

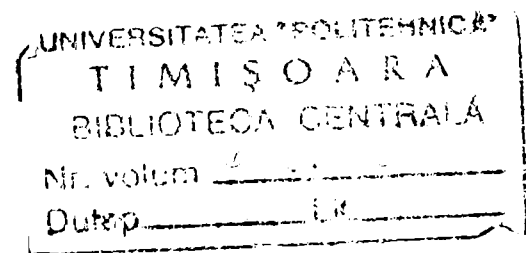
UNIVERSITATEA "POLITEHNICA" DIN TIMIȘOARA
FACULTATEA DE MECANICĂ

Ing. RADU CĂTĂLIN ȚARCĂ

UTILIZAREA INFORMAȚIILOR CU PRIVIRE
LA SITUAREA EFECTORULUI FINAL
ACHIZIȚIONATE PRIN SENZORI,
ÎN VEDEREA CONDUCERII
ÎN TIMP REAL A ROBOȚILOR

TEZĂ DE DOCTORAT

BIBLIOTECA CENTRALĂ
UNIVERSITATEA "POLITEHNICA"
TIMIȘOARA



Conducător științific
Prof. Dr. Ing. Dr. h. c. KOVÁCS FRANCISC VILIAM

TIMIȘOARA
. 2001

Prefață

De multe ori se întâmplă să privim în urmă, în timp, încercând să vedem dacă ceea ce credeam odată că este posibil s-a și materializat conform dorințelor noastre. Și când această materializare se produce ca o finalizare a unei etape de perfecționare a existenței noastre, iar dacă mediul în care s-a concretizat acea realizare este unul de maximă exigență, apare sentimentul de satisfacție, de împlinire.

Tocmai de aceea, în momentul în care, un doctorand al Prof.Dr.Ing.Dr.h.c. Kovacs Francisc Viliam își finalizează activitatea de cercetare printr-o teză de doctorat nu poate decât să încerce un sentiment de împlinire profesională.

Pentru încrederea acordată încă de la început și până în momentul finalizării lucrării, pentru sfaturile și impecabila coordonare pe tot parcursul elaborării tezei de doctorat, mulțumesc distinsului meu conducător științific Prof.Dr.Ing.Dr.h.c. Kovacs Francisc Viliam.

Un cald cuvânt de mulțumire tuturor colegilor și îndeosebi d-lui prep.ing. Cezar Cuc și d-lui ș.l. ing. Tiberiu Vesselenyi care prin sfatul lor, prin propunerile concrete, prin ajutorul dat mi-au fost de un real folos în finalizarea acestei lucrări.

Dar toate acestea au fost posibile numai în măsura în care familia mea a fost aproape întotdeauna prin încurajări, prin sfaturi, prin înțelegerea cu care am fost înconjurat pe tot parcursul acestei perioade.

Ing. Radu Cătălin Țarcă

Cuprins

UTILIZAREA INFORMAȚIILOR CU PRIVIRE LA SITUAȚIA EFECTORULUI FINAL ACHIZIȚIONATE PRIN SENZORI ÎN VEDEREA CONDUCERII ÎN TIMP REAL A ROBOȚILOR

Prefață.....	1
Cuprins	3
Capitolul 1. Stadiul actual al cercetărilor și realizărilor în domeniul conducerii roboților folosind informațiile sistemelor senzoriale	9
1.1. Sistemul de conducere al robotului. Funcții. Arhitectură.	9
1.2. Tehnici de conducere a roboților	18
1.2.1. Principii.....	18
1.2.2. Conducerea nemijlocită a mișcărilor elementelor cuplelor cinematice conducătoare utilizând metode nonadaptive	18
1.2.3. Strategii de conducere specifice roboților adaptivi	19
1.2.3.1. Definiții.....	19
1.2.3.2. Robot adaptiv cu funcții specifice conducerii senzoriale	20
1.2.3.3. Robot adaptiv cu funcții specifice conducerii adaptive	24
1.2.3.4. Robot adaptiv cu funcții specifice conducerii prin învățare	26
1.3. Cercetători și grupuri de lucru care lucrează în visual servoing	31
Capitolul 2. Scopul și obiectivele lucrării	33
Capitolul 3. Conducerea dinamică a robotului.....	35
3.1. Noțiuni introductive.....	35
3.2. Modelarea dispozitivului de ghidare al robotului.....	39
3.2.1. Formalisme.....	39
3.2.2. Formalismul Lagrange.....	39
3.2.3. Formalismul Newton-Euler	43
3.3. Modelarea sistemelor de acționare	46
3.3.1. Introducere	46
3.3.2. Transmisii mecanice. Generalități.....	46
3.3.3. Modelarea și alegerea transmisiei mecanice.....	48

3.3.4. Modelarea acționării electrice.....	54
3.3.4.1. Modelarea acționării electrice cu motor de curent continuu.....	54
3.3.4.2. Modelarea acționării electrice cu motoare pas cu pas.....	58
3.3.4.2.1. Ecuații de bază.....	58
3.3.4.2.2. Generarea impulsurilor.....	61
3.3.5. Modelarea acționării hidraulice.....	63
3.4. Modelul dinamic complet al robotului.....	65
Capitolul 4. Sisteme senzoriale vizuale.....	67
4.1. Noțiuni introductive.....	67
4.1.1. Principii.....	67
4.1.2. Iluminarea.....	67
4.1.3. Suprafețe reflectorizante.....	69
4.2. Senzori vizuali.....	70
4.3. Formarea imaginii.....	73
4.3.1. Transformări în perspectivă.....	73
4.3.2. Modelul "cameră".....	77
4.3.3. Calibrarea camerei.....	80
4.3.3.1. Principii.....	80
4.3.3.2. Calibrarea clasică non-lineară.....	82
4.3.3.3. Calibrarea prin transformări omogene.....	82
4.3.3.4. Calibrarea cu "două plane".....	83
4.3.3.5. Calibrarea "în două etape".....	84
4.3.4. Imaginea stereoscopică.....	85
4.4. Structura și funcțiile sistemelor senzoriale vizuale.....	88
4.5. Etapele prelucrării imaginilor.....	91
4.6. Programe de prelucrare digitală a imaginilor.....	96
Capitolul 5. Contribuții la concepția unor sisteme de conducere senzorială a roboților utilizând informații furnizate de senzori vizuali.....	99
5.1. Noțiuni introductive.....	99
5.2. Principiile metodei propuse de conducere senzorială a robotului pe baza informațiilor din mediu, achiziționate de senzori vizuali.....	104
5.3. Achiziționarea informațiilor privind situarea reală a efectorului final de către subsistemul vizual de urmărire.....	107
5.3.1. Noțiuni introductive.....	107

5.3.2. Modelarea matematică a subsistemului vizual de urmărire	109
5.3.2.1. Scopul modelării	109
5.3.2.2. Modelul geometric al mecanismului subsistemului vizual de urmărire	110
5.3.3. Algoritm pentru determinarea poziției țintei	112
5.3.4. Algoritm pentru determinarea orientării țintei	119
5.4. Achiziționarea informațiilor privind situarea reală a efectorului final prin subsistemul cu senzor vizual fix	122
5.4.1. Noțiuni introductive	122
5.4.2. Determinarea situației reale a OL exprimate în sistemul de referință legat de senzorul vizual fix	123
5.4.3. Determinarea situației reale a EF în sistemul de referință atașat senzorului vizual fix	128
5.5. Robotul	129
5.5.1. Structura robotului	129
5.5.2. Modelul geometric al robotului	131
5.5.3. Determinarea situației ideale a efectorului final al robotului	132
5.6. Determinarea erorilor de situație a obiectului manipulat și calculul comenzilor de corecție	133
5.6.1. Stabilirea matricei de eroare de situație	133
5.6.2. Modul de calcul al mărimilor de corecție de la nivelul fiecărei c.c.c.	136
5.7. Modalități de programare a comenzilor de corecție a erorilor de situație	139
5.8. Concluzii	141
Capitolul 6. Concepția unui sistem concret de conducere senzorială a unui robot utilizând informații furnizate de un sistem vizual de urmărire a situației efectorului final. Modelarea geometrică a sistemului și simularea funcționării lui prin animație.	143
6.1. Introducere	143
6.2. Modelarea geometrică 3D a sistemului robot	144
6.2.1. Noțiuni introductive	144
6.2.2. Etapele parcurse	144
6.2.3. Etapele de parcurs în vederea realizării simulării prin animație în 3D a mișcărilor sistemului robot	150
6.3. Exemplificarea simulării prin animație a robotului pentru o lege de mișcare dată a punctului caracteristic al efectorului final	153

6.4. Concluzii	154
Capitolul 7. Modelarea funcțională a sistemului de conducere senzorială conceput și simularea funcționării lui	155
7.1. Scopul modelării și simulării funcționale a sistemului	155
7.2. Modelarea funcțională a componentelor sistemului de conducere senzorială a robotului	156
7.2.1. Modelarea funcțională a motorului de curent continuu	156
7.2.2. Modelarea funcțională a motorului pas cu pas	157
7.2.3. Modelarea funcțională a senzorului vizual.....	160
7.2.4. Modelarea funcțională a sistemului de calcul.....	163
7.3. Modelarea funcțională a sistemului robot realizat	167
7.4. Prezentarea rezultatelor simulării pe modelul experimental pentru conducerea pentru o c.c.c.	169
7.5. Prezentarea rezultatelor simulării pe modelul funcțional.....	176
7.6. Concluzii	178
Capitolul 8. Construcția unui sistem experimental de conducere senzorială a unui robot cu două grade de mobilitate, pe baza informațiilor achiziționate printr-un subsistem vizual de urmărire. Experimentări.	179
8.1. Introducere.....	179
8.2. Descrierea sistemului experimental conceput și realizat.....	179
8.2.1. Prezentare generală.....	179
8.2.2. Particularizarea modului de calcul al mărimilor de corecție la nivelul fiecărei c.c.c. pentru sistemul robot experimental conceput și realizat.....	184
8.3. Proiectarea și realizarea părților componente.....	187
8.3.1. Proiectarea și realizarea sistemului mecanic al subsistemului vizual de urmărire	187
8.3.2. Proiectarea și realizarea sistemelor de acționare.....	189
8.3.2.1. Noțiuni introductive	189
8.3.2.2. Sistemul de acționare al robotului.....	190
8.3.2.2.1. Structura sistemului de acționare al robotului.....	190
8.3.2.2.2. Parametrii sistemului de acționare ai robotului.....	190
8.3.2.3. Sistemul de acționare a subsistemului vizual de urmărire	192
8.3.3. Proiectarea și realizarea sistemului de comandă a sistemului robot	192
8.3.3.1. Noțiuni introductive	192

<i>Teză de doctorat</i>	
8.3.3.2. Structura sistemului de comandă a sistemului robot realizat	193
8.3.3.3. Interfață pentru comanda motorului pas cu pas	196
8.3.3.4. Interfețe pentru comanda motoarelor de curent continuu	199
8.4. Elaborarea programului de conducere senzorială a robotului utilizând informații furnizate de senzori vizuali	206
8.5. Experimentări	213
8.6. Concluzii	221
Capitolul 9. Contribuții la realizarea unui sistem robot la care comanda de corecție a situării efectorului final se realizează prin utilizarea unui senzor vizual fix	223
9.1. Noțiuni introductive	223
9.2. Sistemul experimental realizat	223
9.2.1. Descrierea sistemului experimental realizat	223
9.2.2. Particularizarea modului de calcul al situării reale a EF în sistemul de referință universal	225
9.2.3. Elaborarea programului de conducere a robotului utilizând informațiile furnizate de senzorul vizual fix	226
9.3. Concluzii	235
Capitolul 10. Contribuții originale. Concluzii finale	237
10.1. Contribuții originale	237
10.1.1. Introducere	237
10.1.2. Contribuții în domeniul sistematizării informațiilor bibliografice	238
10.1.3. Contribuții în plan teoretic	239
10.1.4. Contribuții în domeniul cercetării experimentale	239
10.1.5. Contribuții în domeniul aplicațiilor practice	240
10.1.6. Direcții de cercetare viitoare	241
10.2. Concluzii finale	241
Bibliografie	243
Anexe	

CAPITOLUL 1 Stadiul actual al cercetărilor și realizărilor în domeniul conducerii roboților folosind informațiile sistemelor senzoriale

1.1. Sistemul de conducere al robotului. Funcții. Arhitectură

Rolul sistemului de comandă din cadrul hipersistemului "robot" - privit prin prisma antropomorfismului său structural - este echivalent cu rolul sistemului nervos.

Astfel, dacă sistemul nervos asigură o legătură funcțională reciprocă între organele aparatele și sistemele organismului și realizează unitatea organismului cu mediul înconjurător prin recepționarea excitațiilor externe sau interne, a transmiterii și analizării lor și a elaborării răspunsurilor motorii și secretorii, sistemul de conducere al robotului realizează în mare aceleași funcții, dar la un nivel de complexitate mult mai redus.

Prelevarea și transmiterea informațiilor de la și către celelalte sisteme interne robotului precum și de la și spre mediul exterior se realizează prin intermediul traductoarelor și senzorilor. Rolul lor este topologic echivalent cu cel al analizatorilor, care recepționează excitațiile mediului extern sau intern și conduc excitațiile la scoarța cerebrală.

În figura 1.1 se prezintă o clasificare a modalităților de realizare a conducerii robotului funcție de existența-inexistența unui "contact" între efectorul final sau obiectul prehensat de efectorul final și mediul înconjurător [Soml97].

Astfel, în cazul în care nu există un contact între efectorul final și mediul scopul conducerii robotului este acela de a asigura punctului caracteristic în fiecare moment poziții, respectiv deplasări cu viteze și accelerații impuse sau determinate în raport cu obiectele din mediu, fără coliziuni, dar astfel încât mișcarea să fie "lină" (traectoria pe care o parcurge punctul caracteristic să nu prezinte puncte unghiulare sau de întoarcere).

Se vorbește despre comandă în poziție atunci când poziția punctului caracteristic este mărimea controlată în vederea realizării conducerii robotului, respectiv despre comandă în accelerație sau viteză atunci când punctului caracteristic al robotului i se prescriu și controlează respectiv aceste mărimi.

Dacă există contact între efectorul final și mediul (de exemplu atunci când robotul realizează o asamblare sau atunci când lucrează cu unele capete de forță) în acest caz, de obicei, se necesită realizarea unei "conduceri hibride" [Asad86] atât cinematic cât și dinamic. În cazul conducerii dinamice mărimea prescrisă și controlată o reprezintă forța

generalizată cu care efectorul final trebuie să interacționeze cu mediul. Posibilitățile de rezolvare a "contactului" dintre efectorul final și mediul sunt două și anume: *compliance pasivă* (materializată prin utilizarea unor dispozitive de compliance) dar în special *compliance activă* [Coif92], [Warn89].

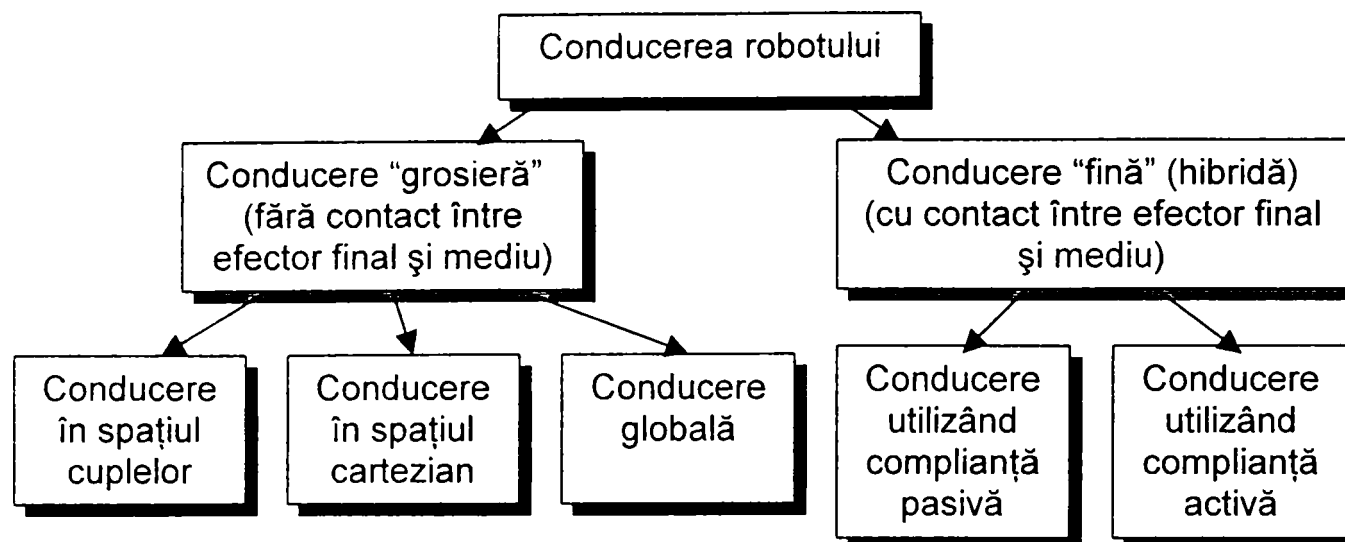


Fig. 1.1. Clasificare a tipurilor de conducere [Soml97]

Sarcina conducerii robotului constă în determinarea vectorului mărimii prescrise (vector de intrare) aferente acționării astfel încât să se realizeze *scopul conducerii robotului* (care în general este de a asigura punctului caracteristic în fiecare moment - cu o precizie prestabilită - poziții, respectiv deplasări cu viteze și accelerații impuse sau determinate în raport cu obiectele din mediul, fără coliziuni, dar astfel încât mișcarea să fie "lină").

Din punct de vedere matematic această problemă poate fi formulată sub următoarea formă:

Sarcina conducerii robotului constă în determinarea vectorului mărimii prescrise aferente acționării - $S(t)$ cu $0 < t < T$ - astfel încât să se realizeze în orice moment relația:

$$|\Psi[t, \Psi(0), A, \Gamma, S(t)] - \Psi_d(t)| < \varepsilon(t) \quad (1.1)$$

unde: - $\Psi(t)$ reprezintă mișcarea realizată de către robot (fie că se controlează situarea efectorului final sau coordonatele generalizate ale cuplelor cinematice conducătoare);

- $\Psi_d(t)$ reprezintă mișcarea prescrisă (fie ca poziție, fie ca viteză, fie ca și cuplu la nivelul cuplelor cinematice conducătoare);

- $\varepsilon(t)$ reprezintă toleranța prescrisă dintre mișcarea realizată și mișcarea prescrisă;

- $\Psi(0)$ reprezintă vectorul condițiilor inițiale;

- $A \in \mathbb{R}_A$ reprezintă vectorul parametrilor acționării;

- $\Gamma \in \mathbb{R}_\Gamma$ reprezintă vectorul parametrilor elementelor dispozitivului de ghidare.

Funcțiile sistemului de conducere sunt următoarele:

- ☞ modelarea sistemului condus, dar și a sistemului de conducere;
- ☞ specificarea, generarea și controlul mișcărilor care se realizează în buclă prin parcurgerea următorilor pași: percepție, decizie, respectiv acțiune;
- ☞ dialogul cu operatorul uman.

În tabelul 1.1.se prezintă o clasificare a tipurilor de conducere a membrilor "familiei robot". [Tarc98d].

Tabelul 1.1. Clasificare a tipurilor de conducere a membrilor "familiei robot"

Tipul de conducere	Cine coordonează conducerea ?	Modalitatea de realizare a conducerii	Utilizarea tipului de comandă în cadrul "familiei robot"
Conducere manuală	- operatorul uman	- prin întrerupătoare, butoane, manete (joy-stick)	-manipulator -manipulator cu comandă manuală.
		- prin copiere	-instalații de teleoperare -exoschelete -proteze -orteze -mașini pășitoare-uneori
Conducere automată	- calculatorul		-robot -manipulator -robocar -mașină pășitoare, târătoare
Conducere mixtă	- operator uman + calculator	- simultan	-robot cu un număr mare de grade de mobilitate; -manipulator medical
		- secvențial (serial)	-proteze -orteze -robot

Conducerea manuală a membrilor "familiei robot" se realizează fie prin întrerupătoare/butoane când se pornește/oprește succesiv/simultan motoarele care acționează mișcările relative ale elementelor cuplelor cinematice conducătoare, respectiv *manete* – cu 1-3 grade de libertate – când se permite transmiterea unor semnale de comandă de tipul celor din instalațiile "master-slave", fie prin *copiere* când operatorul uman "generează" mișcarea, iar "robotul" o execută.

Exemplul tipic pentru acest mod de lucru este cel al instalațiilor sincrone "master-slave", constituite din două lanțuri cinematice asemenea, unul numit "master"(stăpân) și fiind pus în mișcare de operatorul uman, iar cel de-al doilea denumit lanț "slave" executând prin "copiere" mișcările lanțului "master".

Conducerea automată a membrilor "familiei robot" presupune *planificarea și generarea* traiectoriei și *controlul* mișcărilor, în mod automat, de către sistemul de conducere al robotului.

Scopul *planificării traiectoriei* este acela de a stabili punctele țintă impuse de procesul tehnologic și de a impune ca mișcările între aceste puncte (situări) să fie lipsite de coliziuni. Aceste deziderate se realizează utilizând *interblocările* în acest sens.

Prin *interblocări* se înțeleg condiționările logice a începerii respectiv sfârșitului mișcării unor componente (subsisteme), de începutul/sfârșitul altei componente sau realizării/nerealizării unei anumite stări exterioare.

Pozițiile de atins în fiecare etapă pot fi determinate în două moduri și anume: fie prin *învățare (teach in)* fie prin *calcul*. În primul caz al învățării operatorul conduce manual dispozitivul de ghidare al robotului, reglând pozițiile și orientările acestuia, iar când acestea sunt corecte, ele sunt memorate.

În cel de-al doilea caz, este necesar a se *modela* mediul exterior, model ce este utilizat apoi la determinarea prin calcul a situațiilor corespunzătoare punctelor țintă și via din cadrul fiecărei etape a procesului tehnologic. Cel de-al doilea caz se utilizează îndeosebi în cazul în care procesul tehnologic este flexibil.

Arhitectura generală a sistemului de conducere automată de tip multiprocesor, al unui robot industrial ce prezintă k grade de mobilitate se prezintă în figura.1.2.

Se pot identifica următoarele elemente componente ale hipersistemului robot:

- sistem de comandă;
 - sistem de acționare;
 - sistem mecanic;
 - sistem de diagnoză;
 - sistem de preparare a energiei.
- } = sistem de conducere

Dacă primele trei sisteme enumerate mai sus apar întotdeauna în componența robotului, ultimele două sisteme sunt sisteme auxiliare și apar funcție de necesitatea realizării/nerealizării diagnozei robotului și respectiv de necesitatea sau nu a producerii agentului de transmitere a energiei în cadrul robotului.

Sistemul de comandă realizează *funcția de comandă*, care prescrie începutul și sfârșitul diferitelor activități ale robotului și a parametrilor care caracterizează aceste activități.

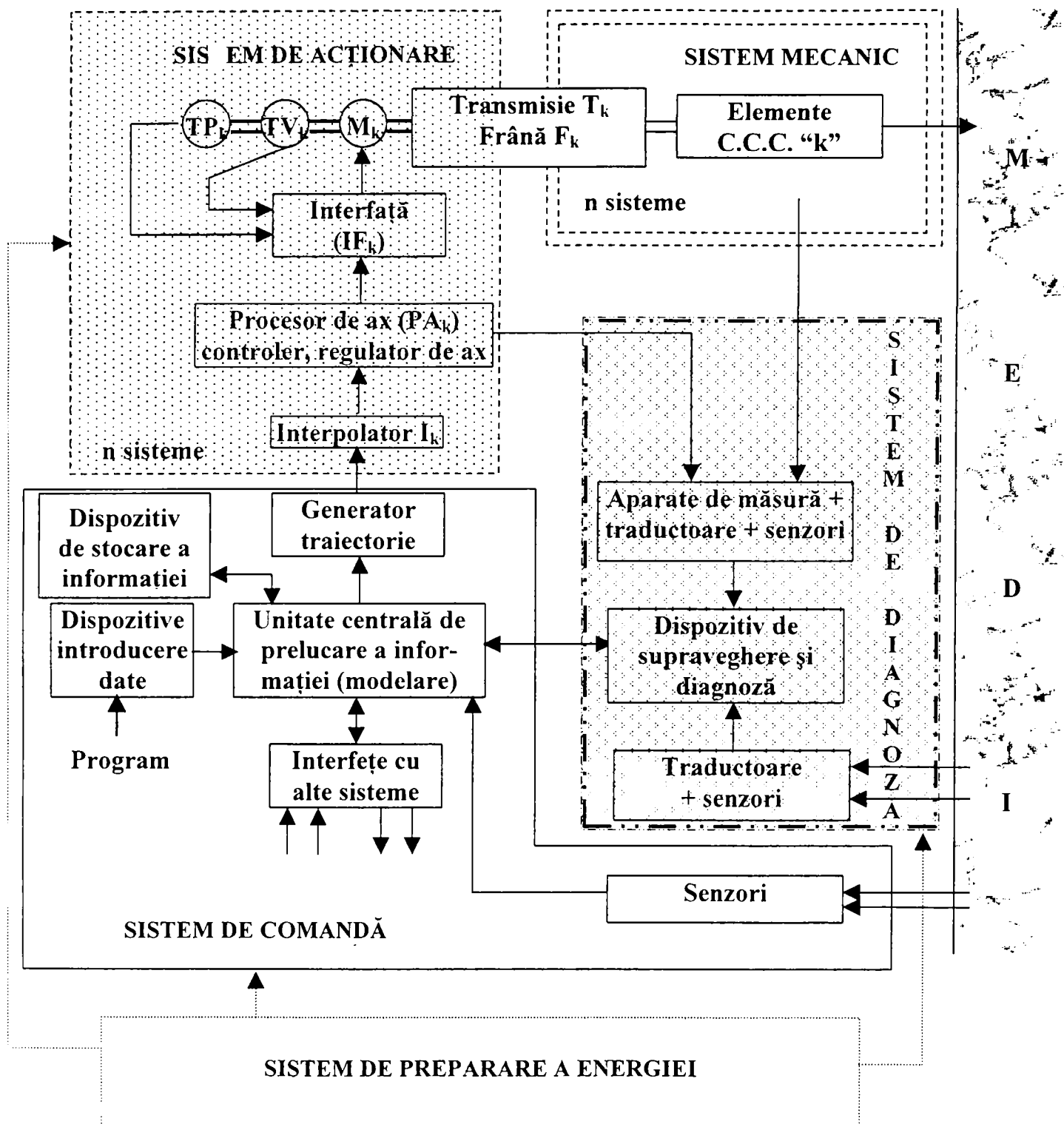


Fig. 1.2. Arhitectura sistemului de conducere văzută din prisma interconexiunilor ce apar în cadrul hipersistemului robot

Din figura 1.2. se observă că elementele componente ale sistemului de comandă sunt:

☛ unitatea centrală de prelucrare a informației (modelare) – componentă ce coordonează sistemul de comandă al robotului și care realizează următoarele funcții:

- evaluarea stării procesului și a mediului în care evoluează robotul;

- stabilirea conexiunilor/legăturilor logice în concordanță cu programul de realizat;
- stabilirea condiționărilor/interblocărilor;
- analiza informațiilor furnizate de senzori;
- analiza informațiilor furnizate de sistemul de diagnoză;
- transmiterea de informații spre generatorul de traiectorie în vederea realizării conducerii robotului;
- schimb de informații cu dispozitivul de stocare a bazelor de date a sistemului de comandă.

☞ programul – un set de informații codificat – ce conține: date, informații referitoare la diferiți parametri ai procesului și mișcărilor comandate, respectiv referitoare la porniri/opriri; conexiuni logice de tip "IF"; salturi peste anumite părți de program; date referitoare la mediu, proces; instrucțiuni de interogare a unor traductoare, senzori.

☞ dispozitivul de introducere date – care, în general, se materializează prin tastatură:

☞ dispozitiv de stocare a informației, care are ca rol memorarea unor date și redarea lor la momentul dorit; din punct de vedere constructiv dispozitivul de stocare a informației poate fi magnetic, electric, mecanic sau fluidic.

☞ senzori ce prelevează informația de la mediul în care evoluează robotul; în unele cazuri atașat senzorilor apar în cadrul sistemului de comandă și procesoare aferente acestora – în special la senzorii vizuali și respectiv la cei de forță/moment;

☞ interfețele cu alte sisteme realizează:

- comunicația cu sisteme de conducere ierarhic superior (spre ex.: intrare paralelă Centronics, intrare/ieșire serială RS232C);

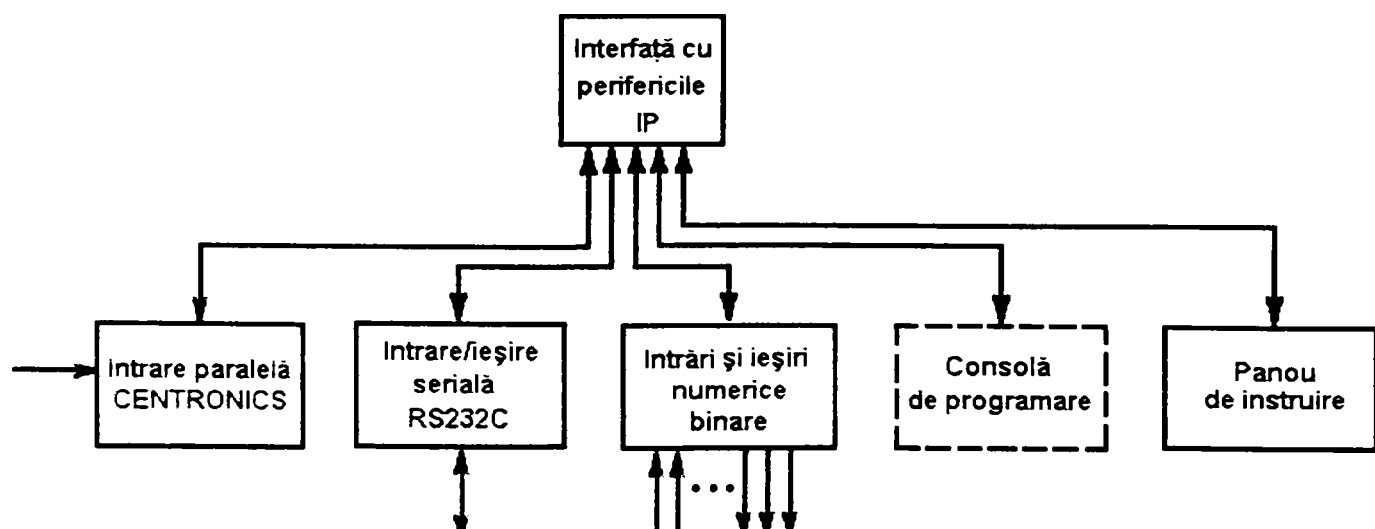


Fig.1.3. Tipuri de interfețe cu perifericele

- sincronizarea programelor de aplicație cu comenzile numerice ale stațiilor deservite, intrări/ieșiri numerice binare;
 - planificarea traiectoriei prin învățare (teach in) prin utilizarea panoului de instruire;
 - consolă de programare opțional, când lipsește unitatea de programare.
- ☞ generatorul de traiectorie.

Generarea traiectoriei este realizată de către un bloc (un software sau un hardware dedicat) din cadrul sistemului de conducere al robotului, numit sugestiv *generator de traiectorie*. Acesta calculează, pe baza unor algoritmi prestabiliți, valorile numerice discrete ale funcțiilor de conducere, la intervale de timp constante, ce poartă numele de perioade de generare.

După cum se observă din schema blocului generator din figura 1.4., acesta are ca mărimi de intrare condiții pentru deplasare, condiții impuse de dinamica robotului și opțional prescrierea unei anumite traiectorii, iar ca ieșire coordonatele, vitezele respectiv accelerațiile fie în coordonate carteziane, fie în coordonatele cuplelor cinematice conducătoare. Valorile numerice calculate ale pozițiilor generalizate, vitezelor generalizate, respectiv ale accelerațiilor generalizate, de către generatorul de traiectorie, reprezintă mărimile de intrare pentru sistemul de acționare.

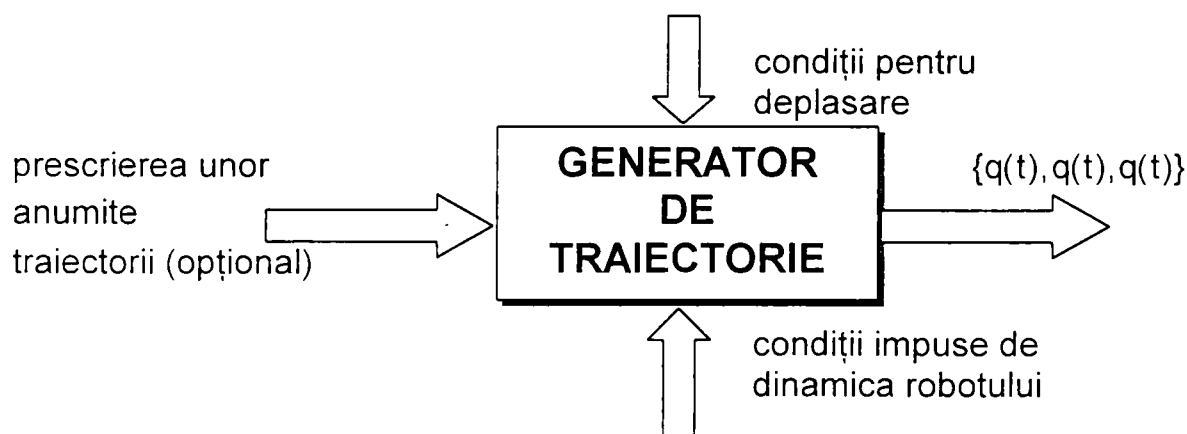


Fig. 1.4. Generatorul de traiectorie

Sistemul de acționare are rolul de a pune în mișcare relativă elementele cuplelor cinematice conducătoare ale robotului sau după caz platforma mobilă pe care este amplasat robotul.

După cum se observă din figura 1.2. părțile componente ale sistemului de acționare sunt:

- ☞ interpolator I_k , $k = \overline{1, n}$ pentru fiecare acționare;

☞ procesoare de ax descentralizate pentru controlul nemijlocit al mișcărilor PA_k $k = \overline{1, n}$ ce realizează conducerea descentralizată a mișcărilor (la nivelul fiecărei cuple cinematice conducătoare); controlul și reglarea mișcărilor se realizează prin intermediul unui algoritm de reglare numerică pe baza informațiilor furnizate de către generatorul de traiectorie prin intermediul interpolatoarelor I_k , respectiv pe baza mărimilor furnizate de traductorul de poziție TP_k și cel de viteză TV_k . Rolul algoritmului de reglare numerică este determinarea periodică a unei mărimi cu care se comandă motorul M_k prin intermediul unor interfețe (IF_k) specifice tipului motorului de acționare;

☞ interfețe descentralizate pentru comanda motoarelor de acționare IF_k $k = \overline{1, n}$;

☞ motorul de acționare M_k , $k = \overline{1, n}$;

Motoarele sistemului de acționare pot acționa direct sau indirect cuplele cinematice conducătoare ale robotului. În cazul în care acționarea este una indirectă, între motor și elementele cuplei cinematice se interpun transmisii mecanice, care asigură adaptarea caracteristicii mecanice a motorului de acționare la cerințele impuse de cupla cinematică conducătoare.

☞ traductorul de poziție TP_k , $k = \overline{1, n}$;

☞ traductorul de viteză TV_k , $k = \overline{1, n}$.

În unele cazuri pe lângă traductoarele de poziție și respectiv de viteză care sunt amplasate la nivelul fiecărei cuple, dacă se dorește o conducere dinamică a robotului, în arhitectura sistemului de conducere a unui robot apar și traductoare de forță. Însă datorită volumului mare de calcule pe care îl implică acest tip de conducere, rareori este utilizat.

Sistemul mecanic al robotului are rolul de a asigura realizarea mișcărilor acestuia și transmiterea energiei mecanice necesare interacțiunii cu mediul. Componentele sistemului mecanic sunt dispozitivul de ghidare prevăzut cu efector final, iar în cazul în care robotul este unul mobil, i se prevede și o platformă mobilă.

Sistemul de diagnoză are în componența sa aparate de măsurare, traductoare și senzori astfel amplasați încât să preleveze informații despre starea sistemului mecanic al robotului și, respectiv, despre starea sistemului exterior - în care evoluează robotul. Aceste informații sunt prelucrate de către *dispozitivele de supraveghere și diagnoză* prelucrate și transmise către unitatea centrală.

În ultimii opt ani s-au pus bazele unei noi concepții de conducere a robotului, care ține seama tocmai de *scopul conducerii robotului*.

Acest tip de conducere se numește “conducere senzorială” (sensory control – ISO 8373), iar arhitectura cea mai simplă a unui sistem robot care să prezinte o astfel de conducere este redată în figura 1.4.

Începând cu anul 1991 [Hash91a] a introdus pentru prima oară sintaxa de “visual servoing”. Acesta definește “sistemul servo-vizual” ca fiind un sistem de conducere globală (sistem de conducere cu buclă de reacție de la mediu) care se bazează în culegerea informațiilor pe sisteme vizuale.

Aceste tipuri de sisteme servo-vizuale se implementează în special la roboții autonomi care lucrează în medii necunoscute.

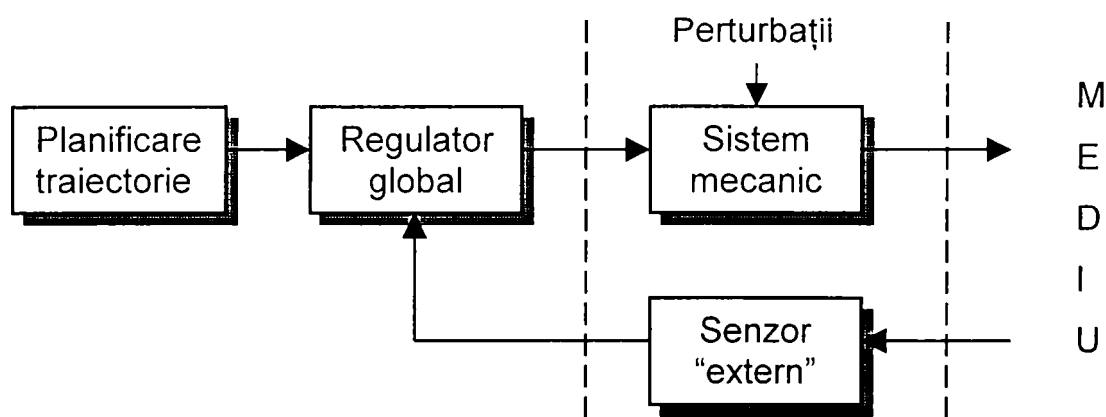


Figura 1.4. Arhitectura unui sistem ce prezintă o conducere senzorială

Conducerea mixtă presupune realizarea unui sistem mixt om-mașină, în care operatorul uman supervizează comanda automată, intervenind la momente bine precizate sau aleatoare, continuu/secvențial pentru a o completa sau corecta. De aici și denumirea acestei comenzi de *comandă de supervizare*.

Acest tip de conducere este întâlnit la roboții cu un număr de grade de mobilitate mare (6-8). În acest caz, spre exemplu operatorul uman comandă sincron mecanismul generator de traiectorie, iar mecanismul de orientare este condus automat.

La unii roboți mai avansați, există posibilitatea modificării automate a repartizării temporale a sarcinilor de conducere între operatorul uman și sistemul de conducere automat. Astfel, într-o primă fază, sistemul de control autonom permite conducerea sincronă a robotului, memorându-se elementele corespunzătoare traiectoriei și a modului de parcurgere a ei, precum și semnalele emise de senzori și respectiv de traductoare. La întâlnirea unor condiții similare în cadrul procesului tehnologic comanda este preluată de sistemul automat. În cazul în care, condițiile diferă de cele “învățate” anterior, comanda este cedată operatorului uman.

1.2. Tehnici de conducere a roboților

1.2.1. Principii

Deoarece conducerea automată este cea mai utilizată metodă de conducere în momentul actual, se vor trata în continuare câteva metode de conducere ce apar în diferite surse bibliografice.

În literatura de specialitate se întâlnesc două metode de conducere nemijlocită a mișcărilor și anume:

- 1) metode nonadaptive:
 - a) conducere cu reglarea mișcărilor la nivelul fiecărei cuple cinematice conducătoare (conducere în spațiul cuplelor cinematice conducătoare);
 - b) conducere în spațiul cartezian;
- 2) metode adaptive.

Oricare din metode este aplicată, problema conducerii nemijlocite a dispozitivului de ghidare constă în:

1. obținerea modelului dinamic al sistemului mecanic;
2. determinarea unor legi și/sau strategii de conducere, pe baza modelului dinamic obținut, cu care să se obțină răspunsuri și performanțe dorite sau impuse pentru efectatorul final al robotului;
3. validarea și ajustarea sistemului de reglare – în primă fază prin simulare, apoi experimental.

1.2.2. Conducerea nemijlocită a mișcărilor elementelor cuplelor cinematice conducătoare utilizând metode nonadaptive

Conducerea nemijlocită a mișcărilor elementelor cuplelor cinematice conducătoare cu reglare la nivelul fiecărei cuple cinematice conducătoare

În literatura de specialitate se prezintă mai multe metode de conducere nemijlocită a mișcărilor elementelor cuplelor cinematice conducătoare cu reglare la nivelul fiecărei cuple cinematice conducătoare, dintre care cele mai cunoscute sunt următoarele:

1. conducere nemijlocită cu tratarea fiecărei cuple cinematice conducătoare ca un servomecanism independent (Vukobratovic (1985), Bogdanov (1989));
2. conducere nemijlocită cu reglare după cuplu – Computed Torque Control (CTC) (Lee, Paul (1972), Bejczy (1973));

3. conducere nemijlocită cu reglare cu timp de răspuns minim – Time Optimal Control (Geering, Hepner, Guzzella; Onder (1986));
4. conducere nemijlocită cu regulator cu structură variabilă – Variable Structure Control (VSC);
5. conducere nemijlocită cu reglare folosind decuplarea neliniară [Dragomir].

Strategii de conducere cu prescrierea mișcării în coordonate operaționale

Metodele cele mai cunoscute în literatura de specialitate sunt:

1. conducere utilizând matricea de conducere $D(r)$, $r = \frac{1}{T}$;
2. comanda în viteză – Resolved Motion Rate Control (RMRC) (Whitney (1969), Luh);
3. comanda în accelerație – Resolved Motion Acceleration Control (RMAC) (Luh, Walker, Paul (1980));
4. comanda în cuplu – Resolved Motion Force Control (RMFC) (Wu, Paul).

Deoarece toate aceste metode de conducere nemijlocită a mișcărilor relative ale elementelor cuplelor cinematice conducătoare, atât cele cu reglare la nivelul fiecărei cuple cinematice conducătoare, cât și cele cu prescrierea mișcării în coordonate operaționale sunt amplu dezbătute în literatura de specialitate ele nu mai sunt reluate în acest paragraf.

1.2.3. Strategii de conducere specifice roboților adaptivi

1.2.3.1. Definiții

În cadrul clasificării roboților – în funcție de tipul de conducere a acestora - dată în [WIR97] o clasă distinctă este materializată de *roboții adaptivi*, pe care aceeași sursă bibliografică îi definește astfel:

Definiție. Robotul adaptiv este acel robot care prezintă funcții specifice conducerii senzoriale, conducerii adaptive sau conducerii prin învățare.

Exemple de astfel de funcții sunt corectarea și schimbarea comportării pe baza informațiilor senzoriale sau a experienței acumulate, planificarea și programarea sarcinilor de executat și instruirea și învățarea din comportări iterative.

Un robot adaptiv tipic este acela al unui robot echipat cu vedere artificială în vederea realizării corecțiilor de poziție la apucarea unor obiecte sau a unui robot cu arc de sudură echipat cu funcții de urmărire continuă a cordonului de sudură.

Definiție. Prin *conducere senzorială* se înțelege acea schemă de conducere în cadrul căreia mișcarea robotului sau forța este ajustată în raport cu mărimile de ieșire de la un senzor extern.

Definiție. Prin *conducere adaptivă* se înțelege acea schemă de control în cadrul căreia controlul parametrilor sistemului sunt ajustați din condițiile detectate în timpul procesului.

Definiție. Prin *conducere prin învățare* se înțelege acea schemă de control în cadrul căreia experiența obținută în cadrul ciclurilor anterioare este automat folosită pentru schimbarea parametrilor de control și/sau a algoritmilor.

Aceste definiții sunt preluate din ISO 8373.

1.2.3.2. Robot adaptiv cu funcții specifice conducerii senzoriale

Robotul al cărui sistem de conducere ajustează în timp real situarea efectorului final sau forța funcție de mărimile de ieșire utilizând în acest sens un senzor extern poartă numele de *robot adaptiv cu funcții specifice conducerii senzoriale*.

În general, la roboții de acest tip senzorul extern se materializează fie printr-un senzor vizual – atunci când se controlează mișcarea robotului – și în acest caz se vorbește de sisteme servo-vizuale, fie printr-un senzor de forță moment – atunci când se urmărește conducerea în forță a robotului.

La sistemele servo-vizuale – *visual servoing* - bucla de reacție vizuală se bazează fie pe poziție, fie pe imagine.

În primul caz – “*traditional*” *visual servoing* – se calculează poziția punctului caracteristic al robotului în spațiul cartezian utilizând imaginile furnizate de senzori și transformata vizuală, aceasta constituind “reacția” de buclă [Hash91b], [Cork94].

Structura generală a unui sistem servo-vizual tradițional adaptat după [Cork94] se prezintă în figura 1.5.

Această metodă necesită determinarea a două funcții de transformare – h , g – fie printr-o cunoaștere a-priori, fie printr-un proces de calibrare. Prima funcție h este cea a modelului sistemului vizual sau funcția de calibrare a sistemului vizual, după cum ea este determinată și ea realizează trecerea de la spațiul imagine la un sistem de coordonate cartezian (de obicei cu centrul situat în baza robotului). Cea de-a doua funcție de transformare g este de fapt un set de funcții obținute din analiza cinematică inversă aplicată robotului în discuție, această funcție transformând poziția punctului caracteristic al efectorului final - dată în coordonate carteziane - în coordonate generalizate la nivelul

fiecărei cuple cinematice conducătoare. Cu alte cuvinte, funcția g face trecerea de la spațiul tridimensional la spațiul coordonatelor generalizate corespunzătoare cuplelor cinematice conducătoare. Precizia de poziționare a robotului depinde de exactitatea acestor funcții.

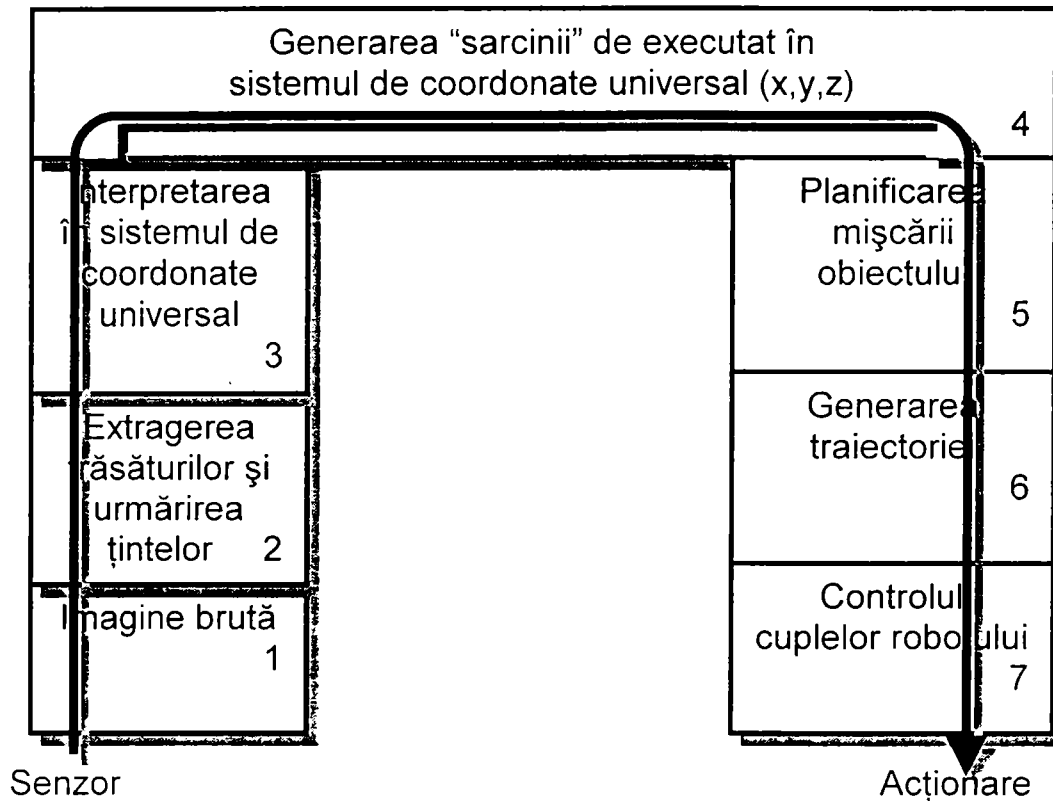


Fig.1.5. Structura generală a unui sistem servo-vizual "tradițional"

Din compunerea acestor două funcții $g(h^*)$ se obține funcția de trecere de la spațiul imagine la spațiul coordonatelor generalizate corespunzătoare cuplelor cinematice conducătoare, funcție ce poate fi folosită în conducerea robotului pornind de la imaginile obținute cu ajutorul senzorilor vizuali externi.

Weiss observă că precizia poate fi îmbunătățită utilizând erorile vizuale reziduale, măsurate după ce s-a executat o mișcare, dacă se fac corecțiile necesare înlăturării acestor erori. Această metodă e denumită în [Weiss84] "*position based (iterated) look-and-move*". Sugestiv această metodă se prezintă în figura 1.6.

Următorul pas a fost acela de a observa că dacă punctele țintă sunt oferite prin coordonatele din spațiul imagine și ele sunt atinse prin acționarea cuplelor cinematice conducătoare, atunci sistemul de coordonate cartezian legat de baza robotului poate fi complet eliminat, utilizând direct funcția compusă $g(h^*)$.

În [Weiss84] această tehnică poartă numele de "*image based look-and-move*" și este prezentată în figura 1.7.

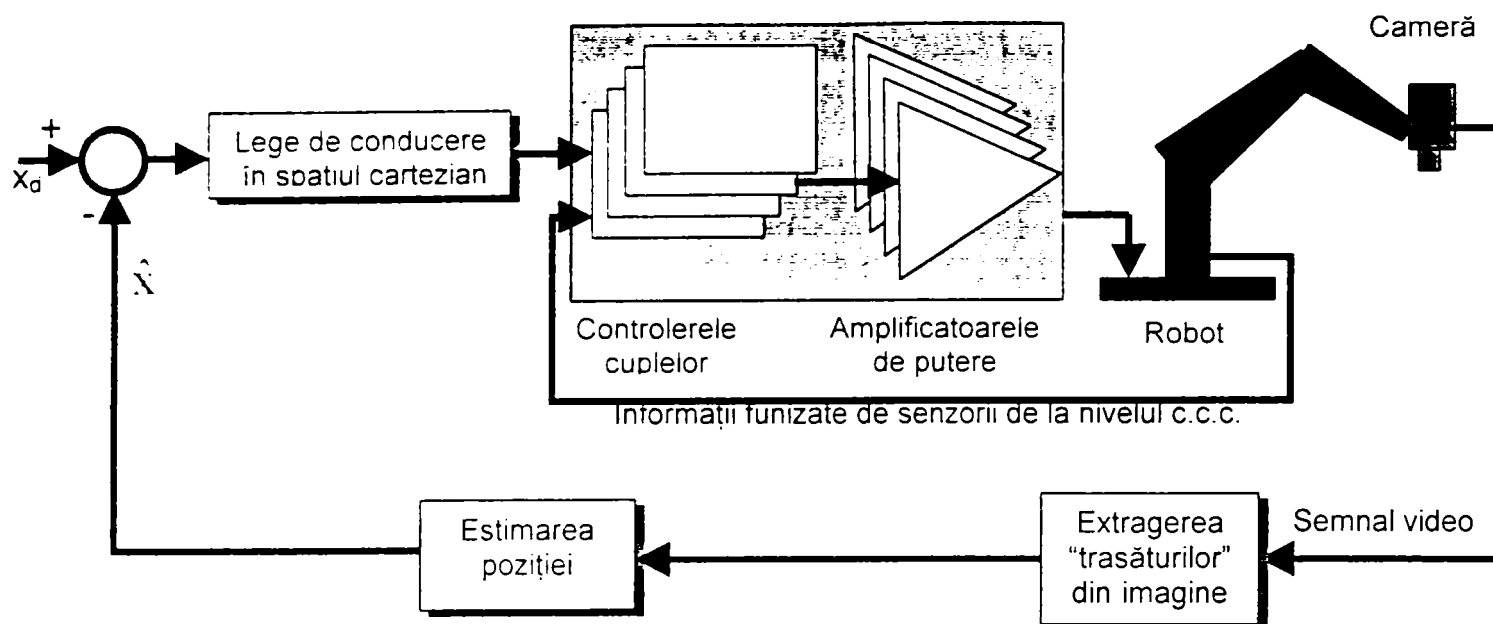


Fig. 1.6. Metoda "position based (iterated) look-and-move" [Weis84]

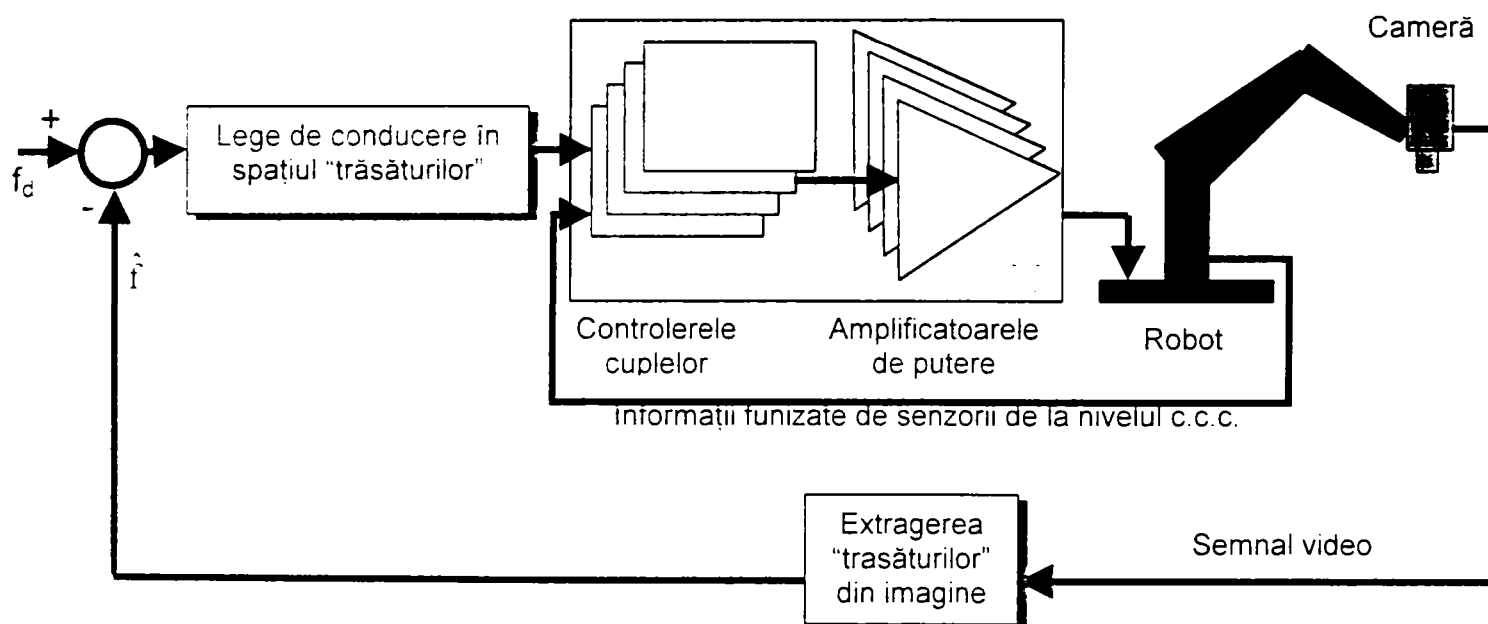


Fig.1.7. Metoda de conducere "image based look-and-move"

Dacă în locul utilizării ocazionale a erorii vizuale reziduale sistemul de comandă de la nivelul fiecărei cuple cinematice conducătoare se folosește de informația preluată de la un semnal vizual continuu, apare în primul caz al conducerii bazate pe poziție aceea tehnică numită de Weiss "position-based visual servoing" (PBVS) – figura 1.8, iar în cazul conducerii bazate pe imagine tehnica "image based visual servoing" (IBVS), figura 1.9.

La această a doua variantă care este întâlnită în literatura de specialitate sub numele de - "uncalibrated" visual servoing - comanda se realizează direct pe baza

“trăsăturilor” (seturi de puncte, muchii ale robotului) detectate în imaginile furnizate de CCD-uri [Hash94], [Hoso94], [Hoso97], [Fedd94], [Bien94], [Khad95].

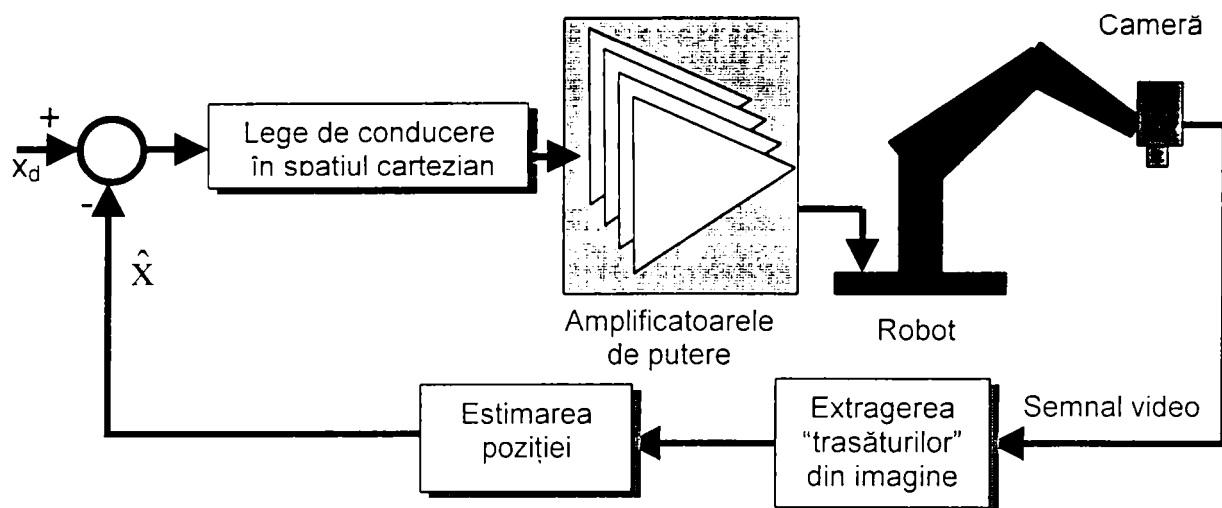


Fig. 1.8. Metoda de conducere a roboților “*position-based visual servoing*” (PBVS)

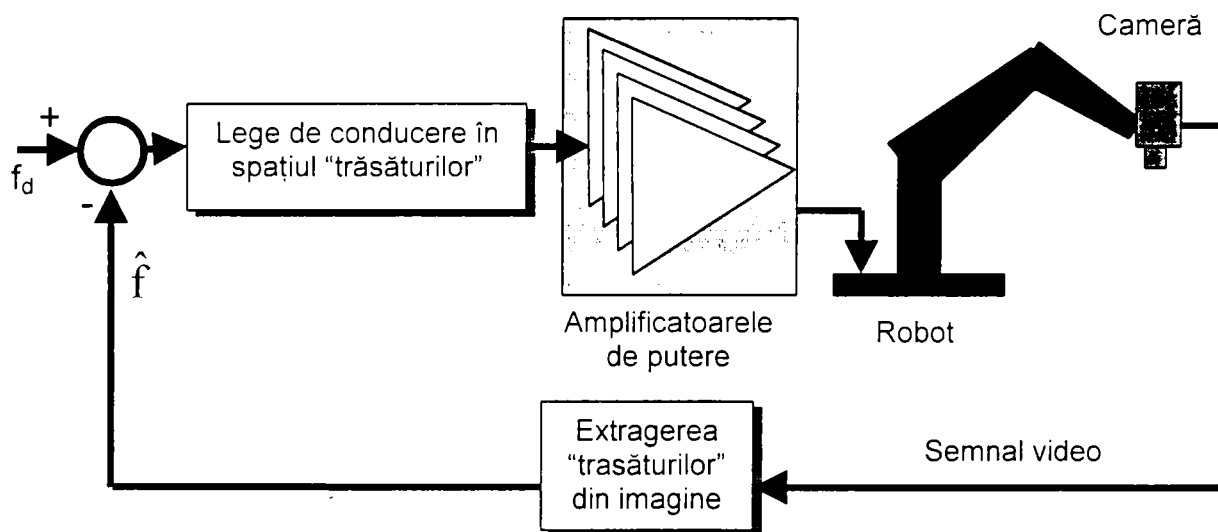


Fig. 1.9. Metoda de conducere a roboților “*image-based visual servoing*” (IBVS)

Deoarece această variantă este în fapt o metodă ce folosește procesul de învățare, ea se tratează în paragraful 1.2.3.4.

În [Cork94] sunt prezentate câteva din aplicațiile comenzii robotice cu reacție vizuală și anume: servirea benzilor rulante, mișcarea ghidată vizual, asamblarea, sudura, aplicații militare pentru urmărirea automată a rachetelor, interceptarea unor obiecte aflate în zbor, conducerea unor roboți subacvatici [Nega91], mașini de cules fructe [Harr89], cuplarea unui conector al rampei de lansare a navetei spațiale, etc.

1.2.3.3. Robot adaptiv cu funcții specifice conducerii adaptive

Un sistem este *adaptiv* dacă este capabil să atingă obiectivele conducerii în situația în care informația privitoare la structura și parametrii modelului matematic al procesului, inițial disponibilă, nu este completă.

Prin urmare, necesitatea dezvoltării conducerii adaptive a roboților cu funcții specifice conducerii adaptive a rezultat din dorința de a utiliza roboții în condiții de lucru în care parametrii și structura procesului nu rămân fixe, cu alte cuvinte de a mări gradul de flexibilitate a sistemului în care operează acesta.

Realizarea conducerii adaptive presupune o reacție de feed-back, astfel încât robotul adaptiv să-și completeze informația despre proces în timp real, pe baza informației de ieșire a sistemului condus, printr-o recunoaștere on-line a procesului.

Prelevarea informației necesare pentru reacția de feed-back este realizată de senzori și traductoare, traductorul achiziționând informație din interiorul sistemului robot, iar senzorii din mediul înconjurător.

Modalitățile de conducere adaptivă sunt diverse, fiind punctate în continuare două metode cel mai des întâlnite în literatura de specialitate:

1. Conducere adaptivă utilizând model de referință – Model Reference Adaptive Control (MRAC).
2. Sisteme de conducere autoacordabile – Self-Tuning Methods.

În domeniul conducerii adaptive cu model de referință primele lucrări au fost cele ale lui Dubowsky, respectiv DesForges (1979) în care dinamica robotului a fost tratată ca un integrator dublu linear decuplat. În 1981 Arimoto și Takegaki propun o metodă prin care modelul se linearizează de-a lungul traiectoriei nominale, iar algoritmul adaptiv se obține prin optimizarea parametrilor locali. Balestrino, De Maria, Sciavico propun în 1983 un algoritm adaptiv "vector unitate", în care însă dinamica acționării este neglijată. Însă în 1986 De Maria, Sciavico și Siciliano publică o lucrare de conducere adaptivă discretă cu model de referință, în care iau în considerare și dinamica acționărilor, care după compensarea gravitației o modelează printr-un sistem de ordin doi. În 1986 Hsia propune un sistem adaptiv de conducere cu model de referință în care presupune că matricea de stare a modelului de referință este egală cu matricea de stare a dinamicii robotului, în care a introdus și informații privind dinamica acționării. Cercetări în acest domeniu au fost făcute de către Vukobratovic, Kircanski (1985), ele fiind prezentate în [Vuk85a], [Vuk85b], [Vuk85c].

Principial, există două metode de realizare a conducerii adaptive cu model etalon (de referință) și anume:

Principiul I: se stabilește un *model de referință* pentru sistemul de reglare de bază al axei robotului; se concepe un *algoritm de adaptare*, care modifică parametrii regulatorului din sistemul de reglare de bază.

Schema bloc al acestui tip de conducere este prezentată în figura 1.10.a.

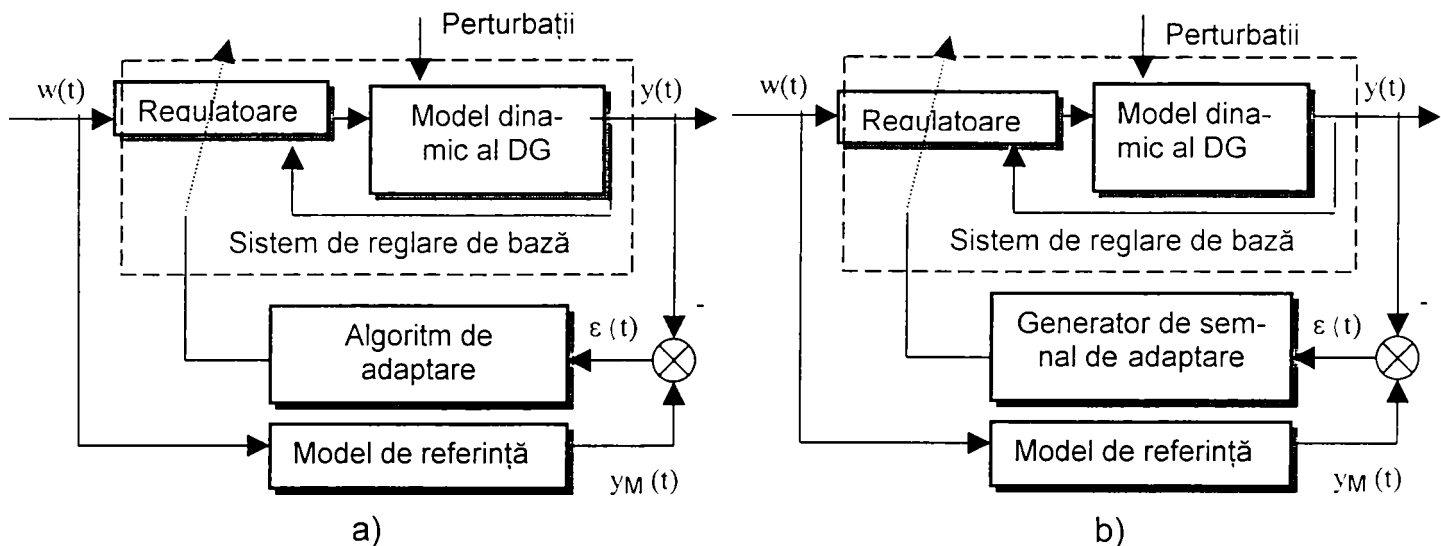


Fig.1.10. Scheme bloc ce utilizează cele două principii de conducere adaptivă cu model etalon

Se face observația că algoritmul de adaptare este comandat cu erorile dintre mărimile de ieșire ale modelului și cele de la ieșirea sistemului de bază.

Principiul II: se stabilește un model de referință pentru sistemul de reglare de bază al robotului; se generează un semnal de adaptare, care să facă dinamica erorii asimptotic stabilă (nulă).

Schema bloc pentru tipul de conducere care utilizează principiul II în conducere este prezentată în figura 1.10.b.

Metode de conducere autoacordabile au fost abordate de un mare număr de cercetători. Metoda presupune o estimare a modificării parametrilor urmată de realizarea controlului acționărilor, figura 1.11.

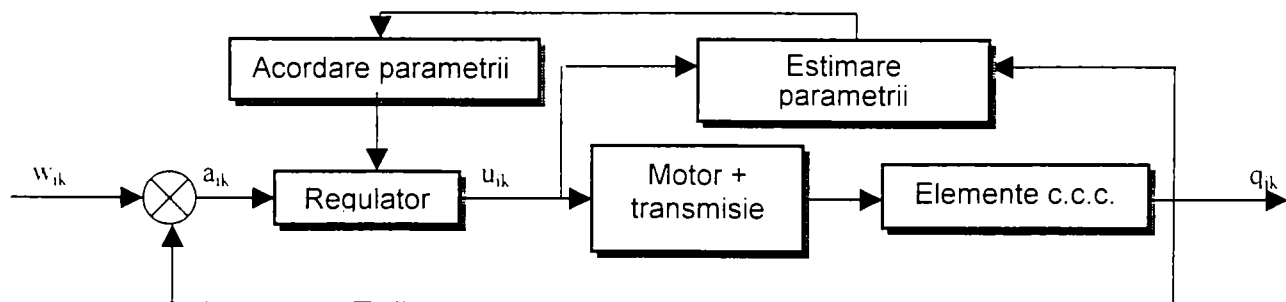


Fig.1.11.Schema bloc pentru un sistem de conducere autoacordabil

Pentru estimarea parametrilor cele mai utilizate metode sunt: metoda celor mai mici pătrate, metoda recursivă a celor mai mici pătrate, metoda gradientului.

1.2.3.4. Robot adaptiv cu funcții specifice conducerii prin învățare

Pornindu-se de la definiția *conducerii prin învățare* prin care se înțelege acea schemă de control în cadrul căreia experiența obținută în cadrul ciclurilor anterioare este automat folosită pentru schimbarea parametrilor de control și/sau a algoritmilor, cel mai elocvent exemplu al unei conduceri de acest tip îl constituie metoda *"uncalibrated" visual servoing* [Jäge94].

Jägersand și Nelson de la Universitatea din Rochester în [Jäge94] oferă câteva caracteristici ale acestei variante:

a) ea nu necesită cunoștințe *a priori* despre parametrii sistemului sau despre structura cinematică a sistemului. De aici rezultă și avantajul de a nu mai fi nevoie de un proces de calibrare a sistemului și în plus nu mai este necesară separarea parametrilor necunoscuți din sistemul de ecuații ce rezultă din analiza structurii sistemului senzori vizual-robot.

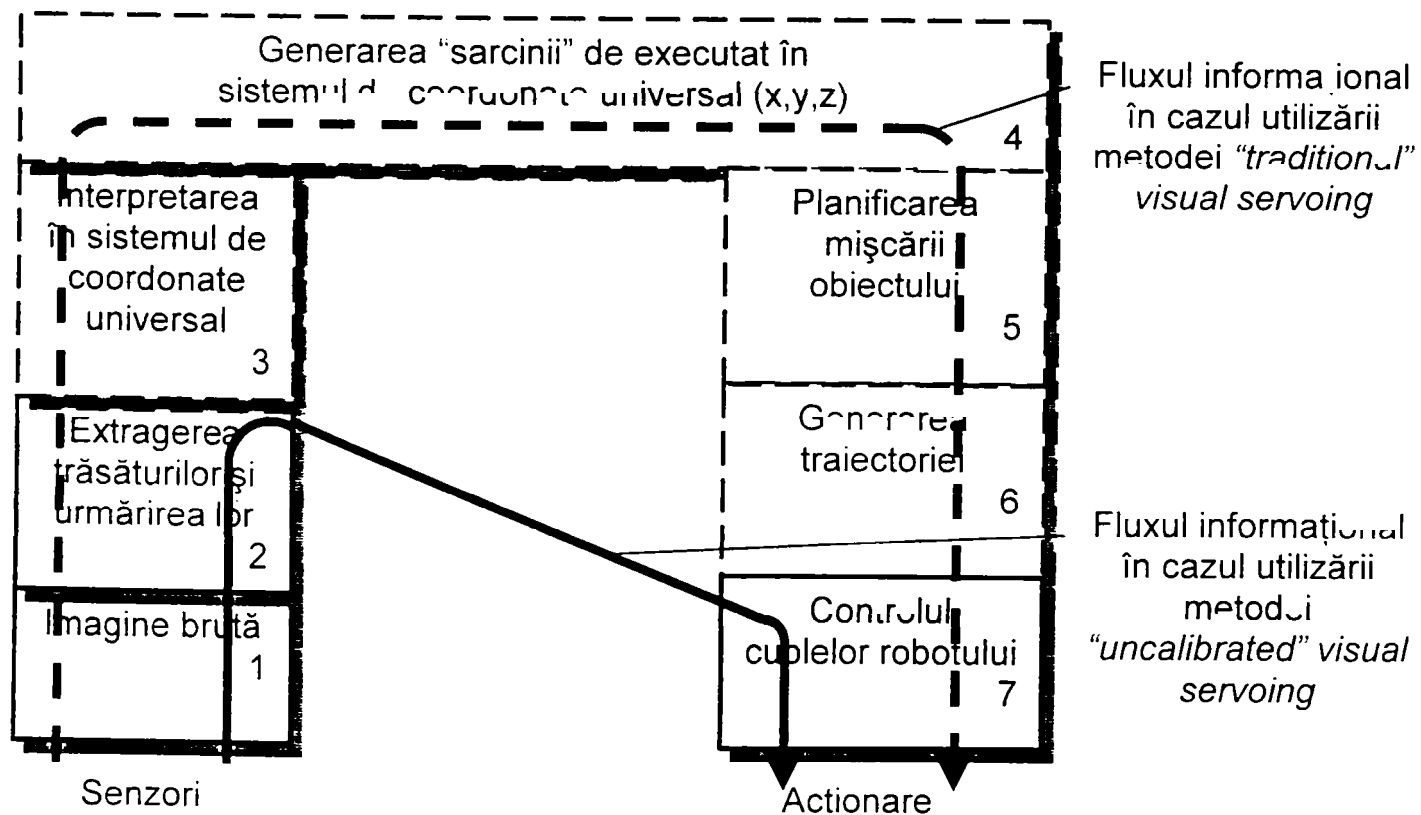


Fig.1.12. Comparație între structura generală a metodei *"traditional" visual servoing* și a metodei *"uncalibrated" visual servoing*

b) nu există restricții de aplicabilitate a sistemului senzor vizual-robot: numărul de senzori vizuali, tipul de trăsături a imaginii, structura sistemului (*hand-eye coordination* sau *camera-and-manipulator*), numărul de intrări sau ieșiri, etc.

Dacă se face o comparație între prima metodă *“traditional” visual servoing* și cea de-a doua *“uncalibrated” visual servoing* se observă că blocurile 3-6 din figura 1.5. dispar deci se poate vorbi de o limitare a lanțului de calcule în cea de-a doua variantă, în schimb este necesară obținerea seturilor de *“trăsături”* („features”).

În figura 1.13. se prezintă componentele constructive necesare realizării conducerii utilizând metoda *“uncalibrated” visual servoing*.

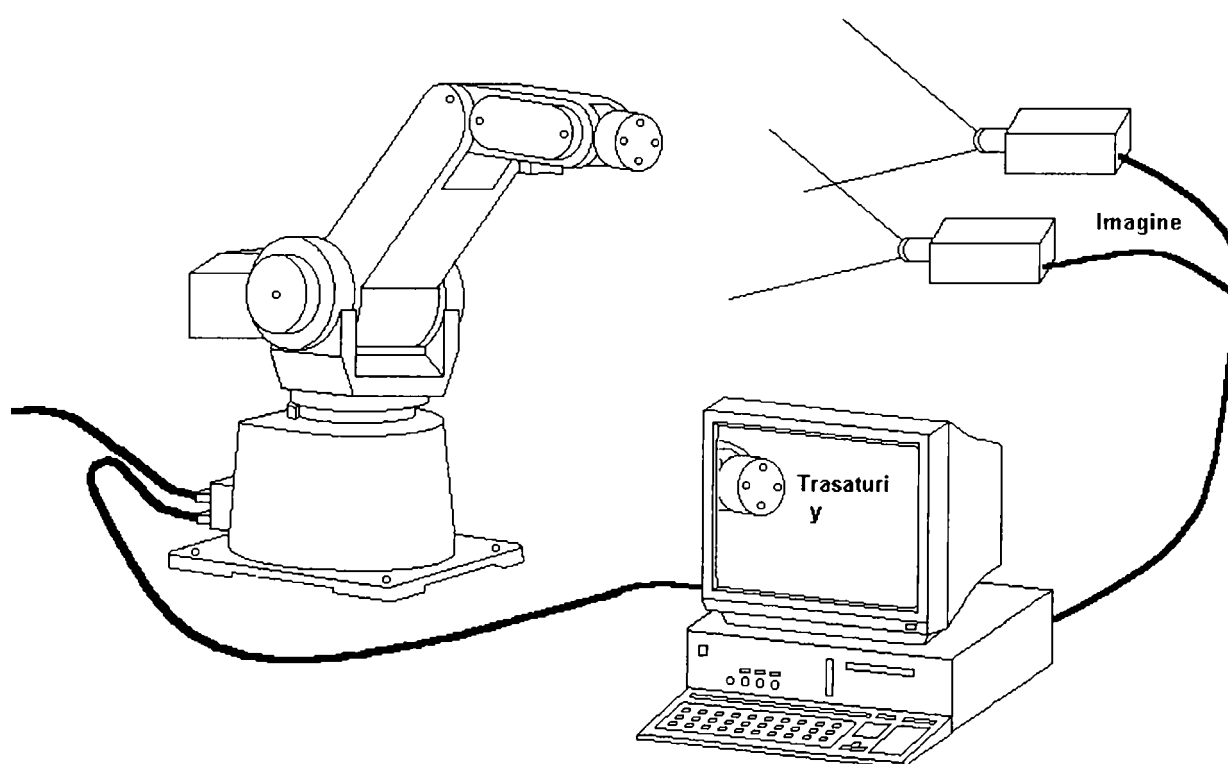


Fig.1.13. Modul de lucru al metodei *“uncalibrated” visual servoing*

Dacă se realizează conducerea unui robot cu trei sau mai multe cuple cinematice conducătoare se vor utiliza doi senzori vizuali.

Câteva variante de realizări pentru un astfel de sistem vizual stereoscopic – întâlnit în literatura de specialitate sub denumirea de cap *“Yorick”* se prezintă în figurile 1.14.

Astfel, în figura 1.14.a) se prezintă un sistem vizual stereoscopic de tip *“eyes in hand”*. Se observă independența mișcărilor de rotație în jurul axei orizontale la cei doi senzori vizuali. În figura 1.14.b) se observă un sistem vizual stereoscopic fix, similar din punct de vedere al mișcărilor ce le poate executa cu cel prezentat în fig 1.14.a)

În figura 1.14.c) cei doi senzori vizuali sunt antrenați în jurul axei orizontale de către un singur motor, în schimb se observă posibilitatea unei mișcări suplimentare a întregului sistem – o mișcare similară cu cea a gâtului uman. În figura 1.14.d) se prezintă un cap

“Yorick” amplasat pe o platformă mobilă. Se remarcă cuplarea mișcării de rotație în jurul axei orizontale a senzorilor vizuali.

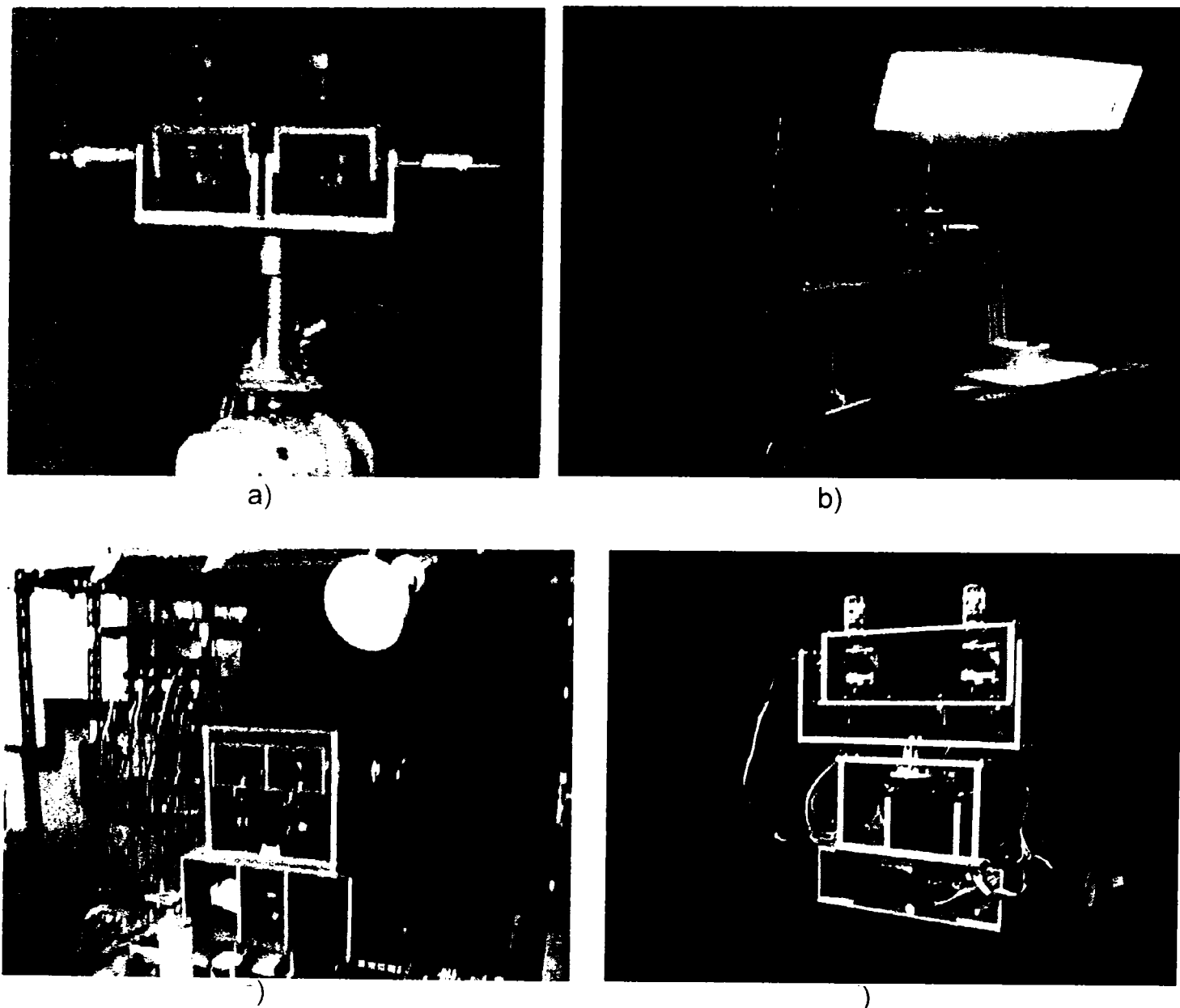
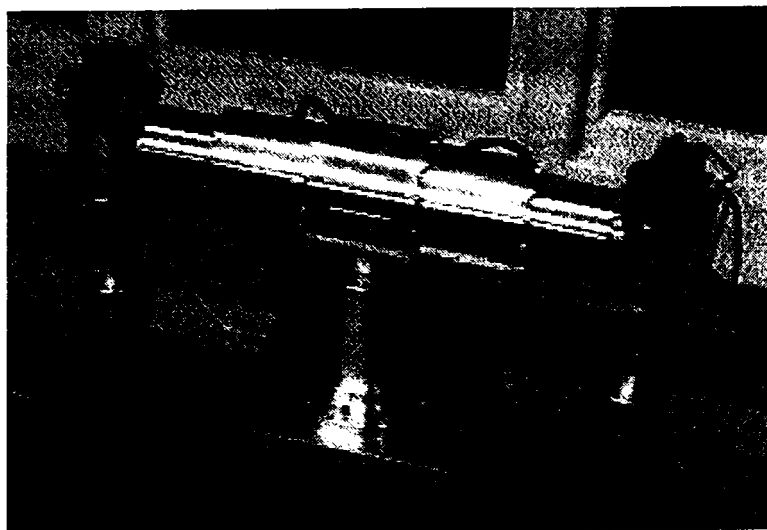


Fig.1.14. Variante de sisteme vizual stereoscopice

Se face observația că primele trei realizări sunt de la Miyazaki Lab, Universitatea din Osaka, Japonia, în timp ce sistemul vizual stereoscopic prezentat în figura 1.14.d) a fost realizat la Asada Lab, Universitatea din Osaka, Japonia.

În figurile 1.15 se prezintă realizările Universității Oxford, Regatul Unit al Marii Britanii în domeniul sistemelor vizuale stereoscopice. Astfel s-au realizat trei capete “Yorick”: 11-14; 8-11; 5-5C.

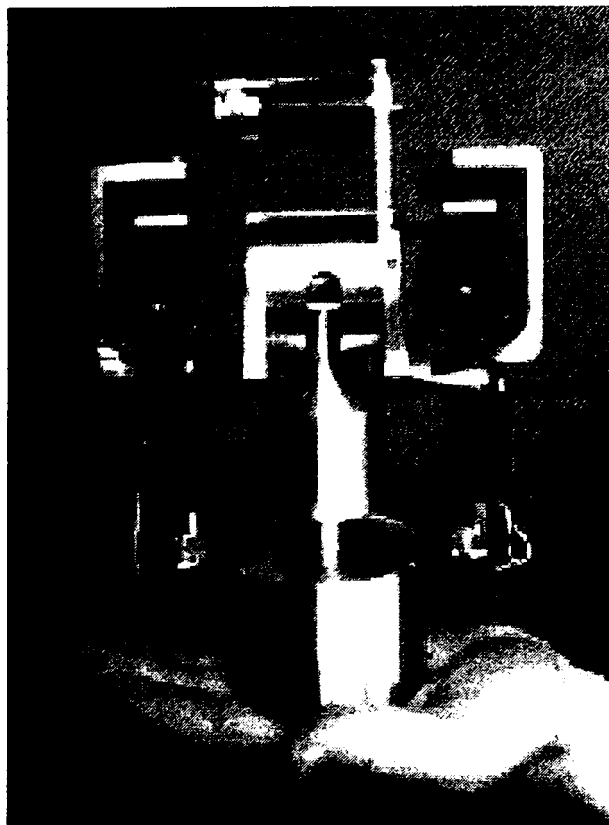
În tabelul 1.2. se prezintă caracteristicile capetelor “Yorick” realizate la Universitatea din Oxford.



a)



b)



c)

Fig.1.15. Capetele "Yorick"-
Universitatea din Oxford
a)Yorick 11-14;
b)Yorick 8-11;
c)Yorick 5-5C.

Tabelul 1.2. Caracteristici ale capetelor "Yorick" realizate la Universitatea din Oxford

Caracteristici	Yorick 11-14			YORICK 8-11			YORICK 5-5C		
	Roll	Pitch	Yaw	Roll	Pitch	Yaw	Roll	Pitch	Yaw
Unghiul de rotație max. (grd)	360	360	360	178.5	177.5	335	28.5	90	235
Acceler. max. (grd/s/s)	6000	5000	3000	38000	25000	20000	28750	28125	19500
Viteza max. (grd/s)	400	400	240	540	470	430	560	680	425
Viteză minimă (grd/s)	0.027	0.027	0.014	-	-	-	0.027	0.027	0.014
Linia de bază	550mm			338mm			110mm		
Greutate	-			8.5kg (exclusiv senzorii)			2kg (cu senzori)		
Sarcină	2 x 5kg			2 x 500g			-		
Repetabilitate	0.003125 până la 0.01055 grd								

Rolul de achiziție a imaginii țintei în mișcare – obiectul de manipulat - revine celor doi senzori vizuali matriciali. Mișcarea de urmărire a țintei de către cei doi senzori vizuali este comandată de la cele trei sau patru acționări cu care este prevăzut sistemul vizual stereoscopic.

La noi în țară la Universitatea Politehnica București s-a construit – sub două variante – un senzor de urmărire a unei ține luminoase de către dr.ing. Constantin Bucșan, senzor prezentat în figura 1.16.

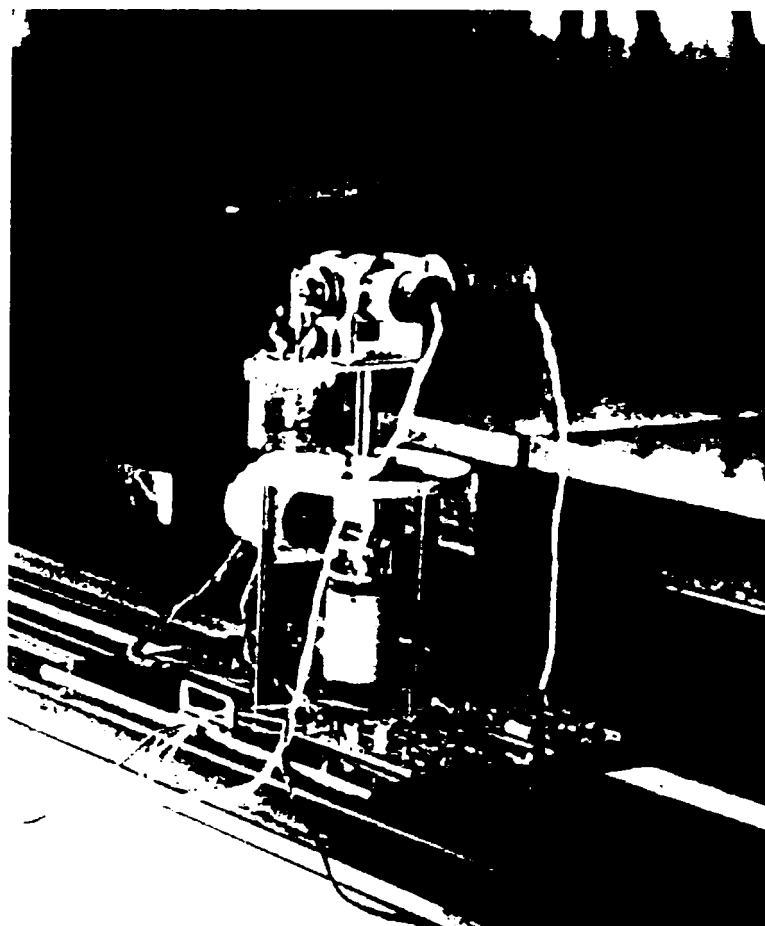


Fig.1.16. Senzor de urmărire a unei ținte luminoase realizat la Universitatea Politehnica București

Sistemul de urmărire prezintă două grade de mobilitate și utilizează în locul sensorului vizual un element fotosensibil de tip ROL 210 de formă circulară împărțit în patru zone fotoconductive. Prin utilizarea microcurenților generați de cele patru zone conductive ale sensorului se obține semnalul necesar urmăririi în timp real al țintei luminoase.

Motoarele de acționare a sistemului de urmărire realizat de colectivul Catedrei de Mecanică Fină a Universității Politehnica București sunt în cazul primei variante constructive a sensorului motoare pss cu pas, iar în cazul celei de a doua variantă motoare de curent continuu.

Din imaginile achiziționate de senzori se extrag și se urmăresc continuu situațiile trăsăturilor reprezentative – de exemplu situarea efectorului final al robotului.

Dacă schimbările în imaginile vizuale sunt înregistrate ca un vector $\underline{y} = (y_1 \dots y_m)^T$ ce în [Jäge96] este numit vector al trăsăturilor sau vector de percepție și dacă se notează cu $\underline{x} = (x_1 \dots x_n)^T$ vectorul semnal de ieșire de la controlerle axelor (și care în cazul cel mai simplu poate fi vectorul de comandă a mișcărilor relative a elementelor cuplelor cinematice conducătoare), atunci pornind de la vectorul trăsăturilor la deplasările de executat în spațiul cuplelor cinematice conducătoare sistemul va "învăța" realizând o mapare între vectorul \underline{y} al trăsăturilor și vectorul \underline{x} . Sistemul nu necesită un model prealabil a funcției de transfer $\underline{y} = f(\underline{x})$, el învață modelul în timpul realizării sarcinilor fără o introducere explicită a unor pași suplimentari de învățare.

1.3. Cercetători și grupuri de lucru care lucrează în visual servoing

În lista următoare se prezintă cercetătorii care au preocupări și realizări deosebite în domeniul "visual servoing" precum și departamentele, respectiv universitățile în care ei activează:

- ☞ Asada Lab, University of Osaka, Japonia;
- ☞ Bill Wilson, University of Waterloo, Canada;
- ☞ Pradeep Khosla, Brad Nelson, CMU, SUA;
- ☞ Dan Koditschek, University of Michigan, SUA;
- ☞ Greg Hager, Yale Vision and Robotics Group, SUA;
- ☞ Koichi Hashimoto, Okayama University, Japonia;
- ☞ Claude Samson, Patrick Rivers, INRIA, Sophia-Antipolis, Franța;
- ☞ Miyazaki Lab, University of Osaka, Japonia;
- ☞ Nikos Papanikolopoulos, University of Minnesota, SUA;
- ☞ Peter Corke, CSIRO, Australia;
- ☞ Peter Allen, Shree Nayar, Robotics Laboratory at Columbia University, SUA;
- ☞ Seth Hutchinson, University of Illinois, Urbana-Champaign, SUA;
- ☞ Chris Brown, Martin Jagersand, University of Rochester, SUA;
- ☞ Giulio Sandini, University of Genova, Italia;
- ☞ Uchida Lab, Waseda University, Japonia.

La noi în țară, în principalele centre universitare sunt colective care au preocupări și realizări în domeniul visual servoing, dintre ele amintim:

☞ Catedra de Mecanică Fină, Universitatea Politehnică București: Prof.dr.ing. Alexandrescu Nicolae, Conf.dr.ing. Bucșan Constantin (senzor pentru determinarea direcției acționat cu motor de curent continuu).

☞ Catedra de Măsurări și Electronică Optică, Catedra de Electronică Aplicată, Universitatea Politehnică Timișoara, Prof. Dr. Ing. Eugen Pop, Prof.Dr.Ing. Virgil Tiponuț, Dr. Ing. Andrei Cimponeriu;

☞ Catedra de Calculatoare, Universitatea Tehnică Cluj-Napoca, Prof.Dr.Ing. Sergiu Nedevschi.

CAPITOLUL 2 Scopul și obiectivele lucrării

În cadrul tezei de doctorat cu tema "Utilizarea informațiilor cu privire la situarea efectorului final achiziționate prin senzori în vederea conducerii în timp real a roboților" se dorește punerea la punct a unui sistem de conducere al roboților, la care prelevarea informației de situație a efectorului final să fie făcută de un dispozitiv exterior robotului.

După cum bine se știe, în momentul de față, roboții prezintă la nivelul fiecărei cuple cinematice conducătoare (c.c.c.) traductoare ce preiau informația referitoare la starea internă a sistemului robot (cantitate de mișcare relativă a elementelor componente a DG - translație sau rotație, viteza relativă generalizată a elementelor, mărimea torsorului de la nivelul c.c.c.), pe care o transmit, prin intermediul unei interfețe, sistemului de comandă.

Acesta, în funcție de modalitatea de conducere a robotului (în poziție, viteză, forță, etc.) utilizează informațiile oferite de traductoarele de la nivelul fiecărei cuple cinematice conducătoare o compară cu cea programată rezultând o eroare. La acel moment peste comenzile necesare de deplasare a efectorului final pe traiectorie, trebuie suprapuse comenzile de corecție în vederea minimizării erorii.

În cazul în care prelevarea informației situației efectorului final se face de un dispozitiv exterior robotului, se necesită doar compararea acestei situații cu cea programată, comanda corecției erorii de situație suprapunându-se peste comenzile necesare deplasării.

În general, cercetările și realizările tehnice privind urmărirea în timp real a situației și mișcării efectorului final al roboților industriali s-au făcut și se fac în scopul conducerii roboților sau a încercării acestora.

Scopul lucrării îl reprezintă dezvoltarea tehnicilor de conducere în timp real a robotului utilizând în acest sens informațiile oferite de către senzori vizuali.

Pentru atingerea acestui scop s-au avut în vedere următoarele obiective fundamentale:

- elaborarea unui studiu al stadiului actual al realizărilor în domeniul conducerii în timp real a robotului folosind informațiile sistemelor senzoriale;

- sintetizarea într-un tot unitar a tehnicilor de modelare a ansamblului – motor-transmisie mecanică-structură mecanică – robotului;
- elaborarea unor noțiuni legate de formarea imaginii și achiziția ei, noțiuni care sunt relevante în conducerea globală a roboților utilizând sisteme senzoriale vizuale;
- conceperea unor sisteme de conducere senzoriale în timp real a roboților utilizând informații furnizate de la senzori și anume: sistem robot la care comanda de corecție a efectorului final se realizează prin utilizarea informațiilor furnizate de un subsistem vizual de urmărire continuă și respectiv sistem robot la care comanda de corecție a efectorului final se realizează prin utilizarea informațiilor furnizate de un senzor vizual fix;
- elaborarea unei ordinograme originale privind modul de realizare a conducerii senzoriale a celor două sisteme robot, care utilizează informațiile furnizate de la subsistemul senzorial vizual, precum și tratarea unitară a aspectelor teoretice pentru cele două cazuri considerate;
- modelarea geometrică 3D a sistemului de conducere senzorială a robotului la care informațiile sunt prelevate de la subsistemul vizual de urmărire și simularea funcționării lui prin animație;
- conceperea modelului funcțional a sistemului de conducere senzorială a robotului și simularea funcționării lui;
- proiectarea și realizarea concretă a unui sistem de conducere senzorială a roboților utilizând informațiile furnizate de la un subsistem vizual de urmărire;
- proiectarea și realizarea concretă a unui sistem servo-vizual “tradițional” de tip “*position based (iterated) look-and-move*”. În vederea analizei, respectiv recunoașterii unor obiecte și de clasificare a acestora pe grupe;
- realizarea de experimente în vederea validării simulărilor;
- formularea unor concluzii privind utilizarea în timp real a sistemului de conducere senzorială a roboților utilizând informații furnizate de la un subsistem vizual de urmărire.

CAPITOLUL 3 Conducerea dinamică a robotului

3.1. Noțiuni introductive

Dinamica roboților tratează modul în care robotul se mișcă sub acțiunea unor forțe generalizate care fie sunt dezvoltate de motoarele de acționare, fie sunt forțe exterioare sistemului robot.

În cadrul analizei dinamice a roboților se necesită a fi tratate două probleme:

☞ *analiza dinamică directă* prin care se determină răspunsul în timp a situației elementelor, exprimată prin valorile coordonatelor generalizate ale unor puncte, la aplicarea unor forțe generalizate, obținute prin integrarea ecuațiilor dinamice ale mișcării robotului;

☞ *analiza dinamică inversă* prin care ecuațiile dinamice ale robotului sunt rezolvate pentru determinarea acelor forțe generalizate care să ofere mișcarea dorită robotului.

Literatura de specialitate propune mai multe metode de obținere a ecuațiilor dinamice de mișcare ale roboților. Primele cercetări în acest sens au fost făcute de către Uicker [Uick65] și Kahn [Kahn69] utilizând formalismul Lagrange. Datorită faptului că algoritmul de rezolvare a ecuațiilor obținute prin formalismul Lagrange este un algoritm cu *complexitatea în timp a algoritmului* de tip polinomial de grad patru $O(n^4)$ și cunoscându-se - conform [Iorg96] - că algoritmi cu complexitate polinomială de grad mai mare de doi sunt irealizabili în timp real, a rezultat imposibilitatea implementării acestora. Totuși, pentru a se putea realiza conducerea dinamică a robotului în timp real mulți autori au propus algoritmi de implementare a ecuațiilor dinamice simplificate, ecuații care neglijau anumiți termeni. Aproximarea cea mai des utilizată [Paul81], [Kova92] a fost aceea de a neglija componenta care ține seama de efectul Coriolis, de efectul centrifugal, efectul cuplării axelor realizându-se așa numitul *calcul dinamic aproximativ*. Alți autori [Khat87], [Shark90] s-au bazat în construirea algoritmului pe observația că modificarea coeficienților ecuațiilor dinamice este lentă, deoarece sunt în funcție de coordonatele generalizate și prin urmare determinarea lor (evaluarea ecuațiilor dinamice) se poate face la o rată destul de mică, rată care depinde de viteza de modificare a coordonatelor generalizate.

O abordare bazată pe metoda Newton-Euler a fost dezvoltată pentru prima dată de Orin, Vukobratovic ș.a. [Orin79]. Armstrong [Arms79] propune o metodă prin care demonstrează cum aplicând un algoritm recursiv se poate reduce complexitatea în timp a algoritmului la $O(n)$. Luh prezintă în [Luh80] un formalism Newton-Euler recursiv în care

vitezele generalizate sunt exprimate în sistemele de referință aferente fiecărei cuplă cinematice conducătoare. Prin metoda prezentată au reușit să reducă timpul de calcul de la 7.9 secunde cât era necesar pentru aplicarea formalismului Lagrange la 4,5ms și astfel aceasta a fost prima metodă care a putut fi implementată on-line. Hollerbach [Holl80] reușește să implementeze un algoritm recursiv și în cazul formalismului Lagrange, reducând astfel timpul de calcul, însă acesta rămânând de trei ori mai mare decât în cazul utilizării algoritmului Newton-Euler recursiv. Echivalența între formalismul Lagrange recursiv și formalismul Newton-Euler recursiv este demonstrată de către Silver [Silv82], autorul demonstrând că diferența în eficiența calcului rezultă din reprezentarea vitezelor generalizate în sistemele de referință aferente fiecărei cuplă cinematice conducătoare.

Formalismul Kane [Kane83] oferă o altă metodologie pentru obținerea ecuațiilor dinamice pentru un robot specificat. Metoda introduce un număr de "z" variabile care nu toate prezintă o semnificație fizică, dar care conduc la micșorarea sarcinii de calcul.

O altă metodă de deducere a ecuațiilor dinamice a roboților este cea a ecuațiilor Appel fiind elaborată de E.P.Popov și colaboratorii săi și dezvoltată în forma sa finală de M.Vukobratović și V.Potkonjak [Vuko85b]. Ea rezolvă atât problema dinamică directă cât și pe cea inversă. Aceste ecuații sunt bazate pe noțiunea lui Gibbs : "energia accelerației". Dinamica mecanismului robotului industrial este descrisă de ecuațiile Gibbs-Appel, conform cărora forțele generalizate exprimate în raport cu cupla cinematică conducătoare aferentă sunt egale cu derivatele parțiale ale "energiei accelerației" în raport cu accelerațiile generalizate.

În general teoriile uzuale de conducere a sistemelor fizice se bazează pe următoarele ipoteze [Cali88]:

- sistemul poate fi modelat și structura sa este cunoscută;
- parametrii modelului sunt cunoscuți.

Cele două ipoteze necesare pentru dezvoltarea unei teorii sistematice de conducere a sistemelor ridică numeroase probleme în practică. În aceste cazuri, modelul utilizat nu reprezintă decât o aproximare a comportării fizice a procesului, de multe ori o aproximare grosieră, valabilă într-un domeniu de funcționare restrâns, în care se neglijează, în scopul simplificării, efectul intrărilor adiacente.

De multe ori parametrii care intervin în reprezentarea aleasă trebuie determinați experimental și de obicei valorile acestora se presupun constante.

Obținerea și menținerea performanțelor sistemelor clasice de conducere - în condi-

țiile prezenței unor mărimi perturbatoare – la cote acceptabile este adesea un obiectiv greu de realizat.

În cazul particular al conducerii dinamice a unui robot sau manipulator aceste ipoteze sunt prezentate sugestiv [Coif92] în tabelul 3.1.

Dacă se descompune robotul (manipulatorul) în subsistemele constituate [Kova99a] se obține schema bloc prezentată în figura 3.1.

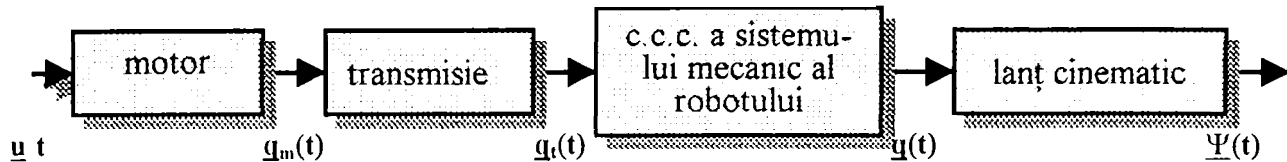


Fig.3.1. Descrierea funcțională a unui robot sub formă sistemică.

După cum se observă și din figura 3.1. la o mărime de intrare $\underline{u}(t)$ (în general această mărime fiind tensiunea sau curentul de alimentare în cazul acționărilor electrice și respectiv presiunea în cazul motoarelor hidraulice) corespunde o mărime de ieșire $\underline{\Psi}(t)$ (în general situarea punctului caracteristic al robotului) – în urma parcurgerii celor patru blocuri:

☞ motor, la ieșirea căruia rezultă un semnal – parametru pozițional relativ generalizat $\underline{q}_m(t)$ în funcție de natura acționării;

☞ transmisie mecanică care are rolul de a asigura adaptarea caracteristicii mecanice a motorului de acționare la cerințele cuplei cinematice conducătoare, semnalul de ieșire $\underline{q}_t(t)$ necesitând a fi riguros proporțional cu cel de intrare $\underline{q}_m(t)$;

☞ ansamblul structurii mecanice care asigură situarea $\underline{\Psi}(t)$ a efectorului final al robotului.

Tabelul 3.1. Ipoteze uzuale pentru modelul dinamic al robotului

Descriere funcțională				
	$\underline{u}(t)$	$\underline{q}_m(t)$	$\underline{q}_t(t)$	$\underline{\Psi}(t)$
Utilizând ipoteze pentru modelul static, coeficienți	(\underline{K}_1)	(\underline{K}_2)	(\underline{K}_3)	utilizând transformări de coordonate
Robot real	- depinde de motor - ecuații diferențiale liniarizate cu aservirea (fata de o alta ecuație) lua-tă în considerare	- inerție; - elasticitate; - frecare uscată; - frecare vâscoasă; - jocuri.	- inerție variabilă; - flexibilitate; - centre de rotații descentrate; - bucle mecanice.	- posibilitate de flexibilitate a elementelor lanțului cinematic.
Ipoteze uzuale pentru modelul dinamic, coeficienți	(\underline{K}_1)	(\underline{K}_2)	- rotații, translații perfecte; - rigiditate perfectă; - lanț cinematic fără bucle mecanice. Se determină relația dintre $\underline{q}(t)$ și $\underline{q}_t(t)$.	- rigiditate perfectă; - modelul geometric $\underline{x}=f(\underline{q}_t)$ rămâne valid.

După cum se observă din tabelul 3.1. utilizând ipotezele pentru modelul static motorul, transmisia mecanică și respectiv sistemul mecanic al robotului pot fi modelate prin matricele cu coeficienți constanți \underline{K}_1 , \underline{K}_2 , respectiv \underline{K}_3 . Transformările de coordonate de la $\underline{q}(t)$ la $\underline{x}(t)$ se realizează ținând cont de geometria lanțului cinematic al robotului.

Dacă se analizează cu atenție aceste modelări ale componentelor și se compară cu realitatea se pot trage următoarele concluzii:

- ☞ semnalul $\underline{q}_m(t) = \underline{K}_1 \cdot \underline{u}(t)$ depinde puternic de natura acționărilor; în realitate, se obține un sistem de ecuații diferențiale. De multe ori însă se preferă a se considera doar o relație de proporționalitate între tensiunea/curentul de comandă și forța generalizată dezvoltată la arborele acționării;

- ☞ a accepta că $\underline{q}_r(t) = \underline{K}_2 \cdot \underline{q}_m(t)$, este mult prea contestabil, datorită faptului că la transmisiile mecanice utilizate la roboți problemele legate de consumarea jocurilor, de frecare uscată, de frecare vâscoasă, de elasticități și inerții rămân prezente și introduc perturbații de mărimi diferite funcție de natura transmisiilor, dar și de gradul de precizie al realizării lor. Pentru a reduce pe cât posibil complexitatea modelului dinamic al robotului de cele mai multe ori se consideră transmisiile perfecte, astfel că doar o parte din influențele prezentate mai sus sunt luate în considerare;

- ☞ relația $\underline{q}(t) = \underline{K}_3 \cdot \underline{q}_r(t)$ nu poate fi aplicată ad literam deoarece sistemul mecanic de articulații al robotului prezintă momente de inerție masice variabile în timp, nu este rigid, axele de rotație ale elementelor care intră în componența unei cuple cinematice conducătoare nu sunt coliniare; în unele cazuri sistemul mecanic de articulații al robotului are în componență bucle mecanice (de exemplu mecanisme de tip paralelogram).

