

**UNIVERSITATEA "POLITEHNICA" TIMIȘOARA  
FACULTATEA DE ELECTRONICĂ ȘI TELECOMUNICAȚII**

Ing. DAN-ANTON ANDREICIUC

**OPTIMIZAREA CONDUCERII  
VEHICULELOR GHIDATE  
AUTOMAT**

TEZĂ DE DOCTORAT

BIBLIOTECA CENTRALĂ  
UNIVERSITATEA "POLITEHNICA"  
TIMIȘOARA

624.126  
181 C

CONDUCĂTOR ȘTIINȚIFIC: Prof. dr. ing. TIBERIU MUREȘAN

-1998 -



## CUPRINS

<b>Cuprins</b>	C.1
<b>Introducere</b>	I.1
<b>1. Sisteme de transport automat. Vehicule ghidate automat.</b>	1.1.
1.1. Descrierea elementelor constructive ale VGA.	1.5
1.1.1. Sistemul mecanic.	1.5.
1.1.1.1. Şasiul.	1.5.
1.1.1.2. Sistemul de rulare.	1.5.
1.1.1.3. Dispozitive de încărcare-descărcare a sarcinii transportate.	1.8.
1.1.1.4. Sistemul mecanic de protecție.	1.9.
1.1.2. Echipamentul electric al robocarului.	1.9.
1.1.2.1. Sursele de alimentare cu energie electrică.	1.10.
1.1.2.2. Motoarele electrice.	1.11.
1.1.3. Echipamentele electronice ale unui VGA.	1.11.
1.1.3.1. Elemente de execuție.	1.11.
1.1.3.2. Traductoarele.	1.12.
1.1.3.3. Senzori utilizați pentru detectarea obstacolelor.	1.12.
1.1.3.4. Echipamente utilizate pentru determinarea situației.	1.12.
1.1.3.5. Echipamente pentru transmiterea datelor.	1.16.
1.1.3.6. Elemente de comandă și afișare.	1.17.
1.1.4. Sistemul de conducere.	1.18.
1.2. Infrastructura sistemului de transport automat.	1.20.
1.2.1. Calea de rulare.	1.20.
1.2.2. Porți de trecere.	1.21.
1.2.3. Lifturi.	1.21.
1.3. Echipamente și instalații periferice.	1.22.
1.3.1. Echipamente staționare pentru transfer de sarcină.	1.22.
1.3.2. Echipamente pentru transmiterea datelor.	1.22.
1.3.3. Instalații de siguranță.	1.22.
1.3.4. Marcaje, reflectori.	1.23.
1.3.5. Stații și instalații pentru încărcarea acumulatorilor.	1.23.
1.4. Concluzii.	1.23.
<b>2. Conducerea vehiculelor ghidate automat.</b>	2.1.
2.1. Considerații generale privind hipersistemul CIM.	2.1.

2.2. Integrarea sistemului de robocare într-un hipersistem CIM.	2.4.
2.3. Conducerea vehiculelor ghidate automat.	2.6.
2.4. Funcțiile îndeplinite de către sistemul de conducere al VGA.	2.11.
2.5. Navigația - parte integrantă a procesului de conducere a robocarului.	2.13.
2.5.1. Procedee de navigare.	2.15.
2.5.2. Cerințe impuse procedeelelor de navigare.	2.17.
2.6. Navigația vehiculelor ghidate automat pe baza hărții spațiului de lucru.	2.19.
2.7. Concluzii.	2.21.
<b>3. Determinarea situației relative a robocarelor.</b>	3.1.
3.1. Determinarea situației relative a VGA prin utilizarea odometriei.	3.2.
3.2. Program pentru realizarea unor deplasări bazate pe odometrie.	3.3.
3.2.1. Generalități referitoare la programul propus.	3.4.
3.2.2. Deplasarea prin comandă manuală.	3.9.
3.2.3. Deplasarea pe traiectorie impusă.	3.13.
3.2.4. Învățarea unei traiectorii.	3.20.
3.2.5. Deplasarea pe o traiectorie învățată.	3.29.
3.3. Concluzii.	3.32.
<b>4. Contribuții aduse metodelor utilizate pentru determinarea situației absolute a robocarelor.</b>	4.1.
4.1. Principiul referențierii discrete.	4.2.
4.2. Metode utilizate pentru determinarea situației absolute a vehiculelor ghidate automat.	4.7.
4.2.1. Determinarea situației vehiculelor cu ajutorul senzorilor ultrasonici.	4.7.
4.2.1.1. Senzori ultrasonici și panouri.	4.8.
4.2.1.2. Determinarea situației robocarului prin utilizarea a doi emițători și doi receptori ultrasonici.	4.9.
4.2.2. Determinarea situației cu ajutorul fasciculelor laser	4.11.
4.2.2.1. Scaner laser cu un reflector.	4.11.
4.2.2.2. Scaner laser cu trei reflectori.	4.13.
4.2.3. Determinarea situației vehiculului cu ajutorul camerei video.	4.15.
4.2.3.1. Camera video și marcajele pe pardoseală.	4.15.
4.2.4. Alte metode și tipuri de senzori utilizați pentru determinarea situației.	4.16.
4.2.4.1. Senzor pentru măsurarea lungimilor și	4.17.



---

marcaje pe pardoseală.	
4.2.4.2. Receptor cu senzor PSD și scanner laser.	4.18.
4.2.5. Determinarea situării unui robocar în spațiu prin vizarea a patru puncte țintă.	4.20.
4.2.5.1. Principiul metodei.	4.20.
4.2.5.2. Determinarea situării robocarului.	4.23.
4.2.6. Determinarea situării unui robocar în plan, prin vizarea a două puncte țintă.	4.27.
4.2.6.1. Principiul metodei.	4.27.
4.2.6.2. Determinarea situării robocarului.	4.29.
4.2.7. Determinarea situării unui robocar aflat în plan, prin vizarea a două puncte țintă.	4.31.
4.2.7.1. Principiul metodei.	4.31.
4.2.7.2. Determinarea situării robocarului.	4.32.
4.2.8. Determinarea situării unui robocar cu cameră video staționară.	4.34.
4.2.8.1. Principiul metodei.	4.34.
4.2.8.2. Determinarea situării vehiculului.	4.35.
4.2.9. Determinarea situării unui robocar cu cameră video amplasată pe vehicul.	4.36.
4.2.9.1. Principiul metodei.	4.36.
4.2.9.2. Determinarea situării vehiculului.	4.36.
4.2.9.3. Determinarea situării cu camera video orientabilă, amplasată pe robocar.	4.38.
4.2.9.4. Determinarea situării cu camera video fixă amplasată pe robocar.	4.39.
4.2.10. Metodă de calibrare a camerei CCD.	4.40.
4.2.11. Procedeu pentru măsurarea unghiurilor cu camera video.	4.42.
4.2.12. Rutină pentru măsurarea distanței în pixeli, între două puncte, într-o imagine digitală.	4.44.
4.2.12.1. Achiziția imaginii.	4.45.
4.2.12.2. Măsurarea distanței.	4.47.
4.2.12.3. Programul utilizat pentru măsurarea distanței dintre două puncte.	4.48.
4.3. Concluzii.	4.52.
<b>5. Concepția și realizarea sistemului mecanic al robocarului.</b>	<b>5.1.</b>
5.1. Șasiul.	5.3.
5.2. Sistemul de rulare.	5.4.
5.2.1. Mecanismul de tracțiune.	5.5.
5.2.2. Mecanismul de virare.	5.7.
5.2.3. Calculul mecanismului de virare.	5.9.

5.2.3.1. Calculul unghiurilor de virare.	5.9.
5.2.3.2. Sinteza mecanismelor de virare prin metoda analitică.	5.10.
5.2.3.3. Observații.	5.13.
5.2.3.4. Rezolvarea grafică.	5.13.
5.3. Elemente mecanice de protecție.	5.15.
5.4. Concluzii.	5.16.
<b>6. Concepția și realizarea sistemului de acționare și de conducere al robocarului.</b>	6.1.
6.1. Sistemul de acționare.	6.2.
6.1.1. Sistemul de acționare a mecanismului de tracțiune.	6.3.
6.1.1.1. Schema bloc și schema de principiu a EEMCC.	6.3.
6.1.1.2. Generarea prin soft a semnalelor de comandă.	6.7.
6.1.2. Sistemul de acționare a mecanismului de virare.	6.10.
6.1.2.1. Schema bloc și schema de principiu a EEMPP.	6.11.
6.1.2.2. Generarea prin soft a semnalelor de comandă.	6.13.
6.1.3. Sistemul de orientare a camerei video.	6.14.
6.1.3.1. Schema bloc și schema de principiu a elementului de execuție utilizat pentru orientarea camerei video.	6.15.
6.1.3.2. Generarea prin soft a semnalelor de comandă.	6.16.
6.2. Sistemul de conducere al vehiculului.	6.22.
6.2.1. Placa de bază SDM - PB.	6.22.
6.2.2. Placa de extensie SDM - EX01.	6.24.
6.2.3. Mediul integrat de dezvoltare PCMON.	6.25.
6.3. Concluzii.	6.29.
<b>7. Rezultate experimentale.</b>	7.1.
7.1. Verificarea sistemului mecanic.	7.1.
7.2. Experimente efectuate cu elementele sistemului de acționare.	7.2.
7.2.1. Verificarea EEMCC.	7.2.
7.2.2. Verificarea EEMPP.	7.5.
7.2.3. Verificarea elementelor de execuție ale sistemului de orientare a camerei video.	7.8.
7.3. Testarea programelor utilizate pentru deplasarea robocarului.	7.8.
7.4. Calibrarea camerei video.	7.31.
7.5. Determinarea poziției unui punct în spațiul 2D cu ajutorul camerei video și simularea deplasării robocarului pe o traiectorie oarecare.	7.35.

---

7.6. Planificarea traiectoriei robocarului în cadrul hiper-	7.39.
sistemului CIM.	
<b>8. Concluzii finale și contribuții.</b>	<b>8.1.</b>
8.1. Considerații referitoare la modul de realizare a unui	8.1.
robocar.	
8.2. Contribuții ale autorului.	8.3.
8.2.1. Obiectivele cercetării	8.4.
8.2.2. Rezultatele obținute referitor la obiectivele pro-	8.6.
puse.	
8.2.3. Contribuții originale.	8.8.
<b>Anexe</b>	<b>A.1.</b>
<b>Bibliografie</b>	<b>B.1.</b>



## INTRODUCERE

Automatizarea producției ridică problema integrării funcționale a diferitelor subsisteme, astfel încât să se obțină sisteme de fabricație adecvate, care să realizeze sarcinile variabile specifice fiecărui gen de producție în parte.

Structura unui sistem de fabricație are la bază sarcinile de producție impuse simultan sistemului de fabricație, care va fi astfel conceput încât toate subsistemele sale componente să fie simultan integrate, în procesul de fabricație. În acest sens semnificative sunt liniile de transfer și centrele de prelucrare, ca fiind instalații specializate destinate producției de serie mare și de masă. Compunerea unor sisteme de fabricație automatizate, destinate producției de serie mică și mijlocie, are la bază ansamblul sarcinilor de producție.

Cum problematica automatizării funcționării unui sistem de fabricație implică două condiții esențiale:

- ◆ condiția de integrare fizică a subsistemelor;
- ◆ condiția de comandă integrată și coordonată;

Întregul ansamblu va trebui orientat spre realizarea acestora, fiind pus un accent deosebit pe cea de-a doua condiție.

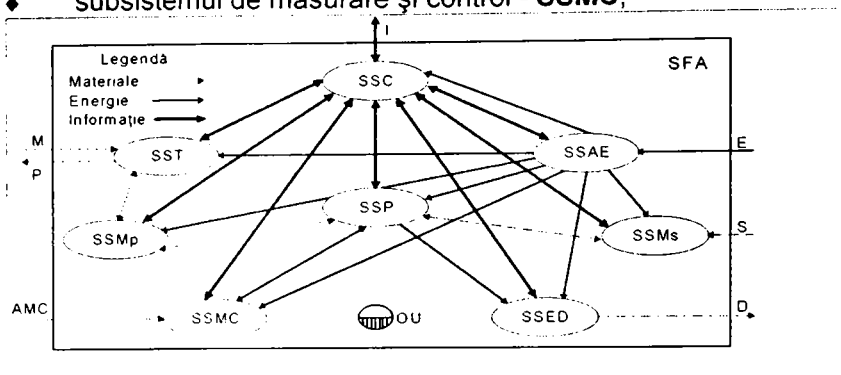
Comanda și reglarea joacă un rol important în realizarea automatizării, materializând cuplarea funcțională a subsistemelor parțiale și structura funcțională a sistemului de fabricație în ansamblu.

Deoarece informația și fluxul de informații constituie categorii caracteristice ale procesului de comandă, sistemul de fabricație automatizat se va prezenta ca un complex de elemente, care comunică între ele conform teoriei informaționale, în vederea îndeplinirii unei sarcini de producție.

Astfel conceput sistemul de fabricație va fi un sistem tehnic caracterizat prin interacțiunea **materiei, energiei și informației**, iar integrarea acestuia într-o structură mai mare (un hipersistem), revine la a integra sistemul de fabricație într-o structură informațională mai complexă, prin conceperea sa ca un sistem de prelucrare a datelor.

Modelul general al unui sistem de fabricație automatizat - SFA, prezentat în figura 1, conține:

- ◆ subsistemul de comandă - **SSC**;
- ◆ subsistemul de prelucrare - **SSP**;
- ◆ subsistemul de manipulare - **SSM** cu subdiviziunile sale:
- ◆ subsistemul de manipulare a pieselor - **SSMP**;
- ◆ subsistemul de manipulare a sculelor - **SSMS**;
- ◆ subsistemul de transport - **SST** a pieselor și subansamblurilor;
- ◆ subsistemul de alimentare cu energie electrică - **SSAE**;
- ◆ subsistemul de evacuare a deșeurilor - **SSED**;
- ◆ subsistemul de măsurare și control - **SSMC**;



**Figura 1.**

Prezența operatorului uman **OU**, este indispensabilă chiar și în cazul sistemului de fabricație automatizată - **SFA**, când toate operațiile necesare transformării semifabricatului într-o piesă finită au loc integral fără intervenția sa. Implicarea **OU** în **SFA** se realizează prin activități de întreținere, reglaje, pregătirea programelor și uneori controlul calității.

**Flexibilitatea constă în capacitatea de modificare a sistemului de fabricație în vederea adaptării sale, în mod automat, unor noi sarcini de producție.**

Spre deosebire de sistemele de fabricație automatizate rigide (ex. liniile de fabricație rigide), care sunt concepute pentru realizarea unei singure sarcini de producție, sistemele de fabricație automatizate flexibile sunt astfel concepute încât să se poată transforma (adapta) în vederea realizării mai multor sarcini de producție diferite. Sarcinile de producție se referă la realizarea propriu-zisă a pieselor, subansamblurilor sau ansamblurilor, concretizându-se în întreaga activitate întreprinsă de la transformarea semifabricatului în produs finit în cazul pieselor, sau a operației de montaj în cazul subansamblurilor sau ansamblului general. Câteva exemple de sisteme de fabricație flexibile sunt: liniile de fabricație adaptive la câteva variante de sarcini de producție, centrele de fabricație

cu comandă numerică, liniile de fabricație cu comandă numerică, sisteme integrate de mașini-unelte și instalații logistice (**robocare**) etc., toate fiind comandate de structuri ierarhizate de echipamente de prelucrare a datelor.

Unul din scopurile finale ale cercetărilor în robotică este acela de creare a unor roboți mobili autonomi.

Cercetările în acest domeniu au debutat în anii '60. Roboții mobili cu un sistem de locomoție cu roți sau bipezi, sunt studiați în laboratoarele universitare sau guvernamentale din întreaga lume.

Motivele care au determinat interesul crescut și deci dezvoltarea rapidă a domeniului sunt două. Primul, roboții mobili au un potențial de aplicabilitate ridicat tocmai datorită mobilității lor. Spre exemplu un manipulator mobil are aplicații potențiale în domeniul industrial, militar și în domeniul serviciilor, deoarece combină flexibilitatea manipulatorului cu mobilitatea platformei la care acesta este atașat. Al doilea motiv este dat de recentele progrese în domeniul calculatoarelor, al senzorilor, al controlului și al procedeelelor de navigare, progrese care fac ca roboții mobili să devină tot mai inteligenți, din ce în ce mai autonomi și cu un cost din ce în ce mai scăzut. Ca rezultat, aria de aplicabilitate a roboților mobili se lărgeste continuu.

Cercetările în domeniu au condus la apariția a trei familii de roboți capabili să se deplaseze.

Prima familie este cea a roboților mobili teleghidați. Denumirea lor în literatura de specialitate este **remote operated vehicles (ROV)**. Acești roboți sunt comandați de către un operator uman, de la distanță, prin unde radio. Utilizarea lor prezintă avantajul protejării operatorului uman, deoarece ei pot acționa în spații greu accesibile sau periculoase. Dezavantajele acestui tip de robot mobil sunt două. Primul, este legat de prezența teleoperatorului. Pe lângă faptul că ocupă timpul unui operator, reclamă în plus prezența acestuia într-o vecinătate mai mică sau mai mare. Spre exemplu, în cazul unor roboți teleghidați care lucrează în mediu subacvatic, este necesară prezența unei nave într-o zonă învecinată aceleia în care aceștia evoluează. Cel de al doilea dezavantaj este cel al vitezei finite de propagare a undelor radio de comandă. Nu este astfel posibilă utilizarea acestui tip de robot mobil pe suprafața planetei Marte, spre exemplu. Timpul necesar undelor radio de comandă pentru a parcurge distanța Terra - Marte dus - întors este de 40 minute, timp în care robotul a navigat necontrolat. Apare astfel o limitare a

spațiului de operare, la zona în care nu se fac încă sensibile întârzierile datorate propagării undelor de comandă.

Cea de-a doua familie este cea a vehiculelor ghidate. Pentru acestea denumirea consacrată în literatură este de **Automated Guided Vehicles** (AGV). Vehiculele ghidate sunt roboți capabili să execute o deplasare de-a lungul unei traiectorii sugerate de către un anumit ghidaj. Firele îngropate sunt ghidajul cel mai des folosit de către producătorii japonezi. Dar poate fi la fel de bine folosit un marcaj trasat pe caldarâm. Obiectivul sistemului de conducere al vehiculelor ghidate automat este acela de a recunoaște și de a urmări marcajul. Sistemele mai avansate sunt capabile chiar să negocieze cu obstacolele staționate pe marcaj, prin ocolirea lor și recunoașterea marcajului pe partea opusă. O serie de dezavantaje umbresc utilitatea acestor sisteme ghidate. În primul rând, costul ridicat al unui asemenea sistem, implicând modelări corespunzătoare ale mediului. Nu trebuie uitat nici faptul că păstrarea acestor marcaje în condiții de funcționare necesită un efort susținut. În plus, este evident că o eventuală modificare a traseelor prezintă numeroase complicații, la un cost nu tocmai scăzut.

Roboții mobili autonomi alcătuiesc cea de a treia familie. Sunt denumiți **roboți mobili** (RM). Din această clasă fac parte sisteme capabile să îndeplinească anumite acțiuni fără intervenția directă a unui operator uman. Referindu-se la deplasare, așa cum s-a făcut și pentru cele două clase anterioare, roboții mobili autonomi sunt capabili să genereze o traiectorie validă între cele două puncte de interes, să urmărească această traiectorie fără o intervenție externă, într-un mediu natural în care nu s-au efectuat modificări.

Deși se consideră că fac parte din clasa roboților mobili, sistemele de conducere automată a autovehiculelor pot fi plasate într-o clasă intermediară ultimelor două. Aceste vehicule au o complexitate superioară în comparație cu cea a vehiculelor ghidate, fiind capabile să proceseze o cantitate mult mai mare și mai complexă de informație. Sistemele de conducere automată a autovehiculelor (*on - road*) urmăresc totuși repere trasate pe calea de rulare. Ideea de bază a acestui gen de sisteme este aceea de recunoaștere și urmărire a benzilor albe care marchează și delimitează carosabilul. Grupa vehiculelor comandate automat în teren accidentat (*off - road*), este grupa care face pasul spre/și legătura cu roboții mobili autonomi. Acestea din urmă sunt nevoite să se descurce într-un mediu nestructurat.



Transferul autorității între cele trei clase mai sus prezentate se face gradat. Dacă pentru clasa roboților teleghidatți deciziile, în totalitate, erau luate de către operatorul uman, robotului rămânându-i doar rolul de executor, pentru clasele superioare autoritatea începe să se împartă între operator și sistem. La autovehiculele ghidate apare deja situația în care sistemul trebuie să ia anumite decizii, cum ar fi spre exemplu ocolirea obstacolului care staționează pe traseu. În cazul roboților mobili autoritatea le este atribuită în exclusivitate. Pe baza unor algoritmi preprogramați sau pe baza cunoștințelor apriori acumulate, robotul decide singur acțiunea care urmează a fi executată.

Nefiind posibilă o delimitare strictă între cele trei clase de mai sus, datorită complexității aplicațiilor, se poate considera că dezideratul principal este realizarea unui vehicul cât mai "inteligent" posibil, capabil să se deplaseze într-un mediu parțial cunoscut sau total necunoscut, fără intervenția operatorului uman. Un astfel de vehicul poate fi utilizat atât pentru rezolvarea unor sarcini de transport în sfera producției, cât și în domeniul serviciilor. În cazul în care este utilizat pentru transport, vehiculului i se atașează un dispozitiv clasic de încărcare-descărcare, iar dacă trebuie să efectueze alte operații, se poate utiliza același vehicul, având montat pe platforma sa un robot.

Tendința cercetătorilor din domeniu conduce spre realizarea unor roboți tot mai avansați, având ca finalitate crearea unor roboți care să accepte descrieri de nivel înalt. Acestor roboți li se va comunica **ce să facă și cum să facă** .

Problema care stă în fața constructorilor este cea dată de programarea roboților astfel încât aceștia să-și îndeplinească sarcinile cât mai flexibil, într-un mod cât mai adaptiv. Lipsa informației complete despre mediu, în momentul proiectării, lipsă la care se adaugă și zgomotul inerent senzorilor, face ca sarcina programatorului să nu fie deloc ușoară. De fapt, complexitatea universului face imposibilă programarea **totală**. Soluția care vine să rezolve această dilemă este cea dată de noile teorii. Acum nu mai este necesară o informație apriori, cât mai vastă despre mediu (uneori nici măcar despre el însuși, robotul) ci robotul este instruit să învețe. Această abordare simplifică pe de o parte sarcina constructorului. Momentul actual este caracterizat de indecizia formei în care să se îmbine metodologia de programare tradițională cu noile forme de adaptare prin instruire.

Robocarele construite astăzi au numeroase aplicații, fiind în special utilizate în sistemele de transport industrial cu grad ridicat de automatizare.

**Un sistem de robocare (de vehicule ghidate automat) este un sistem avansat de transport și de manipulare a pieselor și subsansamblurilor, care implică mai multe vehicule conduse automat de către un sistem de conducere.**

Comparativ cu alte sisteme convenționale de transfer al materialelor cum ar fi: transportoarele cu role, transportoarele cu bandă, conveierele etc., robocarele sunt capabile să-și selecteze singure traseul pentru a ajunge la destinația dorită. Sistemul de conducere și de comandă al robocarelor le facilitează primirea comenzilor de identificare a sarcinilor, care constau în informații privind încărcătura, traseul ce urmează a fi parcurs, destinația și alte informații specifice transportului pe distanțe medii și mari. Primele sisteme de transport cu robocare s-au dezvoltat în domeniul distribuției mărfurilor din incinta depozitelor.

