf" "apucă" sau "desfă".

```
(defun move (loc-nou)
  (transf (listă-coord) ( ))
  (setq lispași (citdepl))
  (fapași lispași)
  (setq x aici (listă-coord) return t)
```

Variabila „x aici” este o variabilă globală care este accesibilă în întreg programul. Ori de câte ori poziția robotului este modificată, se va modifica și valoarea variabilei "x aici".

Pentru funcții ca "grasp" și "ungrasp" s-au definit corpuri de funcții asemănătoare.

```
(defun grasp(mărime-forță)      (defun ungrasp(deschidere)
  (apucă (mărime-forță) ( ))    (desfă(deschidere) ( ))
  (setq lispasi(citdepl))       (setq lispasi (citdepl))
  (fapasi lispasi) return t)    (fapasi lispasi) return t)
```

6.4.2. Nivelul sarcinilor robotice simple

Odată definite comenzile simple la nivelul acțiunilor elementare este necesară completarea lor cu posibilitatea de a face referință la diferite locații din spațiul de lucru al robotului. Eventual ar fi util să se facă referință prin nume iar vectorul de poziție (cartezian) să fie automat extras dintr-o listă globală care cuprinde mai multe astfel de locații. Soluția în această problemă o constituie lista de tipul "a" (asociativă) de felul celei de mai jos:

```
((loc-a (10, 150, 0))
 (deasupra-a(10, 150, 60))
 (loc-b(73, 25, 0)))
```

Pentru ca accesul la listă să se poată face de către orice rutină lista va fi considerată ca valoarea variabilei globale "locații". În acest moment este nevoie de o funcție care să extragă valoarea vectorului de poziție odată ce se dă numele locației. Funcția (locație 'x') returnează vectorul de poziție asociat lui x iar în caz de inexistență dă mesaj de eroare :

```
(defun locație (numeloc)
  (prog(vector)
    (cond((setq vector)
      (car(cdr(assoc numeloc "locații"))))
      (return vector)
      (t(error locație 'nu'este' o' locație))))))
```

În acest moment se poate scrie comandă de genul (move(locație 'x')). Pentru a simplifica această funcție des folosită se poate scrie o nouă funcție "mişcă"

```
(defun mișcă (olocație)
  (move(locație olocație)))
```

Asemănător locațiilor se pot manipula și obiectele în cadrul unei a-liste denumită "x obiecte". O astfel de listă ar putea avea forma de mai jos:

```
((cub -a((120, 30, 25)(120, 30, 60)))  
(cub-b((72, 131, 25)(72, 131, 60))))
```

Dacă se cunoaște poziția unui obiect și se dorește calculul unei poziții deasupra obiectului x acest lucru se realizează prin însumarea unui vector constant care asigură o deplasare pe Z .

```
(defun deasupra (x)  
  (addvec x  
    '( 0 0 35 )))
```

Se poate defini acum o nouă funcție care pornind de la numele obiectului să realizeze apucarea acestuia printr-o deplasare deasupra obiectului, deschiderea mâinii, coborîrea spre obiect și prinderea acestuia:

```
(defun apucă (obi)  
  (prog (deas peobi deasobi)  
    (cond (x holding  
      (error'(apucă eșec)' holding  
        x holding)))  
    (setq deas (du-te obi))  
    (setq peobi (car deas))  
    (move (open deasobi 150))  
    (move (open peobi 150))  
    (move peobi)  
    (setq x holding obj)))
```

În continuare este nevoie de o funcție care să realizeze lăsarea obiectului într-un loc anumit:

```
(defun lasă ( )  
  (cond((null x holding)  
    (error'(lăsarea eșec-nimic  
      apucst)))  
    (t(scoate obi x holding)  
      (setq x obiecte  
        (cons(list x holding  
          (list x here  
            (deasupra x here))))))  
      (setq x holding nil)  
      (move (open x here 150))))
```

Astfel comanda de mișcare a unui obiect într-un anumit loc devine simplă de exprimat, dacă se dă un loc₁ și un

obiect "o" aflat într-o poziție curentă "p".

(defun mută (obi unloc)

(prog (vector)

setq vector (locație unloc)

(apucă obi)

; ia obiectul

(move (deasupra x here))

; ridică obiectul

(move (deasupra vector))

; deasupra un loc

(move vector)

; poziționează obiect

(lasă)

; eliberează obiect

(move (deasupra vector)))

; ridică mîna

6.4.3. Nivelul sarcinilor robotice complexe

Avînd la dispoziție funcția robotică puternică definită în paragraful anterior este posibilă abordarea unor sarcini complexe cum ar fi ordonarea unor cuburi aflate într-o așezare aleatoare. Inițial se dă comanda "privește-lune" care este destinată sistemului de vedere.

Rezultatele obținute de acesta privind numele obiectelor și pozițiile lor vor fi cuprinse în lista obiectelor.

Dependent de comanda recepționată care poate fi de exemplu:

(ordonează ((cub-a cub-b cub-c) un loc (dreapta)) robotul va plasa cubul "A" în poziția "unloc" apoi va calcula un loc pentru cubul "B" la dreapta lui "A" și la fel pentru cubul "C".

Dacă locurile sînt ocupate indiferent că e locul pentru cubul A, B sau C este invocată o procedură de tipul "găsește loc". În cadrul acesteia se caută pe o traiectorie în spirală în jurul locului ocupat o poziție nouă pentru obiectul care ocupă locul, apoi se declanșează automat mutarea obiectului în noul loc.

Înafara sarcinii robotice de ordonare s-au mai definit funcții de "stivuire" și de "soriere" cu ajutorul literelor marcate pe cuburi.