Dacă elementele sistemului mecanic al robotului sunt perfect rigide atunci printr-o analiză cinematică directă se poate determina situarea în timp a punctului caracteristic atașat efectorului final al robotului.

Principalele prezumții care vor fi acceptate pentru modelul fizic al robotului pot fi reduse la următoarele trei:

- ☞ elementele sistemului mecanic al robotului sunt modelate prin corpuri rigide, efectele elasticității fiind neglijate;

- ☞ cuplele cinematice conducătoare sunt perfecte din punct de vedere constructiv (fără jocuri, fără abateri de coaxialitate);

- ☞ lanțul cinematic al robotului nu este nici ramificat, nici închis;

- ☞ gradele de libertate ale robotului nu sunt cuplate cinematic.

3.2. Modelarea dispozitivului de ghidare al robotului

3.2.1. Formalisme

În cazul dispozitivului de ghidare al robotului modelarea dinamică constă în determinarea ecuațiilor dinamice corespunzătoare structurii aferente. În ceea ce urmează se vor prezenta două tehnici care permit scrierea ecuațiilor lor dinamice și anume formalismul Lagrange și respectiv formalismul Newton-Euler.

3.2.2. Formalismul Lagrange

Utilizarea formalismului Lagrange permite obținerea cu ușurință a ecuațiilor dinamice ale robotului, ecuații care pot fi utilizate fie pentru determinarea forțelor pe care trebuie să le dezvolte acționările robotului fie pentru verificarea acționărilor alese.

Deoarece fiecare element al lanțului cinematic al robotului nu dispune decât de un singur grad de libertate – translație sau rotație – caracterizat de coordonatele generalizate q_i , sistemul de ecuații Lagrange de speța a II-a va fi dat de:

$$\frac{d}{dt} \left(\frac{\partial E_c}{\partial \dot{q}_i} \right) - \frac{\partial E_c}{\partial q_i} = Q_i, \quad (3.1)$$

$$i = 1 \dots n.$$

unde:

- n reprezintă numărul gradelor de libertate al robotului;
- E_c reprezintă energia cinetică a robotului;
- $\dot{q}_i = \frac{dq_i}{dt}$ reprezintă viteza (liniară/unghiulară) a elementului „i”;
- q_i reprezintă coordonata generalizată a elementului „i”;
- Q_i reprezintă forța generalizată; și în general asupra robotului pot acționa următoarele tipuri de forțe și momente exterioare:
 - componenta $Q_{i_{EF}}$ din forța generalizată căreia îi corespunde torsorul τ_{EF} care acționează la efectorul final al robotului și care se transmite la cuplele cinematice conducătoare prin intermediul transpusei matricei Jacobiane a robotului;
 - componenta $Q_{i_{act}}$ din forța generalizată căreia îi corespunde torsorul activ τ_{act} din cuple, produs de motoarele de acționare;

- componenta Q_{i_g} aferentă forțelor gravitaționale ale elementelor robotului și ale elementelor acționării dispuse pe structura robotului;
- componenta $Q_{i_{ext}}$ aferentă forțelor și momentelor exterioare de orice altă natură care acționează în puncte oarecare ale robotului.

Deoarece forțele gravitaționale sunt forțe conservative, conform [Drag96] componenta din Q_i aferentă forțelor gravitaționale poate fi scrisă sub următoarea formă :

$$Q_{i_g} = -\frac{\partial E_p}{\partial q_i} \quad (3.2)$$

și cum funcția de forță gravitațională nu depinde în nici un caz de viteze se poate scrie relația 3.1 sub următoarea formă:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_{i_{EF}} + Q_{i_{act}} + Q_{i_{ext}} \quad (3.3)$$

$i = 1 \dots n.$

unde:

$$L = E_c - E_p \quad (3.4)$$

reprezintă funcția lui Lagrange și este energia reală a sistemului adică diferența dintre energia cinetică E_c și energia potențială E_p . Relația este adevărată dacă elementele lanțului cinematic al robotului sunt solide rigide, adică ele nu înmagazinează energie internă.

În cazul particular când asupra robotului nu acționează forțe tehnologice sau perturbatoare externe (cazul unui robot de manipulare) se face observația că cele două componente $Q_{i_{EF}}$ și $Q_{i_{ext}}$ se anulează.

Prin urmare, în vederea realizării efective a calculului, Lagrangianul poate fi scris sub următoarea formă:

$$\sum_{i=1}^n \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \tau_{act,i} \quad (3.5)$$

$i = 1 \dots n.$

unde:

- a) $L = E_c - E_p$ reprezintă funcția lui Lagrange;

în care:

- E_p reprezintă energia potențială a robotului și este dată de $E_p = \sum_1^n m_i \cdot g \cdot z_{S_i}^0$,

unde:

- g reprezintă accelerația gravitațională;
- m_i reprezintă masa corpului „i”;
- $z_{s_i}^0$ reprezintă cota centrului de masă a corpului „i” în sistemul de coordonate de referință;
- E_{C_i} reprezintă energia cinetică a corpului „i”, compusă din:
 - energia cinetică de translație notată $E_{C_i}^{(T)}$, care reprezintă rezultatul semiprodusului dintre masă și pătratul vitezei centrului de masă a elementului „i”, în raport cu un reper fix legat de baza robotului.

$$E_{C_i}^{(T)} = \frac{1}{2} m_i \left[\underline{V}_{s_i}^0 \right]^T \cdot \left[\underline{V}_{s_i}^0 \right] \quad (3.6)$$

unde:

- m_i reprezintă masa elementului „i”;
- $\underline{V}_{s_i}^0$ reprezintă viteza centrului de masă S_i față de elementul fix, exprimate în sistemul de referință „i”, utilizând formulele de recurență:

$$\underline{V}_{s_i}^0 = \underline{V}_{o_i}^0 + \underline{V}_{s_i}^{0i} + \underline{\omega}_i^0 \Lambda^i \underline{O_i S_i} \quad (3.7)$$

$$\underline{V}_{o_i}^0 = A_{i-1}^i \cdot \underline{V}_{o_i}^0 \quad (3.8)$$

- energia cinetică de rotație notată $E_{C_i}^{(R)}$ care reprezintă rezultatul semiprodusului dintre masa și pătratul vitezei unghiulare a elementului „i”, în raport cu un reper fix legat de baza robotului.

$$E_{C_i}^{(R)} = \frac{1}{2} \left[\underline{\omega}_i^0 \right]^T \cdot \underline{J}_i \cdot \left[\underline{\omega}_i^0 \right] \quad (3.9)$$

unde:

- \underline{J}_i reprezintă tensorul de inerție a elementului „i” față de axele principale de inerție;
- $\underline{\omega}_i^0$ reprezintă viteza unghiulară a elementului „i” față de elementul fix, exprimate în sistemul de referință „i”, utilizând formulele de recurență:

$$\underline{\omega}_i^0 = \underline{\omega}_i^{i-1} + A_{i-1}^{i-1} \left[\underline{\omega}_{i-1}^{i-2} + A_{i-2}^{i-1} \left[\underline{\omega}_{i-2}^{i-3} + \dots + A_2^3 \left[\underline{\omega}_2^1 + A_1^2 \underline{\omega}_1^0 \right] \right] \right] \quad (3.10)$$

În acest caz energia cinetică se poate scrie sub următoarea formă:

$$E_{C_i} = E_{C_i}^{(T)} + E_{C_i}^{(R)} = \frac{1}{2} m_i \left[\underline{V}_{s_i}^0 \right]^T \cdot \left[\underline{V}_{s_i}^0 \right] + \frac{1}{2} \left[\underline{\omega}_i^0 \right]^T \cdot \underline{J}_i \cdot \left[\underline{\omega}_i^0 \right] \quad (3.11)$$

Forțele generalizate Q , se determină pornind de la calculul lucrului mecanic virtual al unui sistem de forțe care acționează asupra unui robot cu „n” grade de libertate. Conform [Drag96] presupunând că asupra elementului „i” acționează k forțe \bar{F}_j și h momente \bar{M}_j , lucrul mecanic virtual pentru întregul robot cu „n” grade de libertate este:

$$\delta L = \sum_{i=1}^n \sum_{j=1}^k {}^0 \bar{F}_j \cdot \delta {}^i \bar{r}_j + \sum_{i=1}^n \sum_{j=1}^h {}^0 \bar{M}_j \cdot \delta \bar{\theta}_i \quad (3.12)$$

În acest caz relația (5.213) va deveni:

$$\sum_{j=1}^n \left(A_j \cdot \ddot{q}_j + B_j \cdot \dot{q}_j^2 + \sum_{k=j-1}^n C_{jk} \cdot \dot{q}_j \cdot \dot{q}_k \right) = Q_i \quad (3.13)$$

$i = 1 \dots n.$

sau sub formă matriceală:

$$A(\underline{q}) \cdot \ddot{\underline{q}} + B(\underline{q}) \cdot \dot{\underline{q}}^2 + C(\underline{q}) \cdot \dot{\underline{q}} \cdot \dot{\underline{q}} + \underline{Q}(\underline{q}) = \underline{\tau}_q \quad (3.14)$$

în care:

- $\ddot{\underline{q}} = \left(\ddot{q}_1 \quad \ddot{q}_2 \quad \dots \quad \ddot{q}_n \right)^T$ reprezintă vectorul accelerație;
- $\dot{\underline{q}}^2 = \left(\dot{q}_1^2 \quad \dot{q}_2^2 \quad \dots \quad \dot{q}_n^2 \right)^T$ reprezintă vectorul pătratelor vitezelor generalizate;
- $\underline{\dot{q}\dot{q}} = \left(\dot{q}_1 \dot{q}_2 \quad \dot{q}_1 \dot{q}_3 \quad \dots \quad \dot{q}_1 \dot{q}_n \quad \dot{q}_2 \dot{q}_3 \quad \dots \quad \dot{q}_2 \dot{q}_n \quad \dots \quad \dot{q}_{n-1} \dot{q}_n \right)^T$ reprezintă vectorul produs

cartezian al vitezelor având $n(n-1) \cdot 2$ componente;

- $A(\underline{q})$, $B(\underline{q})$, $C(\underline{q})$ sunt matricele formate cu coeficienții dinamici ai robotului și ele depind de configurația robotului la un moment dat.

Examinând sistemul (3.13) se observă că sistemul se referă la un sistem de ecuații de echilibru dinamic al forțelor. Deci:

- $A(\underline{q}) \cdot \ddot{\underline{q}}$ reprezintă vectorul forțelor de inerție, astfel $A(\underline{q})$ fiind matricea energiei cinetice simetrică, pozitiv definită;

- $B(\underline{q}) \cdot \dot{\underline{q}}^2$ reprezintă vectorul forțelor centrifuge;

- $C(\underline{q}) \cdot \dot{\underline{q}} \cdot \dot{\underline{q}}$ reprezintă vectorul forțelor de cuplaj dintre articulații, sau forțele Coriolis.

În [Tank00] se prezintă determinarea ecuațiilor dinamice utilizând formalismul Lagrange pentru un robot cu cinci grade de libertate de tip TRTRR.

3.2.3. Formalismul Newton-Euler

Utilizându-se formalismul recursiv Newton-Euler se determină torsorul reacțiunilor la nivelul unei cuple cinematice conducătoare pentru un set dat al coordonatelor generalizate, al vitezelor generalizate, respectiv al accelerațiilor generalizate de la nivelul fiecărei cuple cinematice conducătoare.

În cazul recursivității directe se realizează propagarea informațiilor cinematice (viteze, respectiv accelerații generalizate) de la sistemul de referință de bază legat de baza robotului spre efectorul final.

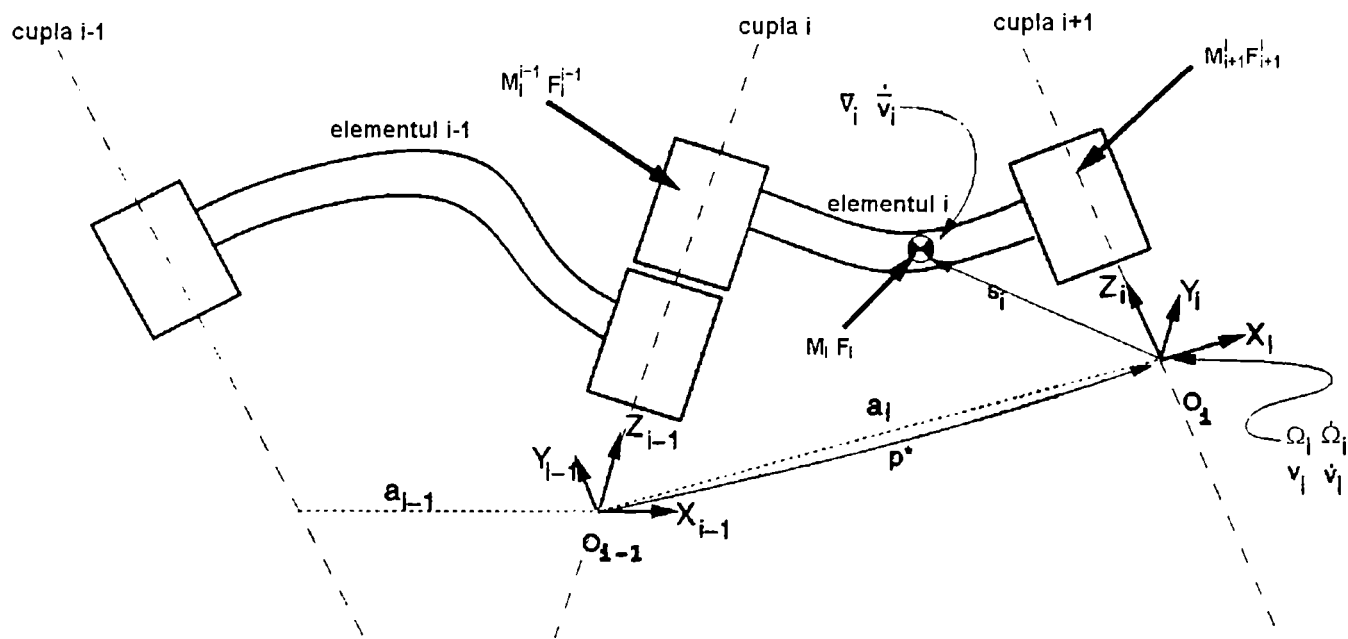


Fig.3.2. Notățiile utilizate pentru deducerea ecuațiilor dinamice, utilizând convenția Denavit-Hartenberg standard

Dacă se utilizează notațiile din figura 3.2 și convenția Denavit-Hartenberg standard se poate scrie:

$${}^{i+1}\underline{\Omega}_{i+1} = {}^{i+1}R_i \cdot ({}^i\underline{\Omega}_i + \xi_i \cdot z_0 \cdot \dot{q}_{i+1}) \quad (3.15)$$

$${}^{i+1}\dot{\underline{\Omega}}_{i+1} = {}^{i+1}R_i \cdot \left\{ {}^i\dot{\underline{\Omega}}_i + \xi_i \cdot [z_0 \cdot \ddot{q}_{i+1}] + {}^i\underline{\Omega}_i \times (z_0 \cdot \dot{q}_{i+1}) \right\} \quad (3.16)$$

$${}^{i+1}\underline{v}_{i+1} = {}^{i+1}\underline{\Omega}_{i+1} \times {}^{i+1}p_{i+1}^* + {}^{i+1}R_i \cdot ({}^i\underline{v}_i + \xi_i \cdot z_0 \cdot \dot{q}_{i+1}) \quad (3.17)$$

$${}^{i+1}\dot{\underline{v}}_{i+1} = {}^{i+1}\dot{\underline{\Omega}}_{i+1} \times {}^{i+1}p_{i+1}^* + {}^{i+1}R_i \cdot ({}^i\dot{\underline{v}}_i + \xi_i \cdot z_0 \cdot \ddot{q}_{i+1}) + 2 \cdot \xi_i \cdot {}^{i+1}\underline{\Omega}_{i+1} \cdot ({}^{i+1}R_i \cdot z_0 \cdot \dot{q}_{i+1}) + {}^{i+1}\underline{\Omega}_{i+1} \times ({}^{i+1}\underline{\Omega}_{i+1} \times {}^{i+1}p_{i+1}^*) \quad (3.18)$$

Prin urmare:

$${}^i \underline{\dot{v}}_{S_i} = {}^i \underline{\Omega}_i \times \underline{s}_i + {}^i \underline{\Omega}_i \times ({}^i \underline{\Omega}_i \times \underline{s}_i) + {}^i \underline{\dot{v}}_i \quad (3.19)$$

$${}^i \underline{F}_{S_i} = m_i \cdot {}^i \underline{\dot{v}}_{S_i} \quad (3.20)$$

$${}^i \underline{M}_{S_i} = J_i \cdot {}^i \underline{\dot{\Omega}}_i + {}^i \underline{\Omega}_i \times (J_i \cdot {}^i \underline{\Omega}_i) \quad (3.21)$$

În cazul propagării inverse a recursivității se determină torsorul care acționează asupra fiecărui element al dispozitivului de ghidare al robotului, pornindu-se de la efectorul final al robotului spre sistemul de referință legat de baza robotului. Atunci pentru $n \geq i \geq 1$, rezultă:

$${}^i \underline{F}_i = {}^i R_{i-1} \cdot {}^{i-1} \underline{F}_{i-1} + {}^i \underline{F}_{S_i} \quad (3.22)$$

$${}^i \underline{M}_i = {}^i R_{i-1} \cdot [{}^{i-1} \underline{M}_{i-1} + ({}^{i-1} R_i \cdot {}^i \underline{p}_i^*) \times {}^{i-1} \underline{F}_{i-1}] + ({}^i \underline{p}_i^* + \underline{s}_i) \times {}^i \underline{F}_{S_i} + {}^i \underline{M}_{S_i} \quad (3.23)$$

Prin urmare, torsorul redus la arborele cuplei cinematice conducătoare i va fi dat de:

$$\underline{Q}_i = \begin{cases} ({}^i \underline{M}_i)^T \cdot ({}^i R_{i-1} \cdot \underline{z}_0) & \text{pentru o cupla de rotație} \\ ({}^i \underline{F}_i)^T \cdot ({}^i R_{i-1} \cdot \underline{z}_0) & \text{pentru o cupla de translație} \end{cases} \quad (3.24)$$

unde:

- J_i reprezintă momentul de inerție al elementului i în raport cu centrul său de masă S_i ;
- \underline{s}_i reprezintă vectorul de poziție al centrului de masă S_i al elementului i în raport cu sistemul de coordonate O_i ;
- $\underline{\Omega}_i$ reprezintă viteza unghiulară aferentă elementului i ;
- $\underline{\dot{\Omega}}_i$ reprezintă accelerația unghiulară aferentă elementului i ;
- \underline{v}_i reprezintă viteza liniară aferentă elementului i ;
- $\underline{\dot{v}}_i$ reprezintă accelerația liniară aferentă elementului i ;
- ${}^i \underline{M}_{i-1}$ reprezintă momentul exercitat de elementul $i-1$ asupra elementului i ;
- ${}^i \underline{F}_{i-1}$ reprezintă forța exercitată de elementul $i-1$ asupra elementului i ;
- ${}^i \underline{M}_{S_i}$ reprezintă momentul total redus la centrul de masă a elementului i ;
- ${}^i \underline{F}_{S_i}$ reprezintă forța totală redusă la centrul de masă a elementului i ;

- ${}^{i-1}R_i$ matricea de orientare care definește orientarea sistemului O_i în raport cu sistemul de coordonate O_{i-1} :

$${}^{i-1}R_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \cdot \sin \theta_i & \sin \alpha_i \cdot \sin \theta_i \\ \sin \theta_i & \cos \alpha_i \cdot \cos \theta_i & -\sin \alpha_i \cdot \cos \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{bmatrix} \quad (3.25)$$

$${}^i R_{i-1} = ({}^{i-1}R_i)^{-1} = ({}^{i-1}R_i)^T \quad (3.26)$$

- ${}^i \underline{p}_i^*$ reprezintă vectorul de poziție a sistemului O_i în raport cu sistemul de coordonate O_{i-1} :

$${}^i \underline{p}_i^* = \begin{bmatrix} a_i \\ d_i \cdot \sin \alpha_i \\ d_i \cdot \cos \alpha_i \end{bmatrix} \quad (3.27)$$

- \underline{z}_0 este versorul unitar după direcția Z și este $\underline{z}_0 = [0 \ 0 \ 1]^T$

Algoritmul recursiv Newton-Euler (RNE) este deosebit de eficient în reducerea timpului de calcul, după cum se observă și din tabelul 3.2.

Tabelul 3.2. Comparație între diferiți algoritmi în funcție de numărul de operații efectuate

Formalism	Multiplicări	Adunări	Pentru M=6	
			Multip.	Adunări
Lagrange [Holl80]	$32 \frac{1}{2} n^4 + 86 \frac{5}{12} n^3 +$ $171 \frac{1}{4} n^2 + 53 \frac{1}{3} n - 128$	$25n^4 + 66 \frac{1}{3} n^3 + 129 \frac{1}{4} n^2$ $+ 42 \frac{1}{3} n - 96$	66.271	51.548
RNE [Holl80]	$150n - 48$	$131n - 48$	852	738
RNE simplificat [Murr84]			224	174

Pentru cazul concret al unui robot după obținerea ecuațiilor dinamice forma generală poate fi simplificată, înlăturându-se adunările, respectiv multiplicările cu zero, cu 1 sau -1 . Termenii de zero și unu sunt datorati fie termenilor trigonometrici din matricea de orientare (multe elemente ale dispozitivului de ghidare fac între ele unghiuri de 0° sau 90°), fie parametrilor cinematici de valoare zero.

În tabelul 3.2. preluat din [Murr84] se prezintă numărul de multiplicări și adunări care sunt necesare pentru un formalism RNE simplificat aplicat la un robot tip Puma cu șase grade de mobilitate. Din tabel se observă o reducere de aproximativ patru ori a

numărului de operații necesare a se realiza în cazul aplicării formalismului RNE simplificat față de formalismul RNE obișnuit și de 295 de ori mai puține decât la utilizarea formalismului Lagrange.

În [Tarc96c] se prezintă deducerea ecuațiilor dinamice utilizând formalismul RNE pentru un robot cu patru grade de libertate de tip RRRR.

3.3. Modelarea sistemelor de acționare

3.3.1. Introducere

Funcția sistemului de acționare este aceea de a pune în mișcare relativă la momentul oportun elementele cuplelor cinematice conducătoare ale sistemului mecanic [Kova99b].

Sistemul de acționare se compune din două subsisteme: motorul de acționare și transmisia mecanică.

În general, efectele dinamice datorate electronicii de putere, motoarelor de acționare și transmisiilor mecanice sunt mai puțin tratate în literatura de specialitate, deși efectele generate de cele trei componente ale robotului nu sunt de o mai mică importanță ca și efectele structurii mecanice a robotului.

3.3.2. Transmisii mecanice. Generalități

Motoarele sistemului de acționare pot acționa direct sau indirect cuplele cinematice conducătoare ale robotului.

Acționarea directă – direct drive – constă în faptul că elementul de ieșire al motorului acționează în mod direct elementele cuplei cinematice conducătoare.

Acționarea indirectă necesită ca între motor și elementele cuplei cinematice conducătoare să se interpună transmisii mecanice.

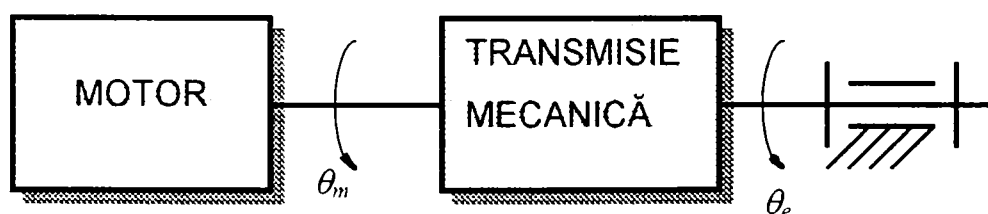


Fig. 3.3. Acționare indirectă

Transmisiile mecanice asigură adaptarea caracteristicii mecanice a mașinii motoare la caracteristica mecanică rezistentă, la nivelul cuplei cinematice conducătoare. Acest rol funcțional se realizează prin transfer de energie și printr-o transformare cinematică.

Transformarea cinematică are în vedere două aspecte și anume:

☞ o transformare cantitativă, adică o schimbare a parametrilor mișcării (viteze, accelerații generalizate);

☞ o transformare calitativă, adică o schimbare a naturii mișcării (transformarea rotației în translație sau invers).

O acționare indirectă se reprezintă schematic în figura 3.3.

Transmisiile mecanice pot fi cu elemente dințate (reductoare), la a căror utilizare se va avea în vedere utilizarea unor dispozitive de scoatere a jocului, cu șurub – piuliță, cu lanț – apare efectul poligonal, prin curele, prin fricțiune, cu mecanisme cu bare, cu cabluri pretensionate, etc.

Indiferent de transmisia mecanică utilizată, esențial este ca raportul de transmitere μ a mișcării de la mașina motoare la cupla cinematică conductoare să fie riguros constant.

Pentru cuplele de rotație, mărimile măsurate la nivelul elementului din cadrul sistemului mecanic al robotului (la ieșirea din transmisia mecanică) sunt legate de mărimile din amonte de transmisia mecanică după relațiile prezentate în tabelul 3.3.

Tabelul 3.3. Relațiile de dependență a mărimilor din aval de mărimile din amonte de transmisia mecanică funcție de raportul de transmitere μ

Mărime	Relația de dependență
Momentul de inerție masic	$J_e = \mu^2 \cdot J_m$
Coeficientul de frecare vâscoasă	$B_e = \mu^2 \cdot B_m$
Coeficientul de frecare uscată	$\tau_{ce} = \mu \cdot \tau_{cm}$
Momentul redus la arborele transmisiei M_e (motorului M_m)	$M_e = \mu \cdot M_m$
Viteza unghiulară	$\omega_e = \frac{\omega_m}{\mu}$
Accelerația unghiulară	$\varepsilon_e = \frac{\varepsilon_m}{\mu^2}$

Motoarele sistemului de acționare pot fi dispuse periferic – amplasare distribuită – sau pot fi dispuse umeral.

Amplasarea distribuită, periferică prezintă avantajul simplificării transmisiei mecanice, dar prezintă dezavantajul încărcării extremităților de ghidare cu forțe masice suplimentare.

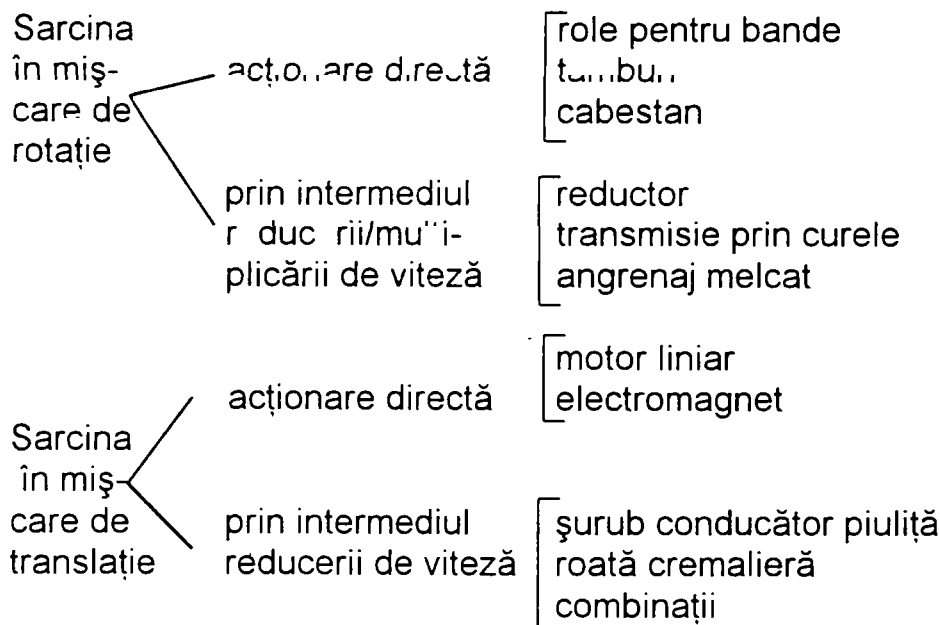
Amplasarea umerală realizează dispunerea motoarelor de acționare în elementul fix al dispozitivului de ghidare. Această variantă prezintă avantajul descărcării dispozitivului de ghidare de forțe masice, dar are dezavantajul unei transmisii mecanice de o construcție complicată.

3.3.3. Modelarea și alegerea transmisiei mecanice

Alegerea tipului de transmisie mecanică se face după următoarele criterii:

- I. specificul obiectului de situat;
- II. corelarea caracteristicilor mecanice statice cuplu (putere) – viteză ale servomotorului cu cele ale sarcinii;
- III. regim dinamic optim;
- IV. corectarea factorului de proporționalitate pentru asigurarea stabilității servomecanismului cu sistem automat.

După primul criteriu tipul transmisiei este determinat de caracterul particular al sarcinii și de tipul mișcării elementului de acționare. O clasificare se poate face astfel:



Ecuțiile de mișcare raportate la ieșirea elementului de acționare sunt:

$$m \frac{d^2x}{dt^2} + D \frac{dx}{dt} + F_r = F_e \quad (3.28)$$

pentru mișcarea de translație, iar în cazul mișcării de rotație

$$J \frac{d^2\theta}{dt^2} + B \frac{d\theta}{dt} + M_r = M_e \quad (3.29)$$

Coeficienții în aceste ecuații sunt:

m, J – masa, respectiv momentul de inerție echivalent, Kg sau respectiv $\text{Kg} \cdot \text{m}^2$;

D, B – coeficienții de frecare vâscoasă, în $\text{N} \cdot \text{s}/\text{m}$, respectiv $\text{N} \cdot \text{m} \cdot \text{s}/\text{rad}$;

F_r, M_r – forța, respectiv cuplul rezistent static, în N, respectiv $\text{N} \cdot \text{m}$;

F_e, M_e – reprezintă forța, cuplul electromagnetic;

x, θ sunt deplasarea liniară, respectiv unghiulară.

Ecuațiile (3.28) și (3.29) reprezintă legătura între elementul de execuție (în dreapta) și ansamblul mecanic transmisie – sarcină (în stânga).

Considerând forța, respectiv cuplul rezistent de tip reactiv, de forma

$$M_r = K\theta$$

$$F_r = K'x \quad (3.30)$$

Funcțiile de transfer ale elementului mecanic rezultă

$$\frac{X(s)}{M_e(s)} = \frac{1}{ms^2 + Ds + K'}$$

și respectiv (3.31)

$$\frac{\theta(s)}{M_e(s)} = \frac{1}{Js^2 + B\theta + K}$$

Se observă că partea mecanică constituie în general un element de ordinul doi cu funcția de transfer

$$G(s) = \frac{h}{a^2 + 2\xi\omega_0 s + \omega_0^2} \quad (3.32)$$

în care:

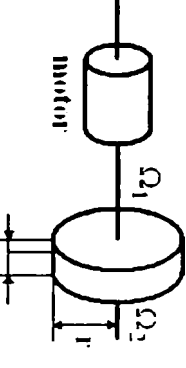
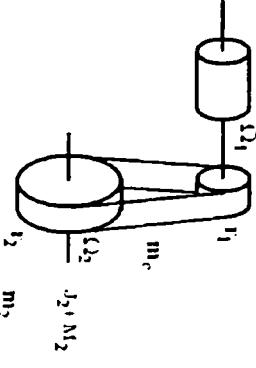
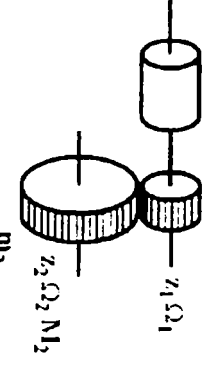
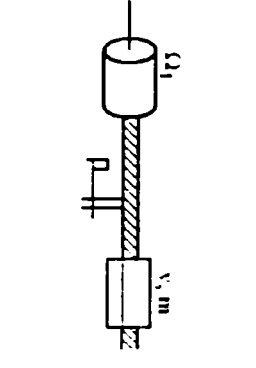
$$h = \frac{1}{m} \text{ sau } \frac{1}{J} - \text{constantă};$$

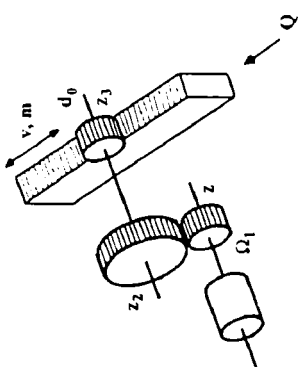
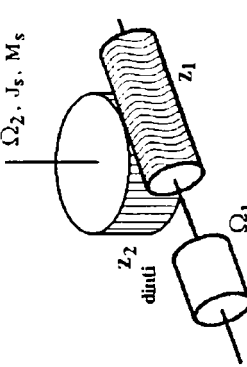
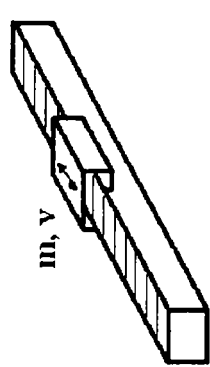
$$\omega_0 = \sqrt{\frac{m}{K'}} \text{ sau } \sqrt{\frac{J}{K}} - \text{pulsăția proprie};$$

$$\xi = \frac{D}{2\sqrt{mK'}} \text{ sau } \frac{B}{2\sqrt{JK}} - \text{factorul de amortizare};$$

Relațiile (3.31) sau (3.32) servesc la identificarea părții mecanice.

În tabelul 3.4 se prezintă cele mai răspândite sisteme de transmisie și unele relații de calcul [Popo92].

Tip transmisie 1	Simbol și date 2	Caracter sarcină 3	Viteza generalizată 4	Inerție generalizată 5	Frezare vâscasă 6	Sarcină statică 7	Observații și precizări 8
Directă, rotație		inerțială	$\Omega_1 = \Omega_2$	$J = J_1 + J_2$ $J_2 = m_2 \cdot \frac{r^2}{2}$	$B = B_1 + B_2 \approx 0$	$M_r = 0$	Valabil și la cilindru
Transmisie prin curea elastică		inerțială, frecări	$\Omega_1 = \frac{r_2}{r_1} \Omega_2$	$J = J_1 + J_2 + J_c$ $J_2 = m_2 \cdot \frac{r_1^2}{2}$ $J_c = m_c r_1^2$	$B \approx B_2 \cdot \frac{r_1^2}{r_2^2}$	$M_r = M_2 \cdot \frac{r_1}{r_2} \cdot \frac{1}{\eta}$	η - randamentul transmisiei
Reductor, rigidă		inerțială, frecări	$\Omega_1 = \frac{z_2}{z_1} \Omega_2$	$J = J_1 + J_2$ $J_2 = m_2 \cdot \left(\frac{\Omega_2}{\Omega_1} \right)^2$	$B \approx B_2 \cdot \frac{z_1^2}{z_2^2}$	$M_r = M_2 \cdot \frac{z_1}{z_2} \cdot \frac{1}{\eta}$	
Șurub conducător - piuliță		inerțială, frecări, statică	$\Omega_1 = \frac{2\pi}{p} v$	$J = J_1 + J_s + J_2$ $J_s = m_s \cdot \frac{d_s^2}{8}$ $J_2 = m_2 \cdot \frac{v^2}{\Omega_2^2}$	$B = B_1 + B_2$	$M_r = Q \mu \cdot \frac{d_s}{2}$ $Q = Q_{long} + Q_{norm}$	date șurub: m_s - masa; d_s - diam. mediu; p - pasul; μ - coef. frecare; Q - forța rezistentă

Tip transmisie	1	2	3	4	5	6	7	8
Tip transmisie	1	2	3	4	5	6	7	8
Pinion cremalieră și reductor		inerțială, frecări, statică	$\Omega_1 = \frac{2v}{d_0} \cdot \frac{z_2}{z_1}$	$J = J_1 + J_2 + J_c$ $J_1 = J_m + J_{z_1}$ $J_2 = J_{z_2} \cdot \frac{z_1^2}{z_2^2}$ $J_c = \frac{md_0^2}{4} \cdot \frac{z_1^2}{z_2^2}$	$B = B_1 + B_c \cdot \frac{z_1^2}{z_2^2}$	$M_r = \frac{d_0}{2\eta} \cdot \frac{z_1}{z_2}$	d_0 - diam. mediu al roții de pe cremaliera; Q - forța rezistentă longitudinală; η - randament reductor.	
Angrenaj melcat		inerțială, frecări, statică	$\Omega_1 = \Omega_2 \cdot \frac{z_2}{z_1}$	$J = J_1 + J_{z_1} + J_2$ $J_2 = J_s \cdot \frac{z_1^2}{z_2^2}$	$B \approx B_1$	$M_r = \frac{M_s}{\eta} \cdot \frac{z_1}{z_2}$	η - randamentul angrenajului melcat	
Directă, translație (motor liniar)		inerțială, statică	$v_1 = v$	$m_1 = m$	$D = D_1$	$F_r = Q_{long} + \mu Q_{norm}$	μ - coef. frecare; Q - forța rezistentă	

Criteriul al II-lea de alegere a sistemului de transmisie are la bază cerința puterii optime transmise. Pentru servomotoarele de curent continuu cu excitație cu magneți permanenți, caracteristică mecanică motoare este liniară ca în figură 3.4.a, iar puterea utilă maximă se obține la coordonatele punctului P de la $\omega_1 = \omega_2$ și aceasta este $P_{max} = M_1 \omega_1$. Dacă sarcina solicită puterea maximă la viteza ω_2 , este necesară o transmisie cu raportul de transmitere $\mu = \omega_1 / \omega_2$.

Metoda de alegere a raportului este indicată de fig. 3.4.b. Alura caracteristicii mecanică rezistentă este o hiperbolă, care suprapusă peste caracteristica mecanică motoare a servomotorului determină suprafața hașurată. În această zonă viteza se poate lua la o valoare convenabilă, cu care se determină raportul de transmitere. În afara zonei hașurate rezultă un raport de reducere neeconomic sub aspectul puterii optime. Dacă cele două caracteristici nu se intersectează, atunci este necesar un servomotor de putere mai mare, iar dacă suprafața hașurată este prea mare mașina motoare nu este utilizată economic din punct de vedere al puterii. Această ultimă variantă însă se utilizează atunci când se dorește existența unei rezerve de putere.

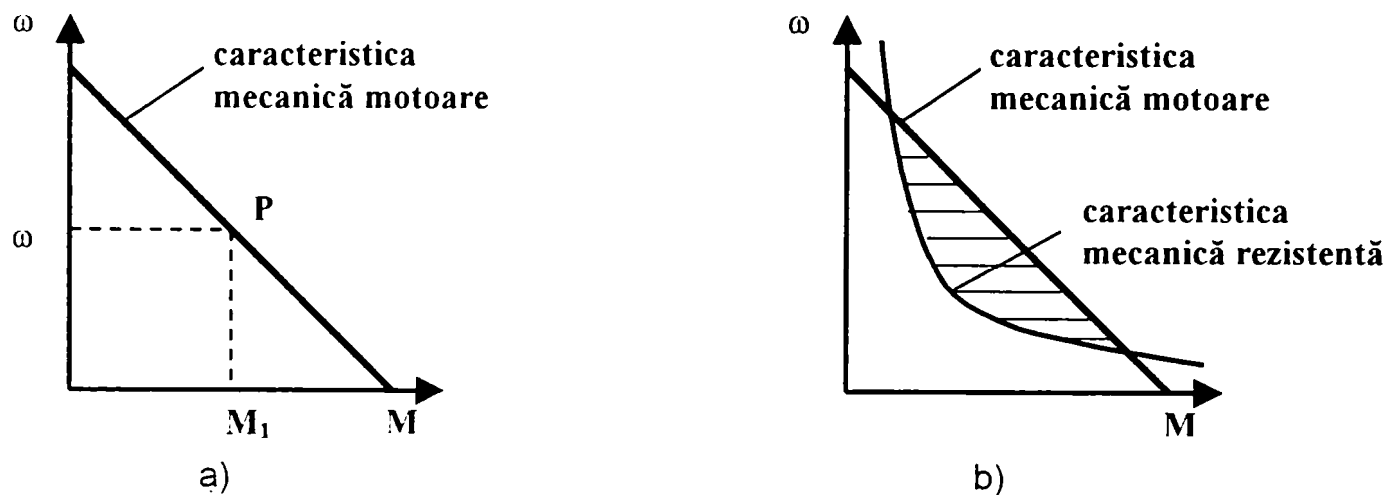


Fig. 3.4. Alegerea raportului de transmisie:
a) puterea utilă a servomotorului; b) metoda grafică.

Cel de-al III-lea criteriu de alegere al transmisiei se referă la utilizarea mașini motoare la un regim dinamic optim. Se ia în considerare influența raportului de transmitere asupra căldurii dezvoltate de motor și evident se dorește minimizarea pierderilor de căldură în timpul pornirii.

Pentru cele ce urmează se consideră sistemul de acționare din figura 3.5 pentru care există următoarele relații:

- coordonata generalizată a elementului în mișcare este dată de:

$$\theta_e = \mu \cdot \theta_m \quad (3.33)$$

- momentul masic de inerție redus la arborele motorului este:

$$J_{ef} = J_m + \mu \cdot J_e \quad (3.34)$$

- momentul redus la arborele motorului este:

$$M_{ef} = J_{ef} \cdot \frac{d\omega_m}{dt} + \mu \cdot M_e \quad (3.35)$$

Considerând servomotorul de curent continuu având cuplul electromagnetic proporțional cu curentul în indus $M_m = k \cdot i(t)$ și variația vitezei liniare în intervalele de timp, se calculează energia pierdută în rotor la pornire.

$$W = \int_0^t Ri^2 dt \quad (3.36)$$

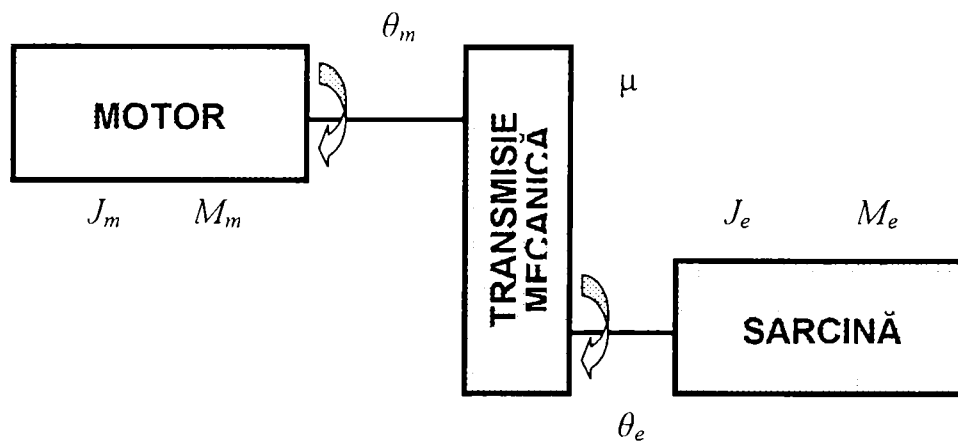


Fig. 3.5. Explicativă pentru deducerea raportului de transmisie dinamic.

Ținând cont de relația (3.33) și de faptul că $d\theta_e \approx 0$ se ajunge la expresia:

$$\begin{aligned} W &= \frac{R}{k^2} \left[\mu \cdot M_e^2 \cdot t + \frac{1}{\mu^2} \cdot (J_m + \mu^2 \cdot J_e)^2 \frac{\theta_e^2}{t^3} \right] = \\ &= \frac{\theta_e^2 \cdot J_e \cdot R}{k^2 \cdot t^3} \left[\left(\frac{J_m}{J_e} + \mu^2 \right)^2 \cdot \frac{1}{\mu^2} + \mu^2 \cdot p \right] \end{aligned} \quad (3.37)$$

în care s-a făcut notația

$$p = \frac{t^4 M_e^2}{\theta_e^2 \cdot J_e^2}$$

Expresia (3.37) se minimizează în raport cu variabila μ^2 și se obține un minim pentru

$$\mu^2 = \frac{J_m}{J_e} \cdot \frac{1}{\sqrt{1+p}} \quad (3.38)$$

Pentru cazul servomecanismelor, la care $M_e \approx 0$ rezultă $p=0$ și se găsește expresia raportului de transmisie dinamic:

$$\mu_{din} = \sqrt{\frac{J_m}{J_e}} \quad (3.39)$$

Raportând expresia energiei (3.35) se poate scrie

$$\frac{W(\mu)}{W(\mu_{din})} = \frac{\frac{1}{\mu^2} \cdot \left(\frac{J_m}{J_e} + \mu^2 \right)^2}{\frac{1}{\mu_{din}^2} \cdot \left(\frac{J_m}{J_e} + \mu_{din}^2 \right)^2} = \frac{\mu_{din}^2}{\mu^2} \cdot \left(\frac{J_m + \mu^2 \cdot J_e}{J_m + \mu_{din}^2 \cdot J_e} \right)^2 \quad (3.40)$$

Concluzia este că trebuie îndeplinită condiția:

$$\mu = \mu_{din} \quad (3.41)$$

pentru ca puterea disipată să fie minimă.

În practică cele două rapoarte sunt de obicei diferite și se încearcă apropierea lor modificând fie J_e , fie ω_l . Dacă diferența între cele două nu este prea mare se poate lua y rezultat din criteriile anterioare întrucât pierderile de putere nu cresc mult.

Al patrulea criteriu permite ajustarea lui μ pentru a putea face sistemul stabil. Se știe că stabilitatea unui sistem automat se poate influența prin ajustarea factorului de amplificare. Cum în acesta intră și raportul de transmitere, rezultă că este posibilă o alegere convenabilă a acestuia.

3.3.4. Modelarea acționării electrice

3.3.4.1. Modelarea acționării electrice cu motor de curent continuu

În figura 3.6. se prezintă cupla cinematică conducătoare "i" a unui robot industrial acționată de un motor de curent continuu.

Întrucât în ceea ce urmează toate mărimile sunt aferente cuplei "i" pentru ușurință acest indice se ignoră.

Notățiile sunt următoarele:

- θ_m, ω_m reprezintă poziția unghiulară și respectiv viteza unghiulară a motorului din cupla "i" [rad], [rad/s];

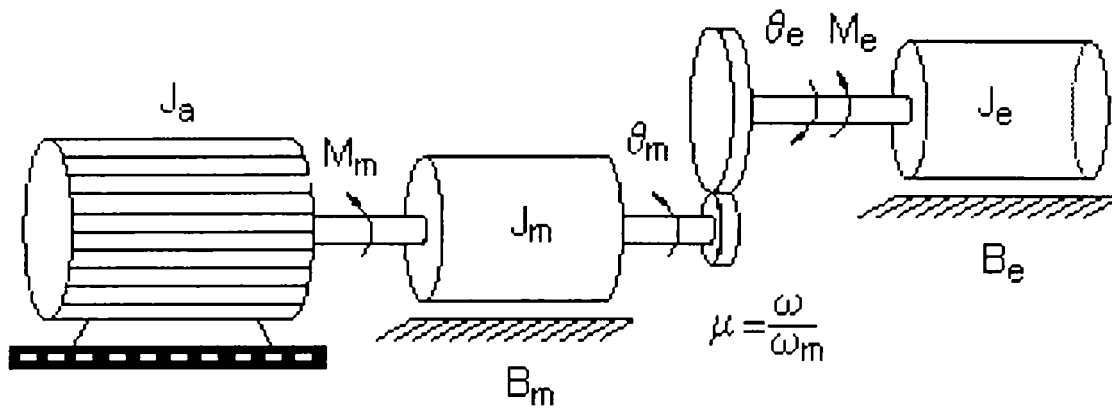


Fig.3.6. Cupla cinematică conducătoare "i" a unui robot industrial acționată de un motor de curent continuu.

- θ , ω reprezintă poziția unghiulară și respectiv viteza unghiulară a cuplei "i" [rad], [rad/s];
- J_a reprezintă momentul de inerție masic al rotorului motorului de curent continuu [kg m^2];
- J_m reprezintă momentul de inerție masic al cuplajului și a părții din reductor cuplată de rotorul motorului de curent continuu [kg m^2];
- J_e reprezintă momentul de inerție masic al elementului dispozitivului de ghidare redus la arborele motorului, care este acționat de cupla respectivă [kg m^2];
- μ reprezintă raportul de transmitere ω/ω_m ;

Pentru un servomotor de curent continuu comandat prin tensiune, modelul simplificat se prezintă în fig.3.7.

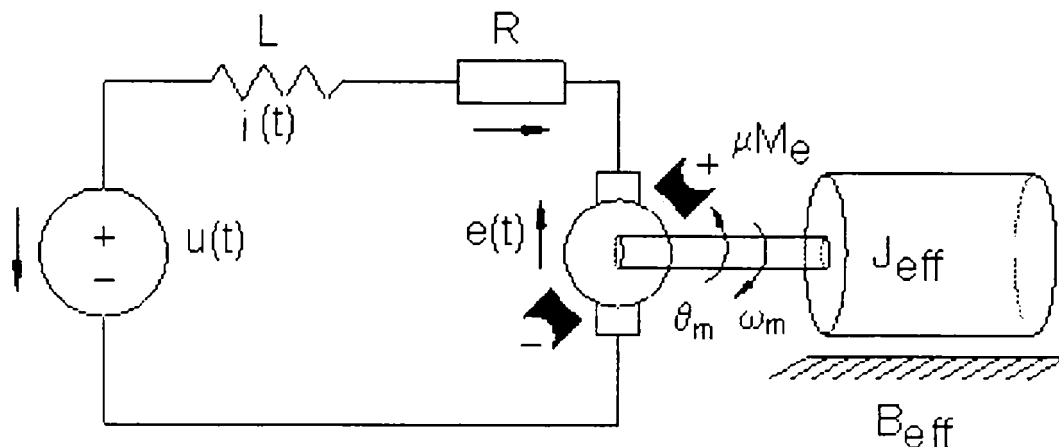


Fig.3.7. Modelul simplificat pentru un servomotor de curent continuu comandat prin tensiune

În figură J_{ef} reprezintă momentul de inerție masic efectiv redus la arborele motorului și este dat de relația:

$$J_{ef} = J_a + J_m + \mu^2 \cdot J_e \quad (3.42)$$

iar:

- $u(t)$ tensiunea de intrare (V);
- $\mu \cdot M_e$ reprezintă momentul redus la nivelul motorului de acționare [Nm];
- R rezistența circuitului rotoric [Ω];
- L inductanța circuitului rotoric [H];
- $e(t)$ pierderea de tensiune datorată tensiunii electromotoare [V], descrisă de

relația:

$$e = K_i \cdot \omega_m \quad (3.43)$$

unde $K_i = K_e \cdot \Phi_e$, iar K_e constanta motorului este dată de $K_e = \frac{p \cdot N}{2 \cdot \pi \cdot a}$ unde:

p – numărul perechilor de poli;

a – numărul perechilor de căi de curent în paralel între bornele motorului;

N – numărul total de conductoare în secțiunea rotorului.

Din figura 3.8. se poate scrie – în concordanță cu legile lui Kirchoff următoarele:

$$L \cdot \frac{d}{dt} i(t) + R \cdot i(t) + K_i \cdot \omega_m = u(t) \quad (3.44)$$

Momentul generat de motor este dat de:

$$M_m = K_i \cdot i(t) \quad (3.45)$$

Momentul generat de motor trebuie să învingă momentul redus inerțial $J_{ef} \cdot \frac{d\omega_m}{dt}$,

să compenseze cuplul rezistent de frecări vâscoase, precum și momentul de încărcare.

Din aceste considerente rezultă:

$$K_i \cdot i(t) = J_{ef} \cdot \frac{d\omega_m}{dt} + B_{ef} \cdot \omega_m + \mu \cdot M_e \quad (3.46)$$

Momentul de încărcare M_e este determinat de efectul Coriolis și cel centrifugal reduse la nivelul cuplei "i", de sarcina apărută în procesul de lucru, de efectul gravitațional și de cel de frecare.

Coeficientul global de frecare vâscoasă B_{ef} [Nm/rad/s] este dat de relația:

$$B_{ef} = B_m + \mu^2 \cdot B_e \quad (3.47)$$

unde: - B_m reprezintă coeficientul global de frecare vâscoasă aferent motorului, iar B_e reprezintă coeficientul global de frecare vâscoasă aferent părții de dincolo de reductor.

Se face observația că $u(t)$ se prezintă sub următoarea formă:

$$N(u) = \begin{cases} -u_{max} & \text{pentru } u \leq -u_{max} \\ u & \text{pentru } -u_{max} \leq u \leq u_{max} \\ +u_{max} & \text{pentru } u \geq u_{max} \end{cases} \quad (3.48)$$

Ecuțiile (3.44) respectiv (3.46) reprezintă modelul matematic al dinamicii cuplei "i" acționate de un servomotor de curent continuu cu magneți permanenți.

Dacă se alege vectorul de stare x dat de:

$$x_i = [\theta_m \quad \omega_m \quad i(t)]^T \quad (3.49)$$

atunci se poate scrie modelul matematic al acționării din cupla "i" sub următoarea formă matricială:

$$\dot{x} = A \cdot x + b \cdot N(u) + c \cdot M_e \quad (3.50)$$

unde matricele și vectorii coloană din ecuația de mai sus sunt date de:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{B_{ef}}{J_{ef}} & -\frac{K_1}{J_{ef}} \\ 0 & -\frac{K_1}{L} & -\frac{R}{L} \end{bmatrix}; \quad b = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix}; \quad c = \begin{bmatrix} 0 \\ -\frac{\mu}{J_{ef}} \\ 0 \end{bmatrix} \quad (3.51)$$

Aplicându-se transformata Laplace ecuațiilor (3.44), respectiv (3.46) acestea vor fi descrise de următoarele relații:

$$u(s) = (sL + R) \cdot i(s) + K_1 \cdot \omega_m(s) \quad (3.52)$$

$$K_1 \cdot i(s) = (J_{ef} \cdot s + B_{ef}) \cdot \omega_m(s) + \mu \cdot M_e(s) \quad (3.53)$$

Pornind de la aceste ecuații, rezultă schema bloc a unei acționări cu motor de curent continuu poate fi construită așa cum se prezintă în figura 3.8.

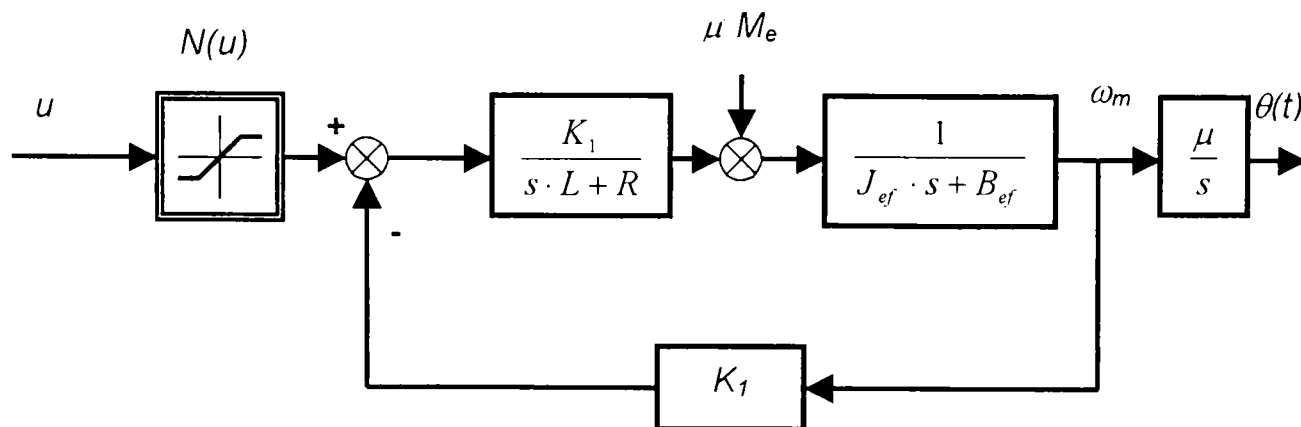


Fig.3.8. Schema bloc a unei acționări de curent continuu

Din ecuațiile (3.52), respectiv (3.53) rezultă:

$$\left[s^2 L J_{ef} + (L B_{ef} + R J_{ef}) s + (K_e^2 + R B_{ef}) \right] \cdot \omega_m(s) = K_i N(u) - \mu(sL + R) \cdot M_e(s) \quad (3.54)$$

sau poate fi pusă sub forma:

$$\left[s^2 + \left(\frac{1}{T_{emg}} + \frac{1}{T_{Bef}} \right) \cdot s + \frac{1}{T_{emg} \cdot T_{emc}} + \frac{1}{T_{emg} \cdot T_{Bef}} \right] \cdot \omega_m(s) = \frac{K_i N(u)}{L J_{ef}} - \frac{\mu(sL + R)}{L J_{ef}} \cdot M_e(s) \quad (3.55)$$

unde:

- $T_{emg} = \frac{L}{R}$ este constanta de timp electromagnetică;
- $T_{emc} = \frac{R \cdot J_{ef}}{K_e^2 \cdot \Phi_e^2}$ este constanta de timp electromecanică;
- $T_{Bef} = \frac{J_{ef}}{B_{ef}}$ este constanta de timp a forțelor de frecare vâscoasă.

Din relația de mai sus se poate deduce modelul matematic pentru acționarea cuplei

“i” funcție de poziția unghiulară θ a cuplei, știind că $\theta(s) = \frac{\mu}{s} \cdot \omega_m(s)$, acesta prezentându-

se:

$$s \cdot \left[s^2 + \left(\frac{1}{T_{emg}} + \frac{1}{T_{Bef}} \right) \cdot s + \frac{1}{T_{emg} \cdot T_{emc}} + \frac{1}{T_{emg} \cdot T_{Bef}} \right] \cdot \theta(s) = \frac{\mu \cdot K_i \cdot N(u)}{L J_{ef}} - \frac{\mu^2 (sL + R)}{L J_{ef}} \cdot M_e(s) \quad (3.56)$$

3.3.4.2. Modelarea acționării electrice cu motoare pas cu pas

3.3.4.2.1. Ecuații de bază

Schema de principiu a unui motor pas cu pas cu două faze se prezintă în figura 3.9.

Rotorul este un magnet permanent având o pereche de poli. Când înfășurările unei faze sunt alimentate se generează pe partea statorului un dipol magnetic. Dacă de exemplu faza 2 este activată, înfășurarea 3 produce un pol “S” iar înfășurarea 4 produce un pol “N”. Fig.2 arată că rotorul stă în poziție fixă orizontală doar dacă faza 2 este activă.

Numărul de pași pe o rotație completă este dat de:

$$S = 2 \cdot n \cdot m \quad (3.57)$$

unde:

n - numărul de perechi de poli ai rotorului;

m - numărul fazelor statorului.

Unghiul de pas este dat de:

$$\Delta\phi = \frac{360^\circ}{S} \quad (3.58)$$

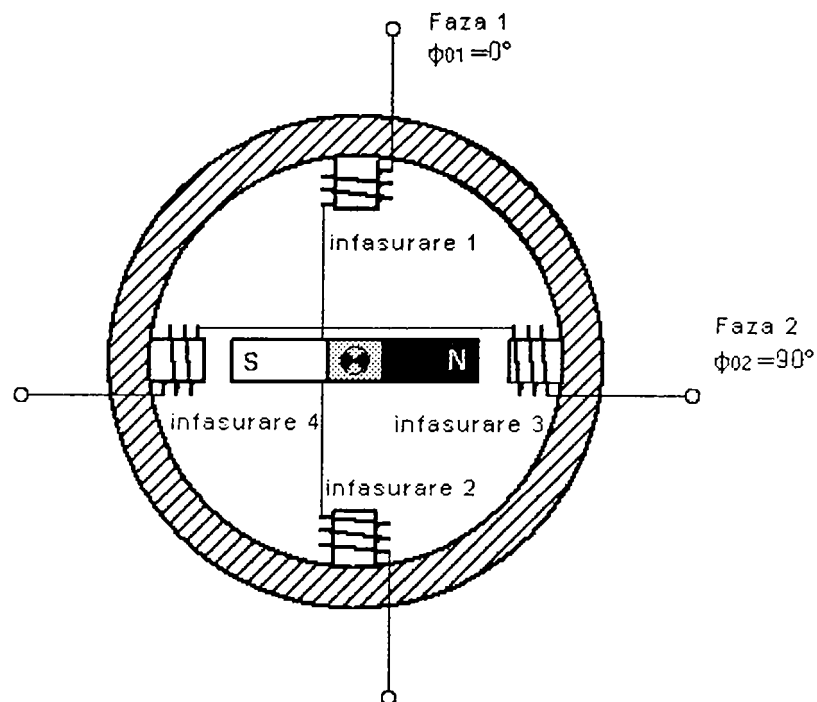


Fig.3.9. Schema de principiu a unui motor pas cu pas cu două faze

Contribuția fazei j la momentul motorului, în cazul în care câmpul magnetic în întrefier are o caracteristică sinusoidală, este dată de:

$$T_{mj} = k_m \cdot \sin(n \cdot \phi(t) + \phi_{oj}) \cdot I_j(t) \quad (3.59)$$

unde:

k_m – constanta motorului, depinde de caracteristicile de proiectare a acestuia;

$\phi(t)$ – poziția unghiulară curentă a motorului;

ϕ_{oj} – poziția bobinei j în stator;

$I_j(t)$ – intensitatea curentului în bobină în funcție de timp.

Curentul $I_j(t)$ în bobină este o funcție de tensiune aplicată U_j și de parametrii bobinei. Relația generală între U_j și $I_j(t)$ este:

$$U_j = F_{emj} + R \cdot I(t) + L \cdot \frac{d}{dt} I(t) \quad (3.60)$$

unde:

F_{emj} – este forța electromotoare indusă în faza j ;

R – rezistența bobinei

L – inductanța bobinei

Forța electromotoare în fiecare bobină poate fi exprimată astfel:

$$F_{emj} = k_m \cdot \sin(n \cdot \phi(t) + \phi_{0j}) \cdot \omega \quad (3.61)$$

unde:

ω_m - viteza unghiulară a rotorului.

Rezistența R și inductanța L sunt aceleași pentru fiecare bobină, deci nu se mai specifică indicele "j".

Aplicând transformata Laplace ecuației diferențiale (3.60), aceasta devine:

$$I = \frac{U}{L \cdot s + R} \quad (3.62)$$

cu s – variabila complexă.

Momentul total produs de MPP este:

$$M_M = \sum_{j=1}^m M_{Mj} \quad (3.63)$$

Momentul total generat de MPP trebuie să învingă momentul redus inerțial $J \cdot \frac{d\omega}{dt}$,

să compenseze cuplul rezistent de frecări vâscoase, și de frecări uscate, precum și momentul de încărcare, rezultând astfel ecuația de mișcare a motorului pas cu pas:

$$\sum_{j=1}^m M_{Mj} = J \cdot \frac{d\omega_m}{dt} + B_{ef} \cdot \omega_m + M_F + \mu \cdot M_e \quad (3.64)$$

în care:

J_{ef} – este momentul de inerție efectiv și este dat de relația (3.42);

B_{ef} – coeficientul global de frecare vâscoasă [Nm/rad/s] este dat de relația (3.47);

M_F – moment produs de frecarea uscată;

M_e – momentul de încărcare.

Pentru MPP cu mai mult de o fază trebuie luate în considerare pozițiile bobinelor.

Pozițiile bobinelor ϕ_{0j} în stator în funcție de numărul de faze m este dat în tabelul 3.5.

Tabelul 3.5. Pozițiile bobinelor ϕ_{0j} în stator în funcție de numărul de faze

Numărul fazelor	m=2	m=3	m=4
ϕ_0 al fazei 1	0	0	0
ϕ_0 al fazei 2	90	60	45
ϕ_0 al fazei 3	-	120	90
ϕ_0 al fazei 4	-	-	135

3.3.4.2.2. Generarea impulsurilor

Generatorul de impulsuri produce comenzi treaptă conform alimentării cu tensiune. Este evident că pentru realizarea în timp a unor secvențe de poziții dorite ale rotorului trebuie asigurate o succesiune bine determinată a tensiunilor pentru fazele 1 și 2, dar în același timp trebuie cunoscută poziția unghiulară la acel moment al rotorului.

Poziția unghiulară a rotorului în orice moment poate fi determinată în 2 moduri:

- ☞ fie prin utilizarea unui traductor de rotație (mai rar);
- ☞ fie prin contorizarea impulsurilor în ambele sensuri.

Numărarea impulsurilor se realizează utilizând contorul:

$$\phi_j = (x - y) \cdot \Delta\phi \quad (3.65)$$

unde:

x – numărul de pași în direcția +;

y – numărul de pași în direcția –.

Poziția comandată ϕ_{i+1} va fi:

$$\phi_{i+1} = \phi_i + \Delta\phi \quad (3.66)$$

Impulsurile pentru ambele faze pentru o rotație sunt date în tabelul 3.6.

Tabelul 3.6. Impulsurile pentru ambele faze pentru o rotație

Poziția rotorului	Tensiunea pentru faza 1	Tensiunea pentru faza 2
$0/360^0$	0	$-u_2$
90^0	u_1	0
180^0	0	u_2
270^0	$-u_1$	0

Impulsul cerut pentru fiecare fază cu ϕ_{i+1} dat este:

$$P_j = \sin(\phi_{i+1} \cdot n + \phi_{0j}) \quad (3.67)$$

Dependența de timp a unei secvențe de impulsuri este descrisă de incrementul de timp t .

$$\Delta t = \frac{|f U t|}{m} \quad (3.68)$$

unde f este un factor de scalare și poate fi folosit pentru a ajusta sistemul. Pentru fiecare pas de timp t poate fi produs un impuls al fazei aferente.

Schema bloc a unei acționări cu motor pas cu pas cu două faze este redată în figura 3.10.

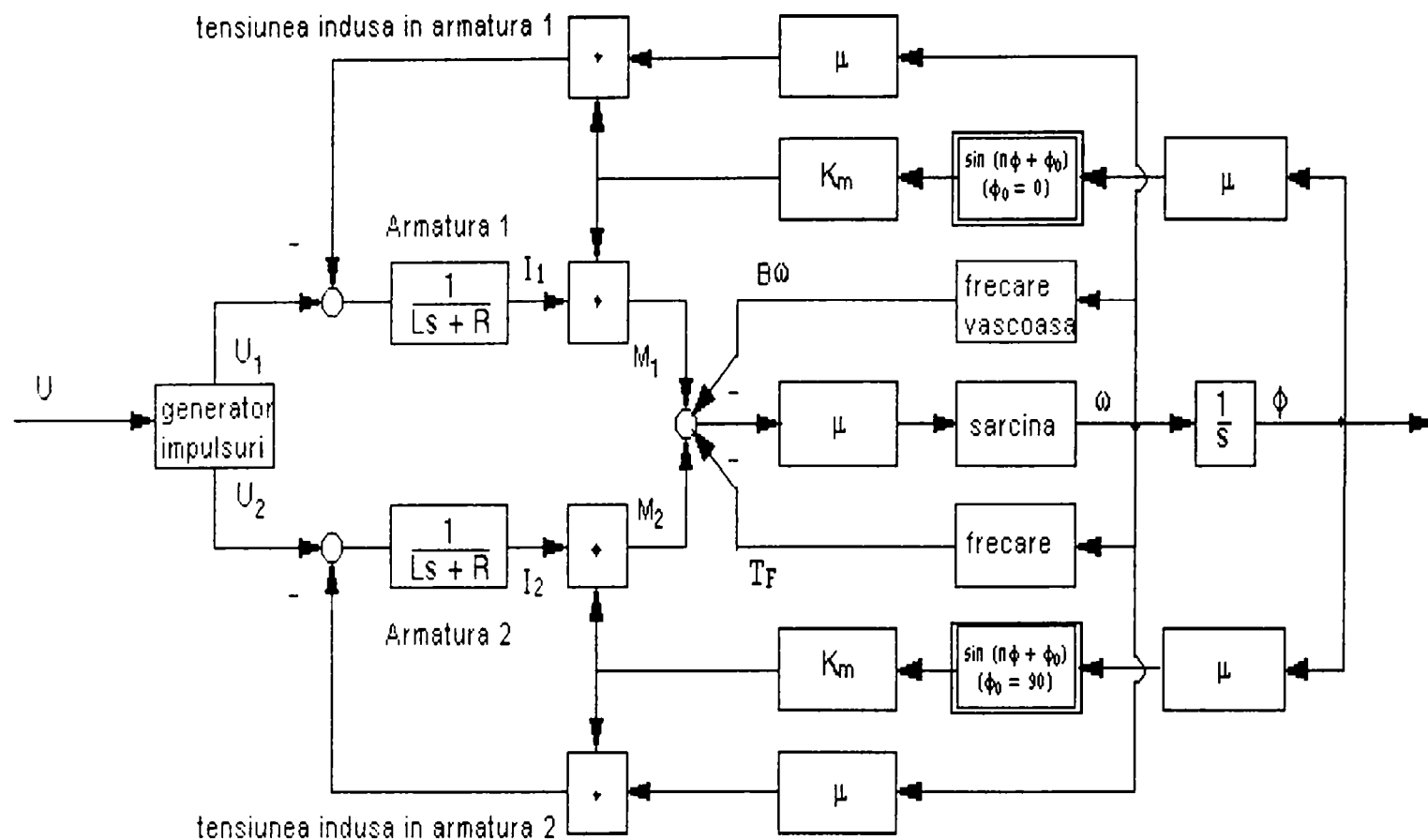


Fig.3.10. Schema bloc a unei acționări cu motor pas cu pas cu două faze

O tensiune U proporțională cu viteza de rotație dorită este utilizată ca intrare pentru generatorul de impulsuri. Cu tensiunile U_1 și U_2 produse de către generatorul de impulsuri sunt alimentate bobinele 1 și 2. Din valorile acestor tensiuni se scad tensiunile induse în cele două armături.

În funcție de succesiunile în care sunt alimentate cele două armături, motorul pas cu pas va prezenta un sens sau altul de rotație. Dacă succesiunea de alimentare a armăturilor este $[(0, -u_2), (u_1, 0), (0, u_2), (-u_1, 0)]$ sensul de rotație a rotorului va fi în sens trigonometric, iar dacă succesiunea de alimentare a armăturilor este $[(0, -u_2), (-u_1, 0), (0, u_2), (u_1, 0)]$ sensul de rotație a rotorului va fi în sens orar.

Elementul numit "sarcină" poate fi orice structură rotită de motor (camera CCD) a cărei moment masic de inerție este J_{ef} .

3.3.5. Modelarea acțiunii hidraulice

Modelarea comportării dinamice a unei cuple acționată de un motor hidraulic comandat de o servovalvă începe pornind de la figura prezentată mai jos.

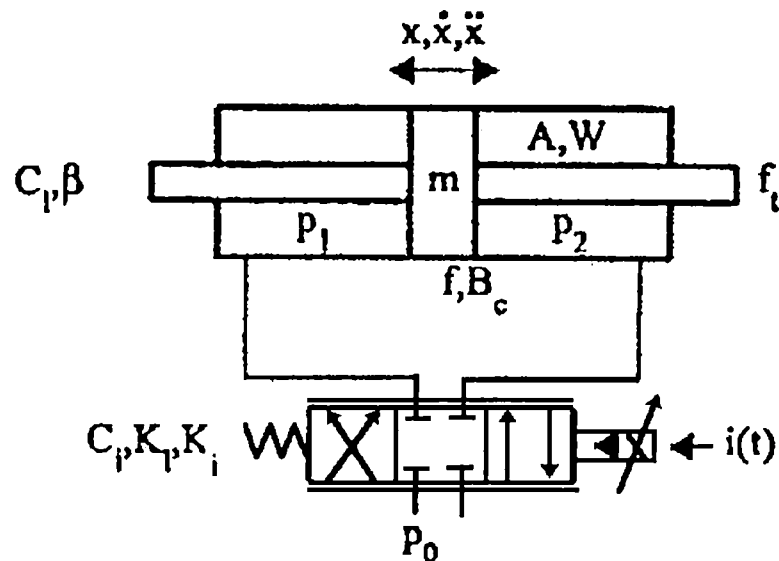


Fig.3.11. Schema acțiunii unui motor hidraulic linear comandat de o servovalvă

Astfel; semnificațiile fizice ale mărimilor prezentate în figura 3.11. sunt următoarele:

- $i(t)$ reprezintă intensitatea curentului de intrare în servovalvă [mA];
- p_0 reprezintă presiunea agentului de lucru [N/m^2];
- W reprezintă volumul de lucru al cilindrului [m^3];
- β reprezintă coeficientul de compresibilitate a fluidului [N/m^2];
- m reprezintă masa pistonului motorului amplasat în cupla "i" [kg];
- x, \dot{x}, \ddot{x} reprezintă cursa, viteza și respectiv accelerația pistonului;
- B_c reprezintă coeficientul de frecare vâscoasă [Ns/m];
- C_i reprezintă panta caracteristicii servovalvei în punctul de lucru;
- A reprezintă suprafața de lucru a pistonului [m^2];
- $C_l = (C_{ie} + C_{ee})/2$ unde C_{ie}, C_{ee} sunt coeficienții de pierdere prin etanșare internă și externă [$(m^3/s)/(N/m^2)$];
- $\Delta p = p_1 - p_2$ diferența de presiune între camerele motorului hidraulic [N/m^2];
- K_l reprezintă coeficientul pierderii prin etanșare a servovalvei [$(m^3/s)/(N/m^2)$];
- K_i reprezintă coeficientul servovalvei din cupla "i" [$m^3/s/mA$];
- f reprezintă forța dezvoltată de motor [N];
- τ_l reprezintă forța externă [N].

Pe baza figurii 3.11 se pot scrie următoarele relații:

$$\underline{Q}_r = A \dot{x} + C_l \Delta p + \frac{W \dot{\Delta p}}{4\beta} \quad (3.69)$$

$$f = A \Delta p = m \ddot{x} + B_c \dot{x} + \tau_l \quad (3.70)$$

$$Q_r = Q_t - K_l \Delta p \quad (3.71)$$

$$K_l i(t) = C_l Q_t \quad (3.72)$$

unde Q_r, Q_t reprezintă debitele reale respectiv teoretice [m^3/s].

Din relațiile de mai sus rezultă ecuațiile modelului dinamic sub următoarea formă:

$$A \Delta p = m \ddot{x}(t) + B_c \dot{x}(t) + \tau_l \quad (3.73)$$

$$K_l i(t) = C_l A \dot{x}(t) + C_l (C_l + K_l) \Delta p + C_l \frac{W \dot{\Delta p}}{4\beta} \quad (3.74)$$

Aplicând transformata Laplace relațiilor (3.73) respectiv (3.74) se obține:

$$(ms^2 + B_c s)x(s) = A \Delta p(s) + \tau_l \quad (3.75)$$

$$\left(\frac{W}{4\beta} \cdot s + C_l + K_l \right) \Delta p(s) = \frac{K_l}{C_l} i(s) - A \cdot s \cdot x(s) \quad (3.76)$$

Schema bloc a unei acționări electrohidraulice se prezintă în figura 3.12.

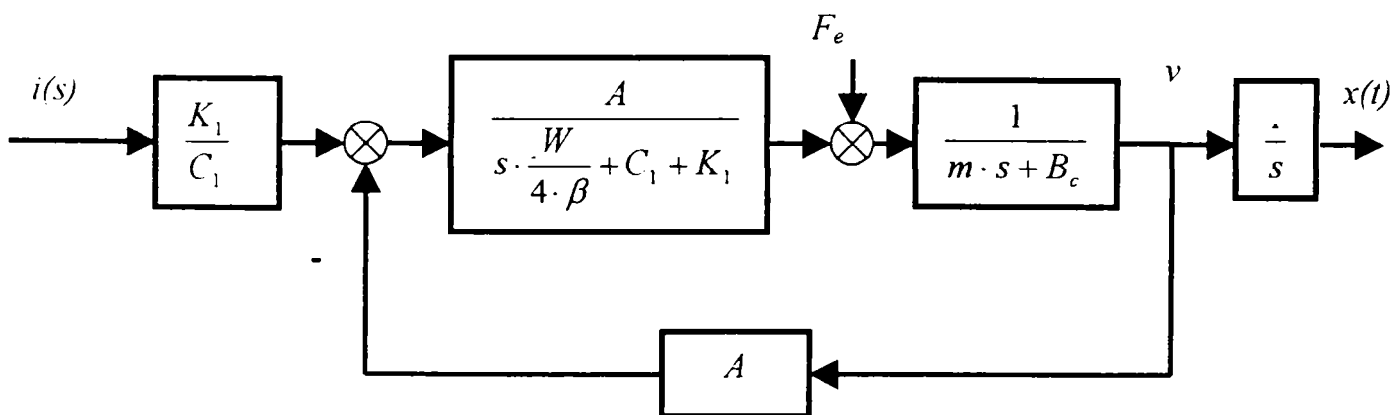


Fig.3.12. Schema bloc a acționării unui motor hidraulic comandat de o servovalvă

Se face observația că structura schemei bloc din figura 3.13 este similară cu cea din figura 3.9. În cazul în care se fac următoarele notații:

$$K_0 = A \quad ; \quad K_1 = \frac{K_l}{C_l} \quad ; \quad K_2 = \frac{A}{C_l + K_l} \quad ; \quad K_3 = \frac{1}{B_c} \quad ; \quad T_1 = \frac{W}{4\beta(C_l + K_l)} \quad ; \quad T_2 = \frac{m}{B_c}$$

ecuația modelului dinamic a unui servosistem electrohidraulic este dată de relația:

$$s[T_1 T_2 s^2 + (T_1 + T_2)s + K_0 K_2 K_3]x(s) = K_1 K_2 K_3 i(s) - K_3 (T_1 s + 1)F_e(s) \quad (3.77)$$

3.4. Modelul dinamic complet al robotului

Din paragrafele anterioare rezultă că modelul dinamic complet al unui robot se constituie din două părți și anume:

- ☞ modelul dinamic al dispozitivului de ghidare;
- ☞ modelul dinamic al acționării (transmisie mecanică și motor).

În cele mai multe cazuri la proiectarea sistemului de conducere al robotului se ignoră sau se simplifică la minimum dinamica părții de acționare, deoarece se presupune că forța generalizată produsă de motor este capabilă de a efectua în condiții normale orice efect asupra mediului.

Astfel sarcina de proiectare se reduce la a determina motorul cu cea mai potrivită forță generalizată, care să confere robotului capacitatea de a urmări orice traiectorie prescrisă.

Modelul dinamic complet al unui robot poate fi prezentat astfel:

- ☞ modelul dinamic al dispozitivului de ghidare:

$$\tau = H(q, \theta) \cdot \ddot{q} + \dot{q}^T \cdot C(q, \theta) \cdot \dot{q} + h_G(q, \theta) \quad (3.78)$$

- ☞ modelul dinamic al acționărilor:

$$(a_0 \cdot s^3 + a_1 \cdot s^2 + a_2 \cdot s) \cdot q_i = b_0 \cdot N(u_i) - (c_1 \cdot s + c_0) \cdot \tau_{ei} \quad (3.79)$$

Parametrii ecuației (3.79) sunt dați în tabelul 3.7. pentru acționarea cu motoare de curent continuu și pentru acționarea electrohidraulică.

În concluzie, problema conducerii robotului este de fapt o problemă de urmărire. Prin urmare sarcina proiectantului este de a determina forța generalizată care urmează a fi dezvoltată de către sistemul de acționare, astfel încât prin conducerea robotului în poziție, viteză, respectiv accelerație, brațul acestuia să urmărească cât mai fidel și cât mai rapid cu putință traiectoria prescrisă fie în coordonatele cuplelor cinematice conducătoare, fie în spațiul cartezian dacă este posibil.

Dar pe lângă o urmărire precisă a traiectoriei este necesar a fi îndeplinite și încă alte câteva criterii și anume:

- ☞ sistemul trebuie să funcționeze în timp real;
- ☞ sistemul trebuie să fie capabil să minimizeze efectul de interconectare care apare între elementele dispozitivului de ghidare;

☞ sistemul trebuie să fie capabil să compenseze variațiile de masă ale obiectului de manipulat (sculei);

☞ sistemul trebuie să fie robust.

Tabelul 3.7. Parametrii modelului dinamic al acționărilor

Parametrii	Motor DC de ordinul trei	Acționare electrohidraulică
a_0	1	1
A_1	$\frac{L \cdot B_{ef} + R \cdot J_{ef}}{L \cdot J_{ef}}$	$\frac{T_1 + T_2}{T_1 \cdot T_2}$
a_2	$\frac{K_i^2 + R \cdot B_{ef}}{L \cdot J_{ef}}$	$\frac{K_0 \cdot K_2 \cdot K_3}{T_1 \cdot T_2}$
c_0	$\frac{\mu^2 \cdot R}{L \cdot J_{ef}}$	$\frac{K_3}{T_1 \cdot T_2}$
c_1	$\frac{\mu^2}{J_{ef}}$	$\frac{K_3}{T_2}$
b_0	$\frac{\mu \cdot K_i}{L \cdot J_{ef}}$	$\frac{K_1 \cdot K_2 \cdot K_3}{T_1 \cdot T_2}$

CAPITOLUL 4 Sisteme senzoriale vizuale

4.1. Noțiuni introductive

4.1.1. Principii

Capitolul introduce unele noțiuni fundamentale legate de formarea imaginii și achiziția ei, noțiuni care sunt relevante în conducerea servovizuală a roboților.

Computer vision reprezintă aplicația unui sistem computerizat dedicat achiziției și procesării informațiilor vizuale din mediu. Ea include două subdomenii vaste și anume: *procesarea imaginii*, respectiv *interpretarea imaginii*.

Primul domeniu realizează îmbunătățirea calității imaginii, astfel încât imaginea rezultată să pună în evidență unele caracteristici dorite de operatorul uman.

Cel de-al doilea, întâlnit în literatura de specialitate sub numele de *machine vision*, este domeniul care prezintă aplicații practice ale primului, în tehnică, de tipul: inspecția și controlul calității obiectelor de lucru și implică extragerea din imaginea achiziționată a unor caracteristici numite generic *trăsături* în vederea utilizării în interpretarea imaginii.

4.1.2. Iluminarea

Natura luminii este una electromagnetică ea având un caracter dual undă și corpuscul. Fenomenele propagării luminii pot fi descrise cel mai bine prin teoria electromagnetică ondulatorie, în timp ce interacțiunea luminii cu substanța, în procesele de emisie și de absorbție, este un fenomen corpuscular [Sear76].

Fotometria este parte a științei radiometriei care se ocupă cu măsurătorile electromagnetice din domeniul vizibil. Prin energie radiantă se înțelege energia transportată de unda electromagnetică [Gherb81].

Energia radiantă care străbate o secțiune perpendiculară pe axa unui con drept - în vârful căruia se găsește sursa de lumină - în unitatea de timp se numește flux de energie radiantă, unitatea de măsură fiind wattul.

Fluxul de energie radiantă evaluat în raport cu senzația vizuală pe care o produce este o mărime biofizică și se numește flux luminos și se exprimă în lumeni.

Raportul dintre fluxul luminos și fluxul de energie radiantă definește luminozitatea, care se măsoară în lumeni/watt.

Curba luminozității fotooptice (răspunsul ochiului la lumină utilizând conul celulelor fotoreceptoare al ochiului) pentru un observator “standard” se prezintă în figura 4.1.

Sensibilitatea spectrală relativă a ochiului este raportul dintre fluxul constant de energie al radiației cu lungimea de undă $\lambda=555$ nm – lungime de undă la care apare cea mai puternică senzație vizuală – și fluxul de energie al radiației cu lungimea de undă care produce aceeași senzație vizuală.

Intensitatea luminoasă I a unei surse punctiforme este fluxul luminos emis în unitatea de unghi solid. Iluminarea se notează cu E și reprezintă fluxul luminos ce cade normal pe unitatea de arie:

$$E = \frac{I}{l^2} \quad [\text{lx}] \quad (4.1)$$

unde l reprezintă distanța de la sursă la suprafața în cauză.

Iluminarea într-o zi cu soare este aproximativ de 10.000 [lx], în timp ce nivelul de iluminare normală a birourilor este de obicei de 1.000 [lx].

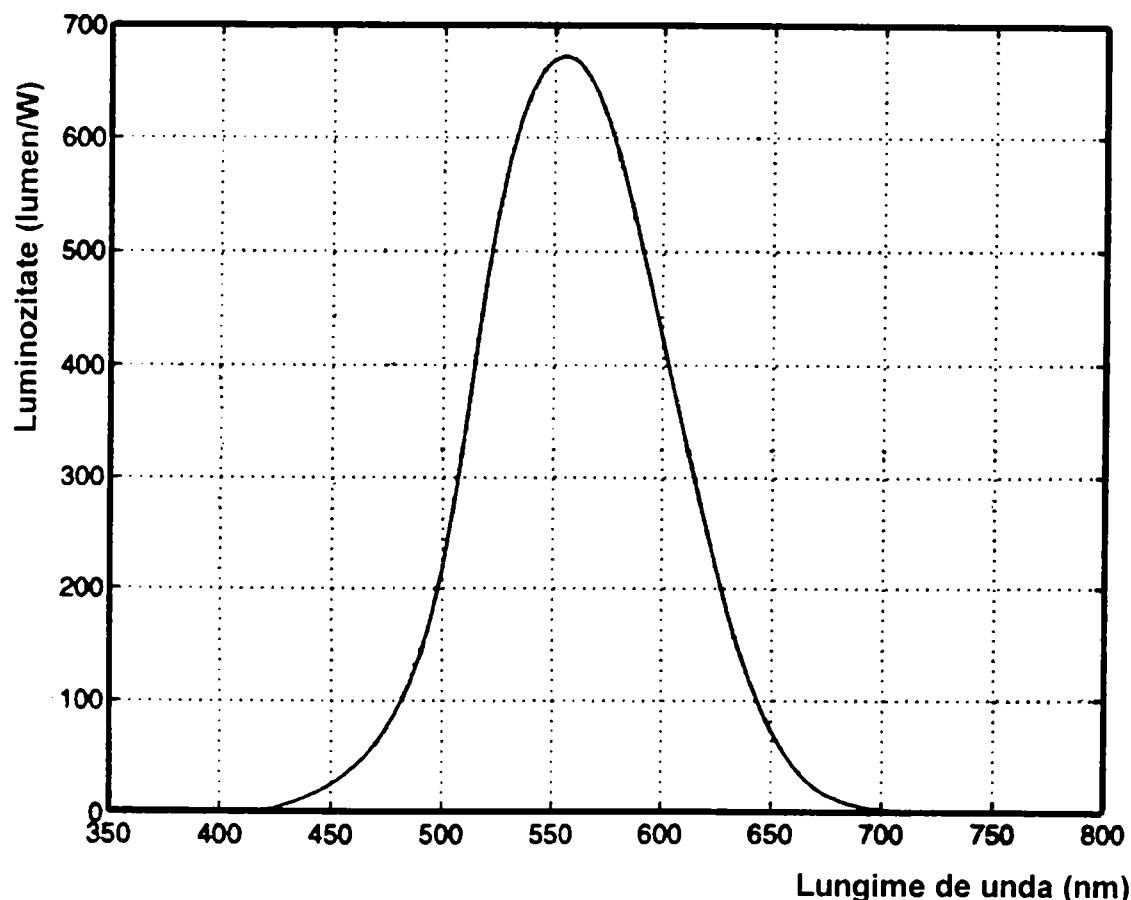


Fig.4.1. Curba luminozității fotooptice pentru un observator “standard”

Luminanța sau strălucirea unei surse se măsoară în nit și este dată de:

$$L_s = E_i \cdot \cos \theta \quad [\text{nt}] \quad (4.2)$$

unde E_i este iluminarea incidentă și θ unghiul format cu normala la suprafață.

4.1.3. Suprafețe reflectorizante

Suprafețele reflectă lumina în diferite moduri funcție de textura suprafeței și lungimea de undă a fascicolului luminos. Cazurile extreme – reprezentate în figura 4.2 – sunt reflexia totală, respectiv reflexia difuză (mată). Legea reflexiei totale (pentru suprafețe tip oglindă) stipulează egalitatea unghiului de reflexie θ_r cu unghiul de incidență θ_i , (conform figurii 4.2a):

$$\theta_i = \theta_r$$

Suprafața difuză împrăștie lumina incidentă în toate direcțiile. O suprafață Lambertiană este o suprafață de difuzie perfectă cu aspect mat ce prezintă aceeași consistență luminoasă indiferent de unghiul de observare (figura 4.2.b). Intensitatea luminoasă a unui punct al suprafeței este:

$$I = rE\theta \cos \theta_r \quad [\text{cd}] \quad (4.3)$$

unde E este iluminarea, iar r gradul de reflexie al suprafeței ($0 \leq r \leq 1$). Pentru o suprafață albă de hârtie gradul de reflexie al suprafeței este de 0,68, pentru griul fotografic de 0,18 și pentru catifea neagră de 0,004.

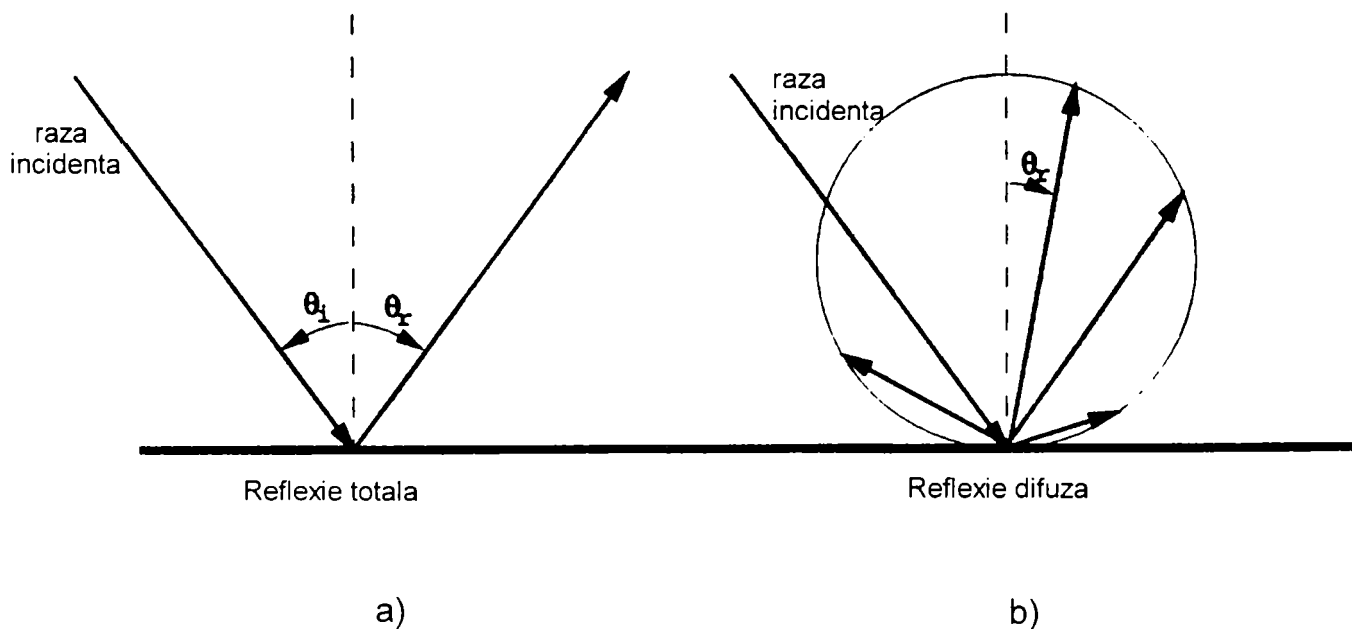


Fig.4.2. Reflexia totală (a) și reflexia difuză (b)

Luminanța (strălucirea) unei suprafețe Lambertiene este dată de:

$$L = \frac{r}{\pi} E \quad [\text{nt}] \quad (4.4)$$

și este independentă de unghiul de incidență sau de cel de reflexie.

În practică, suprafețele reale sunt o combinație a celor două extreme și lumina este reflectată într-o zonă centrată după unghiul de reflexie, care poate fi considerat aproximativ egal cu cel de incidență.

4.2. *Senzori vizuali*

Principial, senzorii vizuali pot fi încadrați în categoria senzorilor optoelectronici, dar se preferă încadrarea lor într-o categorie separată datorită complexității acestora și datorită clasei largi de probleme ce se rezolvă doar cu ajutorul acestor senzori.

În general, senzorii vizuali moderni conțin următoarele tipuri de traductoare video:

- ☞ tuburi receptoare vidate (de tip vidicon, orticon etc.);
- ☞ matrice de fotodiode integrate;
- ☞ traductoare fotoelectrice cu dispozitive cuplate prin sarcină - circuite CCD ("Charge Coupled Devices").

Cei mai răspândiți senzori video sunt cei bazați pe circuite CCD sau pe variante ale acestora.

Senzorii cu dispozitive cuplate prin sarcină (CCD) utilizează conceptul transferului de sarcină, care a fost descris de Boyle și Smith de la Bell Telephone în anul 1970.

Acest principiu constructiv este o variantă aparținătoare tehnologiei dispozitivelor cu transfer de sarcină (CTD - Charge Transfer Device).

Alte variante care pot fi amintite sunt:

- ☞ dispozitive cu acumulare de sarcină BBD (Bucket Brigade Devices);
- ☞ dispozitive cu injectare de sarcini CID (Charge Injection Devices);
- ☞ dispozitive cu sarcini cuplate peristaltice (pCCD – Peristaltic Charge Coupled Device);
- ☞ dispozitive cu sarcini cuplate cu profil peristaltic (p²CCD – Profile Peristaltic Charge Coupled Device).

Din punct de vedere constructiv, senzorul CCD este o matrice de elemente fotosensibile executate pe un cristal semiconductor prin metoda microlitografierii.

Dacă pe suprafața acestui sistem se proiectează un relief luminos, atunci în fiecare element se formează o sarcină localizată de purtători de sarcină minoritară (la potențial minim) și de mărime proporțională cu mărimea fluxului luminos pe elementul dat și cu timpul de acumulare. Distribuția mărimilor sarcinilor acumulate copiază imaginea luminoasă vizualizată.

După expirarea timpului de acumulare, care de obicei este fix, sarcinile sunt deplasate succesiv pe linii și cadre și îndepărtate din structura senzorului prin dispozitivul de ieșire (registru). Informația astfel obținută formează semnalul video.

Construcția principală a unui senzor CCD liniar este prezentată în figura 4.3.

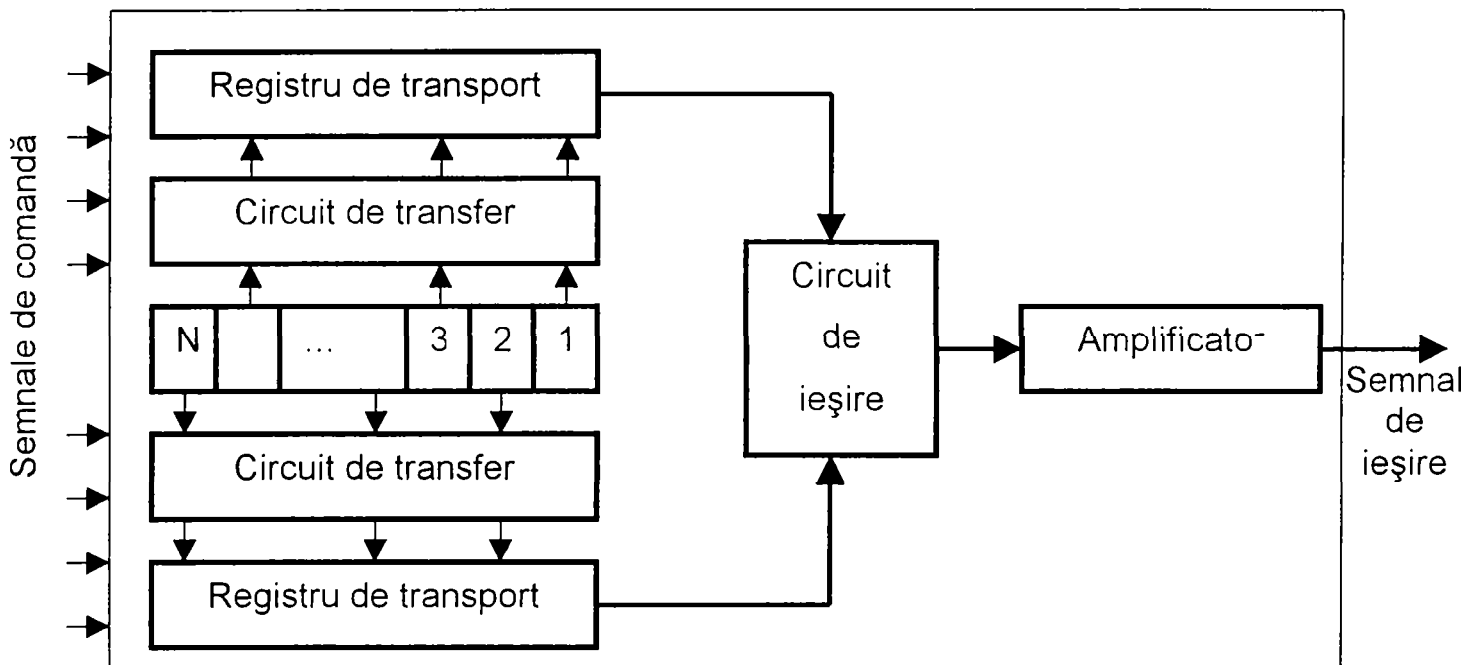


Fig.4.3. Schema bloc a unui senzor CCD liniar

Dintre avantajele acestui tip de senzor se amintesc: reducerea cu un ordin de mărime a masei și a gabaritului, rezistență mecanică mare, fiabilitate ridicată, lipsa necesității alimentării cu tensiune înaltă și a folosirii de tunuri de electroni (filamente încălzite la incandescență). Cel mai mare avantaj îl prezintă precizia geometrică deosebită a imaginii video (datorită legăturii rigide a semnalului video de coordonatele elementelor fotosensibile).

Tehnologia CCD comportă un ansamblu de puncte discrete ale senzorului vizual, fiecare punct discret (imagine elementară) purtând denumirea de pixel.

Senzorii bazați pe acest principiu sunt de două feluri: circuite CCD liniare și circuite CCD matriciale.

Elementele componente sunt următoarele: elementul fotosensibil 1 cu n pixeli, circuitele de transfer 2 și 3, regiștrii de transport 4 și 5, circuitul de ieșire și circuitul de amplificare.

Numărul maxim de pixeli este determinat de lungimea dispozitivului și de ineficiența circuitelor de transfer.

Principial, senzorul CCD bidimensional este constituit dintr-un ansamblu de senzori CCD liniari dispuși în paralel. Dispunerea senzorilor liniari și a circuitelor și registrelor corespunzătoare este prezentată în figura 4.4.

Un astfel de senzor este prevăzut cu circuite de adresare care comută tactul la linia dorită. Funcționarea este similară cu cea a senzorului liniar, deosebirea constând în faptul că la ieșire se obțin cadre formate din linii succesive și nu linii izolate.

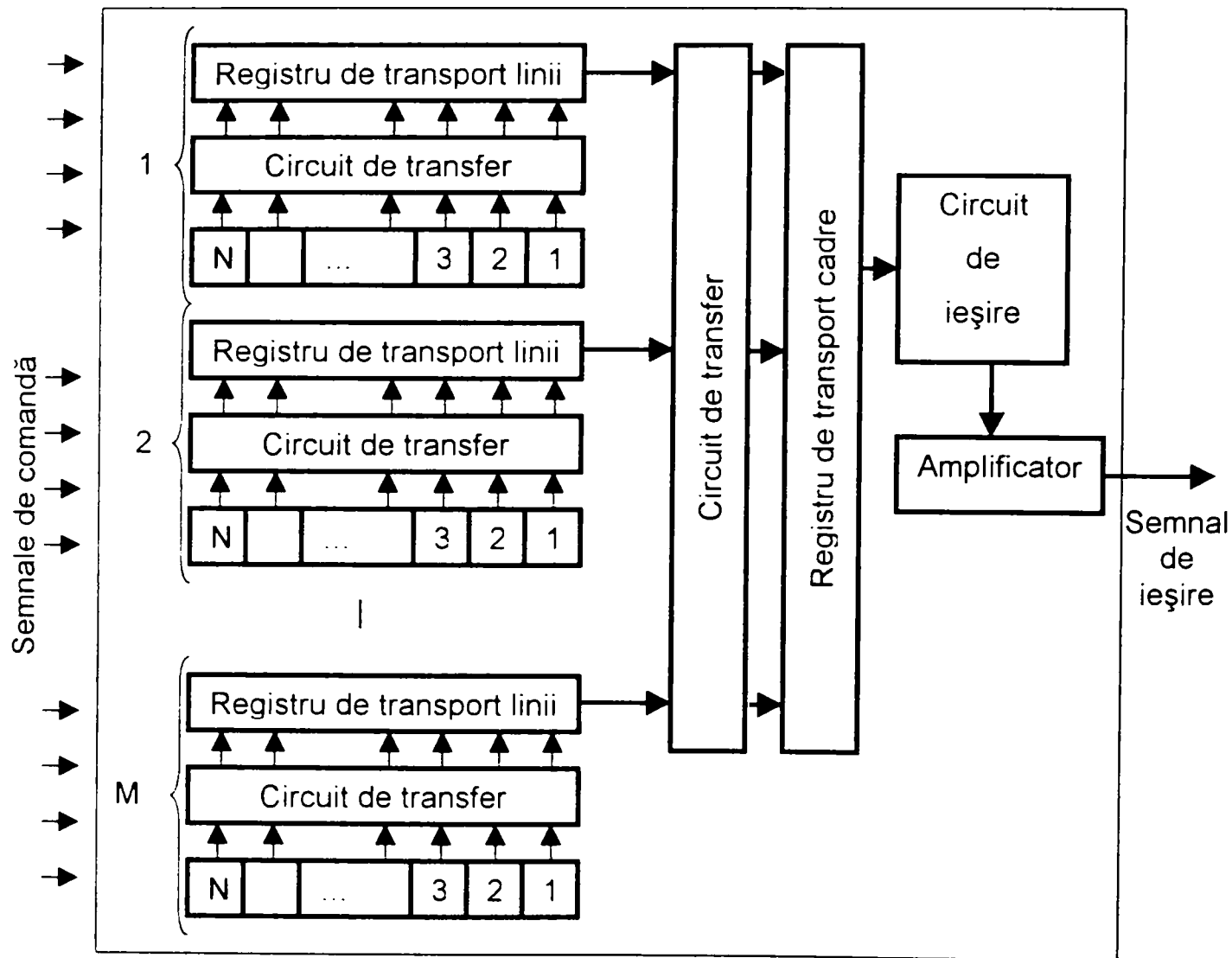


Fig.4.4. Schema bloc a unui senzor CCD matriceal

Avantajele principale ale sistemelor CCD în comparație cu matricele de fotodiode constau într-o rezoluție spațială mai bună, dimensiunea tip a elementului fotosensibil de bază fiind de $10 \times 10 \mu\text{m}$ sau chiar mai mică, precum și prezența unui număr mai mare de elemente pe cristal.

4.3. Formarea imaginii

4.3.1. Transformări în perspectivă

O transformare în perspectivă - care mai poartă numele de transformare de imagine - realizează proiecția punctelor din spațiul tridimensional pe un plan.

Transformările în perspectivă joacă un rol important în procesarea imaginii, deoarece ele dau o aproximare felului în care o imagine este formată, prin observarea mediului înconjurător.

Un model al procesului de formare a imaginii este prezentat în figura 4.5. Se definește sistem de coordonate atașat camerei ($x y z$) acel sistem de referință rectangular drept care are planul imaginii suprapus peste planul xy , iar axa optică - determinată de axa lentilei - în lungul axei z .

Astfel, centrul planului imaginii se află în originea sistemului și centrul optic al lentilei în punctul de coordonate $(0, 0, f)$, cu f - distanța focală a lentilei.

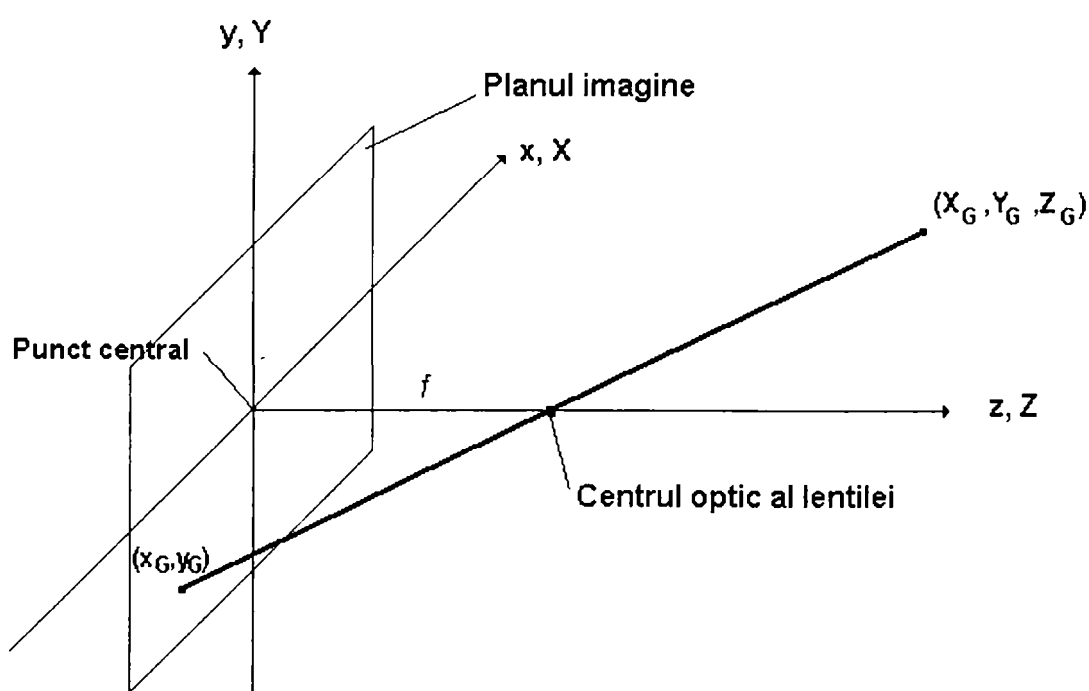


Fig.4.5. Model al procesului de formare a imaginii

Se presupune - în acest paragraf - o suprapunere a sistemului de coordonate legat de cameră și a sistemului de coordonate al universului $^0(X, Y, Z)$. Această restricție va fi

înlăturată în paragrafele următoare.

Fie un punct în spațiul tridimensional cu coordonatele ${}^o(X, Y, Z)$ pe care a fost focalizat camera, scena fiind prezentată în figura 4.5. Se presupune în ceea ce urmează că $Z > f$, adică toate punctele de interes se află în fața lentilei (lentila se află între acest punct și planul imaginii).

Relațiile de legătură între coordonatele (x, y) ale proiecției punctului (X, Y, Z) pe planul imagine și coordonatele acestuia sunt date de relațiile:

$$x = \frac{f \cdot X}{f - Z} \quad (4.5)$$

și

$$y = \frac{f \cdot Y}{f - Z} \quad (4.6)$$

Este important de subliniat ca aceste ecuații sunt neliniare deoarece apare la numitor variabila Z . Deși aceste relații se pot utiliza și sub forma de mai sus, de obicei se va lucra cu ele sub formă matricială. Acest lucru se poate realiza ușor prin folosirea coordonatelor omogene.

Un punct în lumea carteziană poate fi reprezentat și sub forma unui vector, astfel:

$$\underline{w} = [X \ Y \ Z]^T \quad (4.7)$$

iar corespondentul său omogen este dat de:

$$\underline{w}_h = [k \cdot X \ k \cdot Y \ k \cdot Z \ k]^T \quad (4.8)$$

unde k este o constantă arbitrară diferită de zero.