Deși constituie o aplicație importantă în industrie, sistemele de transport cu robocare au evoluat numai în două direcții majore:

- mișcarea materialelor în/din depozite din/spre zonele de fabricație în cazul operațiilor de prelucrare;
- transportul de la un post de lucru la altul în cazul operațiilor de montaj;

O aplicație industrială, care câștigă din ce în ce mai mult teren, este utilizarea robocarelor pentru manipularea materialelor în camere curate (clean-rooms).

Sistemele de transport cu robocare se aplică și în alte domenii, cum ar fi:

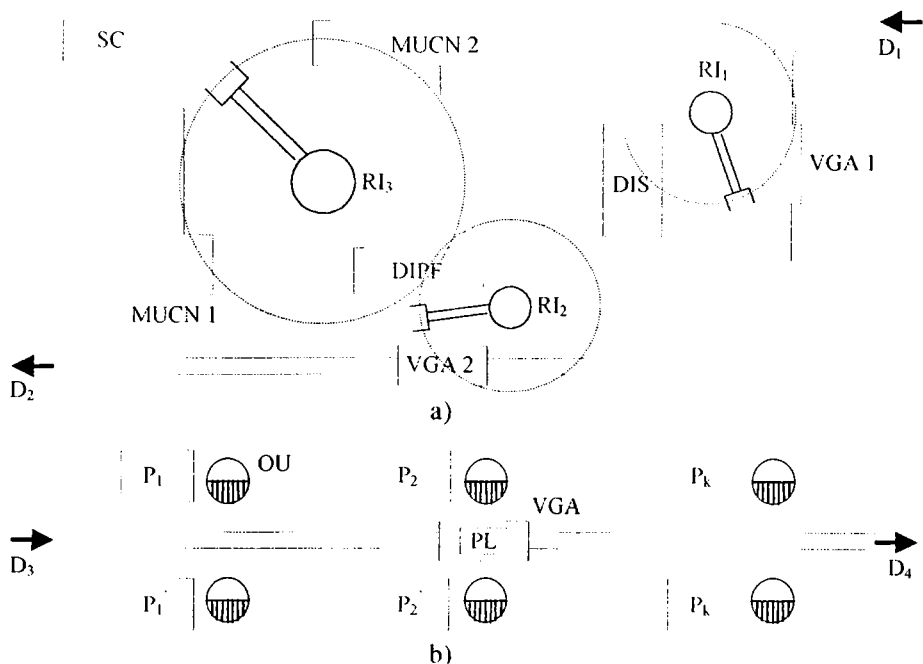
- în birouri, la distribuirea corespondenței, a mesajelor și a coletelor;
- în spitale, pentru distribuirea alimentelor, a rufelor spălate sau colectarea celor murdare;

Deci, sistemele de robocare sunt sisteme flexibile de manipulare și transfer, capabile să se adapteze prompt la orice modificare a proceselor tehnologice, posedând un sistem de comandă și control complex.

Robocarele și sistemele de robocare sunt capabile să conlucreze cu depozite automate **AS/RS** (Automated Storage and Retrieval Systems),

roboți industriali, mașini-unelte **CNC**, instalații de alimentare/evacuare și alte echipamente comandate prin calculatoare de proces.

Două exemple de metasisteme tehnice, care integrează un sistem de robocare, sunt prezentate în figura 2. Astfel, în figura 2a se prezintă o secvență dintr-o rețea de distribuție a semifabricatelor la posturile de prelucrare, posturi care pot fi mașini-unelte, centre de prelucrare sau celule flexibile de fabricație. Din figura prezentată rezultă că celula de fabricație este compusă din: două mașini-unelte cu comandă numerică - **MUCN**, un depozit intermediar pentru semifabricate - **DIS**, un depozit intermediar pentru produse finite - **DIPF**, sistemul de comandă - **SC** al celulei de fabricație, trei roboți industriali - **RI** și un roboțar - **VGA**. Robocarul **VGA** asigură transferul semifabricatelor de la depozitul de semifabricate **D<sub>1</sub>** la celula fabricație și a pieselor prelucrate de la **DIPF** la depozitul **D<sub>2</sub>** de piese finite. Robotul industrial **RI<sub>1</sub>** asigură transferul semifabricatelor de pe **VGA** în **DIS**, iar robotul industrial **RI<sub>2</sub>** transferă piesele prelucrate din **DIPF** în **VGA**. Operațiile de manipulare din cadrul celulei de fabricație sunt realizate de robotul industrial **RI<sub>3</sub>**.



**Figura 2**

În cazul liniilor de asamblare platforma robocarelor constituie chiar platforma de lucru - **PL**, pe care se realizează operațiile de montaj.

Astfel, după cum rezultă și din figura 2b, **robocarul - VGA**, care are pe **PL** componentele necesare ansamblului general, extrase din depozitul -  $D_3$ , se deplasează de la un punct de lucru la altul  $P_1 - P_k/P'_1 - P'_k$  posturi în care operatori umani - **OU** execută operațiile de montaj. În final întregul ansamblu este transferat de la linia de asamblare la depozitul  $D_4$ , după care robocarul își reia ciclul de lucru.

Din figurile prezentate **rezultă necesitatea sistemelor de robocare, pentru a asigura creșterea flexibilității sistemelor de fabricație**. Selectarea sistemului de robocare, în raport cu sistemele convenționale de transfer, este precedată de o analiză a avantajelor și dezavantajelor fiecăruia.

\* \* \*

Teza de doctorat elaborată cuprinde 8 capitole și o listă bibliografică cu 220 de titluri. Lucrările indicate în bibliografie au fost consultate în totalitate de către autor, iar o parte dintre ele se află în posesia acestuia.

**Capitolul 1** conține o prezentare sistematică a principalelor noțiuni referitoare la sistemele de transport automat. Sunt de asemenea prezentate elementele constructive ale unui robocar și se pun în evidență caracteristicile acestora și condițiile pe care trebuie să le îndeplinească fiecare element. Prezentare se face atât cu scopul de a constitui baza de pornire pentru abordările ulterioare, cât și pentru a pune la dispoziția celor interesați în domeniul VGA a unor date necesare aprofundării problemelor pe care le ridică aceste vehicule. **Sistematizarea s-a făcut pe baza consultării unui vast material bibliografic și reprezintă o contribuție a autorului.**

**Capitolul 2** încadrează sistemul de conducere al robocarului în ierarhia sistemelor de conducere a proceselor de fabricație automatizate. Pe baza acestei ierarhizări se pot stabili funcțiile pe care le îndeplinește sistemul de conducere al vehiculului atunci când acesta face parte dintr-un sistem de conducere descentralizat. De asemenea se definește termenul **navigație**, a cărui semnificație nu este reliefată în bibliografia consultată. Autorul își exprimă părerile personale referitoare la funcțiile conducerii și ale navigației și dă o explicație termenului **navigare**, diferit, după părerea sa, de termenul **navigație**.

**Capitolul 3** este capitolul în care se abordează problema determinării situației relative a robocarelor pe baza odometriei. Este prezentată

metoda odometrică și se elaborează un program original utilizat de către sistemul de conducere pentru efectuarea deplasării robocarului în următoarele moduri:

- prin comandă manuală;
- pe o traiectorie impusă;
- pe o traiectorie învățată în urma instruirii.

Deoarece astfel de programe nu au fost întâlnite în literatura de specialitate consultată, autorul a considerat necesară elaborarea acestui program pentru a fi utilizat în aplicațiile ulterioare. **Programul reprezintă în întregime contribuția autorului.**

**Capitolul 4** este rezultatul unei activități îndelungate a autorului în domeniul roboticii și a electronicii. Pe parcursul acestui capitol se prezintă principiul determinării situației absolute a robocarelor. Situația absolută se determină prin măsurarea distanței dintre vehicul și un punct care are coordonatele cunoscute în raport cu un sistem de referință. Se măsoară totodată unghiul format de axa longitudinală a vehiculului cu una dintre axele de coordonate. Metodele și mijloacele utilizate pentru măsurarea acestor mărimi sunt prezentate sistematic în acest capitol. **De asemenea sunt prezentate câteva metode originale care constituie contribuții ale autorului și se face o comparație a acestora.**

**Capitolul 5** conține o prezentare a elementelor mecanice utilizate în construcția robocarului și o justificare a opțiunii autorului pentru această variantă. De asemenea se face prezentarea detaliată a structurii mecanice a vehiculului. **Robocarul a fost construit în totalitate de către autor în această variantă.**

**Capitolul 6** are un caracter preponderent aplicativ. Aici se arată modul în care au fost alese soluțiile pentru realizarea echipamentelor electronice ale robocarului și se prezintă programele necesare sistemului de conducere. **Atât circuitele electronice cât și programele necesare pentru aplicații și experimente au fost concepute și realizate de către autor.**

**Capitolul 7** prezintă rezultatele experimentale. Aceste rezultate se bazează pe o serie de experimente care s-au efectuat în realitate într-o altă ordine decât cea a prezentării materialului cuprins în teză. În primul rând s-au experimentat circuitele electronice necesare alimentării motoarelor și programele corespunzătoare, apoi s-au elaborat și s-au experimentat programele pentru achiziția imaginilor, care au fost în continuare utilizate în scopul calibrării camerei video și apoi pentru

determinarea situației vehiculului. În etapa următoare au fost determinate pe baza metodelor propuse coordonatele unor puncte, aceste determinări confirmând corectitudinea principiilor teoretice. De asemenea s-a efectuat un experiment constând din planificarea traiectoriei și conducerea vehiculului pe traiectoria planificată. Această traiectorie corespunde deplasării în spațiul de lucru aflat în laboratorul de robotică al Facultății de Mecanică. **Toate experimentele și modul de realizare al acestora sunt contribuții ale autorului.**

**Capitolul 8** sintetizează contribuțiile originale ale autorului și prezintă concluziile finale.

\* \* \*

În țara noastră, în Universitatea "Politehnica" există o tradiție în ceea ce privește robotica, care își are începuturile cu mai bine de două decenii în urmă. Personalități din învățământul universitar, cum ar fi Prof. Dr. ing. Tiberiu Mureșan și Prof. Dr. ing. Francisc Kovacs, au contribuit la dezvoltarea roboticii pe plan local, național și internațional.

Sistemele de transport automate nu au stat însă de la începuturile activității în atenția colectivelor de cercetare.

Autorul, făcând parte din Colectivul Multidisciplinar de Roboți Industriali de aproape douăzeci de ani, a acumulat o bogată experiență în domeniul roboticii, sub îndrumarea competentă a șefului de colectiv de pe atunci, Prof. Dr. ing. Tiberiu Mureșan.

De peste zece ani, preocupările sale în robotică au luat direcția sistemelor de transport automate și a vehiculelor ghidate automat.

În România, mai puțin cunoscut, sau mai bine zis mai puțin abordat, domeniul roboților mobili lasă loc fiecăruia de a-și pune la contribuție activitatea creatoare. Noțiunile referitoare la roboții mobili nu sunt standardizate. Bibliografia este relativ restrânsă, numărul persoanelor cu preocupări în domeniu fiind redus. Există deci perspectiva abordării unor noi direcții de cercetare și de realizare a roboților mobili. Iată de ce autorul a luat hotărârea de a-și fructifica experiența acumulată de-a lungul anilor într-o teză de doctorat, cu dorința de a aduce câteva contribuții modeste la promovarea acestui domeniu în cercetarea românească. Prin teza de față, prin lucrările care tratează tema robocarelor, publicate până acum, dar și prin cartea aflată în curs de

publicare se dorește clarificarea anumitor noțiuni referitoare la această temă, atribuirea unor denumiri și termeni care nu au corespondent în limba română și aducerea câtorva noutăți în metodele de conducere.

Cu forțe proprii și mijloace modeste a fost realizat un robocar funcțional, cu ajutorul căruia s-au efectuat experimentări. Acesta va fi pus la dispoziție, în continuare, cu colaborarea autorului, celor care au ca obiectiv studiul vehiculelor ghidate automat.

Se face mențiunea că domeniul **robocare** a devenit o pasiune a autorului și a colectivului pe care l-a format și că dorința sa este de a continua și pe viitor cercetarea în direcția **roboți mobili și VGA**.

Se poate considera că teza de față este o deschidere de drum în acest domeniu pentru colectivul format în cadrul Universității "Politehnica" Timișoara și că succesele nu vor întârzia să apară.

În toată perioada activității în învățământ și cercetare autorul a beneficiat de îndrumarea competentă și plină de înțelegere a domnului prof. dr. ing. Tiberiu Mureșan, de care se simte legat atât sufletește cât și profesional. De aceea, pe această cale îi aduce calde mulțumiri pentru cunoștințele acumulate alături de domnia sa, pentru sfaturile, mobilizarea și îndrumarea susținută de care a beneficiat de-a lungul anilor în care a avut loc elaborarea tezei.

Nu în ultimul rând autorul aduce mulțumiri celor care au participat la realizarea robocarului, studenți și cadre didactice.

Mulțumiri se aduc și colegilor, care prin sfaturi și încurajări permanente au impulsionat finalizarea tezei.

Și în final, autorul aduce mulțumiri soției și celor trei copii ai săi, care nu numai că au dat dovadă de înțelegere și spirit de sacrificiu, dar au participat efectiv la redactarea lucrării. **LOR le este dedicată această teză.**





## Capitolul 1

### SISTEME DE TRANSPORT AUTOMAT. VEHICULE GHIDATE AUTOMAT

În acest capitol sunt sistematizate principalele noțiuni referitoare la sistemele de transport automat (STA) și sunt prezentate elementele constructive esențiale ale unui vehicul ghidat automat (VGA), integrat într-un sistem industrial de transport automat.

Creșterea continuă a flexibilității producției a făcut necesară automatizarea și flexibilizarea sistemelor de transport al materialelor, semifabricatelor sau al produselor finite [VDI 2510]. Față de transportul manual, transportul automat oferă următoarele avantaje :

- flux de materiale și informații organizat;
- cunoașterea exactă a stării transportului;
- scăderea cheltuielilor de personal, mai ales când se lucrează în mai multe schimburi;
- reducerea erorilor care apar în sistemul de transport.

O clasificare a sistemelor de transport este prezentată în figura 1.1.

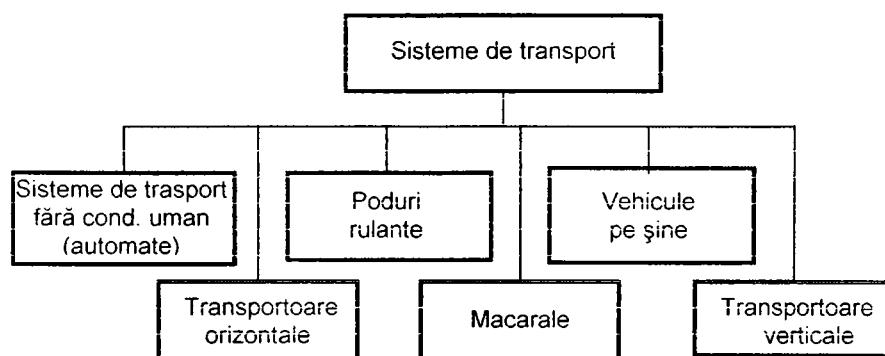


Figura 1.1.

Sistemele de transport pot fi clasificate în două categorii, așa cum rezultă din tabelul 1.1.

Tabelul 1.1.

Conduse de către operator	Conduse automat
Conduse direct de către operator (motostivuitoare, autocamion)	Traseu programat de către operator (sist. de transport fără cond. uman)
Conduse de la distanță de către operator (manipulator)	Traseul nu este programat (roboți autonomi)

Criteriile pe baza cărora utilizatorul ia decizii pentru alegerea unui sistem de transport automat sunt:

- investițiile necesare instalării sistemului;
- costurile;
- necesități legate de suprafețele deservite;
- necesități legate de spațiile deservite;
- erori în transport, aprovizionare etc. [ELB 91].

Pe lângă criteriile enunțate, flexibilitatea are rolul cel mai important. Acest criteriu poate fi tratat din următoarele puncte de vedere:

- dacă sistemul de transport poate fi integrat în structurile existente;
- dacă pot fi transportate materiale diferite;
- dacă pot avea loc modificări ale traseului parcurs;
- dacă sistemul de transport se poate adapta la modificările sistemului de fabricație;
- dacă este acceptată modificarea destinației sistemului de transport;
- dacă sistemul de transport poate fi adaptat unui nivel de automatizare variabil (în schimbare).

Un sistem de transport este flexibil dacă îndeplinește cât mai multe din criteriile de mai sus.

Sistemele de transport automate, performante, oferă următoarele avantaje :

- îmbunătățirea condițiilor de producție, prin ordonarea aprovizionării și prin transportul organizat, în condiții de siguranță maximă;
- efectuarea transportului cu precizie ridicată;
- realizarea ușoară a intersecțiilor și bifurcațiilor;
- transportul este posibil și în afara halelor;
- transparentă ridicată a sistemelor de transport;
- posibilitatea utilizării căilor de rulare existente;

- posibilitatea integrării altor funcțiuni.

Multitudinea sistemelor existente la ora actuală, cât și varietatea aplicațiilor, face dificilă concentrarea tuturor variantelor într-un model care să reprezinte semnificativ vehiculul ghidat automat. Totuși, în cele ce urmează se va încerca o sistematizare a principalelor elemente constructive ale unui **VGA** și a funcțiilor îndeplinite de către acesta [SCH 96].

O definiție a sistemelor de transport automate și a vehiculelor ghidate automat ar fi următoarea:

**Sistemele de transport automate sunt sisteme de transport cu grad ridicat de automatizare, a căror sarcină este transportul de materiale (și nu de persoane)[VDI 22510].** Ele sunt utilizate în mediul interior și/sau exterior și constau din:

- unul sau mai multe vehicule;
- un sistem de conducere;
- sisteme pentru determinarea situației;
- sisteme pentru transmisii de date;
- infrastructură și elemente periferice.

**Vehiculele Ghidate Automat (VGA) sunt vehicule de transport cu sistem motor propriu, conduse automat și alimentate de la sursă proprie de energie. Se utilizează la transportul de materiale, fiind dotate cu sisteme de încărcare / descărcare active sau pasive.**

Un vehicul ghidat automat, ca de altfel orice robot, se bazează pe trei mari categorii de elemente constructive și anume:

- ◆ elemente mecanice;
- ◆ echipamente electrice și electronice;
- ◆ sisteme de conducere.

O reprezentare schematică simplificată a elementelor constructive ale unui VGA este prezentată în figura 1.2.

**Sistemul mecanic** este compus din trei categorii de elemente, care fac parte din construcția oricărui roboțar (șasiul, sistemul de rulare și dispozitivul mecanic de protecție) și un element opțional, dispozitivul de încărcare-descărcare. Acesta poate fi înlocuit, în anumite cazuri, cu un manipulator, sau poate lipsi.

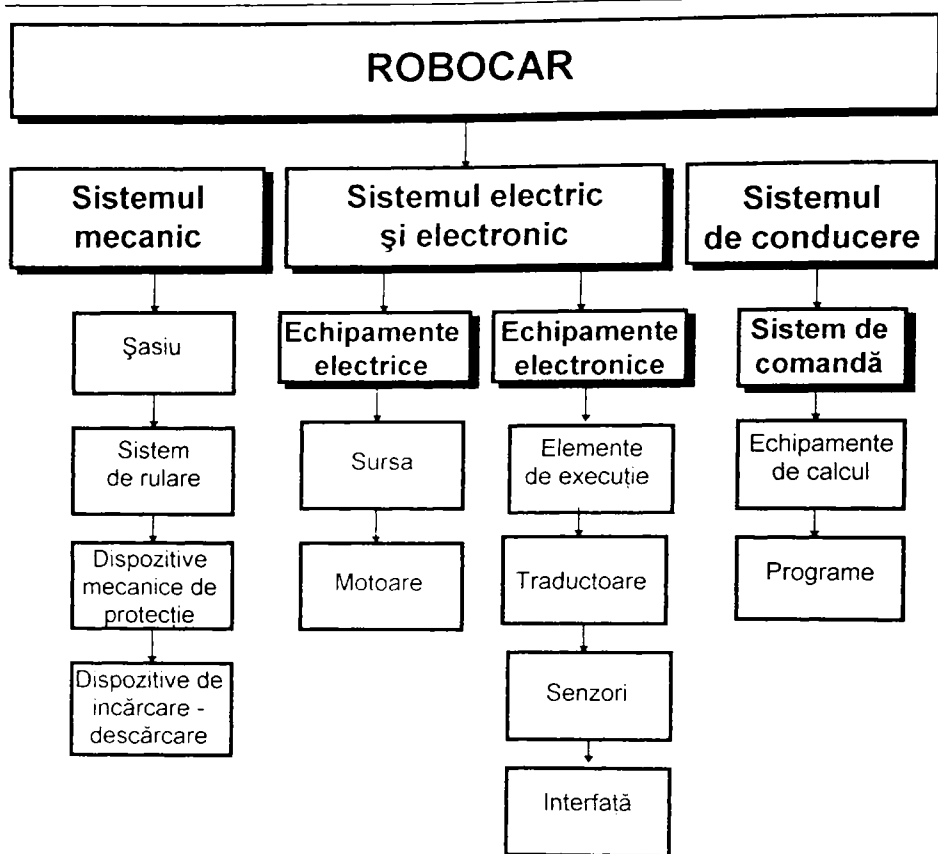


Figura 1.2.

**Echipamentele electrice** sunt: sursa de energie electrică, motoarele utilizate în cadrul sistemului de tracțiune și a sistemului de virare, motoarele care echipează dispozitivele de încărcare-descărcare, etc.

**Echipamentele electronice** principale sunt: traductoarele (care fac parte din sistemul de acționare), sistemele senzoriale (necesare pentru determinarea situației robocarului sau a prezenței obstacolelor), dispozitivele de interfață (tastatură, afișaj) și echipamentul pentru transmiterea informațiilor. Informațiile pot fi transmise spre și dinspre sistemul de conducere ierarhic superior sau spre alte componente ale sistemului de transport situate pe același nivel ierarhic.

**Sistemul de conducere** are în componența sa un echipament de calcul numeric și seturi de programe, necesare conducerii acțiunilor efectuate de către robocar.

---

## 1.1. Descrierea elementelor constructive ale VGA.

În continuare vor fi analizate elementele principale care compun un vehicul ghidat automat și elementele spațiului de lucru, care constituie componente ale sistemului de transport automat și cu care robocarul se află în contact nemijlocit. Aceste componente sunt prezentate în figura 1.3.

### 1.1.1. Sistemul mecanic.

Sistemul mecanic al robocarului este compus din următoarele elemente de bază:

#### 1.1.1.1. Șasiul.

Acesta reprezintă structura mecanică de rezistență și are rolul de a susține sistemul de rulare, sursele de energie, dispozitivul de încărcare-descărcare a sarcinii și echipamentele electrice și electronice. În general șasiul este realizat sub forma unei construcții sudate din profile metalice și are o formă adecvată funcției pe care o îndeplinește robocarul. El oferă posibilitatea montării simple a întregului echipament care compune vehiculul [ AND 94].