6.5. Experimente realizate

6.5.1. Experimente de programare prin învățare

Conceptul de programare prin învățare este esențial în robotică și constituie o caracteristică majoră a roboților industriali. Deși problemele legate de realizarea programării prin

învățare sînt pe deplin clarificate s-a considerat, în momentul terminării construcției robotului P.E.F.I.C.A. (anul 1980) că desfășurarea de experimente fundamentale de robotică este necesară în vederea familiarizării cu conceptele de bază și ca primă evaluare a posibilităților robotului. Programarea prin învățare (teach-in) se poate înțelege din fig.6.7 în care sînt ilustrate procesul de învățare (linie continuă) și procesul de redare (linie întreruptă). În etapa de învățare operatorul uman comandă mișcările robotului de la pupitrul de învățare (tastatura IO-4).



Fig.6.7

Dacă mișcările comandate corespund activității propuse, operatorul decide memorarea lor. Pas cu pas astfel de mișcări executate sub comanda și observația vizuală directă a operatorului sînt incluse în viitorul program de lucru. Odată terminată instruirea se poate trece la verificarea și apoi la redarea automată a programului de lucru. Pe larg, programul SCOPAK- care realizează ambele faze învățare-redare, a fost descris în cap.7 al cărții "Microprocesorul BOBO în aplicații" /MS 81/.

Cu ajutorul acestui program s-au elaborat un număr de experimente printre care construcții din cuburi aranjate într-o varietate de configurații, mutarea unor piese pe tabla de șah, modelarea în laborator a programului unui robot industrial care servește o celulă de fabricație cu patru mașini-unelte (pentru aplicația robotului industrial KMF-1).

6.5.2. Experimente "ochi-mînă-vehicul"

Intr-o fază mai avansată (anul 1983) s-a putut trece la desfășurarea de experimente "ochi-mînă-cărucior", ocazie prin care s-au verificat toate elementele atât teoretice cît și practice ale sistemului de comandă a robotului P.E.F.I.C.A. În fotografia 6.1 este redat aspectul ansamblului "ochi-mînă-vehicul" din laboratorul de Echipamente perife-

rice al Catedrei de Automatică din Institutul Politehnic
"Traian Vuia" Timișoara. Din imagine lipsește doar minisiste-

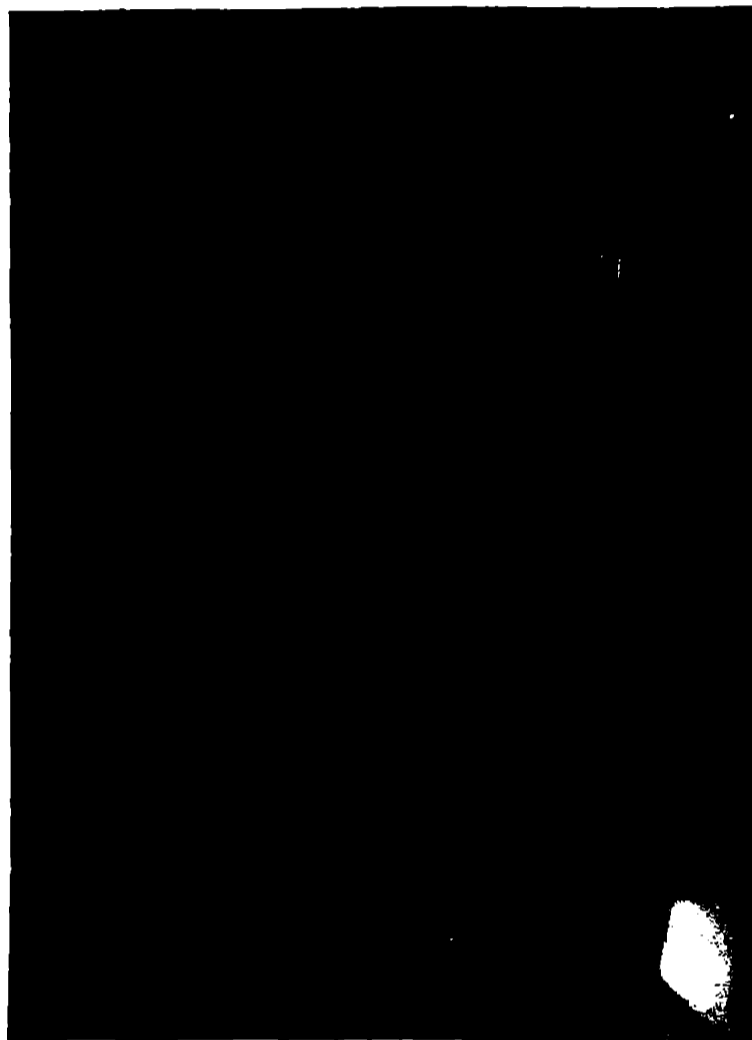


FOTO 6.1.

mul de cal-
culi ROBOTHON
4201 aflat
într-o sală
alăturată.

Succe-
siunea opera-
țiilor prin-
cipale efec-
tuate pentru
ca obiectele
din mediul de
lucru să fie
recunoscute,
poziție și
orientarea
lor să fie
determinată
iar manipu-
lările să
aibă loc

conform programelor nivelului strategic este următoarea :

- 1) Interpretorul LTSP analizează obiectivul cerut de utiliza-
tor și cere imediat o imagine asupra mediului de lucru
(instrucția WATCH) ca urmare, contex-
tul integral al interpretorului LTSP este salvat pe discul
flexibil pentru a face loc imaginii în memoria LCM.
- 2) Camera TV și digitizorul sub comanda LCM și a programului
de achiziție preiau o imagine care este depusă în memoria
microsistemului LCM și apoi transmisă în blocuri de câte un
1 octet spre calculatorul ROBOTHON care depune imaginea pe
bandă magnetică.