Dacă se definește matricea transformării în perspectivă prin:

$$\underline{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{\lambda} & 1 \end{bmatrix} \quad (4.9)$$

atunci produsul $\underline{P} \cdot \underline{w}_h$ este un vector care se notează cu \underline{c}_h :

$$\underline{c}_h = \underline{P} \cdot \underline{w}_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{\lambda} & 1 \end{bmatrix} \cdot \begin{bmatrix} k \cdot X \\ k \cdot Y \\ k \cdot Z \\ k \end{bmatrix} = \begin{bmatrix} k \cdot X \\ k \cdot Y \\ k \cdot Z \\ -\frac{k \cdot Z}{\lambda} + k \end{bmatrix} \quad (4.10)$$

Elementele matricei \underline{c}_h reprezintă coordonatele punctului reprezentat în sistemul de coordonate legat de cameră exprimate în formă omogenă.

Așa cum s-a precizat anterior aceste coordonate pot fi convertite în formă carteziană prin împărțirea primelor trei componente la al patrulea. Astfel, coordonatele carteziene ale oricărui punct reprezentat în sistemul de coordonate legat de cameră se pot exprima sub forma vectorială:

$$\underline{c} = [x \quad y \quad z]^T = \left[\frac{\lambda \cdot X}{\lambda - Z} \quad \frac{\lambda \cdot Y}{\lambda - Z} \quad \frac{\lambda \cdot Z}{\lambda - Z} \right]^T \quad (4.11)$$

Primele două componente ale lui \underline{c} sunt chiar coordonatele punctului (X, Y, Z) care se proiectează pe planul imagine. Cea de-a treia componentă nu interesează în termenii modelului din figura 4.5, această componentă acționând ca o variabilă liberă în transformarea inversă în perspectivă.

Transformarea inversă în perspectivă duce punctul din imagine înapoi în spațiul tridimensional.

Din relația (4.10) rezultă:

$$\underline{w}_h = \underline{P}^{-1} \cdot \underline{c}_h \quad (4.12)$$

unde \underline{P}^{-1} se găsește ușor a fi:

$$\underline{P}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\lambda} & 1 \end{bmatrix} \quad (4.13)$$

Se presupune un punct oarecare din planul imagine de coordonate $(x_0, y_0, 0)$, unde faptul că în locul lui z apare 0 arată că planul imagine se află localizat la $z = 0$. Acest punct poate fi exprimat ca un vector de poziție omogen sub forma:

$$\underline{c}_h = [k \cdot x_0 \quad k \cdot y_0 \quad 0 \quad k]^T \quad (4.14)$$

Aplicând relația (4.10) rezultă vectorul de poziție omogen în raport cu sistemul de coordonate universal:

$$\underline{w}_h = [k \cdot x_0 \quad k \cdot y_0 \quad 0 \quad k]^T \quad (4.15)$$

sau în coordonate carteziene:

$$\underline{w} = [X \quad Y \quad Z]^T = [x_0 \quad y_0 \quad 0]^T \quad (4.16)$$

Deoarece punctul ales $(x_0, y_0, 0)$ din planul imagine a fost unul oarecare, este absurd să obținem pentru orice punct din spațiul tridimensional cota $Z = 0$.

Eroarea provine din faptul că funcția ce atribuie unei scene tridimensionale proiecția ei pe planul imagine este doar o funcție surjectivă fără a fi și injectivă.

Prin urmare punctul imagine (x_0, y_0) corespunde unei mulțimi de puncte coliniare din spațiul tridimensional. Dreapta ce corespunde acestei mulțimi trece prin punctul imagine $(x_0, y_0, 0)$ și prin centrul lentilei $(0, 0, f)$. Ecuația acestei drepte în sistemul de coordonate universal se obține din ecuațiile (4.5), (4.6).

Deci:

$$X = \frac{x_0}{f} \cdot (f - Z) \quad (4.17)$$

și

$$Y = \frac{y_0}{f} \cdot (f - Z) \quad (4.18)$$

Aceste ecuații arată că nu se poate determina coordonatele punctului care a fost proiectat dacă nu se cunoaște ceva în plus în afară de proiecția sa pe planul imagine.

Această observație poate fi utilizată ca un mod pentru formularea transformării inverse în perspectivă doar prin folosirea lui z ca o componentă a lui \underline{c}_h în locul lui 0. Astfel rezultă:

$$\underline{c}_h = [k \cdot x_0 \quad k \cdot y_0 \quad k \cdot z \quad k]^T \quad (4.19)$$

și

$$\underline{w}_h = \left[k \cdot x_0 \quad k \cdot y_0 \quad k \cdot z \quad \frac{k \cdot z}{f} + k \right]^T \quad (4.20)$$

din care după o conversie în cartezian rezultă:

$$\underline{w} = [X \quad Y \quad Z]^T = \left[\frac{f \cdot x_0}{f + z} \quad \frac{f \cdot y_0}{f + z} \quad \frac{f \cdot z}{f + z} \right]^T \quad (4.21)$$

Cu alte cuvinte tratând pe z ca o variabilă liberă se obțin relațiile:

$$X = \frac{f \cdot x_0}{f + z}$$

$$Y = \frac{f \cdot y_0}{f + z} \quad (4.22)$$

$$Z = \frac{f \cdot z}{f + z}$$

Exprimând pe z funcție de Z din ultima ecuație și înlocuind în primele două expresii rezultă:

$$X = \frac{x_0}{f} \cdot (f - Z) \quad (4.23)$$

$$Y = \frac{y_0}{f} \cdot (f - Z) \quad (4.24)$$

în acord cu observația de mai sus.

4.3.2. Modelul "cameră"

Ecuțiile (4.10) și (4.11) caracterizează formarea a unei imagini prin proiecția punctelor din spațiul tridimensional pe un plan imagine. Astfel, aceste două ecuații constituie modelul matematic al unei imagini într-o cameră de luat vederi. Acest model se bazează pe prezumția că sistemul de coordonate universal și cel legat de cameră coincid. Dar în general aceste două sisteme nu coincid.

În figura 4.6 se prezintă un sistem de coordonate universal ${}^0(X, Y, Z)$ folosit pentru localizarea atât a camerei, cât și a punctelor din spațiul tridimensional, notate cu w .

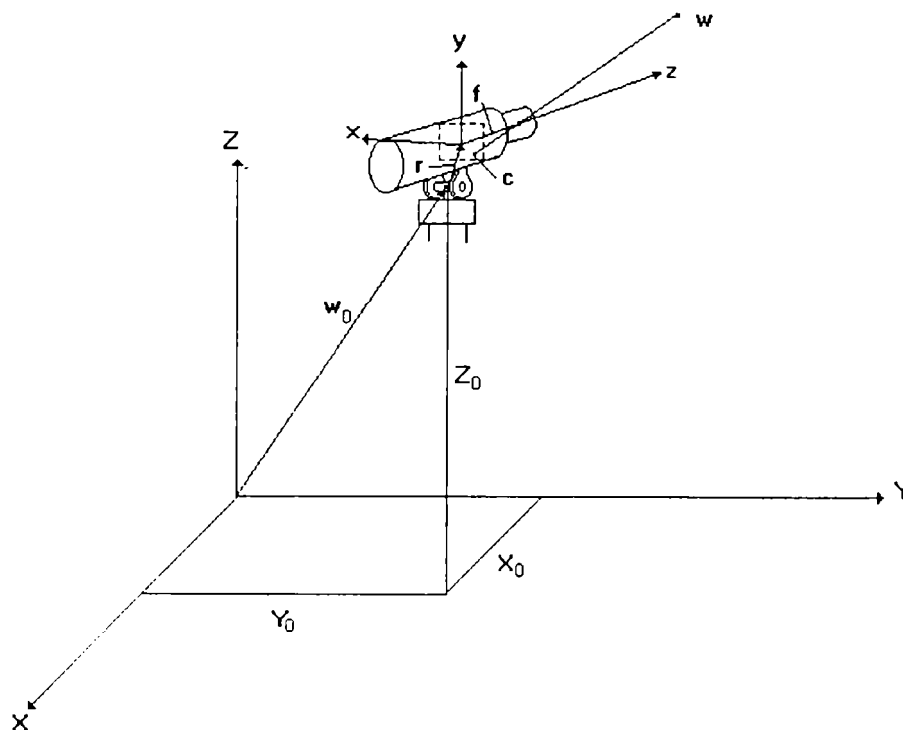


Fig. 4.6. Modelul "cameră"

În figură se prezintă și sistemul de coordonate asociat camerei ${}^0(x, y, z)$, precum și proiecțiile punctelor w pe planul imagine notate cu c . S-a presupus că montarea camerei se face legând-o de un element fix printr-o articulație care permite rotirea acesteia cu unghiul θ și o înclinare a sa cu unghiul α . În această discuție unghiul de rotire se consideră a fi unghiul dintre axele x și X , iar unghiul de înclinare cel dintre axele z și Z .

Poziționarea centrului articulației în raport cu sistemul de referință universal se face cu ajutorul vectorului de poziție w_0 , iar poziționarea centrului planului imagine în raport cu centrul articulației se face cu ajutorul vectorului de poziție r de coordonate (r_1, r_2, r_3) .

Pornind de la cele prezentate în paragraful precedent se trece la obținerea modelului cameră.

Trecerea de la sistemul atașat camerei la sistemul de coordonate universal se face aplicând un set de transformări. După ce acest lucru a fost îndeplinit se vor aplica direct relațiile stabilite în paragraful precedent pentru orice punct din spațiul tridimensional.

Setul de transformări se concretizează în următorii pași: (1) translație după vectorul w_0 ; (2) rotirea cu θ a axei x ; (3) rotirea cu α a axei z ; (4) deplasarea planului imagine în raport cu centrul articulației după vectorul r .

Trecerea de la sistemul de referință universal la cel având originea în centrul articulației se va face prin utilizarea următoarei matrici de transformare:

$$\underline{G} = \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.25)$$

Cu alte cuvinte un punct w_h exprimat în formă omogenă în raport cu sistemul de referință fix și în care are coordonatele (X_0, Y_0, Z_0) poate fi privit ca originea unui nou sistem de coordonate după transformarea $\underline{G} w_h$.

Pentru a exprima rotația axei Ox cu unghiul θ în raport cu axa z se va utiliza matricea transformare:

$$\underline{R}_\theta = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.26)$$

Pentru a roti axa Oz cu unghiul α în raport cu axa x se va utiliza matricea transformare:

$$\underline{R}_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.27)$$

În cele din urmă deplasarea originii planului imagine după vectorul \underline{r} se face utilizând matricea transformare:

$$\underline{C} = \begin{bmatrix} 1 & 0 & 0 & -r_1 \\ 0 & 1 & 0 & -r_2 \\ 0 & 0 & 1 & -r_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.28)$$

În concluzie, un punct având vectorul de poziție \underline{w}_h în reprezentare omogenă raportat la sistemul de referință fix și care este observat printr-o cameră care satisface aranjamentul geometric din figura 4.6, are următoarea reprezentare omogenă în sistemul de coordonate legat de cameră :

$$\underline{c}_h = \underline{P} \cdot \underline{C} \cdot \underline{R}_\alpha \cdot \underline{R}_\theta \cdot \underline{G} \cdot \underline{w}_h \quad (4.29)$$

adică:

$$\underline{c}_h = \begin{bmatrix} k \cdot [(X - X_0) \cdot \cos \theta + (Y - Y_0) \cdot \sin \theta - r_1] \\ k \cdot [(X - X_0) \cdot \cos \alpha \cdot \sin \theta + (Y - Y_0) \cdot \cos \alpha \cdot \cos \theta + (Z - Z_0) \cdot \sin \alpha - r_2] \\ k \cdot [(X - X_0) \cdot \sin \alpha \cdot \sin \theta + (Y - Y_0) \cdot \sin \alpha \cdot \cos \theta + (Z - Z_0) \cdot \cos \alpha - r_3] \\ -\frac{k}{f} \cdot [(X - X_0) \cdot \sin \alpha \cdot \sin \theta - (Y - Y_0) \cdot \sin \alpha \cdot \cos \theta + (Z - Z_0) \cdot \cos \alpha - f - r_3] \end{bmatrix} \quad (4.30)$$

Așa cum s-a indicat în paragraful anterior, coordonatele carteziene (x, y) ale punctului imagine se obțin prin împărțirea primelor două componente ale lui \underline{c}_h la cel de-al patrulea.

Aplicându-se această observație la relația de mai sus se obține:

$$x = -f \cdot \frac{(X - X_0) \cdot \cos \theta + (Y - Y_0) \cdot \sin \theta - r_1}{(X - X_0) \cdot \sin \alpha \cdot \sin \theta - (Y - Y_0) \cdot \sin \alpha \cdot \cos \theta + (Z - Z_0) \cdot \cos \alpha - f - r_3} \quad (4.31)$$

$$y = -f \cdot \frac{(X - X_0) \cdot \cos \alpha \cdot \sin \theta + (Y - Y_0) \cdot \cos \alpha \cdot \cos \theta + (Z - Z_0) \cdot \sin \alpha - r_2}{(X - X_0) \cdot \sin \alpha \cdot \sin \theta - (Y - Y_0) \cdot \sin \alpha \cdot \cos \theta + (Z - Z_0) \cdot \cos \alpha - f - r_3} \quad (4.32)$$

Aceste relații reprezintă coordonatele imaginii a unui punct având vectorul de poziție \underline{w} a cărui coordonate în sistemul de referință sunt (X, Y, Z) .

Se observă că pentru $X_0=Y_0=Z_0=0$, $r_1=r_2=r_3=0$ și $\alpha=\theta=0^\circ$ relațiile de mai sus se reduc la relațiile (4.5), respectiv (4.6).

4.3.3. Calibrarea camerei

4.3.3.1. Principii

După cum se observă din relațiile deduse în paragraful 4.2.2, utilizarea acestora necesită date despre distanțele focale, caracteristicile camerei și unghiurile de rotație și înclinare a camerei.

Deoarece acești parametri pot fi determinați direct, este convenabil de cele mai multe ori - de exemplu când camera se mișcă frecvent - să se determine una sau mai multe din aceste caracteristici, folosind însăși camera ca instrument de măsură.

Prin urmare calibrarea camerei este procesul de determinare a caracteristicilor interne geometrice și optice ale camerei (parametrii intrinseci) și de determinare a situării sistemului de referință atașat camerei față de sistemul de referință de referință universal (parametrii extrinseci).

Caracteristicile cele mai importante ale tehnicilor de calibrare sunt prezentate succint în tabelul 4.1.

Tabelul 4.1. Comparații între cele mai importante caracteristici ale tehnicilor de calibrare

Metodă	Tipul țintei		Distorsiuni
	Coplanară	Necoplanară	
Clasică non-lineară		•	•
Transformare omogenă		•	
“cu două plane”	•		•
“în două etape”	•		•

Metodele diferă atât prin “abilitatea” de calibrare a distorsiunilor lentilei, cât și prin modalitatea ce se necesită pentru calibrarea țintei.

Astfel, pentru generarea unor hărți cu puncte coplanare se pot utiliza imprimante laser, precizia obținută fiind până la 1200 dots/mm.

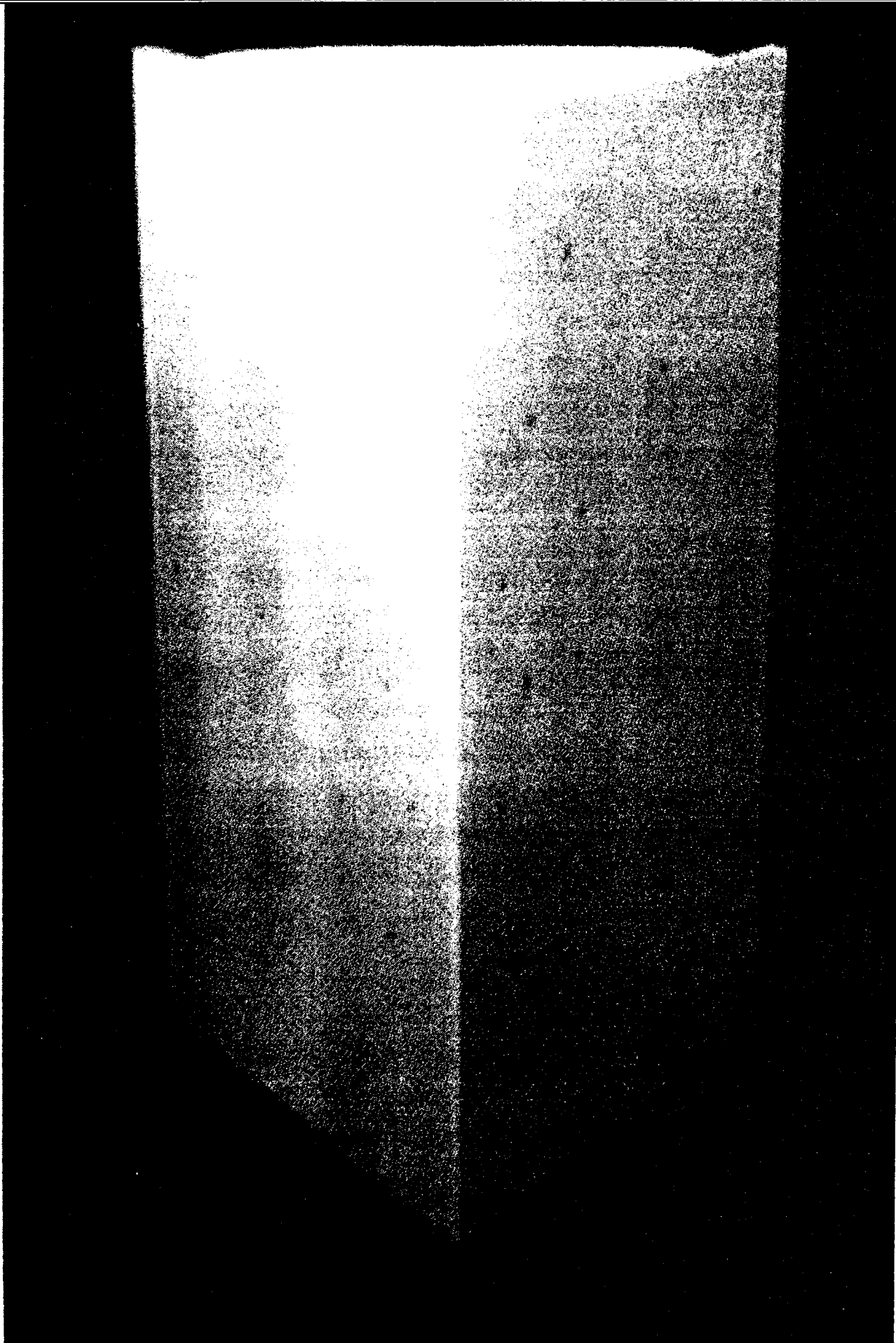


Fig.4.7. Imaginea unui corp de probă utilizat în vederea calibrărilor necoplanare

Tehnicile de calibrare ce utilizează puncte necoplanare necesită fie o structură de calibrare mecanică complexă ca cea prezentată în figura 4.7, fie un dispozitiv spațial de poziționare în raport cu camera a unui reper calibrat.

4.3.3.2. Calibrarea clasică non-lineară

Tehnicile din această categorie au fost dezvoltate în colectivele de cercetători ce s-au ocupat cu fotogrametria și sursele bibliografice în domeniul calibrării clasice non-lineare sunt cele mai vechi [Sobe74], [Wolf74], [Wong75].

Camera este modelată foarte detaliat, în unele cazuri modelele prezentând până la 18 parametrii. Tehnicile de optimizare non-liniare folosesc datele de calibrare pentru o ajustare iterativă a parametrilor modelului.

4.3.3.3. Calibrarea prin transformări omogene

Pentru aceasta se necesită un set de puncte imagine ale căror coordonate în sistemul de referință universal se cunosc, iar procedura de calcul pentru obținerea parametrilor camerei ce folosește aceste puncte cunoscute.

Pornind de la relația (4.29) se notează cu $\underline{A} = \underline{P} \underline{C} \underline{R} \underline{G}$. Elementele matricei \underline{A} conțin toți parametrii camerei, dar din relația (4.29) rezultă $\underline{c}_h = \underline{A} \underline{w}_h$. Luându-se $k = 1$ în reprezentarea omogenă, se poate scrie:

$$\underline{c}_h = \begin{bmatrix} c_{h1} \\ c_{h2} \\ c_{h3} \\ c_{h4} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4.33)$$

și cum se știe că în cartezian coordonatele camerei sunt date de:

$$x = \frac{c_{h1}}{c_{h4}} \quad (4.34)$$

și

$$y = \frac{c_{h2}}{c_{h4}} \quad (4.35)$$

înlocuindu-se $c_{h1} = x \cdot c_{h4}$ și $c_{h2} = y \cdot c_{h4}$ în relația (4.33) și efectuându-se produsul matricial rezultă:

$$x \cdot c_{h4} = a_{11} \cdot X + a_{12} \cdot Y + a_{13} \cdot Z + a_{14} \quad (4.36)$$

$$y \cdot c_{h4} = a_{21} \cdot X + a_{22} \cdot Y + a_{23} \cdot Z + a_{24} \quad (4.37)$$

$$c_{h4} = a_{41} \cdot X + a_{42} \cdot Y + a_{43} \cdot Z + a_{44} \quad (4.38)$$

unde înmulțirea pentru c_{h3} a fost ignorată deoarece ea se referă la z .

Înlocuind c_{h4} în primele două ecuații se obțin două ecuații cu doisprezece coeficienți necunoscuți:

$$a_{11} \cdot X + a_{12} \cdot Y + a_{13} \cdot Z - a_{41} \cdot x \cdot X - a_{42} \cdot x \cdot Y - a_{43} \cdot x \cdot Z - a_{44} \cdot x + a_{14} = 0 \quad (4.36)$$

$$a_{21} \cdot X + a_{22} \cdot Y + a_{23} \cdot Z - a_{41} \cdot y \cdot X - a_{42} \cdot y \cdot Y - a_{43} \cdot y \cdot Z - a_{44} \cdot y + a_{24} = 0 \quad (4.37)$$

Procedura de calibrare constă în următoarele:

- obținerea a $m \geq 6$ puncte la care se cunosc coordonatele (X_i, Y_i, Z_i) , cu $i = 1, 2, \dots, m$ în sistemul de referință universal (se face observația că sunt necesare cel puțin șase astfel de puncte);
- se vizualizează aceste puncte cu camera într-o poziție dată, obținându-se punctele imagine (x_i, y_i) , $i = 1, 2, \dots, m$;
- înlocuind aceste rezultate în ecuațiile de mai sus și rezolvându-le se determină coeficienții necunoscuți.

Pentru rezolvarea sistemului de 12 ecuații se pot găsi un număr foarte mare de modalități de rezolvare a acestuia pe cale numerică.

4.3.3.4. Calibrarea cu “două plane”

Calibrarea utilizând transformările omogene se bazează pe modelul cameră “pinhole” și nu poate fi ușor extinsă și pentru determinarea distorsiunilor lentilei. Metoda de calibrare cu “două plane”, care a fost dezvoltată de [Mart80] înlătură acest neajuns.

Cum se observă și din figura 4.8 o linie în spațiu se definește prin două puncte situate pe cele două plane de calibrare.

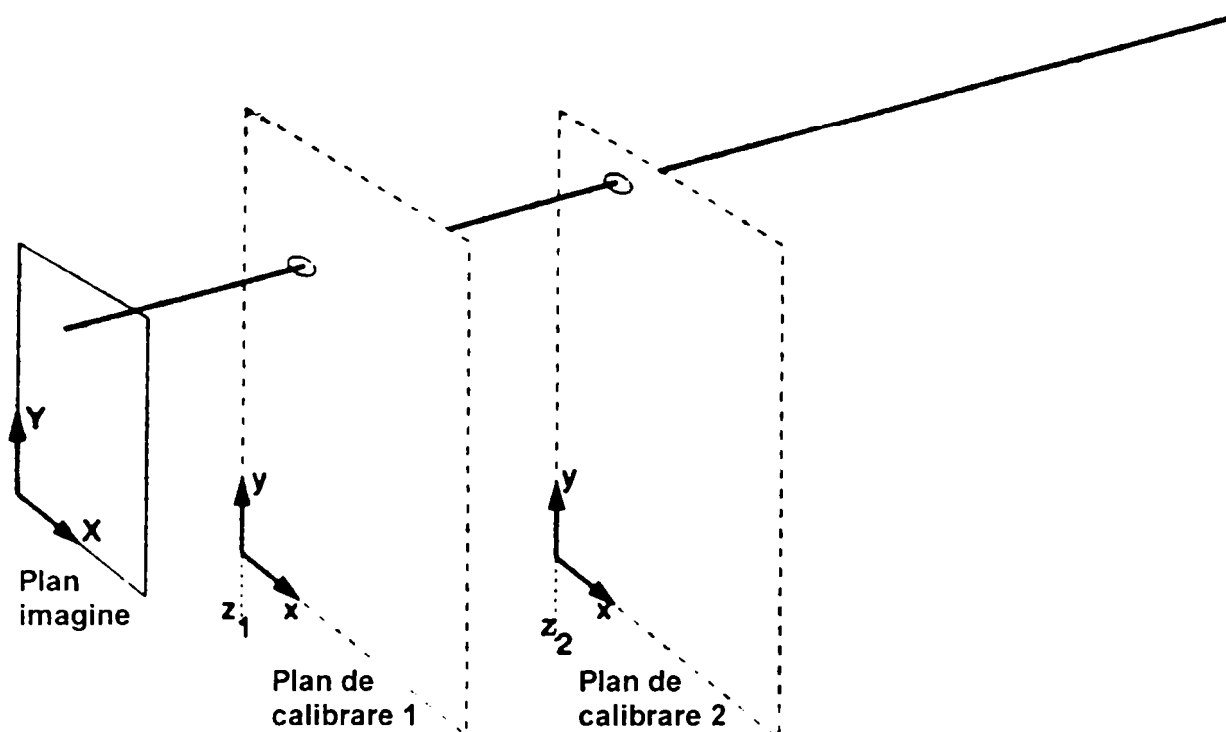


Fig. 4.8. Modelul tehnicii de calibrare cu două plane

Metoda cu două plane de calibrare permite un calcul rapid pentru determinarea transformării inverse de perspectivă, ce este în cazul real o curbă tridimensională ce are ca și origine punctul imagine.

Curba se obține prin interpolare ținând cont de coordonatele punctului imagine, respectiv a punctelor corespunzătoare de pe cele două plane de calibrare.

4.3.3.5. Calibrarea “în două etape”

Metoda de calibrare “în două etape” a fost dezvoltată de [Tsai87] se poate utiliza pentru determinarea atât a parametrilor intrinseci ai camerei cât și a celor extrinseci utilizând o singură poziție de observare a unei ținte plane calibrate.

Metoda se bazează pe așa numita “constrângere de aliniere radială” (radial alignment constraint) ce presupune că distorsiunea lentilei acționează de-a lungul liniilor radiale pornind din punctul central al camerei.

Algoritmul este destul de complex, dar prezintă avantajul unui calcul foarte rapid. Se presupun cunoscute rata de achiziționare a imaginilor de către cameră, coordonatele punctului central al camerei, precum și factorii de scalare specifici matricii de elemente CCD ale camerei.

Metoda este foarte sensibilă la modificarea ratei de achiziție a imaginilor și la precizia de determinare a coordonatelor punctului central al camerei, putându-se eșua foarte ușor la calibrare.

4.3.4. Imaginea stereoscopică

Așa cum s-a specificat și anterior proiecția unei scene tridimensionale pe un plan imagine este o transformare multiplă. Acest lucru înseamnă și faptul că imaginea unui punct nu determină un punct în spațiul tridimensional care să fie bine determinat. În ceea ce urmează se va arăta că informația de adâncime care lipsește poate să fie obținută prin utilizarea tehnicilor stereoscopice.

După cum se arată și în figura 4.9., o imagine stereoscopică implică obținerea a două imagini separate ale unui obiect de interes (de exemplu a unui punct w).

Lungimea dintre centrele optice ale celor două lentile poartă numele de linie de bază și se notează cu B , iar ceea ce se dorește a se obține sunt coordonatele (X, Y, Z) a unui punct w a cărui proiecții pe planele imagine au coordonatele (x_1, y_1) și respectiv (x_2, y_2) .

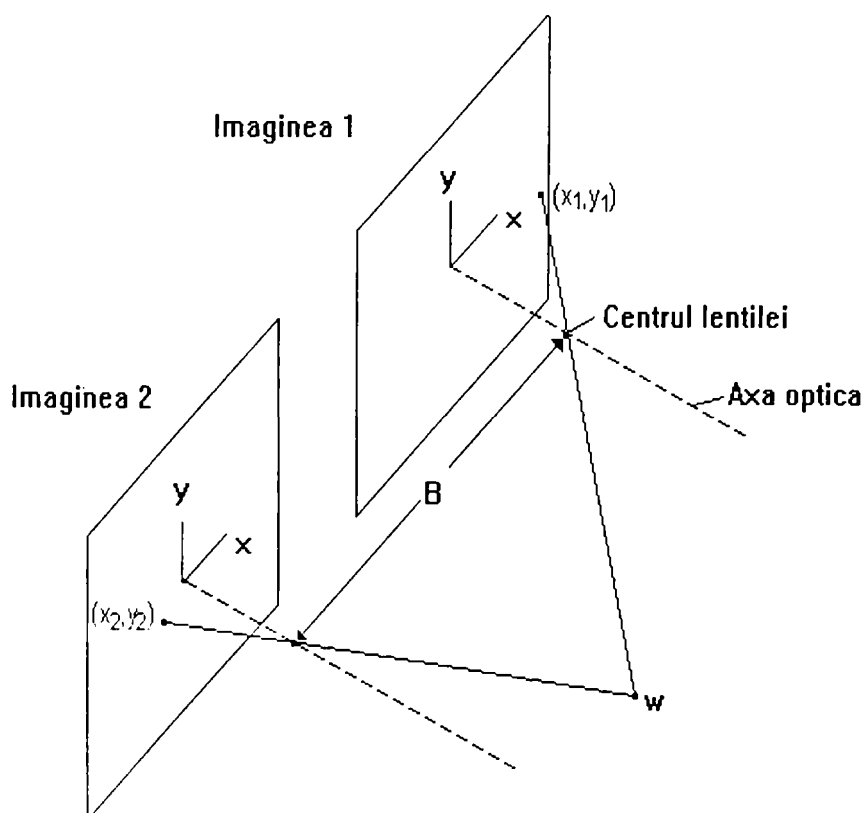


Fig. 4.9. Formarea imaginii stereoscopice

Observație. În ceea ce urmează se presupune folosirea unor camere identice și că sistemul de coordonate ale ambelor camere sunt perfect aliniat, diferind doar plasarea originii lor, condiție întâlnită des în practică.

Revenindu-se la convenția care aduce camera și sistemul de coordonate universal în condiția de suprapunere, planul xy al imaginii va fi aliniat cu planul XY al sistemului de coordonate universal. În acest caz coordonata Z a punctului w este aceeași pentru ambele sisteme de coordonate atașate camerelor.

Se presupune că se aduce prima cameră în coincidența totală cu sistemul de coordonate universal ca în figura 4.10. Atunci, din relația (4.15) rezultă că w se află pe dreapta de coordonate (parțiale):

$$X_1 = \frac{x_1}{f} \cdot (f - Z_1) \quad (4.38)$$

unde indicii lui X și Z indică că prima cameră a fost mutată cu centrul în originea sistemului de coordonate universal, dar păstrându-se aranjamentul din figura 4.10.

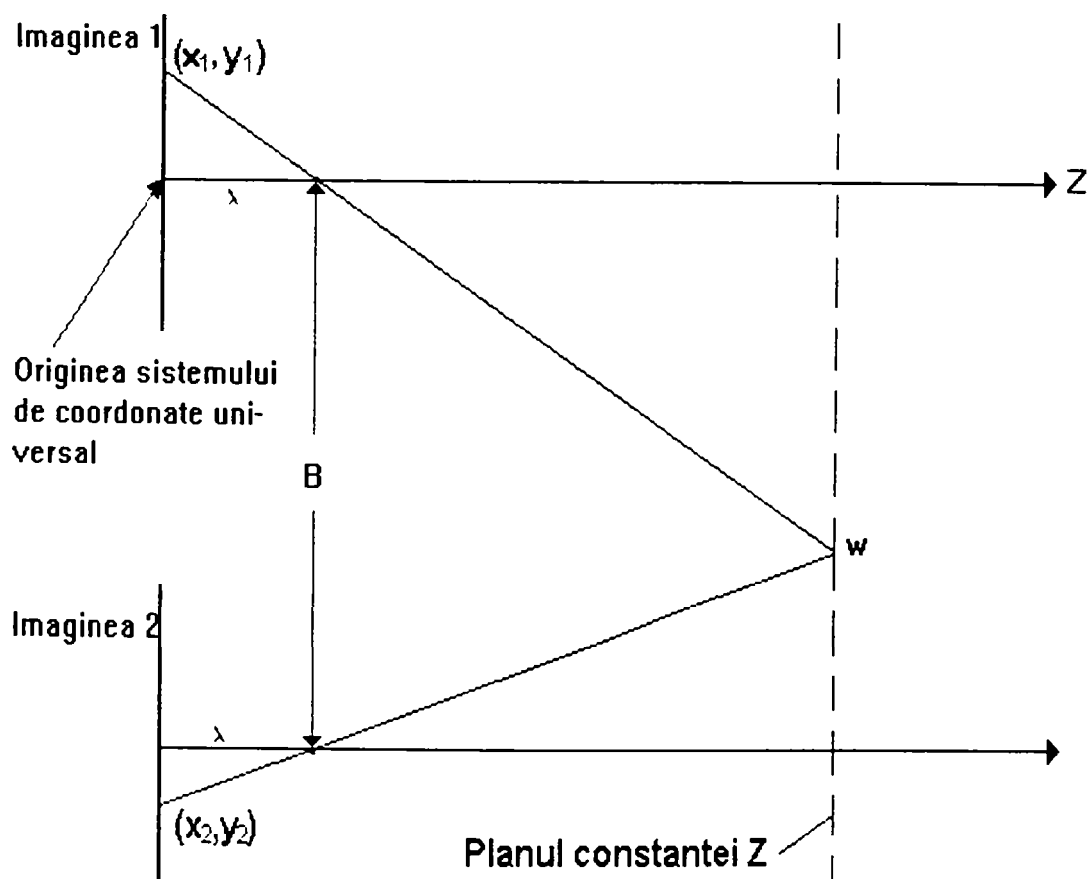


Fig.4.10. Modul de formare a imaginii în vederea stereoscopică

Dacă cea de-a doua cameră ar fi fost adusă în originea sistemului de coordonate universal, atunci ar fi rezultata că w se află pe linia cu coordonatele (parțiale):

$$X_2 = \frac{x_2}{f} \cdot (f - Z_2) \quad (4.39)$$

Datorită distanței ce există între camere și deoarece coordonata Z a punctului w este aceeași pentru ambele sisteme de coordonate, rezultă următoarele egalități:

$$X_2 = X_1 + B \quad (4.40)$$

și

$$Z_2 = Z_1 = Z \quad (4.41)$$

unde B este distanța dintre centrele lentilelor.

Prin substituirea relațiilor (4.40), (4.41) în relațiile (4.38) și (4.39) rezultă următoarele relații:

$$X_1 + B = \frac{x_2}{f} \cdot (f - Z) \quad (4.42)$$

și

$$X_1 = \frac{x_1}{f} \cdot (f - Z) \quad (4.43)$$

Prin scăderea celor două relații se determină cota Z cu relația:

$$Z = f - \frac{f \cdot B}{x_2 - x_1} \quad (4.44)$$

care arată că dacă se poate determina diferența dintre coordonatele imaginilor corespondente x_2 și x_1 și dacă se cunoaște distanța focală și distanța dintre centrele lentilelor B , atunci calculul coordonatei Z a punctului w este o problemă foarte simplă. Coordonatele X și Y rezultă din relațiile (4.17) și (4.18) folosind sau perechea de coordonate (x_1, y_1) sau respectiv (x_2, y_2) .

Cea mai dificilă sarcină în folosirea relației (4.44) pentru determinarea lui Z este de fapt găsirea a două puncte corespondente în imagini diferite ale aceleiași scene. Deoarece aceste puncte sunt în general în aceeași vecinătate, o metodă frecvent utilizată este aceea de a selecta un punct în interiorul unei regiuni mici în una din imagini și apoi de a încerca de a găsi cea mai potrivită regiune din cealaltă imagine folosind tehnica corelației.

Când această scenă conține trăsături distincte, cum ar fi colțuri proeminente, metoda de potrivire a trăsăturilor va da în general o soluție rapidă pentru stabilirea corespondenței.

Observație. Procedura de calibrare dezvoltată în paragraful anterior se poate aplica direct și în cazul achiziționării stereo a imaginilor, prin simpla tratare independentă a camerelor.

4.4. *Structura și funcțiile sistemelor senzoriale vizuale*

Deseori camera este tratată ca și o cutie neagră prevăzută fiind cu câteva "butoane" de ajustare. Dar înainte ca o imagine să fie disponibilă în memoria computerului în vederea prelucrării și analizei, au loc câteva transformări, care sunt prezentate în figura 4.11.

Structura generalizată a unui sistem vizual tehnic se compune din:

- ☛ cameră de achiziție a imaginii, care are în componență: lentila, matricea de senzori de tip CCD, electronica aferentă și respectiv sincronizatorul; ea generează pe baza informației luminoase receptată de la scenă un semnal video analog pe care îl transmite digitizorului;

- ☛ digitizorul, care are în componența sa un separator sync, un bloc care restabilește offsetul semnalului, un filtru, un bloc ce realizează cuantificarea și respectiv un sincronizator, care realizează corespondența între datele transmise în framestore-ul computerului și adresele corespunzătoare.

Structurile sistemelor vizuale industriale sunt de o mare diversitate, rezultată din diversitatea problemelor care se rezolvă prin aceste sisteme.

Pentru folosirea eficientă în condiții industriale, sistemele vizuale trebuie să fie sigure în funcționare, relativ ieftine și comode în exploatare, pentru ca utilizarea lor să fie justificată din punct de vedere economic.

În afară de aceasta, problemele a căror rezolvare necesită utilizare de sisteme vizuale sunt extrem de diverse și nu există o soluție tehnică universal valabilă. Din această cauză, există tendința de a se crea sisteme vizuale modulare, cu o configurație care să poată fi modificată în funcție de problema care trebuie rezolvată.

Din modulele de bază ale sistemelor vizuale fac parte senzorul video, sistemul de iluminare, circuitele de prelucrare a informației video și programul de calcul corespunzător.

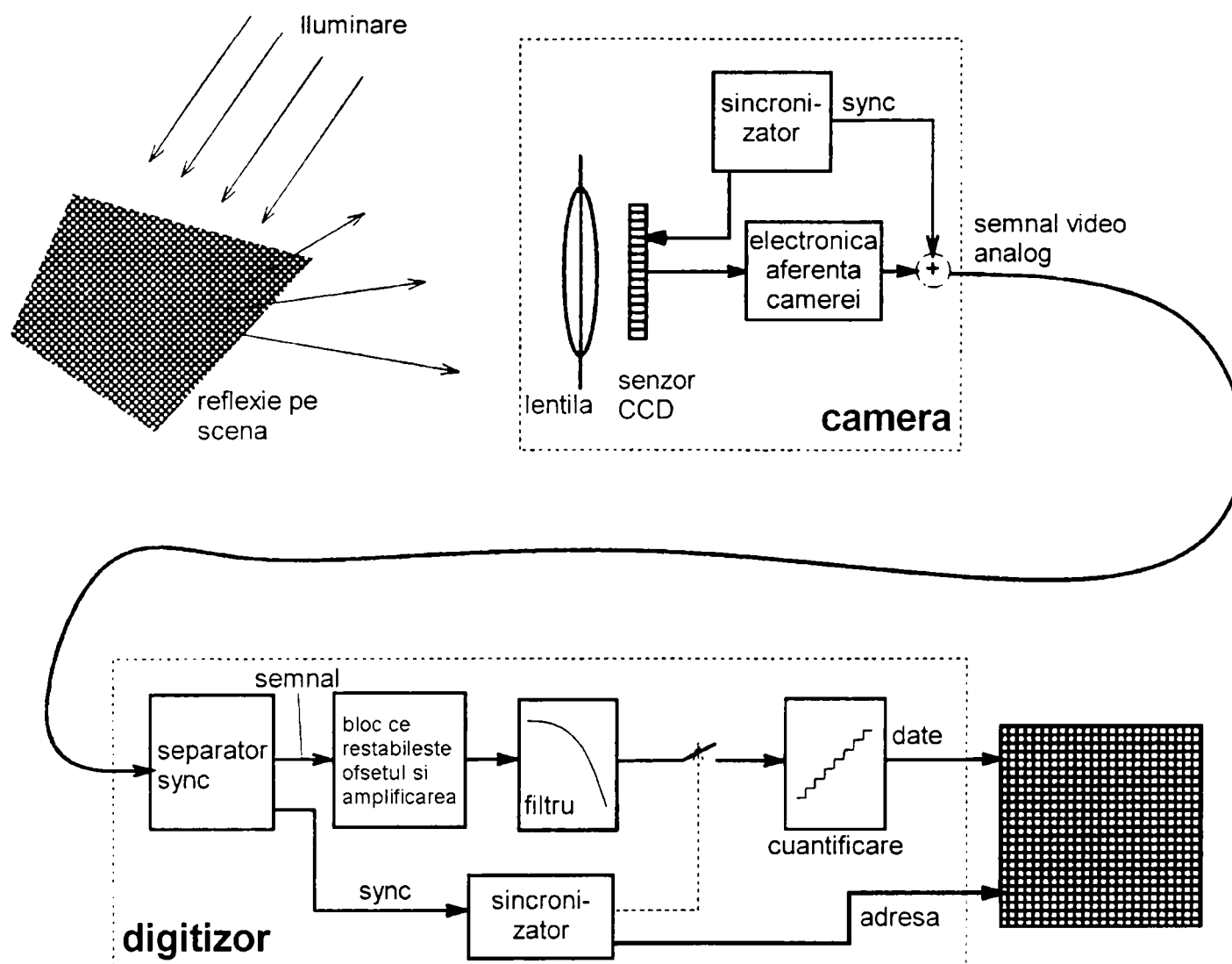


Fig.4.11. Structura unui sistem vizual tehnic

Dacă în cazul operatorului uman – pentru problema discutată - funcția principală o constituie cea de analiză și înțelegere a imaginilor, această funcție fiind realizată de către creier, în cazul sistemelor vizuale, funcțiile principale sunt prezentate în figura 4.12, conform [Nede98].

În cazul sistemelor tehnice – sisteme artificiale - lumea reală este percepută prin intermediul unuia sau mai multor senzori vizuali. Aceștia, în general, achiziționează fie informații bidimensionale de tip intensitate luminoasă, culoare, fie informații tridimensionale de tip informații de profunzime, în scopul creării unui semnal de tip imagine numerică, multimodal, variabil în spațiu și timp.

Prelucrarea imaginii așa cum este prezentată și în figura 4.12. este un proces repetativ aplicabil semnalului primar în vederea îmbunătățirii prin utilizarea unor tehnici de eliminare a zgomotelor și sporire a unor caracteristici.

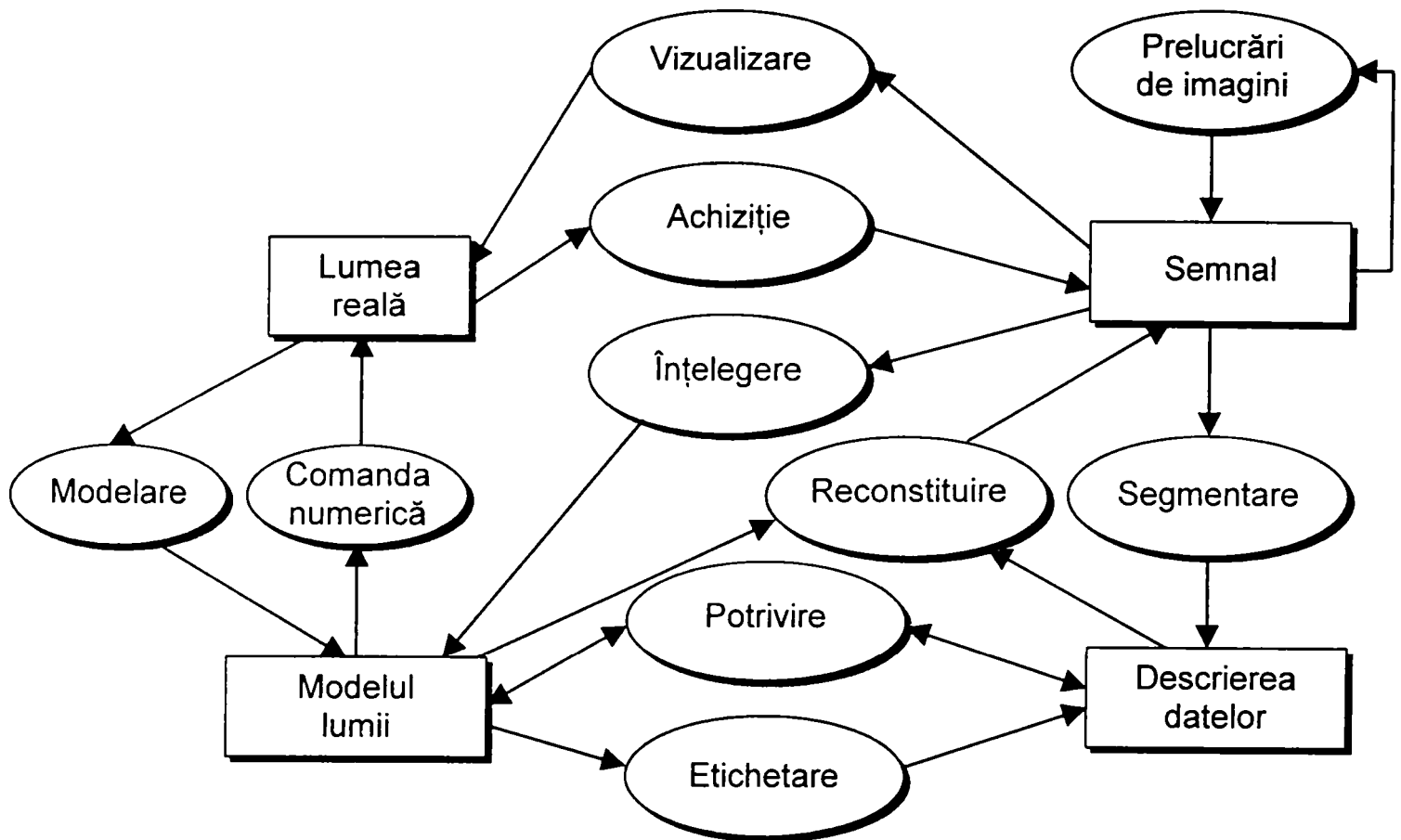


Fig.4.12. Funcțiile principale ale sistemelor vizuale

Segmentarea se aplică semnalului îmbunătățit cu scopul identificării, extragerii și eventual a compresiei informațiilor ce pot fi clasificate funcție de diferite caracteristici alese.

Operatorul uman, utilizând sisteme de modelare, va realiza *modelul lumii*, model care poate reflecta mai mult sau mai puțin fidel structura și caracteristicile lumii reale. Dacă modelarea este corect realizată, între structura, componentele și caracteristicile modelului lumii și semnalele de intrare în sistemul vizual se poate stabili o relație biunivocă.

Stabilirea acestei corespondențe constituie unul din obiectivele principale ale sistemelor vizuale. Interpretarea entităților oferite de procesarea semnalelor de la senzori ca entități ale modelului se realizează prin funcția sistemului vizual de etichetare.

Potrivirea realizează corespondența obiectelor model cu aparițiile lor din scenă, rezolvând fie problema detecției prezenței și localizării unui obiect specificat în scenă, fie problema identificării și localizării tuturor obiectelor din scenă.

Corectitudinea interpretării este oferită de posibilitatea *reconstituirii* semnalului de intrare utilizând cunoștințele oferite de entitățile model detectate în semnal.

Operația inversă, ca o verificare a procesului de segmentare, se realizează prin *reconstituirea* semnalului de intrare din reprezentarea acordată cu senzorul.

Operația de *vizualizare*, operație inversă achiziției, utilizează semnalul imagine pentru reflectarea lumii reale.

Procesul de *înțelegere* suprapune semnalul imagine peste modelul lumii, determinând toate obiectele din scenă și localizările lor.

Operația inversă modelării – *acționarea* - este cea prin care se acționează asupra lumii reale pentru a o pune în concordanță cu modelul lumii. Acționarea se realizează de obicei cu ajutorul mașinilor, din care un rol important îl ocupă roboții.

4.5. Etapele prelucrării imaginilor

Prelucrarea digitală a imaginilor presupune o succesiune de etape de procesare hard și soft, precum și de implementare a unor metode teoretice.

Etapele necesare a fi parcurse pentru realizarea unor prelucrări de imagini sunt prezentate în figura 4.13.

Prima etapă în acest proces este materializată de *achiziția de imagine*. Aceasta necesită un senzor de imagine și capacitatea de a digitiza semnal de la ieșirea acestuia. Senzorul poate fi o camera de televiziune alb negru sau color, care produce o imagine completă la o rată de achiziție care depinde de performanțele senzorului sau o cameră cu scanare liniară, producând la un moment dat o singură linie dintr-o imagine.

În acest caz, mișcarea cu viteză uniformă a obiectului prin fața scannerului duce la obținerea unei imagini bidimensionale. Dacă la ieșirea camerei rezultă un semnal analogic, este necesar transformarea acestuia cu ajutorul unui convertor analogic – numeric în semnal digital.

Cea de-a doua etapă este cea de *preprocesare* a imaginii digitale obținute. Scopul preprocesării este o sporire a calității imaginii în sensul sporirii șanselor de succes ale prelucrării ulterioare. De regulă preprocesarea constă în aplicarea unor algoritmi de îmbunătățire a contrastului, curățare de zgomot și izolare a regiunilor a căror textură indică o asemănare cu informația alfanumerică avută în vedere.

Următoarea etapă este aceea a *segmentării* imaginii, respectiv împărțirea acesteia în zone distincte, caracterizate printr-o omogenitate în raport cu unele caracteristici. De obicei segmentarea este una din cele mai dificile etape în prelucrarea digitală de imagini. Pe de o parte, algoritmi sofisticati de segmentare, cu șanse mari de selectare a regiunilor de interes, necesită calcule complicate și deci se constituie ca un proces mare

consumator de timp, pe de altă parte în cazul în care se utilizează algoritmi simpli sau insuficienți elaborați nu se pot aplica cu șanse mari de succes. Problematika segmentării apare în multe surse bibliografice, rezolvarea ei generând o mulțime de metode.

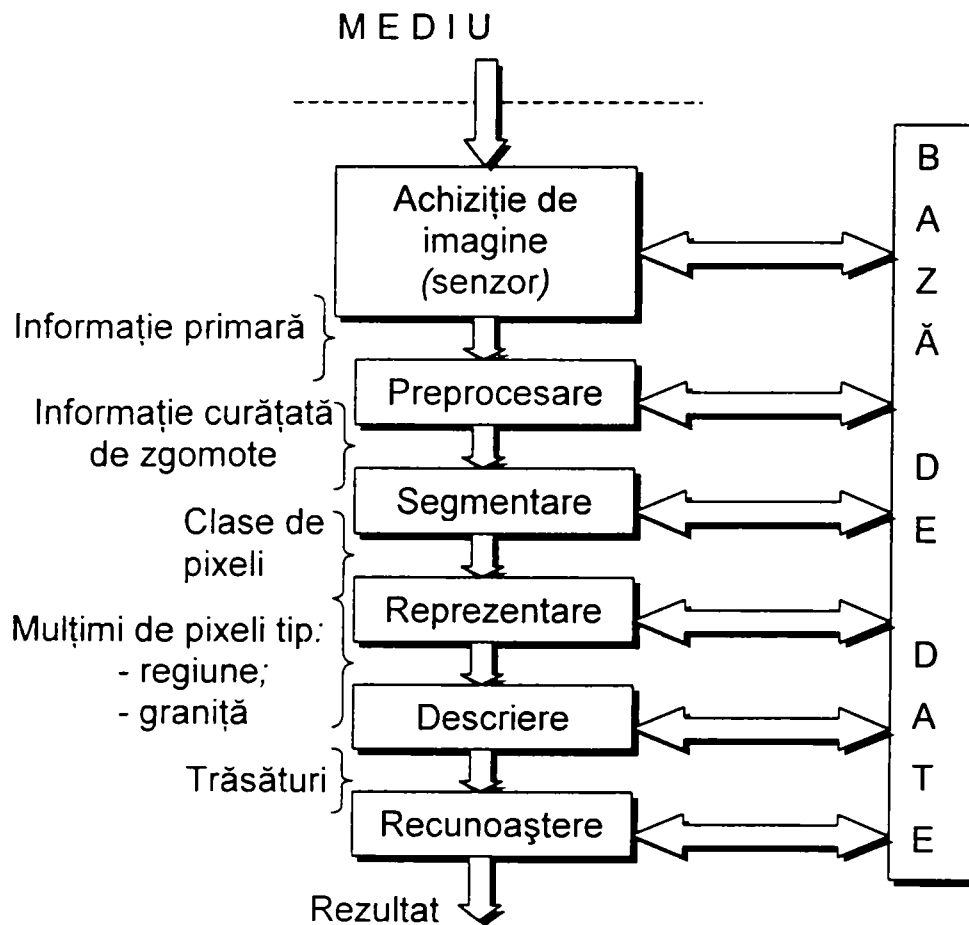


Fig.4.13. Schema bloc a etapelor prelucrării imaginilor

Astfel în [Hara85] se prezintă un algoritm de segmentare aplicabil imaginilor statice, dar care prezintă un inconvenient generat de iterativitatea lui, ceea ce-l transformă într-un algoritm intens cronofag, neputând fi aplicat la rezolvarea problemelor în timp real.

În [Harr89] este descrisă utilizarea culorii în realizarea segmentării pentru o imagine achiziționată de un senzor amplasat pe o mașină de recoltat citrice. În [Hayn91] se propune utilizarea unui "cadru secvențial" care să diferențieze obiectul de interes sau să ignore mediul înconjurător obiectului, în vederea eliminării detaliilor statice din mediu.

Segmentarea binară este simpla clasificare a pixelilor imaginii în două mulțimi (obiect-mediu). În general, aceasta se realizează prin împărțirea pixelilor în două clase, în raport cu o intensitate de prag T :

$$P_{ij} \in \begin{cases} M_o & \text{daca } I(i, j) \geq T \\ M_m & \text{daca } I(i, j) < T \end{cases} \quad (4.45)$$

unde M_o și M_m sunt mulțimile corespunzătoare obiectului, respectiv mediului, $I(i, j)$

intensitatea pixelului situat la coordonatele i și j . Această tehnică non-robustă de segmentare este întâlnită în multe sisteme vizuale ce lucrează în timp real, ca și sistemul de jonglare [Rizz91], robot jucător de ping-pong [Ande87], [Fäss90] sau în sisteme de tip servovizuale [Fedd89], [Hill79], [Cork89].

Pentru o imagine simplă sau pentru una generată prin diferite procedee, segmentarea generează direct obiectele, însă atunci când imaginile sunt complexe rareori se pot diferenția direct obiectele.

În general, rezultatul segmentării este un șir de intensități ale pixelilor, reprezentând fie conturul regiunii de interes, fie întreaga regiune. În fiecare caz este necesară conversia datelor într-o formă mai potrivită pentru prelucrarea pe calculator.

Prima decizie care trebuie luată este dacă datele trebuie să fie reprezentate ca un contur sau ca o regiune completă: reprezentarea ei ca și contur se justifică în cazul în care prelucrarea urmărește evidențierea caracteristicilor de formă (colțuri, inflexiuni, etc.), pe când reprezentarea ca regiune este mai potrivită când se urmărește studiul caracteristicilor interne ale regiunii, cum ar fi textura sau structura internă.

Prin urmare, *reprezentarea* este operația prin care mulțimile spațiale sunt reprezentate într-o formă potrivită unor calcule (prelucrări) viitoare, în general fie ca și regiuni conectate, fie prin granițe.

Alegerea modalității de reprezentare a datelor este urmată de specificarea metodei de descriere a acestora, astfel încât caracteristicile de interes să fie puse în evidență cu prioritate.

Descrierea sau selecția caracteristicilor reprezintă tocmai procesul de selectare a caracteristicilor, având ca rezultat o informație cantitativă sau caracteristici ce diferențiază o clasă de obiecte de altă clasă de obiecte.

Ultima etapă a procesului este aceea de *recunoaștere* și de *interpretare* a datelor.

Prin *interpretarea* imaginii se înțelege descrierea obiectelor fizice dintr-o scenă dată a unei imagini sau a mai multor imagini ale aceleiași scene. O astfel de descriere sub formă numerică poartă numele de trăsături ale imaginii (image features).

Scopul principal al extragerii trăsăturilor imaginii îl constituie reducerea ratei datelor de ieșire de la cameră la un nivel ce poate fi prelucrat de un computer. Prin urmare scopul extragerii trăsăturilor imaginii se confundă cu extragerea esenței scenei respective.

O *trăsătură* a imaginii se definește ca orice relație măsurabilă a unei imagini.

Descrierea obiectelor se face prin intermediul *trăsăturilor globale* sau *locale* extrase din imagini.

Trăsăturile globale reflectă proprietăți globale ale obiectelor. Extragerea lor corectă depinde de achiziționarea unor imagini de bună calitate, ce conțin obiecte simple, perfect separabile între ele și distinctibile de mediu.

Trăsăturile globale la rândul lor pot fi clasificate în: *trăsături geometrice*, *trăsături funcționale* și *trăsături topologice*.

☞ *Trăsăturile geometrice* se împart la rândul lor în:

a) *trăsături caracteristice* care au rol în identificarea obiectelor, iar ele se aleg astfel încât să fie invariabile la transformări rigide. Dintre ele se amintesc:

- A – aria;
 - P – perimetrul;
 - P^2/A – factorul de formă;
 - R_{\max} – raza maximă;
 - R_{\min} – raza minimă;
 - μ_R – raza medie;
 - σ_R – deviația standard a razelor;
 - L_c, l_c – dimensiunile dreptunghiului minim circumscris imaginii;
 - E_{\max} – momentul de inerție maxim al imaginii;
 - E_{\min} – momentul de inerție minim al imaginii;
 - E_{\min}/E_{\max} – factor de alungire;
 - μ - media nivelurilor de gri;
 - σ - deviația standard a nivelelor de gri;
- precum și echivalentele lor tridimensionale;

b) *trăsături de situare* care servesc la determinarea transformării care va supune modelul peste instanța lui din scenă; în această categorie intră:

- x_c, y_c – coordonatele centrului de greutate;
- θ_E – direcția axei de alungire;
- $\theta_{R_{\min}}, \theta_{R_{\max}}$ – direcțiile axelor minime și maxime.

☞ *Trăsăturile funcționale* sunt acelea care realizează o descriere suficient de exactă a obiectelor din imagine utilizând dezvoltări analitice ușor de utilizat cum ar fi:

- coeficienții Fourier ai funcției intrinseci a curbei de contur al imaginii;
- momente de inerție de diferite ordine calculate pe curba de contur: momente de ordin $(p-q)$:

$$m_{pq} = \iint_{\text{imagine}} x^p \cdot y^q \cdot I(x, y) dx dy \quad (4.46)$$

pentru o imagine binară m_{00} reprezintă aria imaginii, iar (m_{10}, m_{01}) este centroidă acesteia.

- Jang [Jang91] oferă o definiție formală a trăsăturii unei imagini prin funcționala:

$$f(x, y) = \iint_{\text{imagine}} F(x, y, I(x, y)) dx dy$$

unde $I(x, y)$ reprezintă intensitatea pixelului în poziția (x, y) ; funcția $F()$ se generează printr-o mapare liniară sau neliniară funcție de trăsătură, putând include și funcția delta;

- invarianții calculați pe baza momentelor de inerție ale imaginii.

☞ *Trăsăturile topologice* exprimă proprietăți ale obiectelor invariante la deformările care nu afectează suprafața, ele fiind independente de distanțe:

- C – numărul componentelor conexe;
- H – numărul cavităților;
- E=C-H – numărul lui Euler;
- graful componentelor conexe.

Trăsăturile locale reflectă proprietățile locale ale obiectelor. Din această cauză ele își păstrează relevanța în cazul truncherilor sau acoperirilor parțiale și sunt mai puțin sensibile la zgomote. Dacă dintr-un set de trăsături locale caracterizând un obiect, o parte sunt acoperite, informațiile conținute de trăsăturile vizibile nu vor fi afectate.

Recunoașterea reprezintă procesul de clasificare a unui obiect într-o anumită categorie, pe baza informației rezultate în urma descrierii imaginii segmentate.

După cum se observă și din figura 4.13. pentru desfășurarea în bune condiții a procesului sunt necesare unele cunoștințe anterioare stocate în baza de cunoștințe. Între această bază de date și celelalte module ale procesului există o strânsă interdependență.

Astfel, cunoștințele necesare pentru rezolvarea unei anumite probleme prin prelucrarea digitală de imagini sunt codificate în sistem sub formă de baze de date. Cunoștințele pot fi fie de tip simplu, de exemplu detalierea unei regiuni dintr-o imagine, unde se presupune că este localizată informația de interes, limitându-se în acest fel procedura de căutare a respectivei informații, fie pot avea un caracter mult mai complex, ca de exemplu o listă a defectelor interne mai importante ale unor produse, în aplicații de tipul inspecției automate sau o bază de date de imagini conținând imagini de înaltă rezoluție preluate de la sateliți, în cazul aplicațiilor de tip detecție din satelit.

Baza de date permite și asigurarea unei reacții între module, pe lângă dirijarea operațiilor în modulele de prelucrare.

Se face remarca – de altfel normală – că majoritatea aplicațiilor prelucrărilor de imagini au o structură mult mai simplă și nu necesită interacțiuni de complexitatea celor prezentate în figura 4.13., iar uneori nici nu sunt necesare toate modulele prezentate în figură.

4.6. *Programe de prelucrare digitală a imaginilor*

Scrierea unui program de prelucrare de imagini presupune existența unui anumit număr de pași.

Primul constă în stabilirea funcțiilor de bază și a operațiilor caracteristice – proiectarea conceptuală – urmată de identificarea celor mai potriviți algoritmi pentru implementarea acestor funcții de baza.

Cel de-al doilea pas este acela de arhivare a programului, obținând astfel prima variantă a programului. După lansarea acestui produs urmează faza testării lui pentru eliminarea problemelor care apar, fiind posibilă încorporarea unor idei noi ce pot conduce la mărirea fiabilității, a performanțelor. Se recomandă ca, odată cu programul, să fie lansată și documentația de folosire a acestuia.

În ultima etapă activitățile vor include servicii de întreținere și de suport tehnic.

Programele dedicate prelucrărilor de imagini implică manipularea unui număr mare de fișiere de imagine, de dimensiuni foarte mari. Acest lucru necesita atât compresia cât și conversia acestor fișiere, întrucât adesea este necesar ca utilizatorii să schimbe date prin intermediul lor.

Organizarea fișierelor de imagini se face în fișiere standard. În tabelul 4.2.se prezintă formatele de imagine standard.

Tabelul 4.2. Formate de imagine standard

Nume	Siglă	Utilizare
Tagged image file format	*.TIF	DOS,UNIX și imagini sub MacIntosh
Encapsulated PostScript	*.EPS	Format dedicat publicațiilor industriale
Graphical interchange format	*.GIF	Format grafic comprimat
Bit-Mapped format	*.BMP	Format realizat de Microsoft
Presentation manager	*.BMP	Format BitMap dedicat IBM OS/2
MacIntosh	*.PICT	Pentru imagini Apple MacIntosh

Primele calculatoare nu aveau interfață cu utilizatorul, programarea făcându-se dificil, prin intermediul codurilor specifice mașinii de calcul.

Cel mai vechi și simplu mod de a realiza o interfață cu utilizatorul este de a realiza un interpretor de comenzi. Acesta folosește un limbaj în exclusivitate textual și pretinde utilizatorului cunoașterea acestuia, fie din memorie, fie din documentație.

Fiecare comandă este interpretată, fiind lansată o rutină corespunzătoare, ce va rezolva o anumită problemă specifică. Dezavantajul este că nu se pot oferi indicații programatorului referitor la corectitudinea programului decât în timpul execuției acestuia. Situația poate fi îmbunătățită prin folosirea unor fișiere de tip "*.bat" – batch files – care specifică o serie de pași de prelucrare și pot fi apelate printr-o singură comandă. Un set bogat de asemenea fișiere poate face lucru cu interpretorul de comenzi mult mai facil.

Incluse în a doua generație, interfețele de tip meniu oferă pe ecranul monitorului – în interiorul unor ferestre – o listă de facilități puse la dispoziția utilizatorului, care pot fi selectate direct cu ajutorul unor taste definite în acest scop. Meniul poate fi definit ca o structură ierarhică, prin selectarea unui submeniu pentru anumite opțiuni ce înglobează la rândul lor alte opțiuni, ș.a.m.d.

A treia generație de interfețe este cea grafică (GUI – "graphical user interface"). Utilizatorul poate controla un proces prin utilizarea unui limbaj vizual-grafic, în locul celui textual. Interfața grafică prezintă de fapt opțiunile pe ecranul monitorului nu doar sub formă de text, ca în meniu, ci și sub formă grafică (simboluri grafice - iconițe). Aceste iconițe reprezintă nu numai procese, ci și date (imagini digitale, spre exemplu) sau chiar echipamente hard (driver de disc, imprimantă, etc.). Folosind o interfață grafică se lansa un program prin așa numita tehnică "drag-and-drop".

Interfața cu fluxul de date este un alt tip de interfață grafică. Aici este utilizat în exclusivitate limbajul vizual (grafic), fiind disponibile niște simboluri grafice (glifuri – "glyphs") reprezentând procese, date sau echipamente. În acest caz utilizarea tehnicii "drag-and-drop" cu ajutorul mouse-ului duce la compunerea pe ecranul monitorului a unei scheme-bloc descriind succesiunea dorită de algoritmi de prelucrare. Fiecare glif are una sau mai multe intrări și/sau ieșiri care pot fi conectate prin linii trasate cu dispozitivul poantor la alte glifuri.

O altă alternativă mai modernă este folosirea unor monitoare de dimensiuni mari pentru a vizualiza mai multe funcțiuni simultan prin folosirea mai multor ferestre grafice. MDI ("multi device interface") este o interfață care oferă această facilitate și care utilizează în mod eficient monitorul. Imaginile sau funcțiunile sunt reprezentate în câte o fereastră,

care poate avea orice dimensiune și poate fi plasată oriunde pe ecran.

Tendința în proiectarea interfețelor cu utilizatorul este de a folosi mai mult limbajul grafic, în detrimentul celui textual, deoarece interfața grafică este mult mai ușor de învățat, în special de către începători, dar și mult mai ușor de utilizat de către specialiști.

Capitolul 5 Contribuții la concepția unor sisteme de conducere senzorială a roboților utilizând informații furnizate de senzori vizuali

5.1. Noțiuni introductive

Pornindu-se de la primul obiectiv al tezei de doctorat și anume acela de a realiza un studiu teoretic al conducerii senzoriale a roboților utilizând ca buclă de reacție informațiile furnizate de senzori vizuali, autorul a conceput două sisteme robot la care controlul situației efectorului final să se poată realiza prin intermediul unor sisteme care au în componență senzori vizuali.

Metoda propusă de către autor are la bază următorul principiu: senzorii vizuali achiziționează informații despre situațiile reale ale efectorului final (interacțiunea sa cu mediul), pe care le transmit sistemului de comandă al robotului, care generează comenzi de corecție a situațiilor reale, astfel încât efectorul final să atingă cu o precizie controlabilă situațiile programate.

Conform definițiilor din ISO 8373 *robotul adaptiv* se definește ca fiind acel robot care prezintă funcții specifice *conducerii senzoriale*, *conducerii adaptive* sau *conducerii prin învățare*, iar prin *conducere senzorială* se înțelege acea schemă de conducere în cadrul căreia mișcarea robotului sau forța cu care acesta acționează asupra mediului este ajustată în raport cu mărimile de ieșire de la un senzor extern.

Sistemul conceput de către autor utilizează senzori vizuali externi în vederea conducerii robotului, prin urmare robotul astfel conceput prezintă o conducere senzorială, iar robotul este un robot adaptiv.

Schema bloc a sistemului robot care utilizează o metodă de control a situației conform principiului enunțat mai sus se prezintă în figura 5.1.

Deoarece schema bloc a robotului astfel concepută prezintă un regulator global, care compară situațiile prescrise cu cele reale obținute de la subsistemul senzorial vizual și care utilizează informația astfel obținută pentru comanda sistemului de acționare a robotului, conducerea robotului este în fapt o conducere globală.

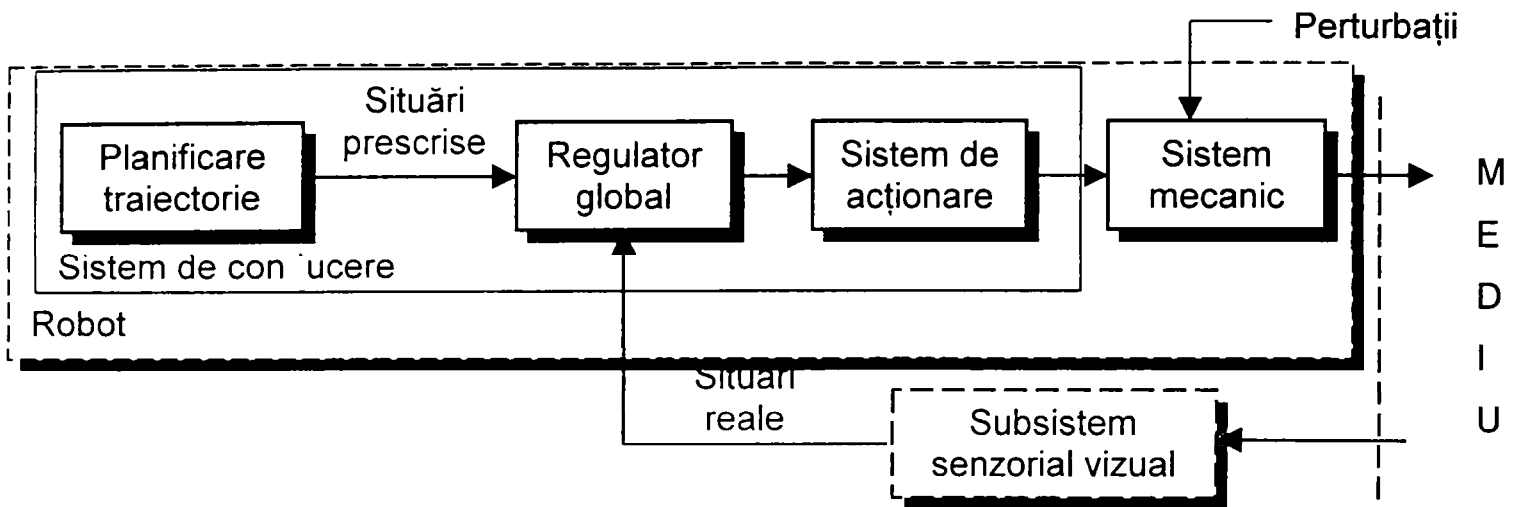


Fig.5.1. Schema bloc a unui sistem robot care prezintă conducere senzorială care utilizează informații furnizate de senzori vizuali

În continuare se vor utiliza următoarele denumiri:

- *subsistemul senzorial vizual* pentru componenta sistemului care achiziționează informațiile despre situațiile reale ale efectorului final prin intermediul senzorilor vizuali;
- *robot* pentru robotul propriu-zis care intră în componența ansamblului „sistem de conducere senzorială a robotului”
- *sistem de conducere senzorială a robotului (sistem robot)* pentru ansamblul format din cele două componente subsistem senzorial vizual și robot a cărei structură este prezentată în figura 5.2.

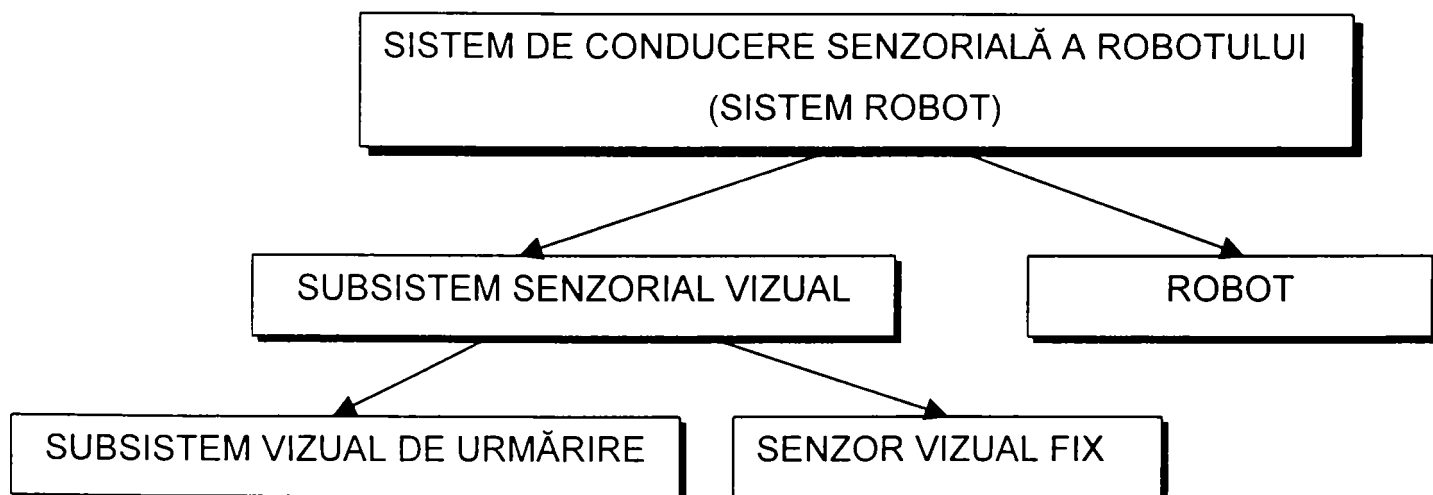


Fig.5.2. Structura sistemului de conducere senzorială a robotului

S-au luat în considerare două cazuri:

- situarea ideală corespunde situației programate, iar situarea reală (realizată) sesizată de subsistemul senzorial vizual este utilizată de către regulatorul global pentru obținerea corecțiilor de situație a efectorului final: în acest caz subsistemul senzorial vizual se materializează printr-un subsistem vizual de urmărire continuă a efectorului final;
- situarea reală este situația scop (spre exemplu situația în care trebuie să ajungă efectorul final pentru a prehensa obiectul de lucru OL), situația OL sesizată de subsistemul senzorial vizual, iar situația efectorului final la momentul inițial este situația ideală – în acest caz subsistemul senzorial vizual se materializează printr-un senzor vizual fix, care va urmări scena din mediu în mod secvențial. discret.

Elementele componente ale sistemului robot cu conducere senzorială sunt: robotul propriu-zis și subsistemul senzorial vizual. așa cum rezultă și din figura 5.2.

S-au conceput două sisteme de conducere. câte unul pentru fiecare caz:

a) *Sistem robot la care comanda de corecție a situației efectorului final se realizează prin utilizarea informațiilor furnizate de un subsistem vizual de urmărire continuă*

În acest caz. din programul de lucru al robotului se determină *situația programată* – situație care convențional se va numi *situație ideală* - a efectorului final și cu ajutorul generatorului de traiectorie, parametrii de situație relativă generalizați la nivelul fiecărei cuple cinematice conducătoare.

Comparându-se această situație programată cu *situația reală* determinată cu ajutorul sistemului vizual de urmărire, rezultă o *eroare de situație*, a cărei cunoaștere prin intermediul unei analize inverse va permite determinarea *comenzilor de corecție* necesare pentru situațiile relative ale elementelor fiecărei cuple cinematice conducătoare ale robotului.

Semnalul rezultat din suprapunerea peste semnalul de intrare de comandă a corecțiilor necesare a fi făcute pentru eliminarea erorilor, se transmite sistemului de acționare în vederea punerii în mișcare relativă a elementelor componente ale sistemului mecanic.

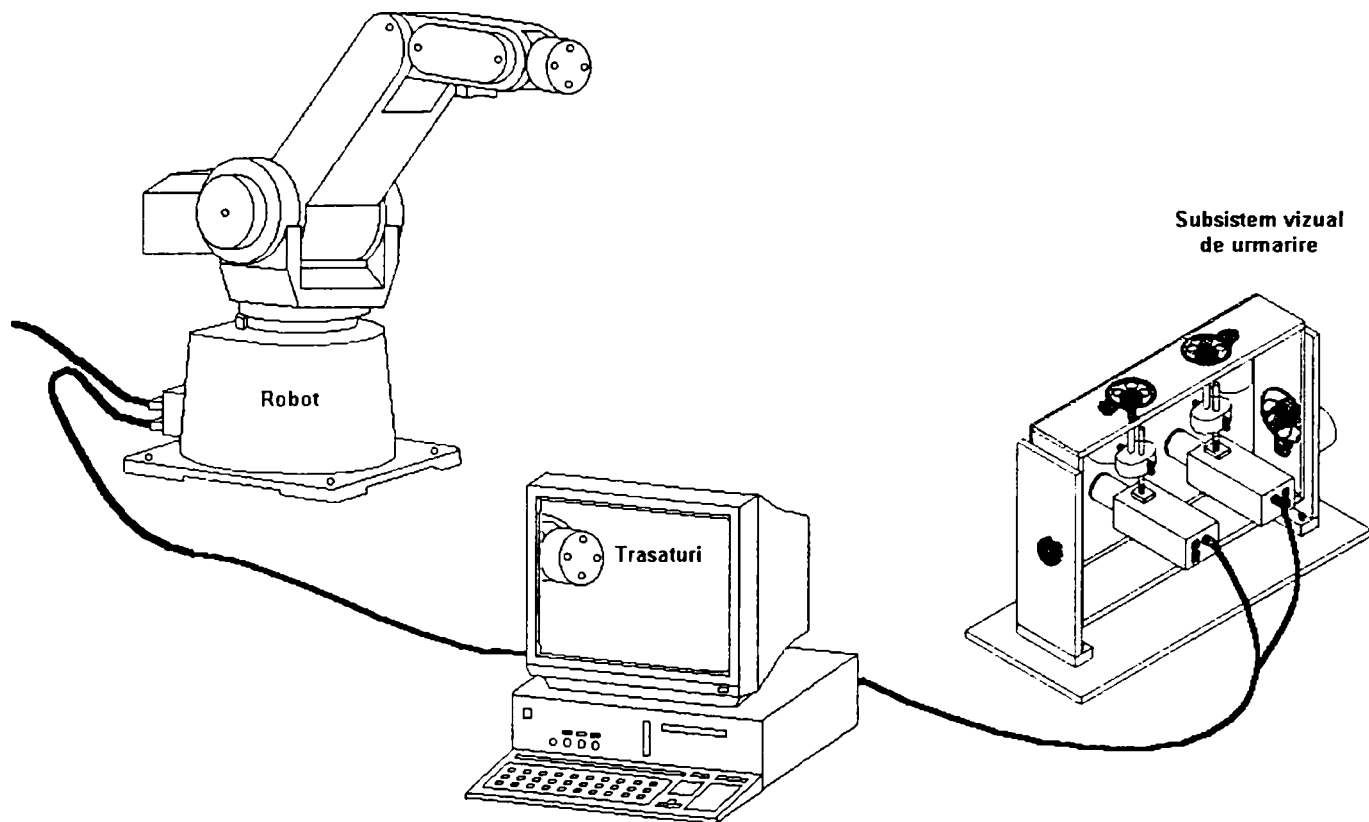


Fig.5.3. Sistem robot la care comanda de corecție a situației efectorului final se realizează prin utilizarea informațiilor provenite de la subsistemul vizual de urmărire

Avantajul unei astfel de abordări îl constituie asigurarea urmăririi obiectivului conducerii robotului prin controlul și corecția în timp real a situației efectorului final.

În figura 5.3 se prezintă sistemul robot, la care comanda de corecție a situației efectorului final se realizează prin utilizarea subsistemului vizual de urmărire continuă a situației efectorului final și care este compus din:

- ☞ subsistemul vizual de urmărire, cu rol de a determina în timp real situația reală a obiectului de manipulat;

- ☞ robotul, cu rol de a compara situația comandată cu situația realizată, de a determina matricea erorii de situație, cu ajutorul căreia, printr-o analiză inversă să se determine corecțiile necesare a fi aplicate peste semnalul de intrare la nivelul fiecărei cuple cinematice conducătoare.

b) Sistem robot la care comanda de corecție a poziției efectorului final se realizează prin utilizarea unui senzor vizual fix, care urmărește scena din mediu în mod secvențial, discret

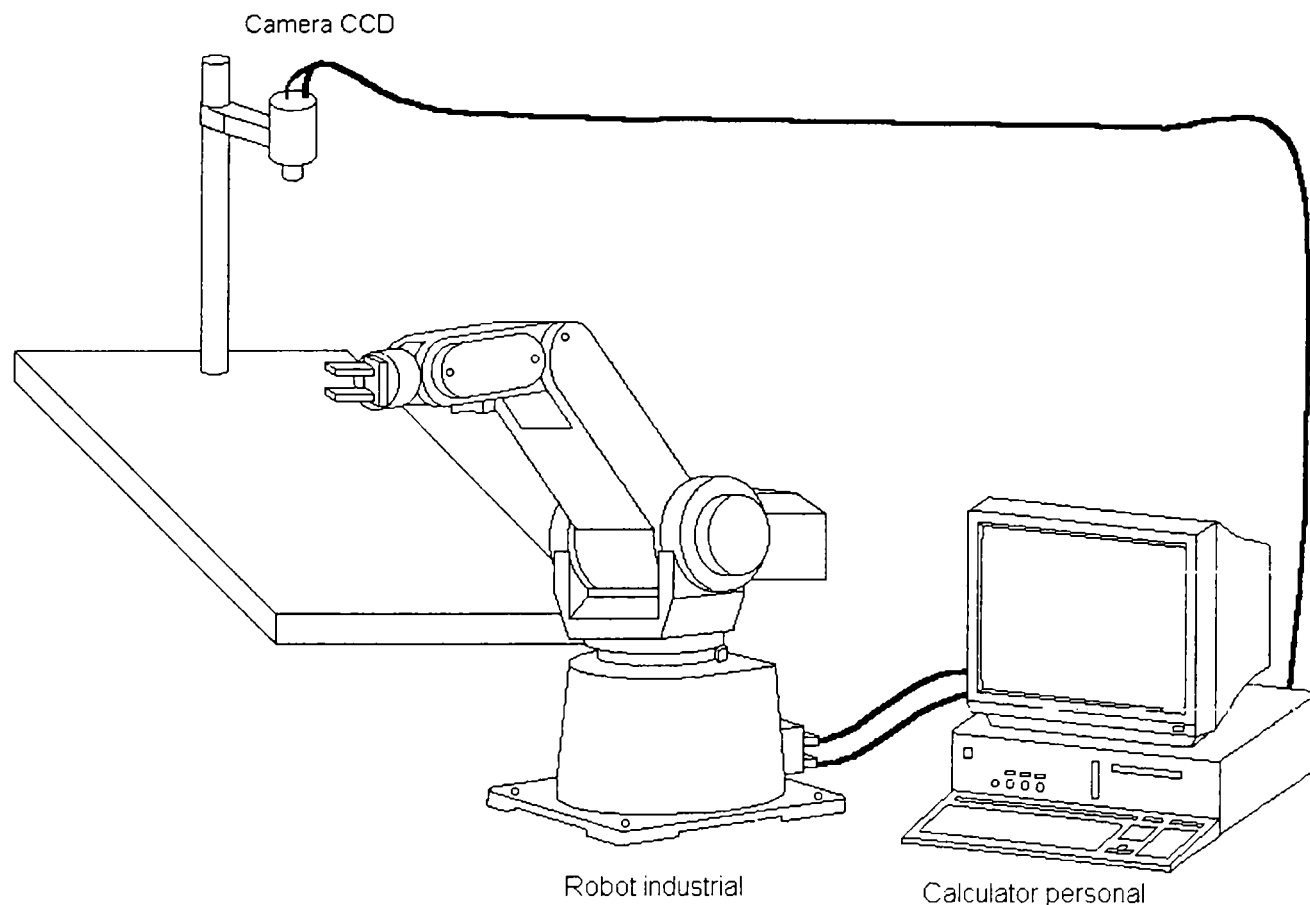


Fig.5.4. Sistem robot la care comanda de corecție a poziției efectorului final se realizează prin utilizarea unui senzor vizual fix

În acest caz, *situarea reală* a efectorului final este situarea în care trebuie să ajungă efectorul final pentru a prehensa OL, determinarea ei realizându-se prin intermediul senzorului vizual fix. *Situarea efectorului final* la momentul inițial se consideră a fi *situarea ideală*.

Structura unui sistem robot la care comanda de corecție a poziției efectorului final se realizează prin utilizarea unui senzor vizual fix se prezintă în figura 5.4 și are următoarele elemente componente:

- ☞ senzorul vizual fix, cu rol de a determina la anumite momente *situarea reală* a obiectului de lucru – *situare* care în mod convențional va fi numită în continuare *situare reală*;

☞ robotul, cu rolul de a manipula OL; sistemul de conducere a robotului folosind informația prelevată de la senzorul vizual fix cu privire la situarea reală a OL și informația cu privire la situarea la acel moment al efectorului final – situare pe care convențional va fi numită în continuare situare ideală – va determina eroarea de situare, după care printr-o analiză cinematico-pozițională inversă se va stabili comanda de corecție necesară pentru situațiile relative ale elementelor cuplelor cinematice conducătoare ale robotului, astfel încât robotul să realizeze corecta prehensare a OL.

În cadrul capitolului sunt tratate aspectele legate de:

☞ stabilirea modelelor geometrice pentru componentele celor două sisteme de conducere senzorială a robotului prezentate mai sus și anume: subsistem vizual senzorial, care se materializează în primul caz printr-un subsistemul vizual de urmărire, iar în cel de-al doilea caz printr-un senzor vizual fix și respectiv componenta robot;

☞ stabilirea algoritmilor de determinare a situației reale a efectorului final;

☞ stabilirea algoritmilor de determinare a situației ideale a efectorului final;

☞ stabilirea algoritmilor de determinare a erorii de situare a efectorului final și

☞ deducerea corecțiilor necesare pentru situațiile relative ale elementelor cuplelor cinematice conducătoare ale robotului.

5.2.Principiile metodei propuse de conducere senzorială a robotului pe baza informațiilor din mediu, achiziționate de senzori vizuali

În vederea modelării matematice și a analizei funcționării celor două subsisteme ale sistemului robot se consideră un sistem de referință universal $O_0x_0y_0z_0$ legat de baza robotului, la care se vor raporta toate mărimile determinate.

Se mai definesc două sisteme de referință, astfel: subsistemul vizual senzorial va avea atașat un sistem de referință $O_vx_vy_vz_v$ cu originea O_v în centrul optic al senzorului, cu o situare oarecare față de sistemul de referință universal; efectorului final al robotului corespunzător situațiilor reale (definite anterior) i se atașează un sistem de referință $O_{Re}x_{Re}y_{Re}z_{Re}$.

Etapele de parcurs pentru realizarea conducerii senzoriale a robotului utilizând informațiile din mediu, achiziționate de senzori vizuali sunt următoarele:

- se stabilește *situarea reală* efectivă sau dorită (după caz) a efectorului final (sistemul de referință $O_{Re}x_{Re}y_{Re}z_{Re}$) în raport cu sistemul de referință atașat

senzorului vizual $O_V x_V y_V z_V$ ${}^{O_V} T_{O_{Re}}$, utilizând informațiile furnizate de subsistemul vizual senzorial;

- se stabilește *situarea reală* efectivă sau dorită (după caz) a efectorului final (sistemul de referință $O_{Re} x_{Re} y_{Re} z_{Re}$) în raport cu sistemul de referință universal $O_0 x_0 y_0 z_0$, conform figurii 5.5:

$${}^{O_0} T_{O_{Re}} = {}^{O_0} T_{O_V} \cdot {}^{O_V} T_{O_{Re}} \quad (5.1)$$

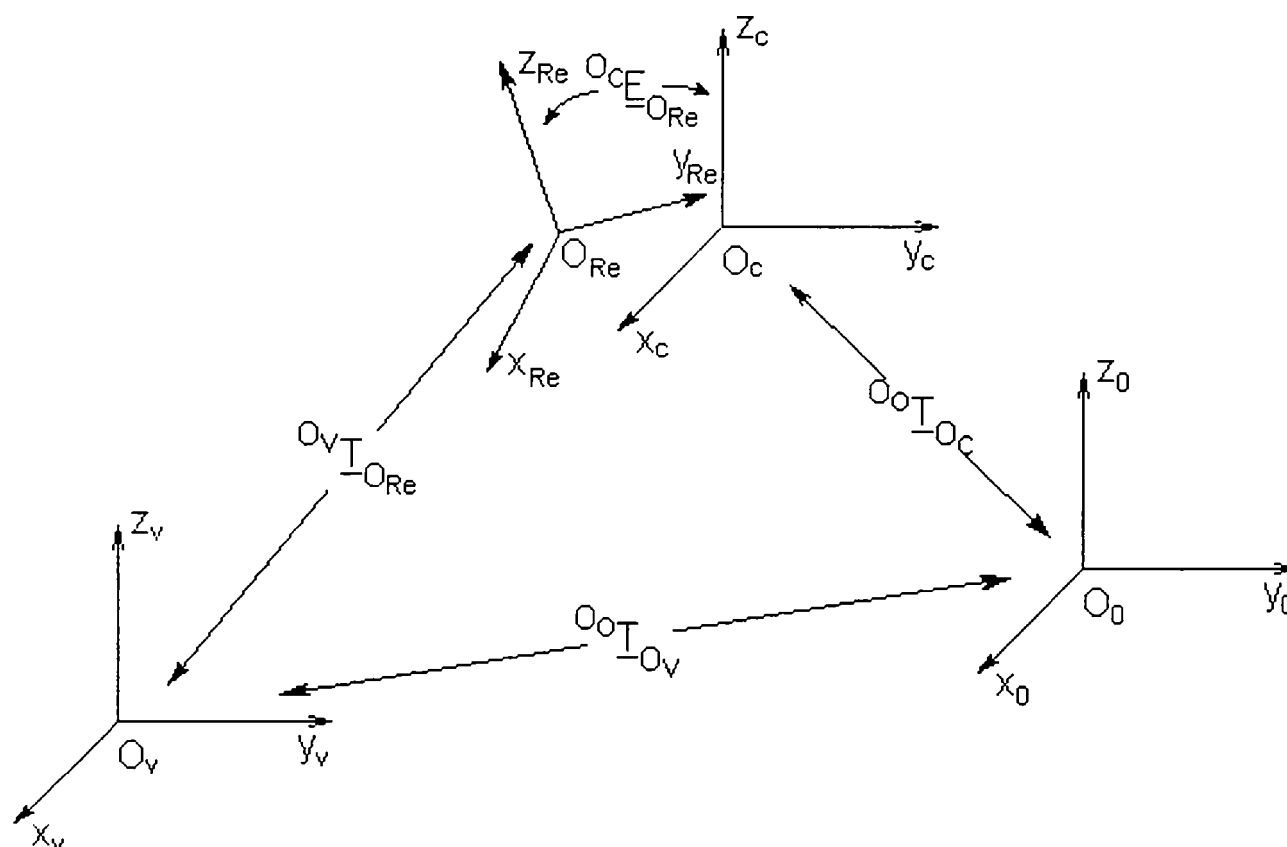


Fig. 5.5. Schemă pentru modul de calcul al matricei erorii de situare

- se stabilește situarea ideală a efectorului final (sistem $O_C x_C y_C z_C$) în raport cu sistemul de referință universal $O_0 x_0 y_0 z_0$: ${}^{O_0} T_{O_C}$
- se stabilește „eroarea situației”, care se definește ca fiind matricea de trecere de la sistemul de referință $O_C x_C y_C z_C$ la sistemul de referință $O_{Re} x_{Re} y_{Re} z_{Re}$:

$${}^{O_C} E_{O_{Re}} = {}^{O_0} T_{O_C}^{-1} \cdot {}^{O_0} T_{O_{Re}} = {}^{O_0} T_{O_C}^{-1} \cdot {}^{O_0} T_{O_V} \cdot {}^{O_V} T_{O_{Re}} \quad (5.2)$$

- se calculează comenzile de corecție necesare pentru situațiile relative ale elementelor cuplelor cinematice conducătoare ale dispozitivului de ghidare al robotului.

În figura 5.6 se prezintă schema logică a metodei propuse de conducere senzorială a robotului pe baza informațiilor din mediu, achiziționate de senzori vizuali.

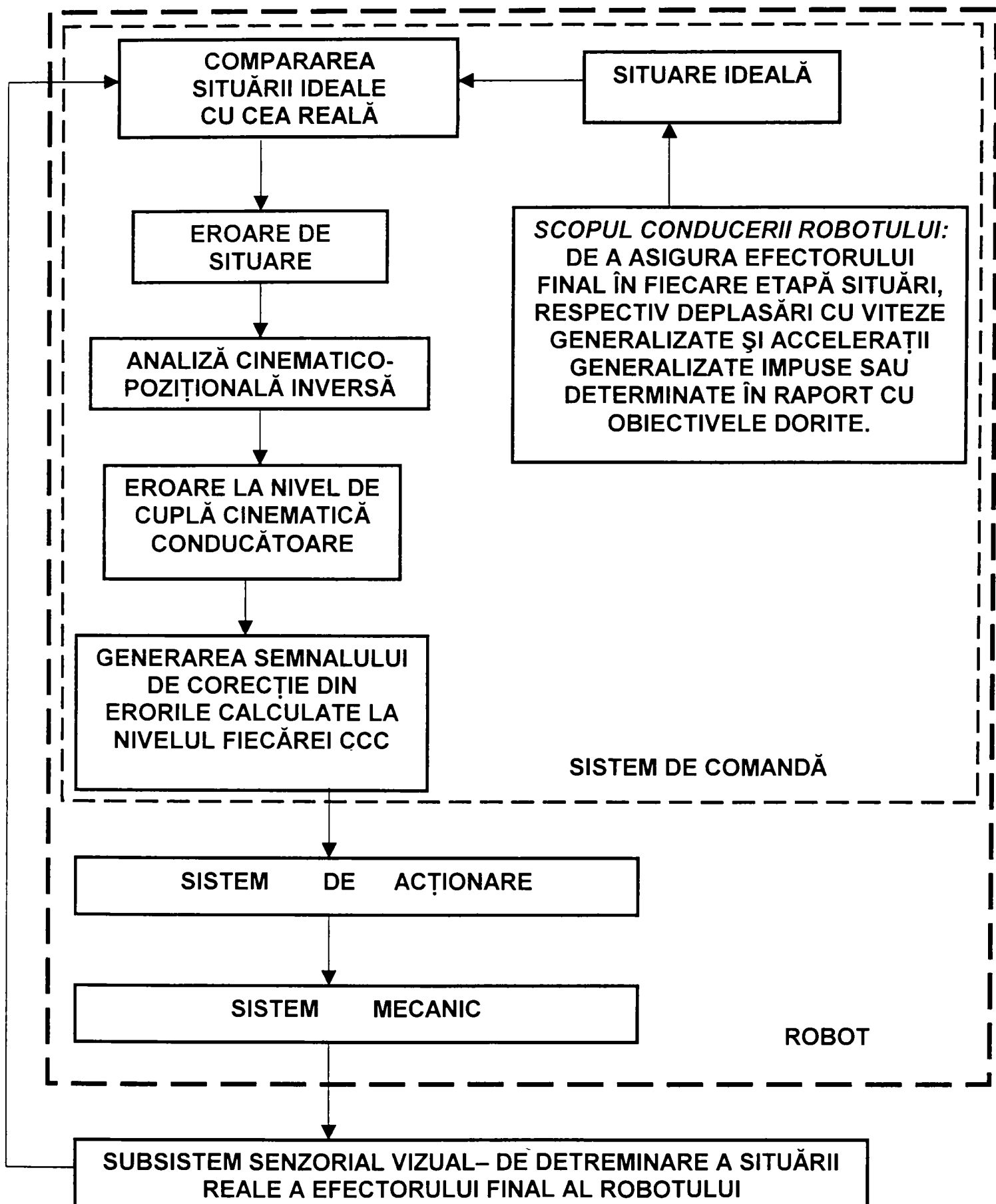


Fig.5.6. Modul de realizare a comenzii la sistemele care utilizează informații

furnizate de un subsistem senzorial vizual

Pornindu-se de la aceste considerente se necesită a se trata aspectele legate de structura, modelarea matematică și analiza funcționării celor două componente necesare realizării conducerii senzoriale a sistemului robot:

☞ robotul, care are rolul de a compara situația ideală cu situația reală, de a determina matricea erorii de situație, pe baza căreia să se determine corecțiile necesare pentru situațiile relative ale elementelor cuplelor cinematice conducătoare ale robotului;

☞ subsistem vizual senzorial, care are rolul de a determina *situația reală* a efectorului final.

5.3. Achiziționarea informațiilor privind situația reală a efectorului final de către subsistemul vizual de urmărire

5.3.1. Noțiuni introductive

Pentru realizarea subsistemului vizual de urmărire, punctul de plecare l-a constituit modul de funcționare a sistemului de vedere al unui operator uman.

Astfel, pentru determinarea situației unui obiect în spațiu sunt necesare doi senzori vizuali, senzori care priviți prin prisma antropomorfismului structural pot să înlocuiască fiecare câte un ochi uman.

Fiecare senzor vizual trebuie să posede cel puțin două grade de mobilitate și anume o rotație în jurul direcției verticale, respectiv o rotație în jurul direcției orizontale (conform figurii 5.7).

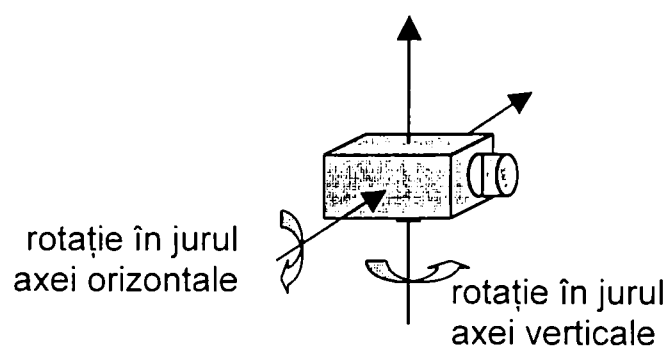


Fig.5.7. Unghiurile de rotație a senzorului vizual

Cele două grade de mobilitate trebuie să fie conferite unui senzor vizual de un mecanism prevăzut cu două cuple cinematice de rotație conducătoare de clasa a V-a, (pentru respectarea condiției desmodromiei mecanismului).

Făcând o analogie cu ochii umani și ținând cont de faptul că mișcarea de rotație a ochilor umani în jurul axei orizontale se realizează simultan, mișcarea de rotație în jurul axei orizontale poate fi o mișcare comună pentru ambii senzori vizuali – generată de un singur motor de acționare.

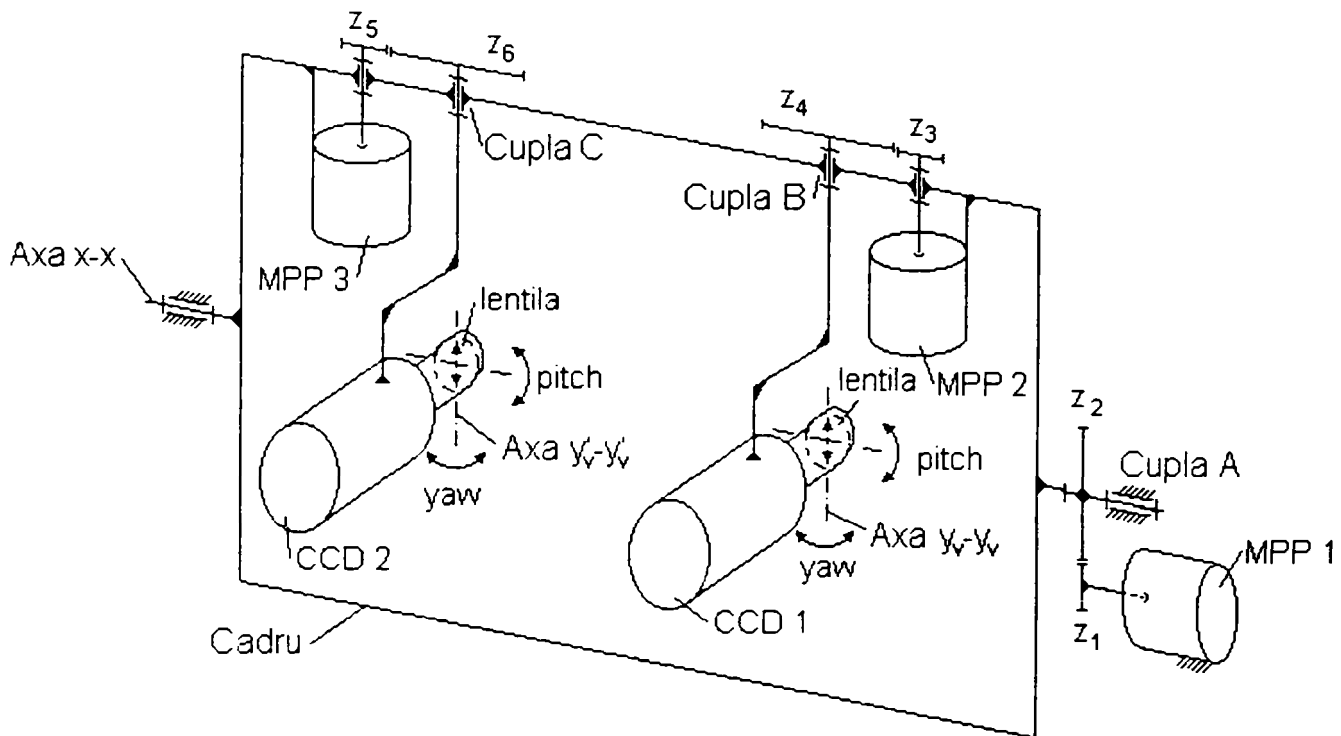


Fig.5.8. Schema cinematică a mecanismului subsistemului vizual de urmărire

Astfel, pentru realizarea unei vederi stereoscopice se necesită un mecanism care să prezinte trei cuple cinematice conducătoare acționate de trei motoare de acționare și anume:

- ☞ un motor (MPP1) pentru mișcarea simultană a cei doi senzori vizuali (CCD 1, CCD2) în jurul axei orizontale – mișcarea "pitch" (flexie-extensie) și
- ☞ câte un motor (MPP2, MPP3) pentru fiecare senzor vizual care să genereze mișcarea de rotație „yaw” (aducție-abducție) în jurul axei verticale, pentru fiecare senzor (CCD 1, CCD2).

Subsistemul se compune din următoarele elemente componente:

- ☞ doi senzori vizuali CCD1, CCD2;

- ☞ cele trei motoare de acționare MPP1, MPP2, MPP3;
- ☞ trei transmisii cu roți dințate Z_1-Z_2 , Z_3-Z_4 , Z_5-Z_6 , cu rol de demultiplicare a mișcării:
- ☞ cadrul C, cu rol de a oferi posibilitatea mișcării de flexie-extensie comună celor doi senzori vizuali:
- ☞ suportul S, pe care este amplasat întreg ansamblul.

Senzorii vizuali, motoarele de acționare și reductoarele au fost astfel amplasate, încât axa de rotație orizontală $x-x$ și respectiv axele de rotație verticale y_1-y_1 și y_2-y_2 aferente fiecărui senzor vizual să fie coplanare, iar punctul în jurul căruia se va roti fiecare senzor vizual să se găsească în centrul optic al lentilei fiecărui senzor vizual.

5.3.2. Modelarea matematică a subsistemului vizual de urmărire

5.3.2.1. Scopul modelării

Din punct de vedere al realizării conducerii globale a robotului, scopul subsistemului vizual de urmărire este acela de a urmări în timp real ținta – materializată în cazul de față de un punct al efectorului final al robotului – în vederea determinării situației acesteia.

Pentru o abordare mai facilă, se vor determina într-o primă fază, relațiile prin care se poate determina poziția unui punct al țintei (punct caracteristic), apoi se va prezenta modalitatea de determinare și a orientării acesteia.

Ținta de urmărit de către cei doi senzori vizuali va fi reprezentată de o diodă electroluminescentă, în cazul în care se urmărește determinarea poziției efectorului final.

Situarea ("poziția-orientarea") unui corp în spațiul tridimensional, poate fi definită cu ajutorul poziției punctului caracteristic și a orientărilor dreptei caracteristice, respectiv a dreptei auxiliare. Ca urmare, pentru determinarea situației unui obiect este suficient cunoașterea coordonatelor a trei puncte aferente obiectului și anume punctul caracteristic și încă două puncte unul aferent dreptei caracteristice, iar celălalt aferent dreptei auxiliare, așezate la distanțe constante, cunoscute de punctul caracteristic.

Prin urmare, atunci când se urmărește determinarea situației efectorului final ținta de urmărit se va materializa prin trei diode electroluminescente.

5.3.2.2. Modelul geometric al mecanismului subsistemului vizual de urmărire

Sistemul vizual de urmărire este în fapt la rândul său un sistem robot, care determină poziția unui obiect de manipulat în spațiu.

Pornind de la această observație, subsistemul vizual de urmărire cu cele trei grade de mobilitate poate fi privit ca un mecanism generator de traiectorie al țintei de urmărit, iar unghiurile α , β , θ vor fi parametrii de situație relativă generalizați ai cuplelor cinematice conducătoare (c.c.c.) A, B, respectiv C ale mecanismului subsistemului vizual de urmărire.

Metodologia de determinare a poziției țintei – punctul G – este în fapt chiar analiza geometrică directă pentru subsistemul vizual de urmărire.

Direcția de urmărit (vizare), pentru fiecare senzor vizual este reprezentată de dreapta care unește ținta cu centrul lentilei senzorului.

Fie $G(x_G, y_G, z_G)$ poziția în care se află la un moment dat ținta, O_v , respectiv O'_v poziția centrelor optice ale lentilelor celor două CCD-uri și $O_v x_v y_v z_v$ sistemul de coordonate triortogonal drept, față de care se determină poziția țintei G. Sistemul de referință $O_v x_v y_v z_v$ se definește ca având centrul în centrul optic al lentilei CCD 1 și axele paralele cu sistemul de referință universal.

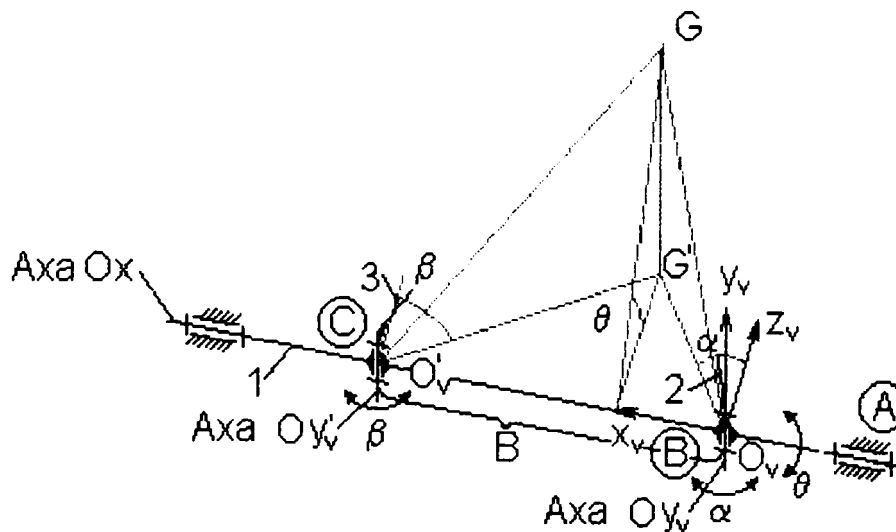


Fig.5.9. Schema cinematică simplificată a subsistemului vizual de urmărire

Pornindu-se de la schema cinematică simplificată a subsistemului vizual de urmărire prezentată în figura 5.9 și cunoscându-se parametrii de situație relativă generalizați (α , β , θ) pentru care imaginea punctului țintă G se formează în centrele

planelor imagine ale celor doi senzori vizuali, prin analiza geometrică directă se determină coordonatele punctului țintă.