#### 1.1.1.2. Sistemul de rulare.

Sistemul de rulare cel mai uzual este cel care utilizează roți. Există și sisteme de rulare cu șenile, dar în continuare vor fi prezentate numai cele care utilizează roți [VDI 4451].

**Sistemul de rulare reprezintă ansamblul de elemente de rulare (roți), active sau pasive, orientabile sau fixe.**

Deci, sistemul de rulare cuprinde atât sistemul de tracțiune, cât și sistemul de virare.

Vehiculele echipate cu diverse tipuri de sisteme de rulare sunt caracterizate prin următoarele:

- comportare în mișcare;
- ușurința conducerii;

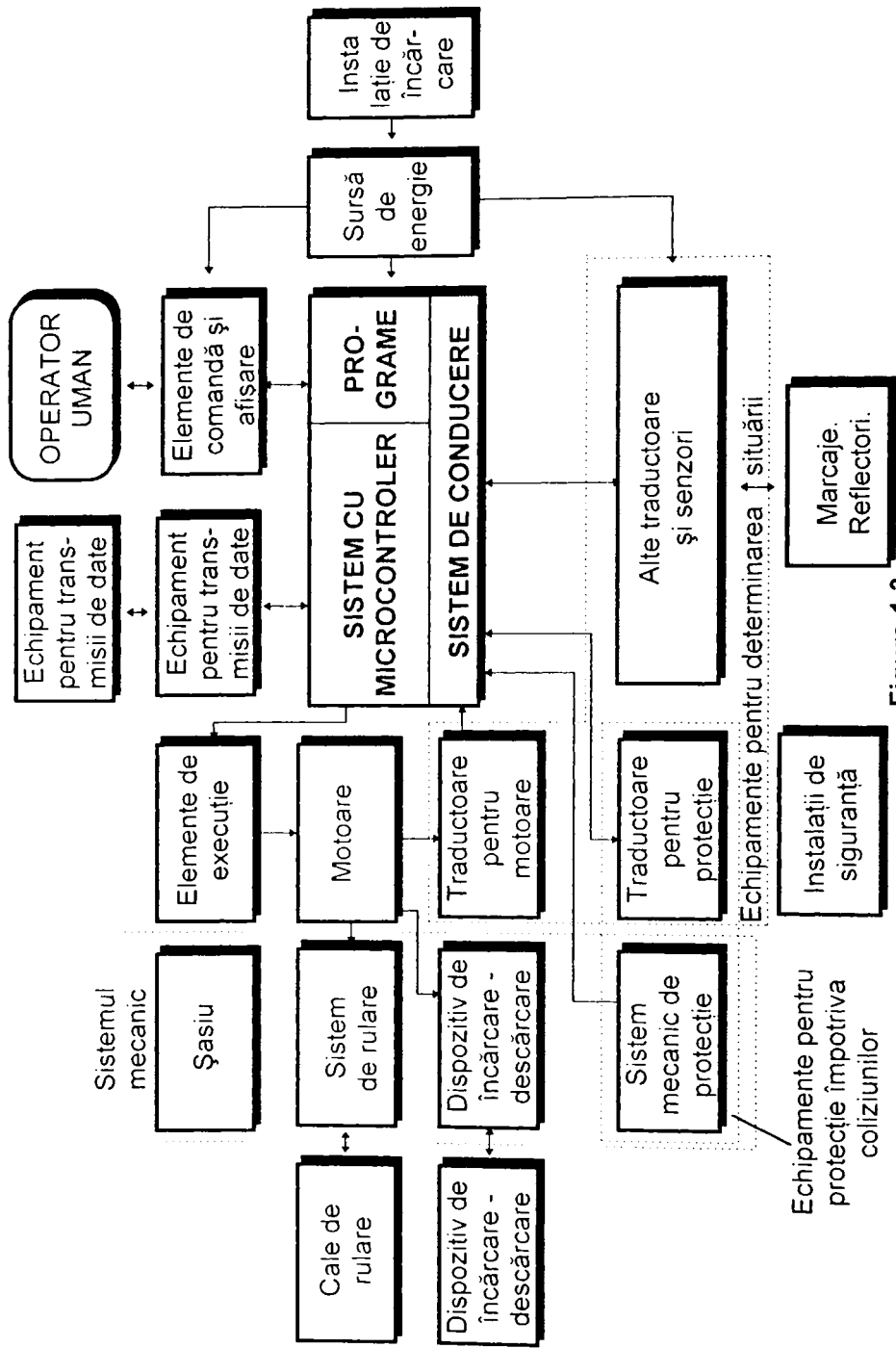
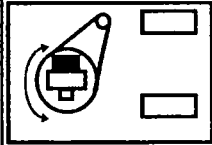
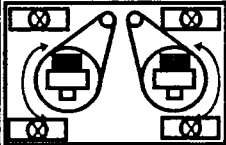
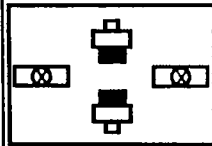
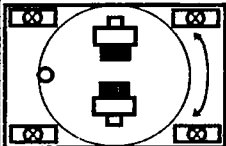
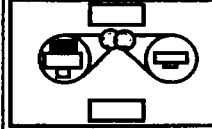
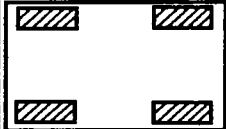



Figura 1.3.

Comportarea în mișcare este determinată de numărul gradelor de libertate ale sistemului de rulare. Vehiculele care se deplasează pe suprafețe au trei grade de libertate. În principiu, este posibil ca vehiculele care se deplasează liniar să fie conduse pe orice fel de traiectorie, dar orientarea șasiului raportată la traiectorie este dinainte stabilită prin sistemul de rulare. De aceea, spațiul necesar pentru deplasare, atunci când sunt parcurse curburi, este mai mare decât pentru alte vehicule.

Vehiculele care se deplasează pe suprafață au orientarea șasiului față de traiectorie independentă de poziția vehiculului. Sunt posibile deplasări în orice mod (translații și rotații) ale șasiului. Realizarea tehnică a acestui tip de vehicule este însă mai complexă.

În figura 1.4. sunt prezentate diverse tipuri de sisteme de rulare [VDI 2510].

Sistem de rulare	Mișcări posibile	Sistem de rulare	Mișcări posibile
 <p>Cu trei roți</p>	<ul style="list-style-type: none"> <li>• în linie (două grade de libertate)</li> <li>• virare în jurul axei posterioare</li> <li>• deplasare preferențială înainte, posibilă înapoi</li> </ul>	 <p>Două grupuri independente de rulare și virare</p>	<ul style="list-style-type: none"> <li>• mișcare pe suprafață (trei grade de libertate)</li> </ul>
 <p>Cu antrenare diferențială</p>	<ul style="list-style-type: none"> <li>• în linie (două grade de libertate)</li> <li>• virare în jurul axei centrale</li> <li>• deplasare înainte-înapoi</li> </ul>	 <p>Antrenare diferențială cu axă de rotire</p>	<ul style="list-style-type: none"> <li>• mișcare pe suprafață (trei grade de libertate)</li> </ul>
 <p>Sistem de virare cuplat</p>	<ul style="list-style-type: none"> <li>• în linie (două grade de libertate)</li> <li>• virare în jurul axei centrale</li> <li>• deplasare înainte-înapoi</li> </ul>	 <p>Antrenare mecanică</p>	<ul style="list-style-type: none"> <li>• mișcare pe suprafață (trei grade de libertate)</li> </ul>



Roată de tracțiune   
  Roată de sprijin   
 Mecanism de tracțiune   
 Roată de sprijin mobilă   
 Mecanism de virare cuplat

Figura 1.4.

Sunt posibile și alte variante decât cele prezentate în figură. De exemplu, elementele de rulare se pot amplasa și lateral (nu numai față - spate).

Aceasta nu influențează însă manevrabilitatea și spațiul necesar pentru deplasare al vehiculului. Un exemplu ar fi automobilul, care are roțile de tracțiune (și de virare) amplasate lateral față și poate fi comparat, din punct de vedere cinematic, cu un vehicul cu trei roți.

Numărul, amplasarea și tipul roților definesc structura sistemului de rulare. Vehiculele pot fi echipate cu trei până la șase roți (uneori chiar mai multe). Liniile care leagă roțile între ele reprezintă liniile de răsturnare ale sistemului de rulare.

Amplasarea roților poate avea loc în formă de :

- triunghi - sistem cu trei roți;
- dreptunghi sau romb - sistem cu patru roți;
- triunghi și dreptunghi - sistem cu cinci roți;
- dreptunghi - sistem cu șase roți;  
(amplasate frontal sau lateral).

Sistemele de rulare care au un număr mai mare de trei roți necesită suspensie. În acest caz (când se utilizează suspensie) se ține cont de faptul că, la deplasarea laterală a sarcinii (încărcare - descărcare), are loc o înclinare a robocarului. Sunt necesare soluții pentru asigurarea operațiilor de transfer fără risc.

Sistemul de virare poate fi realizat în două forme constructive:

- cu orientare geometrică - roțile de virare își modifică orientarea în spațiu, pe baza unor transmisii hidraulice, electrice, etc. și pot fi și roți de tracțiune.
- sistem fără orientare geometrică - roțile de virare sunt în același timp și roți de tracțiune, iar schimbarea direcției de deplasare are loc prin rotirea roților cu turații diferite.

Frânele pot fi acționate mecanic, electromecanic, hidraulic. Ele se combină cu frânarea electrică (prin intermediul motorului electric).

### 1.1.1.3. Dispozitive de încărcare - descărcare a sarcinii transportate.

Integrarea robocarelor în sistemele flexibile de fabricație necesită realizarea unor operații de încărcare-descărcare (transfer) a sarcinilor



transportate de către robocar. Pentru aceasta sunt necesare dispozitive speciale, care echipează atât robocarul, cât și stațiile în care are loc transferul [ AND 94].

Dispozitivele de transfer sunt realizate într-o multitudine de variante și pot fi alese în funcție de necesități și de forma constructivă a vehiculului.

#### **1.1.1.4. Sistemul mecanic de protecție.**

Sistemul mecanic de protecție este realizat cu ajutorul unor elemente elastice, care protejează, în momentul coliziunii cu un obstacol, atât robocarul, cât și obstacolul respectiv.

Sistemele mecanice au fost înlocuite treptat cu senzori ultrasonici, cu fascicule laser sau infraroșii. Utilizând astfel de senzori, zona de siguranță poate fi împărțită în mai multe domenii. Un prim domeniu îl constituie o zonă mai îndepărtată. Dacă obstacolul se află în această zonă, are loc o avertizare acustică și optică și reducerea vitezei vehiculului până când obstacolul este înlăturat. Dacă obstacolul se află în continuare pe traiectoria robocarului, distanța până la acesta se va reduce. În momentul în care obstacolul se va afla în zona a doua, are loc oprirea vehiculului [LEU 97].

Senzorii utilizați pentru detectarea obstacolelor pot fi folosiți în același timp și pentru determinarea situației vehiculului.

În afară de sistemele de protecție se utilizează și dispozitive de alarmare acustice și/sau optice cum ar fi: hupe, difuzoare, surse luminoase intermitente pentru semnalizarea direcției de mers, a deplasării sau a avariei.

#### **1.1.2. Echipamentul electric al robocarului.**

Echipamentul electric face parte din sistemul electric al vehiculului, alături de echipamentul electronic. Echipamentul electric este alcătuit din următoarele componente:

- ◆ sursele de alimentare cu energie electrică;
- ◆ motoarele electrice.

Se pot utiliza și alte tipuri de motoare și surse de energie, dar în studiul de față se vor considera numai cele electrice.

### 1.1.2.1. Sursele de alimentare cu energie electrică.

Sursele de energie uzuale sunt: acumuloarele (pentru motoarele electrice) și combustibilul lichid (pentru motoarele cu ardere internă). Acumuloarele electrice au tensiunile nominale 24, 48 sau 80 V și capacitățile cuprinse între câțiva Ah și 1000 Ah. Ca motoare cu ardere internă se utilizează motoare Otto sau Diesel. Majoritatea vehiculelor sunt însă echipate cu surse și motoare electrice.

Acumuloarele trebuiesc încărcate periodic. Acest lucru are loc de regulă folosind instalații de încărcare staționare, care sunt conectate la intervale regulate de timp la vehicul. În funcție de gradul de automatizare, încărcarea acumuloarelor are loc fără detașarea lor de vehicul sau prin detașare de acesta, iar cuplarea sistemului de încărcare se face manual sau automat.

Modul de realizare a sistemului de încărcare este determinat de următorii factori :

- timpul de lucru al vehiculului (unul, două sau trei schimburi);
- consumul de energie electrică al acestuia;
- spațiul disponibil pentru sursa de alimentare;
- încărcarea manuală sau automată;
- timpul de utilizare (staționări periodice, durata acestora);
- perioada de întreținere a acumuloarelor;
- durata de funcționare a acumuloarelor;
- investiția și costurile necesare întreținerii acumuloarelor.

Se definesc mai multe moduri de utilizare a acumuloarelor [VDI 2510]:

a). utilizarea întregii capacități. Acumulatorul se descarcă după un anumit număr de ore de funcționare și se încarcă într-un anumit interval de timp;

b). utilizarea întregii capacități, cu încărcări intermediare, necesare creșterii duratei de utilizare, urmată de o încărcare totală la sfârșitul schimbului;

c). reîncărcarea acumulatorului în timpul staționării, după intervale regulate de timp, nefiind necesară o încărcare la sfârșitul perioadei de lucru.

Tipul acumuloarelor folosite (cu plumb sau alcaline) depinde de modul de utilizare a acestora. În modurile a și b sunt indicate acumuloarele cu plumb, deoarece oferă cele mai mici costuri pe ora de funcționare. În

modul c sunt utilizate acumuloarele cu NiCd, care permit încărcări de scurtă durată la curenți mari. Acest mod de utilizare permite folosirea acumuloarelor cu capacități mici. Astfel se pot reduce considerabil dimensiunile și greutatea vehiculului. Prețul acumuloarelor cu NiCd este însă de patru până la șase ori mai mare decât al celor cu Pb.

### **1.1.2.2. Motoarele electrice.**

Sistemele de tracțiune și de virare ale VGA sunt în general echipate cu motoare electrice. Există însă și vehicule pentru transportul unor sarcini mari, care folosesc pentru propulsie motoare cu ardere internă.

Numărul și tipul motoarelor electrice depinde de soluția constructivă aleasă. Dacă se utilizează o singură roată pentru tracțiune și pentru virare, un motor electric antrenează roata de tracțiune, iar altul mecanismul de virare. Dacă soluția aleasă pentru tracțiune și virare este utilizarea a două roți antrenate, acestea sunt echipate cu motoare proprii, iar virarea se face prin rotirea roților cu turații diferite.

Datorită faptului că sursa de alimentare este de curent continuu, motoarele utilizate sunt de curent continuu, pas cu pas și fără perii. În ultimul timp sunt din ce în ce mai des utilizate pentru tracțiune motoarele de curent alternativ. Puterea motoarelor diferă în funcție de tipul vehiculului și de sarcina transportată.

### **1.1.3. Echipamentele electronice ale unui VGA.**

Numărul și diversitatea echipamentelor electronice din componența unui VGA este foarte mare și depinde, în primul rând, de complexitatea sistemului de transport din care face parte vehiculul. În continuare vor fi prezentate câteva dintre echipamentele electronice uzuale.

#### **1.1.3.1. Elementele de execuție.**

Acestea sunt alese în concordanță cu tipul motoarelor utilizate. Ele sunt realizate cu tranzistoare bipolare, tranzistoare MOS sau circuite integrate și pot conține circuitele care generează semnalele de comandă corespunzătoare. Uneori semnalele de comandă sunt generate de către sistemul de conducere. Cerințele îndeplinite de către aceste echipamente sunt :

- ◆ fiabilitatea ridicată;
- ◆ consum de energie electrică redus;

- ◆ preț de cost scăzut.

Elementele de execuție sunt realizate sub formă modulară. Aceasta permite înlocuirea lor rapidă cu același tip de elemente sau cu alte tipuri.

### 1.1.3.2. Traductoarele.

Traductoarele utilizate în sistemul de acționare sunt traductoarele incrementale (TIRO). Ele servesc generării mărimilor de reacție necesare buclilor de reglare a vitezei și a poziției. Deoarece sunt utilizate în același timp pentru măsurarea distanțelor parcurse de către vehicul și a unghiurilor de virare, se poate considera că traductoarele incrementale fac parte și din echipamentul utilizat pentru determinarea situației.

### 1.1.3.3. Senzori utilizați pentru detectarea obstacolelor.

Senzorii utilizați pentru detectarea obstacolelor pot fi: ultrasonici, optici sau camere video. În foarte multe cazuri aceștia sunt utilizați și pentru determinarea situației vehiculului.

### 1.1.3.4. Echipamente utilizate pentru determinarea situației.

Complexitatea acestor echipamente este dependentă de natura aplicației [LUZ 95]. Criteriile care se iau în considerare pentru alegerea lor sunt :

- complexitatea rețelei de transport (numărul ramificațiilor, intersecții, stații);
- numărul vehiculelor care deserveșc traseul;
- lungimea traseului dintre stații;
- precizia de poziționare;
- frecvența modificărilor traseului;
- starea pardoselii;
- condițiile de mediu (curățenie, temperatură, umiditate, etc.).

**Determinarea situației vehiculului constă în determinarea precisă a coordonatelor  $x,y$  (poziția) și a unghiului  $\gamma$  (orientarea vehiculului în plan). Coordonatele  $x,y$  sunt coordonatele punctului caracteristic al vehiculului, iar unghiul  $\gamma$  este unghiul format de către axa longitudinală a robocarului cu una dintre axele sistemului de coordonate în raport cu care se determină situația. Aceste mărimi pot fi determinate în două moduri:**

- primul procedeu utilizează integrarea unor mărimi pentru determinarea coordonatelor și a unghiului (acelerație, viteză, etc.). Integrarea are loc începând cu un punct inițial (origine) în intervale mici de timp. Mărimile care se integrează sunt prelevate cu ajutorul unor senzori. Erorile de măsurare a acestor mărimi sunt destul de mari datorită, de exemplu, alunecării roților sau a modificării diametrului acestora. Reducerea erorilor datorate acestor cauze se face prin utilizarea unor roți speciale de măsură, care au alunecare mai redusă și diametrul constant. Senzorii utilizați pentru determinarea situației sunt prezentați sintetic în figura 1.5.

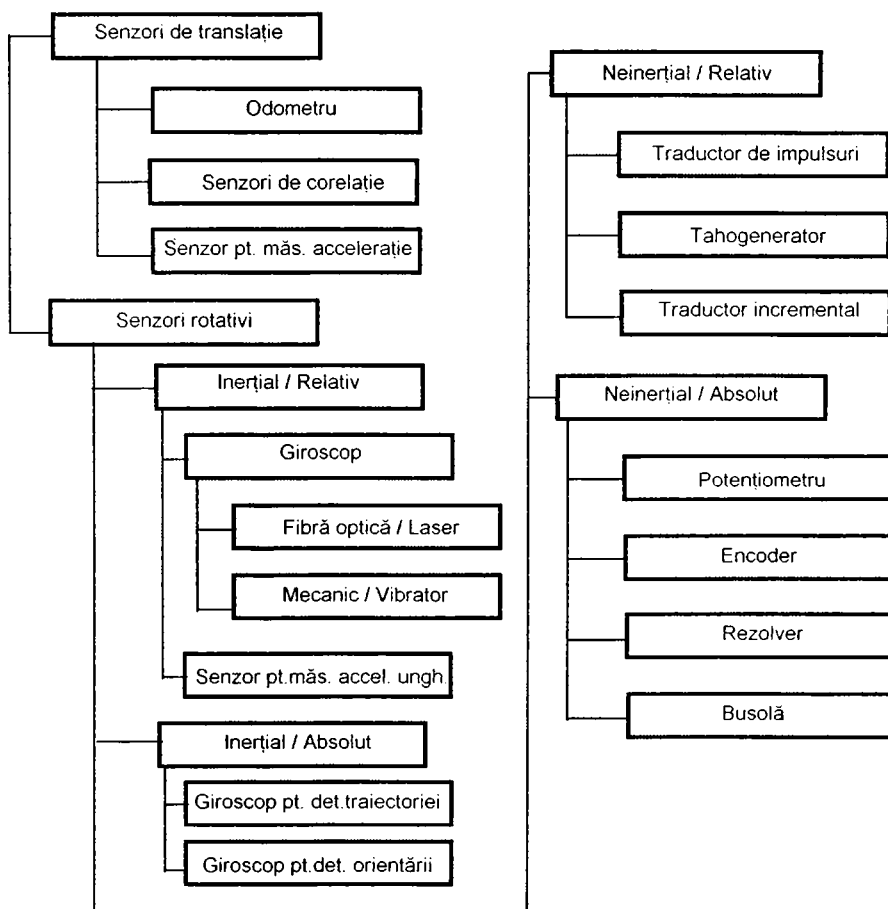


Figura 1.5.

- al doilea procedeu utilizează pentru determinarea poziției și a orientării vehiculului marcaje sau repere (balize) ale căror poziție în spațiu este cunoscută în momentul efectuării măsurării.

Marcajele sunt artificiale (special construite pentru măsurare) sau naturale (obiecte fixe existente în încăpere). Densitatea marcajelor este dependentă de modul în care se face detectarea acestora, precizia cerută pentru determinarea parametrilor de mișcare și precizia determinării poziției. În figurile 1.6, 1.7 se prezintă tipurile de marcaje folosite de către vehiculele care utilizează ghidare virtuală (traseu nematerializat) și ghidare după traseu materializat (fir, linie, etc.) [VDI 2510].

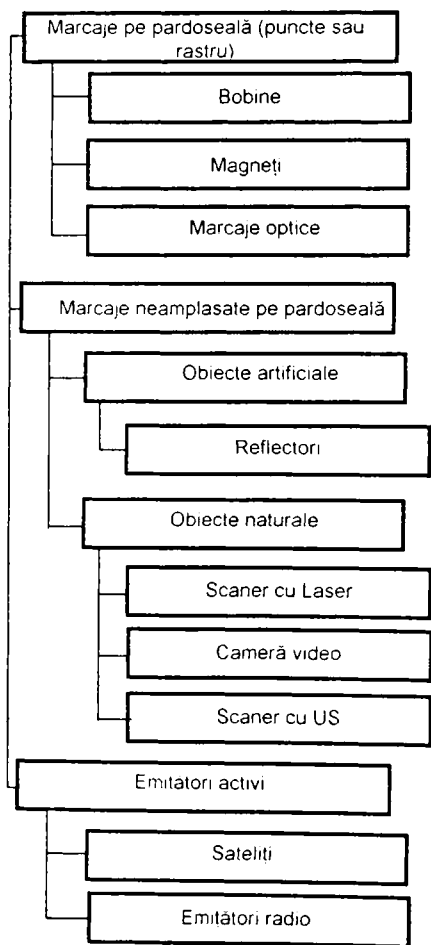


Figura 1.6.