Achiziția imaginii și transferul se face în două etape,
primele 160 de linii TV și următoarele 128 de linii.

- 3) În calculatorul ROBOTHON este lansat programul de analiză
de imagini (prezentat la paragraful 5.4) care determină po-
ziția obiectelor și un număr de invarianți.
- 4a) dacă se efectuează învățarea unui obiect sau a unei clase
de obiecte programul de învățare construiește baza de date

pentru obiectul analizat

- 4b) dacă se efectuează o recunoaștere a cadrului, programul de recunoaștere determină identitatea obiectelor detectate în imagine construind o listă a acestora.
- 5) rezultatele sînt transferate în MCM, unde sînt analizate de programele LTSP în vederea stabilirii operațiilor următoare.
- 6) nivelul strategic de comandă elaborează planul de activitate a robotului care se compune dintr-o secvență de instrucțiuni MOVE X, Y, Z , GRASP și UNGRASP
- 7) una cîte una, aceste instrucțiuni sînt transmise spre calculatorul ROBOTION împreună cu starea robotului și poziția pe sol (Z). Programele nivelului tactic calculează deplasările pe axe și transmite calculatorului MCM rezultatele împreună cu noua stare a robotului și poziția nouă pe sol (Z) (programele MISC, CAL, DEPL, OPT1 și OPT2)
- 8) în calculatorul MCM programe speciale aflate sub comanda programelor LTSP, execută conversia zecimal-binar a datelor de mișcare și întocmesc structuri de date pentru cuvîntul motor ca în fig.6.4.
- 9) cuvintele motor sînt transmise în memoria RAM a robotului P.E.F.I.T.C.A.
- 10) se dă comanda de către MCM a începerii execuției mișcărilor de către robotul P.E.F.I.T.C.A.
- 11) terminarea mișcărilor este semnalată calculatorului MCM care rein operațiile de la punctul 7.

Toate operațiile de mai sus sînt coordonate de programele LTSP și subrutinele asociate acestora precum și denucleul sistemului de operare de pe MCM. Pentru fiecare fază există rapoarte care apar pe ecranul DAF-ului (MCM) astfel cît operatorul poate urmări activitatea întregului ansamblu.

Obiectele manipulate de robot au fost cuburi cu latura de 47 mm acoperite cu negru și avînd pe fiecare față o literă albă astfel încît contrastul să fie cît mai bun. S-au folosit diferite tipuri de iluminări printre care: lumină electrică de la becuri cu incandescentă (2 x 150 W), lumină fluorescentă ambientală obișnuită a laboratorului, lumină de zi și combinații ale acestora. Algoritmul propus la paragraful 5.4 făcînd parte din procedura de recunoaștere a cuburilor marcate cu litere s-au dovedit robuști și eficienți. Majoritatea experimentelor

privind vederea robotică au avut drept scop analiza influenței formei ferestrei de studiu (dreptunghiulară sau patrată cu diferite dimensiuni); pe de o parte și influența rezoluției imaginii digitizate asupra recunoașterii obiectelor.

În cazul rezoluției spațiale de 60 x 285 s-a demonstrat că variabilitatea caracteristicilor CM02P, CM20P și APLI a fost așa de mare cu schimbarea orientării încât s-a renunțat complet la această rezoluție în favoarea celei de 320 x 285 pixeli.

Durata unei achiziții și analize complete pentru 4 cuburi maximum a fost de 13 minute. În covârșitoare parte acest timp este necesar achiziției care folosește benzi magnetice IZOT cu 200 bpi. Dacă FORTIAN-ul de pe ROBOTON ar fi avut instrucții pentru comanda discurilor cartridge acest timp s-ar fi redus probabil de 10 ori. Din aceste 13 de minute:

- 60 de secunde aparțin digitizării imaginii
- 600 de secunde aparțin transferului imaginii din MCM în ROBOTON
- 120 de secunde, prelucrarea efectivă pentru recunoaștere și localizare.

În experiențele efectuate cu robotul P.E.T.H.I.C.A. s-au pus în evidență principalele cauze ale erorilor care afectează precizia de poziționare :

Acestea sînt: jocuri în cuplele roților și a reductoarelor, devierea produsă în rotația căruciorului de către roata pasivă de tip castor.

Ele au fost remediate prin : 1) înlocuirea roții castor cu o bilă pe rulmenți cu posibilitatea de a prelua mișcări în toate direcțiile ; 2) prin reglaje mecanice s-au eliminat, eventual redus, jocurile introduse de reductoare pentru roțile active. Jocul cauzat de către un reductor era de 2 mm. Compensarea acestuia s-a făcut prin programele de mișcare ținîndu-se cont de sensurile de deplasare ale roții.

O problemă care nu a putut fi rezolvată este cea a erorilor necompensate care se produc din cauza dificultăților de tracțiune a roților față de podea. Erorile de tracțiune sînt inerente în orice situație, fie o podea acoperită, fie neacoperită cu variații mici de înălțime. Erorile au fost puse în evidență cînd i s-a dat robotului să manipuleze, în mod repe-

tat, cuburile aflate în colțurile unui patrat cu latura de 500 mm. Eroarea capătului efector a fost de maximum 15 mm după un număr de 30 de mișcări. Din cei 15 mm, 3 mm se datorează erorii de determinare a centrului de greutate de către sistemul de vedere.