În ceea ce urmează se exprimă ecuațiile planelor $GO_vO'_v$, $GG'O_v$ și respectiv $G'O_vO'_v$ în raport cu sistemul de referință atașat subsistemului vizual de urmărire $O_vx_vy_vz_v$.

Ecuția planului $GO_vO'_v$ este:

$$y - z \cdot \operatorname{tg}\theta = 0 \quad (5.3)$$

Ecuția planului $GG'O_v$ este:

$$x - z \cdot \operatorname{tg}\alpha = 0 \quad (5.4)$$

Ecuția planului $G'O_vO'_v$ este:

$$x - z \cdot \operatorname{tg}\beta = B \quad (5.5)$$

unde:

B – distanța între axele cuplelor cinematice conducătoare B și C (este constantă și rezultă din realizarea efectivă a subsistemului vizual de urmărire);

α - unghiul dintre O_vz_v și proiecția direcției de vizare a sensorului vizual 1 pe planul $x_vO_vz_v$ – în fapt parametrul de situare relativă generalizată cu care trebuie rotit senzorul CCD 1 față de axa O_vz_v după axa verticală, pentru a obține imaginea țintei pe o axă verticală dispusă pe mijlocul planului imagine a sensorului CCD 1;

β - unghiul dintre O_vz_v și proiecția direcției de vizare a sensorului vizual 2 pe planul $x_vO_vz_v$ – în fapt unghiul cu care trebuie rotit senzorul CCD 2 față de axa O_vz_v în jurul axei verticale, pentru a obține imaginea țintei pe o axă verticală dispusă pe mijlocul planului imagine a sensorului CCD 2;

θ - unghiul diedru dintre planul $x_vO_vz_v$ și planul format de direcțiile de vizare ale celor doi senzori vizuali – în fapt unghiul cu care trebuie rotit cadrul C și prin urmare ambii senzori vizuali în jurul axei O_vx_v pentru obține imaginile țintei pe o axă verticală dispusă pe mijlocul planelor imagine ai celor doi senzori CCD.

Coordonatele $({}^o x_G \quad {}^o y_G \quad {}^o z_G)$ ale punctului G în sistemul de coordonate $O_vx_vy_vz_v$ se determină ținând cont de faptul că punctul țintă se află la intersecția planelor $GO_vO'_v$, $GG'O_v$ și respectiv $G'O_vO'_v$:

$$\begin{cases} y - z \cdot \operatorname{tg} \theta = 0 \\ x & - z \cdot \operatorname{tg} \alpha = 0 \\ x & - z \cdot \operatorname{tg} \beta = B \end{cases} \quad (5.6)$$

Soluționând sistemul de ecuații (5.6) se obțin coordonatele țintei:

$$\begin{aligned} {}^0 x_G &= B \cdot \frac{\operatorname{tg} \alpha}{\operatorname{tg} \alpha - \operatorname{tg} \beta} \\ {}^0 y_G &= B \cdot \frac{\operatorname{tg} \theta}{\operatorname{tg} \alpha - \operatorname{tg} \beta} \\ {}^0 z_G &= B \cdot \frac{1}{\operatorname{tg} \alpha - \operatorname{tg} \beta} \end{aligned} \quad (5.7)$$

În vederea stabilirii modelului geometric invers se pornește de la coordonatele țintei $({}^0 x_G, {}^0 y_G, {}^0 z_G)$ și se determină parametri de situare relativă generalizați α, β, θ .

Din împărțirea primei și ultimei egalități din relațiile (5.7) se obține:

$$\alpha = \operatorname{arctg} \left(\frac{{}^0 x_G}{{}^0 z_G} \right) \quad (5.8)$$

iar prin împărțirea ultimelor două egalități din relațiile (5.7) rezultă:

$$\theta = \operatorname{arctg} \left(\frac{{}^0 y_G}{{}^0 z_G} \right) \quad (5.9)$$

Cel de-al treilea parametru de situare relativă generalizată rezultă indirect ca fiind:

$$\beta = \operatorname{arctg} \left(\frac{{}^0 x_G - B}{{}^0 z_G} \right) \quad (5.10)$$

5.3.3. Algoritm pentru determinarea poziției țintei

Se consideră $I_1 x_1 y_1 z_1$, respectiv $I_2 x_2 y_2 z_2$ două sisteme de referință atașate planelor imagine aferente fiecărei camere CCD, astfel încât axele $I_i z_i$, $i=1,2$ să fie orientate după axa optică a senzorului vizual corespunzător, axele $I_i x_i$, $i=1,2$ să prezinte întotdeauna o orientare orizontală, iar axele $I_i y_i$, $i=1,2$ se aleg astfel încât să formeze cu primele două un sistem de referință triortogonal drept.

În figura 5.10 sunt prezentate aceste sisteme de coordonate.

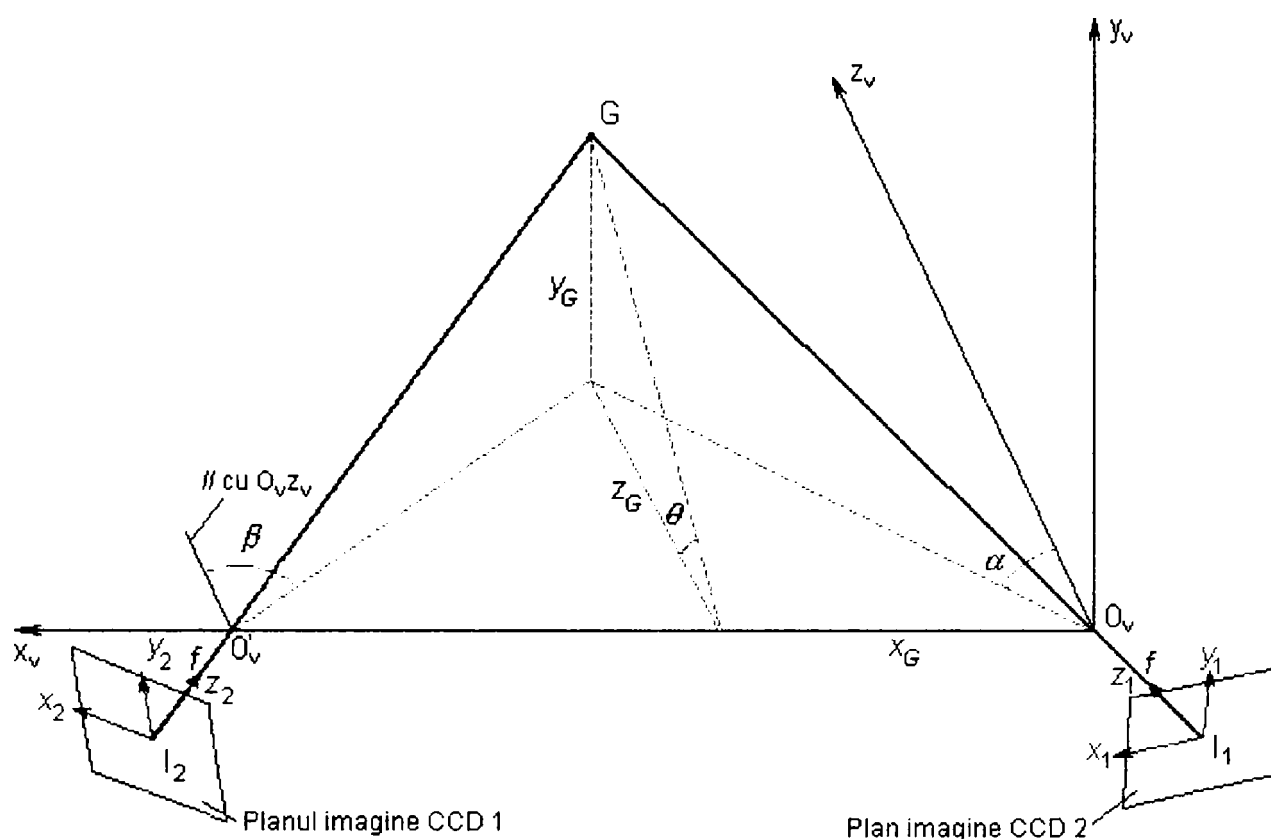


Fig.5.10. Schemă pentru calculul poziției țintei G

Pentru determinarea poziției țintei de urmărit se rotește cadrul C după c.c.c. A cu ajutorul motorului MPP 1 și cei doi senzori vizuali CCD 1 și CCD 2 după c.c.c. B și respectiv C până când imaginile țintei G se vor poziționa în centrele I_1 , I_2 ale celor două plane imagine aferente celor doi senzori vizuali

Coordonatele țintei G în sistemul de coordonate $O_v x_v y_v z_v$ sunt date de relațiile (5.7) deduse în paragraful anterior.

Pentru că efectorul final se află într-o continuă mișcare, în general axa optică a CCD-ului nu se va suprapune peste direcția de urmărire, situație arătată și în figura 5.11.

Modul de determinare a poziției țintei în funcție de parametrii de situare relativă generalizați ai c.c.c. ale mecanismului subsistemului vizual de urmărire și de coordonatele imaginii țintei este următorul: se consideră două poziții pentru ținta G și anume una la momentul inițial t_1 , când imaginea țintei G_1 se află în punctul central al planului imagine și respectiv o a doua G_2 la un moment ulterior t_2 , când imaginea țintei se află deplasată cu mărimile x_G și respectiv y_G față de punctul central al planului imagine – situație prezentată în figura 5.12.

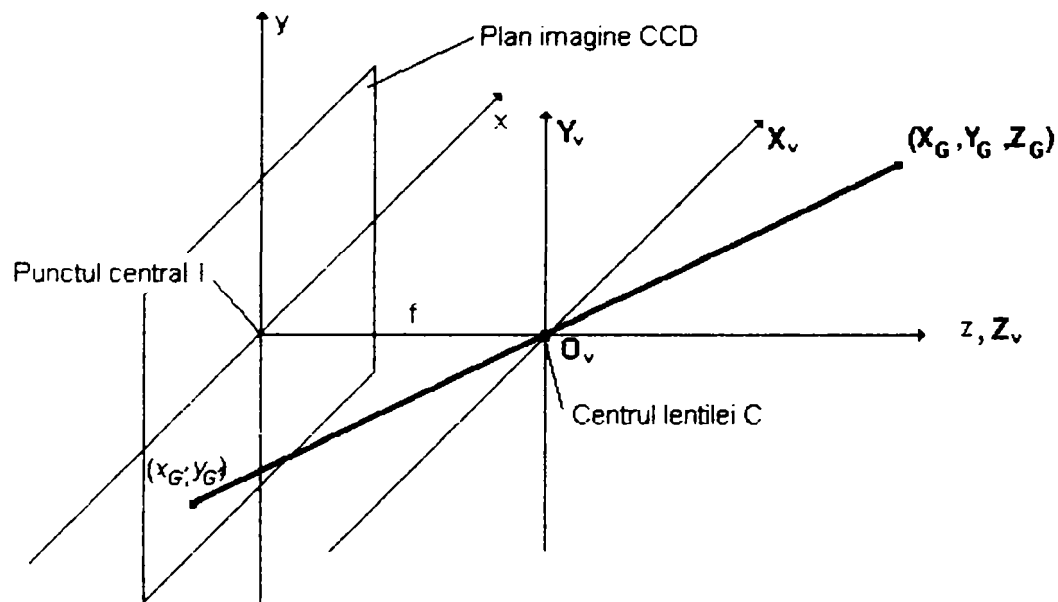


Fig.5.11. Schemă indicând modul de apariție a deplasărilor x_G' și y_G' față de centrul I al planului imagine

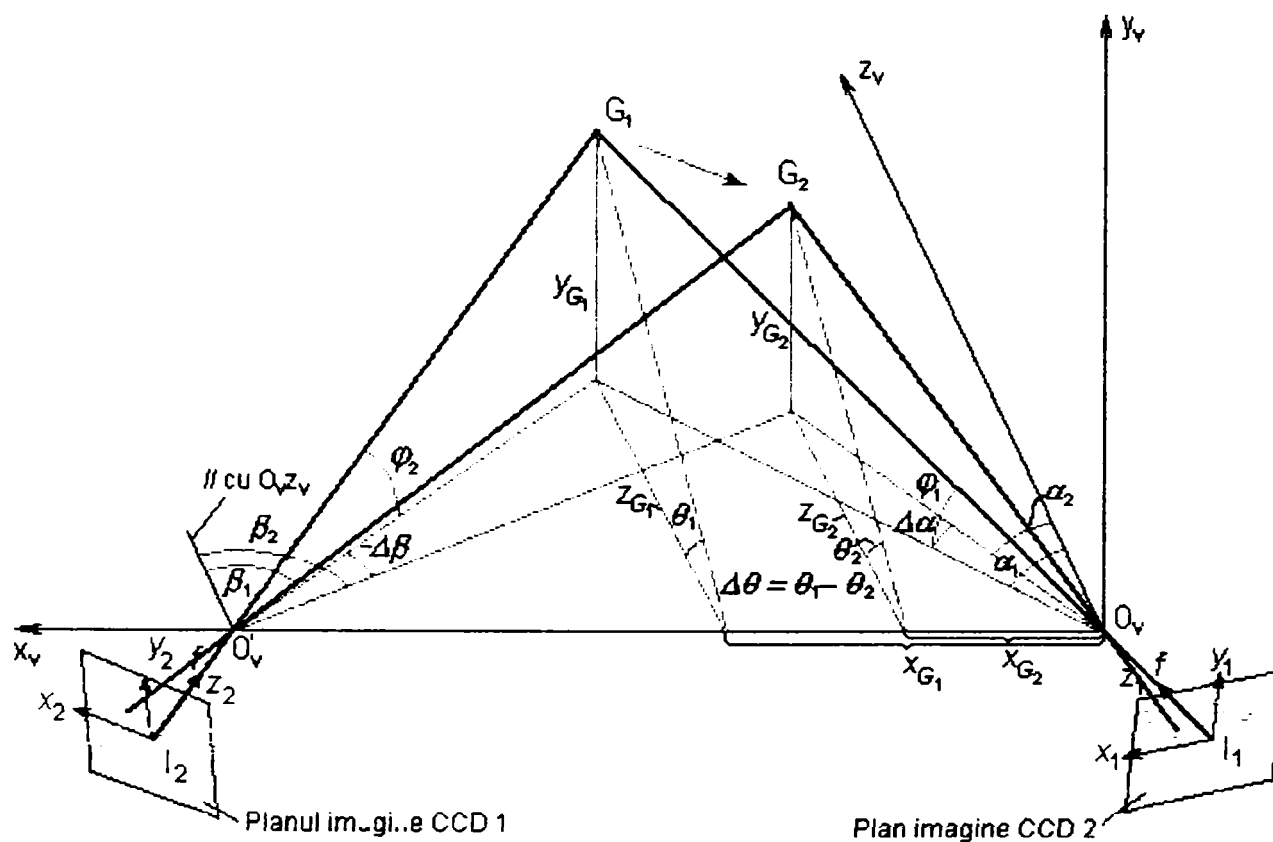


Fig.5.12. Schemă pentru determinarea noii poziții a țintei (G_2) funcție de poziția anterioară (G_1) și deplasările x_{G2} , y_{G2} și x_{G2} , y_{G2} sesizate de senzorii CCD

Din figură se observă că pentru a aduce imaginea punctului G_2 în punctele centrale ale planelor imagine ale celor doi senzori vizuali este necesar să se rotească cei doi senzori CCD în jurul centrelor lentilelor lor O_v și O'_v cu unghiurile $\Delta\alpha$ și $\Delta\beta$ în jurul axelor verticale $O_v y_v$ și $O'_v y_v$, funcție de CCD și respectiv $\Delta\theta$ în jurul axei orizontale $O_v x_v$ conform figurii 5.12.

Coordonatele țintei la momentul t_2 sunt coordonatele punctului de intersecție ale dreptelor $G'_2 O_v$, respectiv $G''_2 O'_v$.

Ecuțiile celor două drepte sunt date de:

$$\frac{{}^{O_v}x - {}^{O_v}x_{C_1}}{{}^{O_v}x_{G'_2} - {}^{O_v}x_{C_1}} = \frac{{}^{O_v}y - {}^{O_v}y_{C_1}}{{}^{O_v}y_{G'_2} - {}^{O_v}y_{C_1}} = \frac{{}^{O_v}z - {}^{O_v}z_{C_1}}{{}^{O_v}z_{G'_2} - {}^{O_v}z_{C_1}} \quad (5.11)$$

pentru dreapta $G'_2 O_v$,

$$\frac{{}^{O_v}x - {}^{O_v}x_{C_2}}{{}^{O_v}x_{G''_2} - {}^{O_v}x_{C_2}} = \frac{{}^{O_v}y - {}^{O_v}y_{C_2}}{{}^{O_v}y_{G''_2} - {}^{O_v}y_{C_2}} = \frac{{}^{O_v}z - {}^{O_v}z_{C_2}}{{}^{O_v}z_{G''_2} - {}^{O_v}z_{C_2}} \quad (5.12)$$

pentru dreapta $G''_2 O'_v$.

Coordonatele punctelor O_v , O'_v în sistemul de coordonate $O_v x_v y_v z_v$ sunt:

$$\begin{cases} {}^{O_v}x_{C_1} = 0 \\ {}^{O_v}y_{C_1} = 0 \\ {}^{O_v}z_{C_1} = 0 \end{cases} \quad (5.13)$$

$$\begin{cases} {}^{O_v}x_{C_2} = B \\ {}^{O_v}y_{C_2} = 0 \\ {}^{O_v}z_{C_2} = 0 \end{cases} \quad (5.14)$$

Relațiile (5.11), (5.12) devin:

$$\frac{{}^{O_v}x}{{}^{O_v}x_{G'_2}} = \frac{{}^{O_v}y}{{}^{O_v}y_{G'_2}} = \frac{{}^{O_v}z}{{}^{O_v}z_{G'_2}} \quad (5.15)$$

$$\frac{{}^{O_v}x - B}{{}^{O_v}x_{G''_2} - B} = \frac{{}^{O_v}y}{{}^{O_v}y_{G''_2}} = \frac{{}^{O_v}z}{{}^{O_v}z_{G''_2}} \quad (5.16)$$

Pentru determinarea coordonatelor punctelor imagine ale țintei în sistemul de coordonate $O_v x_v y_v z_v$ se necesită determinarea matricelor de trecere de la sistemele de referință atașate planelor imagine aferente fiecărui senzor CCD $I_1 x_1 y_1 z_1$, respectiv $I_2 x_2 y_2 z_2$ la sistemul de referință $O_v x_v y_v z_v$.

Trecerea de la sistemul de referință $I_1 x_1 y_1 z_1$ atașat planului imagine la sistemul de referință $O_v x_v y_v z_v$ atașat subsistemului vizual de urmărire se face aplicând următorul set

de transformări: translație în lungul axei z cu $-f$; rotație în jurul axei x cu unghiul $-\varphi_1$; rotație în jurul axei y cu unghiul α_1 .

Matricea de trecere de la sistemul de referință $I_1x_1y_1z_1$ la sistemul de referință $O_vx_vy_vz_v$ va fi dată de:

$${}^{O_v}\underline{T}_{I_1} = Rot_y(\alpha_1) \cdot Rot_x(-\varphi_1) \cdot Trans_z(-f) \quad (5.17)$$

După efectuarea calculelor se obține:

$${}^{O_v}\underline{T}_{I_1} = \begin{pmatrix} c\alpha_1 & -s\alpha_1 \cdot s\varphi_1 & s\alpha_1 \cdot c\varphi_1 & -s\alpha_1 \cdot c\varphi_1 \cdot f \\ 0 & c\varphi_1 & s\varphi_1 & -s\varphi_1 \cdot f \\ -s\alpha_1 & -c\alpha_1 \cdot s\varphi_1 & c\alpha_1 \cdot c\varphi_1 & -c\alpha_1 \cdot c\varphi_1 \cdot f \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.18)$$

Trecerea de la sistemul de referință $I_2x_2y_2z_2$ atașat planului imagine la sistemul de referință $O_vx_vy_vz_v$ atașat subsistemului vizual de urmărire se face aplicând următorul set de transformări: translație în lungul axei z cu $-f$; rotație în jurul axei x cu unghiul $-\varphi_2$; rotație în jurul axei y cu unghiul $-\beta_1$; translație în lungul axei z cu B .

Matricea de trecere de la sistemul de referință $I_2x_2y_2z_2$ la sistemul de referință $O_vx_vy_vz_v$ va fi dată de:

$${}^{O_v}\underline{T}_{I_2} = Trans_x(B) \cdot Rot_y(-\beta_1) \cdot Rot_x(-\varphi_2) \cdot Trans_z(-f) \quad (5.19)$$

După efectuarea calculelor se obține:

$${}^{O_v}\underline{T}_{I_2} = \begin{pmatrix} c\beta_1 & s\beta_1 \cdot s\varphi_2 & -s\beta_1 \cdot c\varphi_2 & B + s\beta_1 \cdot c\varphi_2 \cdot f \\ 0 & c\varphi_2 & s\varphi_2 & -s\varphi_2 \cdot f \\ s\beta_1 & -c\beta_1 \cdot s\varphi_2 & c\beta_1 \cdot c\varphi_2 & -c\beta_1 \cdot c\varphi_2 \cdot f \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.20)$$

Prin urmare, imaginile G'_2 G''_2 ale punctului G_2 pe planele imagine ale celor doi senzori vizuali vor avea următoarele coordonate în sistemul de referință $O_vx_vy_vz_v$ atașat subsistemului vizual de urmărire:

$$\begin{pmatrix} {}^{O_v}x_{G'_2} \\ {}^{O_v}y_{G'_2} \\ {}^{O_v}z_{G'_2} \\ 1 \end{pmatrix} = {}^{O_v}\underline{T}_{I_1} \cdot \begin{pmatrix} x_{G_2} \\ y_{G_2} \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} x_{G_2} \cdot c\alpha_1 - y_{G_2} \cdot s\alpha_1 \cdot s\varphi_1 - s\alpha_1 \cdot c\varphi_1 \cdot f \\ y_{G_2} \cdot c\varphi_1 - s\varphi_1 \cdot f \\ -x_{G_2} \cdot s\alpha_1 - y_{G_2} \cdot c\alpha_1 \cdot s\varphi_1 - c\alpha_1 \cdot c\varphi_1 \cdot f \\ 1 \end{pmatrix} \quad (5.21)$$

$$\begin{pmatrix} {}^0, x_{G_2''} \\ {}^0, y_{G_2''} \\ {}^0, z_{G_2''} \\ 1 \end{pmatrix} = {}^0, \underline{T}_{l_2} \cdot \begin{pmatrix} x_{G_2''} \\ y_{G_2''} \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} x_{G_2''} \cdot c\beta_1 + y_{G_2''} \cdot s\beta_1 \cdot s\varphi_2 + B + s\beta_1 \cdot c\varphi_2 \cdot f \\ y_{G_2''} \cdot c\varphi_2 - s\varphi_2 \cdot f \\ x_{G_2''} \cdot s\beta_1 - y_{G_2''} \cdot c\beta_1 \cdot s\varphi_2 - c\beta_1 \cdot c\varphi_2 \cdot f \\ 1 \end{pmatrix} \quad (5.22)$$

unde:

$$\sin \varphi_1 = \frac{z_{G_1}}{\sqrt{x_{G_1}^2 + y_{G_1}^2 + z_{G_1}^2}} \quad \cos \varphi_1 = \frac{\sqrt{x_{G_1}^2 + y_{G_1}^2}}{\sqrt{x_{G_1}^2 + y_{G_1}^2 + z_{G_1}^2}} \quad (5.23)$$

$$\sin \varphi_2 = \frac{z_{G_1}}{\sqrt{(B - x_{G_1})^2 + y_{G_1}^2 + z_{G_1}^2}} \quad \cos \varphi_2 = \frac{\sqrt{(B - x_{G_1})^2 + y_{G_1}^2}}{\sqrt{(B - x_{G_1})^2 + y_{G_1}^2 + z_{G_1}^2}} \quad (5.24)$$

Ținându-se cont de relațiile (5.7) relațiile de mai sus devin:

$$\sin \varphi_1 = \frac{\operatorname{tg} \theta_1}{\sqrt{\operatorname{tg}^2 \alpha_1 + 1 + \operatorname{tg}^2 \theta_1}} \quad \cos \varphi_1 = \frac{1}{\cos \alpha_1 \cdot \sqrt{\operatorname{tg}^2 \alpha_1 + 1 + \operatorname{tg}^2 \theta_1}} \quad (2.25)$$

$$\sin \varphi_2 = \frac{\operatorname{tg} \theta_1}{\sqrt{\operatorname{tg}^2 \beta_1 + 1 + \operatorname{tg}^2 \theta_1}} \quad \cos \varphi_2 = \frac{1}{\cos \beta_1 \cdot \sqrt{\operatorname{tg}^2 \beta_1 + 1 + \operatorname{tg}^2 \theta_1}} \quad (5.26)$$

Pornind de la aceste relații coordonatele punctelor imagine G_2' G_2'' ale lui G_2 în sistemul de referință $O_v x_v y_v z_v$ date de relații (5.21), (5.22) devin:

$$\begin{pmatrix} {}^0, x_{G_2'} \\ {}^0, y_{G_2'} \\ {}^0, z_{G_2'} \\ 1 \end{pmatrix} = \begin{pmatrix} x_{G_2'} \cdot c\alpha_1 - y_{G_2'} \cdot \frac{s\alpha_1 \cdot \operatorname{tg} \theta_1}{\sqrt{\operatorname{tg}^2 \alpha_1 + 1 + \operatorname{tg}^2 \theta_1}} - \frac{\operatorname{tg} \alpha_1}{\sqrt{\operatorname{tg}^2 \alpha_1 + 1 + \operatorname{tg}^2 \theta_1}} \cdot f \\ y_{G_2'} \cdot \frac{1}{c\alpha_1 \cdot \sqrt{\operatorname{tg}^2 \alpha_1 + 1 + \operatorname{tg}^2 \theta_1}} - \frac{\operatorname{tg} \theta_1}{\sqrt{\operatorname{tg}^2 \alpha_1 + 1 + \operatorname{tg}^2 \theta_1}} \cdot f \\ -x_{G_2'} \cdot s\alpha_1 - y_{G_2'} \cdot \frac{c\alpha_1 \cdot \operatorname{tg} \theta_1}{\sqrt{\operatorname{tg}^2 \alpha_1 + 1 + \operatorname{tg}^2 \theta_1}} - \frac{1}{\sqrt{\operatorname{tg}^2 \alpha_1 + 1 + \operatorname{tg}^2 \theta_1}} \cdot f \\ 1 \end{pmatrix} \quad (5.27)$$

$$\begin{pmatrix} {}^0, x_{G_2''} \\ {}^0, y_{G_2''} \\ {}^0, z_{G_2''} \\ 1 \end{pmatrix} = \begin{pmatrix} x_{G_2''} \cdot c\beta_1 + y_{G_2''} \cdot \frac{s\beta_1 \cdot \operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\beta_1 + 1 + \operatorname{tg}^2\theta_1}} + B + \frac{\operatorname{tg}\beta_1}{\sqrt{\operatorname{tg}^2\beta_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f \\ y_{G_2''} \cdot \frac{1}{c\beta_1 \cdot \sqrt{\operatorname{tg}^2\beta_1 + 1 + \operatorname{tg}^2\theta_1}} - \frac{\operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\beta_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f \\ x_{G_2''} \cdot s\beta_1 - y_{G_2''} \cdot \frac{c\beta_1 \cdot \operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\beta_1 + 1 + \operatorname{tg}^2\theta_1}} - \frac{1}{\sqrt{\operatorname{tg}^2\beta_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f \\ 1 \end{pmatrix} \quad (5.28)$$

Înlocuind relațiile (5.27), (5.28) în relațiile (5.15), (5.16) se obțin ecuațiile dreptelor $G'_2 O_v$, respectiv $G''_2 O'_v$:

$$\begin{aligned} & \frac{{}^0, x}{x_{G_2'} \cdot c\alpha_1 - y_{G_2'} \cdot \frac{s\alpha_1 \cdot \operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} - \frac{\operatorname{tg}\alpha_1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f} = \\ & = \frac{{}^0, y}{y_{G_2'} \cdot \frac{1}{c\alpha_1 \cdot \sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} - \frac{\operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f} \quad (5.29) \\ & = \frac{{}^0, z}{-x_{G_2'} \cdot s\alpha_1 - y_{G_2'} \cdot \frac{c\alpha_1 \cdot \operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} - \frac{1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f} \end{aligned}$$

$$\begin{aligned} & \frac{{}^0, x - B}{x_{G_2''} \cdot c\beta_1 + y_{G_2''} \cdot \frac{s\beta_1 \cdot \operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\beta_1 + 1 + \operatorname{tg}^2\theta_1}} + \frac{\operatorname{tg}\beta_1}{\sqrt{\operatorname{tg}^2\beta_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f} = \\ & = \frac{{}^0, y}{y_{G_2''} \cdot \frac{1}{c\beta_1 \cdot \sqrt{\operatorname{tg}^2\beta_1 + 1 + \operatorname{tg}^2\theta_1}} - \frac{\operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\beta_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f} = \\ & = \frac{{}^0, z}{x_{G_2''} \cdot s\beta_1 - y_{G_2''} \cdot \frac{c\beta_1 \cdot \operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\beta_1 + 1 + \operatorname{tg}^2\theta_1}} - \frac{1}{\sqrt{\operatorname{tg}^2\beta_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f} \quad (5.30) \end{aligned}$$

Coordonatele punctului G_2 exprimate în sistemul de referință $O_v x_v y_v z_v$ rezultă din rezolvarea sistemului de ecuații (5.29), (5.30) astfel încât valorile acestora sunt:

$$\begin{aligned}
{}^{o_v}x_{G_2} &= B \cdot \frac{(x_{G_1'} \cdot c\alpha_1 \cdot \sqrt{tg^2\alpha_1 + 1 + tg^2\theta_1} - y_{G_1'} \cdot s\alpha_1 \cdot tg\theta_1 - tg\alpha_1 \cdot f) \cdot (y_{G_2'} \cdot \sqrt{tg^2\beta_1 + 1} - tg\theta_1 \cdot f)}{(x_{G_1'} \cdot c\alpha_1 \cdot \sqrt{tg^2\alpha_1 + 1 + tg^2\theta_1} - y_{G_1'} \cdot s\alpha_1 \cdot tg\theta_1 - tg\alpha_1 \cdot f) \cdot (y_{G_2'} \cdot \sqrt{tg^2\beta_1 + 1} - tg\theta_1 \cdot f) - (x_{G_2'} \cdot c\beta_1 \cdot \sqrt{tg^2\beta_1 + 1 + tg^2\theta_1} + y_{G_2'} \cdot s\beta_1 \cdot tg\theta_1 + tg\beta_1 \cdot f) \cdot (y_{G_1'} \cdot \sqrt{tg^2\alpha_1 + 1} - tg\theta_1 \cdot f)} \\
{}^{o_v}y_{G_2} &= B \cdot \frac{(x_{G_2'} \cdot c\beta_1 \cdot \sqrt{tg^2\beta_1 + 1 + tg^2\theta_1} + y_{G_2'} \cdot s\beta_1 \cdot tg\theta_1 + tg\beta_1 \cdot f) \cdot (y_{G_1'} \cdot \sqrt{tg^2\beta_1 + 1} - tg\theta_1 \cdot f)}{(x_{G_1'} \cdot c\alpha_1 \cdot \sqrt{tg^2\alpha_1 + 1 + tg^2\theta_1} - y_{G_1'} \cdot s\alpha_1 \cdot tg\theta_1 - tg\alpha_1 \cdot f) \cdot (y_{G_2'} \cdot \sqrt{tg^2\beta_1 + 1} - tg\theta_1 \cdot f) - (x_{G_2'} \cdot c\beta_1 \cdot \sqrt{tg^2\beta_1 + 1 + tg^2\theta_1} + y_{G_2'} \cdot s\beta_1 \cdot tg\theta_1 + tg\beta_1 \cdot f) \cdot (y_{G_1'} \cdot \sqrt{tg^2\alpha_1 + 1} - tg\theta_1 \cdot f)} \\
{}^{o_v}z_{G_2} &= B \cdot \frac{(-x_{G_2'} \cdot s\alpha_1 \cdot \sqrt{tg^2\alpha_1 + 1 + tg^2\theta_1} - y_{G_2'} \cdot c\alpha_1 \cdot tg\theta_1 - f) \cdot (y_{G_1'} \cdot \sqrt{tg^2\beta_1 + 1} - tg\theta_1 \cdot f)}{(x_{G_1'} \cdot c\alpha_1 \cdot \sqrt{tg^2\alpha_1 + 1 + tg^2\theta_1} - y_{G_1'} \cdot s\alpha_1 \cdot tg\theta_1 - tg\alpha_1 \cdot f) \cdot (y_{G_2'} \cdot \sqrt{tg^2\beta_1 + 1} - tg\theta_1 \cdot f) - (x_{G_2'} \cdot c\beta_1 \cdot \sqrt{tg^2\beta_1 + 1 + tg^2\theta_1} + y_{G_2'} \cdot s\beta_1 \cdot tg\theta_1 + tg\beta_1 \cdot f) \cdot (y_{G_1'} \cdot \sqrt{tg^2\alpha_1 + 1} - tg\theta_1 \cdot f)}
\end{aligned} \tag{5.31}$$

Prin urmare, dacă se cunosc parametrii de situație relativă generalizați aferenți c.c.c. ale sistemului vizual de urmărire (unghiurile α, β, θ) la un moment dat și proiecția țintei pe cele două plane imagine ale celor doi senzori vizuali la un moment ulterior, utilizând relațiile (5.31) se poate determina poziția țintei în cea de-a doua poziție.

Vectorul de poziție a țintei determinat prin utilizarea sistemului vizual de urmărire este:

$${}^{o_v}r_G = \begin{pmatrix} {}^{o_v}x_G \\ {}^{o_v}y_G \\ {}^{o_v}z_G \\ 1 \end{pmatrix} \tag{5.32}$$

5.3.4. Algoritm pentru determinarea orientării țintei

Pentru determinarea orientării efectorului final al robotului, autorul propune o metodă originală de utilizare a unui ansamblu de trei puncte țintă luminoase. Astfel, pe efectorul final al robotului, solidar cu acesta, se dispun - în vârfurile unui triunghi dreptunghic isoscel având cateta "l" - trei diode electroluminescente (LED) cu emisii în lumină roșie (R), galbenă (G) și verde (V), astfel încât sursa galbenă să se afle în vârful drept al triunghiului, conform figurii 5.13.

Cei doi senzori vizuali matriciali CCD vor urmări sursele de lumină (roșie, galbenă și respectiv verde). Criteriul de urmărire va fi acela de a minimiza distanța dintre imaginea sursei galbene și punctul central al imaginii captate - iar din procesarea celor două imagini obținute de către cei doi senzori vizuali se va determina în timp real situația ansamblului de LED-uri și deci implicit a situației efectorului final al robotului.

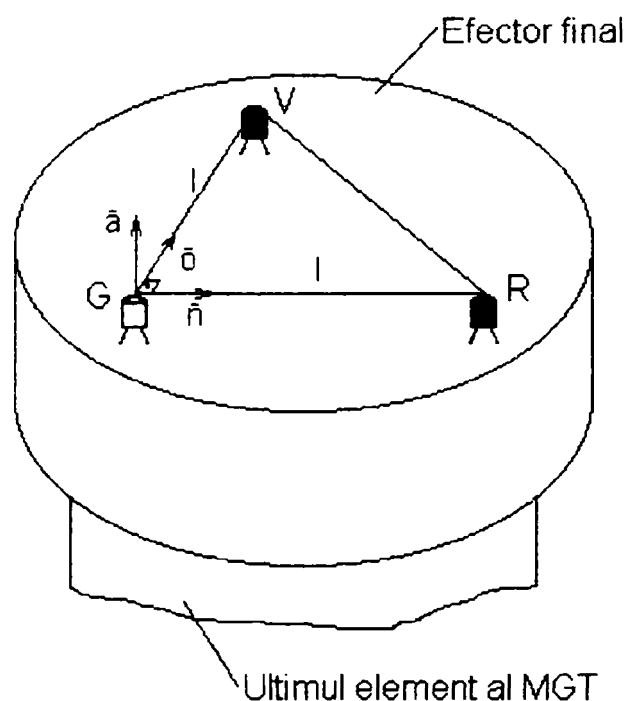


Fig.5.13. Amplasarea sistemului de surse luminoase pe efectorul final al robotului

Utilizând relațiile (5.31) și cunoscându-se coordonatele în planul imagine a imaginilor celor trei surse luminoase, dar și matricile de situație relativă a elementelor c.c.c. $-\alpha, \beta, \theta$ - se calculează pentru fiecare LED poziția, conform metodologiei prezentate în paragraful precedent.

Fie ${}^0 \underline{r}_R = ({}^0 x_R \quad {}^0 y_R \quad {}^0 z_R \quad 1)^T$, ${}^0 \underline{r}_V = ({}^0 x_V \quad {}^0 y_V \quad {}^0 z_V \quad 1)^T$ și ${}^0 \underline{r}_G = ({}^0 x_G \quad {}^0 y_G \quad {}^0 z_G \quad 1)^T$ vectorii de poziție pentru cele trei surse luminoase în sistemul de referință $O_v x_v y_v z_v$, a căror elemente se calculează cu ajutorul relațiilor (5.31).

Pentru determinarea orientării sistemului materializat de cele trei surse luminoase se consideră trei versori $\bar{n}, \bar{o}, \bar{a}$ care formează un sistem de referință triortogonal drept și care sunt atașați sistemului în punctul G, astfel încât versorul \bar{o} să fie orientat după direcția dată de dreapta care unește punctul G cu LED-ul care emite lumină verde, versorul \bar{n} să fie orientat după direcția dată de dreapta care unește punctul G cu LED-ul care emite lumină roșie, iar versorul \bar{a} să determine sistemul triortogonal drept (figura 5.13).

În acest caz matricea de orientare a sistemului de surse luminoase în raport cu sistemul de referință $O_v x_v y_v z_v$ este:

$${}^{O_v} \underline{O}_{ORc} = \begin{pmatrix} {}^{O_v} n_{x_{ORc}} & {}^{O_v} o_{x_{ORc}} & {}^{O_v} a_{x_{ORc}} \\ {}^{O_v} n_{y_{ORc}} & {}^{O_v} o_{y_{ORc}} & {}^{O_v} a_{y_{ORc}} \\ {}^{O_v} n_{z_{ORc}} & {}^{O_v} o_{z_{ORc}} & {}^{O_v} a_{z_{ORc}} \end{pmatrix} \quad (5.33)$$

unde:

$${}^{O_v} n_{x_{ORc}} = \frac{{}^{O_v} x_R - {}^{O_v} x_G}{l}$$

$${}^{O_v} n_{y_{ORc}} = \frac{{}^{O_v} y_R - {}^{O_v} y_G}{l} \quad (5.34)$$

$${}^{O_v} n_{z_{ORc}} = \frac{{}^{O_v} z_R - {}^{O_v} z_G}{l}$$

$${}^{O_v} o_{x_{ORc}} = \frac{{}^{O_v} x_V - {}^{O_v} x_G}{l}$$

$${}^{O_v} o_{y_{ORc}} = \frac{{}^{O_v} y_V - {}^{O_v} y_G}{l} \quad (5.35)$$

$${}^{O_v} o_{z_{ORc}} = \frac{{}^{O_v} z_V - {}^{O_v} z_G}{l}$$

iar l reprezintă lungimea catetei triunghiului dreptunghic isoscel determinat de cele trei surse luminoase.

Pentru determinarea componentelor versorului $\underline{\bar{a}}$ se ține cont de faptul că cei trei versori formează un sistem de coordonate triortogonal drept. Prin urmare versorul \underline{a} se obține ca și produs vectorial între ceilalți doi versori:

$$\underline{\bar{a}} = \underline{\bar{n}} \times \underline{\bar{o}} = \begin{vmatrix} \underline{\bar{i}} & \underline{\bar{j}} & \underline{\bar{k}} \\ {}^{O_v} n_{x_{ORc}} & {}^{O_v} n_{y_{ORc}} & {}^{O_v} n_{z_{ORc}} \\ {}^{O_v} o_{x_{ORc}} & {}^{O_v} o_{y_{ORc}} & {}^{O_v} o_{z_{ORc}} \end{vmatrix} \quad (5.36)$$

unde $(\underline{\bar{i}} \quad \underline{\bar{j}} \quad \underline{\bar{k}})$ sunt versorii sistemului de referință $O_v x_V y_V z_V$ atașat subsistemului vizual de urmărire.

Relația (5.36) se poate scrie și sub forma:

$$\underline{\bar{a}} = \begin{pmatrix} {}^{O_v} n_{y_{ORc}} & {}^{O_v} o_{z_{ORc}} & -{}^{O_v} n_{z_{ORc}} & {}^{O_v} o_{y_{ORc}} \\ {}^{O_v} n_{z_{ORc}} & {}^{O_v} o_{x_{ORc}} & -{}^{O_v} n_{x_{ORc}} & {}^{O_v} o_{z_{ORc}} \end{pmatrix} \cdot \underline{\bar{i}} + \begin{pmatrix} {}^{O_v} n_{z_{ORc}} & {}^{O_v} o_{y_{ORc}} & -{}^{O_v} n_{x_{ORc}} & {}^{O_v} o_{z_{ORc}} \\ {}^{O_v} n_{x_{ORc}} & {}^{O_v} o_{y_{ORc}} & -{}^{O_v} n_{y_{ORc}} & {}^{O_v} o_{x_{ORc}} \end{pmatrix} \cdot \underline{\bar{j}} + \begin{pmatrix} {}^{O_v} n_{x_{ORc}} & {}^{O_v} o_{y_{ORc}} & -{}^{O_v} n_{y_{ORc}} & {}^{O_v} o_{x_{ORc}} \\ {}^{O_v} n_{y_{ORc}} & {}^{O_v} o_{z_{ORc}} & -{}^{O_v} n_{z_{ORc}} & {}^{O_v} o_{y_{ORc}} \end{pmatrix} \cdot \underline{\bar{k}} \quad (5.37)$$

de unde componentele versorului \underline{a} sunt:

$${}^{O_v} a_{x_{ORc}} = {}^{O_v} n_{y_{ORc}} \cdot {}^{O_v} o_{z_{ORc}} - {}^{O_v} n_{z_{ORc}} \cdot {}^{O_v} o_{y_{ORc}}$$

$${}^{O_v} a_{y_{ORc}} = {}^{O_v} n_{z_{ORc}} \cdot {}^{O_v} o_{x_{ORc}} - {}^{O_v} n_{x_{ORc}} \cdot {}^{O_v} o_{z_{ORc}}$$

$${}^{O_v} a_{z_{ORc}} = {}^{O_v} n_{x_{ORc}} \cdot {}^{O_v} o_{y_{ORc}} - {}^{O_v} n_{y_{ORc}} \cdot {}^{O_v} o_{x_{ORc}} \quad (5.38)$$

În concluzie, situarea reală a țintei de urmărit - materializată de sistemul de trei surse luminoase - în raport cu sistemul de referință $O_v x_v y_v z_v$ legat de subsistemul vizual de urmărire este dată de matricea:

$${}^{O_v} T_{O_{Re}} = \begin{pmatrix} {}^{O_v} n_{x_{Re}} & {}^{O_v} o_{x_{Re}} & {}^{O_v} a_{x_{Re}} & {}^{O_v} p_{x_{Re}} \\ {}^{O_v} n_{y_{Re}} & {}^{O_v} o_{y_{Re}} & {}^{O_v} a_{y_{Re}} & {}^{O_v} p_{y_{Re}} \\ {}^{O_v} n_{z_{Re}} & {}^{O_v} o_{z_{Re}} & {}^{O_v} a_{z_{Re}} & {}^{O_v} p_{z_{Re}} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.39)$$

Prin determinarea matricei de situare a țintei în raport cu sistemul de referință $O_v x_v y_v z_v$ legat de subsistemul vizual de urmărire s-a obținut punctul de plecare pentru generarea matricei de situare reală a obiectului manipulat.

5.4. Achiziționarea informațiilor privind situarea reală a efectorului final prin subsistemul cu senzor vizual fix

5.4.1. Noțiuni introductive

Structura unui sistem robot la care comanda de corecție a situației efectorului final se realizează prin utilizarea unui senzor vizual fix se prezintă în figura 5.4.

Un astfel de sistem se utilizează la recunoașterea și manipularea unor obiecte care se găsesc în câmpul vizual al senzorului fix.

Desfășurarea în timp prezintă două etape și anume:

1. etapa de procesare a imaginii achiziționate – în care se recunoaște OL și se determină situarea lui în raport cu sistemul de referință atașat senzorului vizual fix, etapă care se desfășoară conform ordinogramei prezentate în figura 4.11; în această etapă se achiziționează imaginea obiectului de către senzor, după preprocesare și segmentare se determină prin procedeul de descriere unele trăsături caracteristice și de situare ale obiectului, iar pe baza acestor trăsături se recunoaște obiectul și i se calculează situarea reală a OL în raport cu sistemul de referință aferent senzorului vizual fix $O_v x_v y_v z_v$;
2. etapa de manipulare a OL, în care se determină situarea reală a EF în raport cu sistemul de referință al senzorului vizual fix $O_v x_v y_v z_v$, și respectiv situarea reală a EF în raport cu sistemul de referință legat de baza robotului; cunoscându-se această situație și situarea actuală a EF se comandă robotul în vederea prehensării OL.

5.4.2. Determinarea situației reale a OL exprimate în sistemul de referință legat de senzorul vizual fix

Pentru determinarea situației reale a OL exprimate în raport cu sistemul de referință aferent senzorului vizual fix $O_v x_v y_v z_v$, se pornește de la scena prezentată în figura 5.14.

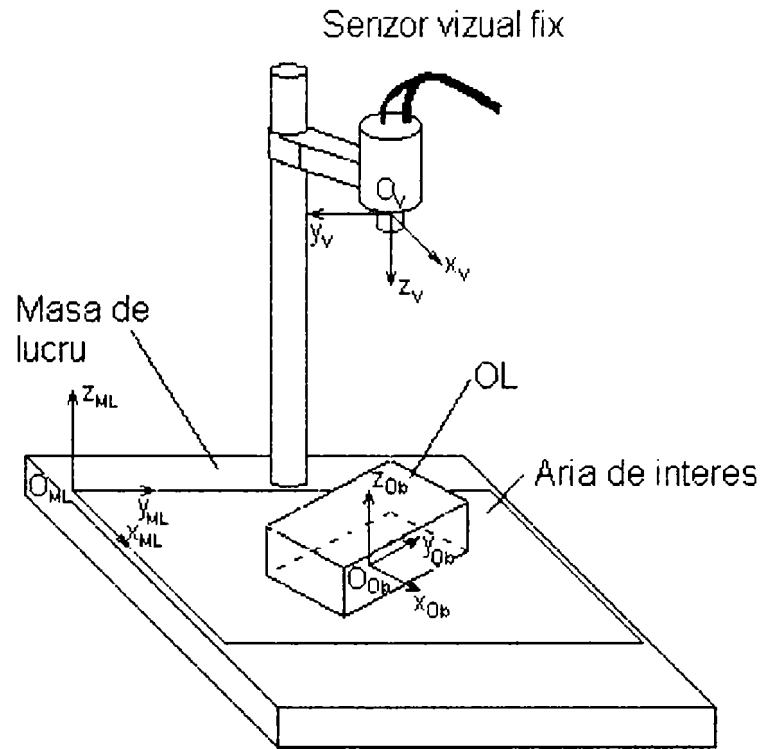


Fig. 5.14. Modul de determinare a situației reale a OL exprimată în sistemul de referință aferent senzorului vizual fix $O_v x_v y_v z_v$

Originea sistemului atașat senzorului vizual va avea coordonatele (x_v, y_v, D) în sistemul de referință $O_{ML} x_{ML} y_{ML} z_{ML}$ legat de masa de lucru. În acest caz matricea de trecere de la sistemul de coordonate $O_{ML} x_{ML} y_{ML} z_{ML}$ la sistemul de coordonate $O_v x_v y_v z_v$ legat de senzor va fi:

$${}^{O_v} T_{O_{ML}} = \text{Trans}_z(D) \cdot \text{Trans}_x(-x_v) \cdot \text{Trans}_y(y_v) \cdot \text{Rot}_x(\pi) \quad (5.40)$$

sau

$${}^{O_v} T_{O_{ML}} = \begin{pmatrix} 1 & 0 & 0 & -x_v \\ 0 & -1 & 0 & y_v \\ 0 & 0 & -1 & D \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.41)$$

Obiectului de lucru i se atașează un sistem de referință $O_{Ob} x_{Ob} y_{Ob} z_{Ob}$, astfel încât centrul O_{Ob} al sistemului va fi amplasat în planul mesei de lucru, pe direcția verticalei care

trece prin centrul de greutate al OL. Axa $O_{ob}Z_{ob}$ va o axă normală pe suprafața mesei de lucru și va trece prin centrul de greutate al OL. Axele $O_{ob}X_{ob}$, $O_{ob}Y_{ob}$ se aleg astfel încât să fie paralele cu direcțiile principale de inerție ale OL și plasate în planul mesei de lucru.

Dacă coordonatele centrului de greutate al OL exprimate în sistemul de referință legat de masa de lucru (se utilizează doar zona de interes a mesei de lucru) sunt $(x_{C_{ob}}, y_{C_{ob}}, 0)$ și dacă se cunoaște unghiul α pe care îl face direcția axei principale de inerție cu axa $O_{ML}X_{ML}$ atunci matricea de trecere de la sistemul de referință al OL la sistemul de referință al mesei de lucru va fi:

$${}^{O_{ML}}T_{O_{ob}} = Trans_x(x_{C_{ob}}) \cdot Trans_y(y_{C_{ob}}) \cdot Rot_z(\alpha) \quad (5.42)$$

După efectuarea calculelor se obține:

$${}^{O_{ML}}T_{O_{ob}} = \begin{pmatrix} c\alpha & -s\alpha & 0 & x_{C_{ob}} \\ s\alpha & c\alpha & 0 & y_{C_{ob}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.43)$$

Matricea de trecere de la sistemul de referință al OL la sistemul de referință $O_vX_vY_vZ_v$ legat de senzor se obține prin înmulțirea matricelor de trecere ${}^{O_v}T_{O_{ML}}$ cu ${}^{O_{ML}}T_{O_{ob}}$:

$${}^{O_v}T_{O_{ob}} = {}^{O_v}T_{O_{ML}} \cdot {}^{O_{ML}}T_{O_{ob}} \quad (5.44)$$

sau

$${}^{O_v}T_{O_{ob}} = \begin{pmatrix} c\alpha & -s\alpha & 0 & x_{C_{ob}} - x_v \\ -s\alpha & -c\alpha & 0 & -y_{C_{ob}} + y_v \\ 0 & 0 & -1 & D \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.45)$$

Atât pentru determinarea trăsăturilor de situare a OL (coordonatele centrului de greutate, unghiul direcției axei principale de inerție cu axa $O_{ML}X_{ML}$) cât și pentru determinarea trăsăturilor caracteristice necesare identificării OL se parcurg următorii pași.

Din observarea scenei din zona de interes, camera achiziționează cu o anumită rată imagini ale scenei, pe care digitizorul le transformă într-un set de semnale digitale matriceale.

Informația primară generată de senzorul vizual fix sub formă de semnal digital se preprocesează în vederea curățirii de zgomote, a îmbunătățirii contrastului prin aplicarea unor filtre.

Pentru un obiect de lucru de formă paralelipipedică prezentat în scena din figura 5.14, o imagine a acestuia achiziționată de senzorul vizual fix și preprocesată se prezintă în figura 5.15.a.

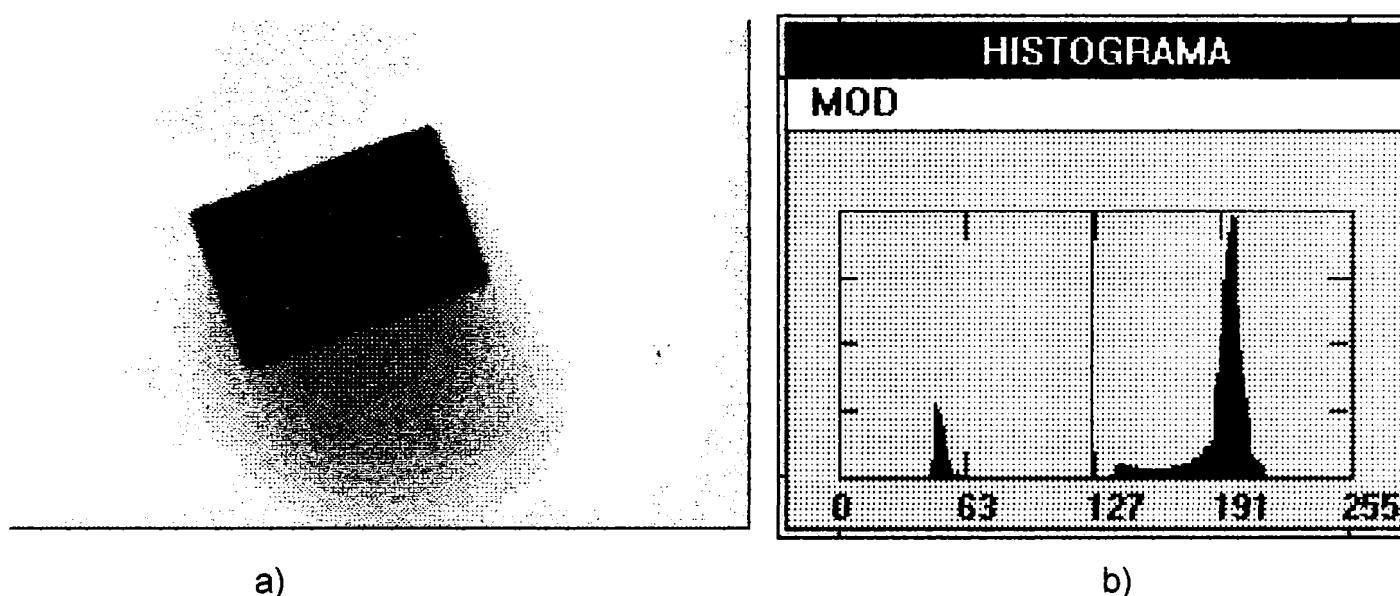


Fig.5.15. Imaginea OL după achiziție a) și histograma imaginii b)

Segmentarea imaginii utilizează tehnica binarizării imaginii prin clasificarea pixelilor imaginii în două clase: clasa OL și clasa mediu, utilizând în acest sens o intensitate prag T . Toți pixelii a căror intensitate se situează sub această valoare sunt încadrați în clasa obiect, iar pixelii a căror intensitate depășește valoarea prag sunt încadrați în clasa mediu. În figura 5.15.b se prezintă histograma imaginii OL achiziționate de senzorul vizual. Imaginea binarizată se prezintă în figura 5.16.a.

Pentru imaginea binarizată de dimensiune $m \times n$ se calculează momente de inerție de diferite ordine date de relația (4.46) și se obține:

- aria obiectului în planul imagine se calculează cu relația:

$$m_{00} = \sum_{i=1}^m \sum_{j=1}^n I(x_i, y_j) \quad (5.46)$$

unde $I(x_i, y_i) = \begin{cases} 0, & \text{daca punctul de coordonate } (x_i, y_i) \in \text{clasei mediu} \\ 1, & \text{daca punctul de coordonate } (x_i, y_i) \in \text{clasei obiect} \end{cases}$

- momentele statice se calculează cu formulele:

$$m_{10} = \sum_{i=1}^m \sum_{j=1}^n x_i \cdot I(x_i, y_j) \quad m_{01} = \sum_{i=1}^m \sum_{j=1}^n y_i \cdot I(x_i, y_j) \quad (5.47)$$

- momentele de inerție axiale sunt date de:

$$m_{02} = \sum_{i=1}^m \sum_{j=1}^n y_i^2 \cdot I(x_i, y_j) \quad m_{20} = \sum_{i=1}^m \sum_{j=1}^n x_i^2 \cdot I(x_i, y_j) \quad (5.48)$$

- momentul de inerție centrifugal se calculează cu relația:

$$m_{11} = \sum_{i=1}^m \sum_{j=1}^n x_i \cdot y_i \cdot I(x_i, y_j) \quad (5.49)$$

- coordonatele centrului de greutate în planul imagine se calculează cu formulele:

$${}^i x_{C_{ob}} = \frac{m_{10}}{m_{00}} = \frac{\sum_{i=1}^m \sum_{j=1}^n x_i \cdot I(x_i, y_j)}{\sum_{i=1}^m \sum_{j=1}^n I(x_i, y_j)} \quad (5.50)$$

$${}^i y_{C_{ob}} = \frac{m_{01}}{m_{00}} = \frac{\sum_{i=1}^m \sum_{j=1}^n y_i \cdot I(x_i, y_j)}{\sum_{i=1}^m \sum_{j=1}^n I(x_i, y_j)}$$

Între coordonatele centrului de greutate ale imaginii OL în planul imagine și coordonatele centrului de greutate ale OL în planul mesei de lucru există relația:

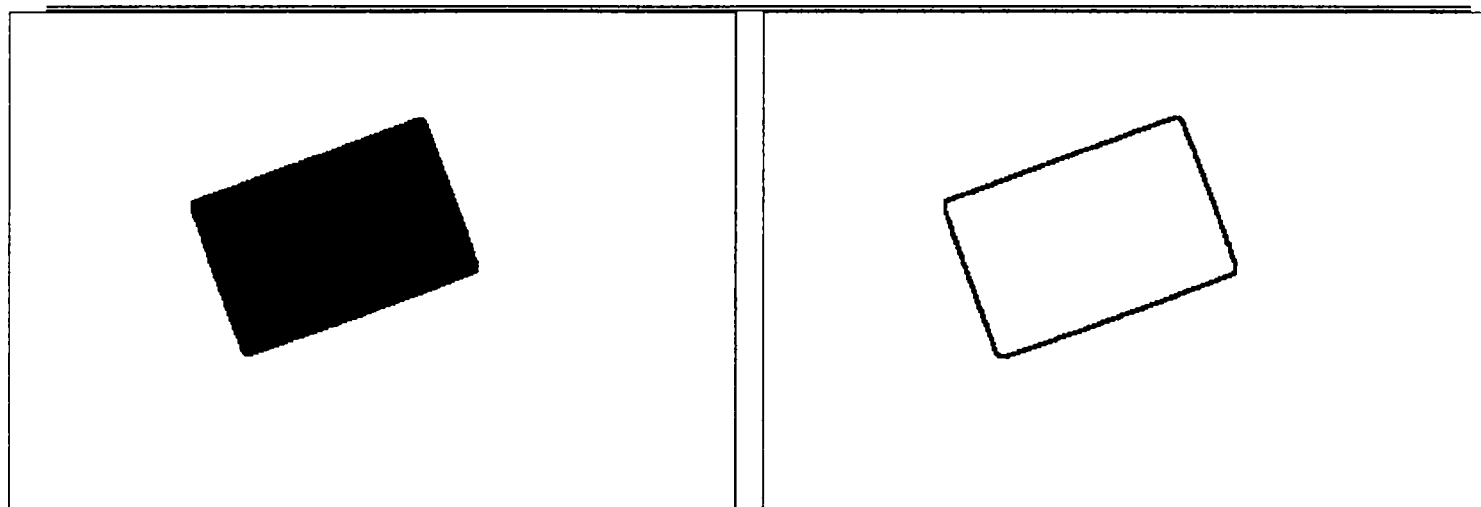
$$x_{C_{ob}} = \frac{{}^i x_{C_{ob}}}{f} \cdot D \quad (5.51)$$

$$y_{C_{ob}} = \frac{{}^i y_{C_{ob}}}{f} \cdot D$$

- orientarea α a direcției axei principale de inerție a OL este dată de relația:

$$\alpha = \frac{1}{2} \cdot \arctg\left(\frac{2 \cdot m_{11}}{m_{02} - m_{20}}\right) = \frac{1}{2} \cdot \arctg\left(\frac{2 \cdot \sum_{i=1}^m \sum_{j=1}^n x_i \cdot y_i \cdot I(x_i, y_j)}{\sum_{i=1}^m \sum_{j=1}^n y_i^2 \cdot I(x_i, y_j) - \sum_{i=1}^m \sum_{j=1}^n x_i^2 \cdot I(x_i, y_j)}\right) \quad (5.52)$$

- pentru determinarea conturului OL în planul imagine se procedează la determinarea muchiiilor OL, utilizând fie tehnica gradientului (a filtrelor Sobel), fie utilizarea grafurilor în detectarea muchiiilor, fie transformatele Hough. În figura 5.16.b se prezintă aplicarea tehnicii gradientului asupra imaginii digitizate din figura 5.16.a.



a)

b)

Fig.5.16. Imaginea binarizată a), detectarea muchiiilor OL b)

Prin atașarea codurilor Freeman conturului astfel obținut perimetrul OL va fi dat de:

$$P = \sum_{i=1}^8 l_i(\text{mod } 2) \cdot \sqrt{2} + (l_i(\text{mod } 2) + 1) \quad (5.53)$$

unde l_i este valoarea curentă a codului Freeman;

- se mai determină valorile maxime și minime ale coordonatelor x și y atinse de OL în planul imagine.

Pentru OL din scena 5.13 reprezentarea pe imaginea binarizată a OL a centrului de greutate a axelor principale de inerție se prezintă în figura 5.17.

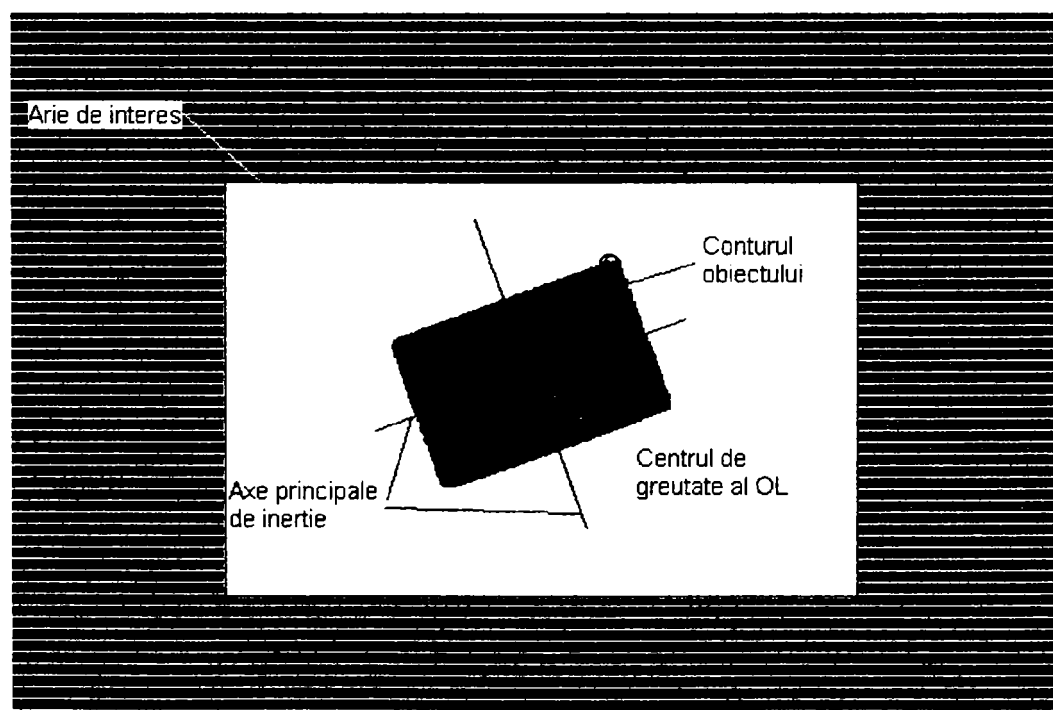


Fig.5.17. Reprezentarea grafică a trăsăturilor de situare pe imaginea binarizată a OL

5.4.3. Determinarea situații reale a EF în sistemul de referință atașat senzorului vizual fix

Pentru a putea prehensa corect OL, sistemul de referință atașat punctului caracteristic al EF al robotului trebuie să ocupe o situație „aliniată” cu sistemul de referință atașat OL (sistemul de referință $O_{Re}x_{Re}y_{Re}z_{Re}$ este translatat față de sistemul de referință $O_{Ob}x_{Ob}y_{Ob}z_{Ob}$ în lungul axei $O_{Ob}z_{Ob}$ cu mărimea d), conform schemei prezentate în figura 5.18. Această situație convențional se va numi situația reală a EF. Situația ideală (comandată) a EF este în acest caz situația de la care se pornește.

Matricea de trecere de la sistemul de referință $O_{Re}x_{Re}y_{Re}z_{Re}$ atașat EF al robotului la sistemul de referință $O_{Ob}x_{Ob}y_{Ob}z_{Ob}$ atașat OL este:

$${}^{O_{Ob}}T_{O_{Re}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.54)$$

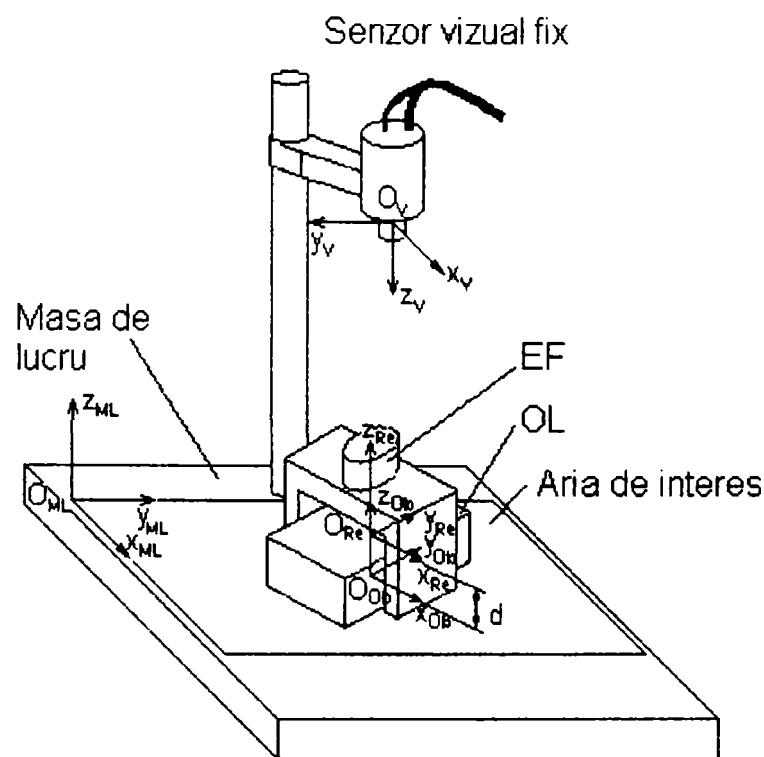


Fig.5.18. Modalitatea de prehensare a OL de către EF al robotului

Matricea de trecere de la sistemul de referință $O_{Re}x_{Re}y_{Re}z_{Re}$ atașat EF al robotului la sistemul de referință $O_vx_vy_vz_v$ legat de senzor va fi dată de relația:

$${}^{O_v}T_{O_{Re}} = {}^{O_v}T_{O_{Ob}} \cdot {}^{O_{Ob}}T_{O_{Re}} \quad (5.55)$$

de unde situarea reală a efectorului final - în raport cu sistemul de referință $O_v x_v y_v z_v$ legat de senzorul vizual este dată de matricea:

$${}^0 T_{O_{Re}} = \begin{pmatrix} c\alpha & -s\alpha & 0 & x_{C_{ob}} - x_v \\ -s\alpha & -c\alpha & 0 & -y_{C_{ob}} + y_v \\ 0 & 0 & -1 & D-d \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.56)$$

Relația (5.56) poate fi scrisă și sub formă generalizată conform relației (5.39).

5.5. Robotul

5.5.1. Structura robotului

Mecanismul dispozitivului de ghidare are la bază un lanț cinematic deschis, care cuprinde o serie de elemente și cuple cinematice conducătoare, având deci o topologie serială.

Robotul considerat în continuare este un robot de tip TTTRRR și deci mecanismul generator al robotului este un mecanism în coordonate carteziene, iar mecanismul de orientare al robotului este de tip Euler cu axele aferente cuplelor cinematice conducătoare de rotație concurente.

Mecanismul generator de traiectorie de tip TTT are în componență următoarele module de translație:

- modul de translație de bază;
- modul de translație pe verticală (ridicare/coborâre braț);
- modul de translație pe orizontală (extensie/retragere braț);

respectiv mecanismul de orientare prezintă trei module de rotație de orientare.

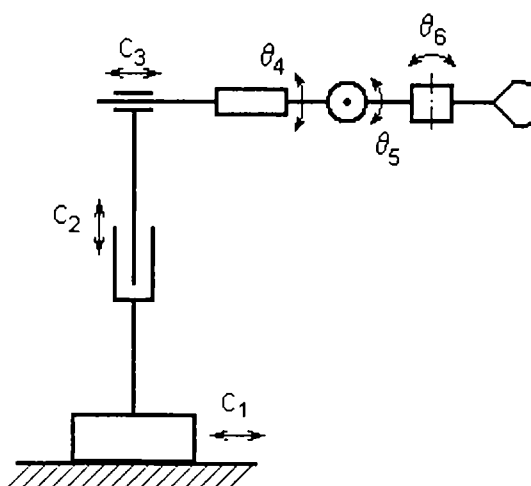


Fig.5.19. Schema structurală modulară a dispozitivului de ghidare

Schema structurală a dispozitivului de ghidare se prezintă în figura 5.19.

În figura 5.20 se prezintă schema cinematică a robotului. De baza robotului se consideră atașat un sistem de coordonate $O_0x_0y_0z_0$ – sistemul de coordonate universal. Se atașează sistemele de referință $O_i x_i y_i z_i$, $i=1 \div 5$ fiecărei cuple cinematice conducătoare, conform convenției Denavit-Hartenberg. Efactorului final i se atașează un sistem de coordonate $O_6 x_6 y_6 z_6$.

S-au notat cu l_i , $i=1 \div 6$ parametrii constructivi ai robotului, iar cu q_i parametrii de stare relativă generalizați:

$$q_i = \begin{cases} c_i, & i = 1 \div 3 \\ \theta_i, & i = 4 \div 6 \end{cases} \quad (5.57)$$

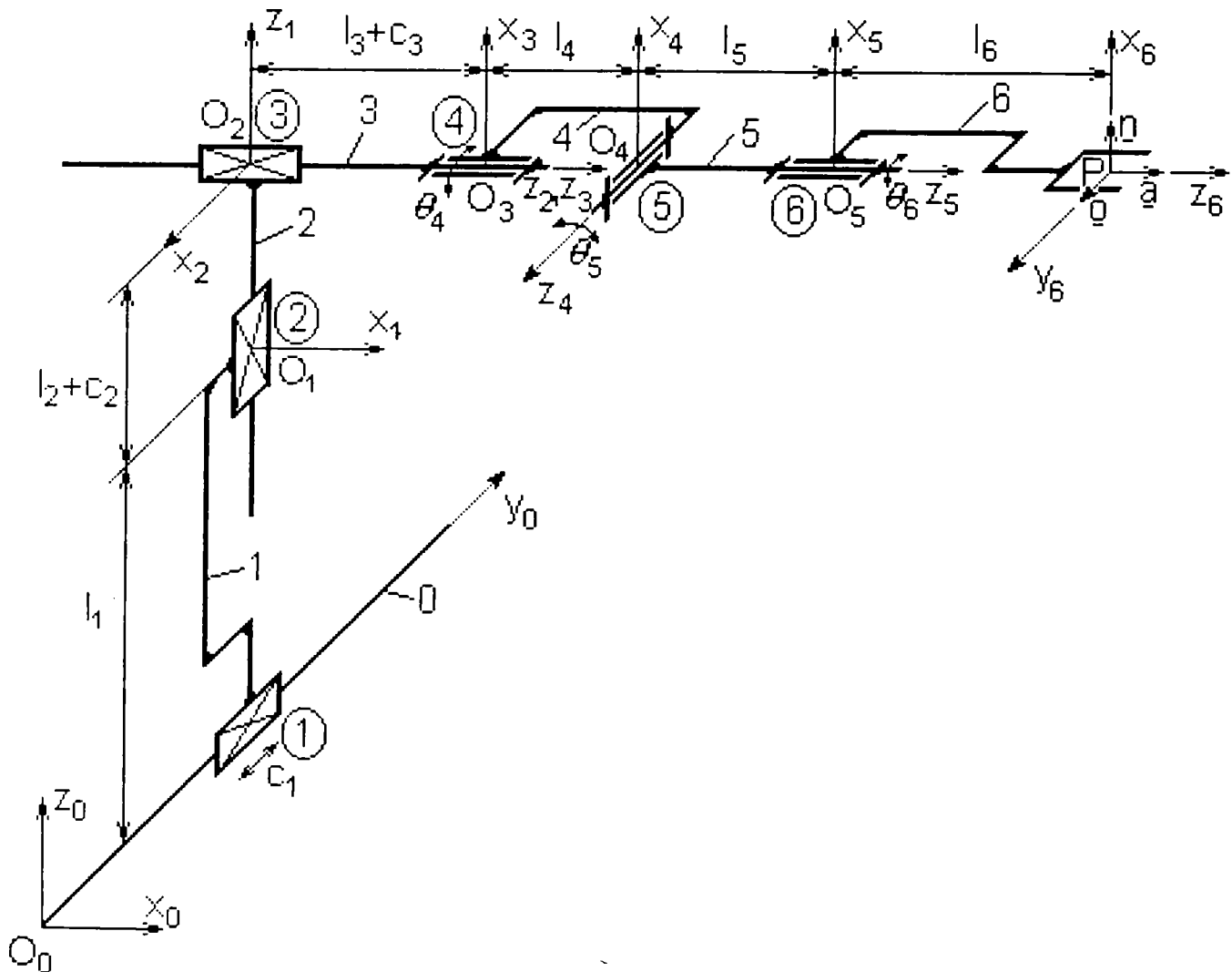


Fig.5.20. Schema cinematică a robotului

5.5.2. Modelul geometric al robotului

Ecuția de control a situației efectorului final pentru un robot în coordonate carteziane este dată de relația [Kova92]:

$${}^0\mathcal{T}_{O_c} = {}^0\mathcal{T}_6 \quad (5.58)$$

în care:

- ${}^0\mathcal{T}_{O_c}$ reprezintă situația ideală (programată) a efectorului final în sistemul de referință universal (legat de baza robotului);

- ${}^0\mathcal{T}_6$ reprezintă matricea de transformare omogenă aferentă dispozitivului de ghidare.

În vederea determinării matricei de transformare omogenă ${}^0\mathcal{T}_6$ aferentă dispozitivului de ghidare se vor utiliza matricile elementare ${}^{i-1}\mathcal{A}_i, i = 1 \div 6$.

Matricea de transformare omogenă ${}^0\mathcal{T}_3$ aferentă mecanismului generator de traiectorie este dată de:

$${}^0\mathcal{T}_3 = {}^0\mathcal{A}_1 \cdot {}^1\mathcal{A}_2 \cdot {}^2\mathcal{A}_3 \quad (5.59)$$

unde:

$${}^0\mathcal{A}_1 = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & c_1 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad {}^1\mathcal{A}_2 = \begin{vmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & l_2 + c_2 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (5.60)$$

$${}^2\mathcal{A}_3 = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & c_1 \\ 0 & 0 & 1 & l_3 + c_3 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Prin urmare:

$${}^0\mathcal{T}_3 = \begin{vmatrix} 0 & 0 & 1 & l_3 + c_3 \\ -1 & 0 & 0 & c_1 \\ 0 & -1 & 0 & l_1 + l_2 + c_2 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (5.61)$$

Matricea de transformare omogenă ${}^3\mathcal{T}_6$ aferentă mecanismului de orientare este dată de:

$${}^3\mathcal{T}_6 = {}^3\mathcal{A}_4 \cdot {}^4\mathcal{A}_5 \cdot {}^5\mathcal{A}_6 \quad (5.62)$$

unde:

$${}^3A_4 = \begin{vmatrix} c\theta_4 & 0 & -s\theta_4 & 0 \\ s\theta_4 & 0 & c\theta_4 & 0 \\ 0 & -1 & 0 & l_4 \\ 0 & 0 & 0 & 1 \end{vmatrix}, \quad {}^4A_5 = \begin{vmatrix} c\theta_5 & 0 & s\theta_5 & s\theta_5 \cdot l_5 \\ -s\theta_5 & 0 & c\theta_5 & -c\theta_5 \cdot l_5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (5.63)$$

$${}^5A_6 = \begin{vmatrix} c\theta_6 & -s\theta_6 & 0 & c\theta_6 \cdot l_6 \\ s\theta_6 & c\theta_6 & 0 & s\theta_6 \cdot l_6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

și deci:

$${}^3T_6 = \begin{vmatrix} c\theta_4 \cdot c\theta_5 \cdot c\theta_6 - s\theta_4 \cdot s\theta_6 & -c\theta_4 \cdot c\theta_5 \cdot s\theta_6 - s\theta_4 \cdot c\theta_6 & c\theta_4 \cdot s\theta_5 & (c\theta_4 \cdot c\theta_5 \cdot c\theta_6 - s\theta_4 \cdot s\theta_6) \cdot l_6 + c\theta_4 \cdot s\theta_5 \cdot l_5 \\ s\theta_4 \cdot c\theta_5 \cdot c\theta_6 + c\theta_4 \cdot s\theta_6 & -s\theta_4 \cdot c\theta_5 \cdot s\theta_6 + c\theta_4 \cdot c\theta_6 & s\theta_4 \cdot s\theta_5 & (s\theta_4 \cdot c\theta_5 \cdot c\theta_6 + c\theta_4 \cdot s\theta_6) \cdot l_6 + s\theta_4 \cdot s\theta_5 \cdot l_5 \\ -s\theta_5 \cdot c\theta_6 & s\theta_5 \cdot s\theta_6 & c\theta_5 & -s\theta_5 \cdot c\theta_6 \cdot l_6 + c\theta_5 \cdot l_5 + l_4 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (5.64)$$

Atunci, matricea de transformare omogenă 0T_6 aferentă dispozitivului de ghidare va avea expresia:

$${}^0T_6 = \begin{vmatrix} -s\theta_5 \cdot c\theta_6 & s\theta_5 \cdot s\theta_6 & c\theta_5 & -s\theta_5 \cdot c\theta_6 \cdot l_6 + c\theta_5 \cdot l_5 + l_4 + l_3 + c_3 \\ -c\theta_4 \cdot c\theta_5 \cdot c\theta_6 + s\theta_4 \cdot s\theta_6 & c\theta_4 \cdot c\theta_5 \cdot s\theta_6 + s\theta_4 \cdot c\theta_6 & -c\theta_4 \cdot s\theta_5 & -(c\theta_4 \cdot c\theta_5 \cdot c\theta_6 - s\theta_4 \cdot s\theta_6) \cdot l_6 - c\theta_4 \cdot s\theta_5 \cdot l_5 + c_1 \\ -s\theta_4 \cdot c\theta_5 \cdot c\theta_6 - c\theta_4 \cdot s\theta_6 & s\theta_4 \cdot c\theta_5 \cdot s\theta_6 - c\theta_4 \cdot c\theta_6 & -s\theta_4 \cdot s\theta_5 & -(s\theta_4 \cdot c\theta_5 \cdot c\theta_6 + c\theta_4 \cdot s\theta_6) \cdot l_6 - s\theta_4 \cdot s\theta_5 \cdot l_5 + l_1 + l_2 + c_2 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (5.65)$$

5.5.3. Determinarea situații ideale a efectorului final al robotului

Situarea comandată se obține prin impunerea unor valori pentru fiecare parametru de situație relativă generalizat al elementelor c.c.c.

Elementele matricei de situație rezultă din rezolvarea problemei de analiză cinematico-pozițională directe, identificând fiecare componentă a matricei de situație respectiv cu elementele matricei de transformare omogenă aferente dispozitivului de ghidare a robotului, conform relațiilor (5.58) și (5.65).

Din identificare rezultă elementele matricei de orientare:

$$\begin{aligned} {}^0n_x &= -s\theta_5 \cdot c\theta_6 \\ {}^0n_y &= -c\theta_4 \cdot c\theta_5 \cdot c\theta_6 + s\theta_4 \cdot s\theta_6 \\ {}^0n_z &= -s\theta_4 \cdot c\theta_5 \cdot c\theta_6 - c\theta_4 \cdot s\theta_6 \end{aligned}$$

$$\begin{aligned}
{}^{0_0} o_x &= s\theta_5 \cdot s\theta_6 \\
{}^{0_0} o_y &= c\theta_4 \cdot c\theta_5 \cdot s\theta_6 + s\theta_4 \cdot c\theta_6 \\
{}^{0_0} o_z &= s\theta_4 \cdot c\theta_5 \cdot s\theta_6 - c\theta_4 \cdot c\theta_6 \\
{}^{0_0} a_x &= c\theta_5 \\
{}^{0_0} a_y &= -c\theta_4 \cdot s\theta_5 \\
{}^{0_0} a_z &= -s\theta_4 \cdot s\theta_5
\end{aligned} \tag{5.66}$$

Elementele vectorului de poziție sunt:

$$\begin{aligned}
{}^{0_0} p_x &= -s\theta_5 \cdot c\theta_6 \cdot l_6 + c\theta_5 \cdot l_5 + l_4 + l_3 + c_3 \\
{}^{0_0} p_y &= -(c\theta_4 \cdot c\theta_5 \cdot c\theta_6 - s\theta_4 \cdot s\theta_6) \cdot l_6 - c\theta_4 \cdot s\theta_5 \cdot l_5 + c_1 \\
{}^{0_0} p_z &= -(s\theta_4 \cdot c\theta_5 \cdot c\theta_6 + c\theta_4 \cdot s\theta_6) \cdot l_6 - s\theta_4 \cdot s\theta_5 \cdot l_5 + l_1 + l_2 + c_2
\end{aligned} \tag{5.67}$$

Deci situarea ideală (comandată) a obiectului de manipulat este dată de:

$${}^{0_0} \underline{T}_{0_c} = \begin{pmatrix} {}^{0_0} n_x & {}^{0_0} o_x & {}^{0_0} a_x & {}^{0_0} p_x \\ {}^{0_0} n_y & {}^{0_0} o_y & {}^{0_0} a_y & {}^{0_0} p_y \\ {}^{0_0} n_z & {}^{0_0} o_z & {}^{0_0} a_z & {}^{0_0} p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5.68}$$

5.6. Determinarea erorilor de situare a obiectului manipulat și calculul comenzilor de corecție

5.6.1. Stabilirea matricei de eroare de situare

În vederea conducerii în poziție a sistemului robot, conform ordinogramei 5.6, este necesară determinarea matricei erorilor de situare ale obiectului manipulat, după care, printr-o analiză cinematico-pozițională inversă stabilirea comenzilor de corecție a situațiilor relative a elementelor fiecărei cuple cinematice conducătoare.

Se consideră sistemul de referință $O_c x_c y_c z_c$ aferent situației comandate (ideale) a obiectului de manipulat conform figurii 5.5. Atunci, matricea erorii de situare este dată de relația:

$${}^{0_c} \underline{E}_{0_{Rc}} = {}^{0_0} \underline{T}_{0_c}^{-1} \cdot {}^{0_0} \underline{T}_{0_{Rc}} \tag{5.69}$$

unde:

- ${}^{0_0} \underline{T}_{0_{Rc}}$ reprezintă matricea de situare reală determinată de subsistemul senzorial vizual (subsistem vizual de urmărire sau senzor vizual fix) în raport cu sistemul de referință universal, fiind dată de:

$${}^{0_0} \underline{T}_{0_{Rc}} = {}^{0_0} \underline{T}_{0_v} \cdot {}^{0_v} \underline{T}_{0_{Rc}} \tag{5.70}$$

în care ${}^{0_0}T_{0_1}$ reprezintă matricea de situare a subsistemul senzorial vizual în sistemul de referință universal și este dată de relația:

$${}^{0_0}T_{0_1} = \begin{pmatrix} 0_0 n_{x_{0_1}} & 0_0 o_{x_{0_1}} & 0_0 a_{x_{0_1}} & 0_0 p_{x_{0_1}} \\ 0_0 n_{y_{0_1}} & 0_0 o_{y_{0_1}} & 0_0 a_{y_{0_1}} & 0_0 p_{y_{0_1}} \\ 0_0 n_{z_{0_1}} & 0_0 o_{z_{0_1}} & 0_0 a_{z_{0_1}} & 0_0 p_{z_{0_1}} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.71)$$

iar matricea ${}^{0_1}T_{0_2}$ este dată de relațiile (5.39), (5.56);

- ${}^{0_0}T_{0_2}^{-1}$ reprezintă inversa matricei de situare comandată (5.68), dată de relația:

$${}^{0_0}T_{0_2}^{-1} = \begin{pmatrix} s_3 \cdot c_4 & s_3 \cdot s_4 \cdot c_4 \cdot c_5 & -c_3 \cdot s_4 \cdot c_4 \cdot c_5 & -l_6 + s_3 \cdot c_4 \cdot (l_4 + l_5 + c_3) + (c_3 \cdot s_4 + s_3 \cdot c_4 \cdot c_5) \cdot (l_2 + l_1 + c_3) - (s_4 \cdot s_5 \cdot c_4 \cdot c_5 \cdot c_6) \cdot c_4 \\ s_3 \cdot s_4 & s_3 \cdot c_4 \cdot c_4 \cdot c_5 & -c_3 \cdot c_4 \cdot c_4 \cdot c_5 & -s_3 \cdot s_4 \cdot (l_4 + l_5 + c_3) + (c_3 \cdot c_4 + s_3 \cdot c_4 \cdot c_5) \cdot (l_2 + l_1 + c_3) - (s_4 \cdot s_5 \cdot c_4 \cdot c_5 \cdot c_6) \cdot c_4 \\ c_4 & -c_4 \cdot s_4 & -s_4 \cdot s_4 & -l_5 \cdot c_4 \cdot (l_4 + l_5 + c_3) + s_4 \cdot s_4 \cdot (l_2 + l_1 + c_3) + c_4 \cdot s_4 \cdot c_4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.72)$$

Utilizând relațiile (5.39), (5.56), (5.71) și (5.72) matricea erorii de situare se poate scrie sub următoarea formă:

$${}^{0_0}E_{0_2} = \begin{pmatrix} {}^{0_0}\Delta n_{x_{0_2}} & {}^{0_0}\Delta o_{x_{0_2}} & {}^{0_0}\Delta a_{x_{0_2}} & {}^{0_0}\Delta p_{x_{0_2}} \\ {}^{0_0}\Delta n_{y_{0_2}} & {}^{0_0}\Delta o_{y_{0_2}} & {}^{0_0}\Delta a_{y_{0_2}} & {}^{0_0}\Delta p_{y_{0_2}} \\ {}^{0_0}\Delta n_{z_{0_2}} & {}^{0_0}\Delta o_{z_{0_2}} & {}^{0_0}\Delta a_{z_{0_2}} & {}^{0_0}\Delta p_{z_{0_2}} \\ 0 & 0 & 0 & 1 \end{pmatrix} = {}^{0_0}T_{0_1}^{-1} \cdot {}^{0_0}T_{0_1} \cdot {}^{0_1}T_{0_2} \quad (5.73)$$

unde componentele matricei erorii de situare sunt:

$$\begin{aligned} {}^{0_0}\Delta n_{x_{0_2}} &= s\theta_5 \cdot c\theta_6 \cdot \left({}^{0_0}n_{x_{0_1}} \cdot {}^{0_1}n_{x_{0_2}} + {}^{0_0}o_{x_{0_1}} \cdot {}^{0_1}n_{y_{0_2}} + {}^{0_0}a_{x_{0_1}} \cdot {}^{0_1}n_{z_{0_2}} \right) + \\ &+ (s\theta_4 \cdot s\theta_6 - c\theta_4 \cdot c\theta_5 \cdot c\theta_6) \cdot \left({}^{0_0}n_{y_{0_1}} \cdot {}^{0_1}n_{x_{0_2}} + {}^{0_0}o_{y_{0_1}} \cdot {}^{0_1}n_{y_{0_2}} + {}^{0_0}a_{y_{0_1}} \cdot {}^{0_1}n_{z_{0_2}} \right) - \\ &- (c\theta_4 \cdot s\theta_6 + s\theta_4 \cdot c\theta_5 \cdot c\theta_6) \cdot \left({}^{0_0}n_{z_{0_1}} \cdot {}^{0_1}n_{x_{0_2}} + {}^{0_0}o_{z_{0_1}} \cdot {}^{0_1}n_{y_{0_2}} + {}^{0_0}a_{z_{0_1}} \cdot {}^{0_1}n_{z_{0_2}} \right) \end{aligned}$$

$$\begin{aligned} {}^{0_0}\Delta n_{y_{0_2}} &= s\theta_5 \cdot s\theta_6 \cdot \left({}^{0_0}n_{x_{0_1}} \cdot {}^{0_1}n_{x_{0_2}} + {}^{0_0}o_{x_{0_1}} \cdot {}^{0_1}n_{y_{0_2}} + {}^{0_0}a_{x_{0_1}} \cdot {}^{0_1}n_{z_{0_2}} \right) + \\ &+ (s\theta_4 \cdot c\theta_6 + c\theta_4 \cdot c\theta_5 \cdot s\theta_6) \cdot \left({}^{0_0}n_{y_{0_1}} \cdot {}^{0_1}n_{x_{0_2}} + {}^{0_0}o_{y_{0_1}} \cdot {}^{0_1}n_{y_{0_2}} + {}^{0_0}a_{y_{0_1}} \cdot {}^{0_1}n_{z_{0_2}} \right) - \\ &- (c\theta_4 \cdot c\theta_6 - s\theta_4 \cdot c\theta_5 \cdot s\theta_6) \cdot \left({}^{0_0}n_{z_{0_1}} \cdot {}^{0_1}n_{x_{0_2}} + {}^{0_0}o_{z_{0_1}} \cdot {}^{0_1}n_{y_{0_2}} + {}^{0_0}a_{z_{0_1}} \cdot {}^{0_1}n_{z_{0_2}} \right) \end{aligned}$$

$$\begin{aligned} {}^{0_0}\Delta n_{z_{0_2}} &= c\theta_5 \cdot \left({}^{0_0}n_{x_{0_1}} \cdot {}^{0_1}n_{x_{0_2}} + {}^{0_0}o_{x_{0_1}} \cdot {}^{0_1}n_{y_{0_2}} + {}^{0_0}a_{x_{0_1}} \cdot {}^{0_1}n_{z_{0_2}} \right) - \\ &- c\theta_4 \cdot s\theta_5 \cdot \left({}^{0_0}n_{y_{0_1}} \cdot {}^{0_1}n_{x_{0_2}} + {}^{0_0}o_{y_{0_1}} \cdot {}^{0_1}n_{y_{0_2}} + {}^{0_0}a_{y_{0_1}} \cdot {}^{0_1}n_{z_{0_2}} \right) - \\ &- s\theta_4 \cdot s\theta_5 \cdot \left({}^{0_0}n_{z_{0_1}} \cdot {}^{0_1}n_{x_{0_2}} + {}^{0_0}o_{z_{0_1}} \cdot {}^{0_1}n_{y_{0_2}} + {}^{0_0}a_{z_{0_1}} \cdot {}^{0_1}n_{z_{0_2}} \right) \end{aligned}$$

$$\begin{aligned}
{}^{0_c} \Delta p_{y_{O_{Re}}} &= s\theta_5 \cdot s\theta_6 \cdot \left({}^{0_{O'}} n_{x_{O'}} \cdot {}^{0_c} p_{x_{O_{Re}}} + {}^{0_{O'}} o_{x_{O'}} \cdot {}^{0_c} p_{y_{O_{Re}}} + {}^{0_{O'}} a_{x_{O'}} \cdot {}^{0_c} p_{z_{O_{Re}}} \right) + \\
&+ (s\theta_4 \cdot c\theta_6 + c\theta_4 \cdot c\theta_5 \cdot s\theta_6) \cdot \left({}^{0_{O'}} n_{y_{O'}} \cdot {}^{0_c} p_{x_{O_{Re}}} + {}^{0_{O'}} o_{y_{O'}} \cdot {}^{0_c} p_{y_{O_{Re}}} + {}^{0_{O'}} a_{y_{O'}} \cdot {}^{0_c} p_{z_{O_{Re}}} \right) - \\
&- (c\theta_4 \cdot c\theta_6 - s\theta_4 \cdot c\theta_5 \cdot s\theta_6) \cdot \left({}^{0_{O'}} n_{z_{O'}} \cdot {}^{0_c} p_{x_{O_{Re}}} + {}^{0_{O'}} o_{z_{O'}} \cdot {}^{0_c} p_{y_{O_{Re}}} + {}^{0_{O'}} a_{z_{O'}} \cdot {}^{0_c} p_{z_{O_{Re}}} \right) \\
&+ \left\{ s\theta_5 \cdot s\theta_6 \cdot {}^{0_{O'}} p_{x_{O'}} + (s\theta_4 \cdot c\theta_6 + c\theta_4 \cdot c\theta_5 \cdot s\theta_6) \cdot {}^{0_{O'}} p_{y_{O'}} - (c\theta_4 \cdot c\theta_6 - s\theta_4 \cdot c\theta_5 \cdot s\theta_6) \cdot {}^{0_{O'}} p_{z_{O'}} + \right. \\
&\left. [-s\theta_5 \cdot s\theta_6 \cdot (l_4 + l_3 + c_3) + (c\theta_4 \cdot c\theta_6 + s\theta_4 \cdot c\theta_5 \cdot s\theta_6) \cdot (l_2 + l_1 + c_2) - (s\theta_4 \cdot s\theta_6 + c\theta_4 \cdot c\theta_5 \cdot s\theta_6) \cdot c_1] \right\} \\
{}^{0_c} \Delta p_{z_{O_{Re}}} &= s\theta_5 \cdot s\theta_6 \cdot \left({}^{0_{O'}} n_{x_{O'}} \cdot {}^{0_c} p_{x_{O_{Re}}} + {}^{0_{O'}} o_{x_{O'}} \cdot {}^{0_c} p_{y_{O_{Re}}} + {}^{0_{O'}} a_{x_{O'}} \cdot {}^{0_c} p_{z_{O_{Re}}} \right) + \\
&+ (s\theta_4 \cdot c\theta_6 + c\theta_4 \cdot c\theta_5 \cdot s\theta_6) \cdot \left({}^{0_{O'}} n_{y_{O'}} \cdot {}^{0_c} p_{x_{O_{Re}}} + {}^{0_{O'}} o_{y_{O'}} \cdot {}^{0_c} p_{y_{O_{Re}}} + {}^{0_{O'}} a_{y_{O'}} \cdot {}^{0_c} p_{z_{O_{Re}}} \right) - \\
&- (c\theta_4 \cdot c\theta_6 - s\theta_4 \cdot c\theta_5 \cdot s\theta_6) \cdot \left({}^{0_{O'}} n_{z_{O'}} \cdot {}^{0_c} p_{x_{O_{Re}}} + {}^{0_{O'}} o_{z_{O'}} \cdot {}^{0_c} p_{y_{O_{Re}}} + {}^{0_{O'}} a_{z_{O'}} \cdot {}^{0_c} p_{z_{O_{Re}}} \right) + c\theta_5 \cdot {}^{0_{O'}} p_{x_{O'}} \\
&- c\theta_4 \cdot s\theta_5 \cdot {}^{0_{O'}} p_{y_{O'}} - s\theta_4 \cdot s\theta_5 \cdot {}^{0_{O'}} p_{z_{O'}} - l_5 - c\theta_5 \cdot (l_4 + l_3 + c_3) + s\theta_4 \cdot s\theta_5 \cdot (l_2 + l_1 + c_2) + c\theta_4 \cdot s\theta_5 \cdot c_1
\end{aligned}$$

5.6.2. Modul de calcul al mărimilor de corecție de la nivelul fiecărei c.c.c.

Cunoscându-se elementele matricei erorii de situare ${}^{0_c} \underline{E}_{O_{Re}}$, prin utilizarea analizei cinematico-poziționale inverse se pot determina mărimile comenzilor de corecție de la nivelul fiecărei c.c.c. (Δq_i , $i = 1 \div 6$). Mărimile comenzilor de corecție de la nivelul fiecărei c.c.c. sunt acele semnale care se necesită a se aplica la nivelul fiecărei c.c.c. pentru a corecta situarea efectorului final al robotului, în vederea atingerii situației ideale (comandate) a efectorului final.

Pentru rezolvarea problemei de analiză cinematico-poziționale inverse se pornește de la:

$${}^{0_c} \underline{E}_{O_{Re}} = {}^0 \underline{T}_6 \quad (5.75)$$

Din rezolvarea ecuației matriceale (5.70) cu necunoscutele Δq_i , $i = 1 \div 6$, rezultă următoarele cazuri (care apar datorită necesității rezolvării nedeterminării care rezultă datorită împărțirii cu zero în cazul general dat de relațiile 5.80 pentru $\Delta\theta_5 = 0$ sau π):

- a) dacă $\Delta\theta_5 = 0$, atunci componentele matricei de transformare omogenă aferentă dispozitivului de ghidare (care reprezintă chiar mărimile de corecție de la nivelul fiecărei c.c.c.) sunt:

$$\begin{aligned}
{}^{0_c} \Delta n_x &= 0 & {}^{0_c} \Delta o_x &= 0 \\
{}^{0_c} \Delta n_y &= \cos(\Delta\theta_4 + \Delta\theta_6) & {}^{0_c} \Delta o_y &= -\sin(\Delta\theta_4 + \Delta\theta_6) \\
{}^{0_c} \Delta n_z &= \sin(\Delta\theta_4 + \Delta\theta_6) & {}^{0_c} \Delta o_z &= \cos(\Delta\theta_4 + \Delta\theta_6)
\end{aligned} \quad (5.76)$$

$$\begin{aligned}
{}^{0c} \Delta a_x &= -1 & {}^{0c} \Delta p_x &= -(l_5 + l_4 + l_3 + \Delta c_3) \\
{}^{0c} \Delta a_y &= 0 & {}^{0c} \Delta p_y &= \cos(\Delta\theta_4 + \Delta\theta_6) \cdot l_6 - \Delta c_1 \\
{}^{0c} \Delta a_z &= 0 & {}^{0c} \Delta p_z &= \sin(\Delta\theta_4 + \Delta\theta_6) \cdot l_6 - l_1 - l_2 - \Delta c_2
\end{aligned}$$

de unde se observă o nedeterminare între $\Delta\theta_4$ și $\Delta\theta_6$.

Parametrii de situare relativă generalizați sunt dați de:

$$\begin{aligned}
\Delta c_1 &= -{}^{0c} \Delta p_y + {}^{0c} \Delta n_y \cdot l_6 \\
\Delta c_2 &= -{}^{0c} \Delta p_z + {}^{0c} \Delta n_z \cdot l_6 + l_1 + l_2 \\
\Delta c_3 &= -{}^{0c} \Delta p_x + l_5 + l_4 + l_3 \\
\Delta\theta_4 &= \alpha \\
\Delta\theta_5 &= 0 \\
\Delta\theta_6 &= \begin{cases} \arctg\left(\frac{{}^{0c} \Delta n_z}{{}^{0c} \Delta n_y}\right) - \alpha, & \text{daca } {}^{0c} \Delta n_y \neq 0 \\ \pm \frac{\pi}{2} - \alpha, & \text{daca } {}^{0c} \Delta n_y = 0 \end{cases}
\end{aligned} \tag{5.77}$$

unde α este o variabilă.

b) dacă $\Delta\theta_5 = \pi$, atunci componentele matricei de transformare omogenă aferentă dispozitivului de ghidare (mărimile de corecție de la nivelul fiecărei c.c.c.) sunt:

$$\begin{aligned}
{}^{0c} \Delta n_x &= 0 & {}^{0c} \Delta o_x &= 0 \\
{}^{0c} \Delta n_y &= -\cos(\Delta\theta_4 - \Delta\theta_6) & {}^{0c} \Delta o_y &= -\sin(\Delta\theta_4 - \Delta\theta_6) \\
{}^{0c} \Delta n_z &= -\sin(\Delta\theta_4 - \Delta\theta_6) & {}^{0c} \Delta o_z &= \cos(\Delta\theta_4 - \Delta\theta_6) \\
{}^{0c} \Delta a_x &= 1 & {}^{0c} \Delta p_x &= -(l_5 + l_4 + l_3 + \Delta c_3) \\
{}^{0c} \Delta a_y &= 0 & {}^{0c} \Delta p_y &= -\cos(\Delta\theta_4 - \Delta\theta_6) \cdot l_6 - \Delta c_1 \\
{}^{0c} \Delta a_z &= 0 & {}^{0c} \Delta p_z &= -\sin(\Delta\theta_4 - \Delta\theta_6) \cdot l_6 - l_1 - l_2 - \Delta c_2
\end{aligned} \tag{5.78}$$

de unde se observă o nedeterminare între $\Delta\theta_4$ și $\Delta\theta_6$.

Parametrii de situare relativă generalizați sunt:

$$\begin{aligned}
\Delta c_1 &= -{}^{0c} \Delta p_y + {}^{0c} \Delta n_y \cdot l_6 \\
\Delta c_2 &= -{}^{0c} \Delta p_z + {}^{0c} \Delta n_z \cdot l_6 + l_1 + l_2 \\
\Delta c_3 &= -{}^{0c} \Delta p_x + l_5 + l_4 + l_3 \\
\Delta \theta_4 &= \alpha \\
\Delta \theta_5 &= \pi \\
\Delta \theta_6 &= \begin{cases} \alpha - \operatorname{arctg} \left(\frac{-{}^{0c} \Delta n_z}{-{}^{0c} \Delta n_y} \right), & \text{daca } {}^{0c} \Delta n_y \neq 0 \\ \alpha \mp \frac{\pi}{2}, & \text{daca } {}^{0c} \Delta n_y = 0 \end{cases}
\end{aligned} \tag{5.79}$$

unde α este o valoare oarecare a unghiului $\Delta \theta_4$.

Prin urmare atât în cazul a) cât și în cazul b) doar suma $\Delta \theta_4 \pm \Delta \theta_6$ va da o imagine corectă a orientării efectorului final.

c) dacă $\Delta \theta_5 \neq 0$ sau π , atunci parametrii de situare relativă generalizați se vor exprima sub forma:

$$\begin{aligned}
\Delta c_1 &= -{}^{0c} \Delta p_y + {}^{0c} \Delta n_y \cdot l_6 + {}^{0c} \Delta a_y \cdot l_5 \\
\Delta c_2 &= -{}^{0c} \Delta p_z + {}^{0c} \Delta n_z \cdot l_6 + {}^{0c} \Delta a_z \cdot l_5 + l_2 + l_1 \\
\Delta c_3 &= -{}^{0c} \Delta p_x + {}^{0c} \Delta n_x \cdot l_6 + {}^{0c} \Delta a_x \cdot l_5 + l_4 + l_3 \\
\Delta \theta_4 &= \begin{cases} \operatorname{arctg} \left(\frac{{}^{0c} \Delta a_z}{{}^{0c} \Delta a_y} \right), & \text{daca } {}^{0c} \Delta a_y \neq 0 \\ \pm \frac{\pi}{2}, & \text{daca } {}^{0c} \Delta a_y = 0 \end{cases} \\
\Delta \theta_5 &= \begin{cases} \operatorname{arctg} \left(\frac{\pm \sqrt{({}^{0c} \Delta a_y)^2 + ({}^{0c} \Delta a_z)^2}}{{}^{0c} \Delta a_x} \right), & \text{daca } {}^{0c} \Delta a_x \neq 0 \\ \pm \frac{\pi}{2}, & \text{daca } {}^{0c} \Delta a_x = 0 \end{cases} \\
\Delta \theta_6 &= \begin{cases} \operatorname{arctg} \left(\frac{-{}^{0c} \Delta a_x}{{}^{0c} \Delta n_x} \right), & \text{daca } {}^{0c} \Delta n_x \neq 0 \\ \pm \frac{\pi}{2}, & \text{daca } {}^{0c} \Delta n_x = 0 \end{cases}
\end{aligned} \tag{5.80}$$

În concluzie, utilizând relațiile (5.77), (5.79) și (5.80) se pot determina mărimile comenzilor de corecție de la nivelul fiecărei c.c.c. (Δq_i , $i = 1 \div 6$).

5.7. Modalități de programare a comenzilor de corecție a erorilor de situație

Pornind de la modalitățile cunoscute de programare a robotului și anume: programe secvențiale, punct cu punct, multipunct și cele care asigură o conducere pe traiectorie continuă, programarea comenzilor de corecție a erorilor de situație determinate la nivelul fiecărei c.c.c. se poate realiza după unul din aceste tipuri de programe.

Modalitățile de programare a comenzilor de corecție în funcție de simultaneitatea mișcărilor relative ale elementele c.c.c. pot utiliza:

- *programe secvențiale*, care permit realizarea corecțiilor la nivelul fiecărei c.c.c. în mod succesiv, prezentând avantajul unei conduceri simplificate (la un moment dat este controlată numeric doar mișcarea unei singure axe), dezavantajul constituindu-l timpul mare de realizare a sarcinii de execuție;

- *programele care permit corecția simultană pe toate c.c.c. acestea fiind:*
 - *programe care vor realiza corecția doar pentru situația finală comandată a obiectului de manipulat și care sunt utilizate atunci când nu este necesară (din punctul de vedere al procesului tehnologic) impunerea unei anumite traiectorii între situația curentă și cea următoare (via sau țintă), dar se dorește mișcări simultane pe toate axele; în acest caz singura condiție impusă este ca traiectoria să nu fie una de coliziune, mișcările pe fiecare axă încep deodată și se termină deodată; traiectoria obținută rămâne una oarecare, imprevizibilă, greu exprimabilă matematic și cu un aspect absolut oarecare pentru operatorul uman;*
 - *programe care vor realiza corecția și pentru unele situații intermediare ale obiectului de manipulat și la care se impune trecerea punctului caracteristic din situația curentă în situația următoare printr-un anumit număr de situații de precizie impuse; astfel se asigură realizarea unei deviații maxime impuse a traiectoriei reale față de cea ideală, care este cu atât mai mică cu cât se impune trecerea punctului caracteristic prin mai multe situații de precizie;*
 - *programele de traiectorie continuă se vor utiliza la acele operații tehnologice la care se impune ca punctul caracteristic să urmeze riguros un anumit drum; de regulă, se încearcă exprimarea analitică a traiectoriei*

impuse sau se procedează la aproximarea acesteia pe porțiuni cu segmente de curbe simple: drepte, arce de cerc, arce de parabolă, fracțiuni de sinusoidă (interpolare).

În cazul sistemului robot la care comanda de corecție a situării efectorului final se realizează prin utilizarea unui senzor vizual fix, care urmărește scena din mediu în mod secvențial, discret, există o singură posibilitate de realizare a comenzii de corecție a erorii de situare care rezultă chiar din principiul metodei: corecția se realizează printr-o singură comandă, de mișcare a EF din situarea ideală (comandată) în situarea reală. Programul utilizat în acest sens poate fi sau un program secvențial sau un program punct cu punct.

Pentru cazul sistemului robot la care comanda de corecție a situării efectorului final se realizează prin utilizarea informațiilor furnizate de un subsistem vizual de urmărire continuă există mai multe posibilități de programare a comenzilor erorilor de situare, după cum rezultă din tabelul 5.1.