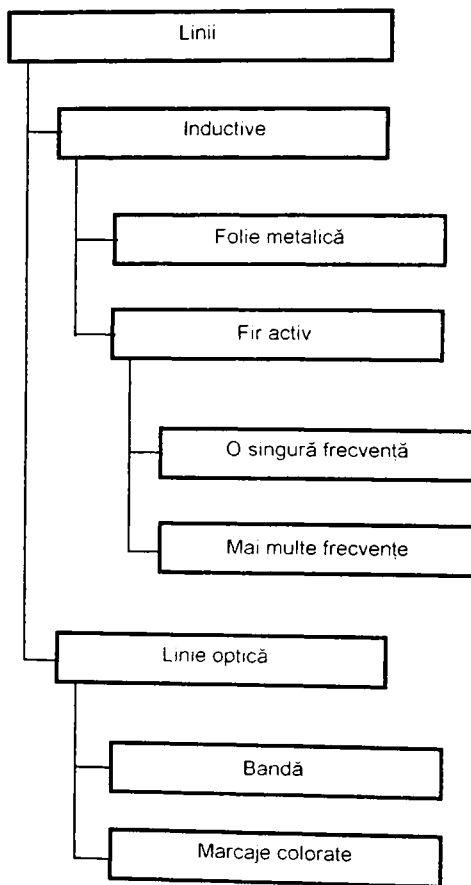


Figura 1.7.

Localizarea vehiculului este necesară pentru ca, prin datele furnizate de echipamentul pentru determinarea poziției să se recunoască locul în care se află vehiculul. Punctele ce trebuiesc localizate sunt: ramificații, intersecții, stații (de lucru, de încărcare - descărcare, de încărcare a acumulatorilor sau de service).

În fig. 1.8 se prezintă tipurile de echipamente uzuale folosite pentru localizare.

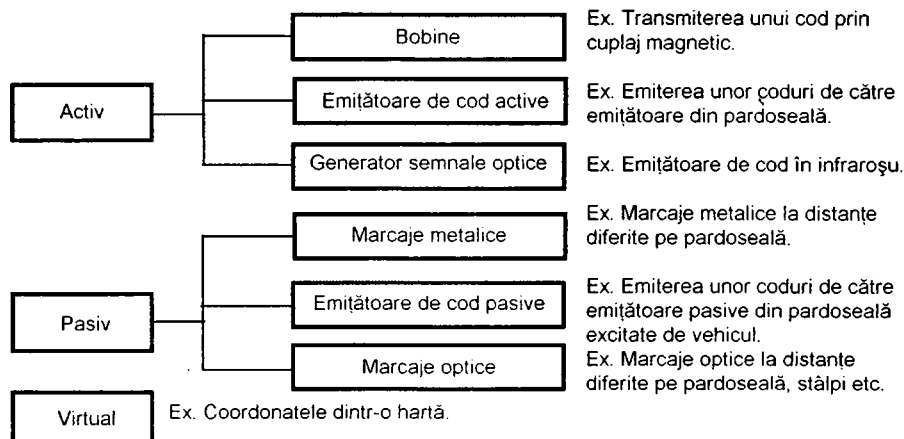


Figura 1.8.

În tabelul 1.2. se prezintă trei procedee clasice de ghidare și se descriu criteriile pentru alegerea acestora.

Tabelul 1.2.

Ghidare cu fir	Ghidare magnetică	Navigație cu laser
<b>Criterii pentru alegerea sistemului de ghidare</b>		
<ul style="list-style-type: none"> <li>• Sunt necesare utilaje pentru săpat șanțuri în pardoseală</li> <li>• Dacă se parcurg porțiuni scurte, cu puține ramificații de către multe vehicule</li> <li>• Dacă nu este necesară flexibilitate ridicată</li> </ul>	<ul style="list-style-type: none"> <li>• Sunt necesare utilaje pentru amplasarea în pardoseală a unor repere magnetice</li> <li>• Dacă se parcurg porțiuni lungi de către puține vehicule</li> <li>• Dacă este necesară flexibilitate mare</li> <li>• Dacă în pardoseală sunt cabluri, grilaje sau plăci metalice</li> <li>• Dacă nu se pot monta marcaje pe pereți</li> </ul>	<ul style="list-style-type: none"> <li>• Dacă nu se pot monta elemente de ghidare în pardoseală</li> <li>• Dacă se pot monta marcaje pe pereți</li> <li>• Dacă este necesară flexibilitate ridicată</li> <li>• Dacă în pardoseală sunt elemente magnetice</li> </ul>

### 1.1.3.5. Echipamente pentru transmiterea datelor.

Legătura între echipamentele fixe și cele mobile are loc prin sistemul pentru transmiterea datelor [VDI 3641]. Pentru a exploata la maximum mobilitatea vehiculelor, transmiterea datelor se bazează pe sisteme de emisie - recepție fără fir. Schimbul de informații are loc între:

- sistemul de conducere fix și vehicul;
- stațiile fixe și vehicul;
- vehicul - vehicul;
- sistemul de conducere și stațiile fixe.

Dintre datele și comenzile care se transmit se amintesc:

- comenzi pentru deplasarea vehiculului;
- coordonatele punctelor finale (ținte), spre care are loc deplasarea robocarului;
- semnale de comandă pentru conducerea vehiculului pe traiectorie;
- semnale pentru sincronizarea sistemelor de încărcare - descărcare a sarcinii de pe vehicul și din stațiile fixe;
- numărul vehiculelor;
- poziția vehiculelor;
- starea vehiculului (încărcarea, avarie);
- starea acumulatorilor;
- informații despre defecțiunile sistemului de transport;
- date referitoare la procesul de producție;
- semnale referitoare la siguranța circulației;
- semnale spre alte periferice (porți, semafoare).

Pentru asigurarea unei funcționări corecte a sistemului de transport automat este necesară stabilirea unor priorități pentru transmiterea datelor și luarea unor măsuri pentru corecția erorilor.

Transmiterea datelor se realizează prin:

- sisteme inductive;
- sisteme radio;
- sisteme cu raze infraroșii;

După distanța acoperită se disting:

- sisteme punctuale
  - inductiv;
  - infraroșu;



- sisteme care acoperă un traseu - inductiv;
  - infraroșu directiv;
- sisteme care acoperă suprafețe - radio;
  - infraroșu omnidirecțional.

**Sistemele de transmitere inductive** utilizează câmpuri electromagnetice cu frecvența de  $20 \div 100$  KHz, cărora le corespund rate de transfer de  $600 \div 4800$  baud. Pentru ca sistemul să fie punctual, transmiterea datelor se poate face în puncte discrete, aflate pe traseul vehiculului, prin bobine.

Sistemele de transmisie pe traseu se bazează pe existența firului de ghidare. Distanța de transmisie este redusă (0,5 m).

**Sistemele de transmitere radio** utilizează frecvențe în benzile 433,050  $\div$  434,790 MHz (banda de 70 cm) și 2,4  $\div$  2,5 GHz [VDI 3641].

**Sistemele de transmitere cu radiație infraroșie (IR)** ating rate de transfer de 115 Kbaud. Sunt capabile de transmiteri punctuale, pe traseu sau pe suprafață. Pentru efectuarea legăturii este necesară vizibilitatea directă între emițător și receptor sau existența reflexiilor. Distanțele acoperite pot fi de ordinul sutelor de metri. Sistemul este perturbat numai de către radiațiile cu conținut mare de IR (lumină solară, bariere optice, telecomenzi, etc).

### 1.1.3.6. Elemente de comandă și afișare.

În anumite situații este necesară intervenția operatorului uman și comandarea manuală a vehiculului. Acest lucru se realizează utilizând un panou de comandă manuală amplasat pe vehicul sau detașat de acesta, legătura cu vehiculul fiind realizată prin radio sau prin cablu. Nu este exclusă nici posibilitatea transmiterii comenzilor prin intermediul tastaturii calculatorului central care se află în legătură cu vehiculul.

Panoul de comandă, amplasat pe vehicul, conține, pe lângă semnalizările acustice și optice, tastatura necesară introducerii comenzilor de către operator și afișajul [VDI 4451]. Comenzile manuale se referă la:

- utilizarea vehiculului;
- deplasare și oprire;
- încărcare / descărcare.

Afișajul prezintă informații referitoare la:

624.126/101

- starea vehiculului;
- defecțiunile intervenite.

Vehiculele complexe pot fi echipate cu tastatură și monitor. Panoul de comandă manuală detașabil se poate conecta la vehicul prin cablu, comunicație radio și radiație infraroșie. El permite operatorului executarea unor operații manuale (deplasare, viraj, pornit, oprit, încărcare, descărcare) de la distanță simultan cu deplasarea vehiculului.

Dacă vehiculul comunică cu un calculator central și este echipat cu cameră video, comenzile manuale se pot transmite de la calculatorul central, urmărindu-se efectuarea acestora pe un monitor special sau chiar pe monitorul calculatorului. În cazul în care **AGV**-ul nu este echipat cu cameră video, dar se cunoaște în fiecare moment poziția și orientarea sa în spațiu, este posibilă executarea anumitor comenzi transmise de la distanță, fără ca vehiculul să se afle în câmpul de vizibilitate al operatorului uman, utilizând un program de afișare grafică a traiectoriei prescrise și a celei parcurse, a obstacolelor și a vehiculului.

#### 1.1.4. Sistemul de conducere.

În figura 1.9 se prezintă o variantă a unui sistem de conducere al **VGA**. Flexibilitatea vehiculului este dictată în primul rând de modul de realizare a sistemului de conducere. Sistemul de conducere **deschis** (cunoscut utilizatorului), oferă mai multe avantaje beneficiarilor acestor aplicații, constructorilor de sisteme de transport și constructorilor de componente.

O structură de conducere se definește ca **deschisă** (cunoscută) dacă este construită modular atât din punct de vedere hard cât și soft și dacă utilizează module standardizate. Aceste module sunt complet cunoscute de către utilizatori [VDI 4451], [VDI 2510].

Managerul abordează funcțiile comunicație, deplasare - preluare sarcină, management energetic și securitate, transmite comenzi și achiziționează informații de stare de la aceste blocuri. Prin funcția **comunicație** se primesc informații de la modulul de orientare și ghidare (de la nivelul **administrativ**) și se revine cu comenzile corespunzătoare.

Complexitatea sistemului de conducere este în concordanță cu modurile de operare ale vehiculului, și anume:

- automat;
- semiautomat;

- manual;
- diagnoză și service;
- instruire.

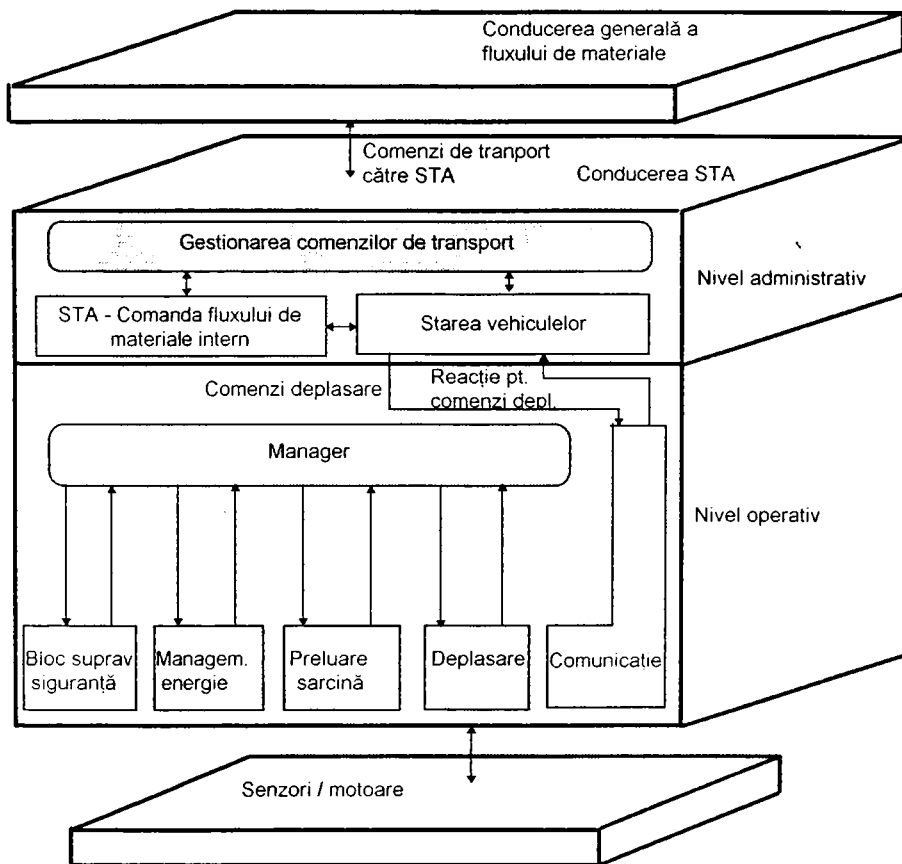


Figura 1.9.

Structurile hard întâlnite prezintă diverse arhitecturi ale părții de calcul și anume:

- o singură placă - soluție cu costuri reduse;
- sisteme de conducere cu programe memorate;
- mai multe plăci (interconectate serial sau paralel);
- prevăzute cu rețele pentru transmisii de date.

Conducerea automată este caracteristica principală a unui **VGA [AND 94]**. Automatizarea are loc pe două niveluri (figura 1.9):

1. Nivelul administrativ, care cuprinde:
  - flux de materiale în interiorul sistemului de transport automat (STA);
  - dispunerea vehiculelor.
  
2. Nivelul operativ, care cuprinde:
  - deplasarea;
  - măsuri de siguranță;
  - management energetic;
  - preluarea sarcinii;
  - comunicația.

Multitudinea aplicațiilor sistemelor de transport automat impune o multitudine a variantelor sistemelor de conducere [AND 95]. Tendința modularizării și a cunoașterii amănunțite a structurii vehiculului este din ce în ce mai evidentă.

## 1.2. Infrastructura sistemului de transport automat

Infrastructura sistemului de transport este constituită din totalitatea construcțiilor și instalațiilor care contribuie la desfășurarea în condiții corespunzătoare a operațiilor de transport [VDI 2199]. Se poate considera că din infrastructură fac parte: căile de rulare, porțile de trecere, lifturile, etc.

### 1.2.1. Calea de rulare.

Calea de rulare reprezintă suportul fizic pe care are loc deplasarea robocarului.

Pentru obținerea unei funcționări cât mai îndelungate a sistemului de transport, sigure și fără defecțiuni, este necesară respectarea anumitor cerințe referitoare la calitatea materialului din care este realizată calea de rulare, cum ar fi [DIN 15185] :

- rezistența la apăsare. Apăsarea exercitată de către vehicul, după natura sarcinii transportate, poate atinge  $2 \div 10 \text{ N/mm}^2$ . Pe de altă parte, vehiculul se deplasează de obicei pe același traseu, ceea ce cauzează o uzură mai mare a căii de rulare. Din acest motiv se impune supradimensionarea rezistenței la apăsare a pardoselii;

- în ceea ce privește coeficientul de frecare, se impune o valoare  $\mu \geq 0,6$ , pentru a avea o frânare eficientă în caz de forță majoră;
- menținerea în stare curată a căii de rulare este importantă, deoarece petele de ulei, umezeală, praf, etc. pot produce o funcționare defectuoasă sau pot reduce siguranța circulației;
- denivelările acceptate sunt de ordinul de mărime 15 mm pentru o distanță de 15 m;
- pantele admise depind de tipul vehiculului și se stabilesc împreună cu constructorul acestuia;
- proprietățile electrice determină gradul de încărcare electrostatică a vehiculului.

Lățimea minimă a căilor de rulare ale vehiculelor este determinată de dimensiunile vehiculului și/sau de dimensiunile sarcinii transportate.

De obicei se alege [ prEN 1525 ]:

- ◆ La circulație într-un singur sens: 1 metru + lățimea vehiculului (0.5m + l + 0.5m);
- ◆ La circulație în ambele sensuri: 1.4 metri + 2 x lățimea vehiculului (0.5m + l + 0.4m + l + 0.5m).

Dacă nu se poate realiza distanța de gardă de 0.5 metri, se vor lua măsuri suplimentare de siguranță [ DIN 51953 ] .

### 1.2.2. Porți de trecere.

Pe traiectoria pe care se deplasează robocarul pot exista porți de trecere, atunci când aceasta se extinde în încăperi alăturate sau atunci când transportul are loc în interior și în exterior [DIN 18225]. Dacă trecerea are loc prin porți, acestea trebuie să nu fie închise sau să nu fie blocate de către alte vehicule. Sistemul de conducere al vehiculului și sistemul de comandă al porților comunică între ele, pentru a se stabili când și cum are loc deschiderea porților și trecerea robocarului.

### 1.2.3. Lifturi.

Pe traseele parcurse de către robocare pot avea loc schimbări de nivel. În aceste condiții trebuie să existe mijloace de transport pe verticală, sub forma unor lifturi, sau calea de rulare trebuie să fie sub formă de rampă.

Mijloacele de transport sau construcțiile realizate sunt alese de către beneficiar și sunt dimensionate de către constructorul sistemului de transport.

### **1.3. Echipamente și instalații periferice.**

Echipamentele și instalațiile aflate în mediul de lucru al robocarului, cu care acesta se află în relații nemijlocite și care contribuie la desfășurarea optimă a procesului de transport sunt următoarele:

- echipamentele staționare de încărcare - descărcare (transfer) a sarcinii transportate;
- echipamentele pentru transmiterea datelor;
- instalațiile de siguranță;
- marcajele și reflectorii;
- stațiile și instalațiile pentru încărcarea acumulatorilor.

#### **1.3.1. Echipamente staționare pentru transfer de sarcină.**

Transportul materialelor începe sau/și este urmat de operații de încărcare, respectiv descărcare. Aceste operații se realizează în stațiile de lucru cu echipamente speciale. Tipul echipamentelor de încărcare - descărcare, cât și cel al stațiilor de preluare a sarcinii depinde de felul sarcinii transportate. Stațiile pot fi:

- stații de transfer pasive - fără echipamente proprii de încărcare descărcare;
- stații de transfer active - cu echipamente proprii de încărcare descărcare.

Coordonarea operației de transfer se face prin comunicare directă între vehicul și stație. [ VDI 2513].

#### **1.3.2. Echipamente pentru transmiterea datelor.**

Aceste echipamente sunt identice cu cele aflate pe robocar, iar caracteristicile lor au fost prezentate în paragraful 1.1.3.5.

#### **1.3.3. Instalații de siguranță.**

Căile de acces pot fi utilizate și de către alți participanți la procesul de fabricație și la trafic. De aceea, în afara sistemelor de protecție aflate pe robocar, se iau măsuri suplimentare de siguranță și anume [prEN 1526]:

- ⇒ instalarea unor oglinzi parabolice în intersecțiile fără vizibilitate;
- ⇒ instalarea semafoarelor, în general comandate cu prioritate de către **VGA**, pentru trecerea acestuia fără oprire;
- ⇒ instalarea unor lămpi de avertizare pentru anunțarea apropierii unui vehicul;
- ⇒ delimitarea traseelor prin marcaje cu vopsea sau benzi din folie de material plastic.

#### 1.3.4. Marcaje, reflectori.

Dacă deplasarea roboarelor are loc în medii în care nu există suficiente repere naturale pentru ca vehiculul să se poată **orienta**, este necesară instalarea unor repere artificiale, față de care se determină situarea vehiculului. Reperetele se realizează sub forma unor marcaje sau reflectori.

Marcajele pot fi atât optice (vopsea), cât și magnetice (magneți introduși în pardoseală).

Reflectorii se amplasează în spațiul de lucru în așa fel încât să poată fi detectați de către vehicul cu un scanner laser. Prin detectarea permanentă a acestora se poate determina poziția vehiculului. Dimensiunile reflectorilor sunt determinate de posibilitatea de detectare a lor cu ajutorul senzorilor.

#### 1.3.5. Stații și instalații pentru încărcarea acumulateorilor.

În funcție de tipul acumulateorilor necesare vehiculului se alege tipul echipamentului de încărcare, astfel încât să se realizeze timpii de încărcare doriți. Sursele necesare încărcării sunt comandate de către microcontrolere, realizându-se controlul încărcării bateriei și adaptarea la tipul și capacitatea acestora [**AND 94**].

### 1.4. Concluzii.

Vehiculele ghidate automat sunt utilizate în numeroase aplicații. De la explorarea spațiului cosmic la cercetarea mediului subacvatic, de la transportul uzinal la ajutorul gospodăresc, din sfera serviciilor la agrement, roboții mobili pot fi întâlniți tot mai des. S-a arătat că între categoria roboților mobili și cea a vehiculelor ghidate automat nu este decât un pas. De multe ori cele două domenii se confundă. În aplicațiile

industriale însă se vorbește de obicei despre vehicule ghidate automat. Acestea se impun în domeniul transportului, oferind o serie de avantaje, dintre care se amintesc:

- ⇒ flux de materiale și informații organizat;
- ⇒ reducerea erorilor referitoare la operațiile de transport.

Au fost amintite doar aceste două avantaje pentru că ele sunt caracteristicile principale ale întregului sistem de producție automatizat.

Deoarece există un sistem de conducere al întregului proces de fabricație, robocarul, încadrat în acest proces, execută cu precizie sarcinile care îi revin. Structura vehiculului poate fi complexă, dar în funcție de aplicația în care este utilizat, structura acestuia se poate simplifica.

Pe baza documentației studiate și a experienței acumulate, autorul descrie un robocar care poate fi utilizat în transportul din mediul industrial.

Mediul industrial oferă anumite avantaje implementării sistemului de transport automat, cum ar fi:

- ⇒ spații de lucru închise, în care nu au loc variații însemnate de temperatură și umiditate;
- ⇒ un mediu de lucru structurat, nefiind necesară ridicarea prealabilă a unei hărți;
- ⇒ existența unui sistem de conducere a procesului, în care trebuie implementată funcția **transport**;
- ⇒ posibilitatea controlului permanent al acțiunilor vehiculului de către operatorul uman;
- ⇒ calitate bună a căii de rulare.

Pe lângă aceste avantaje, există și dezavantaje, ca de exemplu:

- ⇒ posibilitatea murdării sau deteriorării instantanee a căii de rulare;
- ⇒ modificarea necontrolată a structurii mediului;
- ⇒ existența perturbațiilor de natură electromagnetică, luminoasă.

Proiectantul sistemului trebuie să profite de avantajele oferite, dar și să evite neajunsurile care apar datorită dezavantajelor din mediul industrial. Din punctul de vedere al sistemului de rulare, se propune alegerea soluției care oferă în primul rând siguranță în exploatare și fiabilitate



maximă. Sistemul de rulare cu trei roți este o soluție simplă, dar robocarul este instabil, mai ales atunci când are loc transferul sarcinii. De aceea este indicată alegerea sistemului de rulare cu patru roți, cu mențiunea că, în acest caz este necesară suspensia elastică.

Pentru sistemul de virare se preferă varianta care are toate roțile directoare, cu ajutorul căreia se obțin raze minime de virare.

Autorul preferă pentru tracțiune utilizarea servomotoarelor de curent continuu datorită ușurinței cu care acestea se comandă, iar pentru virare, utilizarea motoarelor pas cu pas, cu ajutorul cărora se realizează poziționarea precisă, fără a fi folosite traductoarele de poziție.