Toate acestea se datorează faptului că R.C.I.L.I.C.A. este un sistem robotic funcționând în buclă -deschisă. Inadecvarea buclei deschise, fără reacția vizuală sau alți senzori, este evidentă în următorul experiment de programare prin învățare. Se pornește robotul dintr-un punct X al camerei pînă în punctul Y cu unele schimbări de direcție pe parcurs. Apoi tot în modul de programare prin învățare mișcării robotul înapoi în punctul X. Există în acest moment un program de mișcare a robotului. Din nefericire secvența de mișcări are un foarte slab cuplaj cu realitatea. O eroare de plasare a corpului robotului de cîteva grade în punctul de start, se poate transforma într-o eroare de cîteva zeci de grade eventual zeci de mm după un traseu de o oarecare complexitate. Soluția acestei probleme este îmbunătățirea performanțelor navigaționale, astfel încît să se sesizeze poziția absolută în cadrul geometriei spațiului de evoluție. Există mai multe căi de obținere a acestor date: lumini de referință în cameră unde e plasat robotul, dispozitive sonor dacă există pereți și telemetru laser. Desigur nu trebuie uitate nici posibilitățile vederii robotice. În cazul de față folosind o cameră staționară și un robot mobil există posibilitatea recalibrării robotului sau chiar corectarea poziției capătului efector cu condiția ca mina robotului să fie prezentă în câmpul vizual.

Cu ocazia experimentelor s-a pus în evidență că aparatul matematic a fost corect și nu a produs erori de nici un fel.

Pentru o experiență "ochi-mîni-vehicul", avînd elemente de inteligență artificială, constînd din aranjarea în diferite moduri a cuburilor poziționate inițial aleator s-a obținut un timp total de 30 de minute: 15 minute pentru achiziția și prelucrarea imaginii, 15 minute pentru efectuarea mișcărilor.

CAPITOLUL 7

C O N C L U Z I I F I N A L E

Obiectivul fundamental urmărit în cuprinsul acestei lucrări este acela de a contribui cât mai eficient și original la conceperea, realizarea și experimentarea de sisteme robotice inteligente de tipul "ochi-mână-vehicul", având în vedere studiul atins pe plan mondial dar și nivelul și necesitățile tehnicii robotice românești.

Investigațiile s-au desfășurat în următoarele direcții principale:

i) Analiza evoluției sistemelor robotice inteligente din categoriile "ochi-mână", "ochi-vehicul" și "ochi-mână-vehicul" pe plan mondial între anii 1960-1983;

ii) Desfășurarea cercetărilor sub forma unui proiect amplu intitulat Proiectul Experimental Timișorean de Robot Inteligent pentru Cercetare Avansată - prescurtat P.E.T.R.I.C.A. - care să se dezvolte prin acumulări succesive pe parcursul câtorva ani (1979-1985);

iii) Imprimarea unui puternic caracter experimental cercetărilor întreprinse considerând că efectuarea de experimente-cu roboți compleți (capabili și de manipulare și de locomoție) - are o importanță covârșitoare în robotică;

iiii) Elaborarea sub raport teoretic și aplicativ a unor soluții noi de conducere inteligentă a roboților mobili de manipulare dotați cu vedere artificială.

Principalele contribuții originale ale autorului, prezentate pe capitole, constau în următoarele :

a) Capitolul 2

- evidențierea a două mari etape în dezvoltarea sistemelor robotice inteligente (1960-1973 și 1974-1980);
- extragerea unor caracteristici și performanțe, selectarea și sistematizarea principalelor resurse mecanice, senzoriale de echipament și programat de calcul pentru 20 de sisteme robotice inteligente;
- desprinderea principalelor subdomenii în care se desfășoară lucrările de bază, acestea fiind: Reprezentarea și modelare ,sen-

sori, Manipulare, Locomoție, Suprastructură inteligentă, Integrare și aplicații ;

- punerea în evidență a numărului redus de roboți compleți existenți în laboratoare și dirijarea cercetărilor spre problematica aflată la joncțiunea dintre manipulare și locomoție;
- analiza însestrării sensoriale a sistemelor robotice inteligente studiate a arătat că vederea robotică (VR) este cel mai puternic senzor robotic;

b) Capitolul 3

- elaborarea unei concepții generale asupra manierei de abordare bazată pe ideile de proiect, caracter experimental, inteligență robotică și cercetare avansată;
- proiectarea, construcția și experimentarea robotului mobil cu manipulator P.E.T.h.I.C.A. conceput în vederea unei game largi de cercetări în robotică având ca principale caracteristici:
1) lipsa de specific (nespecializat pentru o sarcină sau un mediu) 2) mobilitatea (limitată omnidirecțională); 3) însestrarea cu manipulator (posibil două) 4) comandă cu microcalculator îmbarcat 5) semiautonom sau autonom prin echipare cu acumulatori și legătură radio;
- alegerea unei soluții cinematice originale pentru manipulator de tipul: pivotare- rotație- rotație- pivotare- rotație;
- adoptarea unor dimensiuni suficiente ale manipulatorului pentru a furniza o paletă de probleme de comandă similare cu cele întâlnite pentru manipulatoarele de sarcini ușoare din industrie;
- obținerea unui braț robotic la un preț foarte scăzut folosind componente și materiale existente;
- introducerea pentru trei din cele cinci grade de libertate a mecanismului cu șurub conducător și piuliță cu avantajul că pot dezvolta forțe ridicate, asigură autoblocarea și produc mișcări line și precise;
- înlocuirea roților castor ale vehiculului care prezintă dezavantaje cu o bilă pe rulmenți;
- proiectarea, construcția și experimentarea unui sistem de comandă cu microcalculator îmbarcat pe robot, având caracteristicile unui microsistem de dezvoltare care prin programe corespunzătoare a asigurat o autonomie completă apoi a devenit expert de mișcare;
- elaborarea unei metode de calcul a timpilor de accelerare și

decelerare pentru acționări cu motoare pas cu pas precum și a numărului de pași de accelerare și decelerare;