Tabelul 5.1. Corelația între tipul de program utilizat și modul de realizare a corecției

Corecție Program	La sfârșitul intervalului de mișcare	La sfârșitul unor intervale intermediare	În mod continuu
Secvențial	x	x	-
Punct cu punct	x	-	-
Multipunct	-	x	-
Traietorie continuă	-	-	x

Din ordinograma prezentată în figura 5.6 rezultă și modalitatea de programare care urmează a fi implementată în cadrul sistemului. Deoarece mărimea controlată o reprezintă situarea (poziția) efectorului final al robotului pentru realizarea corecției la nivelul fiecărei c.c.c. se va utiliza comanda în poziție.

Celelalte trei variante de programare: secvențială, punct cu punct și multipunct se pot realiza utilizând sistemul prezentat de către autor, cu observația că - principal - la o programare secvențială nu se justifică un sistem de conducere globală.

5.8. Concluzii

În cadrul capitolului s-a dezvoltat un studiu teoretic al conducerii senzoriale a roboților utilizând ca buclă de reacție informațiile furnizate de senzori vizuali.

Ținând cont de schema bloc a sistemului robot care prezintă conducere senzorială și care utilizează informații furnizate de senzori vizuali, schemă prezentată în figura 5.1. și comparând-o cu schema bloc din figura 1.4. a unui sistem robot care prezintă o conducere globală se poate concluziona că din punct de vedere funcțional cele două scheme bloc sunt identice.

Prin urmare în continuare se vor utiliza ambele exprimări pentru sistemul robot care utilizează în vederea conducerii sale informații furnizate de senzori.

Din programul de lucru al robotului se determină situarea ideală (programată) a EF și cu ajutorul generatorului de traiectorie parametrii de situare relativă generalizați la nivelul fiecărei cuple cinematice conducătoare.

Se compară această situare ideală cu situarea reală determinată cu ajutorul subsistemului senzorial vizual (subsistem care se poate materializa fie printr-un subsistem vizual de urmărire, fie printr-un senzor vizual fix), rezultând o eroare de situare, care prin intermediul unei analize cinematico-poziționale inverse generează mărimile de corecție a situației relative a elementelor cuplelor cinematice conducătoare ale robotului.

Semnalul rezultat din suprapunerea peste semnalul prescris a mărimilor de corecție a situației relative a elementelor cuplelor cinematice conducătoare ale robotului, se transmite sistemului de acționare în vederea punerii în mișcare a elementelor componente ale sistemului mecanic.

Autorul a stabilit modelele matematice pentru determinarea situației reale a EF în cele două cazuri considerate:

- situarea ideală corespunde situației programate, iar situarea reală (realizată) sesizată de subsistemul senzorial vizual este utilizată de către regulatorul global pentru obținerea corecțiilor de situare a efectorului final; în acest caz subsistemul senzorial vizual se materializează printr-un subsistem vizual de urmărire continuă a efectorului final;
- situarea reală este situarea scop, situarea OL sesizată de subsistemul senzorial vizual, iar situarea efectorului final la momentul inițial este situarea

ideală – în acest caz subsistemul senzorial vizual se materializează printr-un senzor vizual fix care va urmări scena din mediu în mod secvențial, discret.

În cadrul capitolului s-au stabilit relațiile matematice pentru determinarea erorilor de situare a țintei (efectorul final al robotului, respectiv punctul caracteristic al efectorului final) precum și calculul mărimilor de corecție a situației relative a elementelor cuplelor cinematice conducătoare ale robotului.

CAPITOLUL 6 Concepția unui sistem concret de conducere senzorială a unui robot utilizând informații furnizate de un sistem vizual de urmărire a situării efectorului final. Modelarea geometrică a sistemului și simularea funcționării lui prin animație

6.1. Introducere

Pentru confirmarea rezultatelor teoretice obținute în capitolul cinci autorul a conceput un sistem de conducere senzorială concretă a unui robot utilizând informații furnizate de un sistem vizual de urmărire.

Scopul modelării geometrice a sistemului robot și simularea funcționării lui prin animație îl constituie validarea aspectelor teoretice prezentate în capitolul 5.

Întrucât sistemul de conducere senzorială a robotului care utilizează informații furnizate de un subsistem vizual de urmărire are în componența sa două subsisteme – robotul și subsistemul vizual de urmărire, s-a necesitat realizarea modelului geometric 3D pentru fiecare subsistem în parte. Subsistemul vizual de urmărire prezintă doi senzori vizuali, iar dispozitivul de ghidare al robotului s-a ales de tip TTT.

Pentru realizarea simulării funcționării sistemului prin animație s-a programat o traiectorie de realizat pentru punctul caracteristic al efectorului final al robotului, pentru care s-a determinat variația în timp a parametrilor generalizați ai mișcărilor relative ale elementelor cuplelor cinematice conducătoare ale dispozitivului de ghidare al robotului.

Din condiția ca direcțiile de vizare ale celor doi senzori vizuali să treacă totdeauna prin punctul caracteristic aferent efectorului final al robotului, au rezultat parametrii generalizați obținuți din simulare ai mișcărilor relative ale elementelor subsistemului vizual de urmărire.

Verificarea corectitudinii algoritmului pentru determinarea poziției țintei stabilit în cadrul paragrafului 5.3.3. se va realiza prin compararea parametrilor generalizați obținuți din simulare ai mișcărilor relative ale elementelor subsistemului vizual de urmărire cu parametrii generalizați rezultați din relațiile matematice ai mișcărilor relative ale elementelor subsistemului vizual de urmărire.

6.2. Modelarea geometrică 3D a sistemului de conducere senzorială a robotului

6.2.1. Noțiuni introductive

Așa cum s-a prezentat și în capitolul precedent, sistemul de conducere senzorială a robotului, care utilizează informații furnizate de senzori vizuali în vederea realizării conducerii, are în componența sa două subsisteme și anume: subsistemul vizual de urmărire și respectiv robotul. Prin urmare, pentru a putea simula în 3D mișcările sistemului robot în timpul conducerii globale a acestuia sunt necesare modelele geometrice 3D a fiecărui subsistem component al sistemului robot.

Modelarea sistemului robot s-a realizat cu ajutorul programului de proiectare asistată de calculator CATIA-CADAM al firmei franceze Dassault Systemes - Franța.

Dintre produsele componente ale pachetului CATIA-CADAM a fost utilizat programul *CATIA-CADAM Drafting*, cel mai potrivit pentru activitatea de modelare (CADD – Computer Aided Design and Drafting), datorită facilităților destinate creării, editării și managementului desenelor și documentelor tehnice conferite de existența unor unelte puternice de desenare – modelare.

6.2.2. Etapele parcurse

În cele ce urmează se prezintă etapele de realizare ale modelului geometric al sistemului robot. În acest scop, se utilizează modul de lucru *workspace* și sistemul ortogonal de coordonate *Oxyz*, folosind pentru desenare funcția *sketchit* care permite trasarea rapidă a conturilor 2D cu ajutorul comenzilor *line*, *point*, *curve*, fără a fi necesară respectarea strictă a cotelor reperelor componente ale ansamblului.

Într-o primă etapă se realizează conturul plan al dispozitivului, utilizând ca plan de lucru curent planul *Oxy*, datorită faptului că pentru modelare este necesară realizarea regiunilor 2D sau a conturilor închise, condiție obligatorie pentru transformarea lor în elemente 3D.

Această etapă este urmată de extrudarea conturilor obținute, pentru a li se atribui cea de-a treia dimensiune (grosimea), în lungul axei *Oz*. Reperele care au în componență suprafețe de revoluție sunt obținute prin imprimarea unei mișcări de rotație profilului generator al suprafeței în jurul axei de rotație care le generează. De asemenea elementele de formă complexă sunt obținute utilizând funcțiile de modelare 3D ale solidelor (uniune, extragere, intersecție, secționare, etc.)

Pentru realizarea reperelor cu aceeași formă care apar în model de mai multe ori având dimensiuni diferite se utilizează facilitatea de parametrizare, prin specificarea dimensiunilor sub formă literală. Această facilitate este utilizată pentru realizarea reperelor de tip șurub, piuliță, șaibă, bucușă, etc., întrucât astfel ele pot fi modificate foarte ușor din punct de vedere dimensional prin specificarea valorilor efective ale mărimilor parametrizate în momentul introducerii lor în componența modelului.

După modelarea tuturor reperelor acestea sunt transformate din solide 3D în *ditto* și sunt introduse, fiecare separat, într-o librărie de repere (*library ditto*), de unde vor fi accesate în vederea realizării modelului asamblat. Unele repere standardizate au fost alese din bibliotecile globale sau locale de piese standardizate oferite de către programul CATIA.

Modelul se assemblează prin specificarea punctului de inserție în proiect a fiecărui reper, precum și a elementelor care îi definesc poziția, direcția și scara de desenare, eventual constrângerile parametrice necesare, utilizând un sistem de coordonate *Oxyz* față de care se stabilește și poziția principală de vizualizare (proiecție izometrică în acest caz). Poziționarea corectă a componentelor se realizează prin utilizarea unor elemente ajutătoare (puncte, linii, curbe) care trec prin punctele dorite și permit utilizarea lor ca elemente de reper și de aliniere.

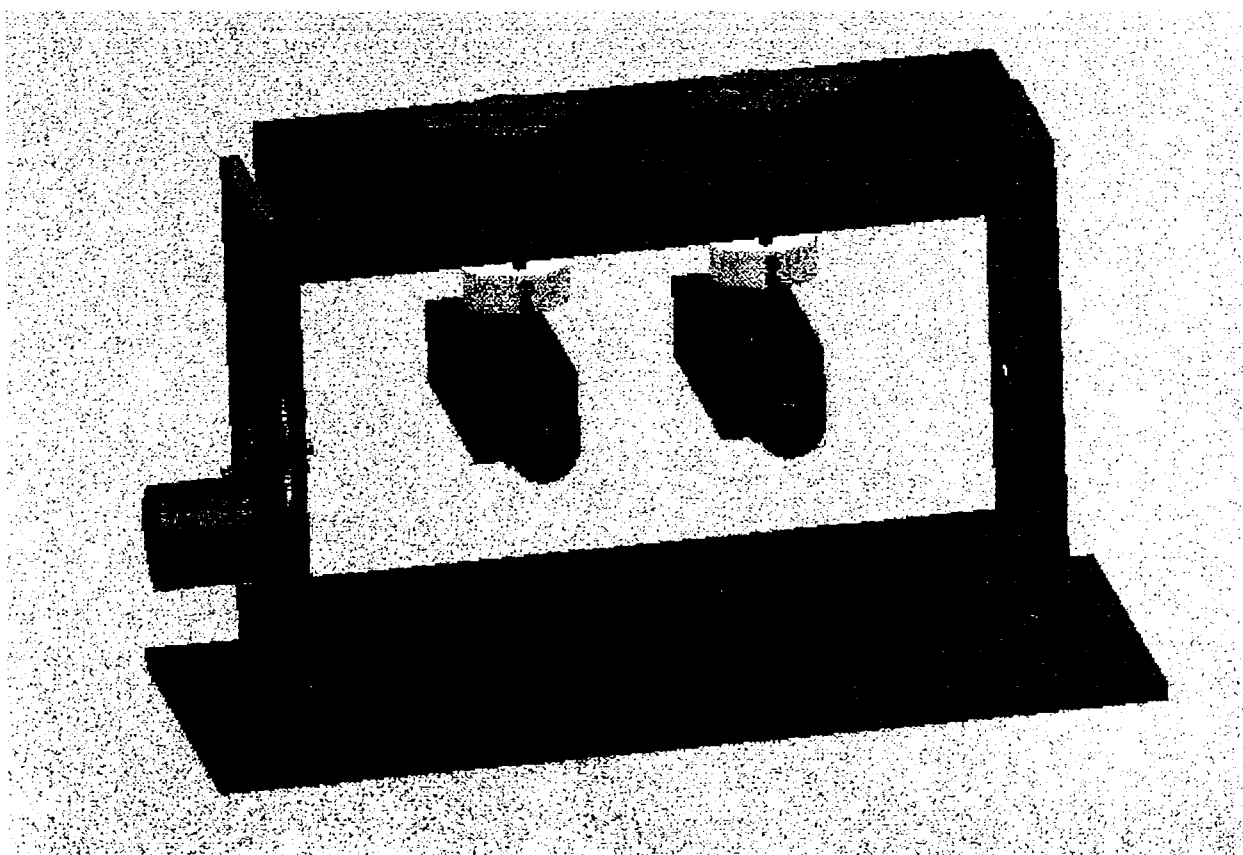


Fig.6.1. Modelare geometrică 3D a subsistemului vizual de urmărire – vedere din față

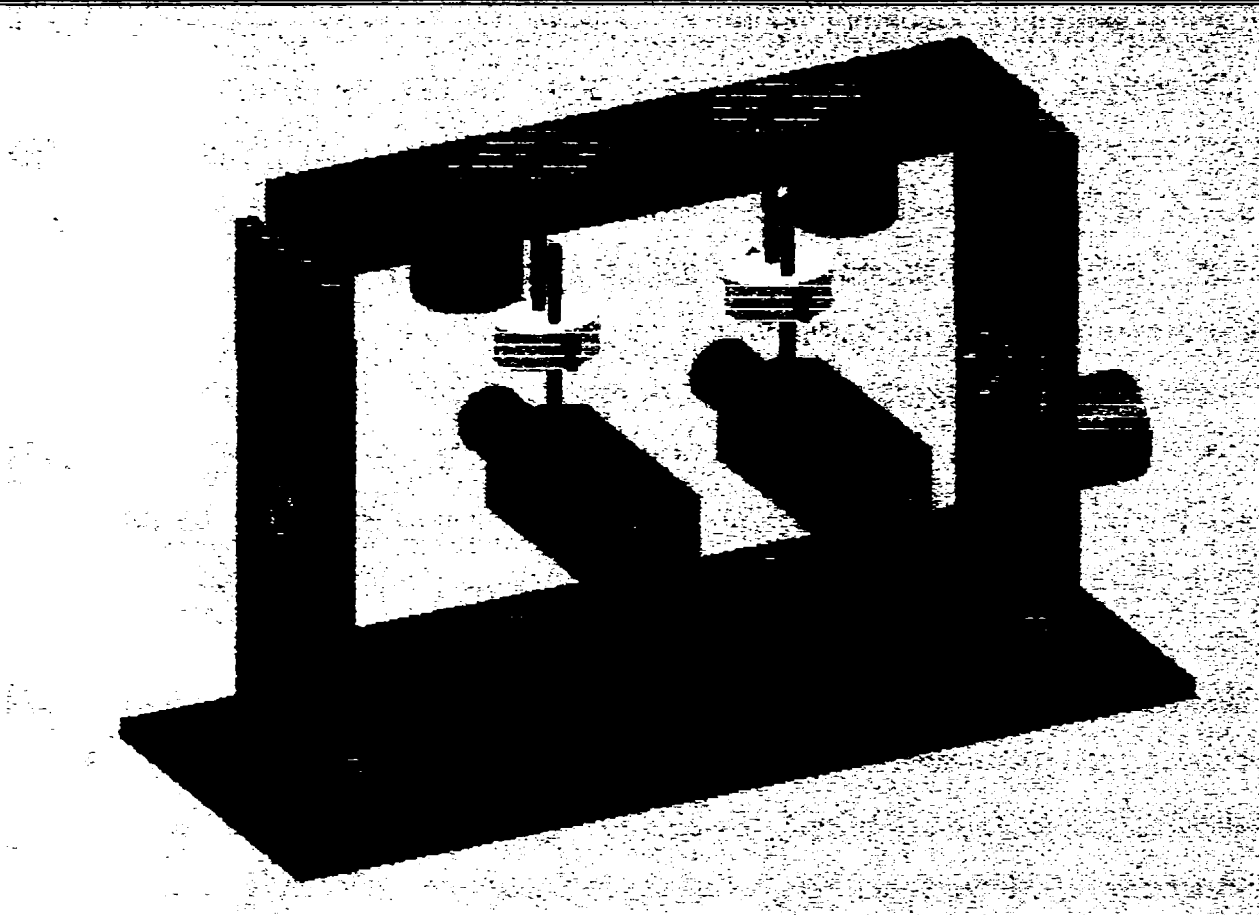


Fig.6.2. Modelare geometrică 3D a subsistemului vizual de urmărire – vedere din spate

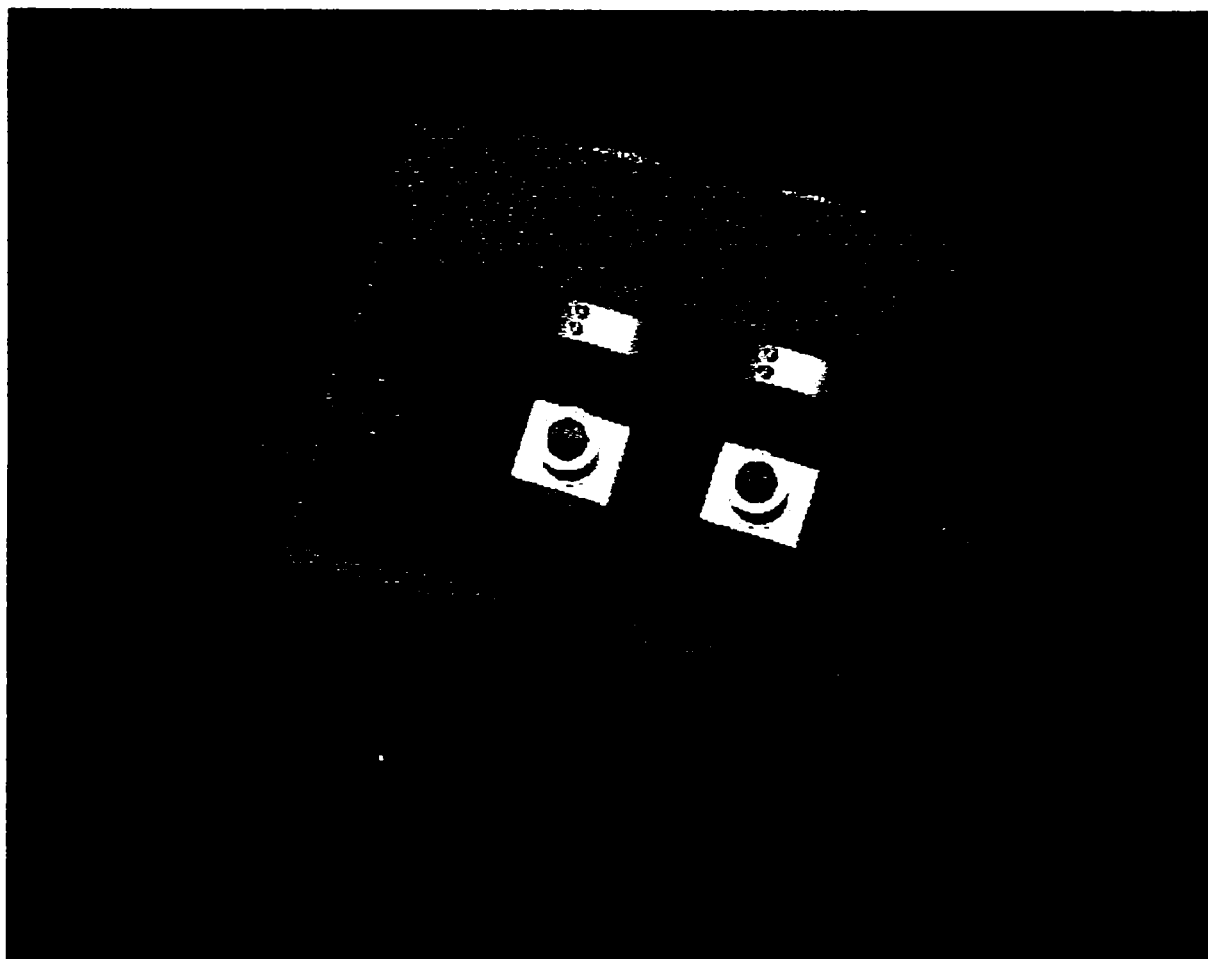


Fig.6.3. Modelare geometrică 3D a subsistemului vizual de urmărire – vedere din față
(subsistemul urmărește o țintă amplasată în stânga-sus)

O facilitate oferită de program și care a fost utilizată pentru realizarea modelului o constituie uneltele de analiză esențiale, care se referă la câteva aspecte importante în proiectare: calculul volumelor, densităților, centrelor de greutate, maselor și momentelor de inerție masice pentru solidele compuse atât din primitive 3D cât și a volumelor obținute prin revoluție sau extrudare din regiuni 2D.

Fiecărui reper îi sunt specificate proprietățile fizice (densitate, material, culoare) pentru a respecta cât mai fidel modelul real, după care se realizează asamblarea dispozitivului. Având specificate proprietățile tuturor elementelor componente, cu ajutorul funcției *wisnecg* se determină centrul de greutate și momentele de inerție masice ale componentelor constitutive ale modelului 3D al sistemului robot.

La sfârșitul modelării fișierul se eliberează de elementele ajutoare asamblării și de liniile ascunse prin utilizarea funcției *clean model* (*point*, *line*, *curve*, *unused ditto*, *unused details*, etc.).

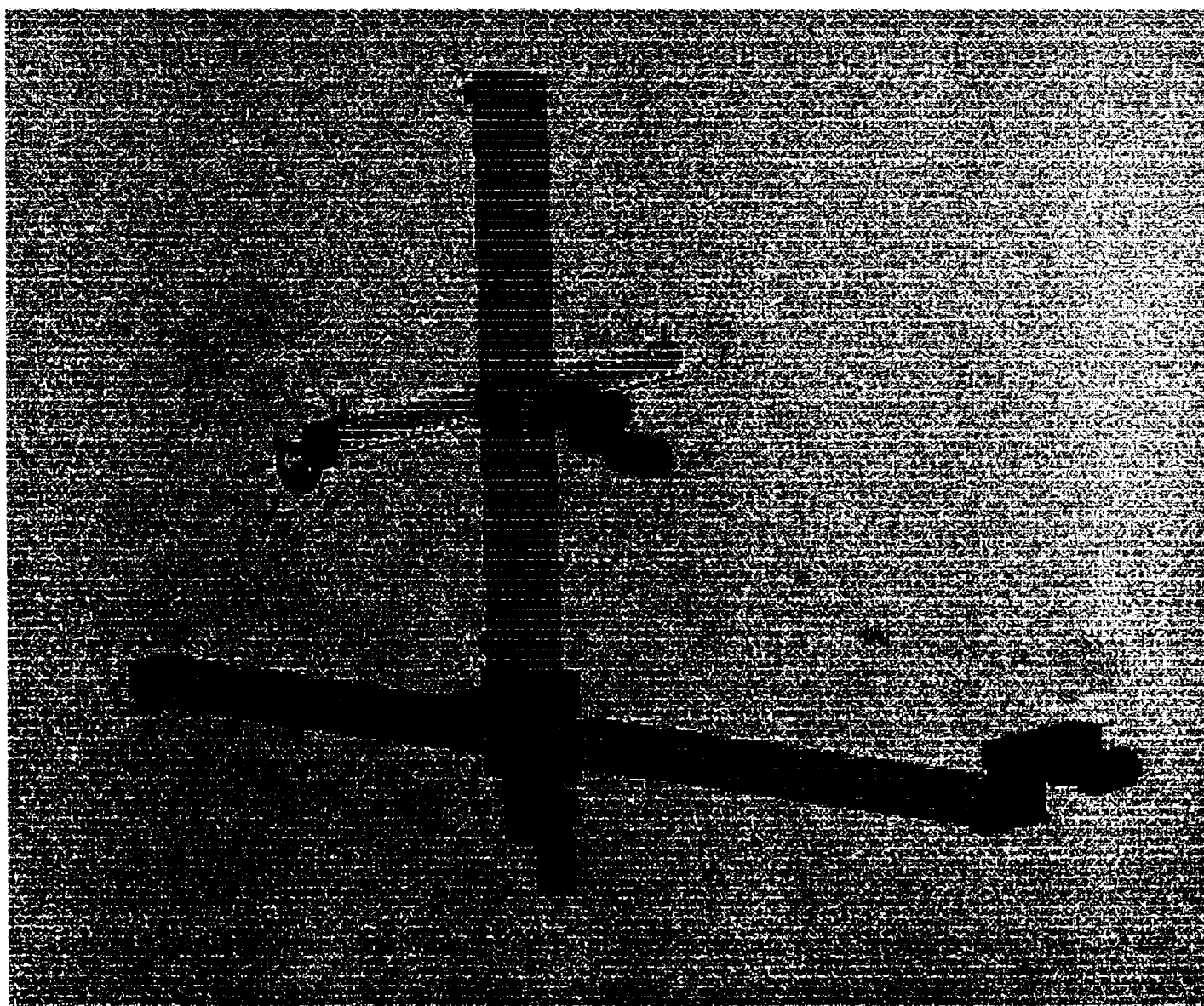


Fig.6.4. Modelare geometrică 3D a robotului

..

Modelarea geometrică 3D a fiecărui subsistem din cadrul sistemului robot astfel realizată se prezintă în figurile de mai jos.

În figurile 6.1, 6.2 și respectiv 6.3 se prezintă trei solide obținute în urma modelării geometrice 3D a subsistemului vizual de urmărire, astfel: vederea din față a sistemului vizual de urmărire în figura 6.1, în figura 6.2 o vedere din spate a subsistemului vizual de urmărire și respectiv în figura 6.3 o vedere din față în cazul în care sistemul urmărește o țintă plasată în stânga-sus a direcției sale de vizare.

Modelarea geometrică 3D a robotului se prezintă în figura 6.4. Se observă mecanismul generator de traiectorie al robotului de tip TTT (cartezian).



Fig.6.5. Modelare geometrică 3D a sistemului robot – vedere din lateral stânga spate după direcția de vizare a subsistemului vizual de urmărire

Modelul este salvat în vederea corespunzătoare proiecției izometrice după care este realizat desenul de ansamblu 2D (*drawing*), respectiv secțiunile (prin alegerea traseelor planelor de secționare) și desenele de execuție, prin extragerea componentelor necesare din desenul de ansamblu.

Vederile se realizează în modul de lucru *drawspaces* cu ajutorul funcției *auxview*.

Pe vederile astfel obținute sunt indicate toate reperele după care este creat și completat tabelul de componență. Vederile subsistemului vizual de urmărire sunt prezentate în capitolul opt.

În final se listează și se verifică desenele și modelul astfel obținut, se corectează eventualele greșeli și după remedierea lor se salvează fișierul aferent modelului 3D al sistemului robot.

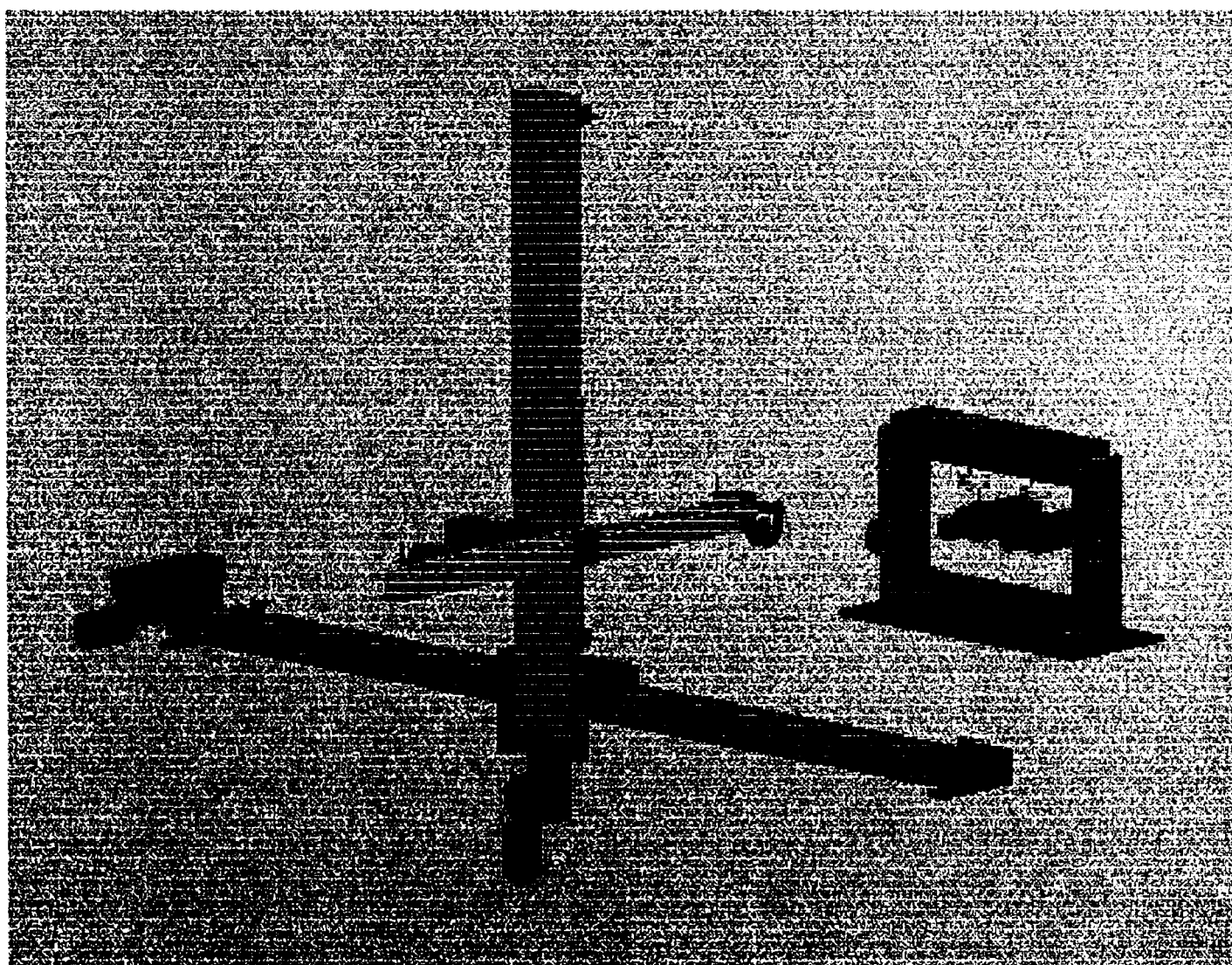


Fig.6.6. Modelare geometrică 3D a sistemului robot – vedere din lateral stânga față după direcția de vizare a subsistemului vizual de urmărire

direcția de vizare a subsistemului vizual de urmărire, iar în figura 6.6 o vedere din lateral stânga față, după direcția de vizare a subsistemului vizual de urmărire.

Programul permite setarea originii sistemului de unde se vizualizează scena. Utilizând această facilitate se pot obține diferite vederi ale aceleiași scene observate.

6.2.3. Etapele de parcurs în vederea realizării simulării prin animație în 3D a mișcărilor sistemului robot

Pentru realizarea simulării funcționării sistemului prin animație este necesară atât realizarea simulării mișcărilor robotului, cât și realizarea simulării mișcărilor subsistemului vizual de urmărire, astfel încât senzorii vizuali să urmărească poziția punctului caracteristic al efectorului final al robotului.

Se consideră legea de mișcare a punctului caracteristic al efectorului final sub următoarea formă:

$$\begin{cases} f_1(x,t) = 0 \\ f_2(y,t) = 0 \\ f_3(z,t) = 0 \end{cases} \quad (6.1)$$

unde (x, y, z) sunt parametrii generalizați ai mișcărilor relative ale elementelor cuplelor cinematice conducătoare ale dispozitivului de ghidare al robotului.

Din relațiile (6.1) se determină modul de variație în timp a parametrilor generalizați ai mișcărilor relative ale elementelor cuplelor cinematice conducătoare ale dispozitivului de ghidare al robotului.

Deoarece direcțiile de vizare ale celor doi senzori vizuali trebuie să fie orientate spre punctul caracteristic aferent efectorului final al robotului pentru ca senzorii să urmărească ținta, utilizându-se o facilitate a programului CATIA, se determină parametrii generalizați (α, β, θ) , obținuți din simulare, ai mișcărilor relative ale elementelor subsistemului vizual de urmărire.

Modul de determinare analitic a parametrilor generalizați (α, β, θ) ai mișcărilor relative ale elementelor subsistemului vizual de urmărire rezultă din condiția ca cei doi senzori vizuali să urmărească ținta conform schemei cinematice a sistemului de conducere senzorială utilizând informații de la un sistem vizual de urmărire care se prezintă în figura 6.7.

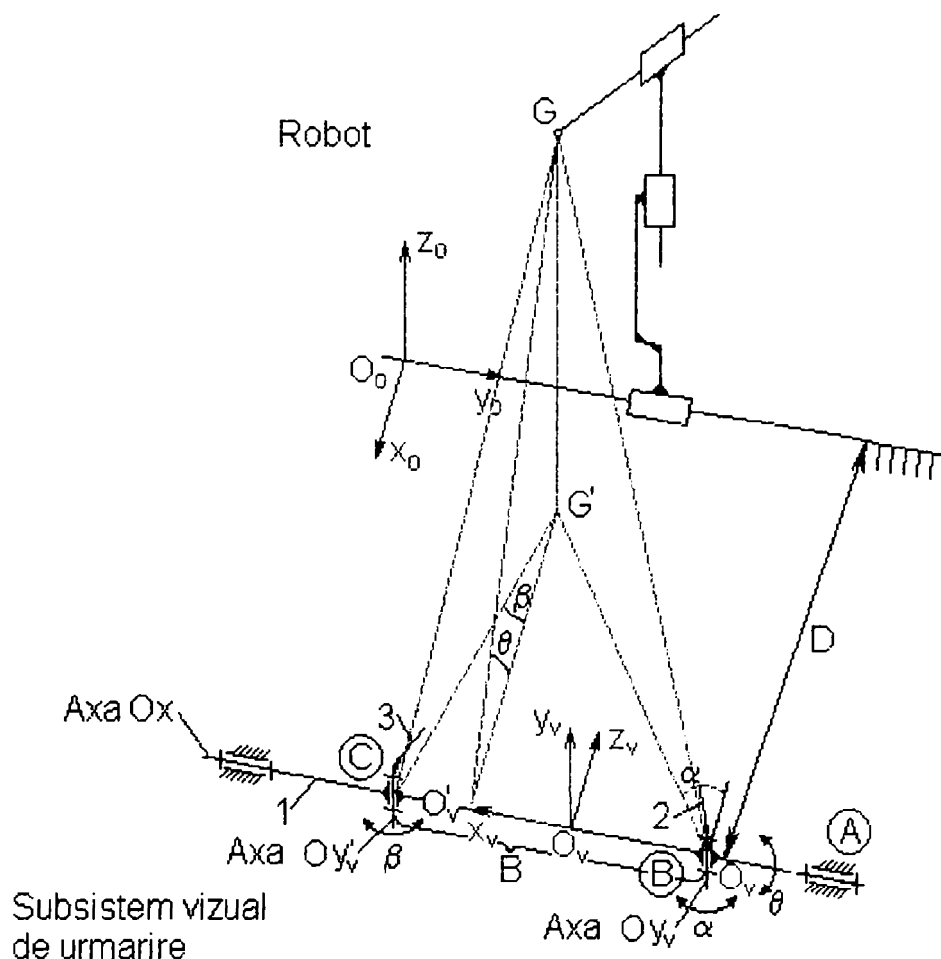


Fig.6.7. Schema cinematică a sistemului de conducere senzorială utilizând informații de la un subsistem vizual de urmărire

Prin compararea parametrilor generalizați, obținuți din simulare, ai mișcărilor relative ale elementelor subsistemului vizual de urmărire cu parametrii generalizați, rezultați din relațiile matematice, ai mișcărilor relative ale elementelor subsistemului vizual de urmărire s-a verificat corectitudinea algoritmului pentru determinarea poziției țintei stabilit în cadrul capitolului cinci.

În cazul în care subsistemul vizual de urmărire se amplasează astfel încât originea O_v a sistemului de coordonate atașat subsistemului vizual de urmărire să prezinte coordonatele $O_v(D, -d_2, 0)$ conform figurii 6.7, atunci matricea de trecere de la sistemul de coordonate atașat subsistemului vizual de urmărire la sistemul de coordonate atașat subsistemului robot va fi dată de:

$${}^{o_0}T_{O_v} = \begin{vmatrix} 0 & 0 & -1 & D \\ -1 & 0 & 0 & -d_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (6.2)$$

unde d_2 reprezintă cursa mișcării relative a elementelor celei de-a doua c.c.c. a robotului în lungul axei O_0y_0 .

În acest caz, pentru ca subsistemul vizual de urmărire să urmărească în timp real ținta, parametrilor generalizați (α, β, θ) trebuie să fie cei dați de relațiile:

$$\begin{aligned}\alpha &= \operatorname{arctg} \left(\frac{{}^{0_0}y_G - \frac{c_1 - B}{2}}{D - {}^{0_0}x_G} \right) \\ \beta &= \operatorname{arctg} \left(\frac{{}^{0_0}y_G - \frac{c_1 + B}{2}}{D - {}^{0_0}x_G} \right) \\ \theta &= \operatorname{arctg} \left(\frac{{}^{0_0}z_G}{D - {}^{0_0}x_G} \right)\end{aligned}\tag{6.3}$$

Etapele parcurse în vederea realizării simulării funcționării robotului sunt următoarele:

- se împarte traiectoria pe care trebuie să o parcurgă punctul caracteristic atașat EF al robotului într-un număr convenabil de segmente egale; fie n numărul de puncte intermediare alese, numite în continuare puncte via;
- pentru fiecare punct via i se calculează parametrii generalizați utilizând relațiile (6.1);
- având aceste date de intrare se realizează câte un model geometric 3D pentru sistem pentru fiecare punct via, cât și pentru punctul inițial și final al traiectoriei descrise de țintă;
- se salvează imagine cu imagine modelul astfel obținut, imaginile urmând a constitui câte un cadru în ansamblul care modelează cinematic subsistemul vizual de urmărire;
- pentru fiecare punct via atins de țintă, se determină parametrii generalizați ai subsistemului vizual de urmărire utilizând în acest sens facilitățile programului CATIA;
- pentru fiecare punct via atins de țintă, se determină parametrii generalizați ai subsistemului vizual de urmărire utilizând relațiile (6.3).
- prin asamblarea tuturor imaginilor obținute într-un program de animație (în cazul de față rularea se realizează utilizând utilitarul Windows Media Player) se obține simularea în 3D a mișcărilor sistemului robot.

6.3. Exemplificarea simulării prin animație a funcționării sistemului robot pentru o lege de mișcare dată a punctului caracteristic al efectorului final

Pentru exemplificarea simulării prin animație a funcționării sistemului robot se va porni de la stabilirea unei legi de mișcare pentru punctul caracteristic aferent efectorului final al robotului.

Deoarece simularea prin animație a funcționării sistemului robot se referă doar la urmărirea mișcărilor efectuate de punctul caracteristic al efectorului final al robotului (deci doar urmărirea poziției efectorului final), s-a acceptat o simplificare a schemei cinematice a robotului prezentată în figura 5.20 păstrându-se doar MGT al acestuia.

Se propune următoarea lege de mișcare de realizat (ecuația legii de mișcare este exprimată în sistemul de referință legat de baza robotului) pentru punctul caracteristic aferent efectorului final al robotului:

$$\begin{cases} \frac{|x(t) - a - c_3|}{c_3} = \frac{|2 \cdot y(t) - c_1|}{c_1} = \frac{|2 \cdot [z(t) - b] - c_2|}{c_2} \\ y(t) = \frac{c_1}{t_1} \cdot t \end{cases} \quad (6.4)$$

unde: - c_1, c_2, c_3 reprezintă cursele programate de realizat ale mișcărilor relative ale elementelor celor trei cuple cinematice conducătoare ale mecanismului generator de traiectorie al subsistemului robot;

- t_1 durata desfășurării mișcării.

Relația (6.4) se explicitează sub următoarea formă:

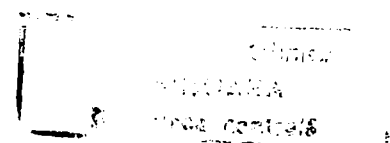
$$\begin{aligned} \frac{x - c_3 - a}{c_3} = \frac{2 \cdot y - c_1}{c_1} = \frac{2 \cdot (z - b) - c_2}{c_2}, \quad \text{daca } y < \frac{c_1}{2} \\ \frac{x - c_3 - a}{-c_3} = \frac{2 \cdot y - c_1}{c_1} = \frac{2 \cdot (z - b) - c_2}{c_2}, \quad \text{daca } y \geq \frac{c_1}{2} \end{aligned} \quad (6.5)$$

$$y(t) = \frac{c_1}{t_1} \cdot t$$

Relațiile (6.4), (6.5) descriu în spațiu două segmente de dreaptă conform figurii 6.8.

Pentru exemplificare se aleg următoarele date ca mărimi de intrare:

- $c_1 = 1200$ [mm], $c_2 = 800$ [mm], $c_3 = 500$ [mm] cursele programate de realizat ale mișcărilor relative ale elementelor celor trei cuple cinematice conducătoare a mecanismului generator de traiectorie al subsistemului robot;



- $a = 100$ [mm], $b = 100$ [mm] coordonatele inițiale pentru mișcările relative ale elementelor c.c.c. ale robotului;
- $D = 1500$ [mm] distanța dintre axa $O_v x_v$ și $O_0 y_0$;
- $B = 200$ [mm] distanța dintre centrele optice ale celor doi senzori vizuali;
- $n = 30$ de puncte via.

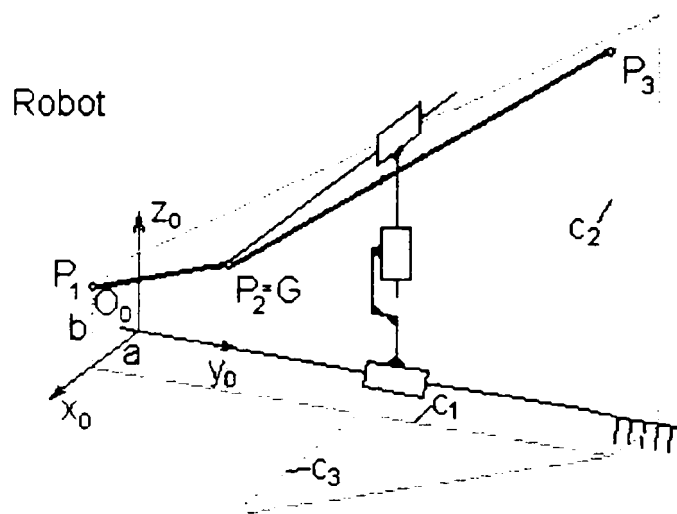


Fig. 6.8. Traiectoria programată pentru punctul caracteristic al EF

Pentru aceste valori în anexa A1 se prezintă rezultatele obținute, utilizând relațiile (6.3) și respectiv (6.5), pentru coordonatele generalizate ale robotului și ale subsistemului vizual de urmărire. Cele trei unghiuri sunt date atât în radiani cât și în grade.

6.4. Concluzii

Simularea funcționării prin animație a sistemului de conducere senzorială a robotului care utilizează informații furnizate de un sistem vizual de urmărire furnizează o imagine concretă despre modul de funcționare a sistemului robot.

Din compararea parametrilor generalizați obținuți din simulare ai mișcărilor relative ale elementelor subsistemului vizual de urmărire cu parametrii generalizați rezultați din relațiile matematice ai mișcărilor relative ale elementelor subsistemului vizual de urmărire s-a verificat corectitudinea algoritmului pentru determinarea poziției țintei stabilit în cadrul capitolului cinci.

Fișierul simulare.mpg cu simularea mișcărilor sistemului robot se află pe CD-ul anexat la teză.

CAPITOLUL 7 Modelarea funcțională a sistemului de conducere senzorială conceput și simularea funcționării lui

7.1. Scopul modelării și simulării funcționale a sistemului

În cadrul capitolului se prezintă modelarea funcțională a sistemului de conducere senzorială a robotului conceput și simularea funcționării lui.

Scopul modelării și simulării funcționale a sistemului robot este acela de a prevedea funcționarea acestuia, înainte de a trece la realizarea fizică a sistemului.

Sistemul de conducere senzorială conceput este un sistem complex, format din componente mecanice, electrice și de calcul care interacționează în timpul funcționării, componente care își modifică stările în mod dinamic, fiind totodată eterogene din punct de vedere al modului de funcționare, unele dintre ele prezentând mărimi de stare continue, altele având mărimi de stare discrete, ca urmare, pentru atingerea scopului simulării s-a necesitat modelarea funcțională și simularea funcționării fiecărei componente a sistemului de conducere senzorială conceput.

Prin simularea funcționării fiecărei componente se dorește să se prevadă modul de funcționare a fiecărei componente, astfel încât să se poată trage concluzii cu privire la:

- stabilirea timpului de achiziție a informației de către subsistemul vizual de urmărire;
- stabilirea timpilor de execuție a comenzilor;
- stabilirea timpului de răspuns al sistemului de calcul,

toate acestea cu scopul de a putea trage concluzia, dacă sistemul robot funcționează în timp real sau nu.

Pentru a modela comportamentul sistemului robot este necesară utilizarea unui program care posedă capacitatea redării comportării dinamice a subsistemelor componente ale acestuia, iar programul Simulink care este parte integrantă a mediului de dezvoltare se aplicații soft MATLAB al firmei MATHWORKS posedă toate atributele necesare în vederea modelării, simulării sistemelor ale căror stări sunt variabile în timp.

7.2. Modelarea funcțională a componentelor sistemului de conducere senzorială a robotului

7.2.1. Modelarea funcțională a motorului de curent continuu

Modelul matematic al motorului de curent continuu a fost prezentat în paragraful 3.3.4.1.

Pornind de la relațiile (3.42) – (3.56), se proiectează modelul funcțional al motorului de curent continuu. Acesta este compus din două funcții de transfer notate "Electric" și respectiv "Sarcina", care exprimă transformatele Laplace ale ecuațiilor diferențiale care descriu comportamentul motorului. Mărimea de ieșire din blocul "Sarcina" este viteza unghiulară a axului motorului, mărime care prin integrare se transformă în parametru generalizat la nivelul c.c.c. respective. Mărimea de intrare în blocul "Electric" este tensiunea de alimentare a motorului.

Frecările care iau naștere în timpul mișcării motorului sunt modelate cu ajutorul blocului "Frecare uscată și vâscoasă" observându-se dependența mărimilor de intrare ale acestuia de viteză. Schema bloc a modelului motorului electric de curent continuu este prezentată în figura 7.1.

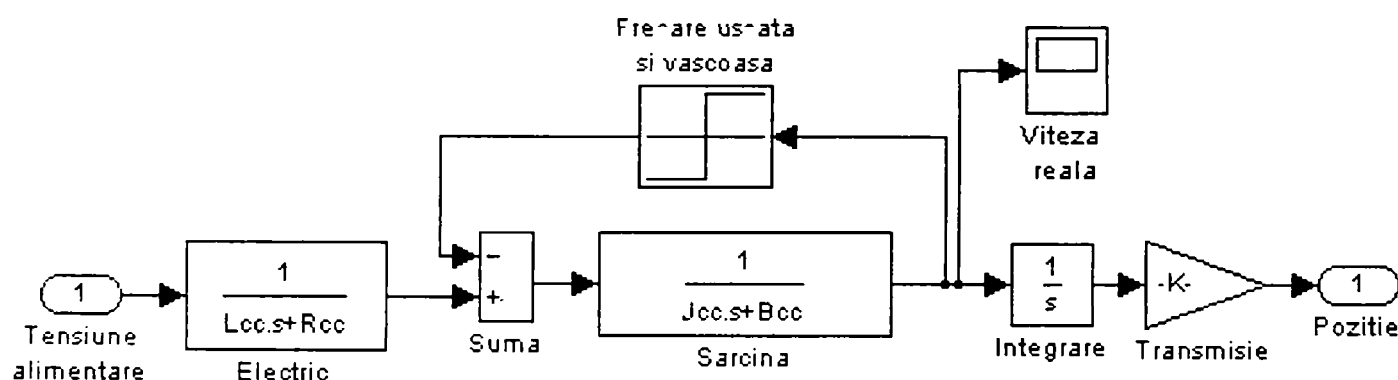


Fig. 7.1. Schema bloc a modelului motorului de curent continuu

Componentele modelului motorului de curent continuu sunt următoarele :

- "Electric" - reprezintă funcția de transfer a înfășurărilor motorului având ca mărimi caracteristice inductanța L și respectiv rezistența R a înfășurării. Pentru cazul analizat în această lucrare acestea au următoarele valori : $L = 0,0015$ [mH], $R = 1,8$ [Ω]. Mărimea de intrare a acestui bloc este tensiunea de alimentare în Volți iar mărimea de ieșire este momentul în [Nm], modalitatea de măsurare a acestor mărimi fiind prezentată în capitolul 8.

- b. "Sarcina" – reprezintă funcția de transfer a încărcării motorului electric cu un anumit moment rezistent. Calculul mărimilor parametrilor acestui bloc se prezintă în capitolul 8. Parametrii caracteristici ai acestui bloc au valorile : momentul de inerție $J = 0,99 \times 10^{-6}$ [kg m²], coeficientul global de frecare vâscoasă $B = 4,57 \times 10^{-6}$ [Nms/rad]. Mărime de intrare a acestui bloc este momentul [Nm] iar mărimea de ieșire este viteza unghiulară în [rad/s].
- c. "Frecare uscată și vâscoasă" - în acest model s-a ținut cont de frecarea vâscoasă care este dependentă de viteză și respectiv de frecarea uscată. Determinarea parametrilor acestui bloc se descrie în anexa A2. Mărimea de intrare a acestui bloc este viteza unghiulară iar mărimea de ieșire este momentul de frecare care se scade din momentul util.
- d. "Integrare " – reprezintă blocul care realizează integrarea vitezei unghiulare pentru a obține parametrul generalizat la nivelul c.c.c. respective [rad].
- e. "Suma" - bloc care realizează însumarea a două mărimi.
- f. "Tensiune de alimentare" - mărimea de intrare la motor [V].
- g. "Poziție" - poziția unghiulară a arborelui motorului [rad].
- h. "Viteza reala" – viteza unghiulară a arborelui motorului [rad/s].

7.2.2. Modelarea funcțională a motorului pas cu pas

Deplasările unghiulare ale senzorului vizual sunt realizate cu ajutorul unor motoare pas cu pas. Modelarea matematică a unui motor pas cu pas a fost prezentată în pragraful 3.3.4.2.

Motoarele folosite în instalația experimentală însă sunt motoare pas cu pas cu 4 faze a cărei schemă de principiu este prezentată în figura 7.2.

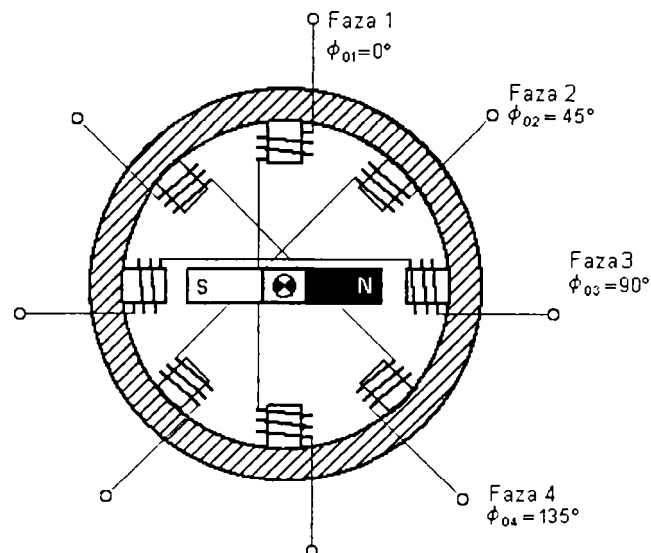


Fig. 7.2. Schema de principiu a unui motor pas cu pas cu 4 faze

figura 7.4. În această schemă bloc s-au mai reprezentat în afară de elementele deja cunoscute :

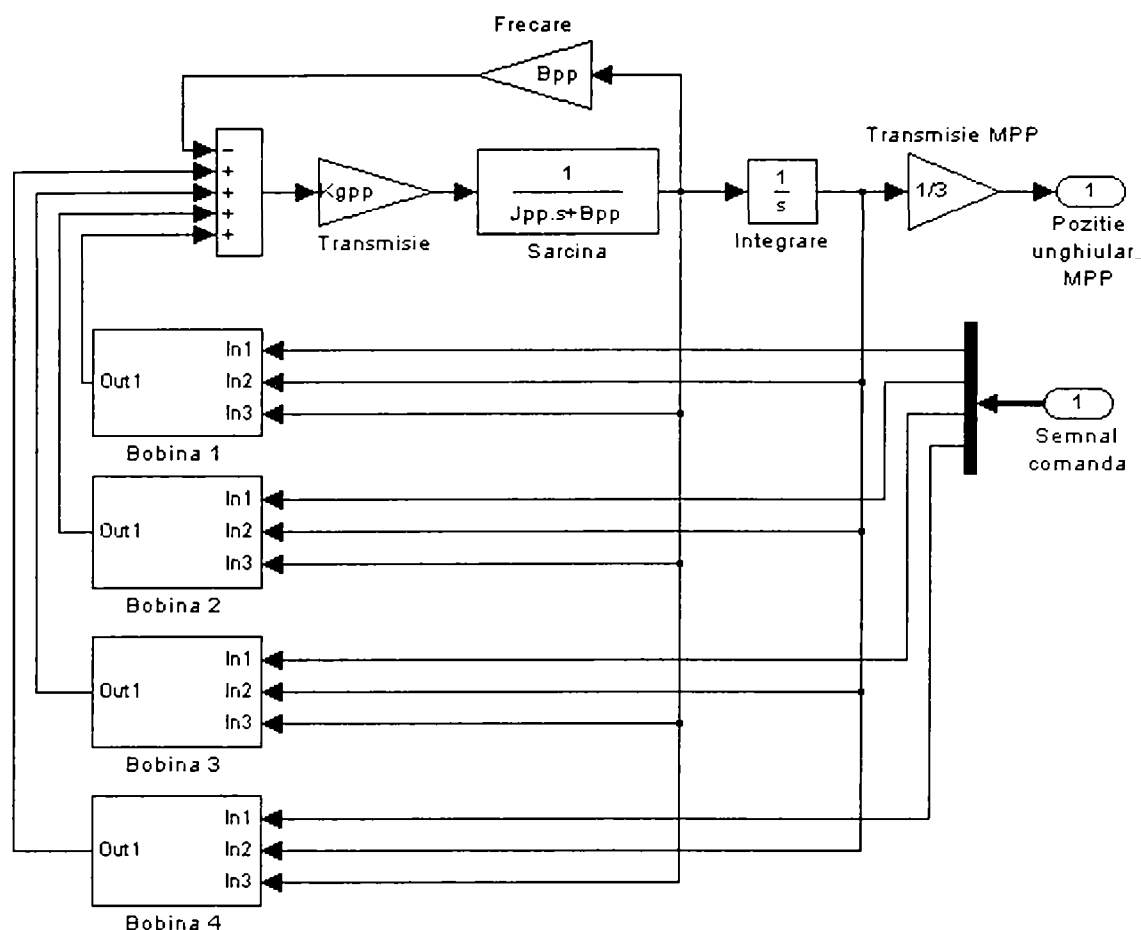


Fig. 7.4. Schema bloc a modelului motorului pas cu pas

- “Sarcina” – funcția de transfer care modelează comportamentul încărcării arborelui motorului, din care fac parte momentul de inerție J și coeficientul de frecare uscată B , a căror determinare se prezintă în capitolul 8.
- “Frecare” funcția de transfer care modelează frecarea vâscoasă.
- “Integrare” – bloc integrator care are același rol ca și în cazul motorului de curent continuu.
- “Pozitia unghiulara MPP” – mărimea de ieșire a motorului pas cu pas [rad].
- “Semnal comandă ” – este un port care transmite impulsurile de comandă ale înfășurărilor, care sunt defazate pentru fiecare înfășurare în parte. Impulsurile de comandă pot fi generate de către un subsistem după cum este arătat în cele ce urmează (paragraful 7.2.4.), sau pot fi folosite blocuri generatoare separate pentru fiecare înfășurare.
- “Bobina ‘n’” – unde ‘n’ reprezintă numărul bobinei, este subsistemul creat din modelul prezentat în figura 7.3.

- g. "Transmisie" - reprezintă modulul mecanic al motorului pas cu pas.
- h. "Transmisie MPP" – reprezintă un reductor cu roți dințate cu o singură treaptă, cu raportul 1/3.

7.2.3. Modelarea funcțională a senzorului vizual

Pentru realizarea modelului senzorului vizual s-au avut în vedere două aspecte majore :

- transformarea perspectivă;
- caracterul discontinuu al achiziției de imagini.

Pentru definirea modelului transformării perspective se definesc următoarele mărimi caracteristice :

- distanța de la focarul lentilei senzorului vizual la planul punctului caracteristic al efectorului final al robotului : "D" (în modelul funcțional : "dist");
- unghiul dintre poziția momentană a axei optice a senzorului vizual și perpendiculara dusă din focarul senzorului vizual la direcția axei robotului "α" (în modelul funcțional se notează "alfa");
- distanța de la focarul lentilei la planul senzorului CCD : "f" (în modelul funcțional: "dist_focal");
- poziția curentă a punctului caracteristic față de un reper absolut (piciorul perpendicularei din focarul senzorului pe planul punctului caracteristic): x_r' ;
- poziția curentă a intersecției axei optice a senzorului cu planul punctului caracteristic : x_{opt} ;
- poziția curentă a punctului caracteristic față de intersecția axei optice a senzorului cu planul punctului caracteristic x_r (în schema bloc din figura 7.7, respectiv în relația (7.6) mărimea s-a notat cu "u");
- poziția curentă a punctului caracteristic față de axa optică în planul imagine x_p (în schema din figura 7.12 respectiv relația 7.6.: "u");

Odată aceste mărimi definite și folosind schema din figura 7.5, se deduc expresiile

$$B = \frac{D}{\cos \alpha} \quad (7.1.)$$

$$x_{opt} = D \cdot \operatorname{tg} \alpha \quad (7.2)$$

$$x_r' = D \cdot \operatorname{tg}(\alpha + \theta) \quad (7.3)$$

$$x_r = x_r' - x_{opt} = D \cdot \left[\operatorname{tg} \left(\alpha + \operatorname{arctg} \frac{x_p}{f} \right) - \operatorname{tg} \alpha \right] \quad (7.4)$$

$$x_p = f \cdot \operatorname{tg} \left(\operatorname{arctg} \left(\frac{x_r}{D} + \operatorname{tg} \alpha \right) - \alpha \right) \quad (7.5)$$

Aspectele teoretice ale transformării de perspectivă s-au tratat în paragraful 4.2.1.

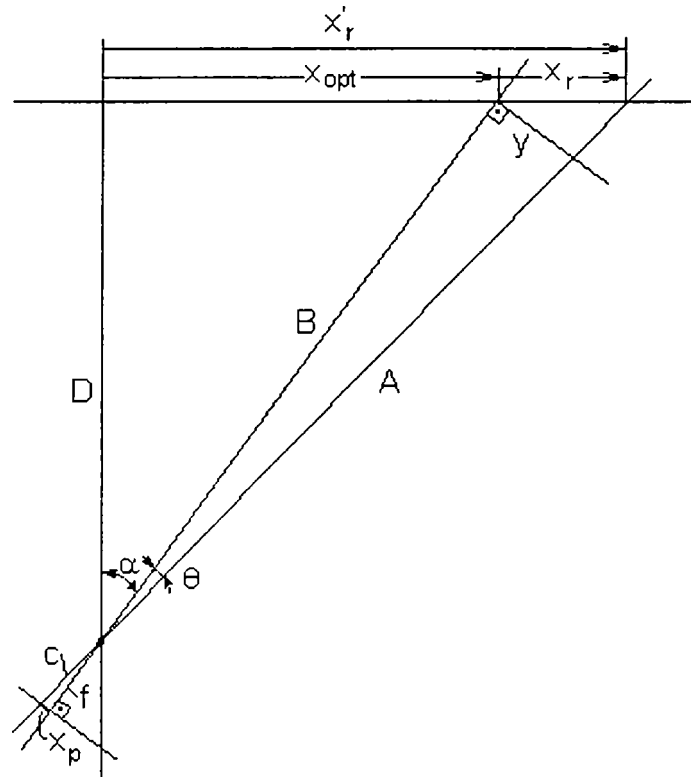


Fig. 7.5. Schema poziției punctului caracteristic și corelarea ei cu imaginea achiziționată de senzorul vizual.

Poziția unghiulară a senzorului vizual este modificată de către motorul pas cu pas în momentul în care imaginea punctului caracteristic depășește limitele "dreptunghiului de interes". Acest fapt se poate reprezenta grafic după cum se arată în figura 7.6.

Fie momentul inițial 1 atunci când imaginea țintei în planul imagine va fi amplasată în centrul acesteia, iar direcția de vizare a senzorului va intersecta planul de mișcare a țintei într-un punct a cărui ordonată este x_1 . La deplasarea țintei spre stânga în planul imagine se va observa o deplasare spre dreapta a imaginii țintei. Atât timp cât imaginea țintei se găsește în dreptunghiul de interes, cu ajutorul relațiilor (7.2), (7.3) se calculează poziția curentă a punctului caracteristic față de intersecția axei optice a senzorului cu planul punctului caracteristic (caz prezentat în imaginea 2 a figurii 7.6.). Dacă imaginea țintei depășește perimetrul dreptunghiului de interes se va comanda motorului pas cu pas execuția unei mișcări în vederea readucerii imaginii țintei în zona dreptunghiului de interes

(caz prezentat în imaginea 3 a figurii 7.6.). Algoritmul se repetă de fiecare dată calculându-se utilizând relațiile (7.2), (7.3), poziția curentă a punctului caracteristic față de intersecția axei optice a senzorului cu planul punctului caracteristic

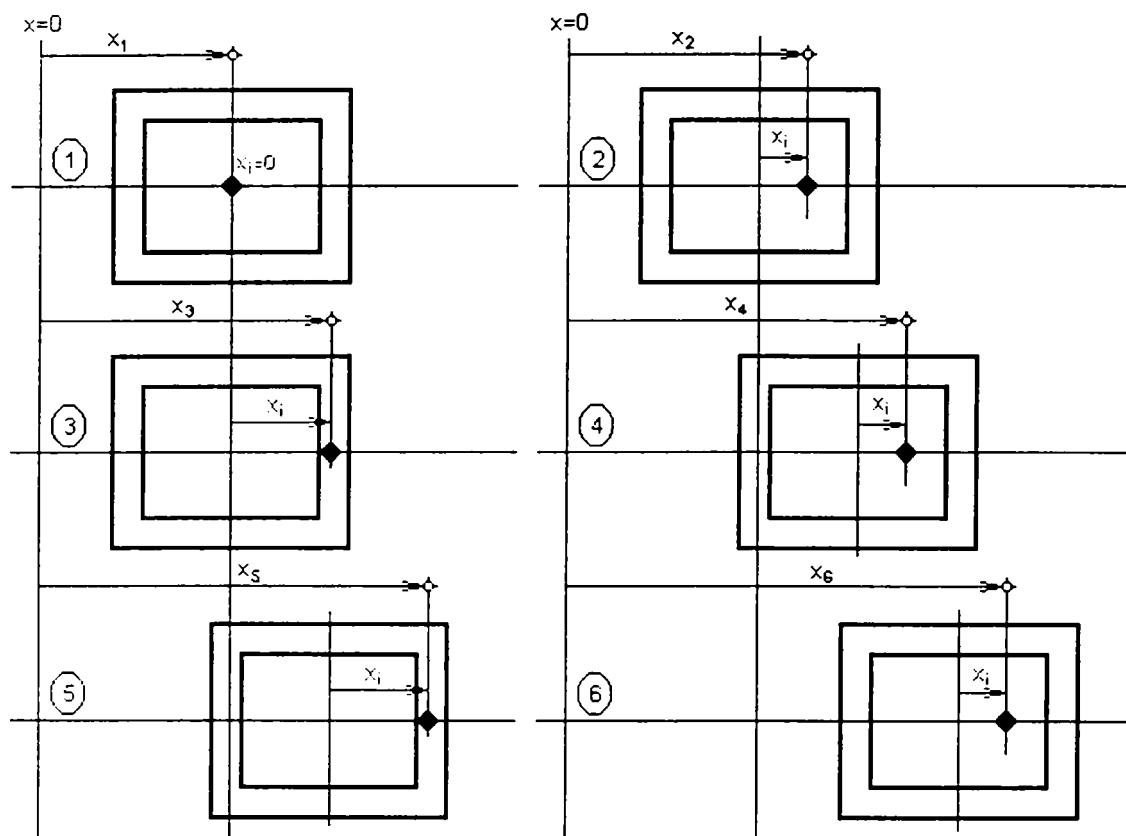


Fig. 7.6. Schema poziției punctului caracteristic și corelarea ei cu imaginea achiziționată de senzorul vizual.

Modelul funcțional al senzorului vizual este arătat în figura 7.7.

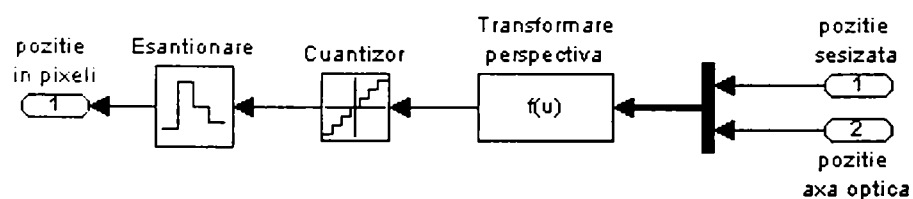


Fig. 7.7. Schema bloc a modelului funcțional al senzorului vizual.

Blocurile componente sunt:

- „Poziție sesizată” – port de intrare care transmite evoluția poziției punctului caracteristic.
- „Poziție axa optică” – port de intrare care transmite evoluția poziției intersecției axei optice cu direcția de deplasare a axei robotului.

- c. „Transformare perspectivă” – bloc de calcul algebric care folosește relația definită anterior și a cărei expresie în limbajul de simulare este :
- $$u = \tan(\operatorname{atan}(u/\operatorname{dist} + \tan(\alpha)) - \alpha) * \operatorname{dist_focal} * n_pixeli \quad (7.6)$$
- în care - “u” reprezintă mărimea de intrare a blocului ($u = x_r$)
- d. “Eșantionare” – reprezintă un bloc care modelează caracteristica discretă a plăcii de achiziție de imagini, și anume faptul că fiecare imagine este prelevată la o rată de 28 de cadre pe secundă, între două achiziții consecutive păstrându-se imagine anterioară; se face observația că rata maximă de achiziție a plăcii de achiziție pentru un format de tip 640x480x8 este de 30 de cadre/ secundă.
- e. “Cuantizor” – reprezintă un bloc de cuantificare a mărimii x_r definită la începutul acestui paragraf, care este o mărime reală într-un număr întreg care reprezintă numărul de pixeli corespunzător acelei mărimi.
- f. “Poziție în pixeli” – reprezintă portul de ieșire al subsistemului care transferă mărimea x_p definită la începutul acestui paragraf, subsistemului de comandă.

7.2.4. Modelarea funcțională a sistemului de calcul

Sistemul de calcul generează mărimile de comandă pentru motoarele de acționare a robotului și pentru motoarele de poziționare a subsistemului vizual de urmărire.

Modelul sistemului de calcul este arătat în figura 7.8.

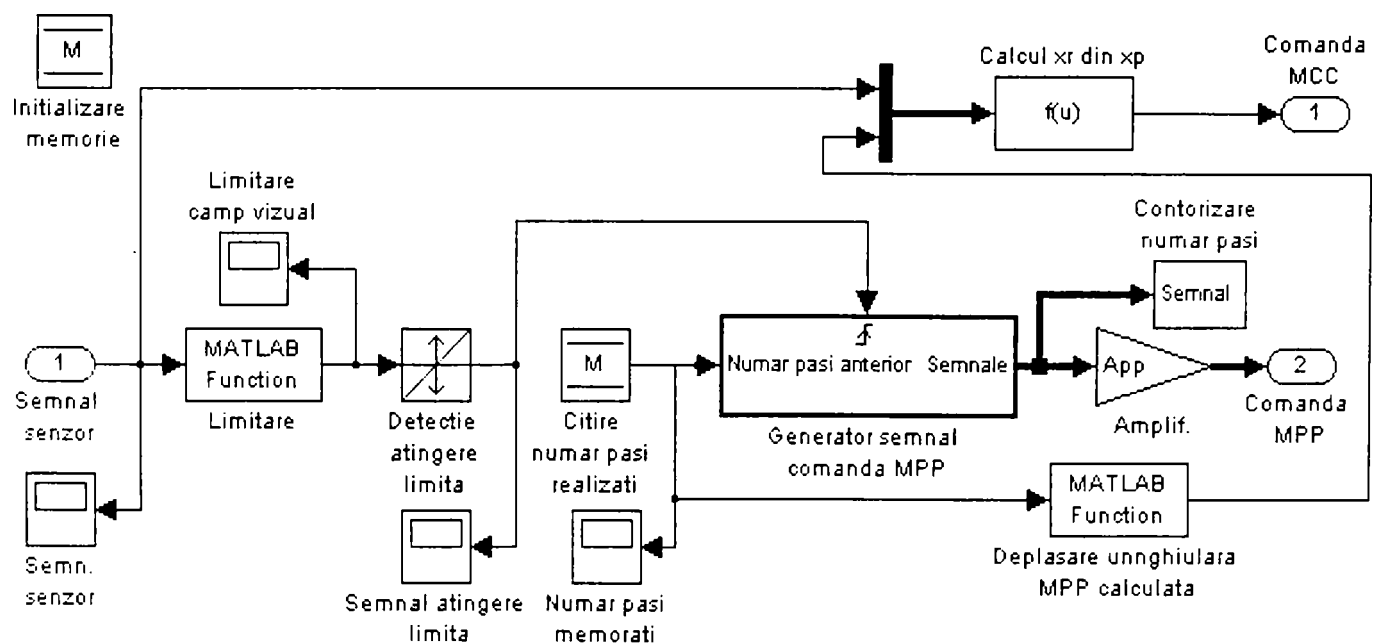


Fig. 7.8. Schema bloc a modelului sistemului de calcul.

În figura 7.8 se observă blocurile de afișare grafică “Semn. senzor”, “Semnal atingere limită” care au rolul de a afișa semnalele pe care le primesc pentru a se putea urmări evoluția simulării.

Mărimile de comandă amintite sunt calculate folosind ca mărime de intrare mărimea x_p dată de către senzorul vizual.

Semnalul de comandă al motorului pas cu pas este generat, după cum s-a arătat și în figura 7.7, la depășirea limitei dreptunghiului de interes aferent planul imagine. În cadrul modelului aceasta se realizează cu ajutorul blocurilor :

- “Inițializare memorie” – definește un spațiu de memorie în care se înscrie și din care se citește numărul de pași realizați de către motorul pas cu pas;
- “Semnal senzor” – portul de intrare al subsistemului;
- “Limitare” – care conține relația :

$$u(1)-x_cadru*(u(2)+1) \quad (7.7)$$

în care :- x_cadru reprezintă poziția marginii cadrului de limitare față de centrul imaginii în pixeli,

- $u(1)$ reprezintă mărimea de intrare a blocului de la senzorul vizual;
 - $u(2)$ reprezintă numărul de pași executați de MPP.
- “Detectie atingere limită” este blocul care declanșează funcționarea blocului “Generator semnal comandă MPP” în momentul atingerii limitei descrise.
 - “Citire număr pași realizați” – citirea din memorie a numărului de pași care au fost deja realizați de către MPP.
 - “Generator semnal comandă MPP” – este un subsistem a cărui structură este arătată în figura 7.9.

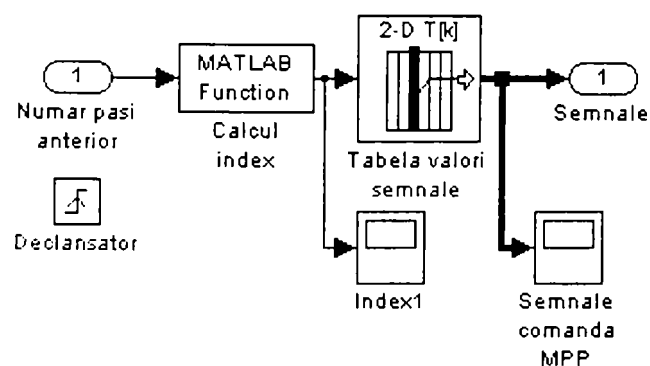


Fig. 7.9. Subsistemul pentru generarea semnalelor de comandă a MPP

Acest subsistem este declanșat de către blocul "Declanșator" în condițiile arătate și primește semnalul de intrare de la portul "Număr pași anterior". Semnalul de intrare se incrementează cu 1 după care îi este aplicată funcția :

$$\text{mod}(u+1,8) \quad (7.8)$$

care reprezintă restul împărțirii cu 8 a semnalului incrementat, calculându-se astfel indexul coloanei tabelului memorate în blocul "Tabela valori semnale". Valorile memorate în acest bloc sunt arătate în tabelul 7.1.

Tabelul 7.1.Valori memorate în blocul "Tabela valori semnale"

Index	0	1	2	3	4	5	6	7
Bobina 1	-1	0	0	0	1	0	0	0
Bobina 2	0	-1	0	0	0	1	0	0
Bobina 3	0	0	-1	0	0	0	1	0
Bobina 4	0	0	0	-1	0	0	0	1

În funcție de valoarea indexului (poziția bobinei care urmează a fi acționată), se generează patru semnale, câte unul pentru fiecare bobină. Aceste semnale se transmit, prin portul denumit "Semnale", sistemului de calcul.

g. "Deplasare unghiulară MPP calculată" – este un bloc care calculează valoarea parametrului "alfa" (definit în paragraful 7.3.4.) cu ajutorul expresiei

$$\text{alfa} = \text{numar_pasi} * \text{pas_unghiular} \quad (7.9)$$

unde : - numar_pasi – numărul de pași realizat de către MPP;

- pas_unghiular – valoarea unui pas al MPP în radiani, care este constant și egal cu $2 * \pi / 200$.

Valoarea parametrului "alfa" este necesară pentru calculul mărimii de comandă a motorului de curent continuu.

h. "Contorizare număr pași" – este un subsistem care contorizează pașii realizați de către motorul pas cu pas și înscrie valoarea contorizată în memorie. Schema bloc a subsistemului de contorizare se prezintă în figura 7.10.

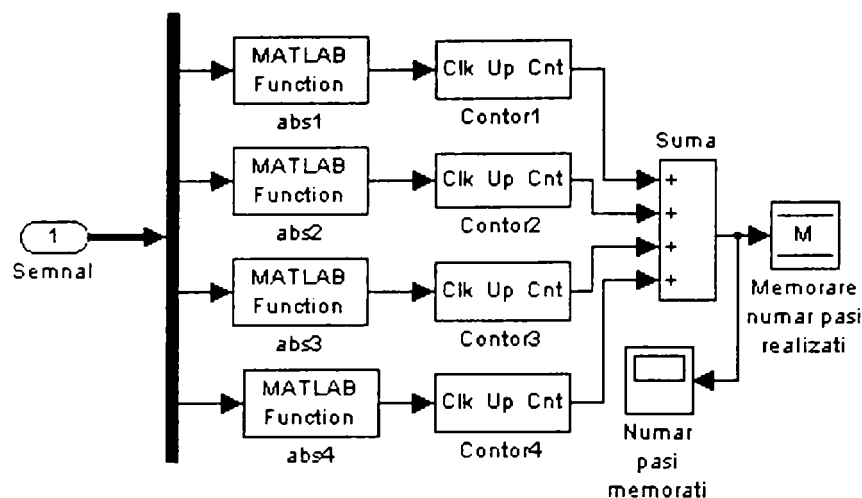


Fig. 7.10. Schema bloc a subsistemului de contorizare pași ai MPP.

Semnificația notațiilor din figura 7.10 este următoarea :

- "Semnal" port de intrare în subsistem;
 - "abs1", ..., "abs4" – returnează valoarea absolută a semnalelor pe cele 4 canale, corespunzătoare celor 4 bobine;
 - "Contor1", ..., "Contor4" – contoare având ca intrare "Clock" (Clk) și ieșire "Count" (Cnt);
 - "Suma" calculează valoarea însumată a pașilor executați pe cele 4 bobine;
 - "Memorare număr pași realizați" – înscrie valoarea contorizată în memorie.
- i. "Comanda MPP" – port de ieșire din sistemul de comandă care realizează transmiterea semnalelor generate către motorul pas cu pas;
 - j. "Calcul x_r din x_p " - componenta semnalului de comandă a motorului de curent continuu este realizată cu ajutorul expresiei (7.4.) care în limbajul de modelare se transcrie :

$$\text{dist}^*(\tan(\text{alfa} + \text{atan}(u/\text{dist_focal})) - \tan(\text{alfa})) \quad (7.10)$$

- k. "Comanda MCC" – este portul de ieșire al sistemului de comandă care transmite valoarea parametrului „ x_r ”, elementelor de comandă a acționării cu motor de curent continuu a c.c.c. ale robotului.

7.3. Modelarea funcțională în ansamblu a sistemului robot realizat

În paragrafele anterioare ale acestui subcapitol au fost prezentate modelele funcționale ale elementelor componente ale sistemului robot. Aceste elemente se integrează în schema completă a modelului.

Modelarea funcțională a sistemului robot s-a realizat atât pentru urmărirea unei ținte după o axă (figura 7.11), cât și pentru sistemul de conducere globală a robotului la care robotul prezintă MGT de tip TT (figura 7.12.). Modelul funcțional al sistemului de conducere senzorială realizat în cazul robotului cu MGT de tip TT s-a obținut pornindu-se de la modelul funcțional al sistemului de conducere senzorială realizat pentru o c.c.c., ținând cont de parametrii sistemului la nivelul fiecărei c.c.c. și de faptul că mișcările și conducerea după fiecare c.c.c. sunt independente.

Pentru a urmări comportamentul dinamic al unui sistem de comandă cu senzor vizual s-a realizat modelul funcțional al sistemului pentru o c.c.c. a robotului. În acest model se regăsesc elementele descrise anterior.

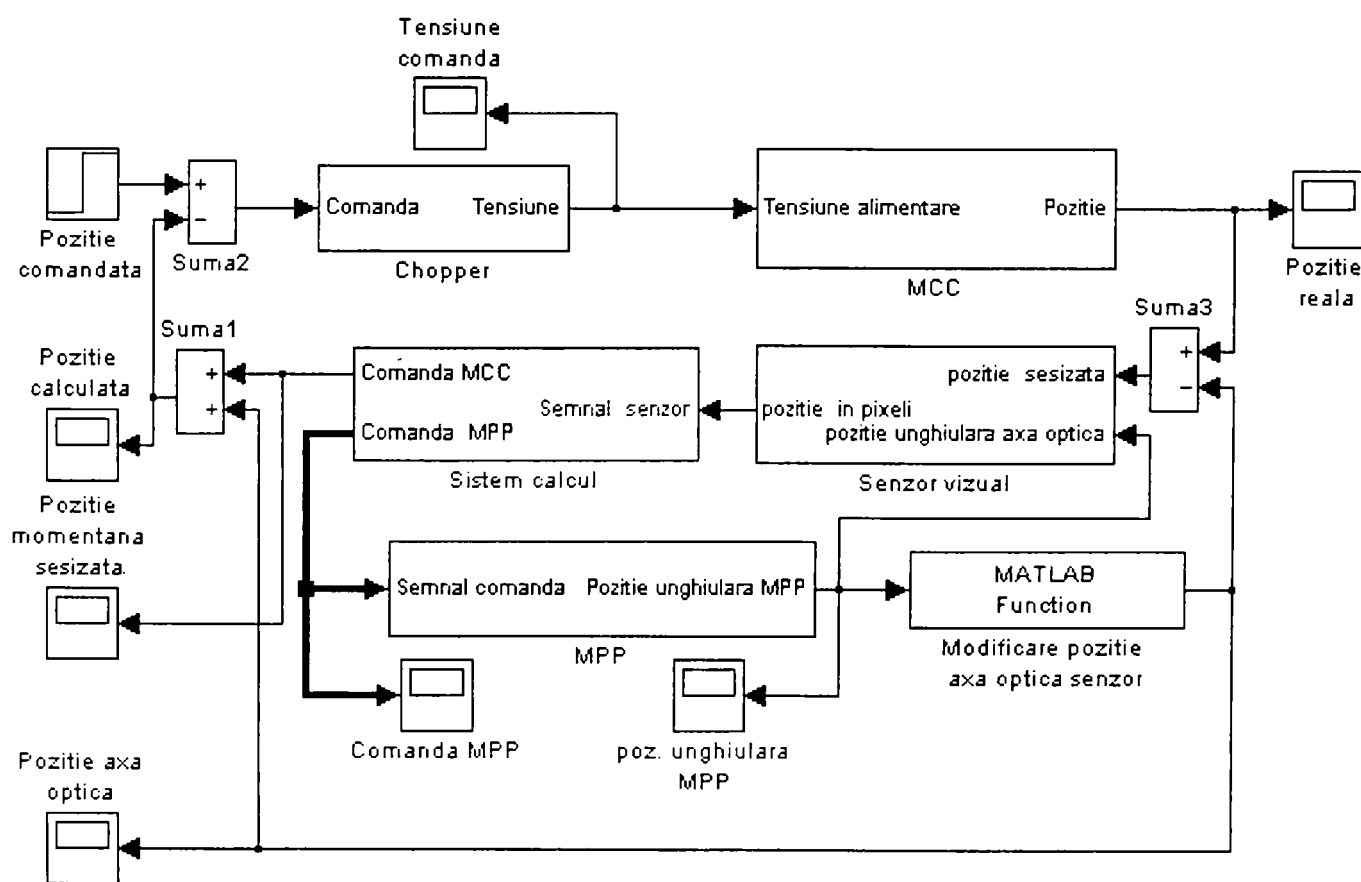


Fig. 7.11 Modelul funcțional al sistemului de conducere senzorială a robotului realizat (pentru o c.c.c. a robotului)

Blocul "Poziție comandată" generează un semnal treaptă care reprezintă poziția programată a axei. În momentul primirii acestui semnal blocul "Chopper" generează semnalul care reprezintă tensiunea necesară funcționării motorului de curent continuu. Blocul "MCC" are ca ieșire semnalul, care reprezintă situația relativă a elementelor c.c.c. modelate.

Această poziție este sesizată de "senzorul vizual" și evaluată de către "sistemul de calcul", comparându-se apoi cu mărimea programată. Rezultatul comparației este eroarea de poziție și reprezintă noua mărime de comandă.

Pe măsură ce mărimea care caracterizează poziția realizată crește, diferența dintre mărimea programată inițial și mărimea reală scade, scăzând în același timp și tensiunea de comandă a motorului de curent continuu, în final acesta oprindu-se în poziția programată. Deoarece câmpul vizual al senzorului nu acoperă întreg domeniul de mișcare al axei modelate, acesta se deplasează în așa fel încât punctul caracteristic să rămână în câmpul vizual al senzorului.

Deplasarea se realizează cu ajutorul unui motor pas cu pas comandat de sistemul de calcul în așa fel încât să execute un pas ori de câte ori punctul caracteristic se apropie la o anumită distanță de limita câmpului vizual al senzorului.

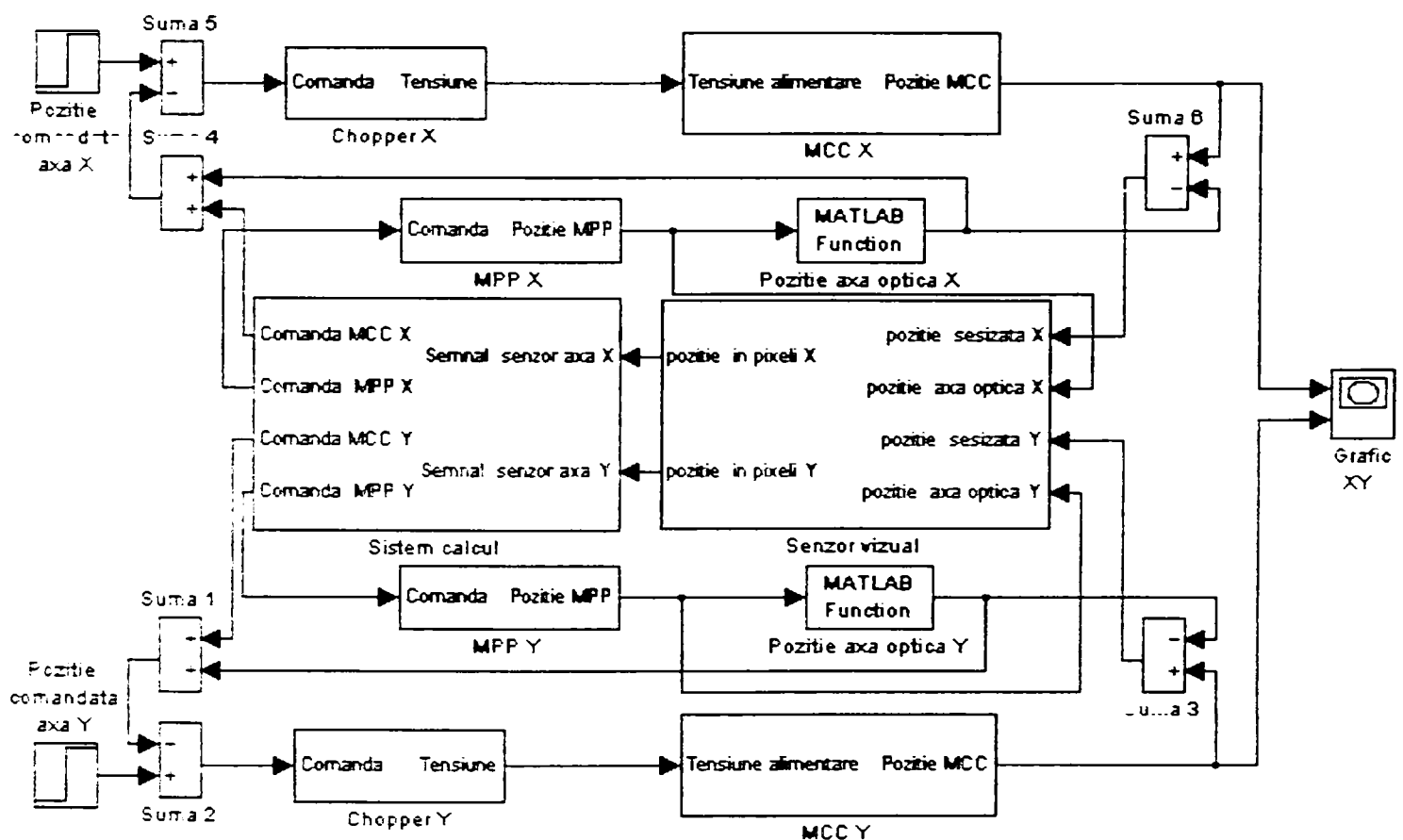


Fig. 7.12 Modelul funcțional al sistemului de conducere senzorială al robotului realizat (robot cu MGT de tip TT)

Structura și modul de funcționare al diferitelor module din cadrul modelului funcțional au fost descrise în detaliu în paragrafele anterioare. Pentru evaluarea comportamentului sistemului modelat, în schema bloc, s-au introdus elemente de afișare grafică a diferitelor semnale în funcție de timp ("Poziția reală", "Poziția axa optică" e.t.c.).

Elementele de acționare ("MCC X", "MCC Y") ale robotului sunt cele două motoare de curent continuu, ambele având asigurată reacția prin același senzor vizual ("Senzor vizual") și același sistem de calcul ("Sistem calcul").

Fiecare dintre elementele de acționare generează deplasările relative ale elementelor aferente c.c.c., având fiecare elemente de programare ("Poziția comandată pe axa Y", "Poziția comandată pe axa X"), elemente de execuție ("Chopper X", "Chopper Y"). Mișcările de compensare a senzorului vizual sunt asigurate de către motoarele pas cu pas ("MPP X", "MPP Y") și pozițiile axei optice a senzorului sunt calculate de către blocurile de calcul ("Poziția axa optică X", "Poziția axa optică Y"). Blocurile sumatoare "Suma 2" și "Suma 5" asigură realizarea buclei de reacție, iar blocurile sumatoare "Suma 1", "Suma 3", "Suma 4", "Suma 6" asigură realizarea compensării datorate urmării punctului caracteristic al robotului de către subsistemul vizual de urmărire.

Traectoria realizată de către punctul caracteristic al robotului poate fi urmărită cu ajutorul elementului de afișare "Grafic XY". Configurația prezentată poate realiza doar traiectorii rectilinii datorită faptului că generatoarele de semnal de comandă pentru axele X respectiv Y generează un semnal constant. Pentru a genera alte tipuri de traiectorii în locul generatoarelor de semnal de tip "treaptă" (step), se vor folosi alte tipuri de module.

7.4. Prezentarea rezultatelor simulării pe modelul experimental pentru conducerea pentru o c.c.c.

Rezultatele simulării sistemului robot pentru o c.c.c. a cărei elemente execută mișcări relative în lungul axei O_0x_0 s-au determinat la aplicarea unui semnal treaptă a cărui variație în timp se prezintă în figura 7.13.

Fișierul „val_par_1.m” cu inițializările pentru simularea sistemului de conducere senzorială a robotului realizat se prezintă în anexa A3.

Acest semnal reprezintă poziția comandată (prescrisă) a țintei luminoase după axa O_0x_0 . Semnal de comandă s-a generat cu o întârziere de 0,01 [s].

Semnalul se aplică chopperului și acesta va genera o tensiune de comandă a cărei variație în timp se prezintă în figura 7.14. După o creștere accentuată semnalul începe să descrească pe măsură ce poziția realizată se apropie ca valoare de cea prescrisă.

Discretizarea semnalului se datorează senzorului vizual care, datorită achiziției digitale a cadrelor, generează în aval de el semnale discrete.

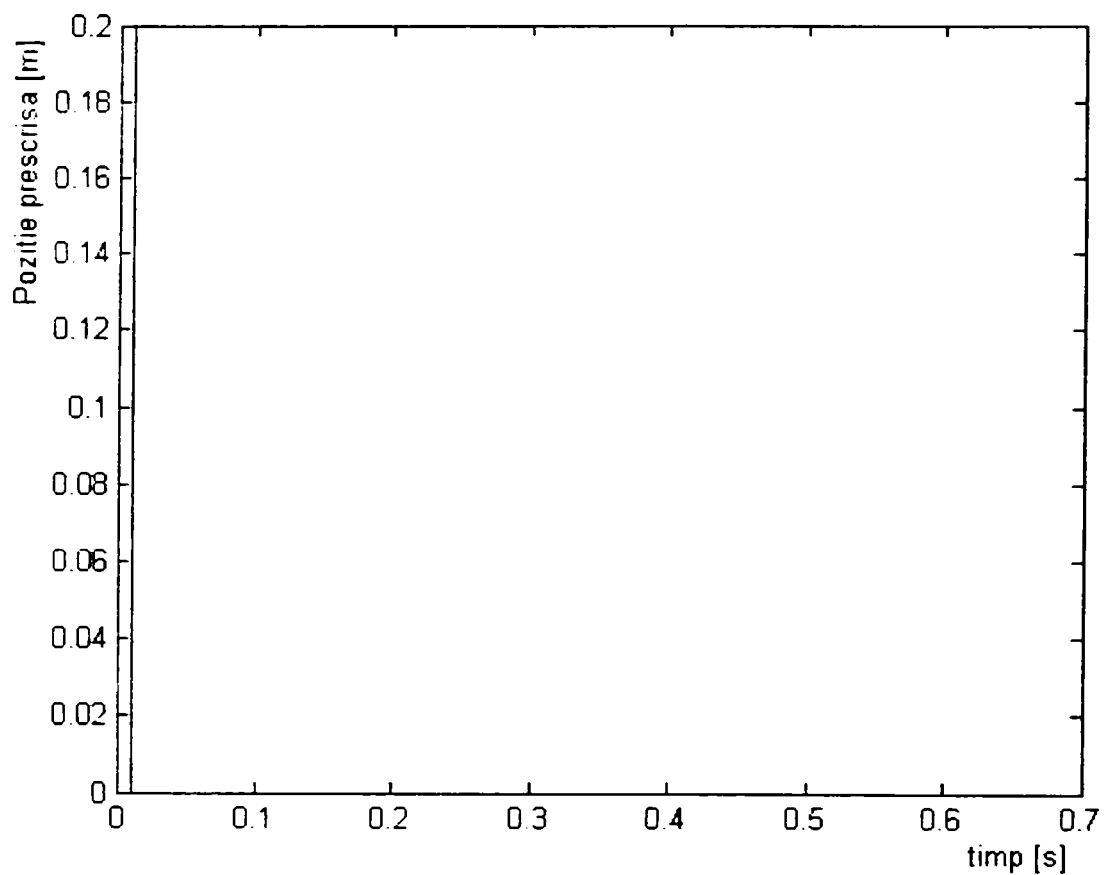


Fig.7.13. Variația în timp a semnalului de comandă aplicat sistemului

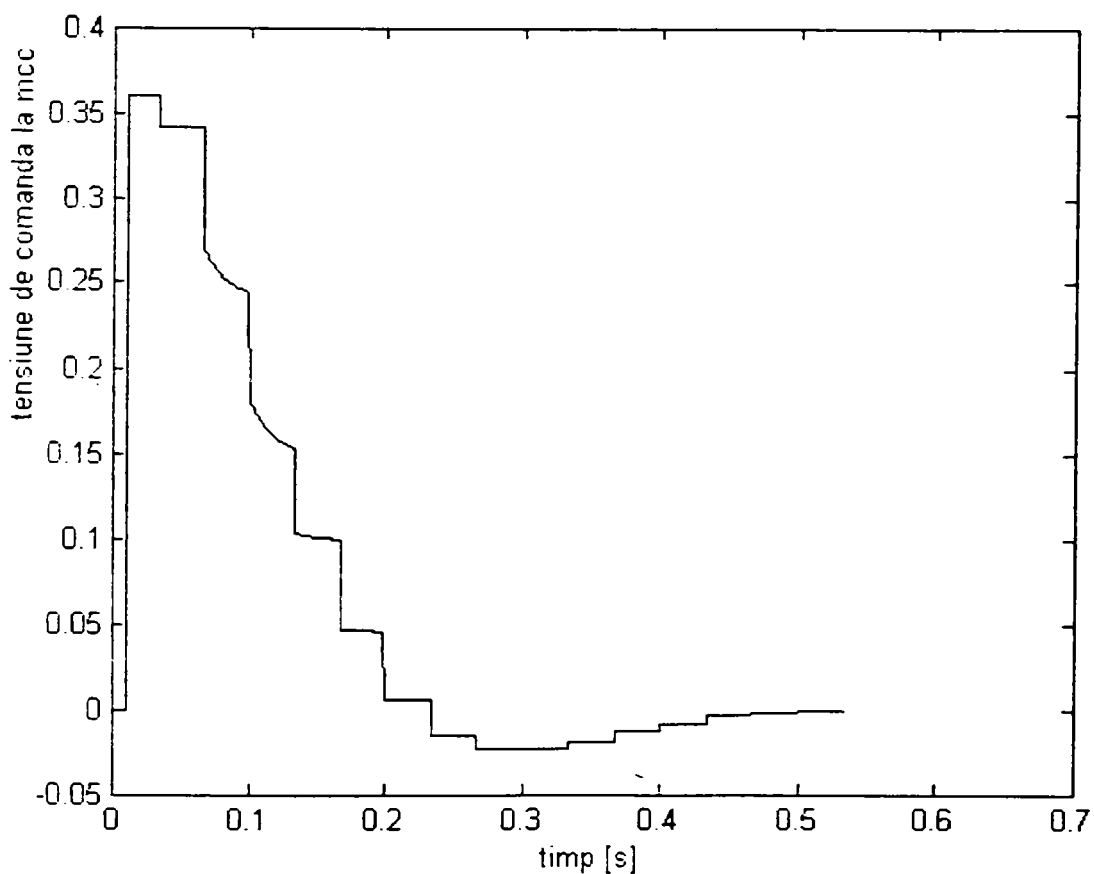


Fig.7.14. Variația în timp a tensiunii de comandă aplicată motorului de curent continuu

În figura 7.15. se prezintă răspunsul sistemului robot la aplicarea semnalului prescris de forma semnal treaptă. Acest semnal reprezintă variația în timp a poziției țintei de urmărit și se culege de la ieșirea din blocul MCC. Răspunsul sistemului robot la semnalul treaptă aplicat oferă unele informații referitoare la timpii de răspuns ai acestuia.

Se observă că timpul de reglare, după care sistemul robot atinge poziția prescrisă de 0,2 [m] cu o eroare de 2,5% este de 0,4 [s], iar dacă se dorește ca eroarea poziției prescrie să fie sub un procent timpul va fi de 0,5 [s].

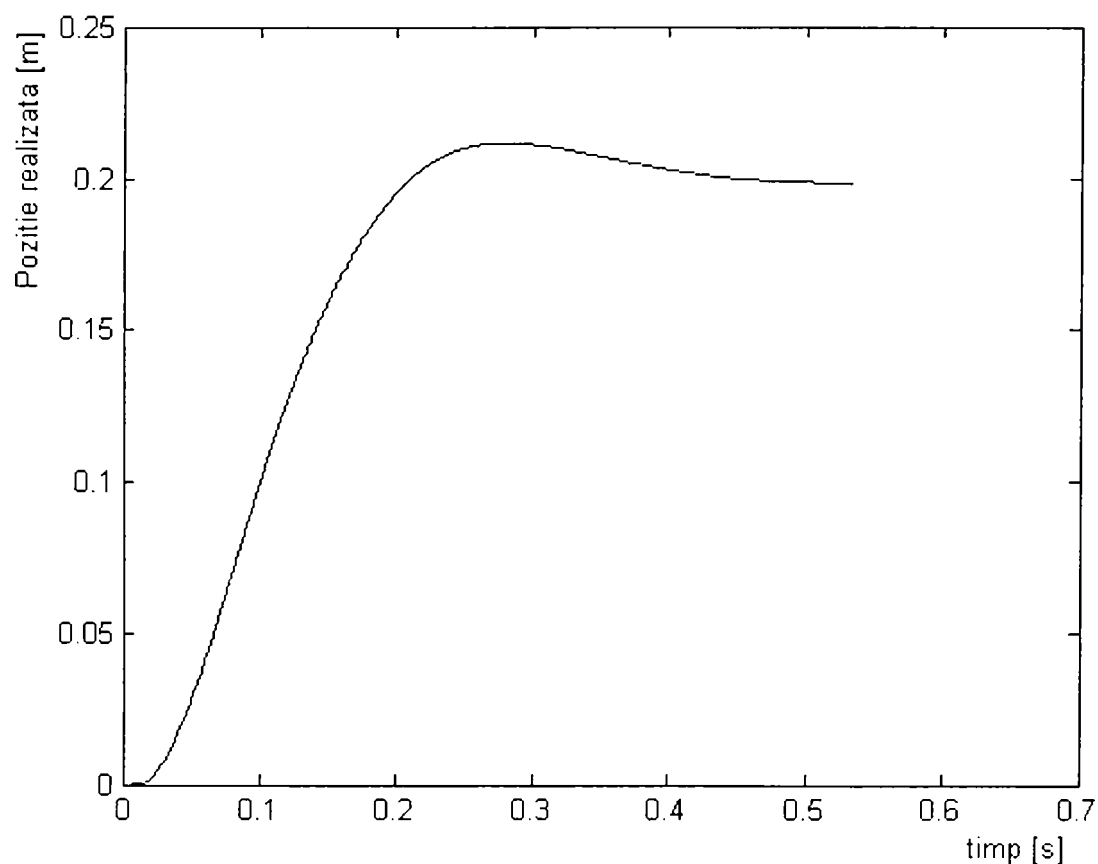


Fig.7.15. Răspunsul sistemului robot la aplicarea semnalului prescris de forma semnal treaptă

Variația în timp a vitezei unghiulare realizate la rotorul motorului de curent continuu se prezintă în figura 7.16. și se culege de la ieșirea blocului „Sarcină” prezentat în figura 7.1.

La simulări nu s-a limitat viteza unghiulară realizată de rotorul motorului de curent continuu, după cum se observă și din figura 7.16, astfel încât să se obțină un profil trapezoidal pentru variația în timp a vitezei.

Răspunsul senzorului vizual, care în același timp este și mărime de intrare pentru sistemul de calcul se prezintă în figura 7.17 și se culege de la ieșirea blocului „semnal senzor” prezentat în figura 7.8.

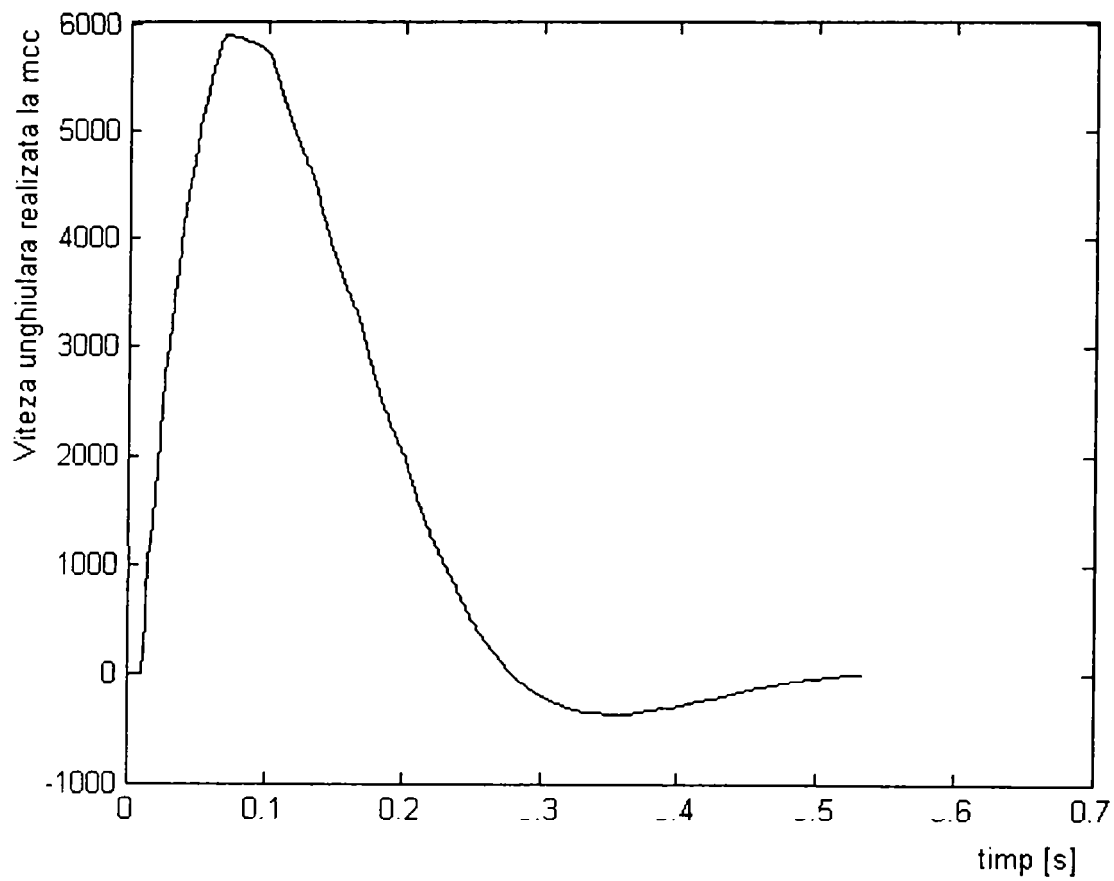


Fig.7.16. Variația în timp a vitezei unghiulare realizate la rotorul motorului de curent continuu

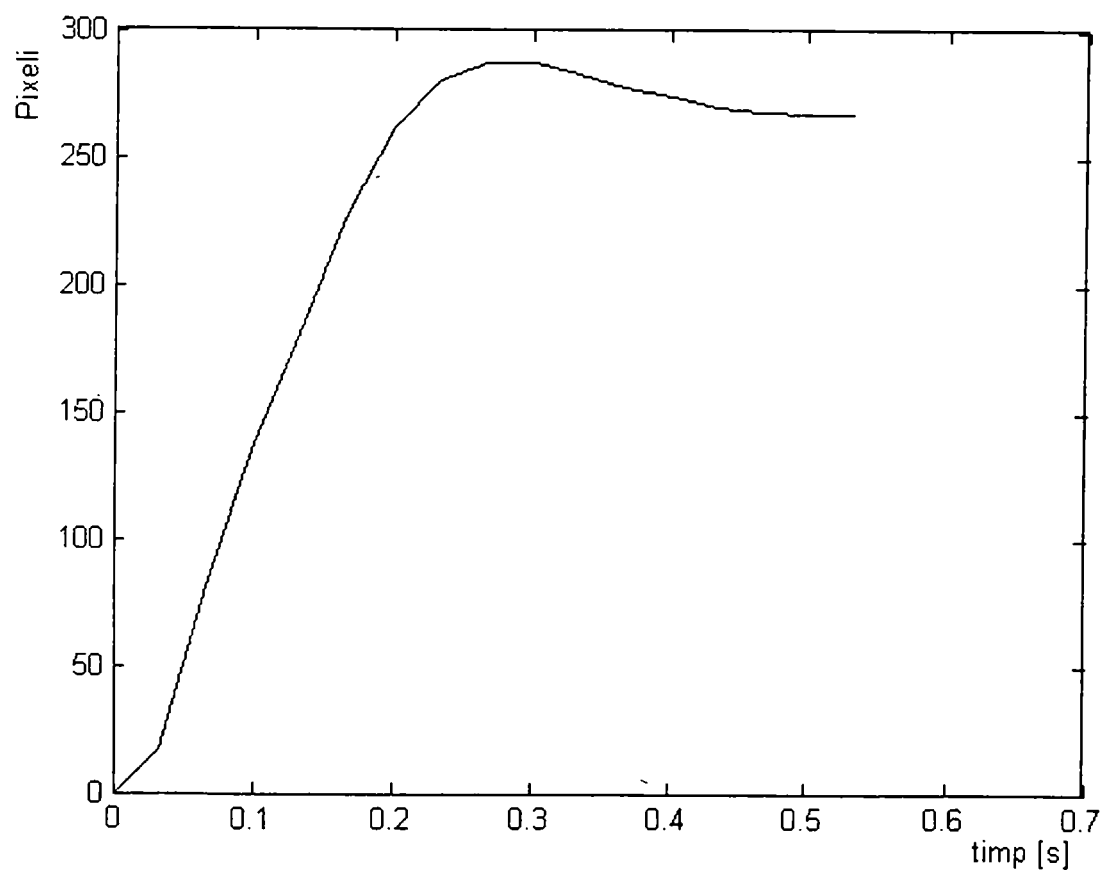


Fig.7.17. Variația în timp a poziției țintei determinată de senzorul vizual

Blocul „Sistem calcul” prezentat în figura 7.8 generează semnalul necesar comenzii motorului pas cu pas, dar și semnalul necesar comenzii motorului de curent continuu. Aceste două semnale se prezintă în figurile 7.18 și respectiv 7.19.