Sistemul senzorial pentru determinarea situației trebuie să funcționeze în timp real. Camera video, împreună cu un hard și un soft adecvat poate fi ușor utilizată atât pentru determinarea situației, cât și pentru detectarea obstacolelor.

Echipamentele pentru transmiterea datelor prin unde radio au acoperire mare, iar datorită informațiilor codate transmise sunt imune la perturbații.

Pentru sistemul de conducere autorul recomandă un sistem de calcul performant, care poate prelua cât mai multe funcții de conducere, oferind un grad ridicat de autonomie și inteligență robocarului.

Sinteza noțiunilor prezentate pe parcursul acestui capitol, cât și propunerile pentru alegerea soluțiilor constructive optime sunt rezultatul studiului unui însemnat material bibliografic și al experienței acumulate de către autor în domeniul roboticii și în special în cel al **VGA**.

De asemenea, documentarea efectuată la **I.P.A. Stuttgart**, unde funcționează un colectiv de cercetare cu preocupări în domeniul vehiculelor ghidate automat, discuțiile purtate cu persoane competente și studiile efectuate asupra unor robocare, au făcut ca imaginea autorului despre acest domeniu să fie clar conturată. Pe parcursul capitolului se fac și anumite comentarii personale legate de problemele cu care se confruntă evoluția vehiculele ghidate automat.

Se dorește ca acest capitol, prin informațiile pe care le aduce și prin problemele pe care le ridică, să deschidă noi perspective cercetării în diverse direcții ale acestui domeniu, atât pentru autor cât și pentru cei care abordează tema **robocare**.

## Capitolul 2

### CONDUCEREA VEHICULELOR GHIDATE AUTOMAT.

Scopul capitolului de față este sistematizarea noțiunilor referitoare la sistemele de conducere a vehiculelor ghidate automat, stabilirea funcțiilor pe care acestea le îndeplinesc și definirea unor termeni care nu sunt explicitați în literatura de specialitate.

Sistemul de conducere al **VGA** și sistemul de fabricație automat din care acesta face parte au o complexitate foarte ridicată, motiv pentru care înțelegerea rolului pe care anumite funcții ale sistemului de conducere al robocarului îl îndeplinesc este superficială sau chiar imposibilă în absența unei prezentări sistematizate.

Studiul amănunțit al unor sisteme de conducere a robocarelor a dus la posibilitatea concentrării noțiunilor legate de conducere și stabilirea unei ierarhii în ceea ce privește nivelurile acestora [AND 94], [AND 95]. Totodată se poate defini cu exactitate poziția sistemului de conducere nemijlocită a robocarului în această ierarhie și este posibilă evidențierea importanței sale în procesul de conducere.

#### 2.1. Considerații generale privind hipersistemul CIM.

Progresul tehnic pe calea automatizării, flexibilizării și integrării producției a condus la conceptul **CIM** - *Computer Integrated Manufacturing, fabricația integrată prin calculator*. Totalitatea activităților, umane și automate din cadrul unei întreprinderi, care concură la realizarea unui produs, este ierarhizată pe patru niveluri de activitate:

- I - conducerea strategică;
- II - compartimentele funcționale;
- III - compartimente de concepție;
- IV - compartimente de execuție

În concepția **CIM**, fiecărui nivel de activitate îi sunt afectate facilități **CIM**, adică sisteme funcționale cu o componentă hardware și software, având rolul de a îndeplini în mod automat activități aferente fiecărui nivel. Utilizarea calculatorului în toate activitățile pe care le implică realizarea unui produs într-o întreprindere, a permis integrarea acestora într-un singur complex de activități, care constituie **hipersistemul CIM**. În figura 2.1 este prezentată schema bloc a hipersistemului CIM [KOV 94a], [KOV 94b].

Facilitățile conceptului **CIM**, rezultate din figura 2.1 sunt următoarele:

- ⇒ asistarea de către calculator (calculatorul central) a conducerii strategice a întreprinderii, a luării deciziilor;
- ⇒ asistarea de către calculator a principalelor activități funcționale (aprovizionare, marketing, desfacere, financiar, contabil, service, personal, logistică externă, întreținere etc.);
- ⇒ automatizarea flexibilă a activității de planificare, programare și urmărire a producției folosind conducerea acestora cu calculatorul (facilitatea **CAPS**);

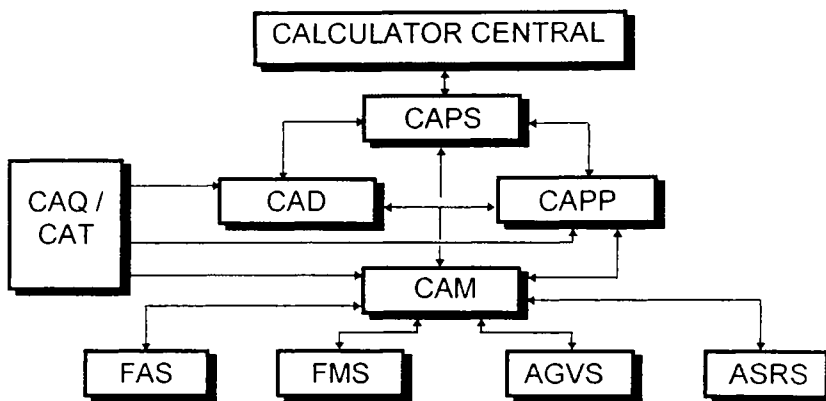


Figura 2.1.

- ⇒ automatizarea flexibilă a activității de concepție constructivă și tehnologică, a pregătirii fabricației prin conducerea acestora cu calculatorul (facilitățile **CAD, CAE, CAPP, CAM**);
- ⇒ automatizarea flexibilă a activităților de execuție (depozitare, transport intern, manipulare, fabricare, finisare, control - testare, sculărie etc.) prin conducerea cu calculatorul (facilitățile **AS/RS, AGVS, FMS, FAS, CAQ/CAT**);
- ⇒ integrarea tuturor activităților asistate de calculator și a facilităților **CIM**, într-un flux informațional unitar, prin rețeaua informațională **CIM**;
- ⇒ integrarea activităților de execuție printr-un flux de materiale unitar, depozitarea, regăsirea, transferul și manipularea lor executându-se în mod automat.

Situată la nivelul II de activitate, facilitatea **CAPS - Computer Aided Planning System, planificare, programare și controlul producției asistate de calculator**, este dispecerul automat al întregii activități productive.

**CAPS** conține bazele de date, care sunt transmise prin rețeaua informațională **CIM**, referitoare la concepția constructivă a produselor (obținute prin facilitatea **CAD** - *Computer Aided Design, proiectare asistată de calculator*), la procesele tehnologice de fabricație (obținute de la facilitatea **CAPP** - *Computer Aided Process Planning, proiectarea procesului tehnologic asistată de calculator*), starea de încărcare a depozitelor (obținute prin facilitatea **AS/RS** - *Automated Storage and Retrieval System, sistem automat de depozitare și regăsire*), circulația obiectelor prin sistemul logistic intern (obținute prin facilitatea **AGVS** - *Automated Guided Vehicle Systems, sisteme de vehicule ghidate automat*), încărcarea mijloacelor de producție, stadiul de execuție al diferitelor procese tehnologice (de la facilitățile **FMS** - *Flexible Manufacturing System, sistem de fabricație flexibilă* și **FAS** - *Flexible Assembly System, sistem de asamblare flexibilă*), starea și numărul pieselor și produselor finite (de la facilitățile **CAO(A)** - *Computer Aided Quality Assurance, asigurarea calității asistate de calculator* și **CAT** - *Computer Aided Testing, testarea performanțelor asistată de calculator*), comenzile de la beneficiari etc. Pe baza datelor menționate mai sus, **CAPS** elaborează *programul de producție principal*, **MPS** - *Master Production Schedule*, care prevede loturile de fabricație, succesiunea operațiilor de manipulare și prelucrare pentru fiecare lot, alocarea mijloacelor de producție necesare, încărcarea în timp a acestora etc. Prevederile **MPS** sunt transmise prin rețeaua informațională **CIM** facilităților de concepție și de execuție ale hipersistemului **CIM** (**CAD**, **CAPP**, **CAM**, **AS/RS**, **AGVS**, **FMS**, **FAS**, **CAQ/CAT**), sub forma unor comenzi start/stop a unor secvențe de program la anumite componente (de fapt sub forma unor ciclograme)[HAM 86a], [HAM 86b].

Un alt program elaborat de **CAPS** este *bonul de materiale*, **BOM** - *Bill of Materials*. Pe baza componentei ansamblului general (subansambluri, piese, componente, materiale, materii prime etc.), se stabilește, în conformitate cu **MPS**, fluxul necesar de intrare al materialelor, materiilor prime și componentelor în depozit, ținând seama de principiul **just in time**. Fluxul de intrare al materialelor este concretizat de către programul **MRP** - *Material Requirement Planning, planificarea necesarului de materiale*, care programează secvențe de achiziție a diferitelor cantități de materiale, materii prime și componente și secvențe de introducere a acestora în depozite, respectiv de scoatere și transferul lor de către **AGVS** [JON 87].

*Programul de evoluție în timp a producției neterminate*, **WPS** - *Work in Progress Schedule*, rezultă la rândul său din **MPS**.

---

Utilizând informațiile cuprinse în baza de date, cu privire la comenzi și informațiile codificate, din programele strategiei elaborate de conducerea întreprinderii (nivelul I de activitate), utilizatorul, asistat de facilitatea **CAPS**, întocmește intrările în lucru a diferitelor produse și succesiunea trecerii acestora prin fazele producției sub forma unor grafice **GANTT**. Pe baza acestora, tot **CAPS** întocmește programele - ciclograme pentru planificarea activității în timp a facilităților (**CAD**, **CAPP** și **CAM** - *Computer Aided Manufacturing, fabricația asistată de calculator*).

Celelalte compartimente ale întreprinderii (aprovizionare, desfacere, financiar, contabil, logistică externă, personal, mecanic șef etc.) își desfășoară activitatea asistată de calculator, fără a avea alocate alte facilități **CIM**, ele fiind racordate prin intermediul rețelei informaționale **CIM** la facilitățile acestuia (în special facilitatea **CAPS**).

În cadrul nivelului III de activitate - compartimentul de concepție - se realizează atât proiectarea constructivă a produselor, cât și proiectarea proceselor tehnologice pentru fabricarea lor. Cele două laturi ale procesului de proiectare, se realizează în mod simultan și se întrepătrund (ingineria *paralelă sau simultană*) [KOV 96b].

Proiectarea constructivă se realizează prin facilitatea **CAD**, în regim conversațional cu participarea utilizatorului. Acesta poate apela informațiile cu privire la date tehnice, respectiv metode de calcul ale sistemelor expert/baze de date **CAE** - *Computer Aided Engineering, inginerie asistată de calculator*.

Proiectarea tehnologică se realizează prin facilitatea **CAPP** de asemenea în regim conversațional. Utilizatorii pot apela informații cu privire la caracteristicile materialelor, ale proceselor tehnologice, regimurilor de așchiere etc. de la sistemul expert /baze de date **MES** - *Manufacturing Expert System, sistem expert de fabricație*.

Documentațiile elaborate de **CAD** și **CAPP**, desene, tabele, instrucțiuni, etc., se transmit celorlalte facilități ale hipersistemului **CIM**.

## 2.2. Integrarea sistemului de robocare într-un hipersistem CIM.

La nivelul IV de activitate - compartimentul de execuție - *fluxul informațional* de la nivelurile superioare se îmbină cu *fluxul de materiale* pentru a permite execuția produselor. Facilitatea **AS/RS** asigură

*depozitarea semifabricatelor*, a materiilor prime, a materialelor, a pieselor, a subansamblurilor și a componentelor necesare începerii procesului de fabricație, *depozitarea intermediară* a pieselor pe fluxul tehnologic de fabricație și *depozitarea produselor finite*.

În funcție de programul de fabricație întocmit de facilitatea **CAPS** și documentația constructivă și tehnologică elaborată de facilitățile **CAD**, respectiv **CAPP**, se emit comenzi de eliberare a conținutului depozitului de semifabricate și componente [AGE 86].

Robotul care deservește depozitul extrage obiectele /paletele în cauză din locașul depozitului și le pune la dispoziția **AGV**-ului, în vederea transportării acestora la sistemele de fabricație flexibilă. Depozitele intermediare primesc de la **AGV** semifabricatele / piesele prelucrate într-o anumită fază a procesului tehnologic de fabricație, în mod individual sau în palete /containere, le stochează și le pun la dispoziția facilităților **FMS** și **FAS**, în vederea unor prelucrări ulterioare, respectiv în vederea asamblării.

În transferul materiei prime, a materialelor, componentelor, pieselor și subansamblurilor se aplică principiul **just in time** - chiar la timp - ceea ce înseamnă că obiectele în cauză ajung la locul unde sunt utilizate, chiar în momentul necesității utilizării lor. Acest lucru presupune o corelare minuțioasă, fără erori, a activității tuturor factorilor implicați, începând cu serviciul de aprovizionare și terminând cu depozitele, respectiv logistica internă a întreprinderii. Această corelare este facilitată prin integrarea activității tuturor factorilor, care concură la realizarea asigurării materiale a procesului de fabricație prin calculator [KOV 86a].

Transferul obiectelor de lucru între depozitele automate - **AS/RS** și sistemele de fabricație flexibilă - **FMS**, inclusiv sistemele de asamblare flexibilă - **FAS**, controlul calității, testarea performanțelor asistată de calculator - **CAQ/CAT**, respectiv elementele componente ale acestor facilități (de exemplu: celule de fabricație, montaj etc.), se realizează prin **sistemul de transfer (transport) automat, în majoritatea cazurilor materializat prin facilitatea AGVS - Automated Guided Vehicle Systems, sistemul de vehicule ghidate automat**.

Conducerea sistemului de vehicule ghidate automat se realizează de către facilitatea **CAPS**. Ea comandă încărcarea /descărcarea vehiculelor ghidate automat în anumite posturi de lucru cu semifabricate, piese, subansambluri sau ansambluri separate sau aflate pe palete, traseele pe

---

care circulă, staționarea lor, încărcarea acumulatorilor la posturile de alimentare cu energie electrică. Schimbul de informații între calculatorul imbarcat pe vehiculul ghidat și calculatorul facilității **CAPS**, se realizează în posturi speciale de schimb de informații (puncte de comunicare) sau prin legături radio.

Informațiile privind datele constructive ale pieselor, respectiv cele privind tehnologia utilizată, se transmit mașinilor-unelte și dispozitivelor de lucru de la facilitățile **CAD** și **CAPP**, prin rețeaua informațională **CIM**, realizându-se pe această cale fabricația asistată de calculator - **CAM**. Intrarea în funcțiune /oprirea mașinilor-unelte și dispozitivelor de lucru, a dispozitivelor instalației aducătoare /de evacuare, a roboților industriali, a modulelor de fabricație flexibilă sunt comandate de sistemul de conducere al celulelor de fabricație flexibilă (Cell controllers). Corelarea activității celulelor de fabricație flexibilă cuprinse într-un sistem ierarhic superior, de exemplu linie de fabricație, se realizează de sistemul de conducere *local*. Corelarea tuturor facilităților **FMS** cu celelalte facilități **CIM** (de exemplu **AGVS**) se realizează de către facilitatea **CAPS** [KOV 96a].

### 2.3. Conducerea vehiculelor ghidate automat.

Odată cu promovarea sistemelor de transport automate, acestea au fost îmbunătățite continuu. Progrese s-au înregistrat și în ceea ce privește sistemele de conducere. În același timp conceptul realizării acestor instalații s-a modificat, având loc o trecere de la sisteme cu multe vehicule, la sisteme cu tot mai puține, numărul vehiculelor fiind redus până la zece [VOG 92], [MEH 91]. Concomitent crește lungimea traseelor pe care acestea se deplasează, cheltuielile de instalare pe vehicul devin tot mai mari. Aceste cheltuieli sunt determinate de cheltuielile de instalare și de exploatare a sistemului. Avantajul esențial este însă adaptarea sistemelor la gradul ridicat de automatizare a procesului și posibilitatea de configurare simplă pe care o oferă utilizatorului [ZEE 91], [HEL 91].

Vehiculele ghidate automat trebuie să se comporte în trafic ca și celelalte componente (vehicule cu conducător uman), adică în primul rând să ofere siguranță maximă transportului. Odată cu creșterea vitezei, se iau măsuri tot mai severe pentru evitarea coliziunilor, nemaifiind corespunzătoare sistemele de protecție mecanice (cu atingere).

---

Cerințele legate de transportul automatizat, prezentate pe parcursul tezei, sunt rezolvate de către **Sistemul de conducere**. O analiză a ierarhizării conducerii pune în evidență următoarele [PRI 92]:

- ◇ la baza modelului ierarhic se află conducerea mașinilor unelte sau a vehiculului (conducerea nemijlocită a vehiculului);
- ◇ conducerea vehiculului este un bloc funcțional al conducerii celulei de fabricație (sau a domeniului de activitate);
- ◇ conducerea celulei (domeniului de activitate) este un bloc ce face parte din conducerea procesului de fabricație;
- ◇ calculatorul de proces este cuprins în nivelul de planificare.

Pe baza acestui concept ierarhic, conducerea automată a procesului de fabricație devine o conducere descentralizată, cu posibilități bune de exploatare. Funcțiile aflate la fiecare nivel sunt astfel alese încât să se reducă pe cât posibil comunicația între elementele componente. Modelul de mai sus conține mai multe funcții de conducere pe fiecare nivel, iar apartenența acestora la diverse niveluri depinde de sarcinile și mărimea sistemului, de cerințele utilizatorului și de hard-ul utilizat.

Conducerea sistemului de transport poate fi structurată deci pe trei niveluri [VDI 2510]:

- ◇ conducerea și urmărirea procesului;
- ◇ conducerea zonei (domeniului) de activitate;
- ◇ conducerea vehiculului.

Aceste niveluri au corespondență în conducerea roboților (a mașinilor unelte) și anume:

- ◇ conducerea vehiculului - conducerea robotului;
- ◇ conducerea zonei - conducerea celulei de fabricație;
- ◇ conducerea și urmărirea procesului - conducerea și urmărirea procesului de fabricație.

O posibilă clasificare a funcțiilor pe domeniile de conducere este prezentată în figura 2.2.

**Sarcina conducerii și urmării Sistemului de Transport Automat (STA)** este gestionarea comenzilor de transport, care sunt primite de la sistemul de planificare (referitoare la starea sistemului sau la dispunerea mijloacelor de fabricație), sau sunt introduse manual de către operator. Generarea sarcinilor pe baza unor criterii de optimizare, ca de exemplu drumul cel mai scurt, deplasare minimă neîncărcat, asigurarea datelor, sunt de asemeni sarcini ale conducerii.



**Conducerea domeniului STA** preia atât comenzile de deplasare destinate vehiculelor, sincronizarea acestora, sincronizarea vehiculelor cu mijloacele fixe, cât și conducerea traficului (intersecții).

**Conducerea vehiculului** are sarcina de a prelucra comenzile de transport, de a urmări traseul și de a găsi destinația. În același timp este necesar să urmărească încărcarea-descărcarea și să ofere posibilitatea comenzii manuale sau a diagnozei.

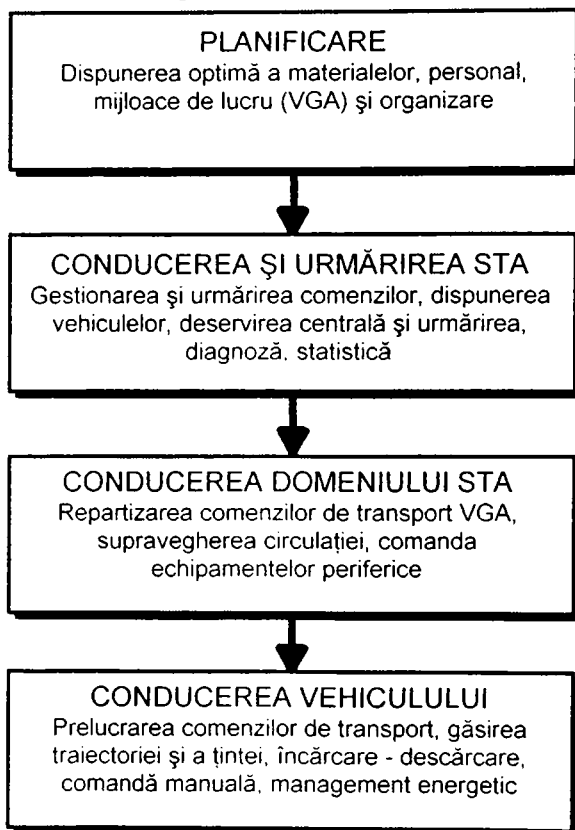


Figura 2.2.

Urmărind structura funcțiilor staționare de conducere, există posibilitatea reprezentării **conducerii centralizate** (figura 2.3.), la care funcțiile **conducere și urmărire**, precum și **conducerea domeniului** sunt concentrate într-un singur bloc [AND 94], [AND 95]. Într-o structură descentralizată de conducere (figura 2.4.), anumite funcții sunt preluate de către sistemele de conducere ale componentelor, care la rândul lor sunt legate prin rețeaua locală cu sistemul de conducere și urmărire.

Sistemele de conducere care se vor realiza în viitor trebuie să ofere posibilitatea transferării unor funcții de pe un nivel pe altul sau eliminarea completă a acestora, pentru o adaptare cât mai ușoară la cerințele utilizatorului.

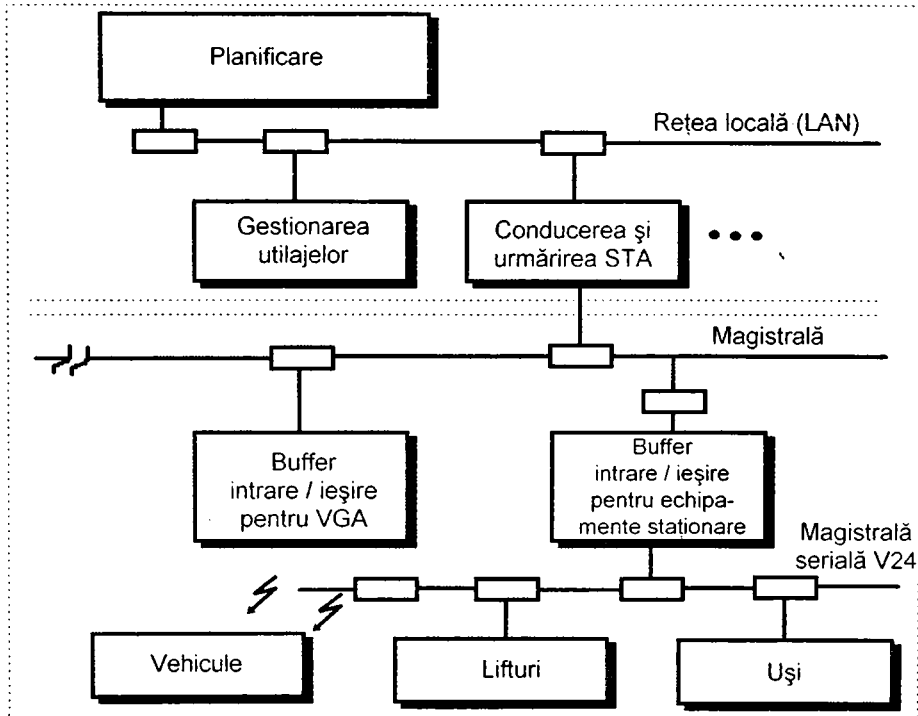


Figura 2.3.