- definirea unei metode și algoritmi noi pentru comanda prin accelerare și decelerare a MPP în condițiile distribuirii impulsurilor de comandă a MPP astfel ca mișcările diferitelor articulații să fie sincrone - constituind o îmbunătățire a algoritmului elaborat de J.W.Hill și prezentate de G. M. Gini;
- conceperea, realizarea și utilizarea unui sistem de conducere bazat pe o arhitectură de calcul ierarhizată pe trei nivele - unu mini și două micro sisteme - în care cea mai puternică resursă de calcul nu joacă rolul de master acesta revenind nivelului intermediar, soluție ce rezolvă problema unui robot ce se poate degrada în mod satisfăcător;

c) Capitolul 4

- sintetizarea principalelor cercetări privind modelele cinematice de ordinul zero;
- formularea algoritmică a problemei atașării coordonatelor Denavit Hartenberg unor cuple cinematice;
- extinderea aplicării coordonatelor D-H la axele cinematice instantanee ale părții de vehicul a unui robot;
- utilizarea gradelor de libertate oferite de manipulator și vehicul împreună în vederea manipulării;
- elaborarea de modele matematice pentru cinematica manipulatorului și vehiculului tratate unitar;
- introducerea matricii constante A_2^{BKAT} (relația 4.18) prin schimbarea căreia se poate trece matematic de la un braț la altul în cazul colaborării a două brațe;
- elaborarea de programe care să permită verificarea experimentală a modelelor teoretice;

d) Capitolul 5

- elaborarea programului pentru conversia și achiziția imaginilor TV cu 320×285 pixeli și 16 nivele de gri;
- studiul a 8 nuclee diferite pentru operatorul de filtrare prin convoluție 3×3 ;
- studierea și elaborarea de programe pentru toți operatorii de extragere a conturului existenți în literatură urmate de analiza comparativă sub raport calitativ și al consumului de timp de calcul;

- elaborarea unui sistem de vedere robotică tip 2D pentru imagini binare destinat recunoașterii și localizării pieselor industriale mici ce nu se scoperă având unele trăsături originale sau sint : algoritmul UC (de urmărirea a conturului) bazat pe conectivitate 6, analiza incluziunii prin două metode, calcul on-line a 7 sume parțiale ce permit evaluarea a lo caracteristicii per componentă de imagine, recunoaștere secvențială și învățare supervizată;
- elaborarea unui al doilea sistem de vedere robotică pentru imagini monocrome, în întregime original, destinat recunoașterii cuburilor marcate cu litere folosind : tehnica ferestrelor, a extragerii conturului prin operator Sobel sau detecție lumină-întuneric, a recunoașterii după metoda celui mai apropiat k-vecin. Se demonstrează astfel posibilitatea dezvoltării de sisteme VH configurate pe aplicație apte de a funcționa în timp real cu resurse de calcul micro;
- experimentarea ambelor sisteme VH în condiții diferite de iluminare în vederea testării robusteții algoritmilor.

e) Capitolul 6

- constituirea unui mediu al sistemului de dezvoltare robotică prin integrarea resurselor robotice cu ajutorul unui sistem de operare independent de limbaj ale cărui dimensiuni au fost păstrate la proporții rezonabile ;
- crearea unui sistem robotic inteligent original compus dintr-un robot mobil cu manipulator și o cameră TV stacionară;
- soluționarea problemelor legate de integrarea vederii cu manipularea și calibrarea sistemului robotic de mai sus;
- definirea unor proceduri noi pentru nivelul strategic pe baza cărora se poate realiza activitatea de planificare a sarcinilor robotice complexe;
- evidențierea necesității calculării poziției robotului pe sol după fiecare manipulare întrucât în relația 6.1 matricea Z nu mai este constantă ;
- posibilitatea simplificării calculelor de coordonare a vederii cu manipularea prin alegerea sistemului de referință de bază chiar în câmpul vizual al camerei, prin care matricea CAM (rel.6.2) devine unitară.

Valoarea aplicativă și direcții de dezvoltare viitoare

Pe baza rezultatelor teoretice și îndecesebi practice obținute în prezenta lucrare și în cursul elaborării ei, autorul a proiectat, în cadrul colectivului de Roboți Industriali de la I. Politehnic "Traian Vuia", sistemul de conducere bazat pe calculatorul de proces ECAKOM-800 al robotului industrial „RMI-1”, programele de aplicație și punerea la punct a primei celule robotizate din România la întreprinderea Electromotor-Mișoara. Lucrarea a fost răsplătită cu premiul "Traian Vuia" al Academiei R.S. România pe anul 1963.

Unele aspecte ale acestei teze au fost valorificate în cadrul a 8 lucrări științifice din care două au fost citate de prof. M. Drăgănescu /Dr 80/. Robotul P.E.T.h.I.C.A. a fost prezentat în cartea "Microprocesorul 8080 în aplicații", Editura Iași 1981 (capitolele 3 și 7).

Autorul a mai participat la 4 contracte de cercetare între care robotul RMI-12,5 construit de "Electrotimig". Deasemenea a fost consultant științific la trei filme de scurt-metraj ale Studioului "Alexandru Sahia" pe teme de robotică. Filmul "Roboți industriali, prezintă între altele și experimentele "ochi-mână-vehicul" realizate cu robotul P.E.T.h.I.C.A.