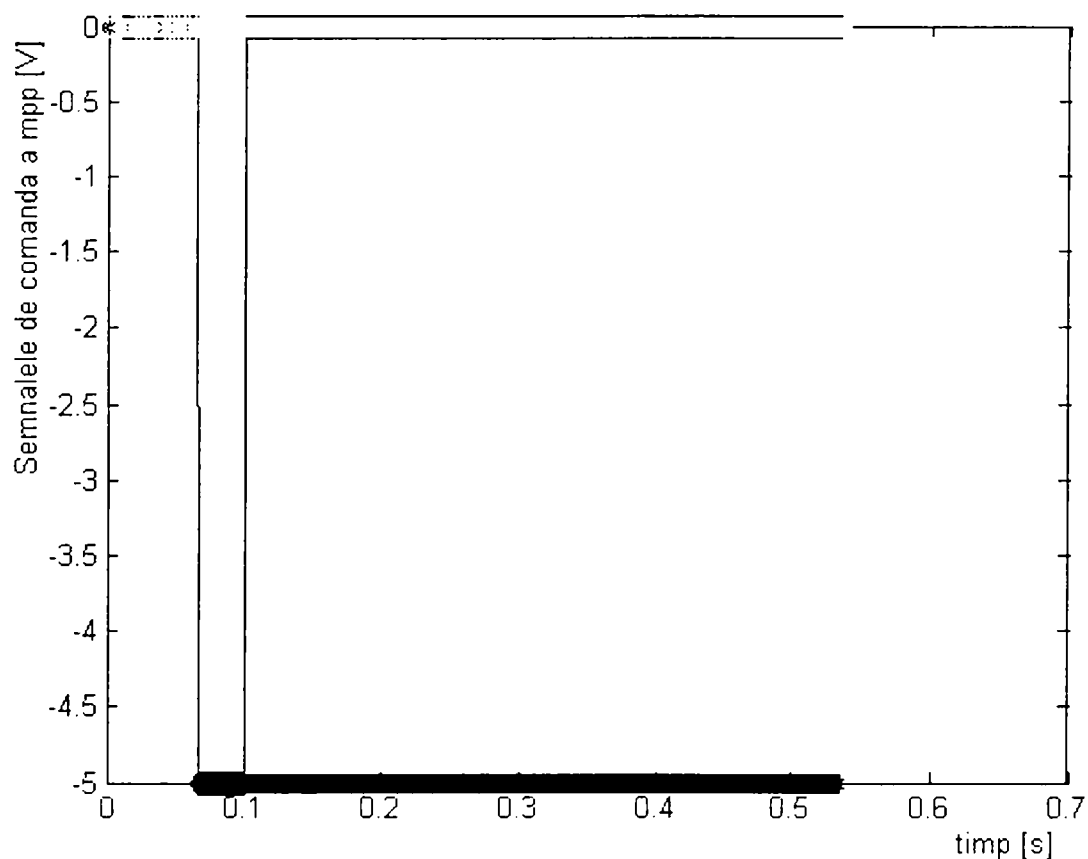


Fig.7.18. Variația în timp a semnalelor de comandă ale m.p.p. generate de blocul „Sistem calcul”

În timpul deplasării țintei blocul „Sistem calcul” a determinat necesitatea generării a două semnale de comandă pentru motorul pas cu pas astfel încât imaginea țintei să fie readusă în „dreptunghiul de interes” din planul imagine.

Semnalul de ieșire din motorul pas cu pas este transformat în blocul „Modificare poziție axa optica senzor” într-un semnal corespunzător mărimii x_{opt} calculate cu relația 7.2., semnal a cărui evoluție în timp se prezintă în figura 7.19.

Peste acest semnal se suprapune semnalul prezentat în figura 7.20., semnalul poziției curente a punctului caracteristic față de intersecția axei optice a senzorului cu planul punctului caracteristic x_r , semnal care se culege de la blocul „poziție momentană sesizată”, rezultând poziția determinată cu ajutorul senzorului vizual (poziția reală), a cărui grafic se prezintă în figura 7.21.

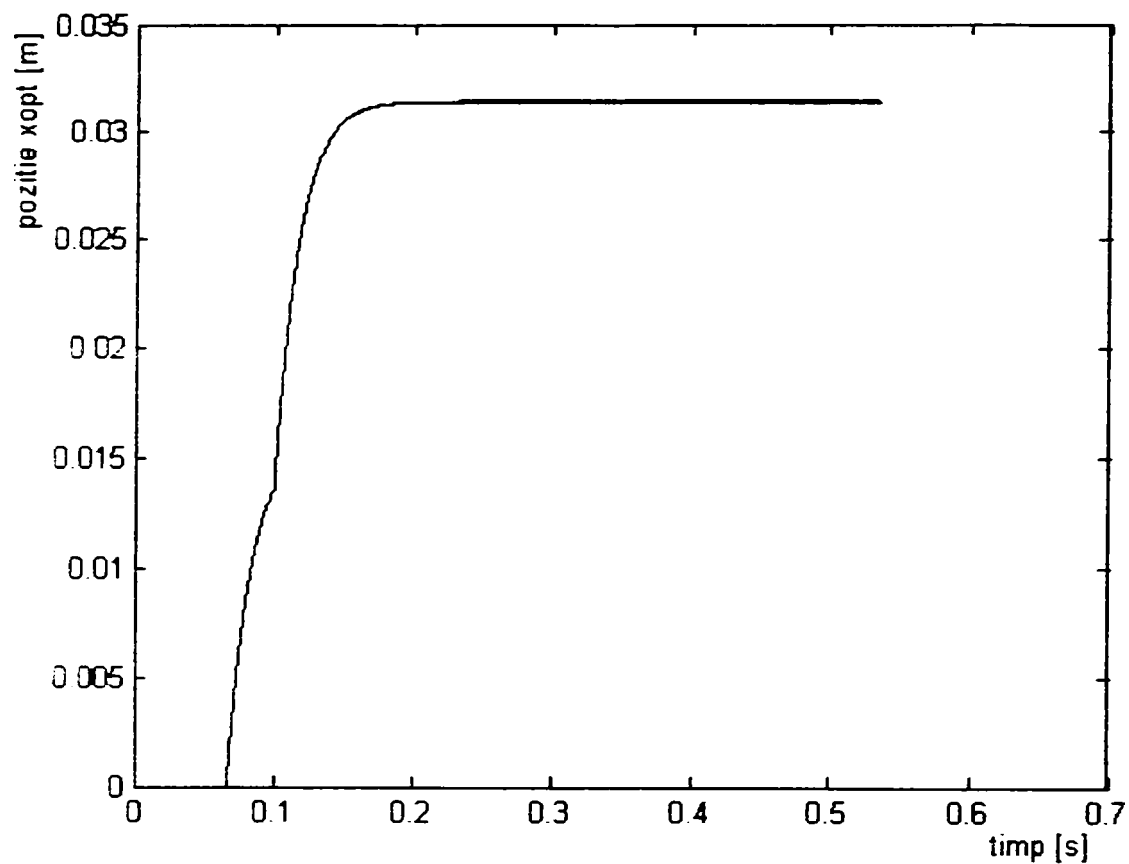


Fig.7.19. Variația în timp a semnalului corespunzător mărimii x_{opt}

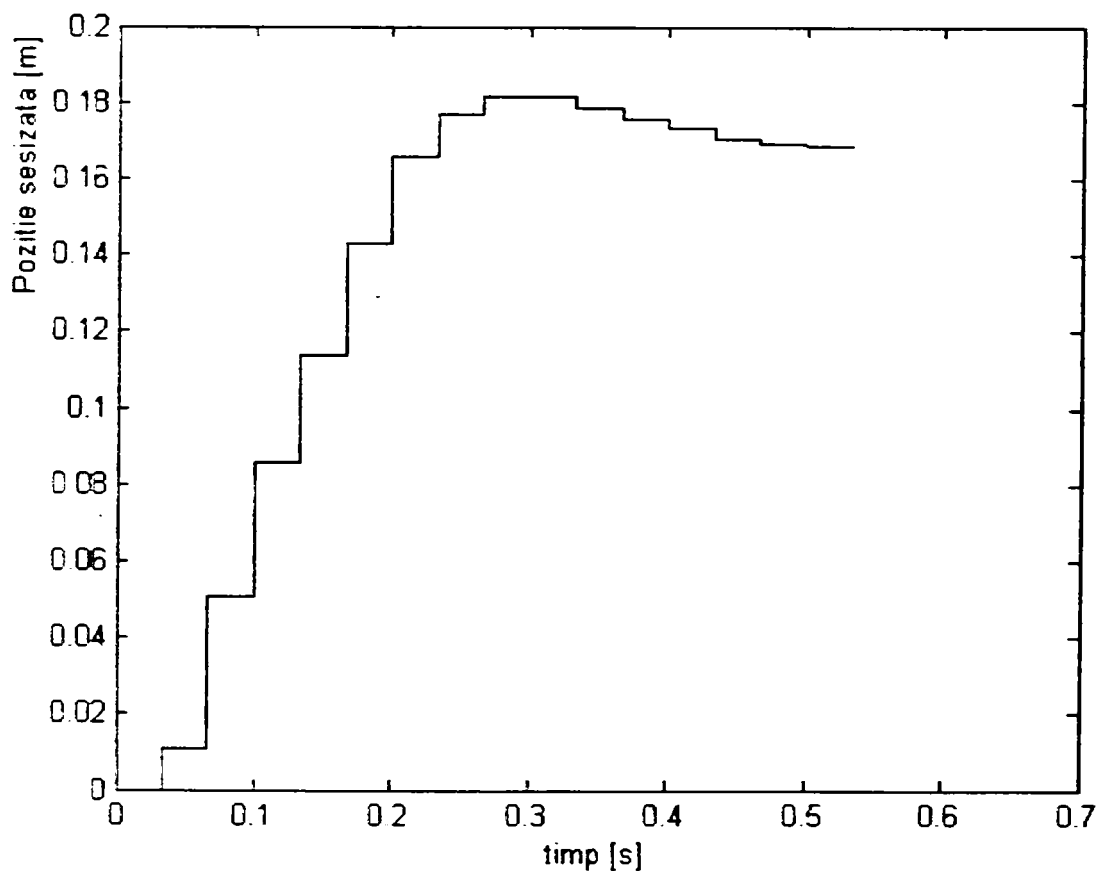


Fig.7.20. Variația în timp a poziției curente a punctului caracteristic x_r față de intersecția axei optice a senzorului cu planul punctului caracteristic

Timpul de reacție a sistemului robot se poate determina și din figura 7.19. Astfel se observă că subsistemul vizual de urmărire realizează doi pași consecutivi la un interval de 0,0332 [s] timp în care și senzorul vizual achiziționează un cadru. Prin urmare timpul necesar sistemului robot pentru a parcurge o buclă completă (inclusiv comanda MPP) este de cel puțin 0,0332 [s].

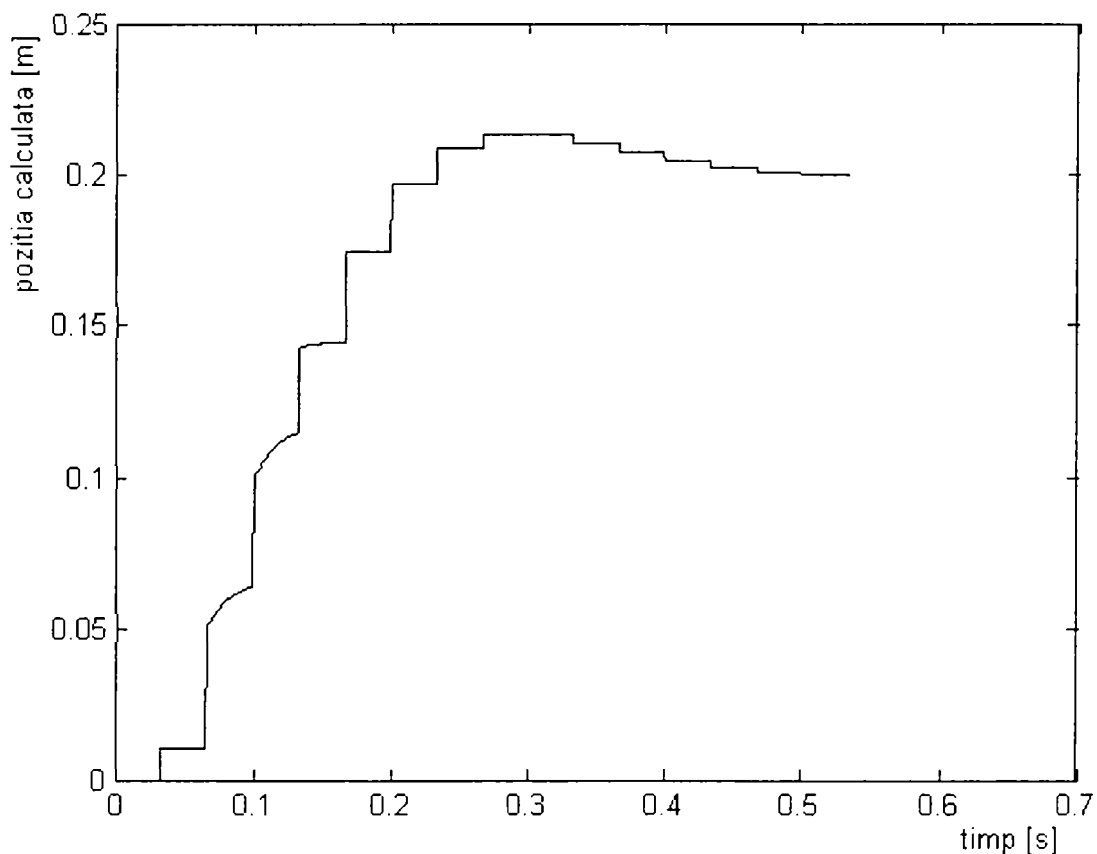


Fig.7.21. Variația în timp a poziției calculate

Poziția calculată este în fapt mărimea de reacție a sistemului robot, care suprapusă peste semnalul de referință generează comanda de corecție necesară a se aplica la nivelul c.c.c.

Deoarece rezultatele simulării funcționării sistemului robot pentru c.c.c. a cărei mișcare este după axa O_0y_0 sunt similare cu cele prezentate pentru c.c.c. a cărei mișcare este după axa O_0x_0 (rezultate prezentate în figurile 7.13-7.21), acestea nu se mai prezintă.

7.5. Prezentarea rezultatelor simulării pe modelul funcțional

Rezultatele simulării sistemului robot realizat la care informațiile sunt achiziționate de subsistemul vizual de urmărire s-au determinat la aplicarea unui semnal treaptă prezentat în figura 7.22. Mărimile înregistrate în timpul simulării sunt prezentate sintetic în tabelul 7.2.

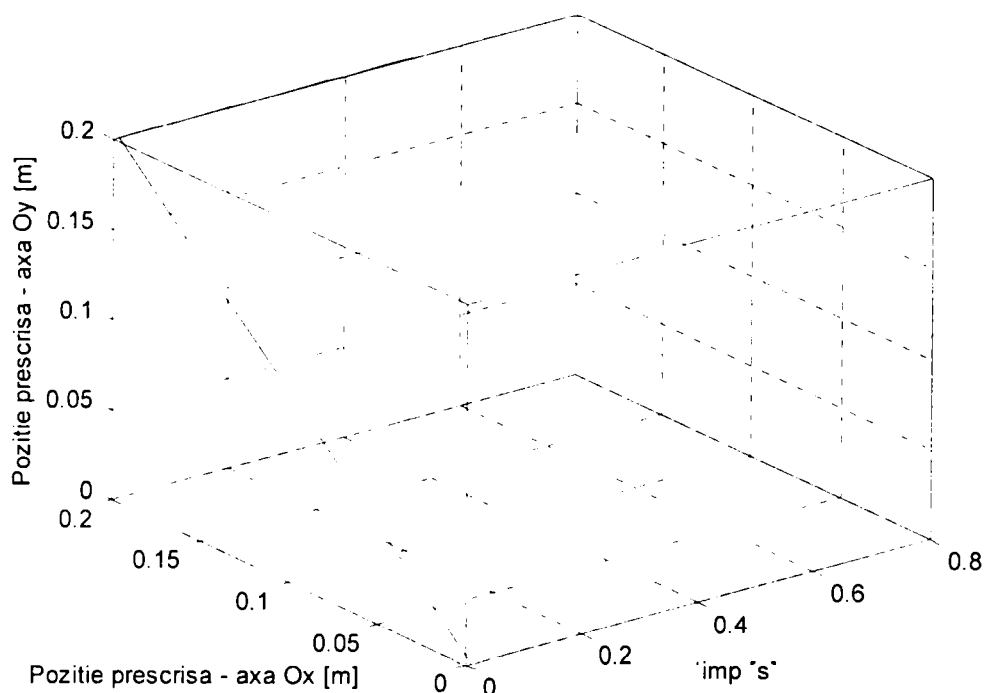


Fig.7.22. Semnalul de comandă aplicat sistemului robot

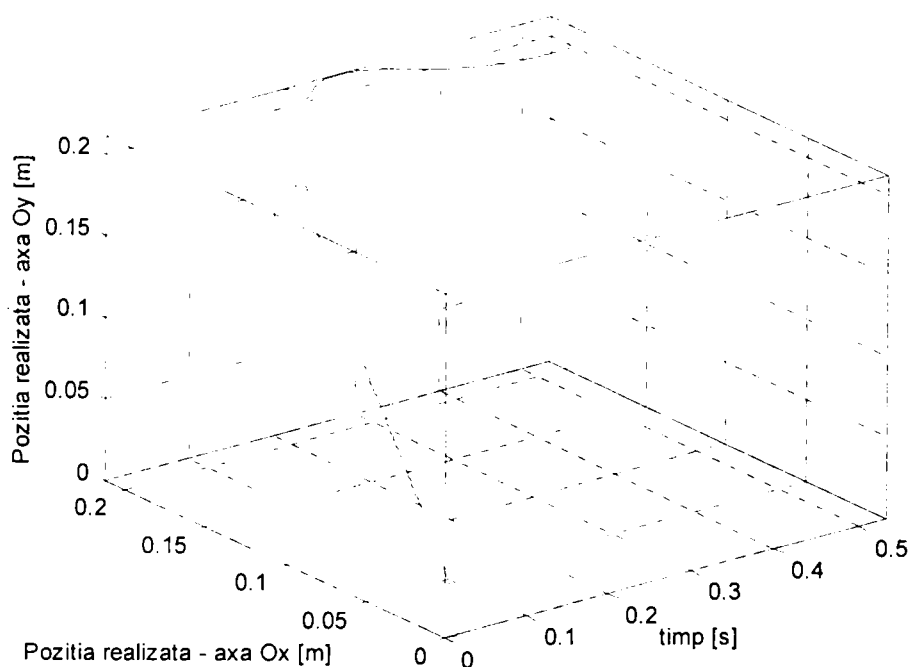
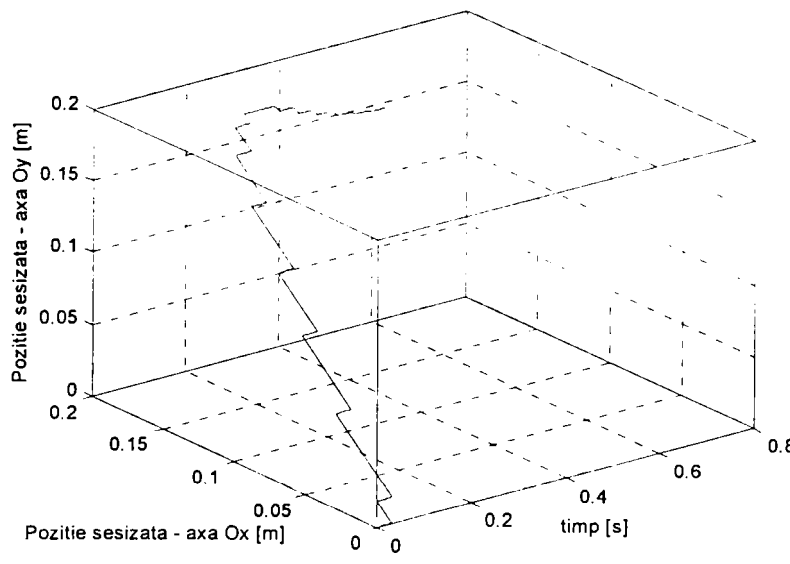
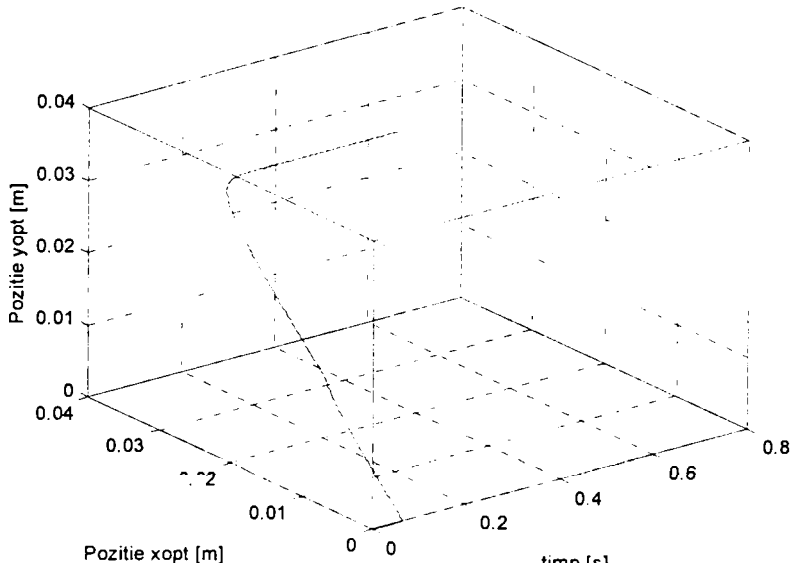
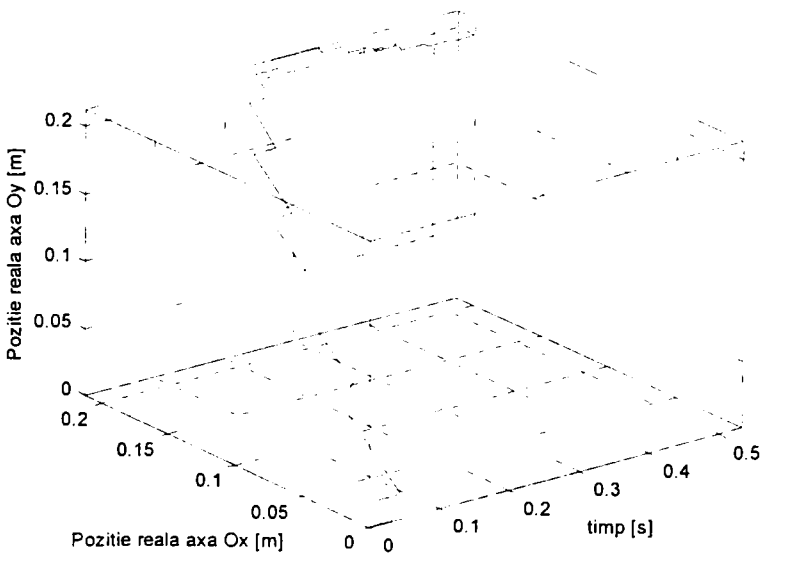


Fig.7.23. Răspunsul sistemului robot la aplicarea semnalului prescris de forma semnal treaptă

Tabelul 7.2. Mărimi înregistrate în timpul simulării

Nr.crt.	Mărime	Grafic
1.	Poziția sesizată a țintei	
2.	Poziții x_{opt} , y_{opt}	
3.	Poziție reală	

7.6. Concluzii

Rezultatele simulării funcționării sistemului robot care utilizează informații de la un subsistem vizual de urmărire validează modelele concepute.

După cum se observă din răspunsul sistemului robot la aplicarea semnalului prescris de forma semnal treaptă, răspuns prezentat în figura 7.23 timpul pentru atingerea poziției prescrise cu o eroare de un procent este de 0,53 [s], timp în care bucla de reglare a sistemului a fost parcursă de 16 ori. Prin urmare, timpul mediu pentru parcurgerea unei bucle de reacție este de 0,033 [s].

Timpul necesar generării comenzii mcc și mpp este sub 0,01 [s].

Timpul aferent achiziției imaginii rezultă din frecvența de achiziție a plăcii Matrox, frecvență care este de 30 de cadre/s. De aici se observă că limitarea în ceea ce privește scăderea timpului de execuție al buclei de reglare o dă sensorul vizual.

În tabelul 7.3. se prezintă timpii obținuți din simularea funcționării sistemului robot.

Tabelul 7.3. Timpii obținuți din simulare în [s]

Proces simulat	Timp obținut din simulare
Generare semnal mcc	0,01
Achiziție imagine	0,03
Calculule în blocul „Sistem calcul”	0,01
Generare semnal mpp	0,01

Rezultatele simulării funcționării sistemului robot se necesită a fi validate de către sistemul fizic, concret al sistemului de conducere senzorială a robotului care utilizează informații de la un subsistem vizual de urmărire.

În concluzie, ținând cont de timpii obținuți la simulare se poate afirma că sistemul de conducere senzorială a robotului care utilizează informații de la un subsistem vizual de urmărire funcționează în timp real.

Capitolul 8 Construcția unui sistem experimental de conducere senzorială a unui robot cu două grade de mobilitate, pe baza informațiilor achiziționate printr-un subsistem vizual de urmărire. Experimentări.

8.1. Introducere

În cadrul capitolului se prezintă construcția sistemului robot cu conducere globală realizat și rezultatele verificării experimentale a funcționării acestuia.

Pornind de la aspectele teoretice referitoare la concepția sistemului de conducere senzorială a roboților utilizând informațiile furnizate de un subsistem vizual de urmărire, prezentate în capitolul 5 și de la modelul funcțional al sistemului conceput, prezentate în capitolul 7, în cadrul capitolului se urmărește validarea experimentală a rezultatelor teoretice și a simulărilor funcționării acestuia pe un sistem fizic.

S-a pornit de la dezideratul ca sistemul de conducere senzorială a robotului să fie un sistem simplu (acest criteriu rezidă evident din factorul economic), dar care trebuie să asigure o experimentare concludentă a principiului conceput, în vederea evaluării corecte și cât mai fidele a sistemului.

Ținând seama de aceste considerente, robotul pe care s-au făcut experimentările prezintă două grade de mobilitate, iar subsistemul vizual de urmărire se compune dintr-un singur senzor vizual.

8.2. Descrierea sistemului robot experimental conceput și realizat

8.2.1. Prezentare generală

Pentru confirmarea experimentală a teoriei referitoare la sistemele de conducere globală a roboților (sistem de conducere cu buclă de reacție de la mediu), care se bazează în culegerea informațiilor cu ajutorul unor sisteme vizuale, autorul a conceput un sistem compus dintr-un robot în coordonate carteziane TT și un subsistem de urmărire cu un senzor vizual, prezentat în fotografia din figura 8.1.

Rolul subsistemului vizual de urmărire este acela de a achiziționa informații cu privire la poziția punctului caracteristic aferent efectorului final al robotului în timp real, astfel încât să poată oferi aceste informații sistemului de comandă al robotului.



Fig.8.1. Vedere a sistemului experimental realizat

Totodată, subsistemul vizual de urmărire trebuie să monitorizeze în timp real deplasarea efectorului final al robotului, astfel încât acesta să nu părăsească spațiul de observabilitate al senzorului vizual.

Schema bloc a sistemului robot care prezintă conducere globală se prezintă în figura 8.2.

Subsistemele componente ale sistemului robot sunt:

- *sistemul de comandă* a sistemului robot, care are rolul de a preleva informațiile de la subsistemul vizual de urmărire referitoare la poziția punctului caracteristic al robotului, de a compara aceste poziții efectiv realizate cu pozițiile programate și printr-o analiză inversă aplicată erorilor

astfel obținute se vor obține mărimile corecțiilor necesare a fi făcute la nivelul fiecărei c.c.c.; aceste corecții se vor suprapune peste noile comenzi care se necesită a se aplica la fiecare c.c.c.; structura sistemului de comandă precum și modul lui de lucru se prezintă detaliat în paragraful 8.3.3;

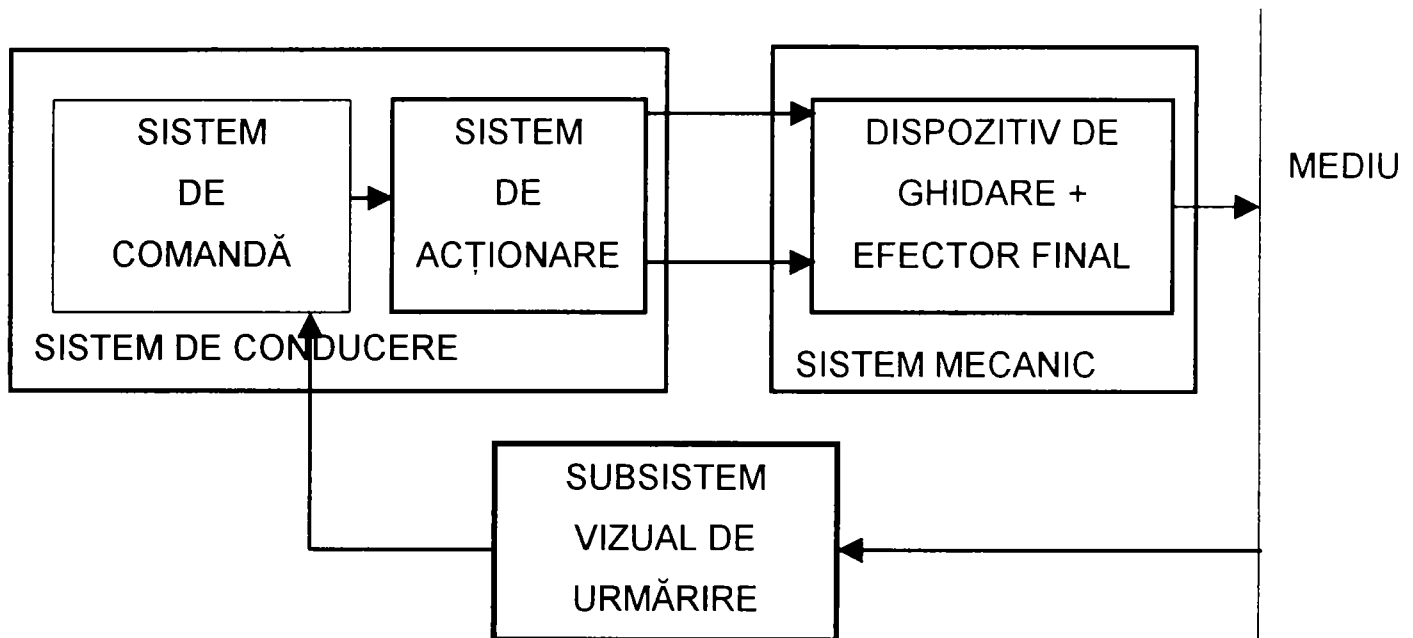


Fig.8.2. Schema bloc a sistemului robot care prezintă conducere globală

- *sistemul mecanic* al robotului; autorul a optat pentru utilizarea unui robot, care servește stația ASRS a sistemului CIM al Universității din Oradea, care are două c.c.c. de translație, în vederea realizării încercărilor; o imagine de ansamblu a robotului se prezintă în figura 8.3;
- *sistemul de acționare* al robotului are rolul de a pune în mișcare elementele aferente c.c.c. ale MGT al robotului în vederea asigurării energiei necesare efectorului final de a interacționa cu mediul și respectiv pentru a asigura acestuia o succesiune de poziții dorite; elementul de acționare de la nivelul c.c.c. se compune din câte un servomotor de c.c., a căror caracteristici sunt prezentate în paragraful 7.3.3., cu transmisia mecanică aferentă (reductor melc-roată melcată și transmisii cu curele dințate); modul de comandă a sistemului de acționare se prezintă detaliat în paragraful 8.3.2.;



Fig.8.3. Vedere de ansamblu a robotului utilizat

- *subsistemul vizual de urmărire* proiectat și realizat de către autor este prezentat în fotografiile din figurile 8.4, respectiv 8.5; conceperea și realizarea subsistemului vizual de urmărire a ținut cont de toate aspectele teoretice prezentate în paragrafele precedente ale acestui capitol; concepția subsistemelor de comandă și acționare și a ansamblului sunt prezentate detaliat în paragrafele 8.3.2 și 8.3.3;

Configurația realizată și concepută a rezultat din posibilitățile efective de hard avute la dispoziție de către autor.

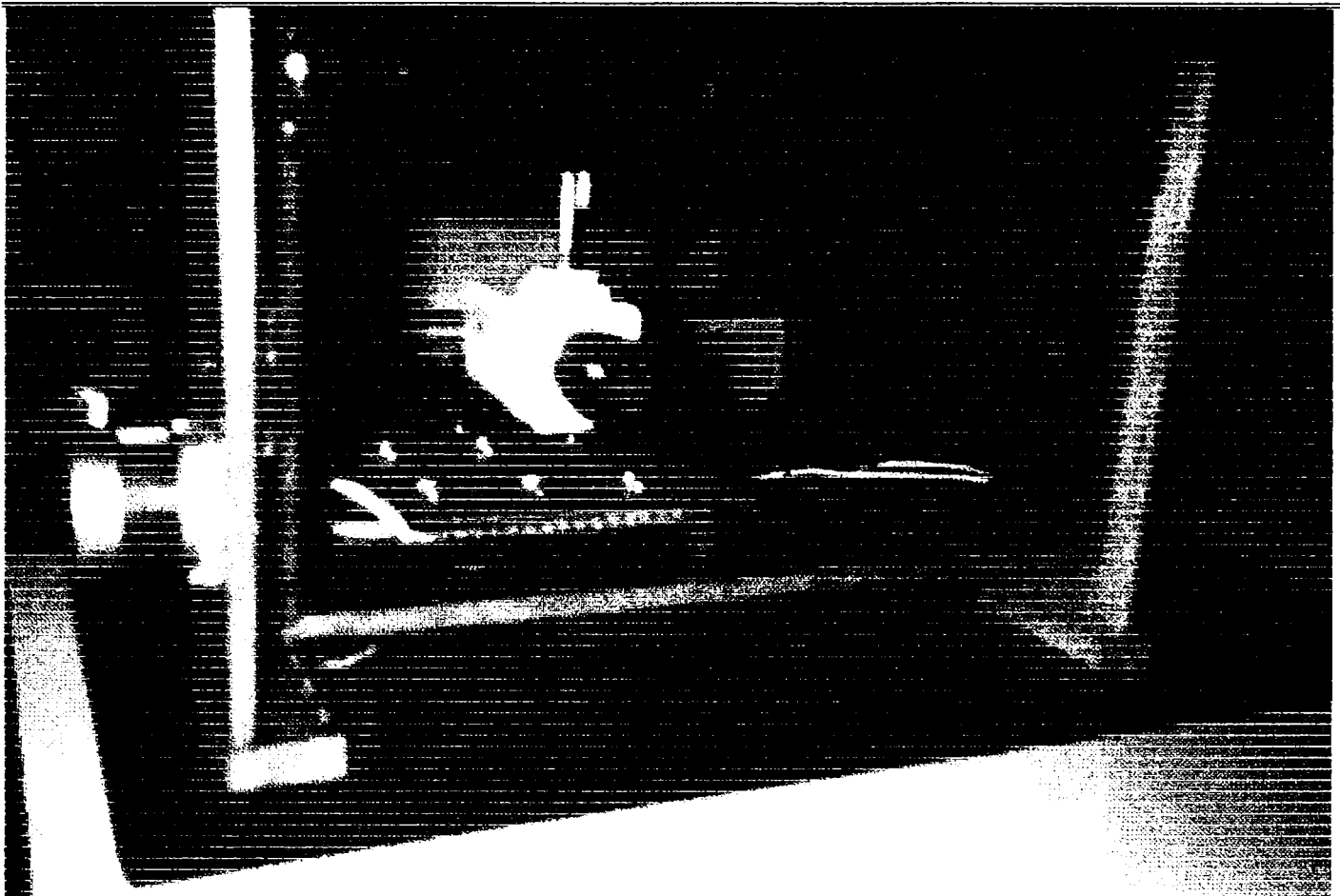


Fig.8.4. Vedere din față a subsistemului vizual de urmărire realizat

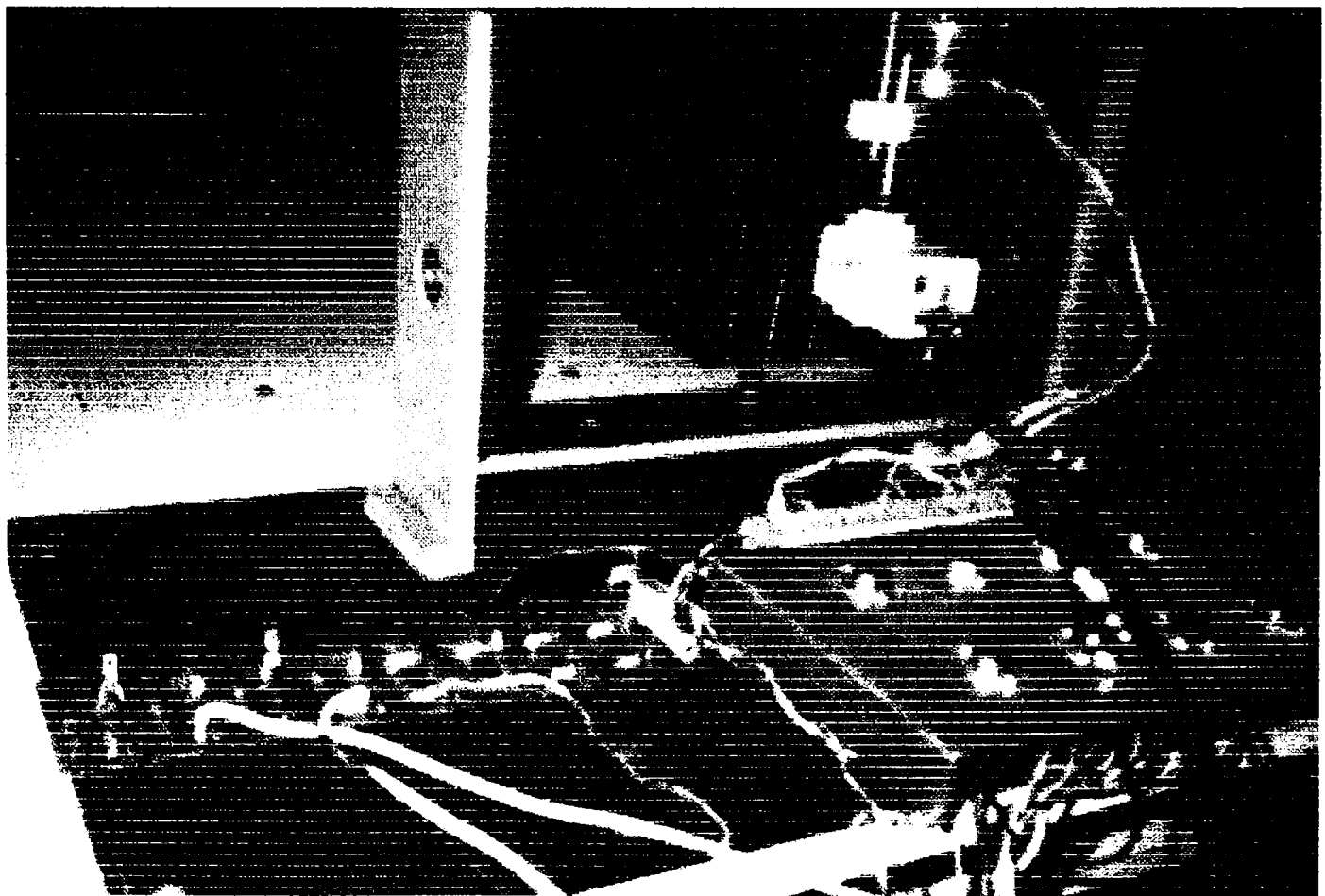


Fig.8.5. Vedere din spate a subsistemului vizual de urmărire realizat

8.2.2. Particularizarea modului de calcul al mărimilor de corecție la nivelul fiecărei c.c.c. pentru sistemul robot experimental conceput și realizat

În figura 8.6 se prezintă schema cinematică a sistemului robot compus din: robot și subsistem vizual de urmărire.

Deoarece robotul are dispozitivul de ghidare de structură TT, matricea de transformare omogenă 0T_2 , dată de relația (5.65) va avea expresia:

$${}^0T_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & c_1 \\ 0 & 0 & 1 & l_1 + l_2 + c_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (8.1)$$

Situarea comandată a obiectului de manipulat dată de (5.68) este:

$${}^{0_0}T_{\alpha_c} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & c_1 \\ 0 & 0 & 1 & l_1 + l_2 + c_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (8.2)$$

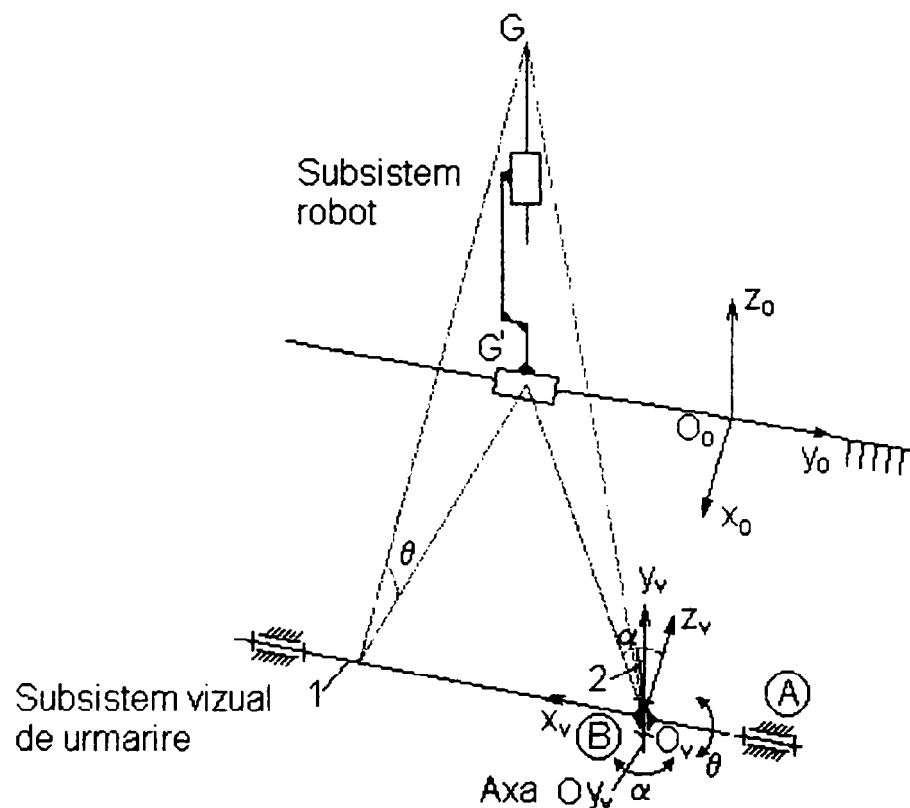


Fig.8.6. Schema cinematică a sistemului robot realizat

Pentru determinarea poziției reale a punctului caracteristic al obiectului de manipulat se ține cont de faptul că, utilizând un singur senzor vizual se poate determina doar direcția pe care se găsește punctul țintă.

Deoarece punctul țintă se mișcă într-un plan, atunci din intersecția direcției de vizare cu planul în care evoluează punctul țintă vor rezulta coordonatele poziției reale ale punctului țintă.

În sistemul de referință $O_0x_0y_0z_0$ ținta, în timpul mișcării ei, va avea cota $x_0=0$. Considerând sistemul de referință $O_vx_vy_vz_v$ amplasat ca în figura 8.6 (astfel încât axele O_0x_0 și O_vz_v , respectiv O_0y_0 și O_vx_v , respectiv O_0z_0 și O_vy_v să fie paralele) și coordonatele punctului O_v în sistemul de referință $O_0x_0y_0z_0$ ca fiind $O_v(D \ 0 \ 0)$, atunci matricea de situare a subsistemului vizual de urmărire în sistemul de referință universal ${}^{0_0}\underline{T}_{O_v}$ se obține aplicând următorul set de transformări: translație în lungul x cu D; rotație în jurul axei z cu unghiul $-\frac{\pi}{2}$; rotație în jurul axei x cu unghiul $\frac{\pi}{2}$:

$${}^{0_0}\underline{T}_{O_v} = \text{Trans}_x(D) \cdot \text{Rot}_z\left(-\frac{\pi}{2}\right) \cdot \text{Rot}_x\left(\frac{\pi}{2}\right) \quad (8.3)$$

sau după efectuarea înmulțirilor:

$${}^{0_0}\underline{T}_{O_v} = \begin{vmatrix} 0 & 0 & -1 & D \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (8.4)$$

Coordonatele punctului țintă în sistemul de coordonate $O_vx_vy_vz_v$ rezultă din relația (5.30) ținând cont că ținta are cota ${}^{0_v}z = D$:

$$\begin{aligned} {}^{0_v}x &= D \cdot \frac{x_{G'_2} \cdot c\alpha_1 - y_{G'_2} \cdot \frac{s\alpha_1 \cdot \text{tg}\theta_1}{\sqrt{\text{tg}^2\alpha_1 + 1 + \text{tg}^2\theta_1}} - \frac{\text{tg}\alpha_1}{\sqrt{\text{tg}^2\alpha_1 + 1 + \text{tg}^2\theta_1}} \cdot f}{-x_{G'_2} \cdot s\alpha_1 - y_{G'_2} \cdot \frac{c\alpha_1 \cdot \text{tg}\theta_1}{\sqrt{\text{tg}^2\alpha_1 + 1 + \text{tg}^2\theta_1}} - \frac{1}{\sqrt{\text{tg}^2\alpha_1 + 1 + \text{tg}^2\theta_1}} \cdot f} \\ {}^{0_v}y &= D \cdot \frac{y_{G'_2} \cdot \frac{1}{c\alpha_1 \cdot \sqrt{\text{tg}^2\alpha_1 + 1 + \text{tg}^2\theta_1}} - \frac{\text{tg}\theta_1}{\sqrt{\text{tg}^2\alpha_1 + 1 + \text{tg}^2\theta_1}} \cdot f}{-x_{G'_2} \cdot s\alpha_1 - y_{G'_2} \cdot \frac{c\alpha_1 \cdot \text{tg}\theta_1}{\sqrt{\text{tg}^2\alpha_1 + 1 + \text{tg}^2\theta_1}} - \frac{1}{\sqrt{\text{tg}^2\alpha_1 + 1 + \text{tg}^2\theta_1}} \cdot f} \\ {}^{0_v}z &= D \end{aligned} \quad (8.5)$$

Vectorul de poziție a țintei determinat prin utilizarea subsistemului vizual de urmărire este conform relației (5.32):

$${}^{o_c} \underline{r}_{O_{Re}} = \begin{pmatrix} D \cdot \frac{x_{G'_2} \cdot c\alpha_1 - y_{G'_2} \cdot \frac{s\alpha_1 \cdot \operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} - \frac{\operatorname{tg}\alpha_1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f}{-x_{G'_2} \cdot s\alpha_1 - y_{G'_2} \cdot \frac{c\alpha_1 \cdot \operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} - \frac{1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f} \\ D \cdot \frac{y_{G'_2} \cdot \frac{1}{c\alpha_1 \cdot \sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} - \frac{\operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f}{-x_{G'_2} \cdot s\alpha_1 - y_{G'_2} \cdot \frac{c\alpha_1 \cdot \operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} - \frac{1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f} \\ D \\ 1 \end{pmatrix} \quad (8.6)$$

Pentru calculul vectorului erorii de poziționare se pornește de la matricea erorii de situație dată de relația (5.68), care devine:

$${}^{o_c} \underline{e}_{O_{Re}} = \left({}^{o_c} \Delta p_{x_{O_{Re}}} \quad {}^{o_c} \Delta p_{y_{O_{Re}}} \quad {}^{o_c} \Delta p_{z_{O_{Re}}} \quad 1 \right)^T = {}^{o_o} \underline{T}_{O_c}^{-1} \cdot {}^{o_o} \underline{T}_{O_r} \cdot {}^{o_r} \underline{r}_{O_{Re}} \quad (8.7)$$

Inversa matricei ${}^{o_o} \underline{T}_{O_c}$ este:

$$\underline{T}_{O_c}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -c_1 \\ 0 & 0 & 1 & -l_1 - l_2 - c_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (8.8)$$

Ținând cont de relațiile (8.5), (8.6), (8.7) și (8.8) vectorul erorii de poziție este:

$${}^{o_c} \underline{e}_{O_{Re}} = \begin{pmatrix} 0 \\ -D \cdot \frac{x_{G'_2} \cdot c\alpha_1 - y_{G'_2} \cdot \frac{s\alpha_1 \cdot \operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} - \frac{\operatorname{tg}\alpha_1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f}{-x_{G'_2} \cdot s\alpha_1 - y_{G'_2} \cdot \frac{c\alpha_1 \cdot \operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} - \frac{1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f} - c_1 \\ D \cdot \frac{y_{G'_2} \cdot \frac{1}{c\alpha_1 \cdot \sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} - \frac{\operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f}{-x_{G'_2} \cdot s\alpha_1 - y_{G'_2} \cdot \frac{c\alpha_1 \cdot \operatorname{tg}\theta_1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} - \frac{1}{\sqrt{\operatorname{tg}^2\alpha_1 + 1 + \operatorname{tg}^2\theta_1}} \cdot f} - l_1 - l_2 - c_2 \\ 1 \end{pmatrix} \quad (8.9)$$

Componentă după x a erorii de poziție nu există, adică:

$${}^{0_c} \Delta p_{x_{O_{Rc}}} = 0 \quad (8.10)$$

Pentru determinarea mărimilor de corecție care se necesită a se aplica la nivelul fiecărei c.c.c. se utilizează relațiile (5.75).

Prin urmare, la prima c.c.c. corecția care trebuie aplicată va fi:

$${}^{0_c} \underline{e}_{O_{Rc}} = D \cdot \frac{x_{G'_2} \cdot c\alpha_1 - y_{G'_2} \cdot \frac{s\alpha_1 \cdot tg\theta_1}{\sqrt{tg^2\alpha_1 + 1 + tg^2\theta_1}} - \frac{tg\alpha_1}{\sqrt{tg^2\alpha_1 + 1 + tg^2\theta_1}} \cdot f}{-x_{G'_2} \cdot s\alpha_1 - y_{G'_2} \cdot \frac{c\alpha_1 \cdot tg\theta_1}{\sqrt{tg^2\alpha_1 + 1 + tg^2\theta_1}} - \frac{1}{\sqrt{tg^2\alpha_1 + 1 + tg^2\theta_1}} \cdot f} + c_1 \quad (8.11)$$

iar pentru cea de-a doua c.c.c. corecția care se necesită a se aplica este:

$${}^{0_c} \underline{e}_{O_{Rc}} = l_1 + l_2 + c_2 - D \cdot \frac{y_{G'_2} \cdot \frac{1}{c\alpha_1 \cdot \sqrt{tg^2\alpha_1 + 1 + tg^2\theta_1}} - \frac{tg\theta_1}{\sqrt{tg^2\alpha_1 + 1 + tg^2\theta_1}} \cdot f}{-x_{G'_2} \cdot s\alpha_1 - y_{G'_2} \cdot \frac{c\alpha_1 \cdot tg\theta_1}{\sqrt{tg^2\alpha_1 + 1 + tg^2\theta_1}} - \frac{1}{\sqrt{tg^2\alpha_1 + 1 + tg^2\theta_1}} \cdot f} \quad (8.12)$$

8.3. Proiectarea și realizarea părților componente ale sistemului

8.3.1. Proiectarea și realizarea sistemului mecanic al subsistemului vizual de urmărire

Subsistemul vizual de urmărire are rolul de a urmări și determina în timp real poziția țintei solidar legată de efectorul final al robotului. Punctul de plecare pentru proiectarea și realizarea subsistemului vizual de urmărire l-a constituit structura mecanismului acestuia prezentată în figura 5.8.

S-a efectuat o proiectare asistată de calculator utilizând programul CATIA. În figura 8.7 se prezintă vederea din față a subsistemului vizual de urmărire, în figura 8.8 vederea de sus a acestuia, iar în figura 8.9 este reprezentată vederea din lateral a subsistemului vizual de urmărire.

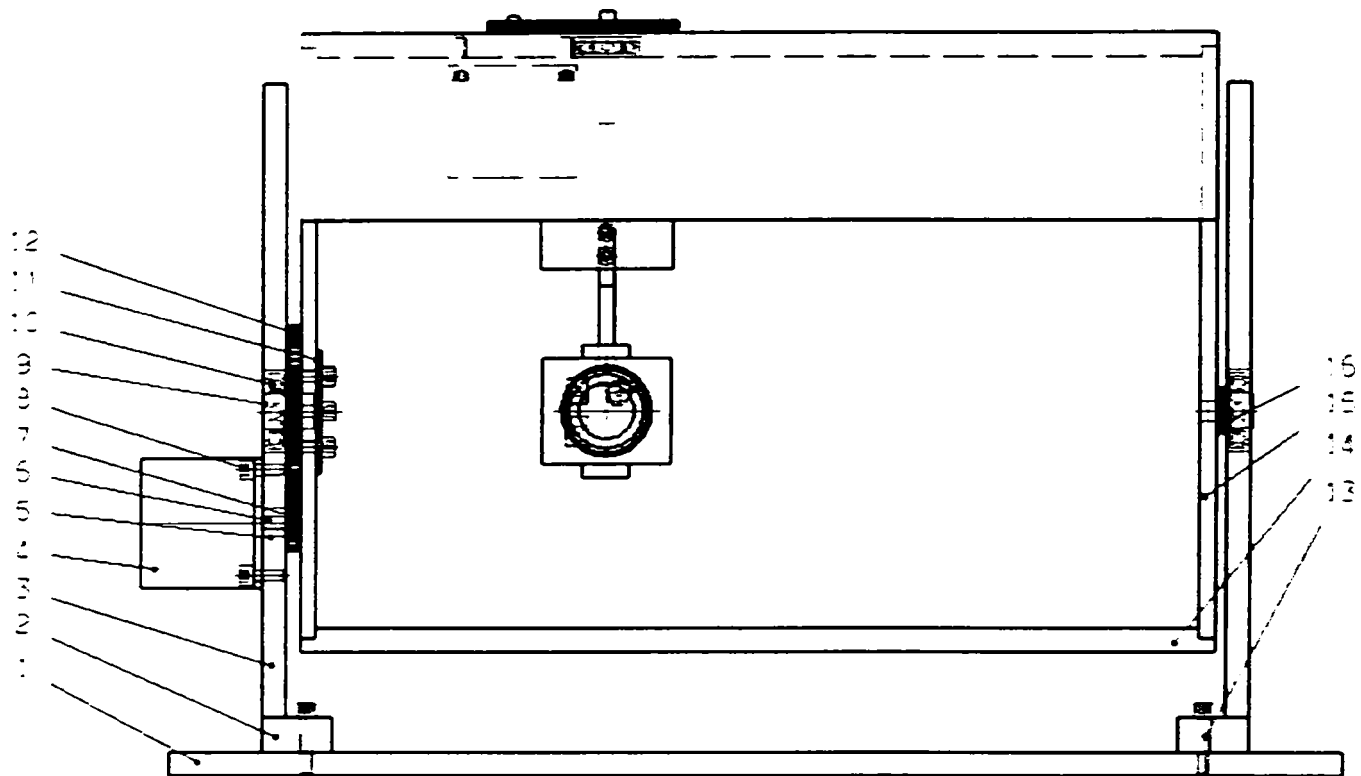
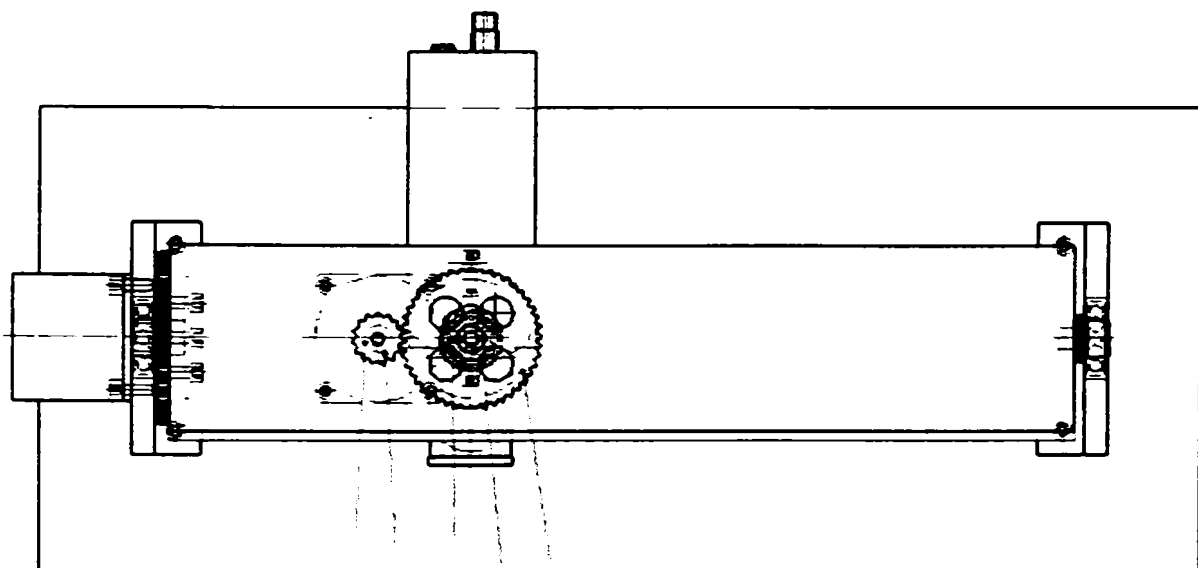


Fig.8.7. Vederea din față a subsistemului vizual de urmărire

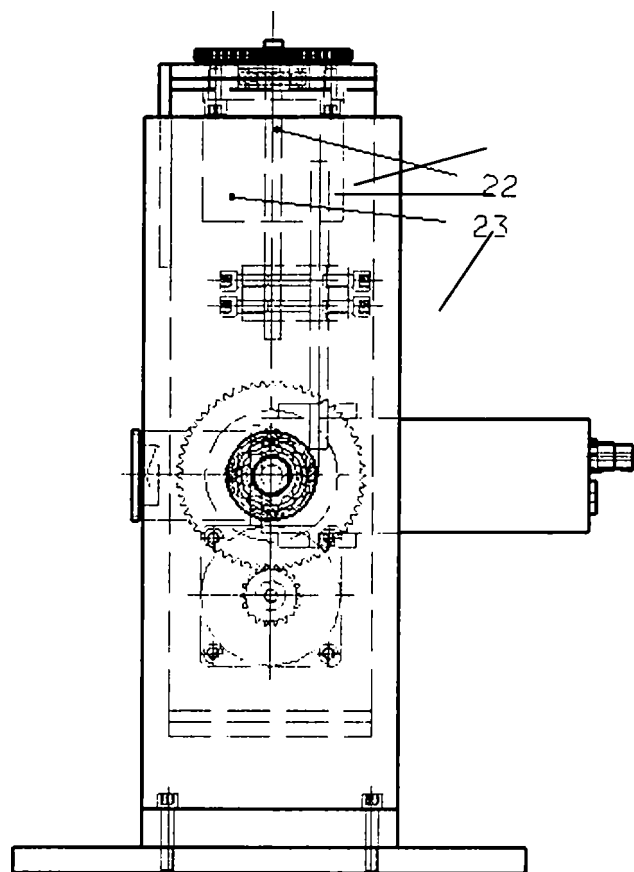


21 20 19 18 17

Fig.8.8. Vederea de sus a subsistemului vizual de urmărire

Subsistemul vizual de urmărire se compune din următoarele părți componente:

1, 2, 3 - plăci ce constituie suportul S;



- 4 - motor pas cu pas;
- 5, 16, 21 - bucșe;
- 6 - ax motor;
- 7, 12, 17, 20 - roți dințate;
- 8, 13, 24 - șuruburi de fixare;
- 9, 16, 22 - ax ;
- 10, 18 - rulmenți;
- 11 - flanșă;
- 14, 15 - plăci din componența cadrului C;
- 19, 21 - pane;
- 23 - ax;
- 25 - excentric;
- 26 - cameră CCD;

Fig.8.9. Vedere laterală a subsistemului vizual de urmărire

8.3.2. Proiectarea și realizarea sistemelor de acționare

8.3.2.1. Noțiuni introductive

În cadrul sistemului robot există două sisteme distincte de acționare – așa cum rezultă și din schema bloc a sistemului prezentată în figura 8.2:

- un sistem de acționare al robotului, constituit din câte un servomotorul de curent continuu și câte un reductor combinat de tip melc-roată melcată și transmisii cu curele dințate pentru fiecare c.c.c.;
- un sistem de acționare al subsistemului vizual de urmărire, constituit dintr-un motor pas cu pas și un reductor cu roți dințate pentru fiecare c.c.c. a subsistemului vizual;

În ceea ce urmează, se prezintă modalitatea de adaptare a sistemului de acționare al robotului pentru asigurarea conducerii globale sistemului robot, dar și sistemul de acționare a subsistemului vizual de urmărire proiectat și realizat de către autor.

8.3.2.2.Sistemul de acțiune al robotului

8.3.2.2.1.Structura sistemului de acțiune al robotului

A fost necesară adaptarea elementelor de acțiune a c.c.c. la cerințele necesare a fi atinse.

Elementul de acțiune pentru o c.c.c. se prezintă schematic în figura 8.10.

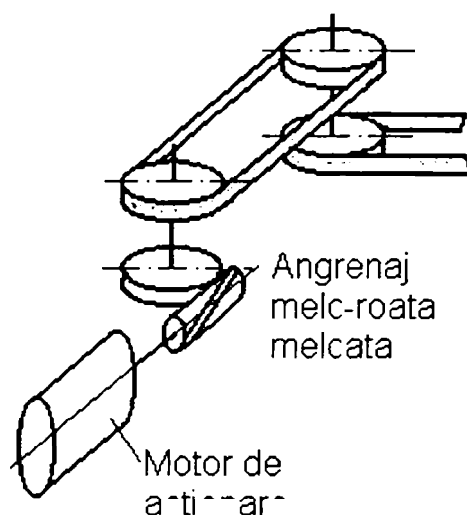


Fig.8.10. Elementul de acțiune a unei c.c.c. al robotului

8.3.2.2.2.Parametrii sistemului de acțiune ai robotului

Sistemul de acțiune a subsistemului robot prezintă pentru fiecare c.c.c. câte un servomotor de curent continuu, având următorii parametrii :

- tensiunea de alimentare: $U_n = 18$ [V];
- curenul maxim absorbit: $I_n = 3$ [A];
- cuplul util: $M_u = 0,0525$ [kgf cm];
- puterea utilă: $P_u = 24$ [W];
- turația nominală: $n = 6000$ [rot/min];

Parametrii de ai servomotorului au rezultat fie direct, fie prin calcul, fie prin determinări experimentale:

- rezistența circuitului rotoric al motorului s-a determinat utilizând un voltmetru: $R = 1,8$ [Ω];

- inductivitatea circuitului rotorului al motorului s-a determinat utilizând o punte RLC: $L = 1,5$ [mH];
- constanta motorului de curent continuu s-a determinat prin clacul, utilizând relația $K_1 = \frac{U_n - R \cdot I_n}{2 \cdot \pi \cdot n} \cdot 60 \left[\frac{V}{rad/s} \right]$; $K_1 = 0,02$ [V/rad/s];
- momentul de inerție efectiv, redus la arborele motorului este dat de relația (3.40), iar valoric este $J_{ef} = 0,99 \cdot 10^{-6}$ [kg m²], unde:
- J_a momentul de inerție a rotorului servomotorului de curent continuu [kg m²] $J_a = 0,339 \cdot 10^{-6}$ [kg m²];
- J_m reprezintă momentul de inerție a cuplajului și a părții din reductor cuplată de rotorul motorului de curent continuu - melcul [kg m²] $J_m = 0,097$ [kg m²];
- J_e reprezintă momentul de inerție a elementului dispozitivului de ghidare redus la arborele motorului, care este acționat de cupla respectivă [kg m²] $J_e = 3562,5 \cdot 10^{-6}$ [kg m²];
- raportul de transmitere a reductorului $\mu = 120$;
- coeficientul global de frecare vâscoasă B_{ef} [Nm/rad/s] $B_{ef} = 14,57 \cdot 10^{-6}$ [Nms/rad] s-a determinat conform metodologiei prezentate în anexa A2.

Determinarea coeficienților de frecare vâscoasă și de frecare uscată a fost necesară pentru a putea realiza modelul pentru acționarea cu servomotoarele de curent continuu dar și pentru a putea simula funcționarea lui. Rezultatele obținute în cadrul acestui paragraf s-au folosit în capitolul 7.

Efectele dinamice datorate transmisiei mecanice și sistemului de acționare sunt importante, datorită creșterii complexității modelului dinamic al robotului.

Dacă se utilizează reductoare rezultă o amplificare a efectelor frecării vâscoase, dar și a frecării uscate.

Determinarea coeficienților frecării vâscoase și uscate utilizează modelul simplu al frecării dat de relația (A2.1) model prezentat în anexa A.2, unde se prezintă și modul de lucru pentru determinarea coeficienților frecării vâscoase și uscate.

Coeficientul de frecare vâscoasă B_a și coeficientul de frecare uscată M_F vor avea valori diferite în cele două sensuri și vor fi:

$$B_a = 14,786 \cdot 10^{-6} \text{ [Nms/rad] pentru deplasarea spre stânga;}$$

$$B_a = 14,356 \cdot 10^{-6} \text{ [Nms/rad] pentru deplasarea spre dreapta;}$$

$M_F = 0,019258$ [Nm] pentru deplasarea spre stânga;

$M_F = 0,019512$ [Nm] pentru deplasarea spre dreapta.

8.3.2.3. Sistemul de acționare a subsistemului vizual de urmărire

Pentru sistemul de acționare a subsistemului vizual de urmărire s-a necesitat conceperea și realizarea efectivă a întregului sistem de acționare. Modul de amplasare a motoarelor și reductoarelor pe structura mecanică a subsistemului vizual de urmărire rezultă din figurile 8.7, 8.8 și respectiv 8.9.

Elementul de acționare a unei c.c.c. a subsistemului vizual de urmărire conține un motor pas cu pas cu patru înfășurări, având următorii parametrii :

- tensiunea de alimentare: $U = 5$ [V];
- curentul maxim absorbit: $I_n = 3$ [A];
- rezistența măsurată a unei faze: $R = 1,49$ [k Ω];
- inductivitatea proprie medie: $L = 17,9$ [mH].

8.3.3. Proiectarea și realizarea sistemului de comandă a sistemului robot

8.3.3.1. Noțiuni introductive

Rolul sistemului de comandă a sistemului robot pe lângă rolul de a comanda conform programului mișcarea sistemului mecanic este acela de a preleva informațiile referitoare la poziția punctului caracteristic al robotului de la subsistemul vizual de urmărire, de a compara aceste poziții cu pozițiile programate și printr-o analiză inversă aplicată erorilor astfel obținute de a obține mărimile corecțiilor situărilor relative a elementelor c.c.c. Comanda de corecție a situării relative a elementelor c.c.c. se comunică elementului de acționare aferent c.c.c. împreună cu comanda corespunzătoare programului de executat.

Prin urmare, sistemul de comandă a unui sistem robot care prezintă conducere globală utilizând în acest sens informațiile furnizate de senzori vizuali trebuie să îndeplinească următoarele funcții:

- 1) achiziționarea și procesarea informațiilor furnizate de senzorul vizual;
- 2) comanda subsistemului vizual de urmărire în vederea urmăririi țintei în timp real;
- 3) calculul pozițiilor atinse de țintă;

- 4) compararea pozițiilor realizate cu cele programate, determinarea vectorului de eroare de poziționare a țintei;
- 5) determinarea corecțiilor situărilor relative a elementelor c.c.c. a robotului;
- 6) Comanda situării relative a elementelor c.c.c. ale robotului cu un semnal rezultat din suprapunerea corecțiilor cu comenzile programate pentru elementele de acționare ale fiecărei c.c.c. a robotului.

Pentru realizarea funcțiilor 1, 2 și respectiv 6 ale sistemului de comandă a sistemului robot s-a necesitat conceperea și realizarea atât a unor dispozitive hardware, cât și a unor programe specifice.

Funcțiile 3, 4 și 5 sistemului de comandă a sistemului robot sunt ușor de realizat din punct de vedere al implementării lor în cadrul programului de comandă. Se necesită doar calculul relațiilor (8.2), (8.6), (8.9), (8.11) și respectiv (8.12).

8.3.3.2. Structura sistemului de comandă a sistemului robot realizat

Rolul sistemului de comandă a sistemului robot este acela de a preleva informațiile referitoare la poziția punctului caracteristic al robotului de la subsistemul vizual de urmărire, de a compara aceste poziții efectiv realizate cu pozițiile programate și printr-o analiză inversă aplicată erorilor astfel obținute de a obține mărimile corecțiilor situărilor relative a elementelor c.c.c. Comanda de corecție a situării relative a elementelor c.c.c. se comunică elementului de acționare aferent c.c.c. împreună cu comanda corespunzătoare programului de executat.

Prin urmare se observă necesitatea realizării integrării a trei funcții în cadrul sistemului de comandă și anume: achiziționarea și procesarea informațiilor furnizate de senzorul vizual; comanda subsistemului vizual de urmărire în vederea urmăririi țintei în timp real; comanda c.c.c. ale robotului cu un semnal rezultat din suprapunerea corecțiilor ce trebuiesc realizate peste comenzile programate a mișcărilor relative a elementelor c.c.c. a robotului.

Schema de principiu a sistemului de comandă a sistemului robot se prezintă în figura 8.17.

În vederea achiziționării și procesării informațiilor furnizate de senzorul vizual s-a utilizat o placă de achiziție de imagini – Matrox Meteor ale cărei caracteristici se prezintă în anexa A4.

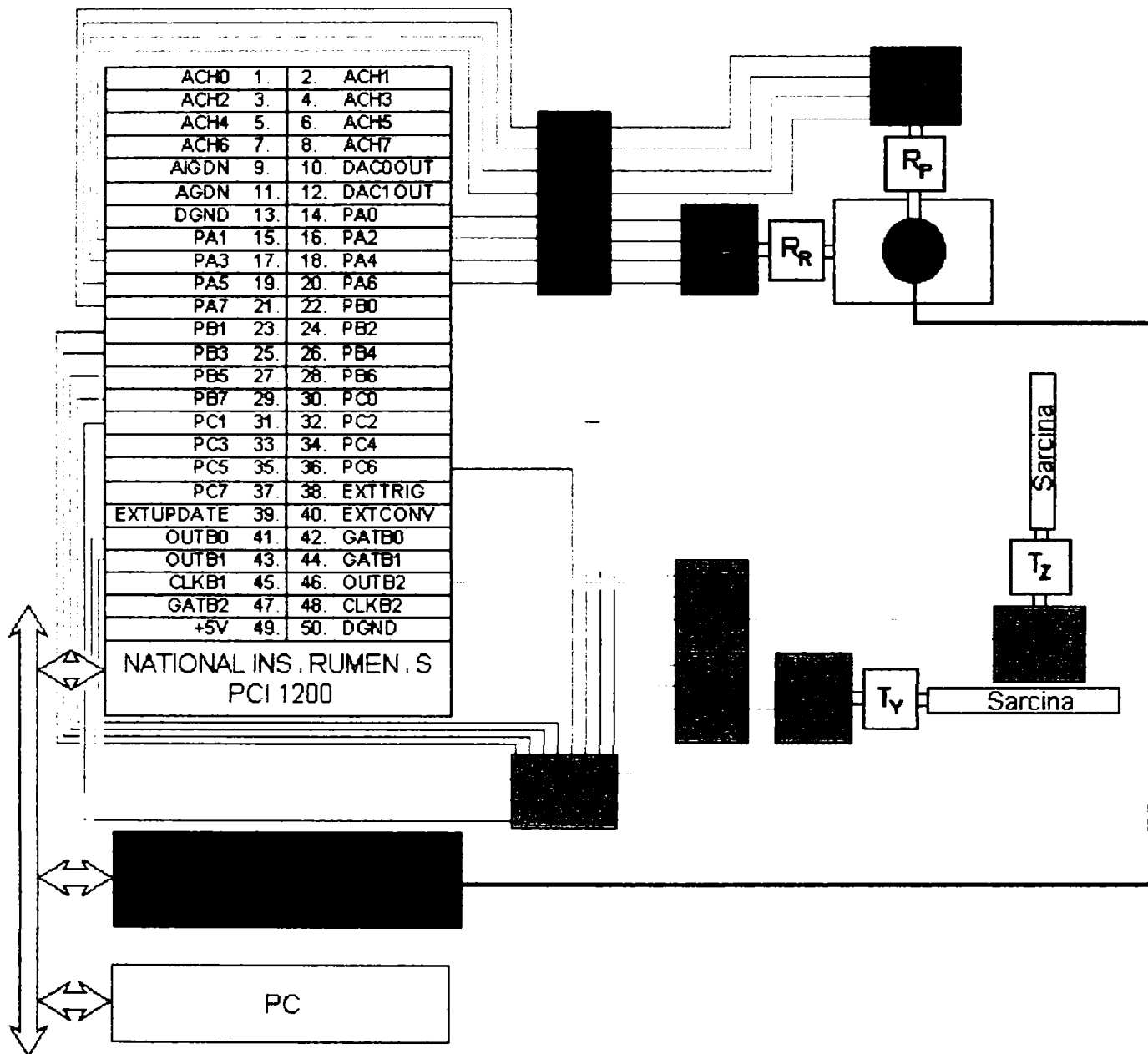


Fig.8.17. Schema de principiu a sistemului de comandă a sistemului robot

Atât pentru comanda subsistemului vizual de urmărire în vederea urmăririi țintei în timp real, cât și pentru comanda c.c.c. ale robotului s-a utilizat o placă de achiziție a firmei National Instruments de tip PCI 1200, a cărei caracteristici se prezintă în anexa A5.

Elementele componente ale sistemului de conducere a sistemului robot sunt următoarele:

1. *Calculatorul PC*, care rulează programul de conducere globală a sistemului robot și care are rolul de a genera pozițiile programate pentru punctul caracteristic al robotului, de a preleva informațiile de la placa de achiziție de imagini Matrox Meteor, pe baza cărora să calculeze pozițiile atinse de țintă și prin compararea pozițiilor realizate cu cele

programate să determine vectorului de eroare de poziție a țintei pe care să îl convertească în corecții necesare a fi realizate la nivelul fiecărei c.c.c. a robotului. Aceste corecții sunt transmise plăcii de achiziție PCI 1200.

2. *Placa de achiziție de imagini Matrox Meteor* cu rol de a prelua informația de la senzorul vizual matriceal de tip CCD și de a o transmite sub un format accesibil calculatorului.

3. *Senzorul vizual matriceal de tip CCD* achiziționează informația cu privire la poziția țintei în timpul mișcării robotului.

4. *Motorul pas cu pas M_R* care generează mișcarea roll a subsistemului vizual de urmărire.

5. *Reductorul R_R* aferent c.c.c. care realizează mișcarea de rotație după axa orizontală.

6. *Motorul pas cu pas M_P* care generează mișcarea roll a subsistemului vizual de urmărire.

7. *Reductorul R_P* aferent c.c.c. care realizează mișcarea de rotație după axa verticală.

8. *Interfețele de comandă I_R, I_P* care asigură amplificarea semnalului de comandă generat de placa PCI 1200 la un semnal corespunzător comenzii motoarelor pas cu pas. Prezentarea constructivă și funcțională a interfețelor se prezintă în paragraful 8.4.3.

9. *Servomotorul de curent continuu M_y* care pune în mișcare c.c.c. aferentă axei O_0y_0 atașată robotului.

10. *Transmisia mecanică T_y* acționată de la motorul M_y , care se compune din reductor melc-roată melcată și două transmisii cu curele dințate.

11. *Servomotorul de curent continuu M_z* care pune în mișcare c.c.c. aferentă axei O_0z_0 atașată robotului.

12. *Transmisia mecanică T_z* acționată de la motorul M_z , care se compune din reductor melc-roată melcată și două transmisii cu curele dințate.

13. *Interfețe de comandă I_y, I_z* a servomotoarelor de curent continuu M_y, M_z . Prezentarea constructivă și funcțională a interfețelor se prezintă în paragraful 8.4.4. Aferent acestor interfețe a fost necesară realizarea unei interfețe suplimentare pentru a se putea controla viteza de deplasare a servomotorului de curent continuu.

De la placa de achiziție NI PCI 1200 s-au utilizat ieșirile digitale PA0- PA7 pentru comanda motoarelor pas cu pas prin intermediul interfețelor de comandă I_R, I_P , ieșirile

digitale PB0, PB2, PB4, PB6 pentru comanda sensurilor servomotoarelor de curent continuu, iar pentru modificarea factorului de umplere în vederea variației vitezelor unghiulare a servomotoarelor s-au utilizat circuite contor/temporizator programabile 8253.

8.3.3.3. Interfață pentru comanda motorului pas cu pas

Semnalul de comandă pentru conducerea motoarelor pas cu pas din componența sistemului de acționare a subsistemului vizual de urmărire este emis de către placa de achiziție PCI 1200. Când programul de conducere globală sesizează că imaginea țintei părăsește dreptunghiul de interes din planul imagine, va genera o comandă de mișcare a cadrului și deci implicit și a senzorului vizual în vederea readucerii imaginii țintei în zona dreptunghiului de interes.

Pentru comanda celor patru înfășurări ale unuia dintre motoarele pas cu pas, s-a necesitat realizarea unei interfețe care să preia semnalele de la porturile digitale și să le amplifice corespunzător pentru a se obține un semnal adecvat comenzii înfășurărilor motorului. Prin urmare pentru fiecare înfășurare a unui motor pas cu pas sunt necesare câte un circuit de amplificare a semnalului, deci pentru un motor sunt necesare patru astfel de celule de amplificare. Acestea sunt prezentate în figura 8.18.

Din figura 8.18 se disting două surse de alimentare a schemei:

- sursa de mică putere de 5V, utilizată ca tensiune de comandă;
- sursa de putere ($5V_{ext}$), pentru alimentarea înfășurării motorului pas cu pas.

Cele două surse de alimentare sunt separate galvanic prin intermediul optocuplorului TIL113, pentru protejarea plăcii de achiziție de curenții mari din circuitul de acționare.

Modul de funcționare al unei celule este următorul: comanda excitației înfășurării motorului pas cu pas este realizată prin intermediul unui program de placa (de achiziție), care ține evidența ordinii de excitație a bobinelor fiecărui motor; ea se realizează prin apariția unui semnal de 5V (unu logic) la pinul corespunzător bobinei excitate; acest semnal este preluat prin rezistorul R11 la baza tranzistorului Q11 (BC171) care funcționează în schemă în regim de comutator; apariția tensiunii de 5V provoacă saturarea joncțiunii colector-emitor a tranzistorului Q11 și punerea la masă a catodului diodei luminescente Q12 din componența optocuplorului; rezistorul R12 are rolul de

Intrarea în conducție a tranzistorului Q13 din interiorul optocuplorului permite apariția pe emitorul acesteia a tensiunii de aproximativ 4,4V (5V-0,6V tensiunea colector-emitor a fototranzistorului). Divizorul rezistiv R13-R14 va determina apariția la baza tranzistorului Q14 (BD683, de alimentare a înfășurării bobinei de excitație L1 a motorului pas cu pas) a unei tensiuni de aproximativ 4V, $\left(U_{R14} = \frac{R14}{R13 + R14} U_{ext} \right)$, unde:

- U_{R14} reprezintă tensiunea la bornele rezistorului R14 (egală cu tensiunea bază-emitor a tranzistorului Q14);

- U_{ext} reprezintă tensiunea de alimentare a circuitului de acționare.

Această tensiune permite trecerea tranzistorului Q14 în saturație, deschizând joncțiunea colector-emitor. În acest fel al doilea capăt al înfășurării bobinei de excitație L1 va fi pus la masă, prin bobină circulând curentul necesar de excitație.

În schemă apar suplimentar dioda Q16 care permite blocarea supratensiunilor cauzate de efectul de autoinducție, care apare în bobina înfășurării de excitație la parcurgerea ei de un curent, în scopul prevenirii deteriorării tranzistorului de acționare Q14, respectiv grupul alcătuit din dioda luminiscentă Q15 și rezistorul de limitare a curentului prin aceasta, cu rol de semnalizare a funcționării tranzistorului Q14, montată în paralel cu bobina L1, în circuitul de colector al tranzistorului Q14.

Neapriinderea lui Q15 este cauzată sau de întreruperea joncțiunii bază-colector a tranzistorului Q14 (joncțiune întreruptă=tranzistor stricat) sau de nealimentarea bazei lui Q14 cu tensiune, cauzată de: deteriorarea lui Q13 sau Q12 (în oricare din cele două cazuri se înlocuiește TIL113) sau lipsa tensiunii de alimentare a circuitului de comandă (se va măsura tensiunea de 5V de comandă sau lipsa ei).

De asemenea, aprinderea continuă a lui Q15 indică o joncțiune străpunsă colector-emitor la Q14, Q13 sau Q11.

Ultima utilitate a lui Q15 este aceea că dacă cumva la o comandă corectă a circuitului de acționare motorul nu se rotește, prin acționarea manuală a succesiunii de comandă (sau oricum cu frecvență suficient de mică încât să facă vizibilă aprinderea și stingerea LED-urilor Qx5 (cu x de la 1 la 4)), aprinderea și stingerea LED-ului Q15 indică funcționarea corectă a celulei de comandă și acționare respectiv deteriorarea bobinei de excitație, care nu mai produce rotirea motorului.

Alimentarea de comandă se face de la o sursă stabilizată de tensiune de mică putere de 5V bine filtrată, la bornele capacitorului C1 (100 μ F/12Vcc), iar cea de acționare V_{ext} , de la o sursă de putere de 5V la bornele capacitorului C2.

Interfața s-a realizat deoarece nu se găsește în stare independentă pentru a fi cumpărată din comerț.

8.3.3.4. Interfețe pentru comanda motoarelor de curent continuu

Aționarea elementelor aferente c.c.c. ale robotului este realizată de servomotoare de c.c. În aval de fiecare servomotor există un reductor melc-roată melcată urmat de două transmisii cu curele dințate.

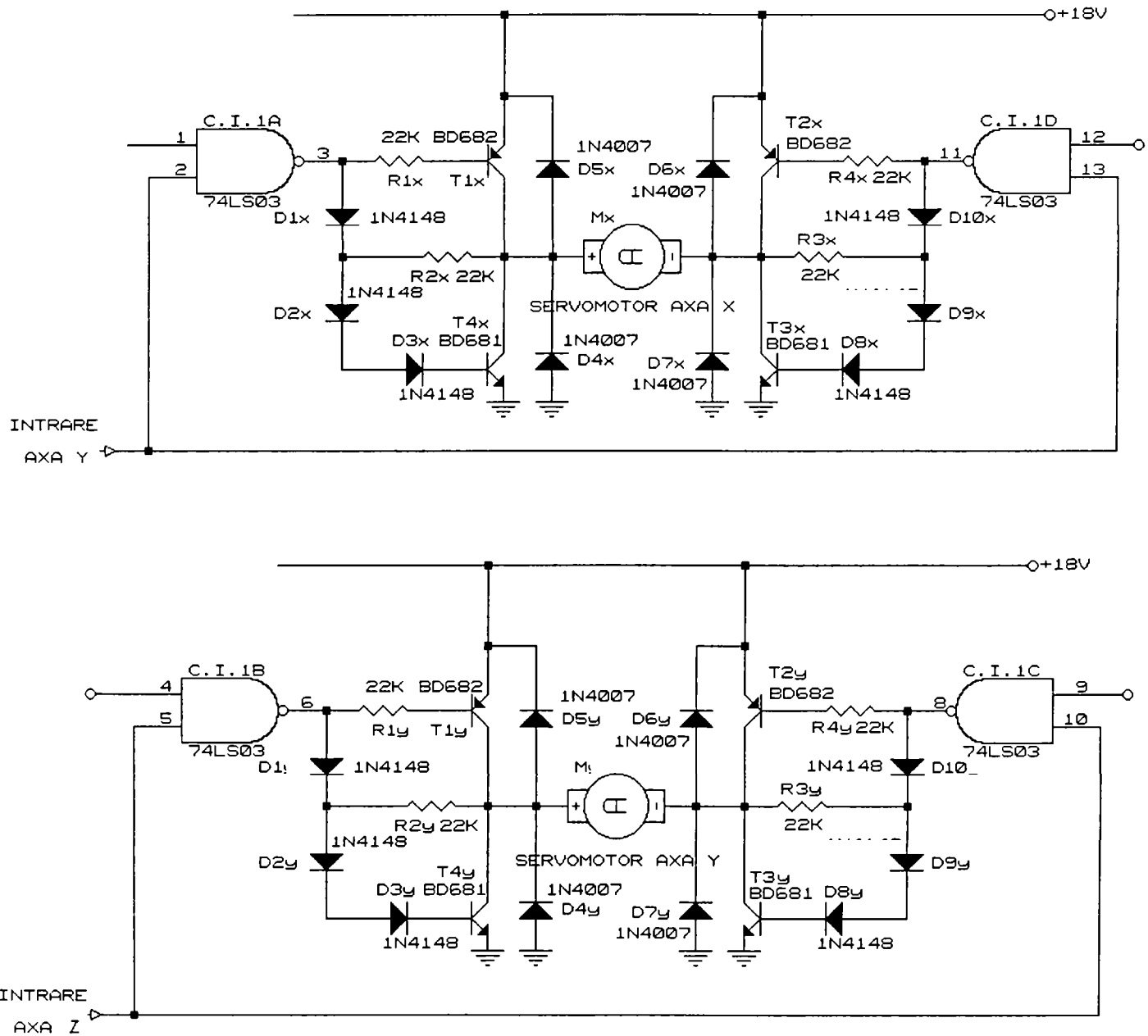


Fig.8.19. Schema circuitului de interfață de comandă a servomotoarelor de c.c.

Parametrii constructivi și de lucru ai servomotoarelor de curent continuu din componența sistemului de acționare a robotului sunt prezentați în paragraful 8.3.2.2.

Pentru a genera semnalul de comandă a celor două servomotoare de c.c. se utilizează placa de achiziție National Instruments PCI 1200 la care se atașează două plăci de interfață prezentate în figurile 8.19, respectiv 8.20.

Servomotorul de curent continuu care acționează elementele c.c.c. ale sistemului mecanic al robotului, conectat în diagonală de ieșire a punții cu tranzistoare, se poate roti în ambele sensuri.

Pentru a se obține sensul de rotație dorit se dau în succesiv comenzile pe intrările 1 și 12 ale circuitului integrat 74LS03 pentru obținerea unui sens, respectiv pe intrările 4 și 9 ale circuitului integrat pentru obținerea sensului de rotație opus. Tranzistoarele T1 și T3 respectiv T2 și T4 vor intra în conducție, în funcție de intrarea comandată, schimbându-se astfel sensul alimentării servomotorului de c.c. Diodele 1N4007 au rol de protejare a tranzistoarelor T1, T2, T3, respectiv T4.

În tabelele 8.1, respectiv 8.2 se prezintă modurile în care se necesită a se da comenzile pe intrările circuitului integrat 74LS03 pentru a pune în mișcare cele două servomotoare de c.c.

Tabelul 8.1. Modurile de comandă ale servomotorului care comandă axa Y

Nr. intrării comandate		
1	12	Servomotor axa Y
0	0	Motor blocat
0	1	Stânga
1	0	Dreapta
1	1	Interzis

Tabelul 8.2. Modurile de comandă ale servomotorului care comandă axa Z

Nr. intrării comandate		
4	9	Servomotor axa Z
0	0	Motor blocat
0	1	Stânga
1	0	Dreapta
1	1	Interzis

Semnalul "Intrarea axa" este furnizat de cea de-a doua interfață realizată de autor. Schema de principiu a acesteia se prezintă în figura 8.20. Prin variația factorului de umplere se variază turația servomotorului de c.c.

Pentru comanda celor două c.c.c. ale robotului sunt necesare două circuite contor/temporizator programabile 8253. Deoarece placa de achiziție PCI 1200 dispune de un astfel de circuit, se necesită utilizarea doar a unui al doilea circuit 8253.

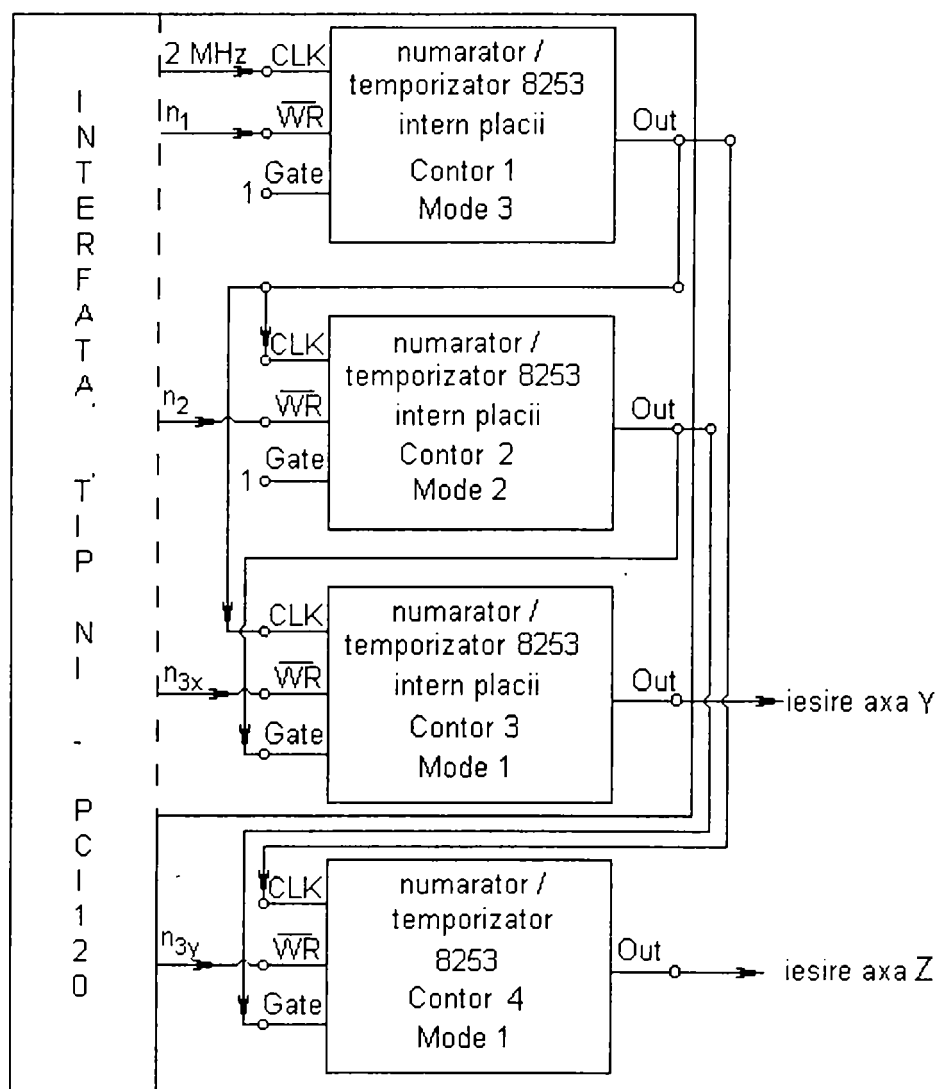


Fig.8.20. Schema interfeței care generează semnalele de comandă la axele Y și Z

Circuitul 8253, realizat în tehnologia NMOS pe o pastilă cu 24 terminale, ale căror specificații sunt prezentate în figurile 8.21, 8.22 și în tabelul 8.3, constituie un contor/temporizator programabil. El este organizat sub forma a trei contoare independente, de câte 16 biți, având asociată logica corespunzătoare pentru comunicația cu unitatea centrală de prelucrare și cu mediul exterior.

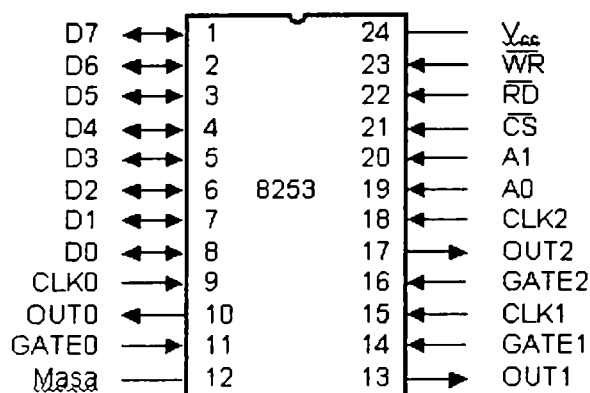


Fig.8.21. Schema terminalelor circuitului 8253

Tabelul 8.3. Semnificația terminalelor circuitului 8253

Nume terminal	Descriere	Tip
D0-D7	Magistrala de date	Bidirecțional, trei stări
CLK 0	Ceas contor 0	Intrare
OUT 0	leșire contor 0	leșire
GATE 0	Poarta contor 0	Intrare
CLK 1	Ceas contor 1	Intrare
OUT 1	leșire contor 1	leșire
GATE 1	Poarta contor 1	Intrare
CLK 2	Ceas contor 2	Intrare
OUT 2	leșire contor 2	leșire
GATE 2	Poarta contor 2	Intrare
A0, A1	Intrări selective contor	Intrări
CS	Selecție circuit	Intrare
RD	Comandă citire	Intrare
WR	Comandă scriere	Intrare
V _{CC}	Alimentare + 5 V	
Masa	Masă	

Schema bloc a circuitului, prezentată în figura 8.24 constă din mai multe componente: tamponul magistralei de date, logica scrie/citește, registrul cuvântului de comandă și cele trei contoare 0, 1 și 2.

Pentru comanda servomotoarelor de c.c. s-au folosit din cele șase moduri de lucru ale circuitului 8253 modurile 1, 2 și 3, care sunt prezentate în ceea ce urmează.

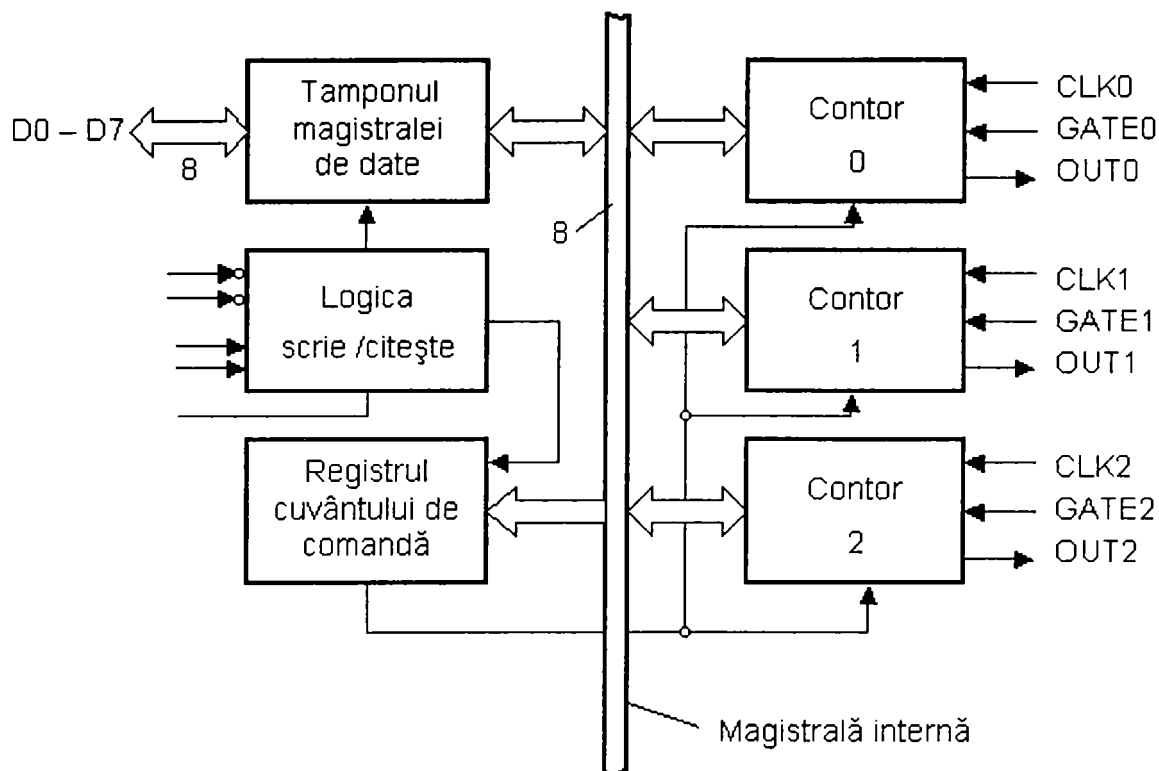


Fig.8.22. Schema structurii interne a circuitului 8253

Modul 1 este definit ca monostabil programabil. Ieșirea contorului va trece de la nivelul ridicat, la unul coborât numai după ce cuvântul de comandă și contorul său au fost încărcate și semnalul aplicat la poartă are o tranziție pozitivă. Această tranziție declanșează decrementarea care, la atingerea valorii finale (zero) va aduce la nivel ridicat ieșirea contorului selectat – fig.8.23.

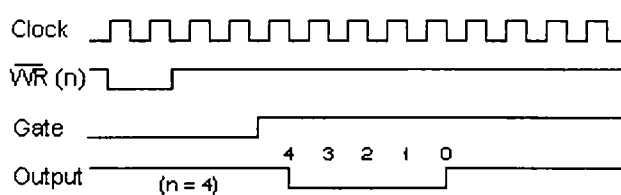


Fig.8.23. Modul 1 de funcționare a circuitului 8253

Dacă în timp ce ieșirea este la nivel coborât, o nouă valoare este înscrisă în contor, aceasta va afecta durata monoimpulsului, până la următoarea declanșare. Valoarea curentă a contorului poate fi citită în orice moment, fără a afecta monoimpulsul. Dacă intrarea la poarta "Gate" are o tranziție pozitivă, indiferent de faptul că decrementarea s-a terminat sau nu, contorul este relansat cu valoarea încărcată inițial. În cazul în care, pe

parcursul decrementării, se încarcă o nouă valoare în contor, acesta va fi luată în considerare la prima tranziție pozitivă a semnalului la poarta "Gate".

Modul 2 este definit ca generator de impulsuri divizate cu N. Ieșirea va fi forțată la nivel coborât, după încărcarea cuvântului de comandă și a numărătorului setat cu constanta N, pe o perioadă egală cu cea a semnalului de ceas. Perioada impulsurilor astfel generate va fi egală cu N – figura 8.24.

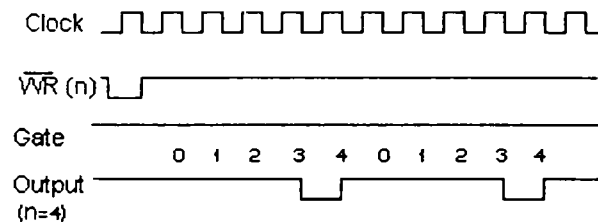


Fig.8.24. Modul 2 de funcționare a circuitului 8253

Dacă pe parcursul operării, intrarea "Gate" va fi forțată la nivel coborât, ieșirea "Output" va fi adusă la nivel ridicat, iar la revenirea lui "Gate" la un nivel ridicat, contorul se va decrementa din nou, de la valoarea inițială. Astfel, intrarea "Gate" poate fi folosită pentru sincronizarea contorului.

Modul 3 al circuitului este definit ca generator de impulsuri dreptunghiulare. El este similar cu modul 2, cu excepția faptului că ieșirea va rămâne la un nivel ridicat până la realizarea decrementării, conform cu $N/2$ sau $(N+1)/2$ intervale de ceas și la nivel coborât pentru celelalte $N/2$ intervale de ceas, după cum N este par sau impar.

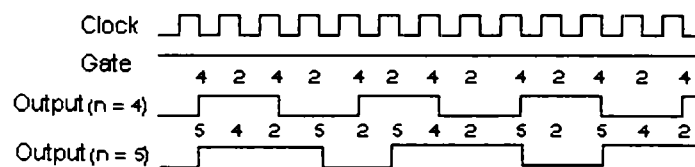


Fig.8.25. Modul 3 de funcționare a circuitului 8253

Aceasta se realizează (figura 8.25) prin decrementarea cu 2 pe frontul căzător al fiecărui semnal de ceas. Atunci când contorul ajunge la zero, ieșirea se modifică, contorul este încărcat cu valoarea inițială și procesul continuă, având un caracter repetitiv.

Primul numărător este programat astfel încât la ieșire se obține semnalul , cu perioada T_x care intra in al doilea numărător.

"Semnalul axă" se obține în modul următor:

- se programează intrarea \overline{WR} a contorului 1 a circuitului 8253 de pe placa de achiziție PCI 1200 la o valoare n_1 ;
- intrarea "Gate" a contorului 1 a circuitului 8253 de pe placa de achiziție PCI 1200 va fi forțată la nivelul ridicat;
- mărimea de ieșire "Output 1" va constitui semnalul de tact pentru toate celelalte contoare ale circuitelor 8253, perioada semnalului astfel obținut fiind

$$T_{Output_1} = \frac{n_1}{2 \cdot 10^6} \text{ [s]}; \text{ pentru contorul circuitului 8253 extern plăcii semnalul se}$$

culege de la ieșirea 41 - OUTB0 a blocului de conectori;

- intrarea \overline{WR} a contorului 2 a circuitului 8253 de pe placa de achiziție PCI 1200 se programează la o valoare n_2 , unde n_2 reprezintă numărul de tacti al perioadei semnalului care se dorește a se obține;
- intrarea "Gate" a contorului 2 a circuitului 8253 de pe placa de achiziție PCI 1200 va fi forțată la nivelul ridicat;
- mărimea de ieșire "Output 2" va constitui semnalul de intrare "Gate" pentru contorul 3 a circuitului 8253 de pe placa de achiziție PCI 1200 și pentru contorul utilizat de pe circuitul 8253 extern plăcii de achiziție; pentru contorul circuitului 8253 extern plăcii semnalul se culege de la ieșirea 43 – OUTB1 a blocului de conectori;
- intrarea \overline{WR} a contorului 3 a circuitului 8253 de pe placa de achiziție PCI 1200 se programează la o valoare n_{3y} , unde n_{3y} reprezintă numărul de tacti ai valorii zero a semnalului "Ieșire axă Y" dintr-o perioadă a semnalului de ieșire;
- semnalul de intrare pentru \overline{WR} a contorului utilizat a circuitului 8253 extern plăcii de achiziție se obține printr-o programare adecvată a ieșirilor digitale PC0-PC7 și PB5, PB7.

8.4. Elaborarea programului de conducere a robotului utilizând informații furnizate de senzori vizuali

Programul realizează conducerea sistemului robot a cărui structură realizată a fost prezentată în paragrafele anterioare ale acestui capitol.

Programul care realizează conducerea globală a sistemului robot este un program de conducere în poziție de tip multipunct. Prin particularizare, dacă numărul punctelor intermediare este zero se obține un program de conducere de tip punct cu punct.

Ordinograma programului prin care se realizează conducerea globală a sistemului robot se prezintă în figura 8.26.

Programul achiziționează - în mod repetat - imagini ale unei surse luminoase cu ajutorul unui senzor vizual matriceal de tip CCD, utilizând o placă de achiziție imagini de tip Matrox Meteor. În anexa A4 se prezintă datele esențiale despre placă.

În prima fază sunt efectuate inițializările:

- declarare headere sistem, necesare programului;
- declarare variabile, inițializare constante;
- declarare prototipuri funcții;
- rezervare memorie pentru plăcile de achiziție;
- inițializări plăci de achiziție;
- inițializare valori luminozitate, contrast, prag binarizare etc. cu valori prestabilite pe cale statistică;
- resetarea și inițializarea axelor CCD-urilor, prin aducerea lor "la zero";

După efectuarea acestor inițializări, se efectuează o prima achiziție de imagine (etapă care se include în inițializarea programului), iar rezultatul este afișat pe ecran.

Scopul acestei prime achiziții este atât de a oferi posibilitatea unei analize a valorilor prestabilite privitoare la parametrii imaginii (luminozitate, contrast, prag binarizare), cât și de a se obține un prim punct de referință pentru calculul poziției de start a țintei luminoase.

După inițializări se prezintă utilizatorului un meniu, prezentat în figura 8.27.