Componentele sistemului de conducere a vehiculului sunt destinate:

- ◇ deservirii (tastatură, comandă manuală, programare);
- ◇ conducerii și urmăririi procesului de fabricație și conducerii domeniului de activitate;
- ◇ comandării altor dispozitive sau periferice.

Aceste componente sunt prezentate în figura 2.5.

Varianta structurală care realizează o deplasare bazată pe sisteme senzoriale cuprinde componentele **poziționare și orientare**, calculul mărimilor de conducere și sesizarea obstacolelor. Obținerea unei flexibilități ridicate necesită o programare optimă a comenzilor de deplasare, manevrare a sarcinilor și pentru alte operații.

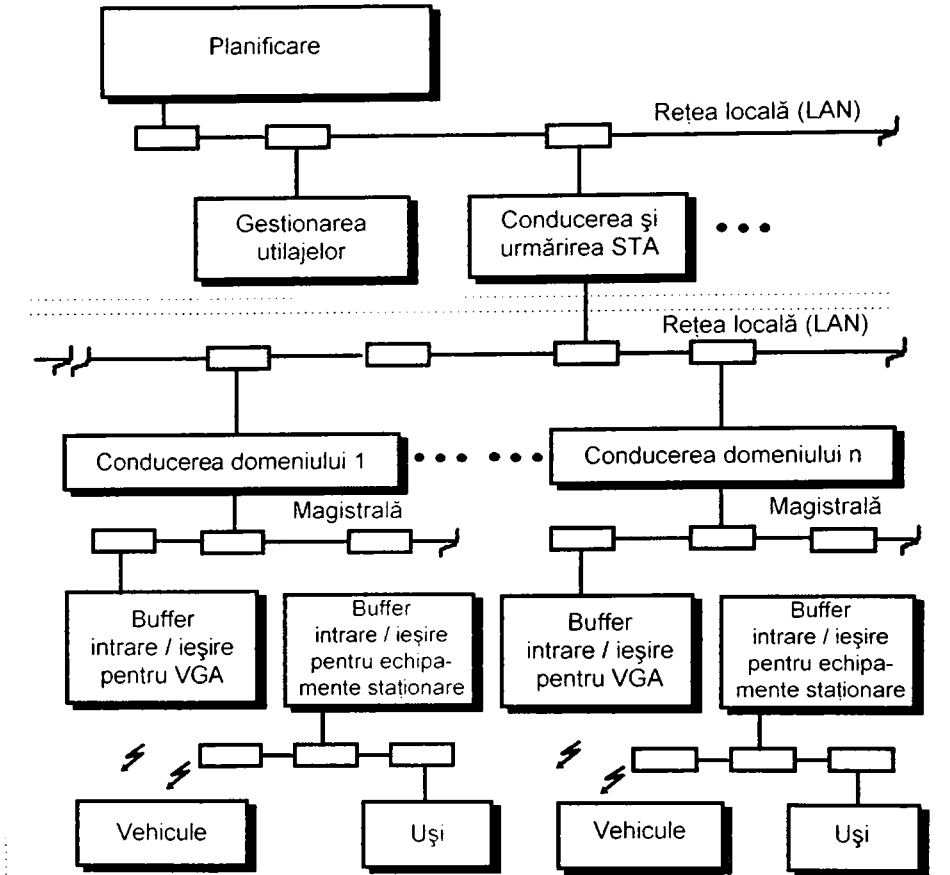


Figura 2.4.

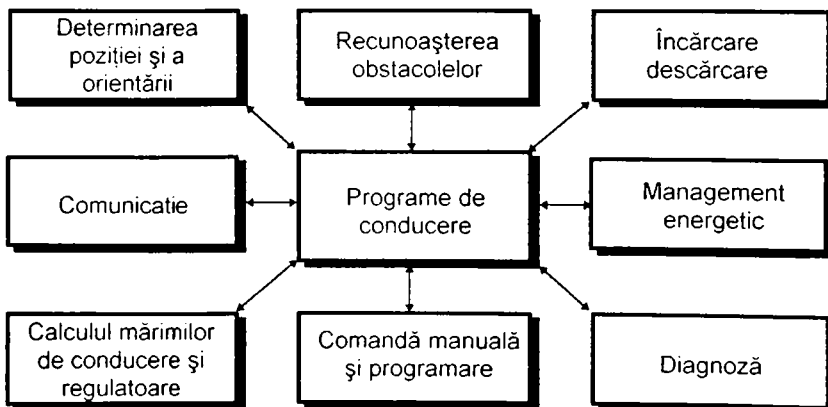


Figura 2.5.

## 2.4. Funcțiile îndeplinite de către sistemul de conducere al VGA.

Din cele prezentate în capitolul de față (figura 2.2), cât și din [AND 94], [HEL 96], [AND 95], rezultă funcțiile pe care trebuie să le realizeze sistemul de conducere al vehiculului. O reprezentare schematică a acestor funcții, **imaginată de către autor** este ilustrată în figura 2.6, iar semnificația lor este următoarea:

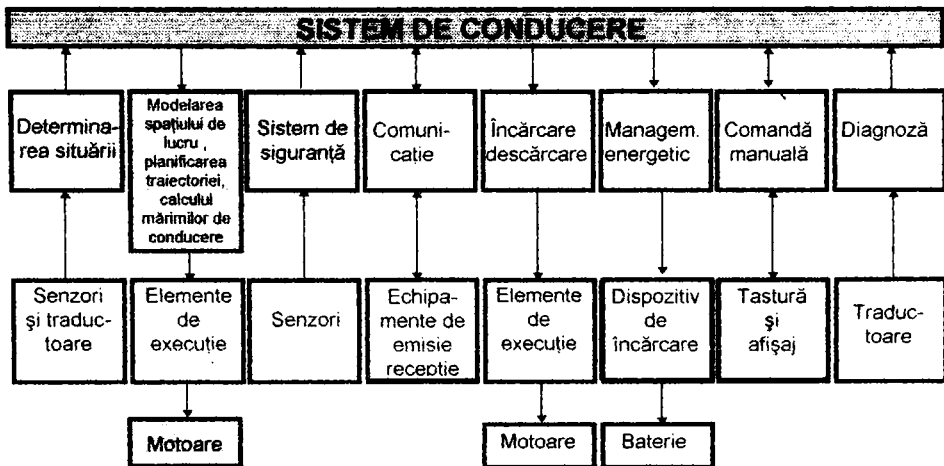


Figura 2.6.

⇒ **determinarea situației.** Pentru a realiza deplasarea robocarului în spațiul de lucru, sistemul de conducere al vehiculului trebuie să fie capabil să determine în fiecare moment situarea acestuia. Determinarea situației se bazează pe prelucrarea unor informații referitoare la coordonatele punctului în care se află robocarul la un moment dat și la orientarea axei acestuia, raportate la un sistem de referință fix. Informațiile sunt achiziționate de către sistemul de conducere cu ajutorul unor traductoare (TIRO) și a unor sisteme senzoriale, amplasate pe vehicul sau în spațiul de lucru.

⇒ **modelarea spațiului de lucru, planificarea traiectoriei și calculul mărimilor de conducere.** Coordonatele punctului inițial din care începe deplasarea vehiculului și coordonatele punctului final în care trebuie să ajungă vehiculul sunt cunoscute pe baza unui model al spațiului de lucru. Cu ajutorul funcției descrise anterior, situarea robocarului este cunoscută de asemenea în fiecare moment. În aceste condiții, sistemul de conducere poate

- planifica, pe baza anumitor criterii, traiectoria pe care are loc deplasarea robocarului. Traiectoria vehiculului fiind planificată, sistemul de conducere trebuie să genereze mărimile de conducere, cu ajutorul cărora se comandă motoarele prin intermediul elementelor de execuție.
- ⇒ **detectarea obstacolelor.** Prin procesarea informațiilor furnizate de către sistemele senzoriale, sistemul de conducere al robocarului stabilește dacă obiectele detectate sunt obstacole aflate pe traiectoria vehiculului, sau sunt elemente cunoscute ale mediului înconjurător. În cazul în care aceste obiecte sunt obstacole, sistemul de conducere decide tipul măsurilor de siguranță care se aplică pentru evitarea coliziunilor.
- ⇒ **comunicația.** Vehiculul automat se poate deplasa liber, dar face parte dintr-un sistem de transport, care la rândul său deservește un proces de fabricație. Din aceste motive trebuie să existe o comunicație fără fir între robocar și echipamentele și instalațiile care participă la procesul de fabricație. Sistemul de conducere este cel care gestionează și utilizează informațiile care se transmit prin sistemul de comunicație.
- ⇒ **conducerea procesului de transfer** a sarcinii transportate. Sarcina care este transportată de către vehicul este încărcată și descărcată cu ajutorul unui dispozitiv de încărcare - descărcare comandat de către sistemul de conducere, pe baza schimbului de informații cu sistemul de conducere ierarhic superior. Uneori acest schimb de informații poate avea loc cu sistemul de comandă al dispozitivului de transfer al stației de lucru în care se află robocarul.
- ⇒ **managementul energetic.** Bateriile de acumulare trebuie încărcate periodic. Controlul stării de încărcare a acestora se face de către sistemul de conducere, care decide momentul în care se efectuează încărcarea și durata acesteia.
- ⇒ comanda manuală. Uneori este necesară intervenția operatorului uman în procesul de transport. Aceasta are loc din următoarele motive :
- ◇ se efectuează o **instruire** a vehiculului;
  - ◇ deplasarea vehiculului are loc eronat;
  - ◇ este necesară efectuarea unor operații pe care robocarul nu le poate executa în mod automat.
- Operatorul realizează schimbul de informații cu sistemul de conducere al vehiculului prin intermediul unui panou de comandă manuală, echipat cu tastatură și afișaj.

⇒**diagnoza**. Dacă are loc defectarea unor echipamente ale robocarului, sistemul de conducere, presupus în stare de funcționare, stabilește care sunt defecțiunile survenite și echipamentele defecte. De asemenea este necesar ca informațiile privitoare la defecțiuni să fie transmise sistemului de conducere ierarhic superior sau operatorului uman, pentru a fi remediate.

Toate funcțiile descrise ale sistemului de conducere au importanță majoră pentru desfășurarea în condiții optime a transportului, dar unele dintre ele nu sunt implicate în mod direct în efectuarea deplasării pe traiectorie a vehiculului. **Primele trei funcții ale sistemului de conducere sunt activate în mod nemijlocit în scopul realizării mișcării impuse robocarului, în concordanță cu procesul de transport și de fabricație.**

## **2.5. Navigația - parte integrantă a procesului de conducere a robocarului.**

Deoarece în literatura de specialitate [HST 96], [HIUT 96] se întâlnește atât termenul **navigație** cât și **navigare**, **autorul consideră necesară explicitarea și definirea acestor termeni**. Semnificațiile lor sunt diferite.

Pentru ca termenii să fie utilizați corect, este necesară precizarea domeniilor la care aceștia se referă. Funcțiile sistemului de conducere:

- ⇒determinarea situației;
- ⇒modelarea spațiului de lucru, planificarea traiectoriei și calculul mărimilor de conducere;
- ⇒detectarea obstacolelor.

Împreună cu echipamentele corespunzătoare pot fi grupate într-un singur bloc, căruia i se atribuie denumirea de **sistem de navigație**.

Pe baza celor arătate mai sus se poate defini acest sistem: **sistemul de navigație este sistemul care cuprinde totalitatea funcțiilor realizate de către sistemul de conducere și totalitatea echipamentelor aferente acestora, activate pentru efectuarea deplasării pe traiectoria prescrisă.**

În cadrul analizei problemelor de bază ale planificării traiectoriei [DRĂ 95.], [LAT 91], [PRU 89], se arată că: asigurarea îndeplinirii sarcinii unui robot (robocar) se bazează pe trei acțiuni majore:

- ⇒stabilirea unei **strategii de navigare**;
- ⇒modelarea spațiului de lucru;
- ⇒planificarea mișcării în spațiul de lucru respectiv.

De foarte multe ori, acțiunea de planificare a mișcării în spațiul de lucru include modelarea acestuia și ca atare cele două acțiuni pot fi considerate una singură [DRĂ 95].

Stabilirea unei strategii de navigare se referă la:

- ⇒deplasarea dintr-un punct al spațiului de lucru într-un alt punct, ambele având poziții bine precizate; strategia constă în a genera o astfel de **funcție de comandă** încât să se atingă poziția dorită.
- ⇒baleierea întregii zone, care se poate realiza prin deplasarea în spirală, pornind de la exterior spre centru sau prin deplasarea în zig-zag.
- ⇒urmărirea unui alt mobil care se deplasează sau a conturilor unui obstacol aflat în spațiul de lucru.

În concluzie, **strategia de navigare definește modul în care are loc mișcarea, fără a face referire la funcțiile sistemului de conducere și la echipamentele și sistemele senzoriale.**

Stabilirea strategiei de navigare nu a fost inclusă în cadrul funcțiilor îndeplinite de către sistemul de conducere al vehiculului. Strategia de navigare este aleasă de către sistemele de conducere ierarhic superioare sau este impusă de către operatorul uman.

O altă noțiune care va fi analizată este **procedee de navigare**. Literatura de specialitate [BYAM 96], [EGAT 96], [HALK 96] prezintă numeroase procedee de navigare. În toate cazurile se analizează modul în care sistemul de conducere realizează urmărirea traiectoriei planificate. De asemenea se analizează tipurile de sisteme senzoriale care furnizează sistemului de conducere informații referitoare la situația vehiculului.

Se observă că aceleași procedee de navigare pot fi utilizate în cadrul oricărei strategii de navigare. Din acest motiv **procedee de navigare** definesc o noțiune distinctă în raport cu **strategia de navigare și navigația**.

În continuare vor fi prezentate câteva procedee de navigare și se vor arăta contribuțiile aduse de către autor la dezvoltarea și optimizarea acestora.

### 2.5.1. Procedee de navigare.

Se va începe analiza procedeeilor de navigare reprezentate sintetic în figura 2.7. cu cele din coloana 3.

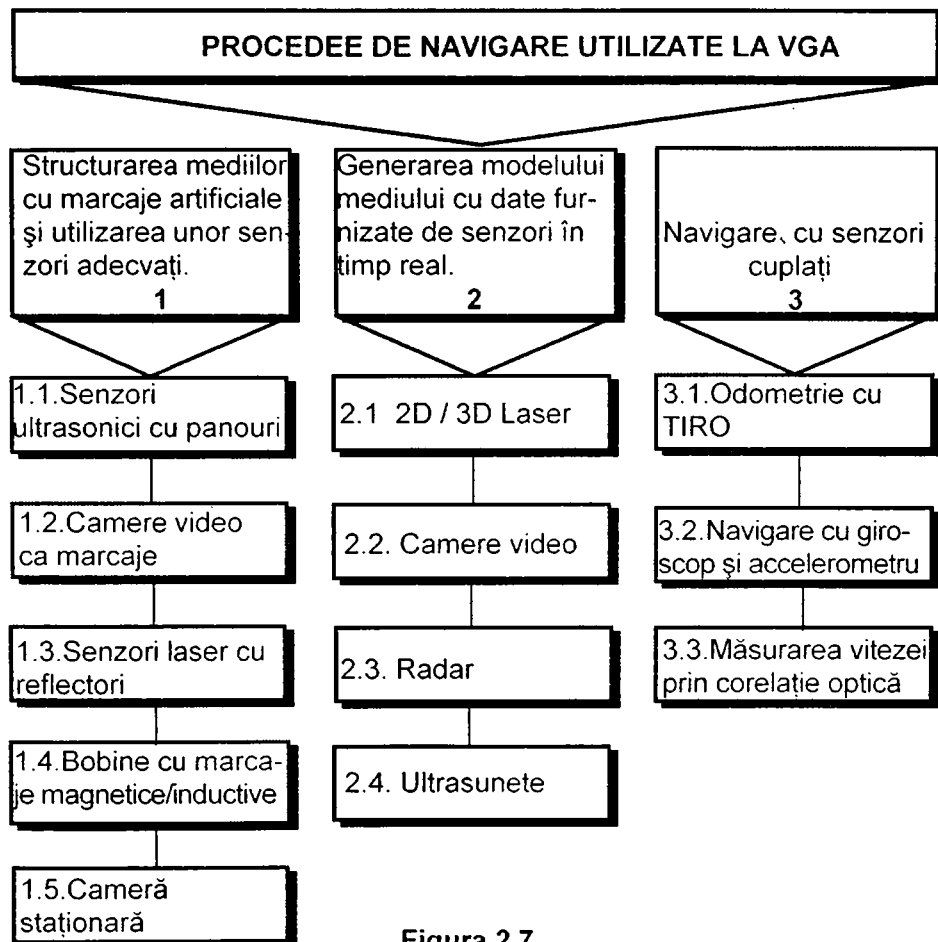


Figura 2.7.

Din această categorie, cea mai răspândită metodă este odometria, adică determinarea situației prin măsurarea, cu ajutorul traductoarelor incrementale, a deplasării și a unghiurilor de virare. În funcție de roțile a căror deplasare se măsoară, de starea solului și de încărcare, nu se pot obține precizii ridicate de măsură [KOB 92], [JAN 90]. O mărire a preciziei de măsură a poziției, și în special a orientării, se obține prin utilizarea giroscopului. Se mai pot folosi și accelerometre pentru determinarea vitezei de translație. Prin dubla integrare a accelerației



---

crește eroarea, ceea ce face ca acești senzori, preluați din navigația aeriană și spațială să nu aibă o răspândire foarte largă.

O sporire a preciziei de determinare a situației are loc prin măsurarea prin corelație optică a vitezei. Un senzor CCD urmărește suprafața căii de rulare și, prin sesizarea denivelărilor (asperităților) acesteia și prelucrarea prin metode de corelație a datelor obținute, se poate calcula viteza de deplasare. Metoda nu este aplicabilă la o iluminare scăzută [KOB 92].

Mărirea preciziei de determinare a poziției și a orientării este necesară deoarece, dacă la deplasarea spre țintă sunt admise erori de urmărire a traiectoriei de câțiva centimetri, la poziționarea pentru încărcare - descărcare sau pentru alte operații, eroarea admisă este de ordinul milimetrilor. Toate procedeele descrise mai sus conduc la **determinarea situației relative** a vehiculului (localizarea relativă).

Metodelor de mai sus li se adaugă, pentru mărirea preciziei metoda de detectare discretă, respectiv cvasicontinuă a unor marcaje fixe (coloana1), ce duce la **determinarea situației absolute** a robocarului (localizare absolută).

Pentru determinarea discretă, respectiv cvasicontinuă a poziției și a orientării, se utilizează scannerul laser cu reflectoare speciale [KVG 90], [WIK 88], [FOJ 91], [VOG 92], camere video cu marcaje pe sol [FRF 95], sau bobine cu magneți permanenți în pardoseală [ARB 95], [MUN 92]. Alte posibilități constau în utilizarea unor repere **naturale** cum ar fi: ramele ușilor și ale ferestrelor, stâlpi, mașini unelte. Dacă aceste repere nu sunt suficiente, se pot instala repere artificiale sub forma unor panouri.

În coloana 2 se prezintă procedeele de navigare bazate pe generarea modelului mediului, utilizând datele furnizate de către senzori și prelucrate de către sistemul de conducere în timp real. Precizia de poziționare și de urmărire a traiectoriei în acest caz este dependentă de tipul senzorilor utilizați, iar complexitatea echipamentelor și a programelor pentru procesarea datelor este ridicată.

Generarea traiectoriei utilizează procedeele de navigare bazate pe sisteme senzoriale și pe calculul mărimilor de conducere. Comenzile referitoare la traiectorie și la viteză sunt generate în timp real și trebuie transformate în mărimi de comandă necesare sistemului de tracțiune și

de virare. Se ține cont și de limitările cinematice și dinamice ale vehiculului.

O problemă importantă o constituie siguranța deplasării, lipsită de coliziuni. Se înlocuiesc senzorii activați prin atingerea unor părți ale vehiculului (bare de protecție) cu senzori de proximitate (ultrasonici, laser sau camere video) [SEV 94], [DRU 90], [HOC 90]. În ceea ce privește senzorii ultrasonici, apar limitări datorate reflexiilor, sensibilității la curenții de aer, la temperatură și umiditate. Alte probleme apar la întâlniri, deplasări pe trasee înguste, deplasări în curbe, aproape de pereți [KUG 90].

Detectarea marcajelor sau recunoașterea obiectelor este dificilă în cazul utilizării camerelor video, atunci când iluminarea spațiului de lucru este insuficientă.

Normele de siguranță prevăd o viteză maximă de deplasare de 6 km/h, un spațiu până la podea de  $10 \div 30$  cm, distanța maximă de frânare, etc [REF 72], [HGB 89], [DIN 15161].

### 2.5.2. Cerințe impuse procedeeleor de navigare.

Vehiculele uzuale sunt echipate cu sisteme de măsurare a unghiurilor și a deplasărilor (TIRO). Din funcțiile unghi - timp și deplasare - timp se pot obține, prin diverși algoritmi de calcul, coordonatele vehiculului. Aceste valori sunt întotdeauna eronate datorită următoarelor cauze:

- ◆ alunecări ale roților și cuplajelor;
- ◆ erori ale sistemelor de măsură;
- ◆ jocuri în reductoare și cuplaje;
- ◆ uzuri ale roților;
- ◆ variații ale sarcinii transportate;
- ◆ erori dinamice;
- ◆ erori de aproximare prin liniarizare la crearea modelului cinematic al vehiculului.

Metoda de navigare mai sus menționată poate fi îmbunătățită prin utilizarea în completare a altor tipuri de senzori (de exemplu giroscop) [BUI 91], [AUC 88], dar în acest mod nu se pot obține rezultate foarte bune. Vehiculul trebuie să pornească dintr-o poziție inițială cunoscută, iar erorile cresc o dată cu distanța parcursă [KOB 92].

Din motivele enumerate mai sus se completează procedeul navigării pe baza senzorilor interni cu o reactualizare a determinării poziției și orientării vehiculului. Reactualizarea are loc prin utilizarea unor sisteme de referință asociate punctelor de referință care au coordonatele cunoscute. Aceste puncte sunt detectate de către sistemele senzoriale ale vehiculului.

Procedeul de navigare se poate aprecia și alege după criteriile prezentate în figura 2.8. [HEL 96].

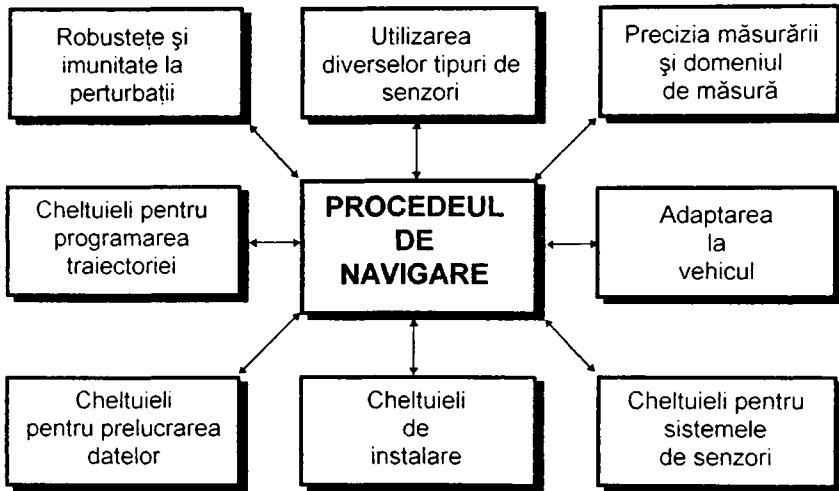


Figura 2.8.