Pentru viitor se întrevăd două direcții de dezvoltare bazate pe rezultatele din această lucrare: prima se referă la dezvoltarea de sisteme robotice industriale inteligente cu vedere, a doua la roboți cu manipulare și locomoție dotați cu vedere și destinați lucrărilor de operare, întreținere și reparare în centralele electrice nucleare.

B I B L I O G R A F I E

1. Ag 77 Agin G., Vision Systems for Inspection and for Manipulator Control, JACC 1977, vol.2, pg.131.
2. Al 81 Albus, J.S. et al, Hierarchical Control for Sensory Interactive Robots, Proc.II Int. Symp.Ind.Robots,pp. 18-39 sept.1981
3. Al 82 Albus J.S. et al, Programming a Hierarchical Robot Control System, ISIR Paris 1982 pg.505-517.
4. Ba 79 Bauzfi G., Ribes P., Robot HILARE : realization d'un robot mobile, Technical Report LAAS - 1979
5. BC 72 Barrow H., Crawford G.P., The Mark 1,5 Edinburgh Robot Facility, Machine Intelligence, Edinburgh Univ. Press, 1972
6. Be 76 Bejczy A.A., Allocation of control between man and computer in remote manipulation
7. Ca 84 Campbell D., An Iterative Algorithm for the Inverse Kinematics Solution of a General N-axis Robot Manipulator Using Powell's Optimization Technique, Robotics Age, August 1985
8. Cl 71 Clowes M.B., On Seeing things, Artificial Intelligence, 1971
9. Co 81 Coiffet P., Les robots-Interaction avec l'environnement, Tome 2, Editura Hermes Franța 1981
10. Cu 81 Cunningham K., Binary segmenting of images, Robotics Age July/Aug.1981, pg.4
11. De 55 Denavit J., Hartenberg E.S., A Kinematic Notation for Lower - pair mechanisms Based on Matrices, ASME J.App.Mech.22, 1955, pg.215-221
12. DH 73 Duda O.K., Hart P.E., Pattern Classification and Scene Analysis, 1973, John Wiley and Sons
13. Di 85 Drimer D. ș.a., Roboți Industriali și manipuloare, Ed.Tehnică 1985
14. Do 77 Dobrotin B.M., et al., An Application of Microprocessors to a Mars Hoving Vehicle, JACC, 1977, vol.2, pg.185
15. Dr 80 Drăgănescu M., A doua revoluție industrială. Microelectronica, automatica- factori determinanți, Ed.

Tehnică, București, 1980.

16. Da.e Duffy J., Rooney J., A Foundation for a Unified Theory of Analysis of Spatial Mechanisms, J.of Eng. Indust. Trans, ASME 97, series B, no.4, Nov. 1975
17. Du.b Duffy J., Analysis of Mechanisms and Robot Manipulators, New York : John Wiley Sons, 1980
18. Ev 77 Evans J.M., et al., Robot loading of an AC machine tool, JACC 1977 pg.720-724.
19. Fe 69 Feldman J.A., Feldman G.M., Falk G., et.al., The Stanford, Hand -Eye Project, First Intern. Joint Conf.on Artificial Intelligence, Washington, 1969
20. Fe 71 Feldman J.A., Sproull R., System Support for the Stanford Hand-Eye System, Proc. 2nd Intern.Joint Conference on A.I., London, 1971
21. Fl 74 Finkel R., et al., AL, A Programming System for Automation, Memo AIM-243, Stanford A.I.L. California, 1974
22. B 83 Flores S., Moangă A., Procedură pentru implementarea unui sistem de programare și comanda roboților, A 5-a Conf. Computer Systems - Control Systems, București, 1983
23. G1 79 Giraud A., Robot de chargement et de chargement d'un poste de travail sur une machine à plateau tournant, Final report, 1979
24. G1 83 Gîrlagă Dan, Implementation of a robot control System for the tactical and executive levels, A 5-a Conf.Computer Systems-Control Systems, București, 1983
25. G3 83 Gini G., Gini M., ROBOT - Controllo, programmazione e interazione con l'ambiente, CLUP-Milano 1983
26. G5 82 Gonzalez R.C., Safabakhsh R., Computer Vision Techniques for Industrial Applications and Robot Control, IEEE Computer, Dec., 1982,pg.17-32
27. Guzman A., Some Aspects of Pattern Recognition by Computer, Gu 67. MAC-TR-37 Project, MAC MIT, Cambridge, Mass,1967.

28. Gz 69 Guzman A., Decomposition of a visual Scene into Three-dimensional Bodies, Proc. NJCC, 1969
29. In 74 Inoue H., Force Feedback in Precise Assembly Tasks, in JACC 1977 vol.2, pg.168
30. HQ 83 Hollerbach J.M., Gideon Sahar, Wrist-partitioned Inverse Kinematic Accelerations and manipulator Dynamics, Int.J.Robotics Res.4 Winter, 1983, pg.61-76
31. HJ 80 Hill, J.W., Introducing Minimover 5, Robotics Age, Summer 1980 pg.18-27
32. Ho 76 Hohn K., Application Flexibility of a Computer-Controlled Industrial Robot, Technical Paper MR76-616, SME Dearborn, Michigan, 1976
33. HP 75 Heghinbotham W.B., Pugh A., Robotar mit Sehvermögen, Vortrag Nr.31 der SIPA -Arbeitstagung Stuttgart, 1975 (in Industrie-Roboter, Warnecke H-J)
34. Hu 69 Hueckel M., Locating Edges in Pictures, AI Memo, Stanford Calif. 1969 (in Integralis Robotis ed. MIK-MOSCOVA, 1972
35. Ja 82 Jarvis F.J., Research Directions in Industrial Machine Vision, IEEE Computer, Dec.1982, pg.55-61
36. Ka 77 Kashioka, S. și a., An Approach to the Integrated Intelligent Robot With Multiple Sensory Feedback: Visual Recognition Techniques, Proceedings of the 7 th ISIE, Tokio, 1977
37. Kq 82 Kovacs F., Cojocaru D., Manipulateare și roboți industriali, Ed. Pacla, 1982
38. Ko 86 Kovacs F., Cojocaru D., Roboții în acțiune, Ed. Pacla, 1986
39. Lu 83 Lumelsky V.J., Iterative Procedure for Inverse Coordinate Transformation for One Class of Robots, Robotics age, aug. 1985, pg.6-14
40. Ma 81 Marr D. VISION, Ed. W.H. Freeman San Francisco 1981
41. MC 68 McCarthy J., Earnest L., Reddy D., Vicens P.J. & A Computer with Hands, Eyes and Ears, Proc. Fall Joint Computer Conf. vol.33, Pt I, 1968.