Opțiunile acestui meniu sunt următoarele:

- *ajustare cameră și axe CCD-uri;*
- *urmărire robot;*
- *ieșire.*

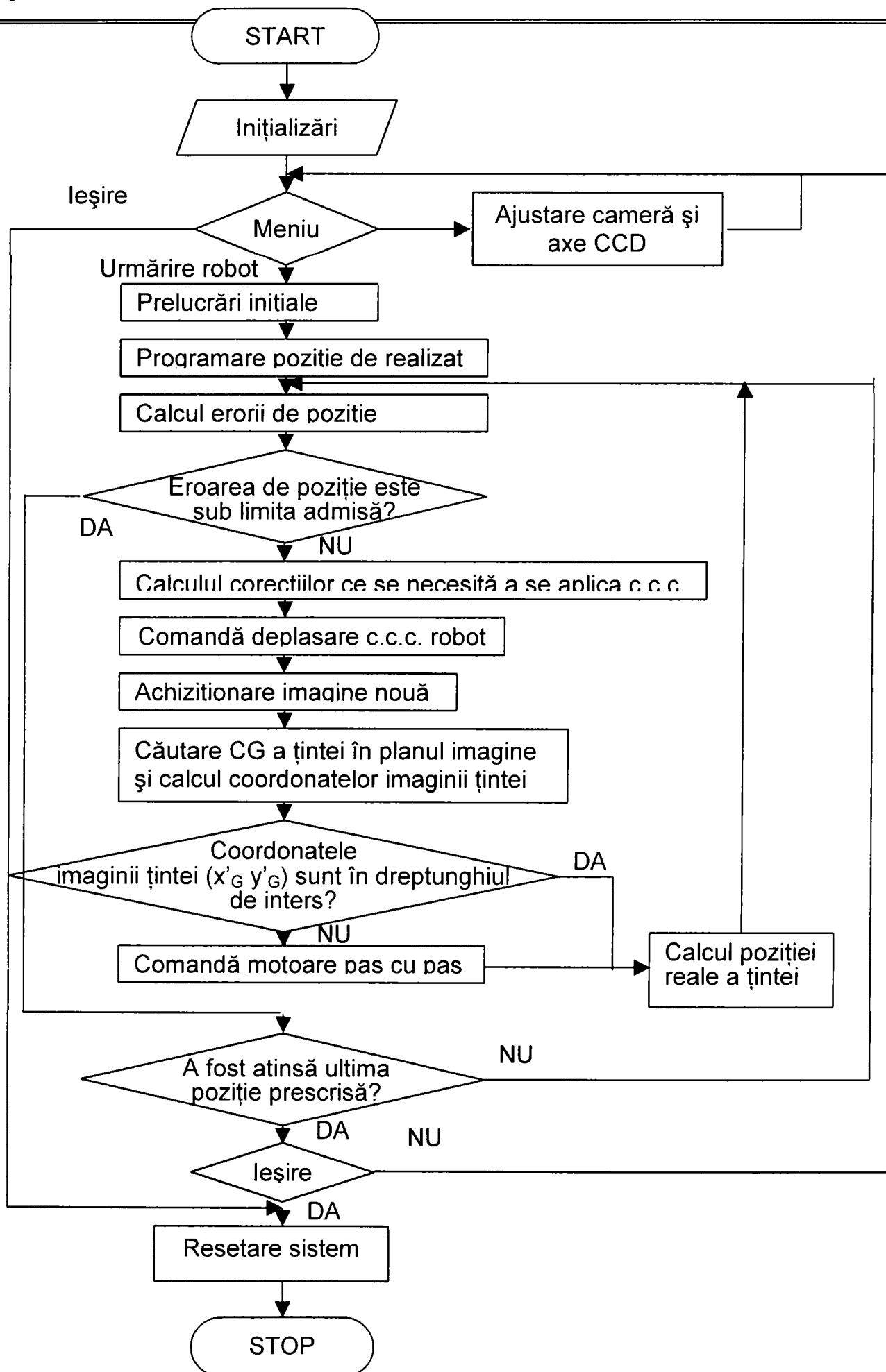


Fig. 8.26. Schema logică a programului de conducere globală a sistemului robot



Fig.8.27. Meniul programului

Opțiunea de ajustare imagine prezentată în figura 8.28 și axe CCD se referă la:

- ajustarea parametrilor imaginii cum ar fi: luminozitate, contrast, prag binarizare, în funcție de condițiile curente de lucru – imaginea achiziționată se prezintă în figura 8.29, iar în figurile 8.30 a), b), c), d) se prezintă imagini ale țintei la diferite setări ale luminozității și contrastului;

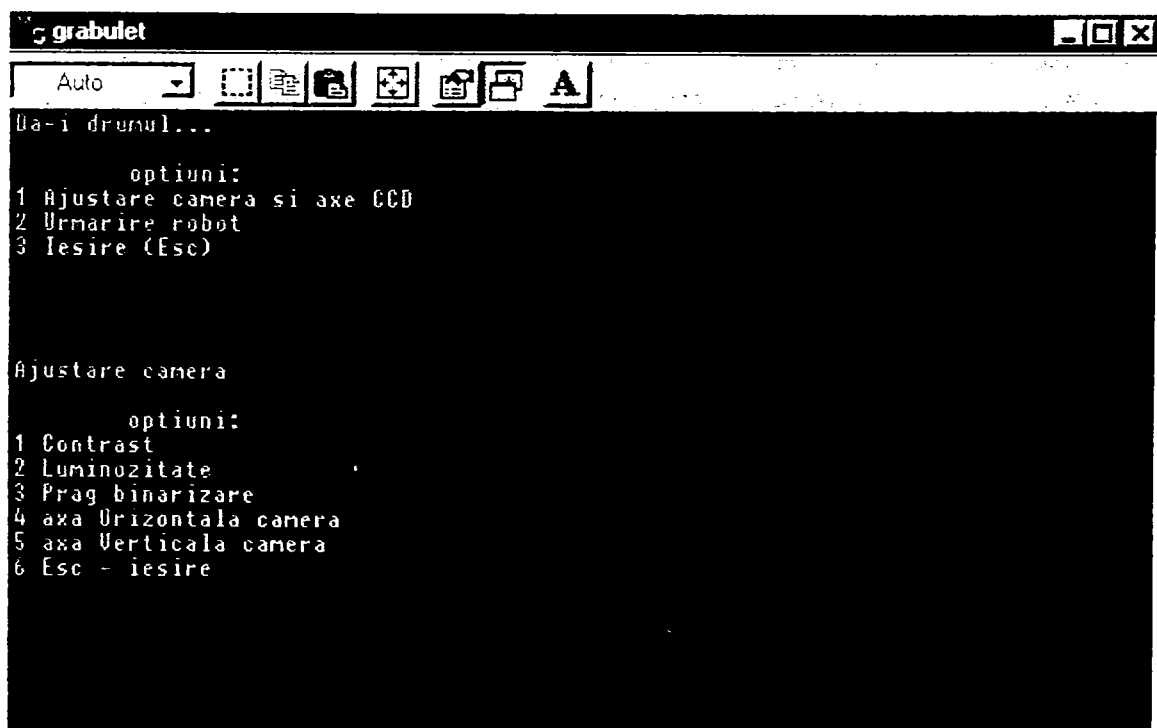


Fig.8.28. Meniul „ajustare camera”

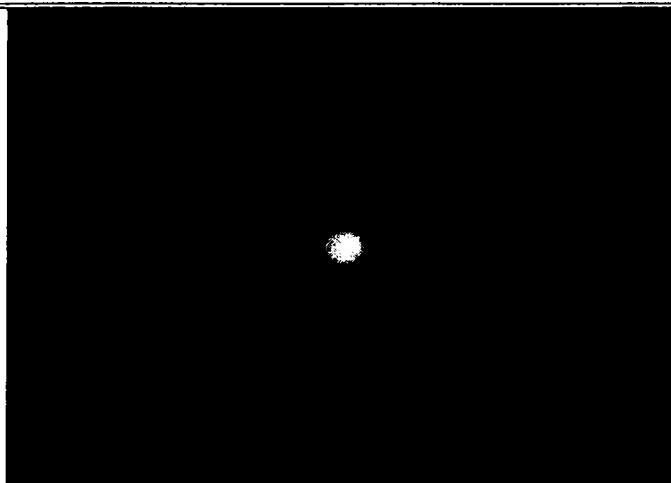


Fig.8.29. Imaginea țintei

- posibilitatea reglării manuale a unui nou punct de referință a axelor CCD-urilor, astfel încât să se pornească cu analiza țintei luminoase întotdeauna dintr-un punct central al ecranului, indiferent de situarea în spațiu a sursei luminoase la începutul experimentului;

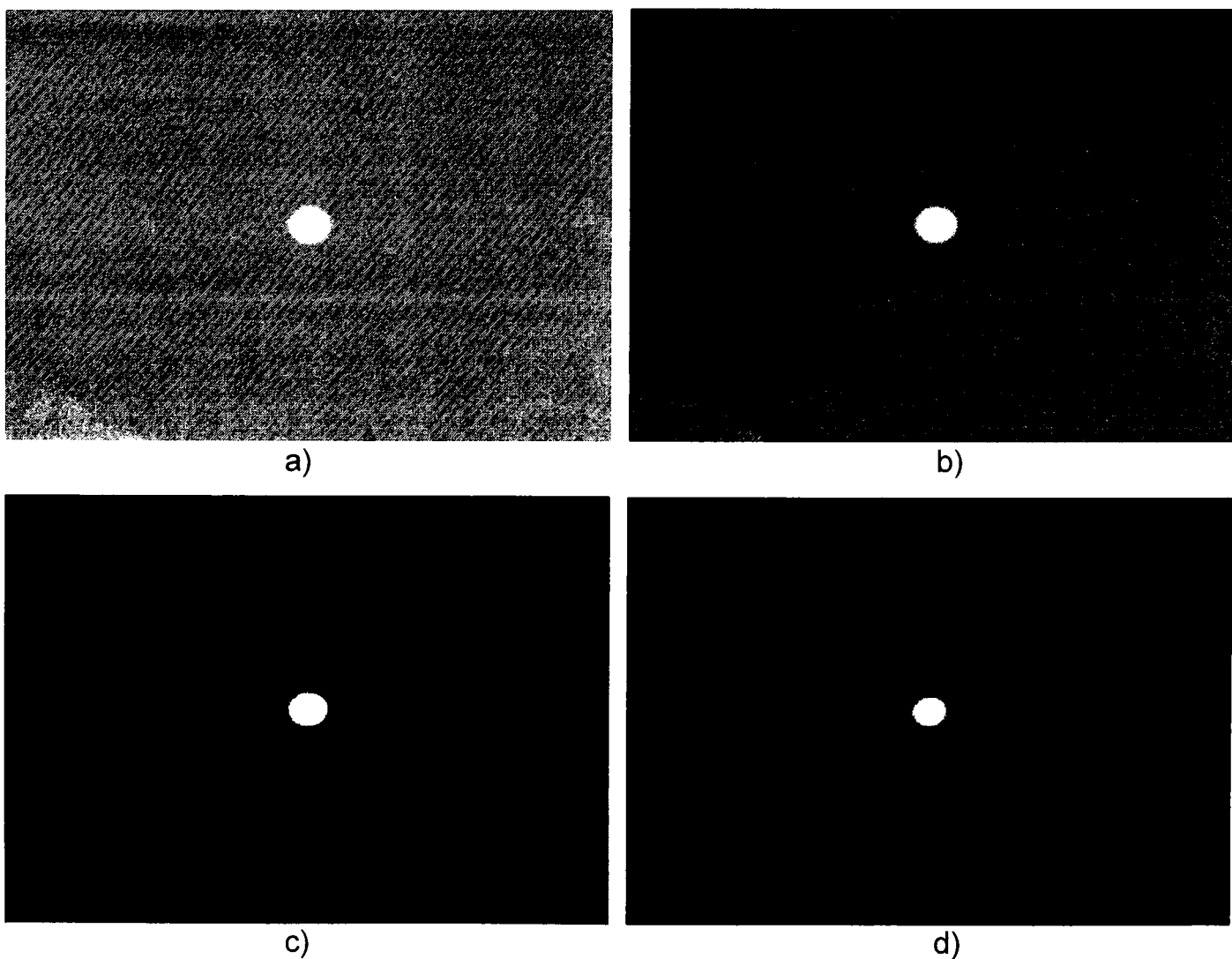


Fig.8.30. Imagini ale țintei la diferite setări ale luminozității, contrastului

În cadrul opțiunii din meniu de *urmărire robot* se realizează următoarele secvențe:

- se achiziționează o nouă imagine;
- se caută centrul de greutate al sursei luminoase în lanul imagine, în acest sens parcurgându-se următorii pași:
 - se binarizează imaginea – imaginea binarizată se prezintă în figura 8.31;

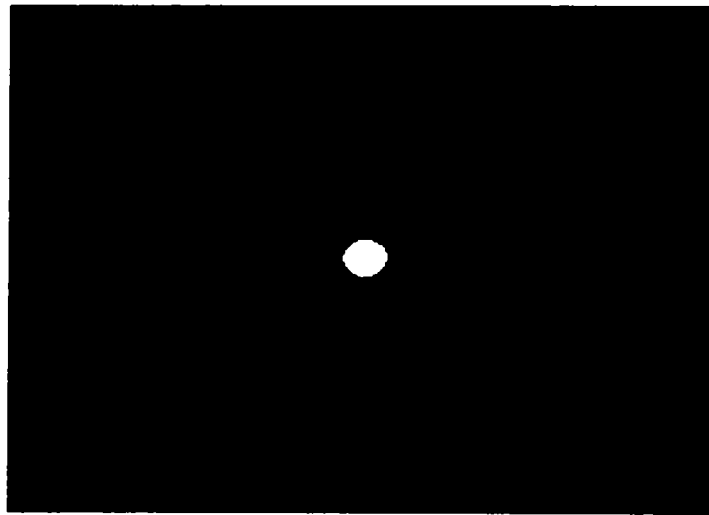


Fig.8.31. Imaginea binarizată

- se scanează imaginea până la găsirea primului pixel care este alb, scanarea realizându-se din două în două rânduri și controlându-se doar pixelii de rang impar (cu scopul de a mări viteza de căutare a țintei luminoase). Acest pixel va constitui centrul unei matrice pătratice, care va fi parcursă în întregime. Tot în scopul mării vitezei, aceasta căutare nu are loc în întreaga imagine achiziționată, ci doar în cadrul unui dreptunghi de dimensiune prestabilită – *dreptunghiul de interes* - al cărui punct central este centrul imaginii (dreptunghiul de interes este marcat în figura 8 32 cu o linie albă pe fondul negru al imaginii).

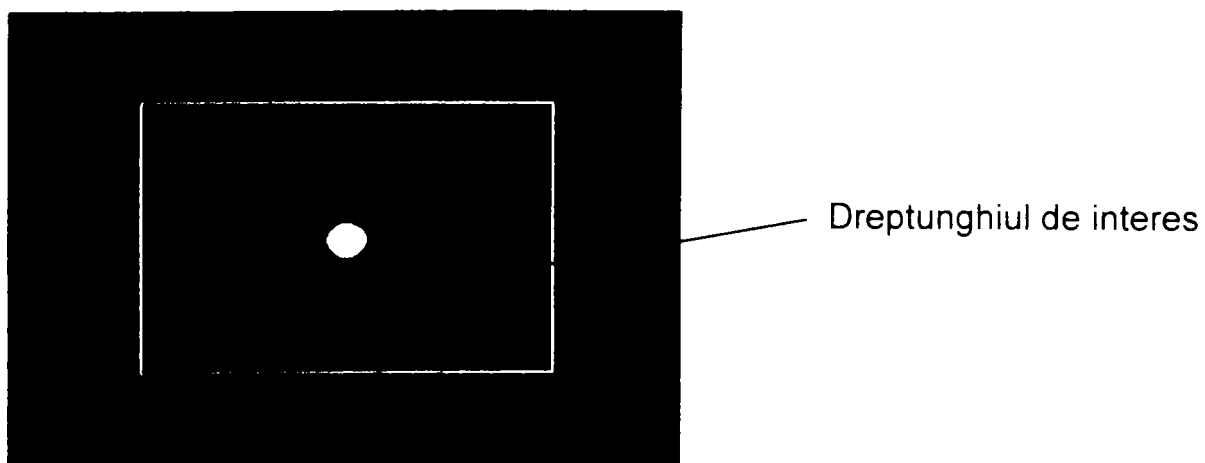


Fig 8 32. Reprezentarea dreptunghiului de interes pe imaginea binarizată

În cazul în care suprafața "albă" este mai mică decât o valoare prestabilită, avem de a face cu un zgomot și procesul de căutare continuă. Odată identificată poziția sursei luminoase în cadrul imaginii, se calculează coordonatele centrului de greutate al imaginii țintei, care vor constitui coordonate imaginii țintei (x'_G y'_G) față de sistemul de coordonate atașat planului imagine;

- dacă coordonatele imaginii țintei sunt în valoare absolută mai mari decât jumătate din dimensiunile dreptunghiului de interes, atunci se vor comanda c.c.c. ale sistemului mecanic al subsistemului vizual de urmărire astfel încât să se readucă imaginea țintei în dreptunghiul de interes din planul imagine. În figurile 8.33 a), b), c), d) se prezintă câteva imagini achiziționate peste care s-a suprapus dreptunghiul de interes; cu săgețile albe se indică sensul în care trebuie să se deplaseze imaginea țintei pentru a fi păstrată în dreptunghiul de interes;

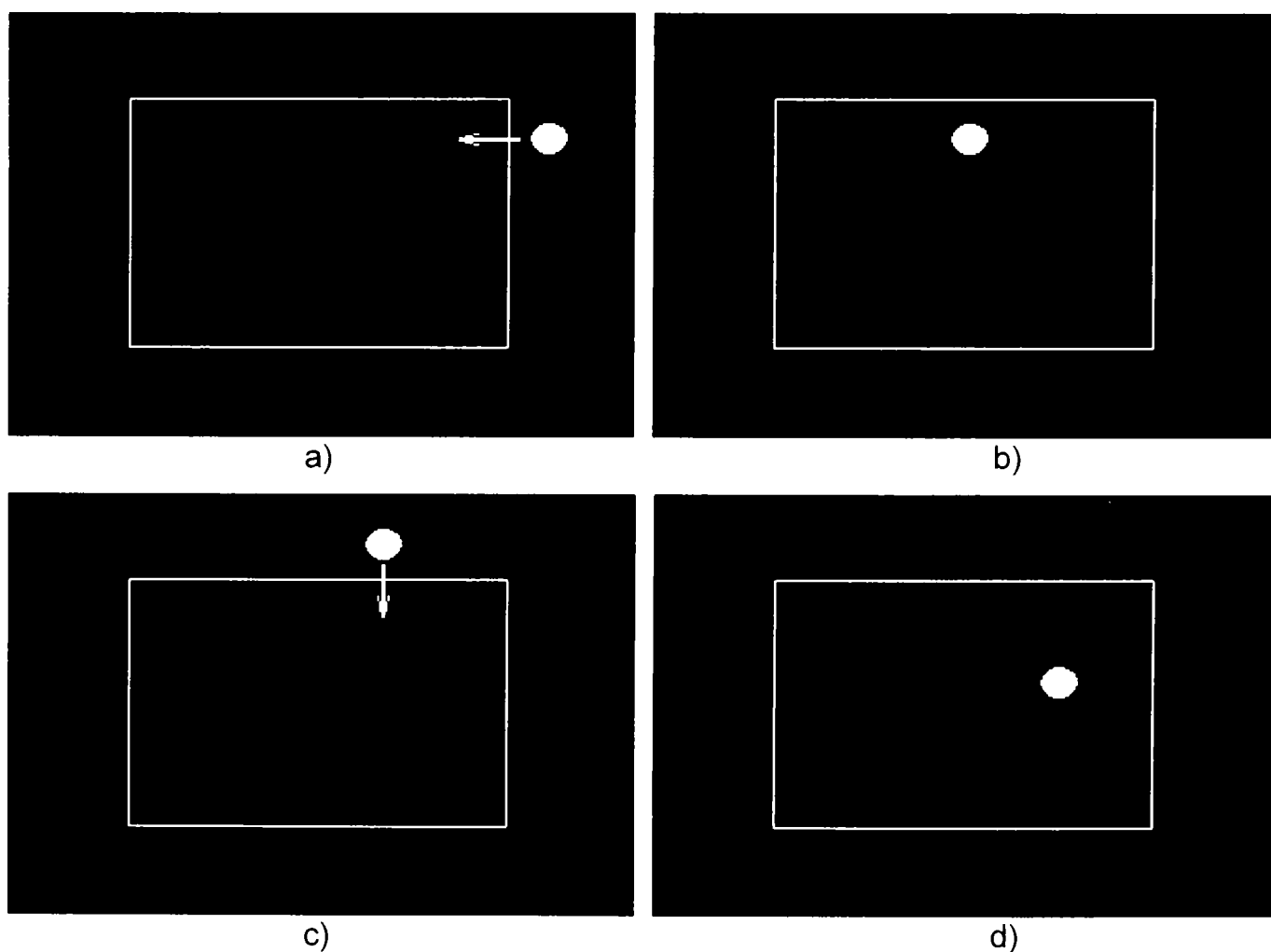


Fig.8.33. Imagini achiziționate în timpul derulării procesului de urmărire a țintei

- se afișează în mod permanent date privitoare la procesul de achiziție, cum ar fi: coordonatele imaginii țintei, timpul scurs de la începerea urmăririi, etc.
- pornind de la informația cu privire la coordonatele imaginii țintei, prin utilizarea relațiilor (8.5) se determină coordonatele punctului țintă în sistemul de coordonate atașat subsistemului vizual de urmărire;
- următorul pas al programului este acela de calcul al erorii de poziție cu ajutorul relației (8.9);
- în cazul în care eroarea de poziție calculată este mai mare decât o eroare limită admisă se trece la calculul corecțiilor ce se necesită a se aplica la c.c.c., corecții care se aplică peste următoarea comandă care se necesită a se realiza;
- dacă eroarea de poziție calculată este sub valoarea erorii limită admise înseamnă că s-a atins poziția curentă programată cu o precizie dată chiar de valoarea erorii limită admise; dacă poziția curentă este și ultima poziție de atins prescrisă atunci operatorului uman i se oferă posibilitatea ieșirii din program sau întoarcerea în meniu; dacă poziția curentă nu este ultima poziție prescrisă atunci programul așteaptă comanda pentru realizarea următoarei poziții de atins.

Opțiunea de ieșire oferă posibilitatea utilizatorului de a părăsi programul la acel moment.

De asemenea (nereprezentat în schema logică pentru a o face mai lizibilă), utilizatorul are posibilitatea în orice moment al execuției programului să oprească derularea acestuia, prin apăsarea tastei Esc.

În momentul în care s-au cules suficiente date, utilizatorul poate alege oprirea execuției programului. La acest moment se efectuează următoarele operațiuni:

- se resetează placa de achiziție și placa de comandă;
- se eliberează zonele de memorie rezervate;
- se redă controlul sistemului de operare.

Programul „grabulet” prezentat în ordinograma din figura 8.26. a fost realizat utilizând limbajul Visual C și se găsește în anexa A6.

8.5. Experimentări

Prin implementarea programului prezentat în paragraful precedent pe sistemul de conducere globală a robotului, s-au îndeplinit toate condițiile necesare pentru realizarea experimentărilor.

Desfășurarea experimentelor s-a realizat în două etape:

- în prima etapă, pentru un ciclu programat pentru robot s-a verificat modul în care subsistemul vizual de urmărire determină poziția reală a țintei pe parcursul deplasării acesteia, precum și valoarea vitezei critice a țintei, valoare până la care subsistemul vizual de urmărire reușește să urmărească ținta;
- în cea de a doua etapă s-au realizat experimentările pe sistemul de conducere globală a robotului, determinând timpii de corecție pentru fiecare secvență de program, cât și pentru program în totalitatea lui:
 - timpul de achiziție a imaginii de către senzorul vizual;
 - timpul de calcul necesar pentru determinarea informațiilor necesare despre poziția țintei;
 - timpul necesar generării mărimilor de comandă;
 - timpul necesar parcurgerii buclei de reacție a sistemului robot.

O imagine a modului de desfășurare a primei etape a experimentelor se prezintă în fotografia din figura 8.34.

Se observă în imagine robotul care are în efectorul său final ținta luminoasă, a cărei poziție, subsistemul vizual de urmărire trebuie să o determine în timp real. Pe monitorul calculatorului apare imaginea țintei achiziționată de senzorul vizual.

Pentru determinarea vitezei critice a EF – viteză până la care subsistemul vizual de urmărire reușește să urmărească deplasarea țintei atașată EF – i s-au programat robotului deplasări după axa O_0X_0 cu viteze crescătoare, obținute prin variația factorului de umplere.

Pentru determinarea vitezei de deplasare a țintei s-a utilizat montajul prezentat în figura 8.35.

Pentru determinarea vitezei unghiulare s-a realizat un program, care citește și scrie într-un fișier valorile unghiului de rotație a axului traductorului în timp. Ordinograma programului se prezintă în figura 8.36.

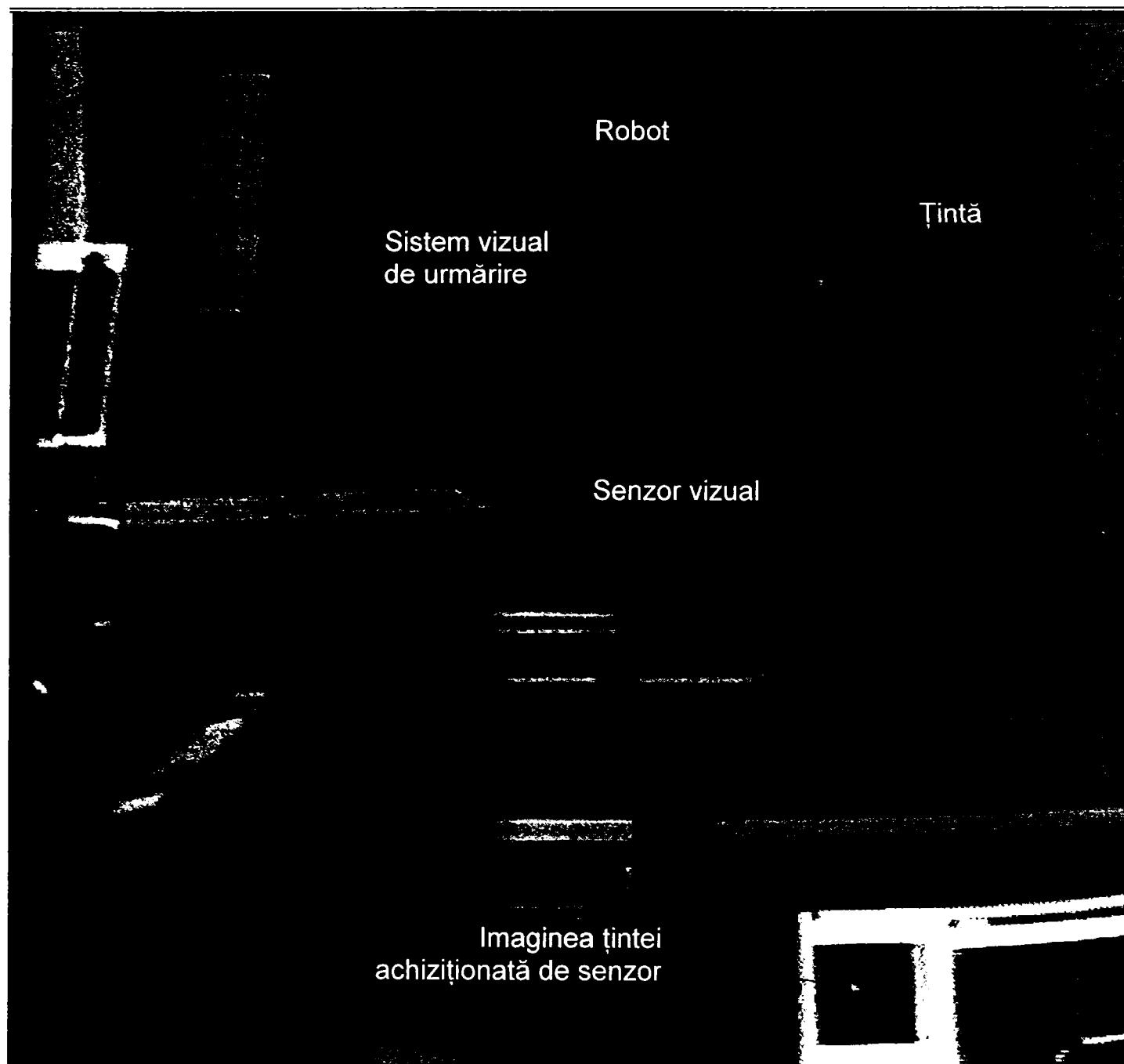


Fig.8.34. Vedere a sistemului robot experimental realizat

Pentru citirea variației în timp a coordonatei generalizate a servomotorului de c.c. se utilizează un traductor incremental de rotație TIRO 1000, sistemul de acționare a c.c.c. al robotului și o placă de achiziție PCI 1200 montată pe un calculator.

După inițializarea plăcii de achiziție PCI 1200 și inițializarea contorului care va realiza numărarea impulsurilor generate de traductor, se va deschide fișierul „Coorodnată_generalizată.dat”.

La apăsarea unei taste se pornește ciclul de măsurare și înregistrare date în fișier, în care se citesc valorile contorului, a valoare a ceasului intern urmată de salvarea acestor date în fișierul deschis.

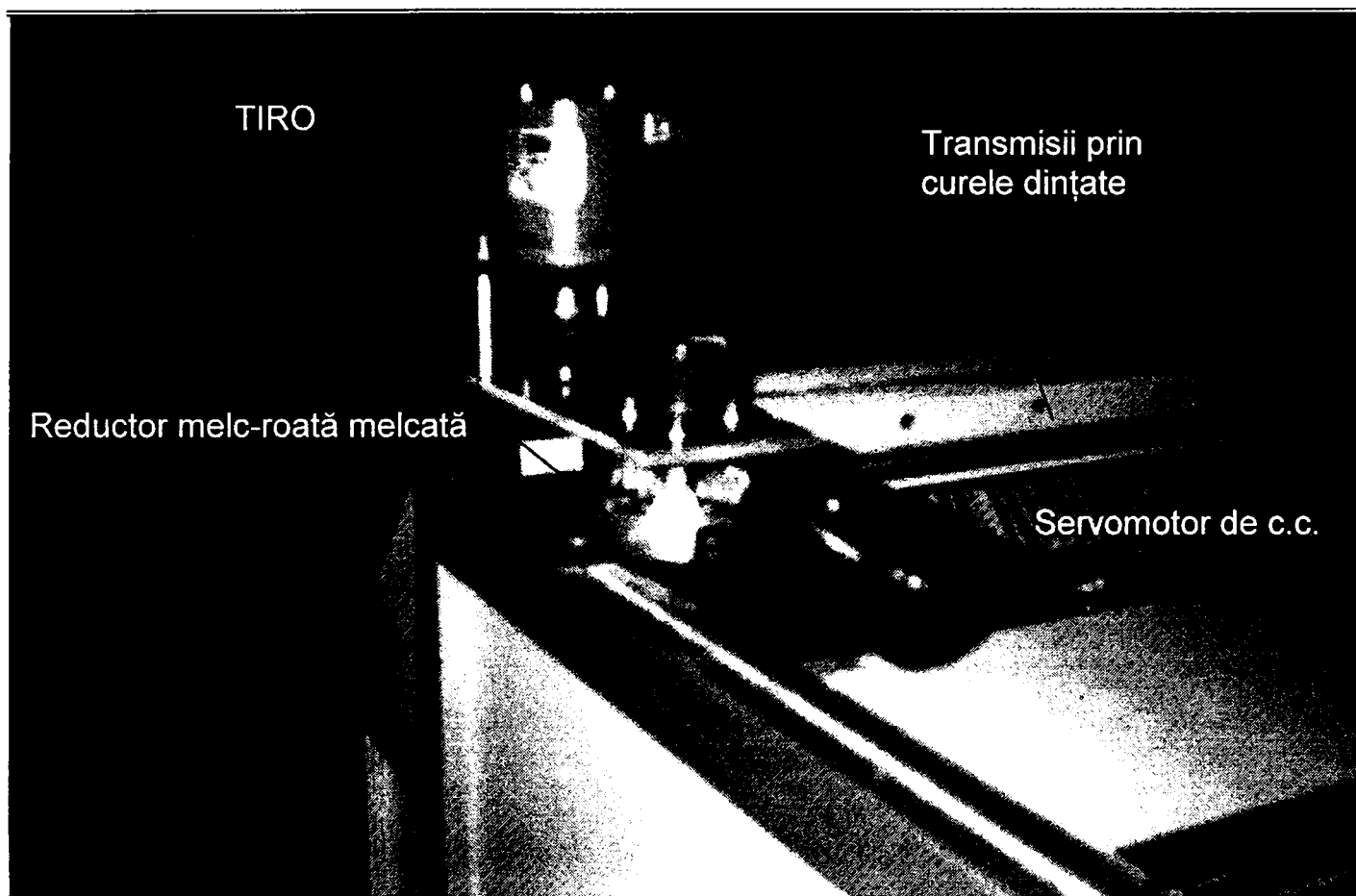


Fig. 8.35. Vedere de ansamblu a standului realizat în vederea determinării vitezei unghiulare a axului servomotorului

Buclo se va repeta până la tastarea unei taste.

La ieșirea din buclă se vor reseta funcțiile plăcii de achiziție PCI 1200.

Utilizând datele astfel obținute se poate calcula ușor viteza unghiulară a servomotorului ținând cont de raportul de transmitere a reductorului și a transmisiei cu curea dințată.

Subsistemul vizual de urmărire a reușit să urmărească deplasarea țintei atașate EF până la viteza de 0,25 [m/s]. Această viteză a fost denumită de autor viteză critică.

Prin urmare, pentru o corectă desfășurare a celui de-al doilea experiment se necesită ca viteza țintei să nu depășească această valoare critică. De aceea este necesar ca să existe posibilitatea limitării vitezei de deplasare a țintei (prin „tăierea” vârfului graficului legii de variație a vitezei în funcție de timp) astfel încât variația vitezei să se realizeze după o lege de mișcare apropiată de o formă trapezoidală.

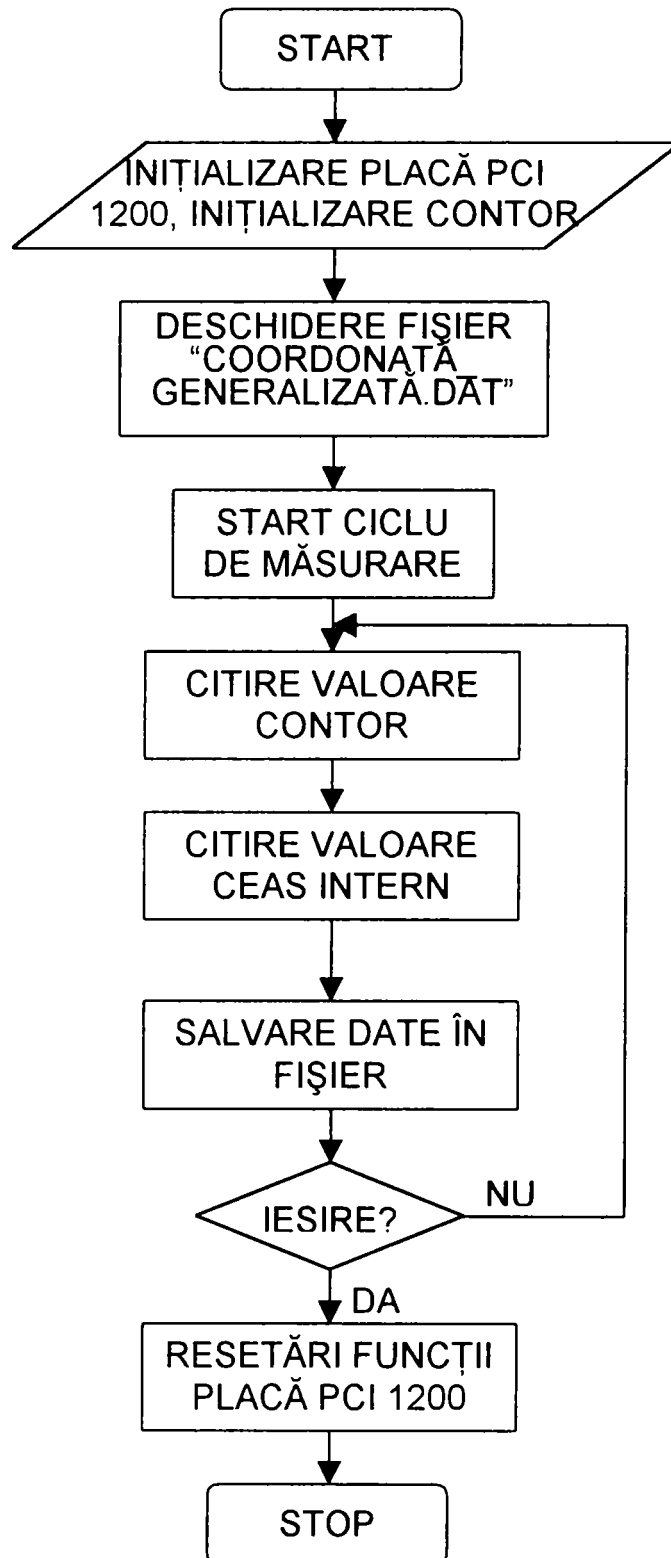


Fig.8.36. Ordinograma programului de citire a variației unghiului de rotație a axului traductorului în timp

Imaginea de pe monitorul calculatorului în timpul desfășurării procesului de urmărire a țintei se prezintă în figura 8.37.

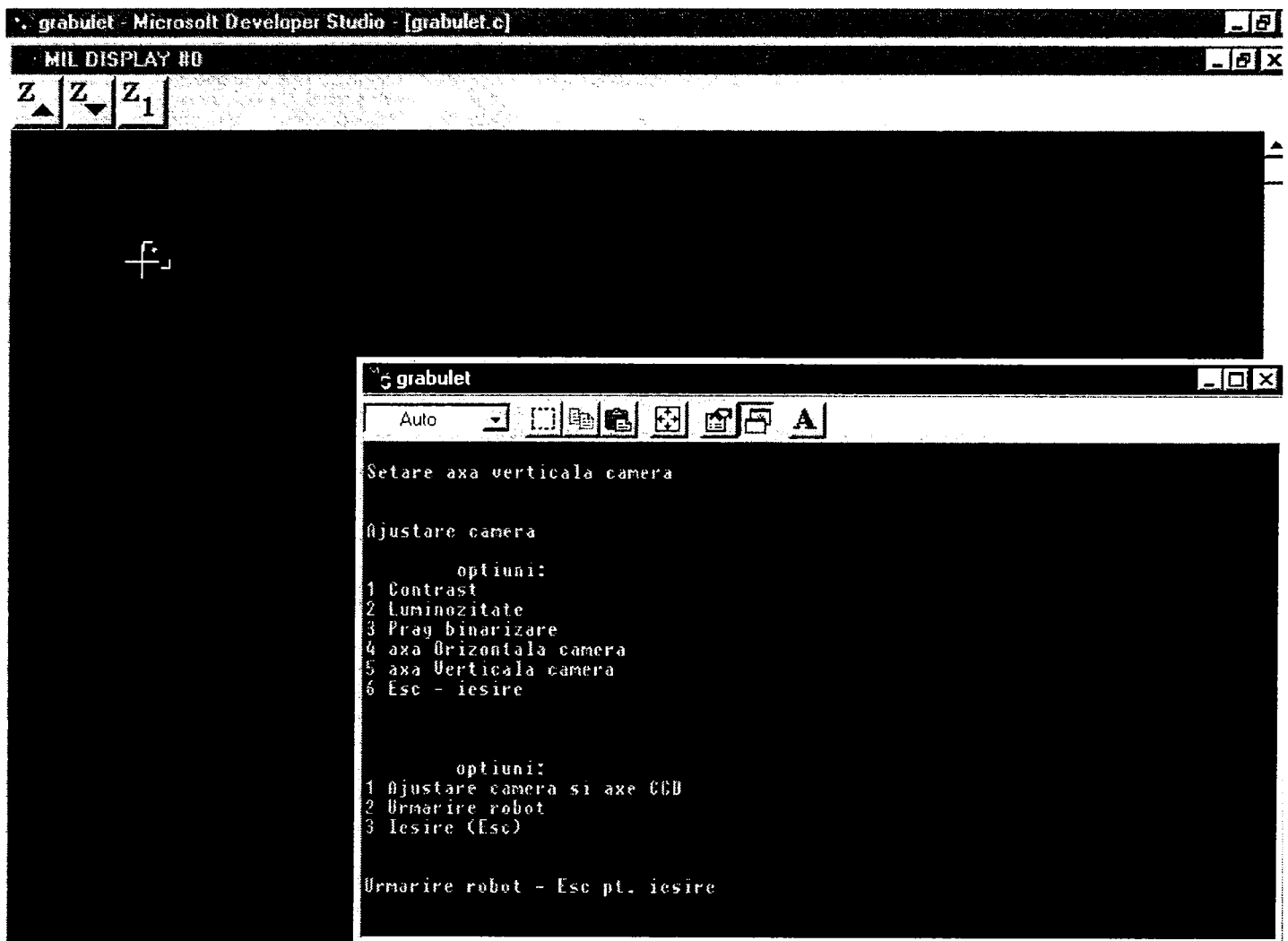


Fig.8.37. Imagine din timpul desfășurării procesului de urmărire a țintei

Secvențele ciclului de lucru al robotului pentru care s-a realizat a doua parte a experimentelor sunt prezentate în figura 8.38.

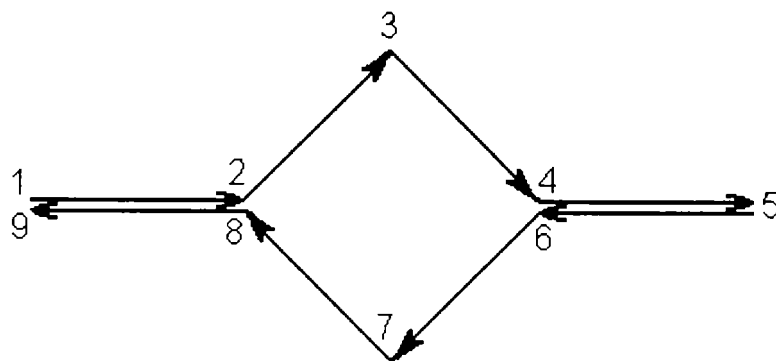


Fig. 8.38. Secvențele ciclului de lucru al robotului

Coordonatele pozițiilor 1 – 9 de atins de către punctul caracteristic în cadrul secvențelor mișcării robotului sunt date în tabelul 8.4.

Tabelul 8.4. Coordonatele pozițiilor 1 – 9 care determină ciclul de lucru al robotului

Puncte	Coordonate poziții
1	(150,450,400)
2	(150,650,400)
3	(150,800,550)
4	(150,950,400)
5	(150,1150,400)
6	(150,950,400)
7	(150,800,250)
8	(150,650,400)
9	(150,450,400)

Semnalul de comandă pentru variația în timp a vitezei de deplasare a țintei se generează prin variația corespunzătoare a factorului de umplere.

Pentru determinarea timpilor de corecție pentru fiecare secvență de program s-a adus o modificare programului de conducere globală, astfel încât la fiecare început de secvență de program (achiziție imagine nouă, căutare centru de greutate a țintei în planul imagine, comandă motor pas cu pas – secvențe prezentate în ordinograma din figura 8.26) se înregistrează timpul generat de ceasul intern al calculatorului.

În momentul în care se pornește programul se setează valoarea curentă a variabilei timp egală cu valoarea curentă a ceasului intern.

Dacă se notează cu:

- t_{i1} valoarea timpului de început al mișcării țintei la parcurgerea buclei „i” de reacție;
- t_{i2} valoarea timpului de început al achiziției imaginii la parcurgerea buclei „i” de reacție;

- t_{i3} valoarea timpului de început al calculelor necesare la parcurgerea buclei „i” de reacție;

- t_{i4} valoarea timpului de început al comenzii motorului pas cu pas la parcurgerea buclei „i” de reacție;

diferențele dintre valorile timpilor de mai sus sunt valorile timpilor aferenți fiecărei secvență din program.

Atunci:

- $\Delta t_{mcc} = t_{i2} - t_{i1}$ reprezintă timpul necesar generării comenzii pentru motorul de curent continuu, la parcurgerea buclei „i” de reacție;

- $\Delta t_{aci} = t_{i3} - t_{i2}$ reprezintă timpul necesar achiziției imaginii, la parcurgerea buclei „i” de reacție;

- $\Delta t_{mcc} = t_{i4} - t_{i3}$ reprezintă timpul necesar calculelor trăsăturilor imaginii, la parcurgerea buclei „i” de reacție;

- $\Delta t_{mcc} = t_{i+11} - t_{i4}$ reprezintă timpul necesar generării comenzii pentru motorul pas cu pas, la parcurgerea buclei „i” de reacție;

O imagine de pe monitorul calculatorului în timpul procesului de conducere globală se prezintă în figura 8.39.

Sunt afișate pe ecran următoarele valori: numărul de pași realizați ($p_{uv}=10$) de motorul pas cu pas care generează mișcarea de rotație yaw pentru subsistemul vizual de urmărire, respectiv numărul de pași realizați ($p_{uo} = 0$) de motorul pas cu pas, care generează mișcarea de rotație pitch pentru subsistemul vizual de urmărire; coordonatele țintei în sistemul de referință legat de baza robotului; coordonatele centrului de greutate ale imaginii țintei în planul imagine; unghiurile yaw și pitch la acel moment pentru subsistemul vizual de urmărire.

Prin programarea traiectoriei prezentate în figura 8.38 și rulând programul de conducere globală a robotului se determină diferențele Δt_k necesare executării secvențelor corespunzătoare s-au obținut valorile prezentate în anexa A7.

Valorile medii ale timpilor obținuți pe fiecare secvență în parte sunt date în tabelul 8.5.

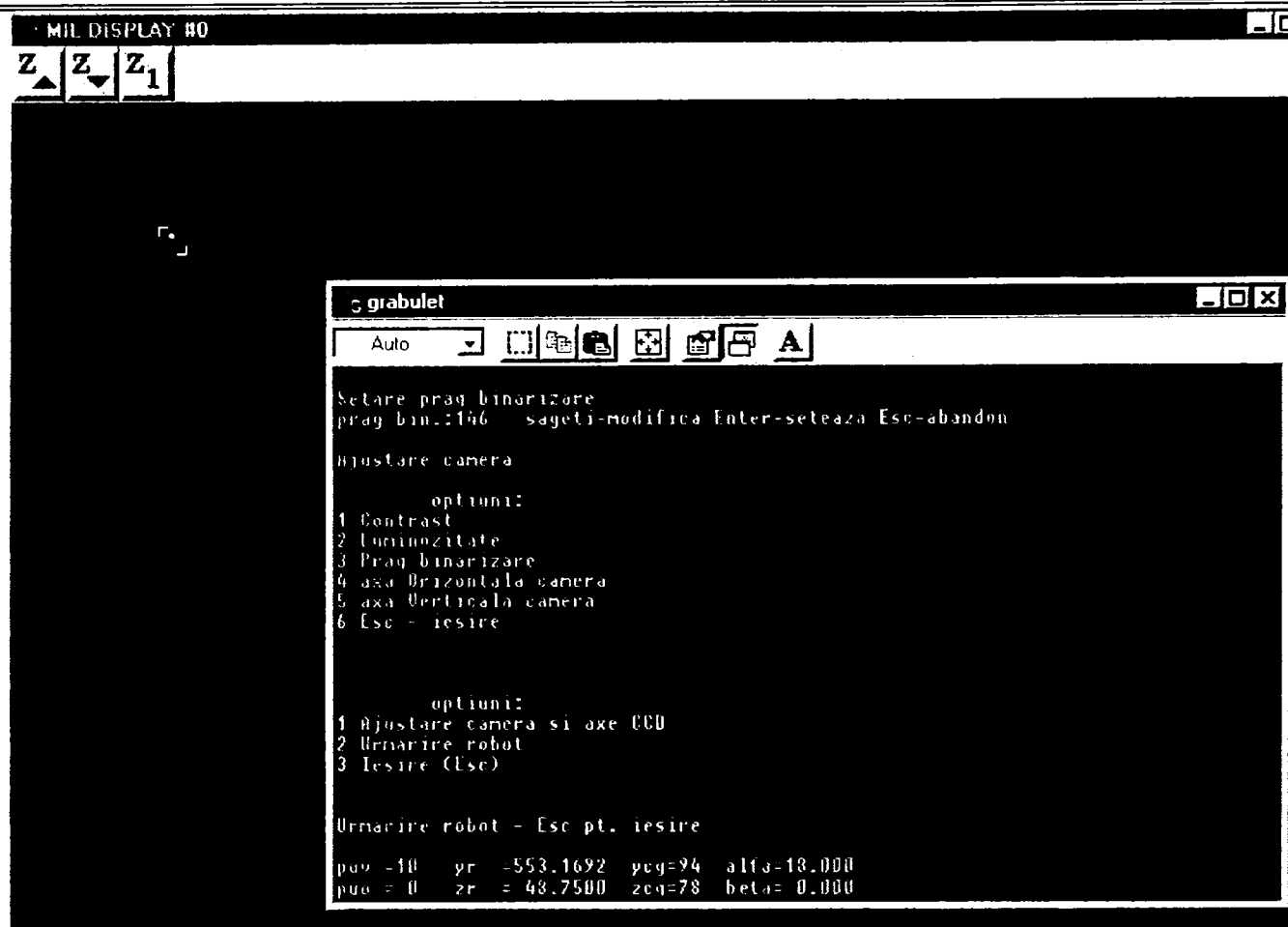


Fig.8.39. Imagine din timpul procesului de conducere globală a robotului

Tabelul 8.5. Valorile medii ale timpilor obținuți pe fiecare secvență [s]

Secvența	Generare semnal mcc	Achiziție imagine		Calculul c.g. al țintei	Generare semnal mpp
		Cu afișare	Fară afișare		
1 – 2	0,034	0,105	0,045	0,038	0,045
2 – 3	0,03	0,110	0,046	0,039	0,045
3 – 4	0,03	0,108	0,046	0,039	0,045
4 – 5	0,034	0,105	0,045	0,038	0,045
5 – 6	0,034	0,103	0,045	0,038	0,045
6 – 7	0,03	0,108	0,046	0,039	0,045
7 – 8	0,03	0,107	0,046	0,039	0,045
8 – 9	0,034	0,104	0,045	0,038	0,045

Pentru a mări viteza de execuție a programului, în etapa de experimentare a sistemului de conducere globală a robotului, s-a renunțat la afișarea imaginii țintei pe

monitorul calculatorului, obținându-se în acest mod o creștere a vitezei de închidere a buclei cu până la 36,4%.

Prin urmare, timpul mediu pentru închiderea buclei de reacție este de 0,16 [s]. Se face observația că sistemul de calcul utilizat a fost un PC Pentium II la 200 MHz, cu memorie de 16Mb RAM și memoria plăcii video de 1Mb.

Performanțele sistemului robot pot fi îmbunătățite cu până la 20 – 25 % prin utilizarea unui sistem de calcul mai performant.

Înregistrarea video a experimentului realizat s-a transferat în fișierul „experiment.wvp”, care se găsește pe CD-ul atașat tezei.

8.6.Concluzii

Pentru confirmarea experimentală a teoriilor stabilite autorul a conceput și realizat un sistem robot care dispune de o conducere globală, utilizând în acest sens subsistem vizual de urmărire prevăzut cu un senzor vizual. Pentru sistemul robot conceput s-au particularizat toate rezultatele obținute pentru cazul general.

Prin realizarea sistemului robot compus din robot și respectiv subsistemul vizual de urmărire, conform cu cele prezentate în cadrul capitolului, s-a atins unul din obiectivele fundamentale ale tezei.

Utilizând sistemul robot realizat și programul de conducere globală în timp real a sistemului robot utilizând în acest sens informații furnizate subsistemul vizual de urmărire, s-au verificat și confirmat atât corectitudinea aspectelor teoretice elaborate și dezvoltate în cadrul capitolului 5, cât și a rezultatelor obținute în cadrul simulării sistemului robot în cadrul capitolului 7.

Dacă se compară rezultatele obținute pentru fiecare din timpii aferenți secvențelor din programul de conducere globală a robotului se observă o creștere a timpilor în cazul utilizării sistemului robot realizat, pe care s-a implementat programul de conducere globală în timp real a sistemului robot, față de timpii obținuți prin simulare pentru aceleași secvențe din program.

În tabelul 8.6. se această comparare de rezultate.

Tabelul 8.6. Compararea timpilor obținuți din simulare cu cei realizați efectiv [s]

Proces simulat	Timp obținut din simulare	Timp realizat efectiv
Generare semnal mcc	0,01	0,03
Achiziție imagine	0,03	0,045
Calcul din program	0,01	0,038
Generare semnal mpp	0,01	0,045

Din compararea rezultatelor se trage concluzia că timpii realizați efectiv sunt de 1,5 – 4,5 ori mai mari decât cei obținuți prin simulare. Acest fapt se datorează următoarelor considerente: viteza rulării programului de conducere globală depinde și de performanțele sistemului de calcul utilizat, pe când simularea este independentă de acest considerent. Achiziția și generările de semnale sunt influențate de viteza cu care plăcile de achiziție „schimbă” informațiile cu sistemului de calcul utilizat. O scădere a timpilor efectiv realizați se poate obține prin utilizarea unor sisteme de calcul mai performante, dar valorile timpilor în orice caz nu pot coborî sub valorile obținute prin simularea funcționării sistemului robot.

Experimentul realizat a confirmat și corectitudinea simulărilor prin care se demonstrează posibilitatea realizării conducerii globale în timp real pentru sistemul robot realizat.

CAPITOLUL 9 Contribuții la realizarea unui sistem robot la care comanda de corecție a situării efectorului final se realizează prin utilizarea unui senzor vizual fix

9.1. Noțiuni introductive

În vederea validării aspectelor teoretice privind concepția unor sistem de conducere senzorială a roboților utilizând informații furnizate de senzori vizuali, autorul a conceput și realizat un sistem la care comanda de corecție a situării efectorului final se realizează prin utilizarea unui senzor vizual fix. S-a utilizat în acest scop un robot Mitshubishi RV-M1 din componența sistemului CIM al Universității din Oradea.

Sistemul a fost conceput în vederea analizei și recunoașterii unor obiecte de lucru – utilizând în acest sens un senzor vizual matriceal de tip CCD fix. Senzorul vizual are rolul de a determina la anumite momente situarea reală a obiectului de lucru. Sistemul de conducere senzorială al robotului, folosind informația prelevată de la senzorul vizual fix cu privire la situarea reală a OL și informația cu privire la situarea la acel moment a efectorului final va determina matricea erorii de situare, după care, printr-o analiză cinematico-pozițională inversă, se vor stabili comenzile de corecție necesare pentru pozițiile relative ale elementelor cuplelor cinematice conducătoare ale robotului, astfel încât robotul să realizeze corecta prehensare a OL.

După prehensarea OL recunoscut și clasificat în grupa corespunzătoare, robotul va depune OL în containerul corespunzător grupei din care face parte acesta.

9.2. Sistemul experimental realizat

9.1.1. Descrierea sistemului experimental realizat

În figura 9.1 se prezintă structura sistemului experimental realizat.

El se compune din:

- ☞ un senzor vizual matriceal fix de tip CCD;
- ☞ o interfață între camera CCD și unitatea de procesare a imaginii;
- ☞ un calculator care primește informația de la CCD și care are în componența sa placa de achiziție de tip Matrox, realizând binarizarea imaginii și filtrarea ei, determină

situarea obiectelor și are rolul de a conduce robotul industrial; totodată, el îndeplinește și funcția de decizie în privința clasificării obiectelor;

- ☞ unitatea de comandă a robotului, care realizează comanda efectivă a mișcării relative a elementelor cuplurilor cinematice conducătoare;
- ☞ robotul industrial RV-M1 cu cinci grade de libertate de tip RRRRR;
- ☞ buncărele în care vor fi depozitate obiectele de diferite clase recunoscute.

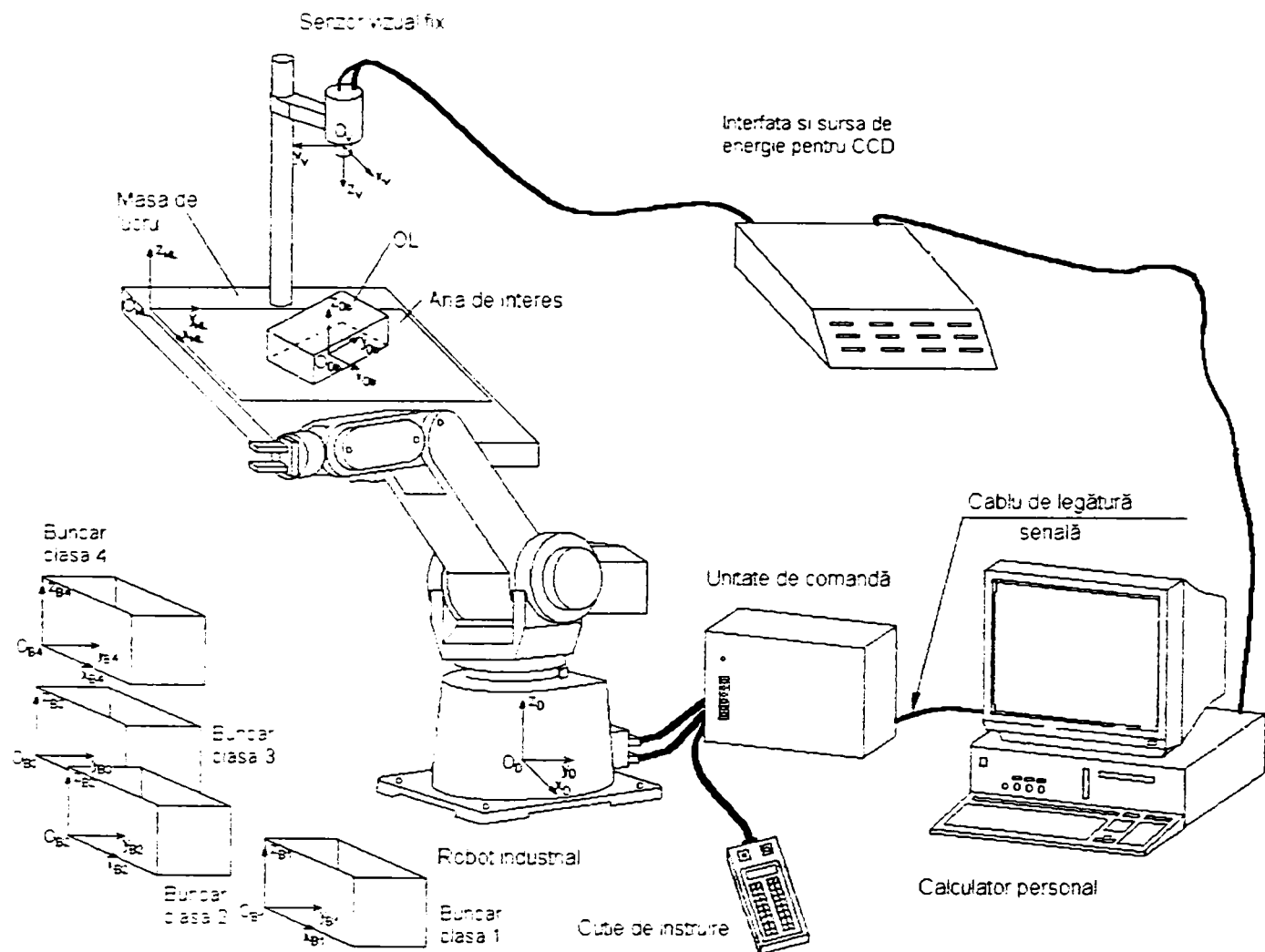


Fig.9.1. Reprezentarea sistemului experimental realizat

OL care trebuie sortate sunt așezate aleator pe masa de lucru situată sub senzorul vizual de tip CCD – scenă prezentată în fotografia din figura 9.3. Masa este de culoare albă, dintr-un material semitransparent, pentru a putea fi iluminată și de jos în sus, atunci când este cazul. Masa este prevăzută și cu posibilitatea iluminării circulare.

Pentru fiecare clasă de obiecte există câte un buncăr în care OL recunoscute de către sistem, vor fi depuse în funcție de clasa de care aparțin.

9.1.2. Particularizarea modului de calcul al situației reale a EF în sistemul de referință universal

Componentelor robot, senzor vizual fix, masă de lucru, OL li se atașează câte un sistem de referință conform celor prezentate în paragraful 5.4. (figurile 5.14, 5.18). Fiecărui buncăr i se atașează de asemenea câte un sistem de referință $O_{B_i}X_{B_i}Y_{B_i}Z_{B_i}$ $i=1,4$. Sistemele de referință sunt prezentate în figura 9.1.

Situarea OL în sistemul de referință $O_vx_vy_vz_v$ atașat de senzorul vizual fix este dată de relațiile (5.44), (5.45) și este:

$${}^{O_v}T_{O_{ob}} = \begin{pmatrix} c\alpha & -s\alpha & 0 & x_{C_{ob}} - x_v \\ -s\alpha & -c\alpha & 0 & -y_{C_{ob}} + y_v \\ 0 & 0 & -1 & D \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (9.1)$$

Situarea reală a efectorului final - în raport cu sistemul de referință $O_vx_vy_vz_v$ legat de senzorul vizual este dată de relația (5.56):

$${}^{O_v}T_{O_{re}} = \begin{pmatrix} c\alpha & -s\alpha & 0 & x_{C_{ob}} - x_v \\ -s\alpha & -c\alpha & 0 & -y_{C_{ob}} + y_v \\ 0 & 0 & -1 & D - d \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (9.2)$$

unde:

- (x_v, y_v, D) sunt coordonatele originii sistemului atașat senzorului vizual în sistemul de referință $O_{ML}X_{ML}Y_{ML}Z_{ML}$ legat de masa se lucru;

- $(x_{C_{ob}}, y_{C_{ob}}, 0)$ sunt coordonatele centrului de greutate al OL calculate cu relațiile (5.46), (5.47);

- α este unghiul pe care îl face direcția axei principale de inerție a OL cu axa $O_{ML}X_{ML}$.

Situarea senzorului vizual fix în sistemul de referință universal este dată de relația (5.66) care particularizată pentru cazul prezentat în figura 9.1 devine:

$${}^{O_o}T_{O_v} = Trans_x(-l) \cdot Trans_z(D) \cdot Rot_z(\pi) \quad (9.3)$$

sau

$${}^{O_o}T_{O_v} = \begin{pmatrix} 1 & 0 & 0 & -l \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & D \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (9.4)$$

unde l este distanța pe O_0x_0 dintre sistemul de referință $O_0x_0y_0z_0$ și sistemul de referință $O_vx_vy_vz_v$.

Situarea reală a EF în raport cu sistemul de referință universal (legat de baza robotului) este dată de:

$${}^{O_0}T_{O_{Re}} = {}^{O_0}T_{O_1} \cdot {}^{O_1}T_{O_{Re}} \quad (9.5)$$

de aici rezultă:

$${}^{O_0}T_{O_{Re}} = \begin{pmatrix} c\alpha & -s\alpha & 0 & x_{C_{Ob}} - x_v - l \\ s\alpha & c\alpha & 0 & y_{C_{Ob}} - y_v \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (9.6)$$

Din relația (9.6) se observă că, pentru a prehensa corect OL, robotul trebuie să prezinte cel puțin patru grade de mobilitate (din punct de vedere al gradelor de libertate pe care le prezintă OL robotul are nevoie doar de trei grade de mobilitate – translații după axele O_0x_0 , O_0y_0 , respectiv rotație după axa O_0z_0 , la care să se adauge mișcarea de apropiere/îndepărtare pe verticală de OL).

9.1.3. Elaborarea programului de conducere a robotului utilizând informațiile furnizate de senzorul vizual fix

Ordinograma programului de analiză și recunoaștere a obiectelor, de clasificare a acestora pe grupe, de comandă a robotului RV-M1 pentru prehensarea obiectelor din câmpul vizual al senzorului CCD și depunerea lor în containerul corespunzător grupei din care face parte se prezintă în figura 9.2.

La momentul inițial, robotul se găsește în situarea ideală - de start (situarea „home”), de unde începe fiecare ciclu de manipulare al OL sortate. În situarea ideală robotul nu intră în câmpul vizual al camerei CCD și prin urmare procesul de achiziție al imaginii OL nu este influențat.

În momentul în care operatorul uman sau un senzor extern emit un semnal de comandă, senzorul vizual CCD trece la achiziția unei imagini pe care o salvează în memoria de lucru a calculatorului (memoria RAM) sub forma unei hărți de biți. Pe monitorul calculatorului se vede în permanență tot ceea ce este în câmpul vizual al camerei CCD, dar imaginea nu este salvată în memoria calculatorului până când nu se începe procesul de analiză a imaginii. După achiziționarea unui cadru și salvarea acestuia în memorie nu se mai achiziționează noi imagini, până la începerea unui nou proces de achiziție de imagini.

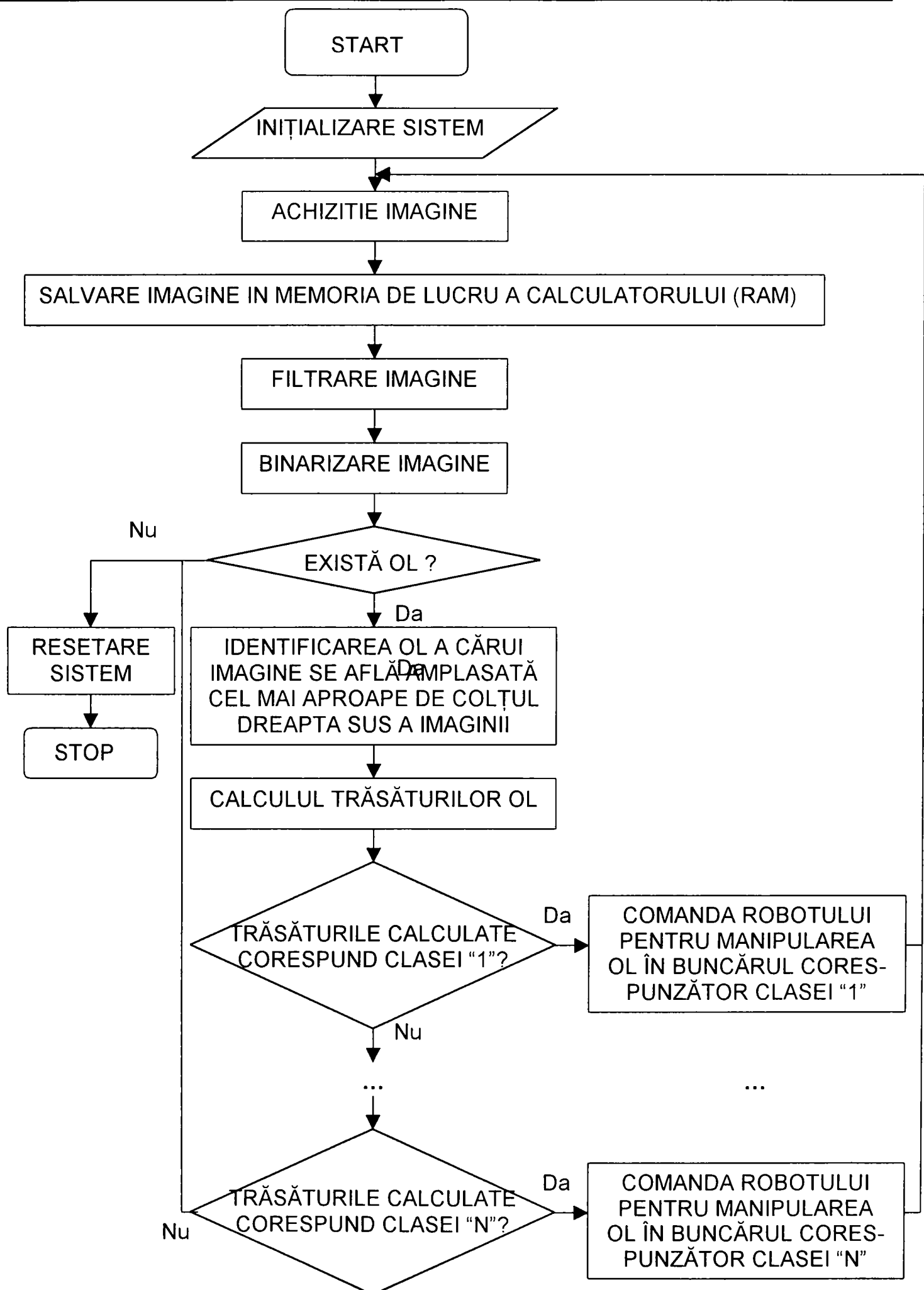


Fig. 9.2 Ordinograma programului de achiziție, recunoaștere și clasificare a OL

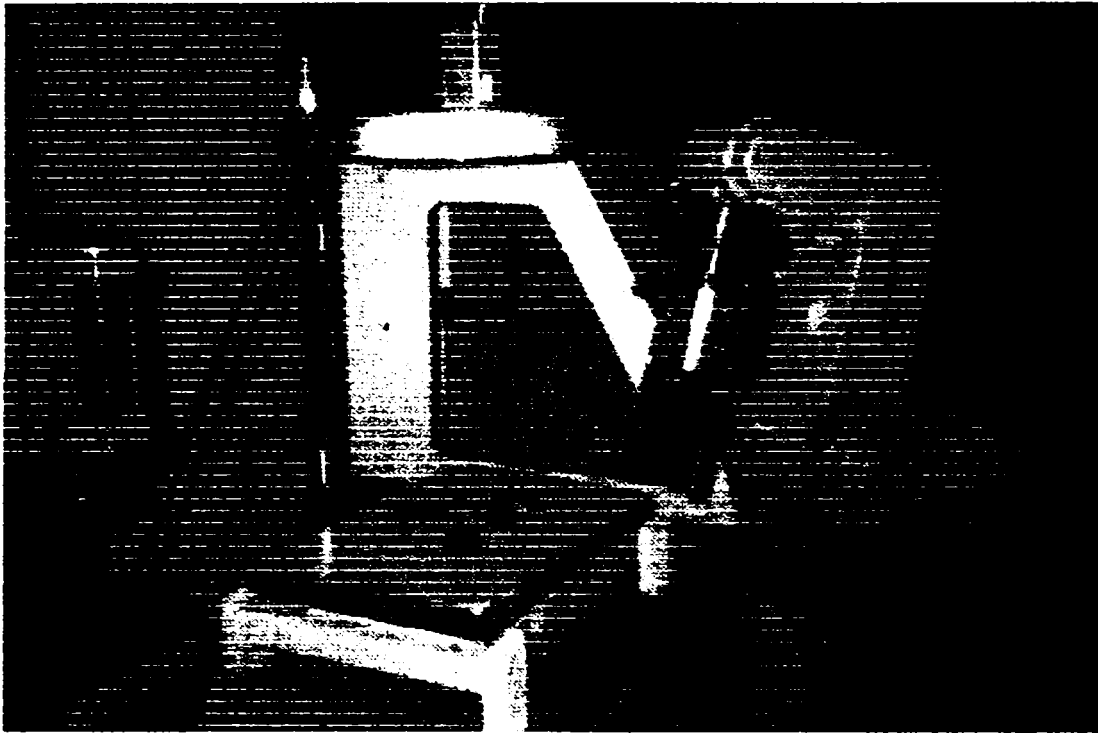


Fig.9.3. Vedere de ansamblu a sistemului experimental realizat

Formatul imaginii de achiziționare în memorie este unul matriceal de dimensiune 640x480, în care indicele rândului și al coloanei reprezintă și coordonatele pixelului de pe ecran. Pixelii sunt memorați în memoria calculatorului pe un interval de un octet, ceea ce înseamnă că pot fi memorate 256 de nivele distincte de gri. Cu această codificare, pixelul a cărui intensitate are valoarea 256 va fi de culoare albă, iar pixelul a cărui intensitate are valoarea 0 va fi de culoarea neagră. Valorile intermediare ale intensității pixelului reprezintă diverse tonuri de gri.

În figura 9.4 se prezintă o imagine achiziționată de către senzorul vizual, care urmează a fi prelucrată.



Fig.9.4. Imagine achiziționată de senzorul vizual

După achiziționarea imaginii și salvarea acesteia în memoria de lucru a calculatorului, se trece la analiza acesteia. Toate operațiile ulterioare se execută asupra imaginii stocate în memoria calculatorului, dar pentru realizarea unui control eficient al derulării programului toate schimbările survenite asupra imaginii se afișează și pe monitor.

Prima etapă în analiza imaginii o reprezintă aplicarea asupra acesteia a unei succesiuni de filtre. Filtrele aplicate pot fi de eliminare a zgomotului, de îmbunătățire a contrastului, de detecție a anumitor culori etc. Succesiunea filtrelor este determinată de către operatorul uman și în general se poate spune că nu există o regulă strictă de aplicare a acestora. Pentru o scenă dată și pentru o aplicație anume trebuie efectuate un număr mare de încercări, luându-se apoi în considerare varianta optimă din punct de vedere al rezultatelor obținute.

În cazul aplicației prezentate, după preprocesarea imaginii prin aplicarea unui filtru de eliminare a zgomotului, urmează etapa de binarizare a imaginii. În cadrul procesului de binarizare se compară valoarea intensității fiecărui pixel cu o intensitate prag, iar dacă valoarea intensității pixelului este mai mică decât valoarea prag, aceasta automat se convertește la culoarea neagră (valoarea 0), iar dacă valoarea pixelului este mai mare decât de prag aleasă, atunci este convertit automat la culoarea alb (valoarea 255).

În acest fel, după aplicarea tuturor filtrelor, în imaginea prelucrată se văd OL de pe masa de lucru, pe un fond alb strălucitor. În figura 9.5. se prezintă imaginea după binarizare.

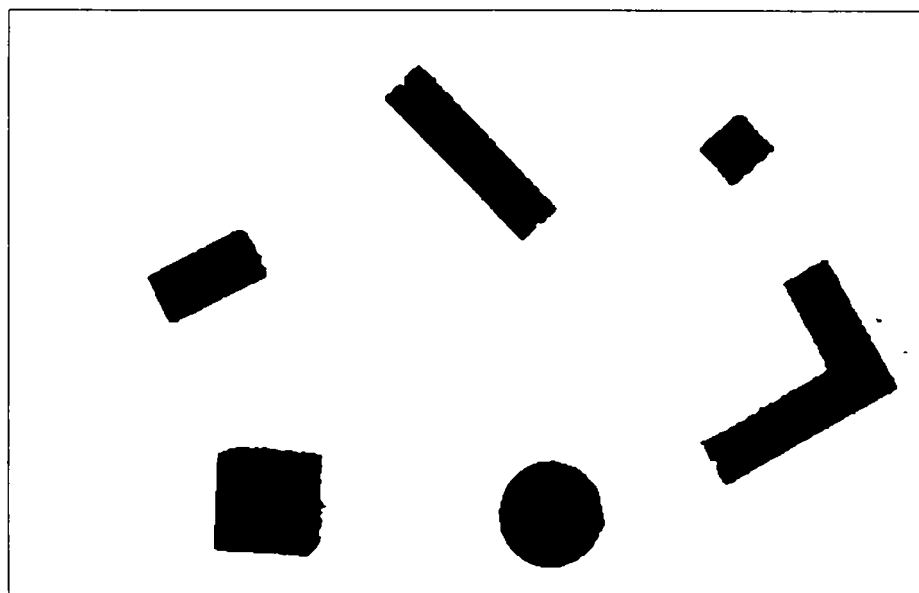


Fig.9.5. Imaginea după binarizare

În funcție de condițiile de lucru și de aplicația dată, uneori este necesar ca și după binarizare să se aplice unele filtre pentru eliminarea unor eventualelor zgomote din

imagine, neîndepărtate până la procesul de binarizare. Aceste incoerențe sunt în special datorate faptului că binarizarea este un procedeu sever de transformare a imaginii, dar care depinde în mod esențial de niște condiții de iluminat foarte stricte.

În cazul unor aplicații pretențioase, cum ar fi recunoașterea unor forme de dimensiuni foarte apropiate, este recomandat ca pragul de binarizare să nu fie impus de către programator ci să fie stabilite în timp real, pe baza unor analize statistice ale histogrammei imaginii.

În figura 9.6 se prezintă imaginea după aplicarea unor filtre de succesive de netezire (smoothing) - se observă dispariția zgomotelor din imagine.

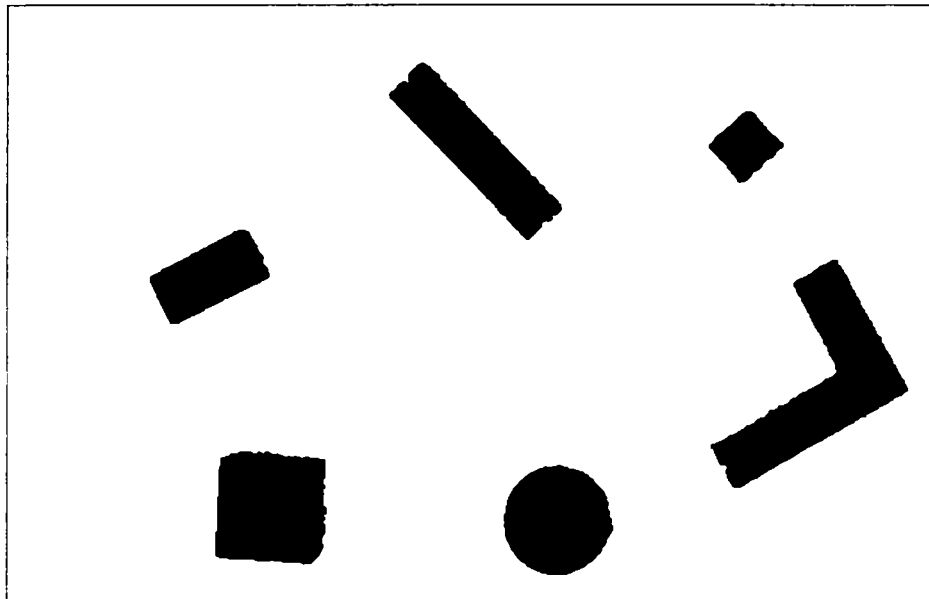


Fig.9.6. Imaginea scenei după aplicarea unor filtre de succesive de netezire (smoothing)

În cadrul exemplului prezentat s-a folosit o iluminare artificială mixtă: iluminare inelară deasupra scenei și iluminarea mesei de lucru din partea de jos a acesteia. În acest fel s-au eliminat aproape complet perturbațiile provenite din fluctuațiile iluminantului natural. În aceste condiții, după efectuarea unui mare număr de încercări s-a ajuns la concluzia că este suficient un prag de binarizare ales "a priori", nemaifiind necesară încărcarea programului cu analize statistice ale imaginii.

După terminarea programului de binarizare a imaginii și de filtrare ulterioară se trece la analiza OL aflate pe masa de lucru.

Pentru analiza și identificarea fiecărui OL se scanează imaginea, prin parcurgerea matricei din memorie pe rânduri, de la stânga la dreapta și de sus în jos. În momentul în care se găsește primul pixel de culoare neagră, înseamnă că s-a atins conturul primului obiect. Coordonatele acestui punct sunt memorate și se trece la "umplerea" primului obiect cu o culoare neutră, pentru a vizualiza și operatorul uman execuția algoritmului de umplere.

Pentru umplere s-a ales un algoritm pe bază de stivă. Acest algoritm parcurge linia curentă până la capăt, în acest timp verificând atât pe linia de deasupra cât și pe linia de dedesubt dacă nu există pixeli de culoare neagră care să aibă un vecin alb pe orizontală (capăt de linie orizontală neagră). Toți pixelii care îndeplinesc această condiție înseamnă că aparțin tot obiectului curent și în plus sunt așezați la una din extremitățile locale ale obiectului. Coordonatele acestor puncte sunt introduse într-o stivă, iar după parcurgerea liniei curente, programul ia primul element din stivă și execută aceeași operațiune până la golirea stivei. În momentul în care s-a ajuns la golirea stivei înseamnă că s-a terminat de parcurs întreaga suprafață a obiectului curent.

În timpul parcurgerii obiectului curent, dacă aplicația o cere, programul poate fi configurat în așa fel încât la apariția unui gol interior în OL, să se aplice același algoritm de umplere și în cazul golului. Acest lucru este folositor în special la aplicații de recunoaștere a OL apropiate ca formă, dar cu o conformație diferită a golurilor interioare.

Algoritmul prezintă avantajul că este foarte robust, necesitând pentru aplicarea lui un spațiu de memorie relativ restrâns în comparație cu alți algoritmi, cum ar fi de exemplu "flood fill" - un algoritm de umplere recursiv, foarte rapid dar care prezintă dezavantajul că în cazul imaginilor de o configurație complexă și de o suprafață mare necesită o mare cantitate de memorie, neputând fi rulat în absolut orice condiții pe calculatoare cu memorie relativ puțină.

Principalul dezavantaj al algoritmului pe bază de stivă aplicat, în comparație cu alți algoritmi consacrați, este viteza mai redusă de procesare a imaginii. După efectuarea unui număr suficient de mare teste și de încercări s-a ajuns la concluzia că algoritmul este suficient de rapid pentru condițiile de lucru date, nefiind necesară alegerea unui algoritm mai complex sau rescrierea acestuia în limbaj de asamblare, fapt care ar fi crescut sesizabil complexitatea acestuia, în favoarea unei creșteri importante de viteză.

În timpul umplerii obiectului curent se calculează o serie de trăsături ale imaginii cum ar fi: aria, utilizând relația (5.46), perimetrul, utilizând relația (5.53), momentele de inerție axiale și centrifugale cu ajutorul relațiilor (5.48) și respectiv (5.49), coordonatele centrului de greutate cu relațiile (5.50), orientarea α a direcției axei principale de inerție a OL cu relația (5.52), etc.

Trăsăturile caracteristice și trăsăturile de situare calculate în vederea recunoașterii și clasificării OL dar și pentru determinarea situării acestuia sunt prezentate în tabelul 9.1.

Tabelul 9.1. Trăsăturile caracteristice și trăsăturile de situare calculate

Nr. crt.	Trăsături	Formula de calcul	Valori pentru obiect nr. 3
1.	Arie	$m_{00} = \sum_{i=1}^m \sum_{j=1}^n I(x_i, y_j)$	1147 [pixeli]
2.	Perimetru	$P = \sum_{i=1}^8 l_i (\text{mod } 2) \cdot \sqrt{2} + (l_i (\text{mod } 2) + 1)$	240 [pixeli]
3.	Factor de formă	$\frac{P^2}{m_{00}}$	50,2
4.	Dimensiunile dreptunghiului minim circumscris imaginii	$L_c = x_{\max} - x_{\min}$ $l_c = y_{\max} - y_{\min}$	$L_c = 72$ $l_c = 83$
5.	Momentele de inerție axiale	$m_{02} = \sum_{i=1}^m \sum_{j=1}^n y_i^2 \cdot I(x_i, y_j)$ $m_{20} = \sum_{i=1}^m \sum_{j=1}^n x_i^2 \cdot I(x_i, y_j)$	$m_{02} = 622250$ $m_{20} = 338112$
6.	Momentul de inerție centrifugal	$m_{11} = \sum_{i=1}^m \sum_{j=1}^n x_i \cdot y_i \cdot I(x_i, y_j)$	$m_{11} = -250352$
7.	Factorul de alungire	$\frac{\sum_{i=1}^m \sum_{j=1}^n y_i^2 \cdot I(x_i, y_j)}{\sum_{i=1}^m \sum_{j=1}^n x_i^2 \cdot I(x_i, y_j)}$	1,84
8.	Coordonatele centrului de greutate	${}^i x_{C_{ob}} = \frac{m_{10}}{m_{00}} = \frac{\sum_{i=1}^m \sum_{j=1}^n x_i \cdot I(x_i, y_j)}{\sum_{i=1}^m \sum_{j=1}^n I(x_i, y_j)}$ ${}^i y_{C_{ob}} = \frac{m_{01}}{m_{00}} = \frac{\sum_{i=1}^m \sum_{j=1}^n y_i \cdot I(x_i, y_j)}{\sum_{i=1}^m \sum_{j=1}^n I(x_i, y_j)}$	${}^i x_{C_{ob}} = 386$ ${}^i y_{C_{ob}} = 95$
9.	Direcția axei principale de inerție	$\alpha = \frac{1}{2} \cdot \arctg \left(\frac{2 \cdot \sum_{i=1}^m \sum_{j=1}^n x_i \cdot y_i \cdot I(x_i, y_j)}{\sum_{i=1}^m \sum_{j=1}^n y_i^2 \cdot I(x_i, y_j) - \sum_{i=1}^m \sum_{j=1}^n x_i^2 \cdot I(x_i, y_j)} \right)$	$\alpha = 59,4$

unde $I(x_i, y_i) = \begin{cases} 0, & \text{daca punctul de coordonate } (x_i, y_i) \in \text{clasei mediu} \\ 1, & \text{daca punctul de coordonate } (x_i, y_i) \in \text{clasei obiect} \end{cases}$

În figura 9.7 se prezintă o secvență a ecranului care afișează o parte din trăsăturile calculate pentru un OL. În figura 9.8 se prezintă imaginea scenei după ce primul obiect a fost identificat.

TRASATURI		
nume	valoare curenta	valoare de referinta
- Arie totală	13986	13975
- Perimetru	587	585
- x centru	323	-
- y centru	225	-
- xmin	239	-
- xmax	406	-
- ymin	157	-
- ymax	293	-

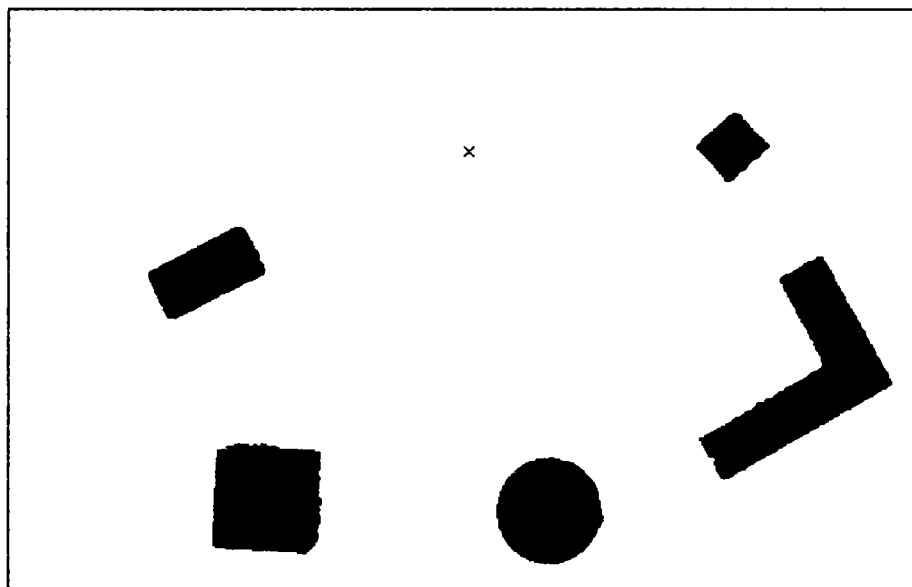


Fig.9.8. Imaginea scenei după ce primul OL a fost identificat

Fig.9.7. Trăsături calculate pentru un OL

După terminarea procesului de umplere și calcularea ariei OL, axelor și a momentelor principale de inerție, se calculează perimetrul OL și în cazul în care aplicația o cere, se execută exact aceeași procedură de identificare și de calcul pentru fiecare dintre găurile OL.

După efectuarea tuturor calculelor privitoare la obiectul curent, se trece la o analiză a trăsăturilor acestuia, pentru a-l clasifica într-o clasă cunoscută de către robot.

Pentru recunoașterea clasei din care face parte OL se pot lua în considerare foarte multe trăsături, dar în condițiile date s-au considerat ca suficiente următoarele: aria obiectului, numărul de găuri ale acestuia, aria găurilor, perimetrul obiectului, perimetrul găurilor, factorul de formă al OL, momentele de inerție principale.

Pentru clasificarea obiectului se trece succesiv la compararea trăsăturilor calculate cu valorile de referință aferente unei clase anume.

După ce primul OL a fost încadrat într-una din clasele cunoscute sau după ce a fost catalogat drept necunoscut, programul va determina matricea erorii de situare, după care printr-o analiză cinematico-pozițională inversă se vor stabili comenzile de corecție

necesare pentru pozițiile relative ale elementelor cuplelor cinematice conducătoare ale robotului, astfel încât robotul să realizeze corecta prehensare a OL.

După prehensarea OL se va comanda robotul astfel încât EF să ajungă într-o situație corespunzătoare depunerii OL în buncărul corespunzător clasei din care face parte, urmată de întoarcerea robotului în poziția "home", poziția de start, de început a oricărui ciclu de manipulare a robotului.

În figura 9.9 se prezintă imaginea scenei după ce primul OL a fost așezat de către robot în buncărul corespunzător clasei sale.

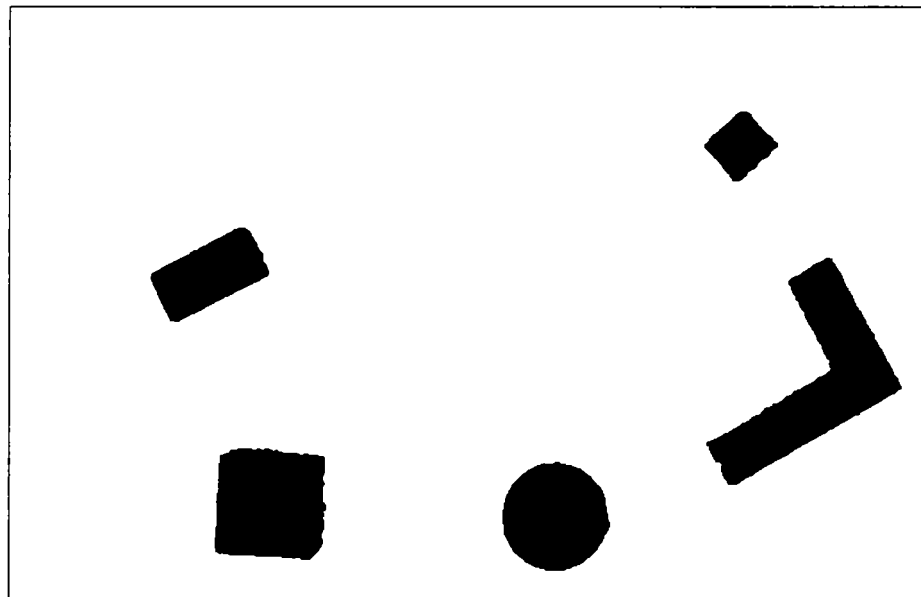


Fig.9.9. Imaginea scenei după ce primul OL a fost depozitat de robot în buncărul corespunzător clasei

În fotografia din figura 9.10 se prezintă o vedere de ansamblu a sistemului servovizual de tip „*position based look-and-move*” în momentul în care robotul urmează să prehenseze un OL.

Programul ajuns la acest punct se poate materializa prin două variante: - se poate continua cu procesul de analiză a imaginii stocate în memoria calculatorului, reluându-se toți pașii descriși mai înainte, mai puțin achiziția imaginii sau se poate face o nouă achiziție a imaginii, iar după aceea să se reia etapele prezentate anterior.

Prima variantă prezintă avantajul unei bune folosiri a timpului, prelucrarea inițială a imaginii (achiziționarea și filtrarea succesivă) chiar dacă nu durează foarte mult, în cazul unui număr mare de OL situate pe masa de lucru duce la întârzieri semnificative.

Varianta a doua prezintă avantajul că în cazul modificării situației OL de pe masa de lucru, programul ia cunoștință despre aceasta, adaptându-se în mod corespunzător (organigrama prezentată în figura 9.2 este realizată pe această variantă).

Programul realizat prezintă câteva avantaje semnificative, mai ales prin prisma utilizării potențiale a acestuia, dar prezintă și câteva limitări.

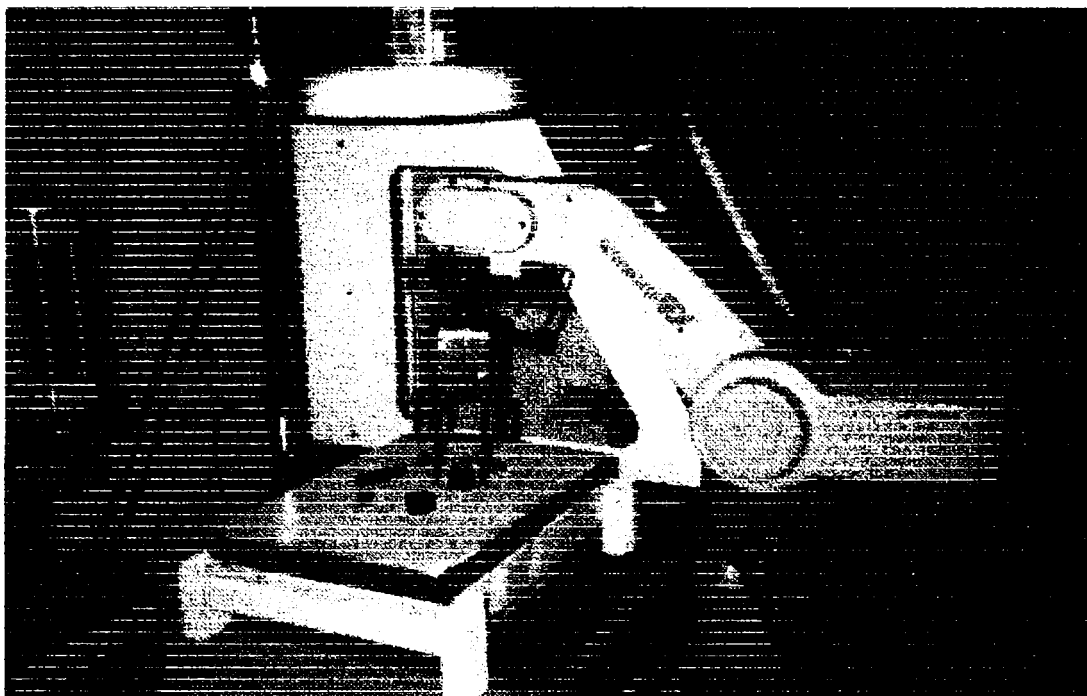


Fig.9.10. Vedere de ansamblu a sistemului experimental realizat în momentul în care robotul urmează să prehenseze un OL

Spre deosebire de alte programe, corectitudinea funcționării acestui program nu este influențată de faptul că dreptunghiul minim al unui OL intersectează dreptunghiul minim al altui OL, atât timp cât cele două OL nu se ating. În cazul în care două OL de pe masa de lucru se ating, calculatorul le interpretează ca și un singur OL.

Sistemul conceput nu are posibilitatea estimării înălțimii OL. Din această cauză, pentru o bună funcționare a acestuia toate OL vor avea înălțimea cuprinsă între două limite acceptabile în vederea prehensării.

Procesul se repetă până când toate OL sunt clasificate și depozitate în buncărul corespunzător.

Programul principal "ROBOT VISION" al algoritmului prezentat în figura 9.2 se prezintă în anexa A8.

9.3. Concluzii

Programul realizat este un program interactiv, dar există posibilitatea de a-l configura astfel încât - pentru o situație dată, după un anumit număr de încercări - să existe posibilitatea execuției automate a recunoașterii și manipulării OL.

Dintre posibilele aplicații suplimentare ale acestui program se amintesc doar câteva:

- controlul dimensional al OL;
- verificarea stării suprafețelor după operații de vopsire etc.;
- automatizarea operațiilor de sortare și ambalare;
- verificarea operațiilor de asamblare.

CAPITOLUL 10 Contribuții originale. Concluzii finale

10.1. Contribuții originale

10.1.1. Introducere

Tema abordată în cadrul tezei de doctorat contribuie la soluționarea principalelor probleme care apar la conducerea în timp real a roboților, utilizând informații cu privire la situarea efectorului final achiziționate prin senzori.

Justificarea abordării temei este oferită de „explozia” tehnologică și informațională din ultimele decenii, care conferă nebănuite posibilități de înzestrare a oricărui membru al familiei robot cu facilități sporite, astfel încât acesta să răspundă inteligent oricărei sarcini primite.

În cadrul tezei se propun câteva variante de roboți adaptivi cu funcții specifice controlului senzorial, a căror subsisteme senzoriale au fost dezvoltate de către autor.

Obiectivul fundamental al tezei de doctorat îl reprezintă validarea experimentală a posibilității conducerii globale în timp real a unui robot utilizând un sistem de urmărire un sistem senzorial de urmărire nemijlocită a poziției obiectului manipulat.

În cadrul tezei s-au urmărit următoarele obiective:

☞ fundamentarea teoretică a celor mai importante noțiuni, problematici, tehnici care sunt necesare a fi utilizate în vederea conducerii globale a roboților;

☞ concepția aspectelor matematice pentru unele sisteme de conducere senzorială a roboților pe baza informațiilor furnizate de sisteme senzoriale vizuale;

☞ modelarea geometrică 3D și simularea funcționării prin animație a unui sistem concret de conducere specifice roboților adaptivi;

☞ modelarea funcțională și simularea funcționării unui sistem de conducere senzorială a roboților pe baza informațiilor furnizate de un sistem vizual de urmărire nemijlocită a situației obiectului manipulat;

☞ realizarea practică a unui sistem de conducere senzorială a roboților pe baza informațiilor furnizate de un sistem senzorial de urmărire nemijlocită a situației obiectului manipulat;

☞ testarea sistemului obținut și compararea mărimilor de ieșire a sistemului cu mărimile de ieșire ale modelului;

☞ realizarea practică a unui sistem servo-vizual "tradițional" de tip "*position based (iterated) look-and-move*";

☞ formularea unor concluzii privind utilizarea în timp real a sistemului de conducere senzorială a roboților utilizând informații furnizate de la un subsistem vizual de urmărire.

Contribuțiile originale ale autorului rezultă din urmărirea realizării obiectivelor propuse prin teza de doctorat și pot fi sintetizate pe patru direcții principale și anume:

- în domeniul sistematizării informațiilor bibliografice;
- în plan teoretic;
- în domeniul cercetării experimentale;
- în domeniul aplicațiilor practice.

10.1.2. Contribuții în domeniul sistematizării informațiilor bibliografice

În cadrul cercetărilor bibliografice s-a realizat o sistematizare a informațiilor extrase din materialul bibliografic după cum urmează:

- prezentarea unor clasificări a tehnicilor de conducere a roboților, evidențiind sarcina conducerii robotului atât analitic, cât și ca rezultat al scopului conducerii robotului;
- elaborarea unei arhitecturi a sistemului de conducere a robotului văzută din prisma interconexiunilor care apar în cadrul hipersistemului robot;
- sintetizarea informațiilor cu privire la tehnicile de conducere a roboților cu o prezentare a celor mai noi realizări în domeniul conducerii adaptive – în special a sistemelor servovizuale;
- o inventariere a cercetătorilor și grupurilor de lucru care lucrează în domeniul conducerii servovizuale a roboților;
- sintetizarea informațiilor despre metodele/tehnicile de modelare a ansamblului – motor-transmisie mecanică-structură mecanică – a robotului;
- prezentarea unor noțiuni legate de formarea imaginii și achiziția ei, noțiuni care sunt relevante în conducerea servovizuală a roboților;
- analiza structurii și funcțiile sistemelor senzoriale vizuale.

10.1.3. Contribuții în plan teoretic

Contribuțiile în plan teoretic se materializează prin următoarele:

- generalizarea – prin exprimarea unitară – modelului dinamic al acționărilor cu servomotor de curent continuu, respectiv cu motor hidraulic comandat de o servovalvă;
- conceperea principilui de lucru privind modul de realizare a conducerii globale a unui sistem robot, care utilizează informațiile furnizate de un sistem senzorial vizual;
- conceperea modelelor geometrice ale sistemelor senzoriale vizuale;
- determinarea relațiilor matematice care permit obținerea coordonatelor punctului țintă în funcție de direcțiile de vizare ale senzorilor vizuali și de imaginile punctului țintă din planele imagine ale celor doi senzori;
- conceperea aparatului matematic care permite determinarea situării unui corp în spațiu, în cazul în care se cunosc poziții ale trei puncte necoliniare, egal distanțate ale sale;
- elaborarea metodologiei de determinare a erorilor de situare a țintei și a metodologiei de calcul al comenzilor de corecție a situațiilor relative a elementelor fiecărei c.c.c.;
- particularizarea aparatului matematic conceput pe sistemele robot concepute și realizate de autor.

10.1.4. Contribuții în domeniul cercetării experimentale

În planul cercetării experimentale contribuțiile sunt:

- modelarea geometrică 3D a fiecărei componente a sistemului robot;
- animația 3D pentru mișcările realizate de către componentele sistemului robot în timpul conducerii sale globale;
- modelarea funcțională a acționării subsistemului vizual de urmărire și simularea funcționării acestuia;
- modelarea funcțională a acționării subsistemului robot și simularea funcționării acestuia;
- modelarea funcțională a senzorului vizual și simularea funcționării acestuia;

- modelarea funcțională a sistemului de calcul și simularea funcționării acestuia;
- modelarea funcțională pentru o c.c.c. a sistemului robot și simularea funcționării acesteia;
- modelarea funcțională pentru sistemul robot realizat și simularea funcționării acestuia.

10.1.5. Contribuții în domeniul aplicațiilor practice

În acest domeniu autorul și-a adus următoarele contribuții originale:

- proiectarea și realizarea unui subsistem vizual de urmărire, necesar realizării conducerii globale a sistemului robot;
- integrarea într-un tot unitar a robotului și a subsistemului vizual de urmărire – sub toate aspectele (sistem mecanic, sistem de acționare, sistem de conducere) - în vederea confirmării aspectelor teoretice și validării rezultatelor simulărilor cu privire la conducerea globală a robotului care utilizează informații furnizate de senzori;
- realizarea unui stand care să permită determinarea vitezei unghiulare a motorului care acționează subsistemul robot;
- realizarea unui stand care să permită determinarea vitezei critice pentru ținta de urmărit;
- proiectarea și realizarea interfeței pentru sistemul de comandă a subsistemului vizual de urmărire;
- proiectarea și realizarea interfețelor pentru sistemul de comandă a robotului;
- întocmirea subprogramului de achiziție imagini din cadrul programului de conducere globală a sistemului robot;
- întocmirea subprogramului de comandă a subsistemului vizual de urmărire din cadrul programului de conducere globală a sistemului robot;
- întocmirea subprogramului de comandă a subsistemului robot din cadrul programului de conducere globală a sistemului robot;
- realizarea programului de conducere globală a sistemului robot utilizând informații furnizate subsistemului vizual de urmărire;
- proiectarea și realizarea un sistem servo-vizual "tradițional" de tip "*position based (iterated) look-and-move*", în vederea analizei, respectiv recunoașterii

- unor obiecte și de clasificare a acestora pe grupe (sistem de conducere senzorială care utilizează în vederea prelevării informației un senzor vizual fix);
- întocmirea programului de achiziție, recunoaștere, clasificare OL utilizând un sistem servo-vizual "tradițional" de tip "*position based (iterated) look-and-move*".

10.1.6. Direcții de cercetare viitoare

Lucrarea pune la dispoziția celor interesați un material original cu privire la conducerea globală – în poziție – a sistemelor robot utilizând pentru aceasta informațiile furnizate de un subsistem senzorial vizual.

Rezultatele obținute de autor prezintă importanță în vederea aplicării în viitor a tehnicilor de conducere globală a roboților. Astfel, rezultatele pot fi dezvoltate astfel încât următorii pași în tehnica de conducere globală i-ar constitui trecerea la o conducere globală în viteză, respectiv în accelerație.

Dintre domeniile în care se pot implementa cu succes conducerea globală a sistemelor robot amintim:

- industrie (servirea benzilor rulante, mișcarea ghidată vizual, asamblarea, sudura);
- agricultură (mașini de recoltat fructe, mașini de sortat produse agricole);
- construcții;
- armată (aplicații militare pentru urmărirea automată a rachetelor, interceptarea unor obiecte aflate în zbor, dezamorsarea de mine, etc.);
- alte aplicații (conducerea unor roboți subacvatici, cuplarea unui conector al rampei de lansare a navetei spațiale).

10.2. Concluzii finale

Teza de doctorat cu titlul "Utilizarea informațiilor cu privire la situarea efectorului final achiziționate prin senzori în vederea conducerii în timp real a roboților" tratează într-o variantă originală aspectele referitoare la conducerea globală a sistemelor robot.

Lucrarea este structurată pe zece capitole incluzând 141 figuri, 22 tabele și 238 relații pe parcursul a 249 pagini. Anexele la prezenta teză cuprind 49 pagini.

După sintetizarea tehnicilor de conducere a roboților cu o prezentare a celor mai noi realizări în domeniul conducerii adaptive – în special a sistemelor servovizuale, în

cadrul tezei s-a realizat o prezentare a metodelor/tehnicilor de modelare a sistemului robot și a noțiunilor legate de formarea imaginii și achiziția ei, aspecte care sunt relevante în conducerea servovizuală a roboților.

Pornind de la primul obiectiv al tezei de doctorat de a se realiza un studiu teoretic al conducerii în timp real al roboților utilizând ca buclă de reacție informațiile furnizate de senzori vizuali, autorul prin soluțiile propuse, a conceput două sisteme robot la care controlul situării efectorului final să se poată realiza fie prin intermediul unui subsistem vizual de urmărire, fie utilizând un senzor vizual fix.

Din programul de lucru al robotului se determină situarea programată a efectorului final și cu ajutorul generatorului de traiectorie coordonatele generalizate la nivelul fiecărei cuple cinematice conducătoare, iar din informațiile obținute de către subsistemul senzorial vizual, se determină situarea reală a obiectului manipulat. Comparându-se situarea programată cu cea reală determinată cu ajutorul subsistemului senzorial vizual, rezultă o eroare de situație, care prin intermediul unei analize inverse va fi transformată în erori la nivelul fiecărei cuple cinematice conducătoare. Semnalul rezultat din suprapunerea peste semnalul de intrare a corecțiilor necesare a fi făcute pentru eliminarea erorilor, se transmite sistemului de acționare în vederea punerii în mișcare a elementelor componente ale sistemului mecanic.

Avantajul unei astfel de abordări îl constituie asigurarea urmăririi obiectivului conducerii robotului prin controlul și corecția în timp real a situării efectorului final.

Autorul a modelat și simulat funcționarea sistemului robot proiectat. Din simulare s-a demonstrat posibilitatea conducerii în timp real a sistemului robot realizat.

Experimentele efectuate și rezultate obținute au confirmat atât corectitudinea aspectelor teoretice elaborate și dezvoltate în cadrul tezei de doctorat, cât și a rezultatelor obținute în cadrul simulării sistemului robot, demonstrând posibilitatea realizării conducerii în timp real al sistemului de conducere senzorială al robotului care pentru prelevarea informației privitoare la situarea efectorului final utilizează un subsistem vizual de urmărire.

Teza de doctorat prin cercetările efectuate și realizările obținute constituie o noutate prin modul de abordare și de rezolvare a problematicii, deschizând căi noi de cercetare în domeniul conducerii sistemelor robot și nu numai.

Bibliografie

- [**Ande87**] Anderson, R., Real Time Expert System to Control a Robot Ping-Pong Player. PhD thesis, University of Pennsylvania, June 1987.
- [**Asad86**] Asada, H., Slotine, J.J.E. Robot Analysis and Control, John Wiley&Sons Ltd., 1986.
- [**Arim81**] Arimoto, S., Takegaki, M. An Adaptive Method for Trajectory Control of Manipulators. IFAC 8th World Congress, Kyoto, 1981.
- [**Arms79**] Armstrong, W. Recursive Solution to the Equations of Motions of an n-link Manipulator. In Proc. 5-th World Congress on Theory of Machines and Mechanism, pp. 1343-1346, Montreal, July, 1979.
- [**Bogd89**] Bogdanov, I. Microprocesorul în comanda acționărilor electrice, Editura Facla, Timișoara, 1989.
- [**Bucș96**] Bucșan, C., Senzori pentru subansambluri cinematice ale roboților industriali, Teză de Doctorat, Universitatea Politehnica București, 1996.
- [**Cali88**] Călin, S., Popescu, Th., Jora, B., Sima, V., Conducerea adaptivă și flexibilă a proceselor industriale, Editura Tehnică, București, 1988.
- [**Coif92**] Coiffet, Ph. La Robotique - principes et applications, 3e édition revue et complétée, Hermes, Paris, 1992.
- [**Cojo85**] Cojocaru, G., Kovacs F.V., Roboții în acțiune. Sisteme flexibile și fabricația în serie. Editura Facla, Timișoara, 1985.
- [**Cork89**] Corke, P., Paul, R., Video-Rate Visual Servoing for Robots. In V. Hayward and O. Khatlib, Editors, Experimental Robotics I, volume 139 of Lecture Notes in Control and Information Sciences, pp. 429-451. Springer-Verlag, 1989.
- [**Cork94**] Corke, P.I. High-Performance Visual Closed-Loop Robot Control, PhD Thesis, University of Melbourne, 1994.
- [**Crai86**] Craig, J.J., Introduction to Robotics. Addison-Wesley Publ. New-York, 1986.
- [**Deac86**] Deacu, L., ș.a. Tehnica hidraulicii proporțională, Editura Dacia Cluj-Napoca, 1986.

- [Dolg96] Dolga, V., Traductoare și senzori, Lito Universitatea "Politehnica" Timișoara, 1996.
- [Drag97] Drăgulescu, D., Dinamica roboților, Editura Didactică și Pedagogică RA, București, 1997.
- [Drim85] Drimer, D., Alexandrescu, N., ș.a. Roboți industriali și manipolatoare, Editura Tehnică, București, 1985.
- [Dubo79] Dubovsky, S., DesForges, D.T. The Application of Modelling Identification and Control, vol.6, 1979.
- [Dudi87] Dudiță, Fl., Diaconescu, D., Curs de mecanisme. Fascicula 4. Robotomecanisme. Lito Universitatea Brașov, 1987.
- [Dumi96] Dumitriu, A., Bucșan, C., Demian, T., Sisteme senzoriale pentru roboți, Editura Medro, București, 1996.
- [Fäss90] Fässler, H., Beyer, H., Wen, J., A Robot Ping-Pong Player: Optimised Mechanics, High Performance 3D Vision and Intelligent Sensor Control. Robotersysteme, 6:161-170, 1990.
- [Fedd89] Feddema, Mitchell, O., Vision-Guided Servoing with Feature-Based Trajectory Generation. IEEE Trans. Robot. Autom., 5(5):691-700, October 1989.
- [Gherb81] Gherbanovschi, N., Prodan, M., Levay, Șt., Fizică, Editura Didactică și Pedagogică, 1981.
- [Hand96] Handra Luca, V., Măties, V., Brișan, C., Tinca, T., Roboți. Structură, cinematică și caracteristici. Editura Dacia, Cluj-Napoca, 1996.
- [Hara85] Harrell, R.C., Slaughter, D.C., Adsit, P.D., A Fruit-Tracking System for Robotic Harvesting. Machine Vision and Applications, 2:69-80, 1989.
- [Hash91a] Hashimoto, K. Model-Based Control of Manipulator with Image-Based Dynamic Visual Servo, PhD Thesis, Osaka University, 1991.
- [Hash91b] Hashimoto, K. Kimoto, T., Ebine, T., Kimura, H. Manipulator Control with Image-Based Visual Servo, IEEE International Conference Robotics and Automation, Sacramento, California, 1991.
- [Hash94] Hosoda, K., Asada, M. Versatile Visual Servoing without Knowledge of True Jacobian, Proc. of IROS, 1994.
- [Hayn91] Bukowski, R., Haynes, L., a.o. Robot Hand-Eye Coordination Rapid Prototyping Environment. In Proc. ISIR, pp. 16.15-16.28, October, 1991.
- [Hill79] Hill, R.E., A New Algorithm for Modelling Friction in Dynamic Mechanical Systems. TDA Progress Report 42-95. Pp. 51-57, July 1979.