- ◆ **precizia măsurării** coordonatelor și a orientării vehiculului în cadrul sistemului de referință determină **deplasarea autonomă** între două astfel de sisteme. Dacă se utilizează procedee odometrice, erorile nu pot fi compensate nici cu senzori speciali (giroscop). Eroarea cea mai deranjantă este cea care apare la măsurarea orientării vehiculului. Pentru eroarea  $\Delta\gamma_F$  a orientării vehiculului, după o distanță parcursă  $l_F$  apare o deplasare laterală  $\Delta l_S$

$$\Delta l_S = l_F \operatorname{tg} \Delta\gamma_F$$

Se înțelege prin **deplasare autonomă** spațiul parcurs între două puncte de către robocar pe baza măsurării odometrice.

- ◆ **domeniul de măsură** al senzorilor determină deviațiile vehiculului de la traiectoria prescrisă și influențează flexibilitatea în ceea ce privește modificarea acesteia;

- ◆ **adaptarea la vehicul**; majoritatea vehiculelor sunt echipate cu dispozitive de încărcare - descărcare și transportă sarcini care limitează posibilitățile de utilizare a unor senzori (de exemplu un scanner laser necesită un domeniu de măsură de  $360^0$  - limitat de sarcina transportată);
- ◆ **cheltuieli de instalare**; instalarea marcajelor artificiale necesită anumite cheltuieli. Cu cât acestea sunt mai reduse, se poate obține o creștere a flexibilității sistemului de transport cu costuri scăzute;
- ◆ **cheltuieli pentru prelucrarea datelor obținute cu ajutorul sistemelor senzoriale**; transformarea acestor date în coordonate ale vehiculului se face prin utilizarea calculatoarelor adecvate;
- ◆ **cheltuieli de programare**; o generare simplă a traiectoriei aduce beneficiarului flexibilitatea dorită. Din acest motiv marcajele se montează numai acolo unde sunt necesare. Costuri ridicate pentru sistemele de măsurare sunt nedorite, la fel ca și cheltuielile datorate senzorilor;
- ◆ **robustețea și imunitatea la perturbații**; fiind utilizate în mediu industrial, sistemele senzoriale sunt supuse unor condiții neprielnice și anume: murdărie, praf, variații ale iluminării, zgomote, câmpuri electromagnetice etc. De aceea sunt necesare sisteme robuste, fără erori de măsurare și cu posibilitatea repunerii rapide în funcțiune după avarie;
- ◆ **utilizarea unor tipuri diverse de senzori**. Se impune ca funcționarea sistemelor de navigare și conducere să fie independente de tipul senzorilor folosiți;
- ◆ **cheltuielile necesare pentru sistemul senzorial** vor influența acceptarea sau neacceptarea sistemelor de transport automat în diferite aplicații. Folosirea componentelor simple și robuste precum și a modulelor standardizate duce la reducerea costului sistemului senzorial.

Criteriul costurilor este mai pregnant acolo unde se utilizează puține vehicule, iar acestea funcționează pe trasee lungi.

## 2.6. Navigația vehiculelor ghidate automat pe baza hărții spațiului de lucru.

În cadrul concluziilor de la sfârșitul capitolului 1 se prezintă avantajele și dezavantajele pe care le oferă mediul industrial implementării sistemelor de transport automat. În paragraful de față se va analiza modul în care

are loc deplasarea vehiculului pe o traiectorie oarecare, într-un spațiu de lucru aflat în mediu industrial.

În primul rând se consideră că deplasarea este posibilă numai dacă există un sistem senzorial pentru perceperea informațiilor despre mediu, necesare activității desfășurate. Vehiculul fiind echipat cu acești senzori și fiind în posesia unui sistem de calcul corespunzător, deplasarea poate avea loc în mai multe moduri, și anume:

- ⇒ se cunosc, pe baza informațiilor achiziționate anterior de către programator, toate detaliile referitoare la structura mediului. Aceste date fac posibilă construirea unei hărți a spațiului de lucru. Etapa următoare este planificarea traiectoriei pe care are loc deplasarea robocarului;
- ⇒ spațiul de lucru nu este cunoscut, dar sistemul de conducere al robocarului **învață** traiectoria pe care se deplasează, condus de către operatorul uman și în același timp sistemul de conducere memorează informații referitoare la obiectele din jur și construiește o hartă. În continuare poate avea loc deplasarea pe traiectoria învățată;
- ⇒ spațiul de lucru nu este cunoscut și nu are loc o **instruire**. Prima acțiune este deplasarea pe o traiectorie oarecare. Simultan cu deplasarea, sistemul de conducere determină structura mediului cu ajutorul sistemelor senzoriale, după care construiește harta. Are loc apoi planificarea traiectoriei pe care vehiculul se va deplasa .

Din cele de mai sus se poate observa că, **oricare ar fi modul de lucru, este necesară cunoașterea structurii mediului sub forma unei hărți**. Harta este, de fapt, o reprezentare exactă a coordonatelor unor obiecte care sunt utilizate drept repere de către robocar și sunt percepute cu ajutorul sistemelor senzoriale. Sistemul de conducere determină coordonatele vehiculului în raport cu aceste repere și compară coordonatele curente cu cele prescrise. Din această comparație rezultă informațiile necesare corectării traiectoriei.

Considerând că mediul industrial are o structură cunoscută apriori, construirea hărții acestuia nu constituie o problemă majoră și nu este sarcina sistemului de conducere al robocarului. Totodată, majoritatea obiectelor fiind fixe și având forme și dimensiuni cunoscute, ele pot fi utilizate drept sisteme de referință pentru determinarea situației vehiculului. În cazul în care reperele existente (naturale) nu sunt

suficiente, se poate apela și la repere artificiale, amplasate, acolo unde este necesar, de către constructorul sistemului de transport.

Deoarece, după cum s-a arătat, orice deplasare se poate executa numai prin raportarea poziției vehiculului la anumite sisteme de referință care pot fi percepute cu ajutorul sistemului senzorial, **autorul consideră o necesitate primordială studiul modului în care are loc determinarea situării vehiculului de către sistemul de conducere al acestuia.**

## 2.7. Concluzii.

În literatura de specialitate autorul nu a întâlnit o sistematizare a problemelor referitoare la **conducerea și navigația VGA**. De asemenea, definirea unor termeni este superficială sau lipsește. **Din aceste motive s-a considerat oportună tratarea conducerii și a navigației și definirea unor termeni.**

**Pe parcursul capitolului sunt evidențiate numeroase contribuții originale ale autorului și anume:**

- ⇒ prezentarea sistematică și ierarhizată a nivelurilor de conducere, încheiată cu prezentarea sistemului de conducere al VGA;
- ⇒ evidențierea funcțiilor îndeplinite de către sistemul de conducere al robocarului;
- ⇒ atribuirea termenului **navigație** unui grup de funcții;
- ⇒ definirea **navigației**;
- ⇒ explicitarea termenului **strategii de navigare**;
- ⇒ explicitarea termenului **procedee de navigare**.

*Pe baza acestor precizări se poate determina care este poziția unei anumite funcții în cadrul sistemului de conducere al robocarului sau al procesului de fabricație.*

Vehiculele ghidate automat prezintă anumite particularități față de roboții industriali și anume:

- ⇒ mișcarea are loc într-un mediu care poate suferi modificări ale structurii sale;
- ⇒ spațiul de lucru are dimensiuni mari, ceea ce face dificilă raportarea la un sistem de referință singular. De asemenea este necesară echiparea vehiculului cu sisteme senzoriale necesare percepției mediului și a sistemului de referință;

- ⇒deoarece vehiculul se deplasează, este necesară asigurarea alimentării cu energie electrică de la o sursă proprie și realizarea unei comunicații fără fir cu sistemele de conducere ale instalațiilor din mediul înconjurător;
- ⇒fiind integrat într-un proces amplu, robocarul este subordonat unui sistem de conducere ierarhic superior.

Din aceste motive, sistemul de conducere al robocarului este mai complex și îndeplinește mai multe funcții decât sistemul de conducere al unui robot industrial.

Numărul senzorilor și diversitatea acestora este foarte mare. Senzorii sunt necesari pentru:

- ⇒măsurarea distanțelor și a unghiurilor;
- ⇒determinarea situației;
- ⇒detectarea obstacolelor.

Uneori, mai mulți senzori pot îndeplini aceleași funcții. Sistemul de conducere procesează informațiile primite de la sistemele senzoriale pentru desfășurarea în condiții optime a acțiunii programate.

Pentru ca deplasarea să fie posibilă, se impune existența unei hărți a spațiului de lucru. Această hartă poate fi construită de către programator sau de către sistemul de conducere al robocarului în faza de instruire. Dacă faza de instruire nu există, robocarul execută o mișcare oarecare, iar în timpul acesteia acumulează informații despre mediu, percepute cu ajutorul sistemelor senzoriale. Informațiile îi sunt necesare pentru construirea hărții, după care poate fi planificată traiectoria.

Fiind cunoscute cerințele pe care trebuie să le îndeplinească sistemul de conducere al vehiculului, **se poate afirma că la baza procesului de conducere se află modalitățile de determinare a situației și modul de percepție a spațiului de lucru. Din acest motiv autorul va încerca să aducă câteva contribuții la perfecționarea metodelor de determinare a situației robocarului.**

## Capitolul 3

# DETERMINAREA SITUAȚII RELATIVE A ROBOCARELOR

Deplasarea unui robocar poate avea loc prin urmărirea unei căi de ghidare, dispusă de-a lungul traiectoriei impuse. Această cale de ghidare poate avea diverse forme constructive, și anume:

- ⇒șine metalice;
- ⇒un fir parcurs de curenți având o anumită frecvență;
- ⇒benzi metalice sau vopsea, etc.

Flexibilitatea redusă a metodelor amintite se datorează în primul rând dificultăților întâmpinate la modificarea căii de ghidare. De aceea, deplasarea bazată pe căi de ghidare fizice este înlocuită cu alte metode, care nu necesită existența unor trasee materializate (ghidare virtuală). Aceste metode necesită însă sisteme senzoriale și de conducere complexe.

În paragraful 2.6. al capitolului precedent se arată că **orice deplasare se poate efectua numai prin raportarea poziției vehiculului la anumite sisteme de referință**. Cu alte cuvinte, este necesar ca în fiecare moment situarea vehiculului să fie cunoscută.

Se consideră că robocarul pornește dintr-un punct inițial, ale cărui coordonate, raportate la un sistem de referință unic, sunt cunoscute. În spațiul de lucru nu există alte sisteme de referință percepute de către sistemul senzorial al vehiculului. În acest caz, determinarea situații constă în calculul poziției și a orientării curente a vehiculului pe baza coordonatelor inițiale și a informațiilor provenite de la senzorii de deplasare, de viteză sau de accelerație. Metoda de localizare descrisă are dezavantajul că, prin integrarea mărimilor de mai sus, erorile de măsurare și de calcul care apar se cumulează.

Prin procedeul prezentat se face **determinarea situații relative**. Unii autori [PRU 89], [NIT 97] utilizează denumirea **localizare relativă** a vehiculului.

*Scopul acestui capitol este de a prezenta succint în ce constă determinarea situații relative a robocarului și de a propune un program original, conceput în întregime de către autor, pe baza căruia poate avea loc deplasarea vehiculului.*



### 3.1. Determinarea situații relative a VGA prin utilizarea odometriei.

Odometria este o metodă curent utilizată pentru localizarea VGA a căror deplasare are loc cu ajutorul roților, deoarece este relativ simplă, iar axele vehiculelor sunt prevăzute cu traductoare incrementale rotative. Pentru obținerea informațiilor necesare determinării situații unui robocar se măsoară deplasările:

- ⇒ roții motoare, în cazul vehiculelor cu trei roți;
- ⇒ roții auxiliare (montată în acest scop);
- ⇒ roților motoare, în cazul vehiculelor cu patru roți;
- ⇒ roților pasive.

Dintre metodele odometrice se preferă metoda **odometrică diferențială**, care utilizează informațiile referitoare la deplasarea a două roți, de preferință coaxiale (întrucât simetria avantajează calculele ulterioare) [NIT 97]. Ele pot fi chiar roțile motoare (pentru structura cu roți diferențiale), sau roțile pasive ale punții posterioare, destinate exclusiv susținerii (pentru vehiculele cu trei roți). În figura 3.1. se prezintă modul în care are loc determinarea situații relative a robocarului pe baza odometriei. Se notează cu S și D elementele din partea stângă, respectiv dreaptă a vehiculului. În foarte multe cazuri vehiculul este substituit printr-un punct unic aparținând structurii sale mecanice, ales astfel încât să ducă la simplificarea modelului. Acesta este **punctul caracteristic**, notat în figură cu P.

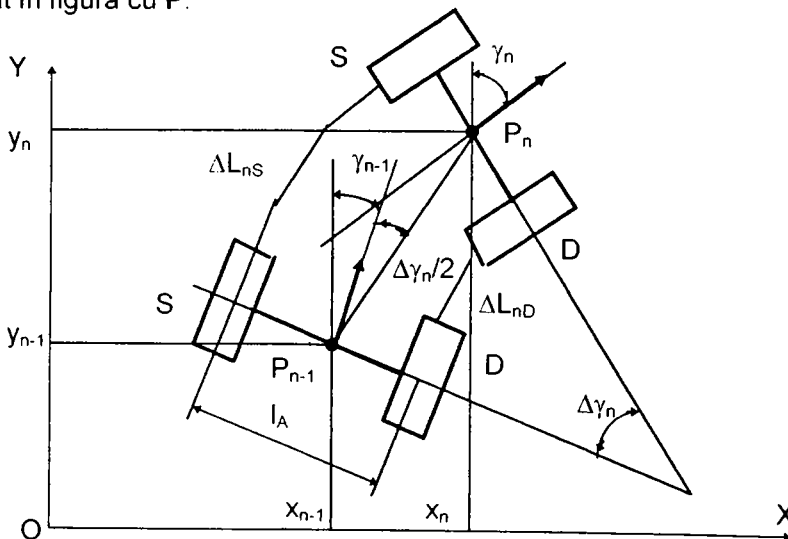


Figura 3.1.

Se vor nota cu  $\Delta L_{nS}$ , respectiv  $\Delta L_{nD}$  deplasările elementare măsurate cu ajutorul traductoarelor incrementale ale roților stânga, respectiv dreapta (figura 3.1). Rotația elementară ( $\Delta\gamma$ ) a vehiculului este dată de relația:

$$\Delta\gamma = \frac{\Delta L_{nS} - \Delta L_{nD}}{l_A} \quad (3.1)$$

iar deplasarea elementară, liniară sau circulară, a punctului caracteristic este:

$$\Delta L_n = \frac{\Delta L_{nS} + \Delta L_{nD}}{2} \quad (3.2)$$

Considerând cunoscute valorile  $(x_{n-1}, y_{n-1})$  - poziția punctului caracteristic (P) la momentul inițial,  $\gamma_{n-1}$  - orientarea vehiculului la momentul inițial; ( $\Delta\gamma_n, \Delta L_n$ ) - rotația și deplasarea elementară a vehiculului, date de relațiile (3.1) și (3.2), se poate estima poziția și orientarea vehiculului cu ajutorul următorului set de ecuații recursive:

$$x_n = x_{n-1} + \Delta L_n \cdot \sin\left(\gamma_{n-1} + \frac{\Delta\gamma_n}{2}\right) \quad (3.3)$$

$$y_n = y_{n-1} + \Delta L_n \cdot \cos\left(\gamma_{n-1} + \frac{\Delta\gamma_n}{2}\right) \quad (3.4)$$

$$\gamma_n = \gamma_{n-1} + \Delta\gamma_n \quad (3.5)$$

Relațiile (3.3) - (3.5) sunt valabile atât pentru o deplasare elementară rectilinie (caz în care  $\Delta L_n = 0$ ), cât și circulară (caz în care  $\Delta L_n \neq 0$ ). Pentru deplasări elementare de-a lungul unor tronsoane care au curbura ( $k$ ) variabilă, se impune utilizarea unor traductoare care au rezoluția mai bună, în vederea prevenirii acumulării erorilor prin efectul integrator al relațiilor stabilite. Alte erori care pot afecta tehnica odometrică sunt introduse de imprecizia valorii geometrice a lungimii axei ( $l_A$ ) și de variațiile diametrului roților, care afectează direct valorile măsurate ( $\Delta L_{nS}$ ) și ( $\Delta L_{nD}$ ). Prevenirea lor conduce la precizii de estimare suficiente pentru traiectorii cu lungimi de ordinul metrilor (lungimea crește odată cu preponderența deplasărilor liniare față de cele circulare sau clotoidice), după care se impune o nouă recalibrare prin situarea absolută.

### 3.2. Program pentru realizarea unor deplasări bazate pe odometrie.

Programul prezentat în continuare a fost conceput atât pentru măsurarea deplasărilor și a unghiurilor de virare ale robocarului, cât și pentru realizarea unor deplasări și viraje pe baza mărimilor de conducere prescrise de către utilizator sau obținute prin instruire.

### 3.2.1. Generalități referitoare la programul propus.

Programul prezentat în acest capitol este scris în limbajul C, apoi este convertit în format hexazecimal. În urma conversiei se obține un fișier cu extensia .HEX, care este încărcat în sistemul de dezvoltare construit în jurul microcontrolerului 80C552. Conversia se realizează în mediul integrat de dezvoltare PCMON cu ajutorul utilităților C51, L51, OH51. Pentru conversie sunt parcurse următoarele etape:

- ◆ compilatorul C51 realizează conversia din fișierul sursă scris în limbajul C în fișier obiect cu extensia .OBJ;
- ◆ link-editorul L51 realizează harta legăturilor dintre program și funcțiile pe care acesta le utilizează, aflate în biblioteci.
- ◆ programul de conversie în format hexazecimal, OH51 realizează conversia fișierului obiect .OBJ în fișier format hexazecimal cu extensia .HEX.

Programul complet, utilizat pentru deplasarea robocarului pe baza odometriei, este scris în limbajul C și va fi prezentat în detaliu în anexa 1. Sunt utilizate o serie de funcții standard, definite în bibliotecile cu care este dotat sistemul de dezvoltare și anume:

#### ***sorin.lib***

iar prototipurile acestor funcții se găsesc în fișierul *header*.

#### ***sorin.h***

În această bibliotecă se găsesc funcții pentru:

- ◆ comunicația serială (prin intermediul interfeței seriale RS232);
- ◆ generarea semnalului PWM la buzzer;
- ◆ afișare pe ecranul LCD cu care este prevăzut sistemul de dezvoltare;
- ◆ înscrierea memoriei EEPROM a sistemului de dezvoltare;
- ◆ scanarea unei tastaturi cu 4x3 taste;
- ◆ conversia analog-numerică;

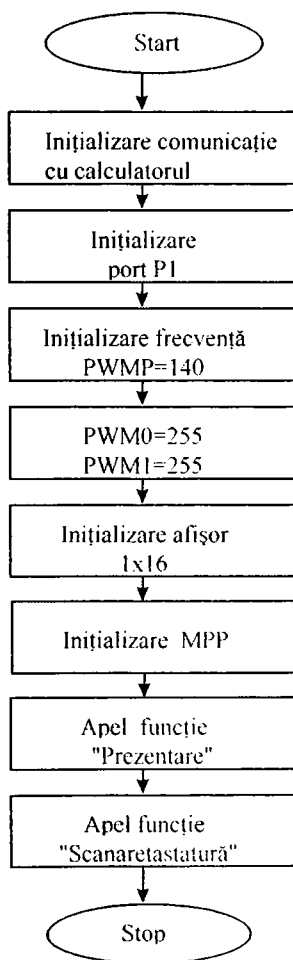
Biblioteca este inclusă la începutul fiecărui program care utilizează aceste funcții.

```
#include "..\..\lib\sorin.h"
```

De asemenea se va include și fișierul header al microcontrolerului 80C552, *reg552.h*, pentru a informa compilatorul că microcontrolerul utilizat este 80C552:

```
#include <reg552.h>
```

În figura 3.2. se prezintă schema logică a programului principal:



**Figura 3.2**

Programul principal începe prin inițializarea comunicației cu calculatorul. ***initserial()***; Această funcție se găsește în biblioteca de funcții a sistemului de dezvoltare. Comunicarea sistemului de dezvoltare cu calculatorul se realizează prin intermediul interfeței seriale RS232. Urmează inițializarea portului P1, având ca efect frânarea motorului de curent continuu, pentru a se evita apariția unor deplasări neprevăzute ale robocarului.

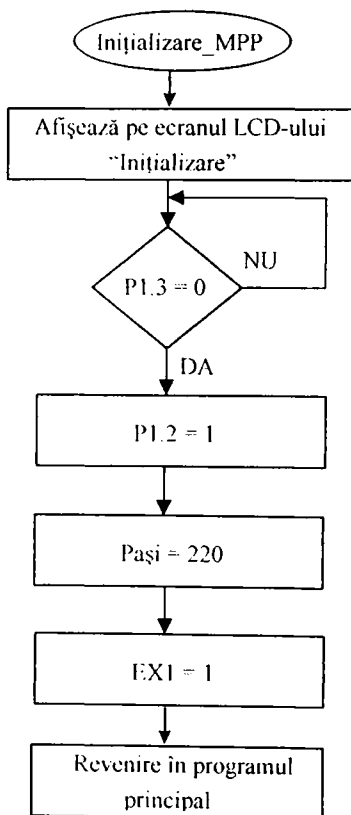
***P1&=0xF8;***

Se stabilește frecvența semnalului PWM prin încărcarea registrului PWMP:

**PWMP=150;**

Afișarea pe ecranul LCD se inițializează cu ajutorul funcției din bibliotecă: **initdisplay1( 0 );**

Funcția **inițializare\_MPP()** realizează inițializarea mecanismului de virare al robocarului. Schema logică a programului care implementează această funcție este prezentată în figura 3.3. Prin această procedură, mecanismul de virare al robocarului este adus în poziția pentru care are loc deplasarea rectilinie a vehiculului.



**Figura 3.3.**

Inițializarea se realizează pe baza următorului algoritm:

- ◆ se rotește MPP spre stânga până când traductorul optic pentru inițializare, montat pe șasiul robocarului, furnizează un impuls. Pe baza acestui traductor, sistemul de conducere stabilește originea sistemului de referință al mecanismului de virare. Impulsul provenit de la traductorul optic constituie un semnal de intrare pentru sistemul de dezvoltare și este aplicat pe pinul P1.3.
- ◆ se rotește MPP un număr de 220 de pași spre dreapta, față de originea sistemului de referință. Pentru acest număr de pași mecanismul de virare este poziționat pe direcția de deplasare rectilinie.