42. MH 81 Hildreth E.C., Edge Detection in Man and Machine T.F.-MIT 1981. (bazat pe lucrările lui Marr)
43. MK 71 Munetaka J., Kohel S., Tadashi N., ETL Robot-1: Artificial Intelligent Robot, IEEE Japan Electronic Engineering, 1971
44. Ma 80 Marchiando C., Robot articulation control by micro-processor. Institute de microtechnique, Lausanne, Elveția.
45. MT 71 Masakazu Ejiri, Takehasu, et.al., An Intelligent Robot, with Cognition and Decision Making Ability, 2 nd Intern. Joint Conf.on Art.Intell., London 1971
46. Ma 66 Narasimhan R., Syntax Directed Interpretation of classes of Pictures, Comm. ACM, 9, No 3 (1966)
47. Mi 69 Nilsson N.J., Mobile Automation, An Application of Artificial Intelligence Techniques, Proc. of Intern. Joint Conf.on Artificial Intelligence, Washington, May 1969 (in Integralne Roboti, 1972, MIR-MOSCOVA)
48. Pa 76 Paul R., WAVE, A Model-Based Language for Manipulator Control, Technical Paper MH76-615, SME Dearborn, Michigan 1976
49. Pa 77 Paul R., Modelling, Trajectory Calculation and Servoing of a Computer Controlled Arm, JACC 1977
50. Pa 81 Paul R., Robot manipulators: mathematics, programming, and control, The MIT Press 1981
51. PB 75 Popplestone H.J., Brown C. M. et al., Forming models of plane - and cylinders faceted bodies from light Strips, 4 th IJCAI, Tbilisi, USSR, sept.1975
52. Pe 77 Perkins N.A., Model-based vision system for industrial parts, IEEE Transactions on Computers, febr.1977
53. Pl 68 Pieper D.L., Kinematic of Manipulators Under Computer Control, Ph. D. Thesis, Stanford Univ.1968
53. PE 69 Pieper D.L., Roth B., The Kinematics of Manipulators Under Computer Control, Proc.II Int.Cong.Theory Machine and Mechanisms, 2(1969), pg.159-168
54. PK 77 Park J.W., Minicomputer software organization for control of industrial robots, JACC-1977 vol.2,pg.164

55. Po 72 Poddniak B., Integralne roboti, Editura MIR
Moscow, vol.1 si vol.2 (1975)
56. Po 84 Powell M.J.D., An Efficient Method for Finding the
minimum of a Function of Several Variables Without
Calculating Derivates, The Computer Journal, 1984
57. Pr 80 Prajoux R., Sobek K, et al., A robot System utili-
zing task specific planning in a block world assem-
bly experiment, Proc. of 10 th Int.Symp.Ind.Robot
Milan 1980
58. PS 68 Pingle K.K., Singer J.A., Wiseman W.M., Computer Con-
trol of a Mechanical Arm through Visual Input, Proc.
IFIP-68, 1962
59. Ra 76 Raibert M., A State Space Model for Sensorimotor
Control and Learning, Memo 351, A.I.Lab. Cambridge,
1976
60. RM 77 Rosen G.A., Nitzan D., Use of sensors in Programmable
Automation, IEEE Computer, decembrie 77.
61. Ro 63 Roberts L.G., Machine Perception of Three-Dimensional
Solids, Optical and Electro-Optical Information Pro-
cessing, MIT Press, 1963
62. Ro 65 Roberts L.G., Machine Perception of Three-Dimensional
Solids, Optical and Electro-Optical Information Pro-
cessing, MIT Press, 1965
63. Rs 74 Rosen G, et al., Exploratory Research in Advanced Au-
tomation, JACC 1977 vol.2, pg.169
64. Rs 78 Rosen G.A., Machine vision and robotics : industrial
requirements, in Computer vision and sensor based ro-
botics, Plenum Press, 1978
65. IV 80 Reinhold A.G., Vanderbrug G., The Autovision System,
Robotics Age, Fall 1980 pg.22-28
66. Sh 84 Shahinpoor M., Campbell D., An Iterative Solution to
the Generalized Robotics Inverse Kinematic Problem
by Means of Powell's Optimization Technique, Res.Rep.
No MTE-106, Clarkson University, Aug.1984
67. Sh 85 Shahinpoor M.M., Exact Solutions to the Inverse Kine-
matics Problem of a Standard 6-axis Robot Manipulator,