- [**Holl80**] Hollerbach, J. A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamic Formulation Complexity. IEEE Trans. Syst. Man Cybern., SMC-10(11):730-736, November, 1980.
- [**Hoso97**] Hosoda, K., Ishida K., Asada, M. Utilizing Affined Description for Adaptive Visual Servoing, Workshop on New Trends in Image-Based Robot Servoing, sept 1997, pag. 13-18.
- [**Hsia86**] Hsia, T.C. Adaptive Control of Robot Manipulators. A Review. IEEE Trans., 1986, pp.183-189.
- [**Ione86**] Ionescu, V., Popeea, C., Conducerea structurală a sistemelor liniare, Ed. Tehnică, București, 1986.
- [**Iorg96**] Iorga, V., Jora, B., Nicolescu, C., Lopătan, I., Fătu, I. Programare numerică, Ed. Teora, București, 1996.
- [**Ispa99**] Ispas, V., Roboți pentru aplicații speciale, Editura Dacia, Cluj-Napoca, 1999.
- [**Ivăn94**] Ivănescu, M., Roboți industriali, Editura "Universitaria" Craiova, 1994.
- [**Jäge94**] Jägersand, M., Nelson, R. Adaptive Differential Visual Feedback for Uncalibrated Hand-Eye Coordination and Motor Control, <http://www.cs.rochester.edu/u/{jag,nelson}/> December 14, 1994.
- [**Jäge95**] Jägersand, M., Nelson, R. Visual Space Task Specification, Planning and Control, in Proc of IEEE Int. Symp. On Computer Vision, 1995, p 521-526.
- [**Jäge96**] Jägersand, M., Nelson, R. On-line of Visual Models using Active Vision, in Proc. of ARPA Understanding Workshop, 1996.
- [**Kahn69**] Kahn, M. The Near-Minimum Time Control of Open-Loop Articulated Kinematics Linkages. Technical Report AIM-106, Stanford University, 1969.
- [**Kane83**] Kane, T., Levinson, D. The Use of Kane's Dynamical Equations in Robotics. Int. Journal Robot. Res., 2(3):3-21, Fall, 1983.
- [**Khat87**] Khatib, O., A Unified Approach for Motion and Force Control of Robots Manipulators: the Operational Space Formulation. IEEE Trans. Robot. Autom., 3(1), pp.43-53, February, 1987.
- [**Kova76**] Kovács, F.,V., Perju, D., Savii, Gh., Metode noi în sinteza mecanismelor. Editura Facla Timișoara, 1976.
- [**Kova81a**] Kovács, F.,V., Considerații cu privire la arhitectura generală a roboților industriali. Primul Simpozion Național de Roboți Industriali, București, 1981.
- [**Kova81b**] Kovács, F.,V., Contribuții la elaborarea unei terminologii de robotică. Primul Simpozion Național de Roboți Industriali, București, 1981.

- [Kova82a]** Kovács, F.,V., ș.a., Principii în tipizarea seriilor unitare și familiilor de roboți industriali. Al treilea Simpozion Național de Roboți Industriali, 1982.
- [Kova82b]** Kovács, F.,V., Cojocaru, Gh., Manipulatoare, roboți și aplicațiile lor industriale. Editura Facla, Timișoara, 1982.
- [Kova91]** Kovács, F.,V., Cu privire la determinarea exactității și incertitudinii de poziționare a roboților industriali programați prin instruire. MERO 91 – București, 1991.
- [Kova92]** Kovács, F.,V., Rădulescu, C. Roboți Industriali, vol.I-II, Lito. Universitatea "Politehnica" Timișoara, 1992.
- [Kova94]** Kovács, F.,V., Grigorescu, S., Rădulescu, C. Sisteme de fabricație flexibilă robotizate, vol.I-II, Lito. Universitatea "Politehnica" Timișoara, 1992.
- [Kova97]** Kovács, F.,V., The "Robotic Family" Simpozion Internațional, "Teoria și practica mecanismelor", SYROM97, București, 1997.
- [Kova99a]** Kovacs, Fr.V., Tarcă, R., Blaga, Fl., Tripe, V.A. Sisteme de fabricație flexibilă, Editura Universității din Oradea, 1999.
- [Kova99b]** Kovacs, Fr.V., Tusz, Fr., Varga, Șt., Fabrica viitorului, Editura Multimedia Internațional Arad, 1999.
- [Kova99c]** Kovacs, Fr.V., Utilizarea noțiunii de conexiune cinematică în Teoria mecanismelor și Robotică. Dizertație pentru obținerea titlului științific onorific Doctor Honoris Causa, Universitatea Craiova, 1999.
- [Kova00]** Kovács, F.,V., Varga, Șt., Pau, V.C., Introducere în Robotică, Editura Printech, București, 2000.
- [Lipo94]** Lipovszki, G., Automatic Parameter Search for Model Reference Adaptive Control. PhD Thesis, Technical University of Budapest, 1994.
- [Luh80]** Luh, J.Y.S., Walker, M.W., Paul, R.P.C. On-line Computation Scheme for Mechanical Manipulators. ASME Journal of Dynamic Systems, Measurement and Control, 102:69-76, 1980.
- [Mart80]** Martins, H., Birk, J., Kelley, R., Camera Models Based on Data from Two Calibration Planes. Computers Graphics and Image Processing, 17:173-180, 1980.
- [Mani98]** Maniu, I., Sisteme hidraulice și pneumatice de acționare, Editura Orizonturi Universitare, Timișoara, 1998.
- [Murr84]** Murray, J.,J., Computational Robot Dynamic, PhD Thesis, Carnegie-Mellon University, 1984.
- [Nega91]** Negahdaripour, S., Fox, J. Undersea Optical Stationkeeping: Improved Methods, J. Robot Syst., vol.8, pp.319-338,1991.

- [Negr97] Negrean, I., Vușcan, I., Haiduc, N., Robotică. Modelul cinematic și dinamic. Editura Didactică și Pedagogică, București, 1997.
- [Negr98] Negrean, I., Vușcan, I., Haiduc, N., Robotics. Kinematic and Dynamic Modelling. Editura Didactică și Pedagogică, București, 1998.
- [Orin79] Orin, D., McGhee, R., Vukobratovic, M., Hartoch, G. Kinematics and Kinetic Analysis of Open-Chain Linkages Utilising Newton-Euler Methods, Mathematical Biosciences, an International Journal, 43(1/2):107-130, February, 1979.
- [Paul81] Paul, R.,P. Robot Manipulators: Mathematics, Programming and Control. MIT Press, Cambridge, Massachusetts, 1981.
- [Petr84] Petrescu, A., Moisa, T., Țăpuș, N., Gayraud, A., Botez, C., Microcalculatoarele Felix M18, M18B, M118, vol.1, Editura Tehnică, București, 1984.
- [Popo92] Popovici, D., Servomecanisme, Litografia Universității Tehnice Timișoara, 1992.
- [Rank85] Ranky, P.,G., Ho, C., U., Robot Modelling. Control and Application with Software. Springer Verlag, Berlin, 1985.
- [Rizz91] Rizzi, A., Koditschek, D., Preliminary Experiments in Spatial Robot Juggling. In Proc 2nd International Symposium on Experimental Robotics, Toulouse, France, June, 1991.
- [Sear76] Sears, F.W., Zemasky, M.W., Young, H.D. Univesity Physics, Addison-Wesley Publishing Co., 1976.
- [Shar90] Sharkey, P., Daniel, R., Elosegui, P., Transputer Based Real Time Robot Control, In Proc. 29th CDC, vol. 2, pp.1161, December, 1990.
- [Silv82] Silver, R.A. On the Equivalence of Lagrangian and Newton-Euler Dynamics for Manipulators. Int. Journal Robot. Res., 1(2):60-70, Summer, 1982.
- [Sobe74] Sobel, I., On Calibrating Computer Controlled Cameras for Perceiving 3-D Scenes, Artificial Intelligence. An International Journal, 5:185-198, 1974.
- [Soml97] Somló, J., Lantos, B., Cat, P.T. Advanced Robot Control, Akadémiai Kiadó, Budapest, 1997.
- [Tank00] Tanko, T., Construcția modelului dinamic pentru un robot de tip TRTR utilizând ecuațiile Lagrange. Analele Universității din Oradea, 2000.
- [Tarc94a] Țarcă, R., Țarcă, I., Tripe V., A., Aspecte teoretice privind determinarea în timp real a poziționării efectorului final la roboți industriali folosind tehnici de percepție artificială, Analele Universității din Oradea, 1994.
- [Tarc94b] Țarcă, R., Țarcă, I., Tripe V., A., Aspecte teoretice privind determinarea în timp real a orientării efectorului final la roboți industriali folosind tehnici de percepție artificială, Analele Universității din Oradea, 1994.

- [Tarc96a] Țarcă, N., Țarcă, R., Țarcă, I., Program de analiză, recunoaștere și ordonare a corpurilor, Al IV-lea Congres al Academiei de Cibernetică "Ștefan Odobleja", Băile Felix, 1996.
- [Tarc96b] Țarcă R., Cuc C., Țarcă I., Tripe V., A., Program de analiză și recunoaștere a corpurilor, Simpozionul național de roboți, ediția a XIII a, Reșița, 1996.
- [Tarc96c] Țarcă, R., Țarcă, I., Tripe V., A., Construcția modelului dinamic pentru un robot de tip RRRR, Simpozionul național de roboți, ediția a XIII a, Reșița, 1996
- [Tarc98a] Țarcă R., Țarcă I., Vesselenyi T., Tocuț D., Simularea conducerii adaptive – utilizând un model de referință – a unei axe de robot – partea I, Simpozionul național de Robotică, ediția a XIV a, Brașov, 1998.
- [Tarc98b] Țarcă I., Țarcă R., Polojințef N., Cuc, C., Simularea conducerii adaptive – utilizând un model de referință – a unei axe de robot – partea I, Simpozionul național de Robotică, ediția a XIV a, Brașov, 1998.
- [Tarc98c] Țarcă, R., Țarcă, I., Tripe, V., A., The Simulation of Hydraulic Robot Axe Using an Adaptive Reference Model Control, part I,II, Second International Conference on Renewable Sources and Environmental Electro-Technologies, Felix-Spa, 1998.
- [Tarc98d] Țarcă, R. Conducerea adaptivă a roboților industriali utilizând informații furnizate de senzori, Referatul doctorat nr.3, Universitatea Politehnica Timișoara, 1998.
- [Tarc99] Țarcă, R., Țarcă, I., Theoretical Aspects about Stability Analysis in a Visual Servosystem, The Third International Conference on Renewable Sources and Environmental Electro-Technologies, Felix-Spa, 1998.
- [Tarc00a] Țarcă, R., Țarcă, I., Utilizarea sistemelor servovizuale în comanda unor componente ale sistemelor de fabricație flexibilă, pp. 407-416, Tendințe în Cibernetică Contemporană, Editura Europa Nova, București, 2000.
- [Tarc00b] Țarcă, R., Țarcă, I., Strategii de conducere specifice roboților adaptive, Analele Universității din Oradea, 2000.
- [Tarc00c] Țarcă, R., Țarcă, I., Țarcă N., Program de conducere a unui sistem de urmărire a corpurilor luminoase în mișcare, Conferința națională a Academiei de Cibernetică "Ștefan Odobleja", Lugoj, mai, 2000.
- [Tarc01] Țarcă, R., Țarcă, I., Polojințef, N., 3D Model for a Visual Servoing Device, MicroCAD Symposium, Miskolc, febr., 2001.
- [Trip00] Tripe Vidican, A., Optimizarea constructivă și funcțională a manipuletoarelor și roboților industriali cu acționare hidraulică. Teză de doctorat. Universitatea "Politehnica" Timișoara, 2000.

- [Tsai87]** Tsai, R., A Versatile Camera Calibration Technique for High Accuracy 3-D Machine Vision Metrology Using off-the-shelf TV Cameras and Lenses. IEEE Trans. Robot. Autom. 3(4):323-344, August, 1987.
- [Uick65]** Uicker, J. On the Dynamic Analysis of Spatial Linkages Using 4 by 4 Matrices. PhD thesis, Dept. Mechanical Engineering and Astronautically Sciences, NorthWestern University, 1965.
- [Vlai97]** Vlaicu, A., Prelucrarea digitală a imaginilor, Editura Albastră, Cluj-Napoca, 1997.
- [Vuko85a]** Vukobratovic, M., Kirkanski, N. An Approach to Adaptive Control of Robotic Manipulators. Automatica vol. 21, No. 6, 1985.
- [Vuko85b]** Vukobratovic, M., Kirkanski, N. Real-Time Dynamics of Manipulators Robots. Springer Verlag, Berlin, 1985.
- [Vuko85c]** Vukobratovic, M., Kirkanski, N. Nonadaptive and Adaptive Control of Manipulators Robots. Springer Verlag, Berlin, 1985.
- [Warn89]** Warneke, H.J., Schraft, R.D.: - Industrieroboter. Krausskopf-Verlag, GmbH Mainz, 1989.
- [Weis84]** Weiss L.E. Dynamic Visual Servo-Control of Robots: An Adaptive Image-Based Approach, PhD Thesis, Carnegie-Mellon University, 1984.
- [WIR97]** World Industrial Robots 1997, United Nations – Economic Commission for Europe (UN/ECE), International Federation of Robotics, New York and Geneva, 1997.
- [Wolf74]** Wolf, P., Elements of Photogrammetry, McGraw-Hill, 1974.
- [Wong75]** Wong, K.W., Mathematic Formulation and Digital Analysis in Close-Range Photogrammetry. Photogrammetric Engineering and Remote Sensing, 41(11):1355-1373, November, 1975.

Exprimarea tabelară a valorilor parametrilor generalizați ai robotului și ai subsistemului vizual de urmărire determinați - utilizând relațiile (6.3) și respectiv (6.5) – pentru legea de mișcare dată și cele 30 de puncte via

Nr.crt.	x	y	z	α (rad)	β (rad)	θ (rad)	α (deg)	β (deg)	θ (deg)
1.	100,00	0,00	100,00	-0,34	-0,46	0,07	-19,65	-26,57	4,09
2.	133,33	40,00	126,67	-0,32	-0,45	0,09	-18,60	-25,78	5,30
3.	166,67	80,00	153,33	-0,31	-0,44	0,11	-17,48	-24,94	6,56
4.	200,00	120,00	180,00	-0,28	-0,42	0,14	-16,29	-24,04	7,88
5.	233,33	160,00	206,67	-0,26	-0,40	0,16	-15,03	-23,09	9,27
6.	266,67	200,00	233,33	-0,24	-0,39	0,19	-13,67	-22,07	10,71
7.	300,00	240,00	260,00	-0,21	-0,37	0,21	-12,23	-20,97	12,23
8.	333,33	280,00	286,67	-0,19	-0,35	0,24	-10,68	-19,80	13,80
9.	366,67	320,00	313,33	-0,16	-0,32	0,27	-9,02	-18,54	15,45
10.	400,00	360,00	340,00	-0,13	-0,30	0,30	-7,25	-17,18	17,18
11.	433,33	400,00	366,67	-0,09	-0,27	0,33	-5,36	-15,71	18,97
12.	466,67	440,00	393,33	-0,06	-0,25	0,36	-3,32	-14,12	20,84
13.	500,00	480,00	420,00	-0,02	-0,22	0,40	-1,15	-12,41	22,78
14.	533,33	520,00	446,67	0,02	-0,18	0,43	1,19	-10,55	24,80
15.	566,67	560,00	473,33	0,06	-0,15	0,47	3,68	-8,53	26,89
16.	600,00	600,00	500,00	0,11	-0,11	0,51	6,34	-6,34	29,05
17.	566,67	640,00	526,67	0,15	-0,06	0,51	8,53	-3,68	29,44
18.	533,33	680,00	553,33	0,18	-0,02	0,52	10,55	-1,19	29,79
19.	500,00	720,00	580,00	0,22	0,02	0,53	12,41	1,15	30,11
20.	466,67	760,00	606,67	0,25	0,06	0,53	14,12	3,32	30,42
21.	433,33	800,00	633,33	0,27	0,09	0,54	15,71	5,36	30,70
22.	400,00	840,00	660,00	0,30	0,13	0,54	17,18	7,25	30,96
23.	366,67	880,00	686,67	0,32	0,16	0,54	18,54	9,02	31,21
24.	333,33	920,00	713,33	0,35	0,19	0,55	19,80	10,68	31,44
25.	300,00	960,00	740,00	0,37	0,21	0,55	20,97	12,23	31,66
26.	266,67	1000,00	766,67	0,39	0,24	0,56	22,07	13,67	31,87
27.	233,33	1040,00	793,33	0,40	0,26	0,56	23,09	15,03	32,06
28.	200,00	1080,00	820,00	0,42	0,28	0,56	24,04	16,29	32,24
29.	166,67	1120,00	846,67	0,44	0,31	0,57	24,94	17,48	32,42
30.	133,33	1160,00	873,33	0,45	0,32	0,57	25,78	18,60	32,58
31.	100,00	1200,00	900,00	0,46	0,34	0,57	26,57	19,65	32,74

Determinarea coeficientului de frecare vâscoasă și a coeficientului de frecare uscată pentru servomotorul de curent continuu

În figura A2.1 se prezintă variația momentului de frecare funcție de viteza generalizată. Linia întreruptă reprezintă modelul simplu al frecării, având ecuația:

$$M_e = B_a \cdot \dot{q} + M_F \quad (\text{A2.1})$$

unde panta dreptei B_m reprezintă *frecarea vâscoasă*, iar offset-ul reprezentându-l *frecarea uscată*. Acest ultim termen se poate modela printr-o funcție nelineară:

$$M_F = \begin{cases} 0 & \text{daca } \dot{q} = 0 \\ M_F^+ & \text{daca } \dot{q} > 0 \\ M_F^- & \text{daca } \dot{q} < 0 \end{cases} \quad (\text{A2.2})$$

și în general $|M_F^+| \neq |M_F^-|$.

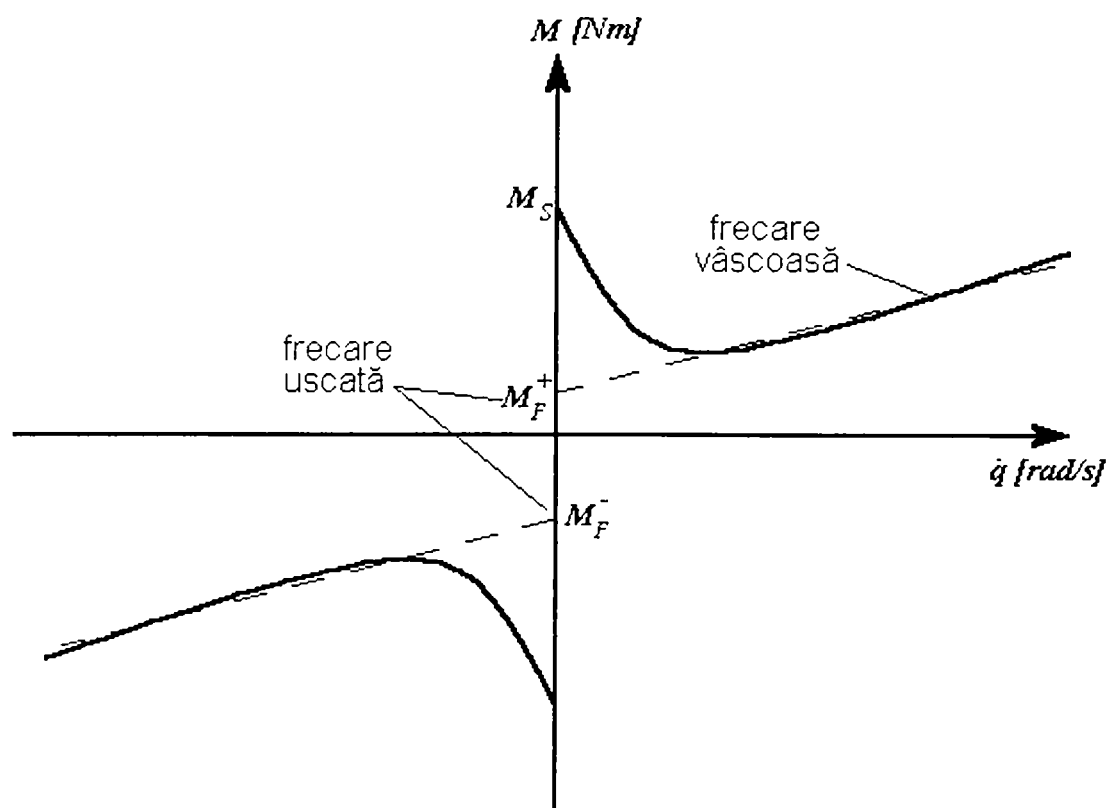


Fig.A2.1. Variația momentului de frecare în funcție de viteza generalizată

Frecarea statică este torsorul necesar pentru a trece un motor de acționare din stare de repaus în stare de mișcare și după cum se observă din figura A2.1 este mai mare în valoare absolută decât frecarea uscată. Modelul complex al frecării, reprezentat prin linie continuă devine important doar pentru viteze generalizate scăzute. Panta negativă

poate fi atribuită efectului Stribeck datorat ungerii mixte și datorită ei conducerea utilizând un algoritm PID simplu poate fi instabil la viteze generalizate reduse.

Modul de lucru pentru determinarea coeficienților frecării vâscoase și uscate este următorul:

Motorul de acționare se pune în mișcare la o viteză unghiulară constantă și se calculează momentul generat de motor (se măsoară curentul absorbit de motor, iar cu relația (3.45) se determină momentul generat de motor). Se repetă măsurătorile pentru un set de viteze unghiulare (vitezele unghiulare se determină utilizând un traductor incremental de rotație), atât într-un sens cât și în celălalt, după care se determină panta dreptei (coeficientul de frecare vâscoasă B_a) dar și intersecția ei cu axa forței generalizate (coeficientul de frecare uscată M_F). Pentru determinarea frecării statice se crește intensitatea curentului de alimentare a motorului cuplei până când cupla se pune în mișcare.

Dacă se repetă experimentul, dar cu motorul în sarcină (motorul realizând deplasarea axei dispozitivului de ghidare) cu o bună aproximare se poate considera că panta dreptei ce aproximează graficul - moment de frecare în funcție de viteză generalizată – este chiar coeficientul global de frecare vâscoasă B_{ef} .

Schema de principiu a standului pentru determinarea coeficientului global de frecare vâscoasă se prezintă în figura A2.2.

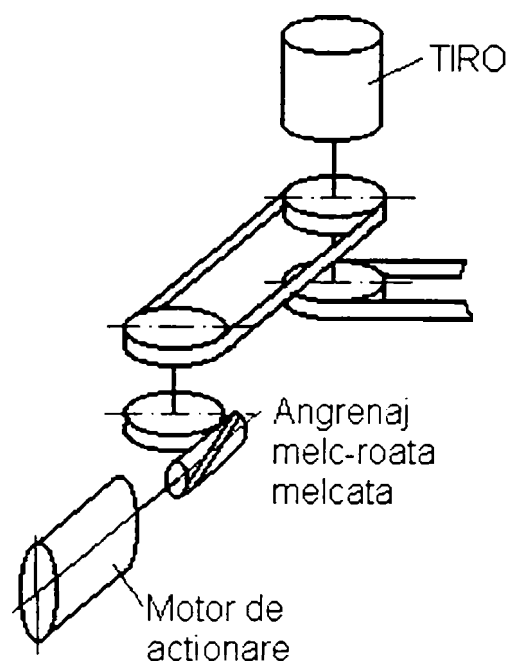


Fig.A2.2. Schema de principiu a standului pentru determinarea coeficientului global de frecare vâscoasă

Vederea de ansamblu a standului pe care s-a determinat coeficientul global de frecare vâscoasă B_{ef} pentru axa orizontală a dispozitivului de ghidare este prezentată în figura A2.3.

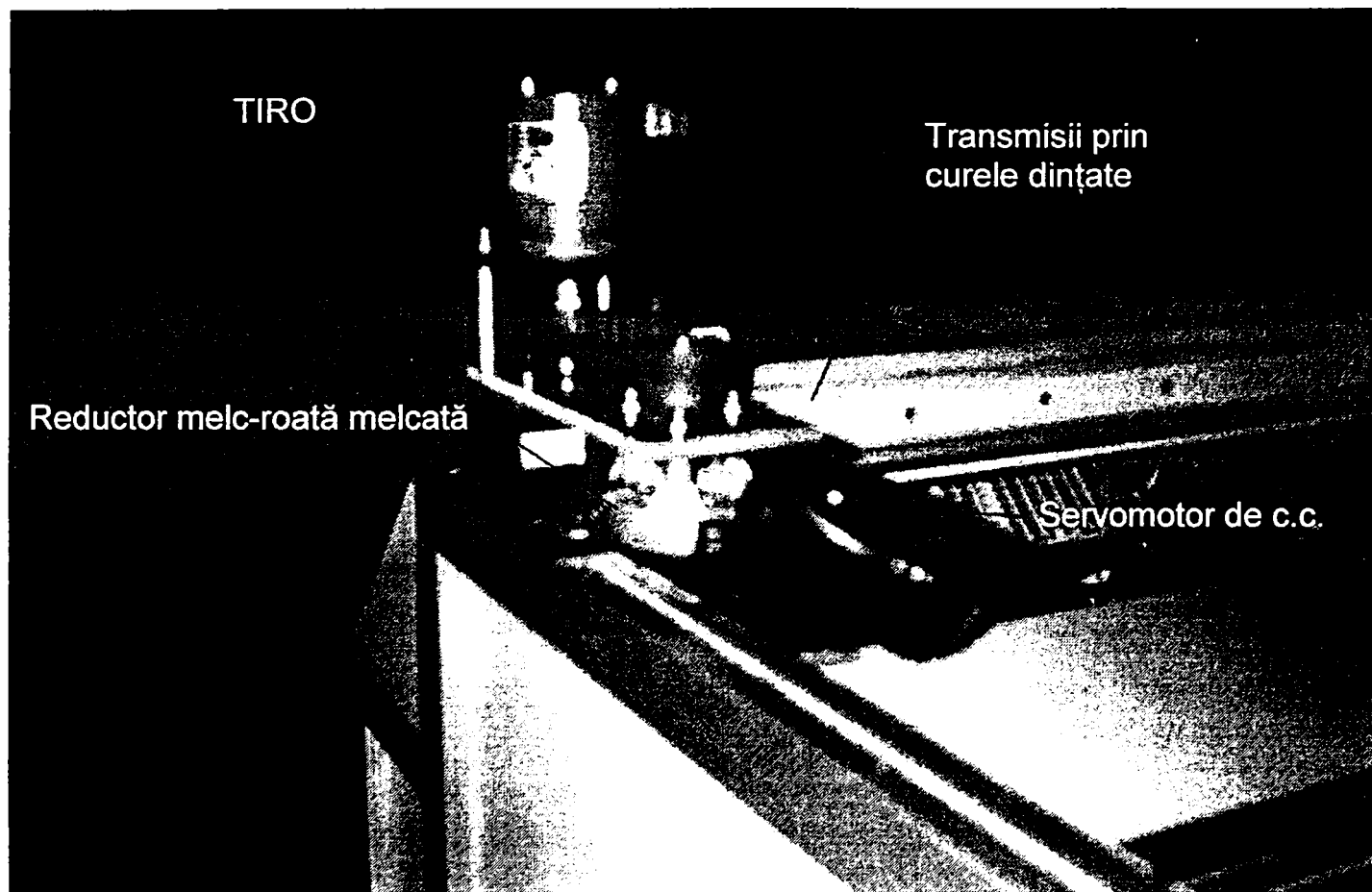


Fig. A2.3. Vedere de ansamblu a standului realizat în vederea determinării coeficientului global de frecare vâscoasă B_{ef} pentru axa orizontală a dispozitivului de ghidare a robotului

Pentru determinarea vitezei unghiulare s-a realizat un program, care citește și scrie într-un fișier valorile unghiului de rotație a axului traductorului în timp. Ordinograma programului se prezintă în figura A2.4.

Pentru citirea variației în timp a coordonatei generalizate a servomotorului de c.c. se utilizează un traductor incremental de rotație TIRO 1000, sistemul de acționare a c.c.c. placa de achiziție PCI 1200 montată pe un calculator.

După inițializarea plăcii de achiziție PCI 1200 și inițializarea contorului care va realiza numărarea impulsurilor generate de traductor, se va deschide fișierul „Coorodnată_generalizată.dat”.

La apăsarea unei taste se pornește ciclul de măsurare și înregistrare date în fișier, în care se citesc valorile contorului, a valoare a ceasului intern urmată de salvarea acestor date în fișierul deschis.

Buclo se va repeta până la tastarea unei taste.

La ieșirea din buclă se vor reseta funcțiile plăcii de achiziție PCI 1200.

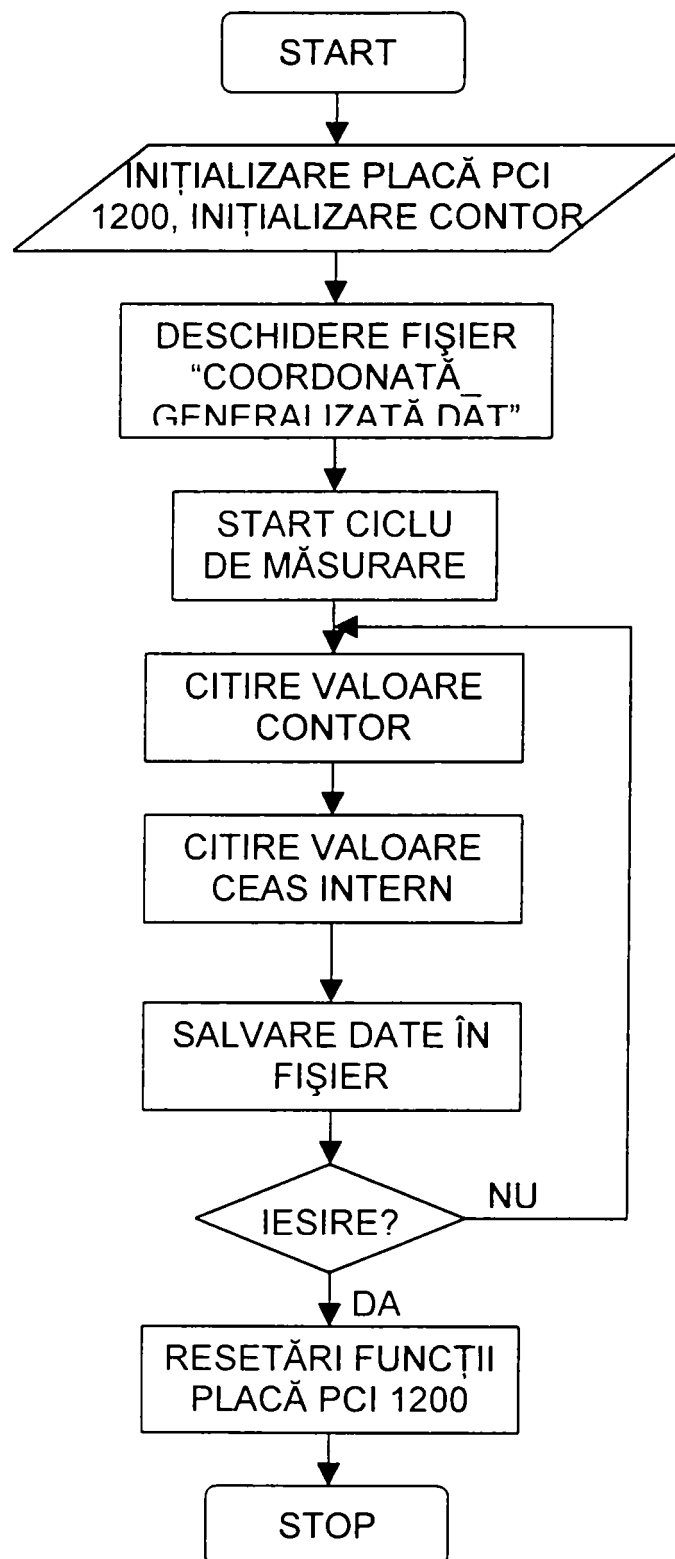


Fig.A.4. Ordinograma programului de citire a variației unghiului de rotație a axului traductorului în timp

Utilizând datele astfel obținute se poate calcula ușor viteza unghiulară a servomotorului ținând cont de raportul de transmitere a reductorului și a transmisiei cu curea dințată.

În figurile A2.5 și respectiv A2.6 se prezintă rezultatele obținute ale variației momentului de frecare în funcție de viteza generalizată pentru deplasarea spre stânga, respectiv dreapta a c.c.c. de translație orizontală a robotului.

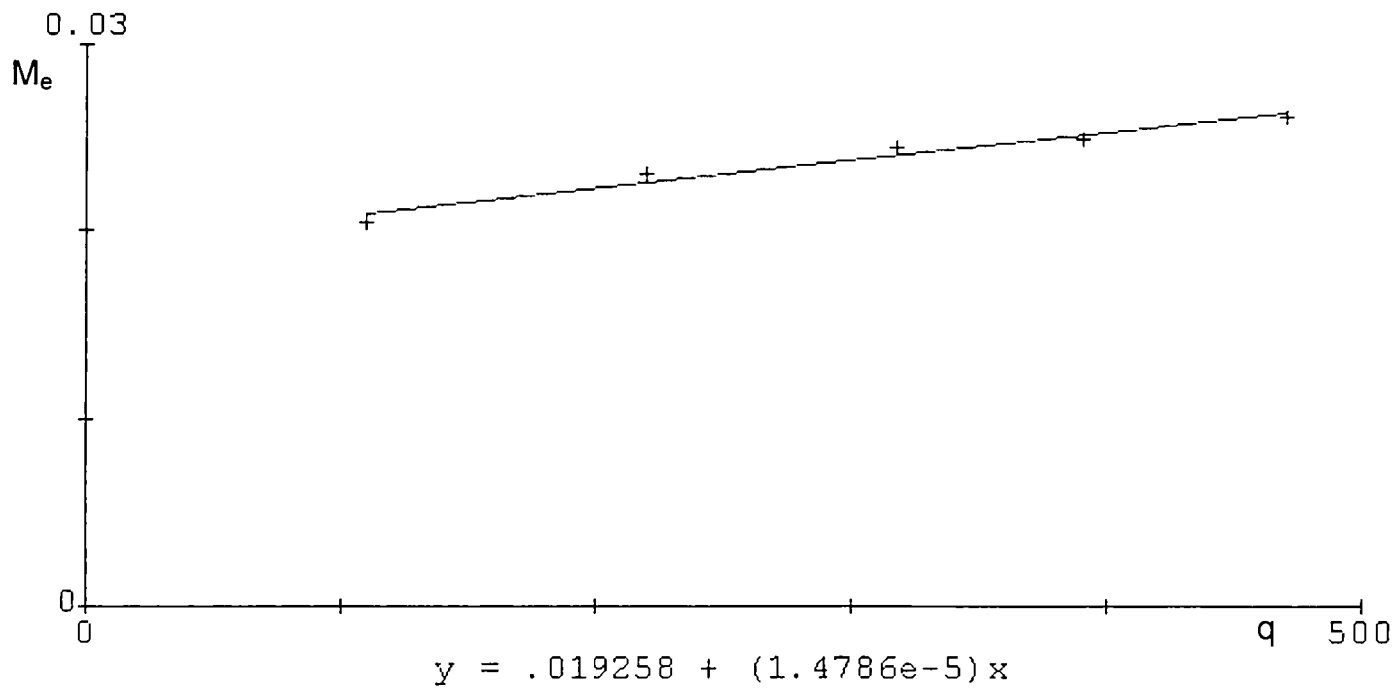


Fig.A2.5. Variația momentului de frecare funcție de viteza generalizată pentru deplasarea spre stânga a c.c.c. de translație orizontală a robotului

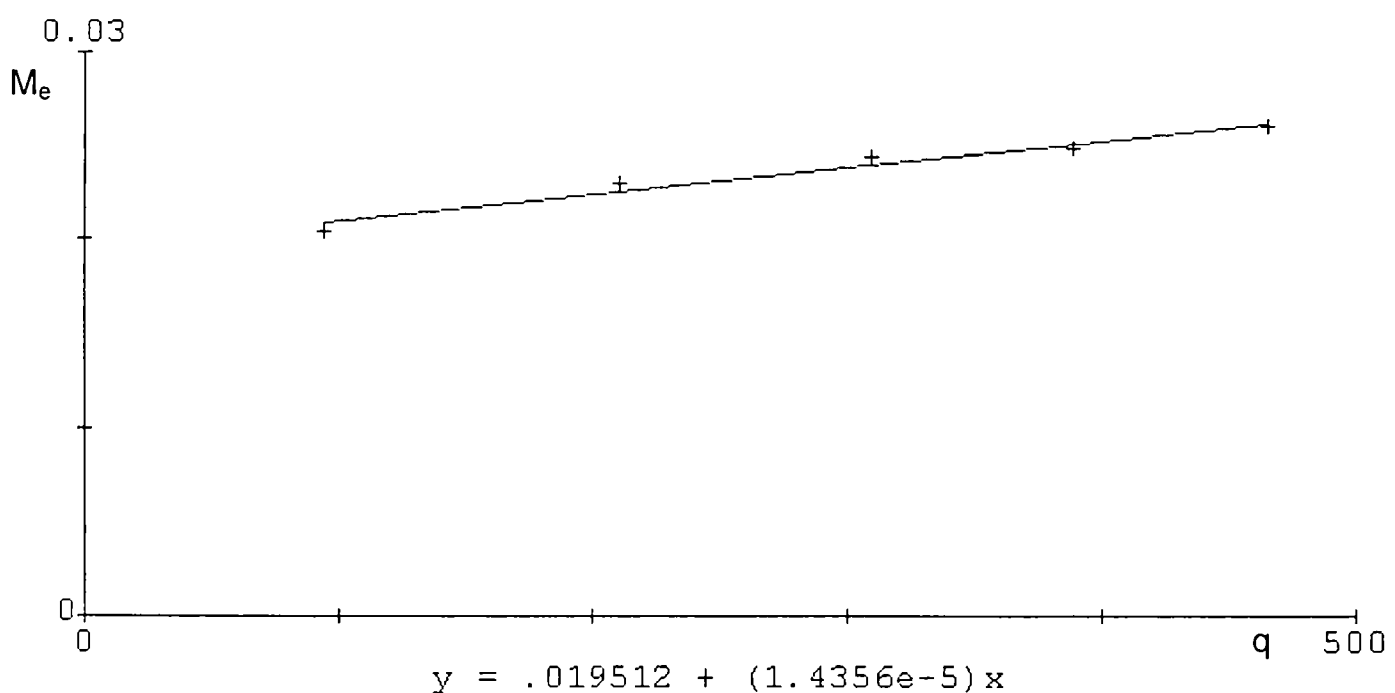


Fig.A2.6. Variația momentului de frecare funcție de viteza generalizată pentru deplasarea spre dreapta a c.c.c. de translație orizontală a robotului

Din rezultate experimentale se pot determina:

- coeficientul de frecare vâscoasă B_a din valoarea pantei dreptei care aproximează printr-o funcție lineară valorile obținute experimental;

- coeficientul de frecare uscată M_F din valoarea obținută din intersecția dreptei cu axa forței generalizate.

Prin urmare, coeficientul de frecare vâscoasă B_a și coeficientul de frecare uscată M_F vor avea valori diferite în cele două sensuri și vor fi:

$$B_a = 14.786 \cdot 10^{-6} \text{ [Nms/rad]} \text{ pentru deplasarea spre stânga;}$$

$$B_a = 14.356 \cdot 10^{-6} \text{ [Nms/rad]} \text{ pentru deplasarea spre dreapta;}$$

$$M_F = 0,019258 \text{ [Nm]} \text{ pentru deplasarea spre stânga;}$$

$$M_F = 0,019512 \text{ [Nm]} \text{ pentru deplasarea spre dreapta.}$$

**Fișierul „val_par_1.m”
cu inițializările pentru simularea sistemului de conducere adaptivă realizat**

```
% Se specifica marimile, unitatile de masura
% si apoi se atribuie valori marimilor.

% Motor pas cu pas

%Parametrii generatorului de semnale
% - A -> Amplitudine - Volti
% - St -> Sample time - secunde

App = 5
Stpp = 0.1

% Parametrii sarcinii
% J -> Moment de inertie Nm^2
% B -> Amortizare - coeficient adimensional

Jpp = 0.000001
Bpp = 0.0001

% Parametrii motorului
% L -> Inductanta - Henri
% R -> Rezistenta - Ohm
% n -> Numar de poli
% delta -> pasul unghiular al motorului pas cu pas

Lpp = 0.00176
Rpp = 5
npp = 25
delta = 2*pi/(2*npp*4)

% Parametrii cinematici
% Kg -> Transmisia mecanica - coeficient adimensional
% Km -> Coeficient motor - coeficient adimensional

Kgpp = 1
Kmpp = 1

% Motor de curent continuu

%Parametrii generatorului de semnale

% - Kch -> Parametru chopper

Kch = 18/0.01

% Parametrii sarcinii
% J -> Moment de inertie Nm^2
% B -> Amortizare - coeficient adimensional
```

Jcc = 0.99e-6
Bcc = 14.57e-6

% Parametrii motorului
% L -> Inductanta - Henri
% R -> Rezistenta - Ohm

Lcc = 0.0015
Rcc = 1.8

% Parametrii cinematici
% Kg -> Transmisia mecanica - coeficient adimensional
% Km -> Coeficient motor - coeficient adimensional

Kgcc = 1/120
Kg1cc = 0.03
Kmcc = 1
Ker = 3e-5
% Senzor vizual

alfa = 0
dist = 1.5
dist_focal = 0.03
n_pixel = 640/0.008

% Sistem de calcul

x_cadru = 80

PREZENTAREA PLĂCII MATROX METEOR

În această anexă se face o prezentare generală a plăcii MATROX METEOR, a caracteristicilor acesteia a modului de achiziție și de transfer a imaginii și a pachetelor de programe utilizate împreună cu această placă.

MATROX METEOR este o placă de achiziție de imagini monocrome sau color (după opțiune) ce asigură transferul în timp real a imaginii achiziționate de către un senzor vizual matriceal în vederea stocării sau prelucrării sale de către un calculator și este în același timp și suportul pentru software-ul interactiv de imagine oferit de către firma furnizoare (fie pachetul de programe MIL – Matrox Imaging Library, fie pachetul de programe MI – Matrox Inspector).

Funcție de modalitatea de achiziție a imaginii, MATROX METEOR se poate prezenta în două versiuni și anume:

- MATROX METEOR CLASIC ce realizează achiziții de imagini monocrome sau color standard de tipul:

- Luminanță/Crominanță Y/C (S-video; S-VHS, HI8);
- RS – 170 (640x480)/CCIR (768x576);
- NTSC (640x480) / PAL (768x576) / SECAM (768x576).

La această variantă a plăcii de achiziție sunt disponibile până la patru canale de intrare video selectabile din soft.

- MATROX METEOR / RGB reprezintă o versiune evoluată a variantei anterioare, care prezintă în plus un modul RGB (red – green - blue) ce suportă achiziția imaginilor color pe 24 de biți de la o cameră standard RGB, permițând controlul independent al câștigului și offset-ului pentru canalele R, G și B. Placa dispune de o intrare TTL trigerată în vederea sincronizării achiziției cu un eveniment extern.

Placa de achiziție transferă informațiile de tip imagine în timp real fie în memoria RAM a CPU în vederea procesării, fie către bufferul display-ului în vederea afișării lor.

Semnalul video de intrare poate fi reglat prin soft, putându-se ajusta luminozitatea, contrastul, nuanțele, saturația și detaliile.

MATROX METEOR asigură sincronizări foarte bune chiar dacă achiziția de la camerele video sau videorecordere se face în mod playback sau în mod pauză.

În cazul în care placa trimite informația către bufferul displayului în vederea afișării imaginii într-o fereastră, fereastra poate fi scalată la orice dimensiune și poziționată oriunde pe ecran.

Imaginea digitală spre unitatea de interfață PCI suportă multiple formate de transfer de date (8 biți mono, 15 biți și 24 biți RGB).

Într-un calculator pot fi instalate mai multe plăci, dar numărul lor maxim depinde de specificul sistemului și respectiv de specificul aplicației.

MATROX METEOR lucrează împreună cu placa video Matrox MGA Millenium pentru afișarea în timp real a imaginii dar și cu orice placă VGA în acest caz însă realizându-se o pseudocolorare a imaginii.

În figura A4.1. se prezintă diagrama bloc a plăcii MATROX METEOR.

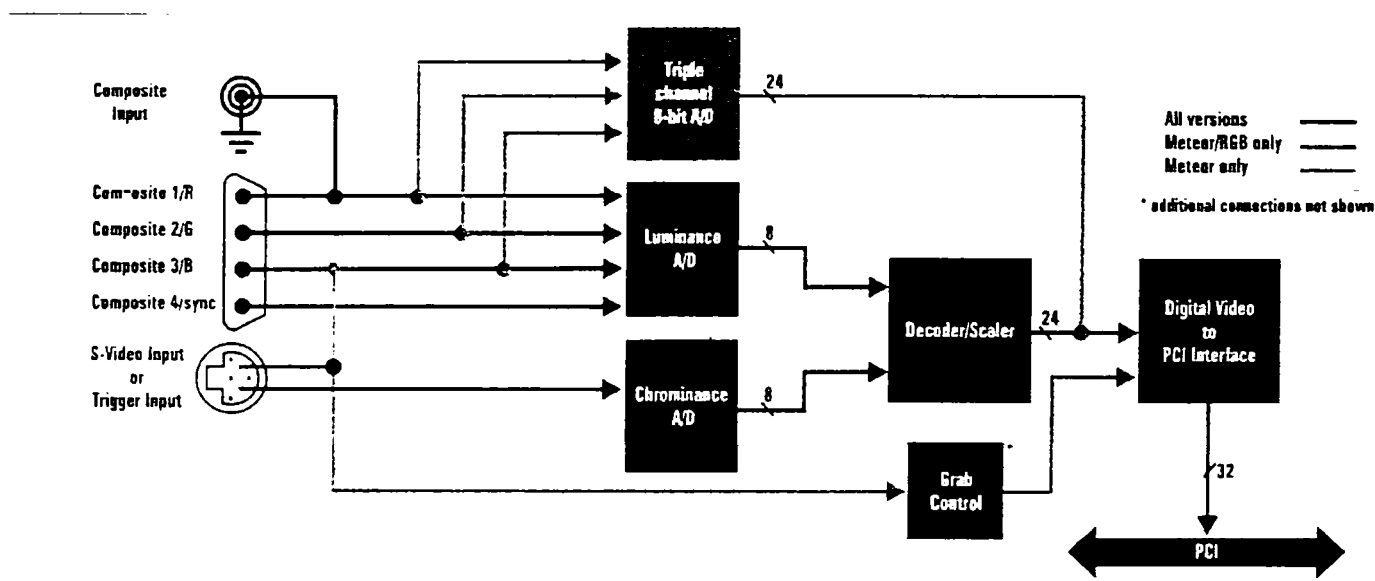


Fig. A4.1. Diagrama bloc a plăcii MATROX METEOR

Transferul în timp real al imaginii depinde de placa VGA sau de sistemul utilizat și este de până la 45 MB/sec. Achiziția în timp real pe sistem sau în memoria ecranului necesită:

- 9 MB/sec pentru 640x480x8 @ 30 fps RS – 170;
- 11 MB/sec pentru 768x576x8 @ 25 fps CCIR;
- 35 MB/sec pentru 640x480x24 @ 30 fps NTSC sau RGB;
- 42 MB/sec pentru 768x576x24 @ 25 fps PAL sau RGB.

Interfața în cazul acestei plăci este una PCI pe 32 de biți.

În continuare sunt prezentate legăturile corespunzătoare pinilor conectorilor de intrare:

- conectorul video de intrare

PIN	DESCRIERE
1.	Compoziția 1 sau R (roșu)
2.	Compoziția 2 sau G (verde)
3.	Compoziția 3/sau luminanța de intrare sau B (albastru)
4.	Compoziția 4 sau compoziția Sync
5.	Masa
6.	Masă.
7.	Masă.
8.	Masă.
9.	Masă.

□ conector de intrare Y/C

PIN	DESCRIERE
1.	Masă.
2.	Masă.
3.	Luminanța de intrare(la varianta clasică) / Intrare trigger (varianta RGB)
4.	Crominanța de intrare

□ Jack audio pentru intrare compozită:

PIN	DESCRIERE
1.	Compoziția 1
2.	Masă.

Data fiind complexitatea echipamentului hard realizarea efectivă a achiziției/controlului se face simplu, prin programe de aplicație. Prelucrarea propriu-zisă a datelor se face cu programe de aplicație, fie cele livrate de firma producătoare odată cu placa, fie programe scrise de utilizator.

Programele utilizator pot apela funcțiile din biblioteca proprie a plăcii, din programe scrise în limbaje de nivel înalt.

PREZENTAREA PLĂCII NATIONAL INSTRUMENTS PCI 1200

În această anexă se face o prezentare generală a plăcii NI PCI 1200, a circuitelor pentru intrările și ieșirile analogice ale acesteia și diverse configurații de utilizare în regim static.

Placa de achiziție NI PCI 1200 conține opt canale de intrări analogice, două multiplexoare CMOS, un amplificator cu câștig programabil soft, un convertor analog-numeric de 12 biți și o memorie FIFO de 512 cuvinte. Grație circuitelor de sincronizare, placa poate sincroniza în mod automat conversii analog-numerice multiple. Cele opt semnale analogice sunt preluate de cele două multiplexoare CMOS, amplificate, convertite numeric și păstrate în memoria FIFO. În fig. A5.1 este prezentată schema bloc a plăcii, iar în fig. A5.2 diagrama bloc a circuitelor de achiziție analogică.

Sistemul multicanal al intrărilor analogice poate fi utilizat în analiza semnalelor și în stocarea de date. Convertorul analog-numeric poate fi folosit în aplicații de înaltă rezoluție cum ar fi cromatografia sau măsurătorile de temperatură. Canalele de ieșiri analogice pot fi folosite pentru generarea de stimuli experimentali, de funcții analogice, sau pot controla procese fizice. Cele 24 linii digitale de intrare-ieșire compatibile TTL pot folosi la comutarea unor unități externe (tranzistoare sau relee), pentru citirea de stări logice sau pentru generarea de întreruperi. Elementele de sincronizare-numărare pot sincroniza procese, să genereze impulsuri, să măsoare frecvențe și durate.

Placa NI PCI 1200 se conectează la slotul de extensie a magistralei de date, adrese și control a calculatorului.

Legătura plăcii cu exteriorul (de fapt, cu procesele fizice controlate) se face prin conectorul de intrare-ieșire de 50 pini prezentat în fig. A5.3. Conform figurii A5.3 placa dispune de :

- 8 pini pentru intrările analogice ;
- 2 pini de masă pentru intrările-ieșirile analogice ;
- 2 pini pentru ieșirile analogice ;
- 24 de pini pentru intrările-ieșirile digitale ;
- 2 pini de masă pentru intrările-ieșirile digitale ;
- 8 pini pentru controlul celor trei numărătoare ;
- 1 pin pentru generarea unei tensiuni de +5V ;
- 3 pini pentru controlul extern al convertoarelor.

În tabelul A5.1. sunt prezentate funcțiile semnalelor corespunzătoare pinilor conectorului de intrare/ieșire.

Tabelul A5.1. Funcțiile semnalelor corespunzătoare pinilor conectorului de intrare/ieșire

PIN	SEMNAL	DESCRIERE
1...8	ACH0..ACH7	Canale de intrări analogice.
9	AISENSE/AIGND	Masa pentru intrări analogice în modurile RSE, AISENSE în modul NRSE. Bidirecțional.
10	DAC0OUT	Canalul 0 de ieșire analogică.
11	AGND	Masă analogică. Bidirecțional.
12	DAC1OUT	Canalul 0 de ieșire analogică.
13	DGND	Masă digitală. Ieșire.
14...21	PA0.. PA7	Linii bidirecționale pentru portul digital PA. PA7 este MSB; PA0 este LSB.
22...29	PBO..PB7	Linii bidirecționale pentru portul digital PB. PB7 este MSB; PB0 este LSB.
30...37	PCO..PC7	Linii bidirecționale pentru portul digital PC. PC7 este MSB; PC0 este LSB.
38	EXTTRIG	Semnal extern de control, pentru startul unei secvențe sincronizate. Intrare.
39	EXTUPDATE*	Semnal extern de control, pentru preluarea ieșirilor CAN. Intrare.
40	EXTCONV*	Semnal extern de control, pentru controlul conversiilor A/N. Bidirecțional.
41	OUTBO	Ieșirea numărătorului B0.
42	GATBO	Poartă numărător B0. Intrare.
43	COUTB1	Ieșirea sau setarea numărătorului B1. Selectabil.
44	GATB1	Poartă numărător B1. Intrare.
45	CCLKB1	Ceas numărător B1 (selectabil). Intrare.
46	OUTB2	Ieșirea numărătorului B2.
47	GATB2	Poartă numărător B2. Intrare.
48	CLKB2	Ceas numărător B2. Intrare.
49	+5V	Ieșire +5V, max. 1A.
50	DGND	Masă digitală. Ieșire.

*Indică semnal activ pe nivel logic zero.

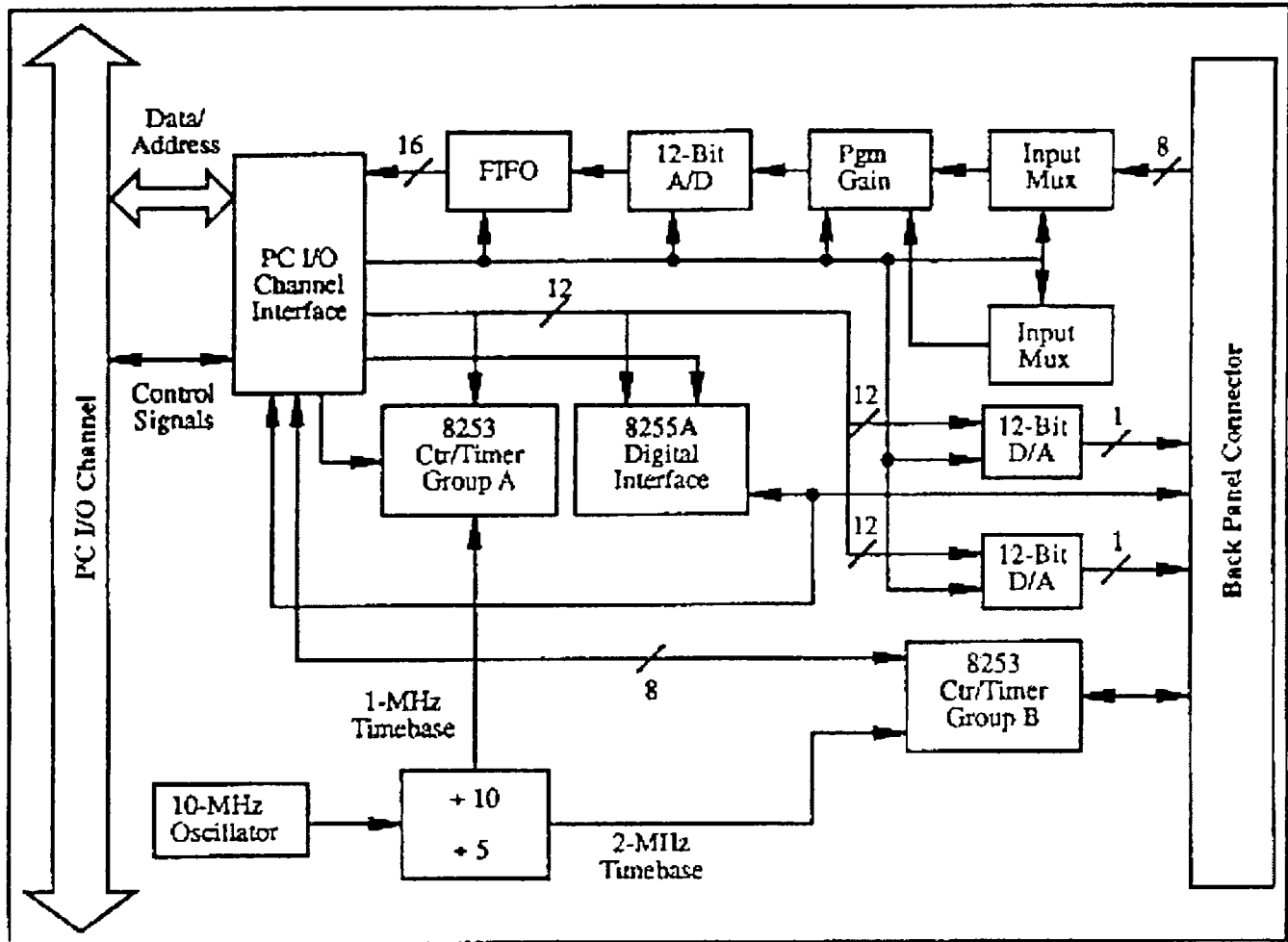


Fig. A5.1 Schema bloc a plăcii NI PCI 1200

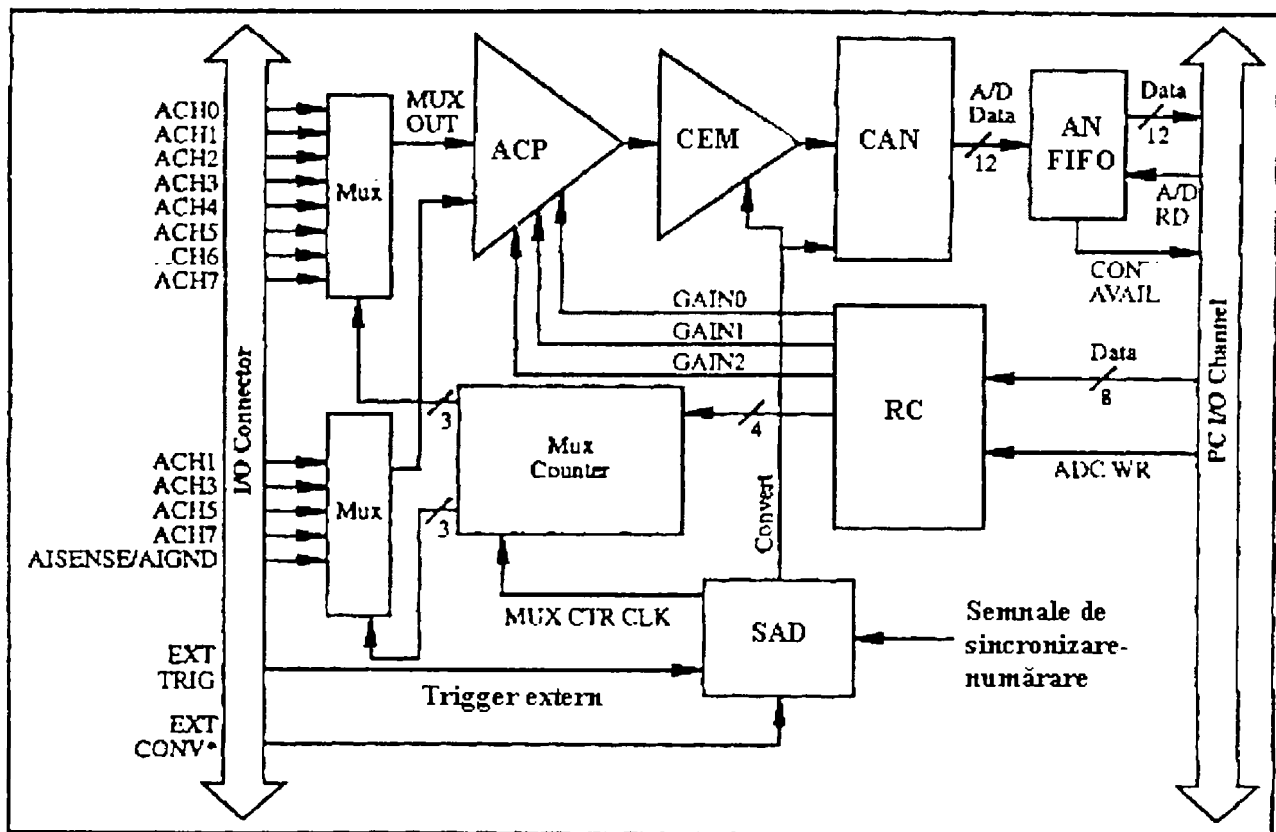


Fig. A5.2. Schema bloc a circuitelor de achiziție analogică a plăcii NI PCI 1200

SEMNAL	NR.	PIN	SEMNAL
ACH0	1	2	ACH1
CH2	3	4	ACH3
ACH4	5	6	ACH5
ACH6	7	8	ACH7
AISENSE/AIG	9	10	DAC0OUT
AGND	11	12	DAC1 OUT
DGND	13	14	PA0
PA1	15	16	PA2
PA3	17	18	PA4
PA5	19	20	PA6
PA7	21	22	PB0
PB1	23	24	PB2
PB3	25	26	PB4
PB5	27	28	PB6
PB7	29	30	PC0
PC1	31	32	PC2
PC3	33	34	PC4
PC5	35	36	PC6
PC7	37	38	EXTTRIG
EXUPDATE*	39	40	EXTCONV*
OUTBO	41	42	GATBBO
COUSB1	43	44	GATB1
CCLKB1	45	46	OUTB2
GATB2	47	48	CLKB2
+5V	49	50	DGND

Fig. A5.3. Schema conectorului cu exteriorul

Data fiind complexitatea echipamentului hard realizarea efectivă a achiziției/controlului se face simplu, prin programe de aplicație. Prelucrarea propriu-zisă a datelor se face cu programe de aplicație, fie cele livrate de firma producătoare odată cu placa, fie programe scrise de utilizator.

Programele utilizator pot fi la nivel de limbaj de asamblare, caz în care se citesc/scriu cuvinte de comandă și control în/din regiștrii funcționali ai plăcii sau se pot apela funcțiile din biblioteca proprie a plăcii, din programe scrise în limbaje de nivel înalt (Pascal, C, Basic).

Programarea directă, în limbaj de asamblare, a plăcii se face printr-un grup special de regiștri aflați pe placă: fiecare funcție a plăcii este asigurată de scrierea de cuvinte de comandă și control în regiștrii corespunzători.

Există 6 grupe de regiștri:

- grupul de configurare și stare, care controlează și monitorizează marea majoritate a operațiunilor plăcii ;
- grupul de intrări analogice, pentru citirea rezultatelor convertorului analog/numeric ;
- grupul de ieșiri analogice, pentru comanda celor două convertoare numeric/analogice ;
- două grupuri de temporizare-numărare (A și B), care accesează cele două circuite 8253 din componența plăcii;
- grupul pentru intrările și ieșirile digitale, constituit pe baza circuitului 8255A de pe placă ;
- grupul regiștrilor numărătorului de intervale.

Pachetul de programe și funcții soft - NI-DAQ - care însoțește placa de achiziție este destinat controlului funcționării acesteia de către calculator. Este un pachet soft destinat întregii familii de plăci National Instruments.

Aplicațiile de utilizare a plăcilor National Instruments se pot scrie în limbaje de nivel înalt - Microsoft sau Borland C, Pascal sau Basic. Programele de aplicație pot fi rulate atât sub sistemul de operare DOS cât și din mediul WINDOWS.

Pentru achiziția semnalelor analogice în regim static este prezentată shema din fig. A1.2, în care s-au făcut notațiile:

- ACP - amplificator cu câștig programabil;
- CEM – circuit eșantionare-memorare;
- CAN – convertor analog-numeric;
- RC – regiștrii de comandă;
- SAD – sincronizare achiziție de date;
- AN FIFO – registru intrare-ieșire (first-in first-out)

Unul dintre multiplexoare are opt canale de intrări analogice (0..7). Acest prim multiplexor este folosit pentru achiziția unor semnale analogice cu referință (masă) comună.

Celălalt multiplexor este conectat la canalele 1, 3, 5, 7 pentru semnale diferențiale (fiecare semnal având propria sa referință).

Amplificatorul cu câștig programabil permite amplificarea semnalului de intrare înainte de eșantionarea și conversia sa în numeric, asigurându-se astfel o creștere a rezoluției și preciziei măsurătorii. Câștigul amplificatorului instrumental se selectează prin control software. Placa NI PCI 1200 asigură câștiguri de 1; 2; 5;10; 20; 50 și 100.

Placa NI PCI 1200 folosește un convertor analog-numeric cu aproximări succesive de 12 biți. Rezoluția de 12 biți face ca rezultatul conversiei să se obțină în 4096 de pași. Rezultatul conversiei reprezintă expresia numerică a valorii nivelului voltajului de intrare. Domeniul propriu de intrare a convertorului este de 0...5V. Circuitele adiționale măresc acest domeniu de intrare la +/- 5V sau 0...10V.

Rezultatul unei conversii analog-numerice se depune în memoria FIFO după ce în prealabil a suferit o extensie de semn la 16 biți. Memoria FIFO servește ca buffer al convertorului analog-numeric și prezintă două avantaje. În primul rând, după completarea unei conversii, valoarea numerică este salvată în memoria FIFO, în vederea unei citiri ulterioare, iar convertorul este liber pentru o nouă conversie. Pe de altă parte, memoria FIFO poate colecta rezultatele până la 512 conversii analog-numerice, înainte să se producă o pierdere de informație, permițând astfel software-ului să se alinieze cu hardware-ul. Marja de timp acordată este, desigur, de 512 ori intervalul de eșantionare.

Pentru siguranță, după fiecare 16 scrieri în memoria FIFO, se intră pe condiție de eroare - A/D FIFO overflow - dacă încă nu a avut loc nici o citire din memoria FIFO. Rezultatul conversiei analog-numerice poate fi interpretat fie în cod binar natural, pentru semnal de intrare unipolar, fie în cod complement de doi, pentru semnal de intrare bipolar. În modul unipolar rezultatul conversiei este prezentat ca un număr binar natural din domeniul 0..+4095. În mod bipolar, el este un număr în codul complement de doi din domeniul -2048..2047. În acest mod, bitul de semnificație maximă a rezultatului conversiei este inversat, în vederea obținerii complementului lui doi.

Rezultatul conversiei analog-numerice este apoi extins de la 12 la 16 biți, adică i se adaugă în față un 016 sau un F16, funcție de cod și de semn. Astfel, valorile citite din memoria FIFO au o lungime de 16 biți.

Operația de achiziție se referă la procesul de prelevare a unei secvențe de rezultate ale unor conversii analog-numerice, la intervale de timp (sample interval)

sincronizate cu grijă. Circuitele de sincronizare a achiziției generează diverse semnale de ceas și de sincronizare. Placa NI PCI 1200 poate realiza achiziții pe unul sau mai multe canale (scanat), în mod continuu sau la intervale de timp anume. Placa folosește un numărator care va realiza trecerea automată de la un canal de intrare analogică la altul.

Sincronizarea achiziției presupune semnale care inițiază operația de achiziție, inițiază conversiile analog-numeric individuale, direcționează (gate) achiziția de date și generează ceasurile (clock) de scanare. Aceste semnale sunt furnizate de către circuite de sincronizare din componența plăcii. În acest scop, placa folosește unul din cele două circuite 8253.

O conversie analog-numerică se inițiază fie printr-o tranziție 1→0 a ieșirii număratorului 0 (OUT A0) a chipului 8253(A), fie printr-o tranziție 1→0 a intrării EXTCONV. Intervalul de prelevare a eșantioanelor este de 2...65535 msec. și este furnizat de un numărator (cu decrementare) de 16 biți, cu un ceas de 1 MHz (la fiecare trecere prin 0 a număratorului se generează un impuls și se reîncarcă la valoarea programată). Software-ul trebuie să țină însă socoteala conversiilor efectuate și să inițializeze număratorul A0 după realizarea numărului dorit de conversii.

Achiziția pe mai multe canale (scanată) este controlată de un numărator de decrementare (de la numărul cel mai mare de canal la 0). Frecvența de achiziție maximă (numărul de eșantioane pe secundă) este dată de timpul de conversie plus timpul de eșantionare. În achiziția scanată acestuia i se adaugă timpul de stabilizare a multiplexoarelor de intrare și amplificatorului (după ce multiplexoarele comută, trebuie să se aștepte stabilizarea amplificatorului pe noua valoare a semnalului de intrare, înainte de a iniția conversia, pentru a obține precizia de 12 biți a rezultatului acesteia).

Timpul de stabilizare este funcție de câștigul selectat. Circuitele de sincronizare ale achiziției de date ale plăcii sesizează când frecvența de achiziție mare poate duce la pierderea rezultatelor conversiilor analog-numeric, prin setarea fanionului de eroare (overrun error flag) din registrul de stare. Se recomandă, deci, frecvențe de achiziție conform tabelului A5.2

Tabel A5.2. Frecvențe de achiziție recomandate

MOD DE ACHIZIȚIE	CÂȘTIG	FRECVENȚĂ
Mono-canal	1; 2; 5; 10; 20; 50; 100	83,3 K eșant./s; 71,4 K eșant./s
Multi-canal	1; 2; 5; 10; 20; 50; 100	83,3 K eșant/s; 55,5...62,5 K eșant/s; 20,0 K eșant/s.

Program "grabulet.c"

/*
 Programul urmareste un punct luminos (LED) care se misca dupa o lege de miscare oarecare, si calculeaza la anumite intervale de timp situarea in spatiu a punctului luminos. Eventual se poate adapta si pt. calculul vitezei si acceleratiei.

LED-ul se urmareste cu ajutorul unui senzor video CCD care este miscat de doua motorase pas cu pas, comandate de calculator, in functie de miscarea LED-ului

Programul executa urmatoarele:

- initializeaza placile de achizitii
- reseteaza axele camerei video (se aduce la pozitia "zero")
- se executa intr-o bucla infinita urmatorul meniu, pana cand se abandoneaza de catre utilizator executia meniului:
 - optiuni MENUU:
 - resetare/reglare manuala axe camera video
 - urmarire LED
 - iesire
- se reseteaza placile de achizitii si se elibereaza zonele de memorie alocate pt. executia programului

*/

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

```
#include <mil.h>
//headere functii placa de achizitii video
#include "nidaqex.h"
//headere functii placa de achizitii si comanda semnale analog/numerice
```

```
MIL_ID MilApplication, /* Application identifier. */
      MilSystem,      /* System identifier.    */
      MilDisplay,    /* Display identifier.  */
      MilDigitizer, /* Digitizer identifier.*/
      Millmage;
```

```
unsigned char *buf;
```

```
/*
```

```
pointer la o zona de memorie in care se salveaza imaginea
achizitionata de senzorul video. Imaginea este stocata liniar,
necodificat, pe 8 biti, alb-negru. Valoarea 255 inseamna albul
cel mai stralucitor, valoarea 0 este negrul absolut.
```

```
*/
```

```
unsigned char ch;
```

```
double      xcg,
            ycg;
long        sum_x,
```

```

                sum_y,
                pos_x,
                pos_y,
                nr_x,
                nr_y;
long           parcurse;
long           contrast,
                luminozitate,
                tempo;
int            PRAG=145;
/*
PRAG este pragul de luminozitate pt. sesizarea sursei
luminoase. Daca un pixel este mai luminos decat PRAG, atunci
el apartine sursei luminoase.
Teoretic poate fi stabilit in mod dinamic, practic in cazul de
fata s-a stabilit experimental, pt. a nu complica extrem de
mult programul. Valoarea poate fi modificata de catre
utilizator, in functiile de conditiile de lucru, in momentul
ajustarii camerei video si a parametrilor acesteia
*/
int            curent_xcg,
                curent_ycg;

long contor=0;
i16 iStatus = 0;
i16 iRetVal = 0;
i16 iDevice = 1;
i16 iPort = 0;
i16 iLine = 3;
    i16    iMode = 0;//no hand-shaking mode
i16 iDir = 1;//standard output port = 1
i16 iStateON = 1;
i16 iStateOFF = 0;
i16 ilgnoreWarning = 0;

unsigned char horiz_last_pulse;//poate fi 1,2,3 sau 4, unde
                                //1=2+8
                                //2=8+32
                                //3=32+128
                                //4=2+128 la actionarea MCC

unsigned char vert_last_pulse;//idem
unsigned char pulse_array[5]={0,2+8,8+32,32+128,2+128};

//prototip cautare centru greutate
void get_cg(void);

//prototip functie miscare motor
void misca_placa(void);

//prototip functie ajustare camera

```



```
void camera_setup(void);
```

```
void rotate_horizontal(int);
```

```
//prototip functie prelucrare imagine si control sistem
```

```
void step_by_step(void)
```

```
/*
```

Functia step_by_step este practic functia principala din program.

Functia executa intr-o bucla infinita, care poate fi oprita de utilizator prin apasarea tastei Esc, urmatoara succesiune de operatii:

- se achizitioneaza o imagine de la camera video, imagine care se salveaza in memorie si care poate fi accesata in mod direct.

- se parcurge imaginea liniar si se cauta sursa luminoasa. Deoarece sursa luminoasa (LED-ul) nu este punctiforma in realitate, se considera pozitia sursei luminoase pozitia centrului de greutate (CG) al petei luminoase.

- se compara pozitia actuala a CG cu pozitia CG avuta in momentul ultimei actualizari/corectari a axelor ce misca senzorul video (CG de referinta). In cazul in care diferenta pe orizontala si/sau verticala depaseste o anumita valoare, atunci se executa urmatoarele operatii:

- se calculeaza si se afiseaza pozitia (si eventual viteza/acceleratia punctului luminos)
- se comanda axele camerei video a.i. centrul imaginii ce urmeaza a fi achizitionate sa corespunda cu CG curent
- CG curent devine CG de referinta

Altfel (diferenta de deplasare mai mica), camera video ramane nemiscata si se executa doar urmatoarea operatie:

- se calculeaza si se afiseaza pozitia (si eventual viteza/acceleratia punctului luminos)

```
*/
```

```
{
```

```
int contor=0;
```

```
/* Grab . */
```

```
MdigGrab(MilDigitizer, MillImage);//achizitioneaza o imagine
```

```
contor++;
```

```
printf(" iteratia nr.:%d",contor);
```

```
//cauta CG
```

```
get_cg();

curent_xcg=(int)xcg;
curent_ycg=(int)ycg;

do {

    ch=255;

    MdigGrab(MilDigitizer, Millmage);//achizitioneaza o imagine

    contor++;
    printf(" iteratia nr.:%d",contor);

    //cauta CG
    get_cg();

    if (abs((int)(curent_xcg-xcg))>1) {
        rotate_horizontal((int)((curent_xcg-xcg)*(80/34)));
        curent_xcg=(int)xcg;//se va completa si cu deplasarea pe y
    }

    if (kbhit()) {
        ch=getch();
    }

} while (ch!=27);//se termina la Esc

ch=255;

};

void get_cg(void) {
/*
functia cauta centrul de greutate al sursei luminoase.
*/
    int    primul_x,
           primul_y,
           colt_sus_x,
           colt_sus_y,
           colt_jos_x,
           colt_jos_y;

    if (buf == NULL) {
        printf("\nbufferul de imagine nu este alocat!!!\n");
        exit(0);
    }//daca nu s-a alocat bufferul de imagine, se iese

    xcg=0;
    ycg=0;
    pos_x=-2;
```

```

pos_y=-2;
parcourse=-2;
primul_x=0;
primul_y=0;
colt_sus_x=0;
colt_sus_y=0;
colt_jos_x=0;
colt_jos_y=0;

```

```

MdigGrabWait(MilDigitizer,M_GRAB_END);//waiting for the end of the grab

```

```

//bucla cautare primul punct alb

```

```

/*

```

```

pt. optimizarea cautarii, nu se cauta linie cu linie si punct cu
punct, ci din doua in doua linii, si din doi in doi pixeli.
Experimental s-a observat ca este suficient de "des" ca sa se
gaseasca in imagine sursa luminoasa, in conditii normale de lucru

```

```

*/

```

```

do {
    pos_x+=2;
    parcourse+=2;
    if (pos_x>767) {
        pos_x=0;
        pos_y+=2;
        parcourse+=768;
    }
}
while ((* (buf+parcourse)<PRAG)&&parcourse<442368);
//pana cand avem ceva mai luminos ca PRAG sau s-a parcurs totul

```

```

//daca inca ne aflam in interiorul imaginii
if (parcourse<442368) {

```

```

    primul_x=pos_x;
    primul_y=pos_y;

```

```

/*

```

```

in momentul in care s-a gasit primul punct "alb", acesta se
incadreaza intr-un dreptunghi care va fi parcurs "pixel cu pixel"
Se tine cont la stabilirea dreptunghiului de lucru de
posibilitatea ca punctul luminos sa fie intr-un colt al imaginii.
Se calculeaza suprafata si CG-ul petei luminoase.

```

```

*/

```

```

//stabilire coordonate x ale dreptunghiului de lucru

```

```

if (primul_x > 7)
    colt_sus_x=primul_x-7;
else
    colt_sus_x=0;

```

```

if (primul_x < 151)

```

```

        colt_jos_x=primul_x+9;
    else
        colt_jos_x=159;

    //stabilire coordonate y ale dreptunghiului de lucru
    if (primul_y > 1)
        colt_sus_y=primul_y-2;
    else
        colt_sus_y=0;

    if (primul_y < 145)
        colt_jos_y=primul_y+14;
    else
        colt_jos_y=159;

/*
pt. etapa de debugging, se evidentiaza cu linii albe colturile
dreptunghiului de lucru
*/
        MgraLine(M_DEFAULT, Millmage, colt_sus_x,
colt_sus_y,colt_sus_x+5,colt_sus_y);
        MgraLine(M_DEFAULT, Millmage, colt_sus_x,
colt_sus_y,colt_sus_x,colt_sus_y+5);
        MgraLine(M_DEFAULT, Millmage, colt_jos_x, colt_jos_y,colt_jos_x-
5,colt_jos_y);
        MgraLine(M_DEFAULT, Millmage, colt_jos_x,
colt_jos_y,colt_jos_x,colt_jos_y-5);
//end linii ajutatoare
    nr_x=0;
    nr_y=0;
    sum_x=0;
    sum_y=0;

    for (pos_y=colt_sus_y;pos_y<=colt_jos_y;pos_y++) {
        for (pos_x=colt_sus_x;pos_x<=colt_jos_x;pos_x++) {
            if (*(buf+pos_y*768+pos_x)>PRAG) {
                //daca este alb
                nr_x++;
                nr_y++;
                sum_x+=pos_x;
                sum_y+=pos_y;
            }
        }
    }

    xcg=(float)(sum_x)/nr_x;
    ycg=(float)(sum_y)/nr_y;
    printf("\tX cg =%3.2f",xcg);
    printf("\tY cg =%3.2f\r",ycg);

```

```
    }
};

//functie delay
void stai(void) {

long   tmp=0;

    do {
        tmp++;
    } while (tmp<180000);
}

void misca_placa(void) {
/*
functie de initializare porturi numerice de iesire
*/

    iStatus = DIG_Out_Port(iDevice, iPort, 2+8);
    stai();
    iStatus = DIG_Out_Port(iDevice, iPort, 32+8);
    stai();
    iStatus = DIG_Out_Port(iDevice, iPort,128+32);
    stai();
    iStatus = DIG_Out_Port(iDevice, iPort,2+128);
    stai();

    iStatus = DIG_Out_Port(iDevice, iPort, 0);
//se scoate de aici si se va pune dupa fiecare apel al functiei

}

/*
functie de rotire a motorului pas cu pas ce efectueaza miscarea
in plan orizontal a camerei video. Se specifica numarul de pasi
doriti ca si argument al functiei, sensul de miscare fiind dat de
semnul argumentului, pozitiv sau negativ.
*/
void rotate_horizontal(int rotations) {
    int contor_rotatii=rotations;

    if (contor_rotatii>0) {
        do {
            if (horiz_last_pulse==4) {
                horiz_last_pulse=1;
            }
            else {
                horiz_last_pulse++;
            }
            contor_rotatii--;
            iStatus = DIG_Out_Port(iDevice,0,pulse_array[horiz_last_pulse]);
        }
    }
}
```

```

        stai();
    } while (contor_rotatii>0);
}
if (contor_rotatii<0) {
    do {
        if (horiz_last_pulse==1) {
            horiz_last_pulse=4;
        }
        else {
            horiz_last_pulse--;
        }
        iStatus = DIG_Out_Port(iDevice,0,pulse_array[horiz_last_pulse]);
        stai();
        contor_rotatii++;
    } while (contor_rotatii<0);
}

iStatus = DIG_Out_Port(iDevice,0,pulse_array[0]);

}

```

```
void camera_setup(void) {
```

```
/*
```

functie de ajustare a camerei video. Functia achizitioneaza in timp real imagini pe care le afiseaza pe monitor, oferind posibilitatea efectuării urmatoarelor reglaje:

- reglare contrast imagine
- reglare luminozitate imagine
- miscare axa orizontala camera video
- miscare axa verticala camera video
- modificare prag de selectie a sursei luminoase
- iesire

Toate modificarile efectuate se reflecta in timp real in imaginile achizitionate.

```
*/
```

```
MdigGrabContinuous(MilDigitizer,MillImage);
```

```
do {
```

```

    //afisare optiuni
    printf("\n\nAjustare camera\n");
    printf("\n\toptiuni:\n");
    printf("1 Contrast\n");
    printf("2 Luminozitate\n");
    printf("3 Prag binarizare\n");
    printf("4 axa Orizontala camera\n");
    printf("5 axa Verticala camera\n");
    printf("6 Esc - iesire\n");
    printf("\n\n");

```

```

    /*selectarea unei optiuni din meniu*/
    ch=getch();

```

```

//executie optiuni din meniu

//contrast
if ((ch=='1')||(ch=='c')||(ch=='C')) {
    printf("\nSetare contrast\n");
    tempo=contrast;

    do {
        printf("\rcontrastul:%d \tsageți-modifica Enter-setează Esc-
abandon",tempo);
        ch=getch();
        if (ch==224) {
            ch=getch();
            if ((ch==72)||(ch==77)) { //sageata sus sau la dreapta
                if (tempo<M_MAX_LEVEL) {
                    tempo++;
                    MdigReference(MilDigitizer,
M_CONTRAST_REF,tempo);
                }
            }

            if ((ch==80)||(ch==75)) { //sageata jos sau la stanga
                if (tempo>M_MIN_LEVEL) {
                    tempo--;
                    MdigReference(MilDigitizer,
M_CONTRAST_REF,tempo);
                }
            }
        }
    }

} while ((ch!=27)&&(ch!=13));

if (ch==13) {
    contrast=tempo;
}
else {
    MdigReference(MilDigitizer, M_CONTRAST_REF,contrast);
}

ch='1';
}

//luminozitate
if ((ch=='2')||(ch=='l')||(ch=='L')) {

    printf("\nSetare luminozitate\n");
    tempo=luminozitate;

    do {

```

```

printf("\rluminozitate:%d \tsageți-modifica Enter-seteaza Esc-
abandon",tempo);
ch=getch();
if (ch==224) {
    ch=getch();
    if ((ch==72)||(ch==77)) { //sageata sus sau la dreapta
        if (tempo<M_MAX_LEVEL) {
            tempo++;
            MdigReference(MilDigitizer,
M_BRIGHTNESS_REF,tempo);
        }
    }
    if ((ch==80)||(ch==75)) { //sageata jos sau la stanga
        if (tempo>M_MIN_LEVEL) {
            tempo--;
            MdigReference(MilDigitizer,
M_BRIGHTNESS_REF,tempo);
        }
    }
}
} while ((ch!=27)&&(ch!=13));

if (ch==13) {
    contrast=tempo;
}
else {
    MdigReference(MilDigitizer,
M_BRIGHTNESS_REF,luminozitate);
}

ch='1';
}

//prag binarizare
if ((ch=='3')||(ch=='p')||(ch=='P')) {

    printf("\nSetare prag binarizare\n");
    tempo=PRAG;

    do {
        printf("\rprag bin.:%d \tsageți-modifica Enter-seteaza Esc-
abandon",tempo);
        ch=getch();
        if (ch==224) {
            ch=getch();
            if ((ch==72)||(ch==77)) { //sageata sus sau la dreapta
                if (tempo<M_MAX_LEVEL) {
                    tempo++;

```



```

        }
    }

    if ((ch==80)||(ch==75)) { //sageata jos sau la stanga
        if (tempo>M_MIN_LEVEL) {
            tempo--;
        }
    }

}

} while ((ch!=27)&&(ch!=13));

if (ch==13) {
    PRAG=tempo;
}

ch='1';

}

//axa orizontala CCD
if ((ch=='4')||(ch=='o')||(ch=='O')) {

    printf("\nSetare axa orizontala camera\n");

    do {
        ch=getch();
        if (ch==224) {
            ch=getch();
            if ((ch==72)||(ch==77)) { //sageata sus sau la dreapta
                rotate_horizontal(1);
            }

            if ((ch==80)||(ch==75)) { //sageata jos sau la stanga
                rotate_horizontal(-1);
            }
        }
    }

} while ((ch!=27)&&(ch!=13));

ch='1';

}

//axa verticala CCD
if ((ch=='5')||(ch=='v')||(ch=='V')) {
    printf("\nSetare axa verticala camera\n");
}

} while ((ch!=27)&&(ch!='6'));
```

```

        ch=255;
        MdigHalt(MilDigitizer);
    }

void main(void) {
/*
functia principala
*/
    iStatus = DIG_Prt_Config (iDevice,iPort,iMode,iDir);
    iRetVal = NIDAQErrorHandler(iStatus, "DIG_Line_Config",
ilgnoreWarning);

    /* Allocate defaults. */

    MappAlloc(M_DEFAULT, &MilApplication);
    MsysAlloc(M_DEF_SYSTEM_TYPE, M_DEF_SYSTEM_NUM, M_SETUP,
&MilSystem);
    MdigAlloc(MilSystem, M_DEFAULT, "M_CCIR" , M_DEFAULT, &MilDigitizer);
    MdigChannel(MilDigitizer , M_CH0);
    MdispAlloc(MilSystem, M_DEFAULT, M_DEF_DISPLAY_FORMAT, M_DEFAULT,
&MilDisplay);
    MbufAlloc2d(MilSystem,
                (long)(MdigInquire(MilDigitizer, M_SIZE_X, M_NULL)),
                (long)(MdigInquire(MilDigitizer, M_SIZE_Y, M_NULL)),
                8L+M_UNSIGNED,
                M_IMAGE+M_GRAB+M_PROC+M_DISP, &MillImage);

    /*display activation*/

    MbufClear(MillImage, 0x0);
    MdispSelect(MilDisplay, MillImage);

    /*setting the image size and offset*/

    MdigControl(MilDigitizer,M_SOURCE_OFFSET_X,304);
    MdigControl(MilDigitizer,M_SOURCE_OFFSET_Y,208);

    MdigControl(MilDigitizer,M_SOURCE_SIZE_X,160);
    MdigControl(MilDigitizer,M_SOURCE_SIZE_Y,160);

    contrast=(long)((M_MIN_LEVEL+M_MAX_LEVEL)/2);
    luminozitate=(long)((M_MIN_LEVEL+M_MAX_LEVEL)/2);
    MdigReference(MilDigitizer, M_BRIGHTNESS_REF,luminozitate);
    MdigReference(MilDigitizer, M_CONTRAST_REF,contrast);

    /*getting the image buffer address*/

    buf = (unsigned char *)MbufInquire(MillImage,M_HOST_ADDRESS,M_NULL);

    misca_placa();//pt. a ne asigura ca suntem cu motorul in pozitia de start

```

```
horiz_last_pulse=4;

// meniu principal
do {
    printf("\n\toptiuni:\n");
    printf("1 Ajustare camera si axe CCD\n");
    printf("2 Urmare robot\n");
    printf("3 Iesire (Esc)\n\n\n");

    ch=getch();

    if ((ch=='1')||(ch=='a')||(ch=='A')) {
        camera_setup();
    }

    if ((ch=='2')||(ch=='u')||(ch=='U')) {
        printf("Urmare robot - Esc pt. iesire\n\n");
        step_by_step();
    }

} while ((ch!=27)&&(ch!='3')&&(ch!='i')&&(ch!='I'));

printf("\nLa revedere!!!\n");

//resetare port digital de iesire
iStatus = DIG_Out_Port(iDevice, iPort, 0);

/* Release defaults. */

MbufFree(MilImage);
MdispFree(MilDisplay);
MdigFree(MilDigitizer);
MsysFree(MilSystem);
MappFree(MilApplication);
}

/* End of program */
```

**Valorile timpilor obținuți în decursul conducerii globale a sistemului robot pentru
ciclul prezentat în figura 8.38**

Generare semnal mcc	Achiziție imagine	Calculul c.g. al țintei	Generare semnal mpp
0.034	0.105	0.038	0
0.0341	0.105	0.038	0
0.034	0.105	0.038	0
0.0338	0.105	0.038	0
0.0341	0.105	0.039	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.0338	0.105	0.038	0
0.034	0.105	0.038	0
0.0341	0.105	0.038	0
0.034	0.105	0.038	0.045
0.034	0.105	0.038	0
0.0339	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.0341	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.039	0
0.0339	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.031	0.110	0.038	0
0.03	0.110	0.039	0
0.03	0.110	0.038	0
0.032	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0

0.031	0.110	0.039	0.045
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.038	0
0.032	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.031	0.110	0.039	0
0.029	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.029	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.031	0.108	0.039	0
0.045	0.108	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.032	0.108	0.039	0
0.03	0.108	0.039	0,045
0.031	0.108	0.039	0
0.03	0.108	0.039	0
0.029	0.108	0.039	0
0.03	0.110	0.038	0
0.03	0.108	0.039	0
0.032	0.108	0.039	0
0.03	0.108	0.039	0
0.031	0.108	0.039	0
0.03	0.108	0.039	0
0.027	0.108	0.038	0
0.03	0.108	0.039	0
0.03	0.108	0.038	0
0.029	0.108	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.038	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.034	0.105	0.038	0
0.0341	0.105	0.038	0
0.034	0.105	0.038	0

0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.031	0.110	0.038	0
0.03	0.110	0.039	0
0.03	0.110	0.038	0
0.032	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.031	0.110	0.039	0.045
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.038	0
0.032	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.031	0.110	0.039	0
0.029	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.029	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.110	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.031	0.108	0.039	0
0.045	0.108	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.032	0.108	0.039	0
0.03	0.108	0.039	0.045
0.031	0.108	0.039	0
0.03	0.108	0.039	0
0.029	0.108	0.039	0
0.03	0.110	0.038	0

0.03	0.108	0.039	0
0.032	0.108	0.039	0
0.03	0.108	0.039	0
0.031	0.108	0.039	0
0.03	0.108	0.039	0
0.027	0.108	0.038	0
0.03	0.108	0.039	0
0.03	0.108	0.038	0
0.029	0.108	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.038	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.03	0.108	0.039	0
0.034	0.105	0.038	0
0.0341	0.105	0.038	0
0.034	0.105	0.038	0
0.0338	0.105	0.038	0
0.0341	0.105	0.039	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.0338	0.105	0.038	0
0.034	0.105	0.038	0
0.0341	0.105	0.038	0
0.034	0.105	0.038	0.045
0.034	0.105	0.038	0
0.0339	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.0341	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.039	0
0.0339	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0
0.034	0.105	0.038	0

Program "ROBOT_VISION.c"

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <ctype.h>
#include <string.h>
#include <alloc.h>
#include <process.h>
#include <dos.h>

#define BLACK 0;
#define WHITE 255;
#define PERIMETER 1;
#define True 1
#define False 0
#define bool int
#define POINTER void*
#define PPOINTER void **

#include <stdio.h>
#include <iplib.h>
#include <errno.h>
#include <math.h>
#include "exam.h"
#include USEMACRO

#include "stiva.c"
#include "default.c"
#include "obiecte.h"
#include "image1.h"
#include "robot.h"

void main(void)
{
    extern int errno;
    signed char rez;
    unsigned long int err_code;
    PUNCT *pct2;
    char ch;
    extern unsigned char huge imagine[480][512];
    extern unsigned long sup_crt;
    extern long xcg_crt;
    extern long ycg_crt;
    char c1[7];
    char c2[7];
    char c3[7];
    char sir_temp[10];
    float ftmp;
```

```

long xr,yr,tmp;
int nr_obiecte=0;
clrscr();
err_code=VG_InitLib();
init(&err_code);

//initializare robot
// spawnl(P_WAIT,"test.exe","test.exe","go",0);
// delay(500);
//pozitia 1
// spawnl(P_WAIT,"test.exe","test.exe","0.0","275.0","300.0","-90.0","175.0",0);
//pozitia 2
// spawnl(P_WAIT,"test.exe","test.exe","-185.0","-200.0","300.0","-90.0"," 175.0",0);
//pozitia 3
// spawnl(P_WAIT,"test.exe","test.exe","-250.0"," -248.0"," 280.0"," -83.0"," 176.0",0);

defaultsettings();
StergeEcran(0);
/* VG_SetContGrabSt(ON);
ch=getch();
VG_SetContGrabSt(OFF);*/
VG_SingleGrab();
while(VG_GetGrabSt());
VG_ReadBlock(0,0,512,480,imagine);
/* err_code=VG_SavePicture("unu.tga",0,0,512,480,BIT_MAP);*/
/* VG_LoadPicture("unu.tga");*/
binarizare(170);
// StergeEcran(0);
// VG_WriteBlock(0,0,512,480,imagine);
// VG_DilateImage(); //functia dilata elementele WHITE, NU BLACK
// VG_ErodeImage();
delay(4000);
pct2=(PPUNCT) malloc(sizeof(PUNCT));
if (FirstObject(pct2))
    {
    printf("\a");
    }
else
    printf("nu s-au gasit obiecte\n");
while (FirstObject(pct2))
    {
    nr_obiecte++;
    printf("obiectul cu numarul:%d",nr_obiecte);
    FillFirstObject(*(pct2));
    tmp=319-ycg_crt;
    ftmp=(tmp*205)/480;
    tmp=(long)ftmp;
    xr=tmp-385;
    tmp=xcg_crt-265;
    yr=tmp*276/512;
    ltoa(xr,c1,10);
    ltoa(yr,c2,10);

```

```

tmp=90-alfa;
ltoa(tmp,c3,10);
if ((sup_crt>500) && (sup_crt<1100))/*piulita*/
{
//deplasare deasupra CG
spawnl(P_WAIT,"test.exe","test.exe",c1,c2," 280.0"," -93.0",c3,0);
delay(500);
//coboara
spawnl(P_WAIT,"test.exe","test.exe",c1,c2," 250.0"," -93.0",c3,0);
delay(500);
//apuca
spawnl(P_WAIT,"test.exe","test.exe","gc",0);
delay(500);
//ridica
spawnl(P_WAIT,"test.exe","test.exe",c1,c2," 280.0"," -93.0",c3,0);
delay(500);
//se duce
spawnl(P_WAIT,"test.exe","test.exe","-180.0"," -280.0"," 270.0","-
90.0","0.0",0);
delay(500);
//coboara
spawnl(P_WAIT,"test.exe","test.exe","-180.0"," -280.0"," 250.0","-
90.0","0.0",0);
delay(500);
//arunca
spawnl(P_WAIT,"test.exe","test.exe","go",0);
//se ridica
spawnl(P_WAIT,"test.exe","test.exe","-180.0"," -280.0"," 270.0","-
90.0","0.0",0);
delay(500);
//
ch=getch();
}
if ((sup_crt>1100) && (sup_crt<2000)) /*arc*/
{
//deplasare deasupra CG
spawnl(P_WAIT,"test.exe","test.exe",c1,c2," 280.0"," -93.0",c3,0);
delay(500);
//coboara
spawnl(P_WAIT,"test.exe","test.exe",c1,c2," 250.0"," -93.0",c3,0);
delay(500);
//apuca
spawnl(P_WAIT,"test.exe","test.exe","gc",0);
delay(500);
//ridica
spawnl(P_WAIT,"test.exe","test.exe",c1,c2," 280.0"," -93.0",c3,0);
delay(500);
//se duce
spawnl(P_WAIT,"test.exe","test.exe","-210.0"," -270.0"," 270.0","-
90.0","0.0",0);
delay(500);
//coboara

```

```

spawnl(P_WAIT,"test.exe","test.exe",-210.0," -270.0"," 250.0",-
90.0","0.0",0);
    delay(500);
    //arunca
    spawnl(P_WAIT,"test.exe","test.exe","go",0);
    //se ridica
    spawnl(P_WAIT,"test.exe","test.exe",-210.0," -270.0"," 270.0",-
90.0","0.0",0);
    delay(500);
//    ch=getch();
    }
    if (sup_crt>2000) /*surub*/
    {
        //deplasare deasupra CG
        spawnl(P_WAIT,"test.exe","test.exe",c1,c2," 280.0"," -93.0",c3,0);
        delay(500);
        //coboara
        spawnl(P_WAIT,"test.exe","test.exe",c1,c2," 250.0"," -93.0",c3,0);
        delay(500);
        //apuca
        spawnl(P_WAIT,"test.exe","test.exe","gc",0);
        delay(500);
        //ridica
        spawnl(P_WAIT,"test.exe","test.exe",c1,c2," 280.0"," -93.0",c3,0);
        delay(500);
        //se duce
        spawnl(P_WAIT,"test.exe","test.exe",-250.0," -248.0"," 270.0",-
90.0","0.0",0);
        delay(500);
        //coboara
        spawnl(P_WAIT,"test.exe","test.exe",-250.0," -248.0"," 250.0",-
90.0","0.0",0);
        delay(500);
        //arunca
        spawnl(P_WAIT,"test.exe","test.exe","go",0);
        //se ridica
        spawnl(P_WAIT,"test.exe","test.exe",-250.0," -248.0"," 270.0",-
90.0","0.0",0);
        delay(500);
//    ch=getch();
    }
}

/* //apelare program extern pentru comanda robot
    rez=spawnl(P_WAIT,"test.exe",
                "test.exe",
                "go ",0);

    //mesaje eroare
    if (rez)
        switch (errno)
        {

```

```

        case E2BIG :printf("Arg list too long\n");
                    break;
        case EINVAL:printf("Invalid argument\n");
                    break;
        case ENOENT:printf("Path or file name not found\n");
                    break;
        case ENOEXEC:printf("Exec format error\n");
                    break;
        case ENOMEM :printf("Not enough core\n");
                    break;
    }*/
printf("\a");
ch=getch();

//inchidere biblioteca functii placa MATROX
VG_QuitLib();
free(pct2);
printf("\a");
}

```

```
#include "stiva.h"
```

```
bool InitalizareStiva(PSTIVA p)
```

```

{
    p->VirfulStivei=NULL;
    return True;
}

```

```
bool TestStivaGoala(PSTIVA p)
```

```

{
    return (p->VirfulStivei == NULL);
}

```

```
bool IntroducereInStiva(PSTIVA ps, POINTER e)
```

```

{
    PELEMENT p;
    p = (PELEMENT) malloc(sizeof(ELEMENT));
    if (!p)
        return False;
    else
    {
        /* completare inregistrare */
        p->valoare=e;
        p->precedentul=ps->VirfulStivei;
        ps->VirfulStivei=p;
        return True;
    }
}

```

```
bool ExtrageDinStiva(PSTIVA p, PPOINTER e)
```

```

{
    PELEMENT t;

```

```

    if (TestStivaGoala(p))
        return False;
    else
        {
            *e=p->VirfulStivei->valoare;
            t=p->VirfulStivei->precedentul;
            free(p->VirfulStivei);
            p->VirfulStivei=t;
            return True;
        }
}

```

```

bool ValoareDinVirf(PSTIVA p, PPOINTER e)
{
    if (TestStivaGoala(p))
        return False;
    else
        {
            *e=p->VirfulStivei->valoare;
            return True;
        }
}

```

/* se va folosi o constructie de felul:

```

typedef struct
{
    int x;
    int y;
} PUNCT,*PPUNCT;

STIVA variabila_stiva;

PPUNCT pp;

pp=malloc(sizeof(PUNCT));
if (pp)
    {
        pp->x=9;
        pp->y=10;
        if (IntroduceInStiva(&variabila_stiva,(POINTER)pp))
            totul este OK;
        else
            nu mai este loc!
    }
else
    este problema mare!;
*/

```

```
#include "functii1.h"
```

```
void binarizare(unsigned char arg_prag,unsigned char * arg_imagine,int dim)
```

```
/*
```

```
- arg_prag este pragul de binarizare.Valorile mai mari  
ca pragul de binarizare sunt convertite in WHITE (255),  
cele mai mici in BLACK (0).
```

```
- arg_imagine - pointer la o imagine (un vector de elem.  
pe 8 biti
```

```
- dim - dimensiunea vectorului
```

```
*/
```

```
    {
    int contor;

    contor=0;

    while (contor<dim)//pina la poiterul nul(sfirsit)
        {
            if ( *(arg_imagine+contor) < arg_prag )
                *(arg_imagine+contor) = 0;//negru
            else
                *(arg_imagine+contor) = 255;//alb
            contor++;
        }
    }
```

```
void dilate(unsigned char* arg_linie,int arg_dim)
```

```
/*
```

```
functia umple spatiile goale cu latimea de un pixel,marginite  
de pixeli WHITE,pe o lungime de arg_dim.
```

```
se poate optimiza pt. viteza.
```

```
*/
```

```
    {
    int cd;
    for(cd=1;cd<arg_dim;cd++)
        {
            if( *(arg_linie+cd-1) == *(arg_linie+cd+1) )//daca vecinii sunt egali
                if (*(arg_linie+cd+1)==255)
                    *(arg_linie+cd)=*(arg_linie+cd+1);
        }
    }
```

```
int pozitie_centru(unsigned char * arg_linie, int arg_poz)
```

```
/*
```

```
- returneaza centrul unui sir de elemente WHITE
```

```
- arg_linie este un tablou binarizat cu functia binarizare
```

```
- arg_poz este pozitia de unde se cauta in stinga si in  
dreapta.De obicei este vechiul centru.
```

```
*/
```

```
{
```

```

long int suma;
int c,nr_el;
nr_el=0;
suma=0;
c=0;

while ( *(arg_linie+arg_poz+c) == 255 )
    {
    suma+=arg_poz+c;
    c++;
    nr_el++;
    }
c=1;
while ( *(arg_linie+arg_poz-c) == 255 )
    {
    suma+=arg_poz-c;
    c++;
    nr_el++;
    }
if (suma!=0)
    return (int)(suma/nr_el);
else
    return -1;//??
}

```

```

int suma(unsigned char*arg_linie,int centru,unsigned char lung)
/*

```

returneaza suma elementelor WHITE pe o lungime 2*lung+1
in cadrul vectorului arg_linie.Centrul intervalului este in
centru.

```

*/
{
int c,s;
s=0;
for(c=centru-lung; c<=centru+lung; c++)
    if (*(arg_linie+c) == 255)
        s++;
return s;
}

```

```

int suma_conexa(const unsigned char *arg_linie,const int arg_centru)
/*

```

functia returneaza suma pixelilor conecsi de culoare
WHITE (255).Cautarea incepe din pozitia arg_centru.

```

*/
{
int c,suma;

c=0;
suma=0;//initializari

while (*(arg_linie+arg_centru+c)==255)

```



```
        {
            suma++;
            c++;
        }
    c=-1;
    while (*(arg_linie+arg_centru-c)==255)
        {
            suma++;
            c++;
        }
    return suma;
}
```

```
typedef struct tip_punct
/*
structura care defineste tipul PUNCT
PPUNCT = pointer la tipul PUNCT
*/
{
    int x,y;
}PUNCT,*PPUNCT;
```

```
unsigned char huge imagine[480][512];
```

```
void binarizare(const unsigned char arg_prag)
/*
Functia realizeaza binarizarea matricii imagine.
Toate elementele cu culoarea mai mica sau egala cu arg_prag
sunt setate la culoarea alb (WHITE).
Toate elementele cu culoarea mai mare sunt setate
la culoarea negru (BLACK).
*/
```

```
    {
        int x,y;
        char ch;

        x=0;
        y=0;
        while(y<481)
            {
                if (imagine[y][x]<=arg_prag)
                    {
                        imagine[y][x]=BLACK;
                    }
                else
                    imagine[y][x]=WHITE;
                x++;
                if(x==513)
                    {
```

```
        x=0;
        y++;
    }
};

void StergeEcran(const unsigned char arg_culoare)
/*
functia sterge ecranul cu culoarea arg_culoare
*/
{
    PIXEL_8 p;

    p.gray=arg_culoare;
    VG_ClearScreen(&p);
};

void putp(const unsigned int argx,const unsigned int argy,
          const unsigned char arg_culoare)
{
    imagine[argy][argx]=arg_culoare;
};

unsigned char getp(const unsigned int argx,const unsigned int argy)
{
    if (argx>511)
        return WHITE;
    if (argy<1)
        return WHITE;
    if (argy>479)
        return WHITE;
    return (unsigned char) imagine[argy][argx];
};

bool FirstObject(PPUNCT arg_ppunct)
/*
Functia identifica primul obiect (pixeli de culoare BLACK)
Returneaza True daca a gasit un obiect,altfel False.
Argumentul arg_ppunct va contine coordonatele primului pixel.
*/
{
    int pozx,pozy;

    pozx=0;
    pozy=479;
    while (True)
    {
        if (getp(pozx,pozy)==0)
        {
            arg_ppunct->x=pozx;
            arg_ppunct->y=pozy;
        }
    }
}
```

```

        return True;
    }
    pozx++;
    if (pozx==512)
    {
        pozx=0;
        pozy--;
    }
    if (pozy==0)
        break;
}
printf("x=%d y=%d\n",pozx,pozy);
return False;
};

```

```
void FillFirstObject(void)
```

```

{
    int x,y,dx;
    unsigned long n;
    STIVA st1;
    PUNCT pct1;
    PPUNCT ppct;
    PIXEL_8 cul;
    char ch;

    cul.gray=200;
    n=0;

    ppct=(PPUNCT) malloc(sizeof(PUNCT));
    VG_SetPenColor(&cul);
    VG_SetPenWidth(0);
    InitializareStiva(&st1);
    FirstObject(&pct1);
    x=pct1.x;
    y=pct1.y;
    if (ppct==NULL)
    {
        printf("\nmemorie insuficienta\n");
        VG_QuitLib();
        exit(-1);
    }
    ppct->x=pct1.x;
    ppct->y=pct1.y;
    IntroducereInStiva(&st1,(POINTER) ppct);
    while (!TestStivaGoala(&st1))//cit timp mai exista in stiva
    {
        dx=0;
        ExtrageDinStiva(&st1,(PPOINTER) &ppct);
        x=ppct->x;
        y=ppct->y;
        free(ppct);
        while (getp(x+1,y)==0)

```

```

        x++; //se avanseaza pina la capatul din dreapta al liniei
while (getp(x-dx,y)==0)
    {
        n++;
        dx++;
    }
VG_MoveTo(x,y);
VG_LineRel(-dx+1,0);
dx=0;
while (getp(x-dx,y)==0) //cit timp pixelul curent este BLACK
    {
        if (dx==0) //se introduc in stiva punctele de deasupra si de
            {
                //dedesubt, daca sunt negre
                if (getp(x,y+1)==0)
                    {
                        ppct=(PPOINTER) malloc(sizeof(PUNCT));
                        ppct->x=x;
                        ppct->y=y+1;
                        if (!IntroduceInStiva(&st1,(POINTER) ppct))
                            {
                                free(ppct);
                                printf("memorie insuficienta\ala");
                                VG_QuitLib();
                                exit(-1);
                            }
                    }
                if (getp(x,y-1)==0)
                    {
                        ppct=(PPOINTER) malloc(sizeof(PUNCT));
                        ppct->x=x;
                        ppct->y=y-1;
                        if (!IntroduceInStiva(&st1,(POINTER) ppct))
                            {
                                free(ppct);
                                printf("memorie insuficienta\ala");
                                VG_QuitLib();
                                exit(-1);
                            }
                    }
            }
        if ( (getp(x-dx,y+1)==0) && (getp(x-dx+1,y+1)==255) && (dx!=0) )
            {
                /*
                daca deasupra pixelului curent exista un pixel negru
                marginit la stinga de un pixel alb, acesta se introduce
                in stiva
                */
                ppct=(PPUNCT) malloc(sizeof(PUNCT));
                ppct->x=x-dx;
                ppct->y=y+1;
                if (!IntroduceInStiva(&st1,(POINTER) ppct))
                    {

```

```
        free(ppct);
        printf("memorie insuficienta");
        VG_QuitLib();
        exit(-1);
    }
}
if ( (getp(x-dx,y-1)==0) && (getp(x-dx+1,y-1)==255) && (dx!=0) )
{
/*
acelasi lucru pt. pixelii de dedesubt
*/
ppct=(PPUNCT) malloc(sizeof(PUNCT));
ppct->x=x-dx;
ppct->y=y-1;
if (!IntroduceInStiva(&st1,(POINTER) ppct))
{
    free(ppct);
    printf("memorie insuficienta");
    VG_QuitLib();
    exit(-1);
}
}
//blocul de prelucrare a pixelilor din "interior"
putp(x-dx,y,150);
dx++;
}
}
printf("\nn=%d\n",n);
};
```