IEEE Journal of Robotics, 1985

68. SI 73 Silver D., The Little Robot System, in Joint Automation Control Conf. vol.2, 1977, pg.170
69. So 69 Sobel I., Visual Accomodation in Machine Perception, 1969
70. Sp 76 Speckert G., Hand Eye Coordination, Working Paper 127, MIT Artificial Intelligence Lab., 1976
71. SS 70 Sproull F.F., Swinehart D., SAIL, Stanford Artificial Intelligence Project Operating Note, No 57, 1970
72. St 76: Stoinescu F., Conversia numerică a imaginilor TV prin eşantionare încetată, în Probleme Actuale de Conducere și Informatică, editura Dacia, 1975
73. St 78 Stoinescu F., Enătescu V., Principii peinoiberne- tice aplicate la roboții cu inteligență senzorio- toare, în Probleme Actuale de conducere și Informa- tică, editura Dacia 1978
74. St 81 Stoinescu F. și colectiv, Robotul industrial IMAI-1 Al III-lea Simpozion Internațional de Teorie și Prac- tica Mecanismelor, București 1987
75. St 82 Stoinescu F. și colectiv, Robotul industrial IMAI-2, Al II-lea Simpozion Național de Roboți industriali, București 1982.
76. St 80a Stoinescu F., Strugaru C., Microprocesorul 8080 în configurație minimă, Buletin Stiințific al I.P. "Traian Vuia" Tom 25, Fasc.2, 1980.
77. St 80b Stoinescu F., Strugaru C., Interfața serie programe- bilă pentru DAF 1001, Buletin Stiințific al I.P. "Traian Vuia" Tom 25, Fasc.2, 1980
78. St 85 Stoinescu F., Hand-eye experiments with the mobile robot P.A.I.E.C.S., Simpozionul Microprocesore, microcalculatoare și aplicații în economie, Timișoara 29-30 noiembrie 1985
79. St 85 Stoinescu F., Văduva Gh., O metodă de partiționare a spațiului de lucru a unui robot de manipulare, Simpo- zionul Național de roboți industriali, Timișoara, 1985

80. St 84 Stoinescu K., Structura Sistemelor de comandă a roboților actuali , Referat științific de doctorat, Biblioteca I.P. "Traian Vuia" Timișoara 1984
81. St 85 Stoinescu K., Sisteme ierarhizate de comandă în robotică, Referat științific de doctorat, Biblioteca I.P. "Traian Vuia" Timișoara, 1985
82. Sm 82 Saurad E., A microcomputer based real-time robotics system, Robotics Age, Nov./Dec. 1982, vol.4, No.6, pg.27
83. Sn 80 Snyder E.W., Microcomputer based path control, Robotics Age, Spring 1980 pg.7
84. Ta 77 Takeyasu, K. g.a., An Approach to the Integrated Intelligent Robot with Multiple Sensory Feedbacks: Construction and Control Functions, Proceedings of the 7th ISIF, Tokio, 1977
85. TB 79 Tenenbaum J.A., Barrow H.G. et al., Prospects for industrial vision, in Computer vision and sensor-based robots, Plenum Press, 1979
86. Th 79 Thompson M.A., Introduction to robot vision, Robotics Age, Summer 1979
87. Th 81 Thompson M.A., Camera geometry for robot vision, Robotics Age Mar/Apr.1981, pg.20
88. La 84 Tsai L.W., Morgan A.P., Solving the Kinematics of the most General Six - and Five - Degree of Freedom Manipulators by Continuation Methods, General Motors Research Report, No GM-4631; Oct.1984
89. Wa 79 Warnacke H., Schraft I., Industrie roboter, Krausskopf Verlag, 1979
90. WB 82 Winston P.H., Horn E.H., Artificial Intelligence: An MIT Perspective, The MIT Press 1982, vol.1, și vol.2
91. We 76 Masaka J.S., A Survey of Threshold Selection Techniques, Computer Graphics and Image Processing, 5-1976 p382
92. WG 75 Mill P., Grossman D., An Experimental System for Computer Controlled Mechanical Assembly, IEEE Transactions on Computers Vol.C.24, No.9, Sept.1975

93. wh 77 Whitney D.E., Robot and manipulator control by exte-
receptive sensors, JACC 1977 -pg.155-163
94. Wt 81 Winston P.H., Inteligența artificială, Ed.Tehnică,
București 1981
95. Wl 81 Wilf J.A., Chain-code, Robotics Age Mar/Apr.1981
pg.12
96. WA 78 Ward M.E., Rosol L., Holland S.W., COMSIGHT-a prac-
tical vision - based robot guidance System, 9 th
ISIR Washington, SUA 1979
97. x x x UNIMATE 4000 B, manual de prezentare și întreținere
al firmei UNIMATION INC
98. x x x Proceedings of the 12 th -ISIR - Paris 1982
99. x x x Distanțioane upravliaemie roboti-manipulatori,
editura MIE , Moscova 1976
100. x x x Course on pattern recognition and image processing,
Delft University of Technology, Department of Applied
Physics, 1976
101. x x x Documentația de firă a sistemului de calcul
ROBOTRON 4201, 1976
102. x x x Inteligența artificială și robotica, Editura Acade-
miei R.S. România, București, 1983
103. x x x Programme automation et robotique avansé -
Laboratoire LAAS Toulouse, Franța 1979
104. x x x Proceedings of the 1 st International Conference on
Robot Vision and Sensory Controls, Stratford-upon-
Avon, Marea Britanie, 1981
105. x x x Video signal input, Robotics Age, Mar./Apr.1981
106. AD75 Agin G.J., Duda R.O., SAI vision research for advanced
industrial automation, Proc. 2nd ISA, Japan Computer
Conference,1975
107. MS81 Morgan T., Lstraganu G., Steinescu N., Petriu M.,
Microprocesorul 8086 în aplicații, ed. revia 1981