Pentru a detecta semnalul furnizat de către traductorul optic, se poziționează pe "1" logic linia P1.3, iar apoi se așteaptă până când P1.3 devine "0" logic. Această tranziție este detectată cu ajutorul subrutinei **inițializare\_MPP()**:

```
if(P1==0xF0)
```

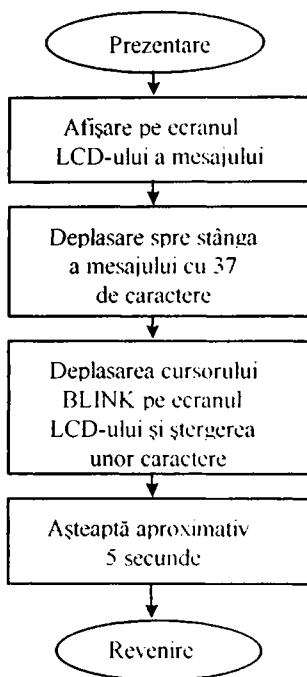
```
{  
    PWM1=255;  
    delay(800);  
    P1 |= 0x04;  
    pasi=220;  
    PWM1=125;  
    EX1=1;  
    break;  
}
```

După stabilirea originii sistemului de referință, are loc schimbarea sensului de rotație a MPP prin poziționarea pe "1" logic a liniei P1.2, iar apoi este părăsită funcția **inițializare\_MPP()**.

Se menționează că originea sistemului de referință, nu va fi atinsă decât la inițializare. Mecanismul de virare nu mai poate ajunge în origine decât în caz de avarie sau la ieșirea din program, deoarece se **realizează o limitare a mișcării acestuia la stânga și la dreapta prin soft**. Această limitare se realizează de către sistemul de conducere prin tratarea cererii de întreruperupere sosite pe linia INT1. Impulsurile furnizate de către traductorul mecanismului de virare sunt aplicate pe linia de intrare în sistem INT1.

În capitolul 6 se va descrie detaliat algoritmul care realizează limitarea rotirii la stânga și la dreapta a mecanismului de virare.

Funcția **prezentare()** este o funcție care prezintă pe afișajul LCD conectat la sistemul de conducere al robocarului informații referitoare la acțiunile care se execută. Schema logică a programului care realizează această funcție este reprezentată în figura 3.4.



**Figura 3.4.**

Funcția **Scanare\_tastatură()** este funcția cea mai importantă și cea mai complexă a programului executat de către microcontroler.

Se realizează un salt din programul principal în această funcție, iar revenirea nu mai are loc decât în caz de avarie.

Schema logică a acestei funcții este de complexitate ridicată, motiv pentru care ea va fi prezentată detaliat în continuare, împreună cu modulele de deplasare a robocarului.

### 3.2.2. Deplasarea prin comandă manuală.

Pe durata experimentelor, deplasarea robocarului s-a realizat în conformitate cu comenzile transmise prin intermediul tastaturii unui calculator compatibil PC.

Comenzile manuale pentru deplasarea robocarului sunt:

- ◆ deplasare prin comandă manuală înainte: prin apăsarea tastei **8**;
- ◆ deplasare prin comandă manuală înapoi: prin apăsarea tastei **2**;
- ◆ rotire spre dreapta un grad: apăsarea tastei **6**;
- ◆ rotire spre stânga un grad: apăsarea tastei **4**.

După cum se observă din schema logică prezentată în figura 3.5, în cadrul funcției **Scanare\_tastatură()** se citește bitul **RI**, care semnifică, dacă este "1" logic, că a sosit un octet prin interfața serială RS232. Acest octet este transferat din registrul **S0BUF** (registrul de comunicație serială) în variabila **Data**.

Apoi se compară această variabilă pe rând cu codul ASCII al numărului **8**, respectiv **2**, etc. (Codul ASCII al numărului este diferit de numărul respectiv. El se obține prin adunarea numărului respectiv cu constanta 48).

Dacă s-a apăsat tasta **8**, se va urma bucla logică marcată cu culoare albastră și are loc deplasarea robocarului înainte.

Se testează bitul **înainte**. Testarea are ca scop determinarea sensului de deplasare anterior primirii comenzii. Acest bit detectează schimbarea sensului de deplasare, iar pentru a se evita solicitarea excesivă a mecanismelor de transmisie de la motor la roată (reductor), este apelată funcția **opritMCC()**. Această funcție realizează o întârziere soft de circa 0,5 secunde și este descrisă în capitolul 6. Dacă bitul **înainte** are valoarea "0" logic, înseamnă că, anterior aplicării noii comenzi, robocarul a efectuat o deplasare înapoi. Deoarece comanda actuală este **8** (deplasare înainte și având loc o schimbare a sensului de mișcare), este apelată funcția **opritMCC()**.

În continuare este poziționat pe "1" logic bitul **înainte**. Acest bit indică faptul că deplasarea robocarului are loc înainte.

Pe ecranul afișajului LCD se va scrie mesajul "Deplasare înainte".



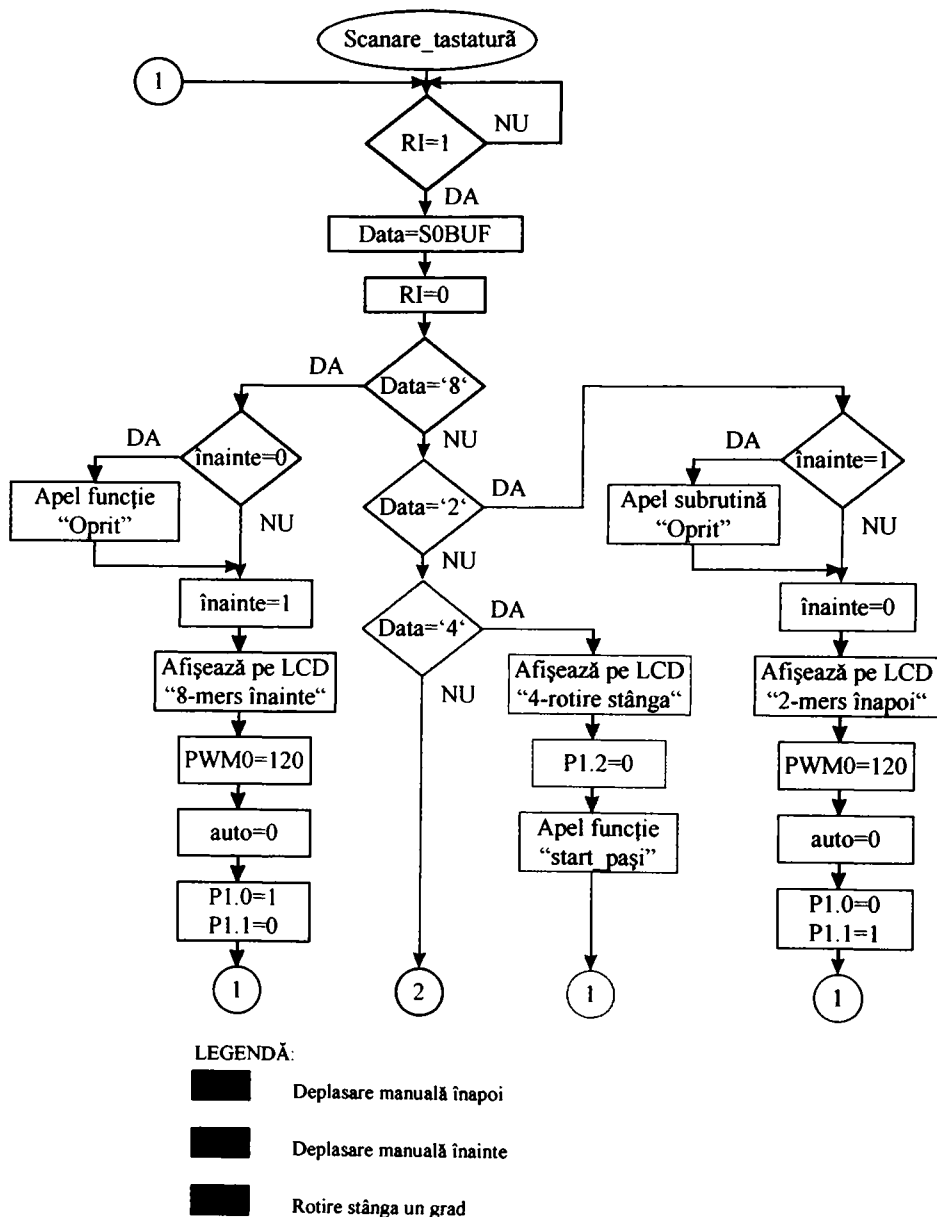


Figura 3.5.

În registrul PWM0 se încarcă valoarea 120, care corespunde unui factor de umplere 0,5 al semnalului PWM. Acest semnal comandă elementul de execuție care aparține motorului de tracțiune (MCC).

Bitul **auto** este un indicator pentru determinarea tipului comenzii. El indică faptul că deplasarea robocarului are loc prin comandă manuală sau automat pe traiectorie impusă.

În continuare se poziționează pe "1" logic linia de port P1.0 și pe "0" logic linia P1.2, ceea ce corespunde deplasării înainte.

Secvența de program pe baza căruia se realizează deplasarea robocarului înainte este următoarea:

```
if(Data[0]==50)
{
    if(!inainte)
        oprit();
    inainte=1;
    clearscreen();
    write("2-mers inapoi");
    automat=0;
    PWM0=120;
    EX0=1;
    P1 &= 0xFE;
    P1 |= 0x02;
}
```

Dacă s-a apăsata tasta **2**, se va urma bucla logică marcată cu culoarea roșie, iar deplasarea robocarului are loc înapoi.

Algoritmul de comandă este identic cu cel utilizat pentru deplasarea înainte, cu specificația că bitul **înainte** va fi poziționat de data aceasta pe "0" logic. P1.0 este poziționat pe "0" logic, iar P1.1 este poziționat pe "1" logic pentru schimbarea sensul de deplasare.

Descrierea detaliată a funcționării elementului de execuție al MCC va fi prezentată în capitolul 6.

*În concluzie, în momentul în care sistemul de conducere primește comanda de deplasare **înainte** sau **înapoi**, MCC este alimentat cu jumătate din tensiunea nominală și astfel robocarul se va deplasa cu viteză constantă până la primirea comenzii de oprire sau a unei comenzi pentru schimbarea sensului de deplasare. Deci, pentru deplasarea manuală înainte sau înapoi este suficientă apăsarea tastei dorite o singură dată.*

Dacă s-a apăsat tasta **4**, se va efectua o rotație spre stânga cu un grad. În schema logică traseul urmărit este cel din figura 3.5, marcat cu linie verde.

Secvența de program este descrisă în continuare:

```
if(Data[0]==52)
{
    clearscren();
    write("4-rotire stinga");
    P1 &= 0xFB;
    P1 |= 0x08;
    start_pasi(1,&pasi);
}
```

Pe ecranul LCD se va afișa mesajul **“Rotire la stânga”**.

Are loc poziționarea liniei de port P1.2 pe “0” logic (ceea ce corespunde sensului de rotire spre stânga).

Se apelează funcția **start\_pași()** cu parametrul unu.

```
start_pasi(unsigned char grade,unsigned char *point)
{
    *point=grade*2;
    PWM1=125;
    EX1=1;
}
```

Se observă că această funcție este apelată cu parametrii: **grade** și **point**. Parametrul **grade** specifică numărul de grade cu care se va roti mecanismul de virare și este introdus de către utilizator prin intermediul tastaturii (în cazul în care este prescrisă o rotație la stânga sau la dreapta), iar în cazul comenzii manuale este implicit unu (un grad). Funcția descrisă returnează un pointer **point**. Acesta indică adresa de memorie la care se află numărul de impulsuri pe care trebuie să le furnizeze TIRO al MPP pentru a se efectua rotirea cu numărul de grade prescris. Se încarcă registrul PWM1 cu valoarea 125, ceea ce corespunde unui factor de umplere 0,5 al impulsurilor care constituie semnalul de comandă al elementului de execuție al MPP și se validează întreruperea externă EX1.

În cadrul subrutinei pentru tratarea întreruperii EX1, se numără impulsurile provenite de la TIRO, iar în momentul în care s-a ajuns la numărul de impulsuri prescris se întrerupe semnalul pe linia PWM1.

Dacă tasta apăsată este **6**, se va efectua o rotație spre dreapta cu un grad, iar schema logică corespunzătoare este cea prezentată în figura 3.6 (traseul de culoare albastră).

*În concluzie, semnalul de tact care comandă elementul de execuție al MPP este generat la ieșirea PWM1, iar semnalul care stabilește sensul de rotație este furnizat pe linia de port P1.2. În toate modurile de lucru (atât manual cât și automat), rotirea MPP se realizează pe baza numărării impulsurilor provenite de la TIRO care aparține sistemului de virare. Numărul de impulsuri/grad este 2,38. În cazul virării prin comandă manuală, la fiecare apăsare a tastei corespunzătoare comenzii de virare la stânga sau la dreapta, mecanismul de virare va fi rotit cu un grad.*

### 3.2.3. Deplasarea pe traiectorie impusă.

Deplasarea pe traiectorie impusă a robocarului poate avea loc în conformitate cu comenzile transmise prin intermediul tastaturii. Aceste comenzi sunt următoarele:

- ◆ deplasare înainte pe traiectorie impusă (se realizează prin intermediul tastei **F**);
- ◆ deplasare înapoi pe traiectorie impusă (se realizează prin intermediul tastei **A**);
- ◆ virare la dreapta cu un număr prescris de grade (se realizează prin intermediul comenzii de la tastatură **R**);
- ◆ virare la stânga cu un număr prescris de grade (se realizează prin intermediul comenzii de la tastatură **L**).

Programul este flexibil, permițând robocarului efectuarea în același timp a unei mișcări pe traiectoria impusă și a unei mișcări comandate manual. Spre exemplu, poate avea loc o deplasare automată pe o traiectorie impusă (înainte sau înapoi), în timp ce virarea este realizată prin comandă manuală. De asemenea, se poate prelua de către operatorul uman controlul robocarului în orice moment prin oprirea lui, apăsând tasta **O**.

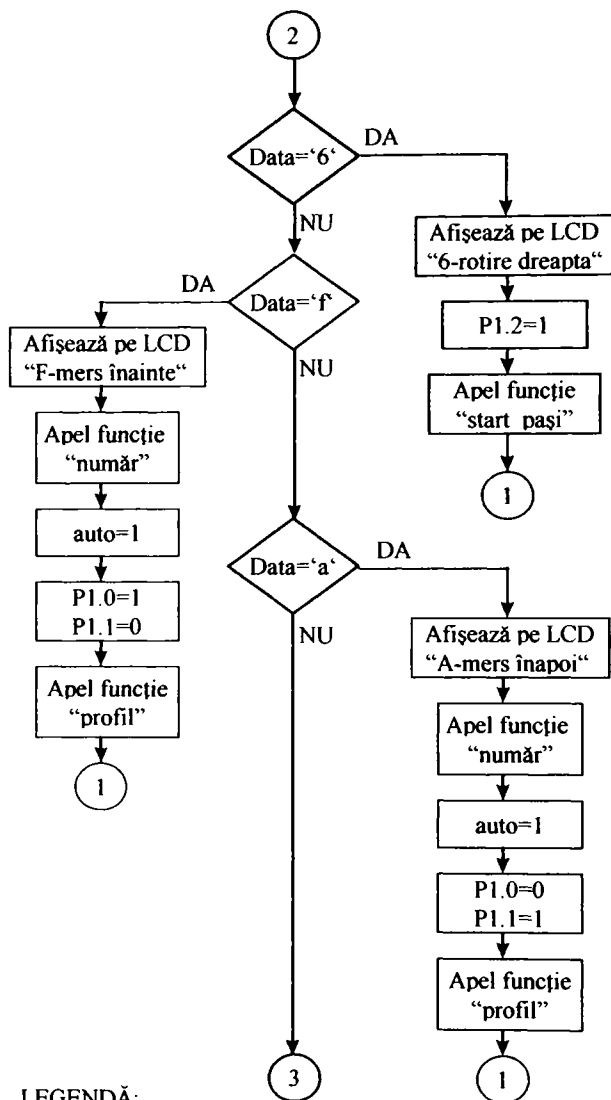


Figura 3.6.

După introducerea comenzii de deplasare pe traiectoria impusă, se așteaptă introducerea numărului de centimetri care trebuie parcurși (în

cazul în care comanda a fost de deplasare pe traseu impus: **F** sau **A**), respectiv numărul de grade cu care trebuie să vireze robocarul (în cazul în care comanda a fost de rotire cu un număr de grade prescris **R** sau **L**).

Numărul de centimetri trebuie să fie cuprins între 0 ÷ 999, iar numărul de grade între 0 ÷ 60 (la stânga sau la dreapta). Numărul de centimetri, respectiv grade, trebuie introdus corect (nu se acceptă litere, ci numai cifre).

Funcția care realizează citirea numărului prescris de centimetri, respectiv grade, este **int număr()**. Această funcție este comună atât pentru citirea numărului de centimetri, cât și pentru citirea numărului de grade și va fi prezentată în continuare:

```

int numar()
{
  if(m)
  {
    txstr("\nIntroduceți numărul de cm (0-999):\n",5);
    j=0;
  }
  else
  {
    txstr("\nIntroduceți numărul de grade (0-60°):\n",5);
    j=1;
    Data[0]='0';
  }
  for(k=j;k<3;k++)
    while(1)
      if( rxch(&Data[k]))
        if(Data[k]<58 && Data[k]>47)
          break;
        else
          txch(8);
  i=(Data[0]-48)*100+(Data[1]-48)*10+(Data[2]-48);
  return i;
}

```

Bitul **m** indică faptul că are loc o deplasare sau o rotație (dacă **m** este "0" logic are loc o rotație cu numărul de grade prescris, iar dacă **m** este "1" logic are loc o deplasare cu numărul de centimetri prescris).

Se citește valoarea în grade, respectiv în centimetri prin interfața serială RS232. Aceste valori trebuie reconstituite, deoarece se transmit codurile ASCII ale tastelor apășate. Reconstituirea valorii în centimetri, respectiv grade se realizează în variabila  $i$ . Valoarea variabilei  $i$  (centimetri sau grade) este returnată de funcția **numar()** și este utilizată în continuare în cadrul programului.

Dacă de la tastatura calculatorului se primește comanda deplasare automată înainte (**F**) cu un număr prescris de centimetri, se urmărește traseul de culoare roșie din schema logică din figura 3.6.

Pe ecran se va afișa mesajul "Deplasare înainte", după care este apelată funcția **număr()**. Această funcție a fost descrisă mai sus.

Bitul **auto** este poziționat pe "1" logic, deoarece mișcarea nu se efectuează prin comandă manuală (are loc în mod automat pe traiectoria impusă).

În modul automat de deplasare pe traiectoria impusă, are loc o accelerare sau o decelerare la pornirea, respectiv la oprirea vehiculului.

În cazul de față, spațiul pe care robocarul se deplasează accelerat pentru a atinge viteza maximă este 40 cm, egal cu spațiul pe care acesta se deplasează decelerat (la oprire). Această distanță este parcursă pe baza numărării impulsurilor furnizate de către traductorul incremental al sistemului de tracțiune.

Vor exista două situații:

- ◆ distanța care urmează a fi parcursă este mai mică de 80 cm;
- ◆ distanța care urmează a fi parcursă este mai mare de 80 cm.

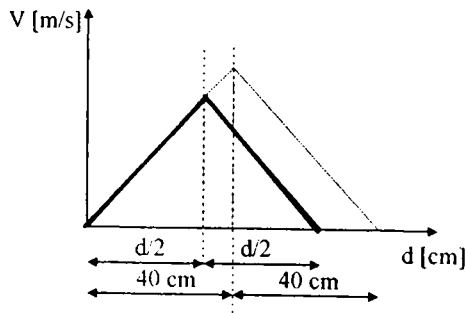


Figura 3.7.

În primul caz, robocarul se va deplasa accelerat până la jumătatea distanței, iar cealaltă jumătate va fi parcursă decelerat (figura 3.7) și nu va fi atinsă viteza maximă.

În al doilea caz, robocarul se va deplasa accelerat primii 40 cm, apoi urmează o deplasare cu viteza maximă, urmând ca ultimii 40 cm ai traiectoriei să fie parcurși decelerat (figura 3.8).

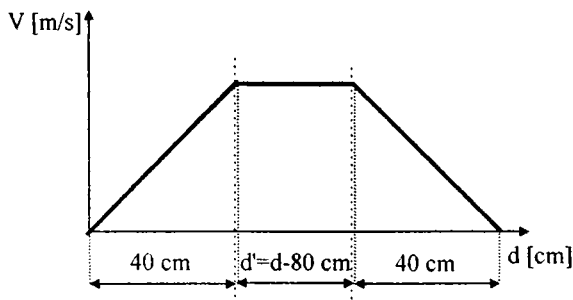


Figura 3.8.

Acest profil de viteză se obține apelând funcția **profil** (**int nr, unsigned char \*acc, int \*palier, unsigned char \*decc**) cu parametri corespunzători.

```

profil(int nr,unsigned char *acc,int *palier,unsigned char *decc)
{
if(nr<80)
    {
        *palier=0;
        *acc=nr/2;
        *decc=nr/2;
    }
else
    {
        *acc=40;
        *decc=40;
        *palier=nr-80;
    }
EX0=1;
l=165;
PWM0=l;
}

```



Parametrul **nr** cu care este apelată funcția, reprezintă numărul de centimetri prescris de către utilizator, iar funcția **profil()** va returna trei pointeri după cum urmează:

- ◆ **acc** pointer, indică spre zona de memorie în care se găsește numărul de centimetri care trebuie parcurs accelerat;
- ◆ **decc** pointer, indică spre zona de memorie în care se găsește numărul de centimetri care trebuie parcurs decelerat;
- ◆ **palier** pointer, indică spre zona de memorie în care se găsește numărul de centimetri care trebuie parcurs cu viteză maximă.

În continuare se încarcă în registrul PWM0 valoarea 165, ceea ce înseamnă că motorului de curent continuu i se aplică 1/3 din valoarea maximă a tensiunii de alimentare. Apoi are loc incrementarea, respectiv decrementarea registrului PWM0 la fiecare cm parcurs, pentru a se obține profilul de viteză descris anterior.

Impulsurile furnizate de către TIRO sunt semnale de intrare în sistemul de dezvoltare cu microcontroler 80C552. Ele se aplică pe intrarea de întrerupere INT0 a microcontrolerului. Numărul de impulsuri este divizat cu 53 în subrutina de tratare a întreruperii INT0, deoarece după 53 de impulsuri robocarul s-a deplasat 1 cm.

Cazul corespunzător comenzii **A** (deplasare pe traiectorie impusă înainte), este reprezentat pe schema logică din figura 3.6 cu culoare verde.

Algoritmul este la fel ca și în cazul comenzii **F**, cu modificările privitoare la sensul de deplasare.

Dacă comanda care se primește este **R**, robocarul va efectua un viraj la dreapta cu un număr prescris de grade, iar traseul care se urmărește pe schema logică este cel marcat cu culoare roșie în figura 3.9.

Pe ecranul afișajului LCD se va scrie mesajul "Rotire dreapta", apoi se poziționează linia de port P1.2 pe "0" logic (corespunzător rotirii spre dreapta). Bitul **m** este poziționat pe "0" logic (semnificația lui a fost prezentată mai sus).

Se apelează funcția **număr()**, care realizează citirea numărului de grade. Această funcție a fost prezentată anterior.

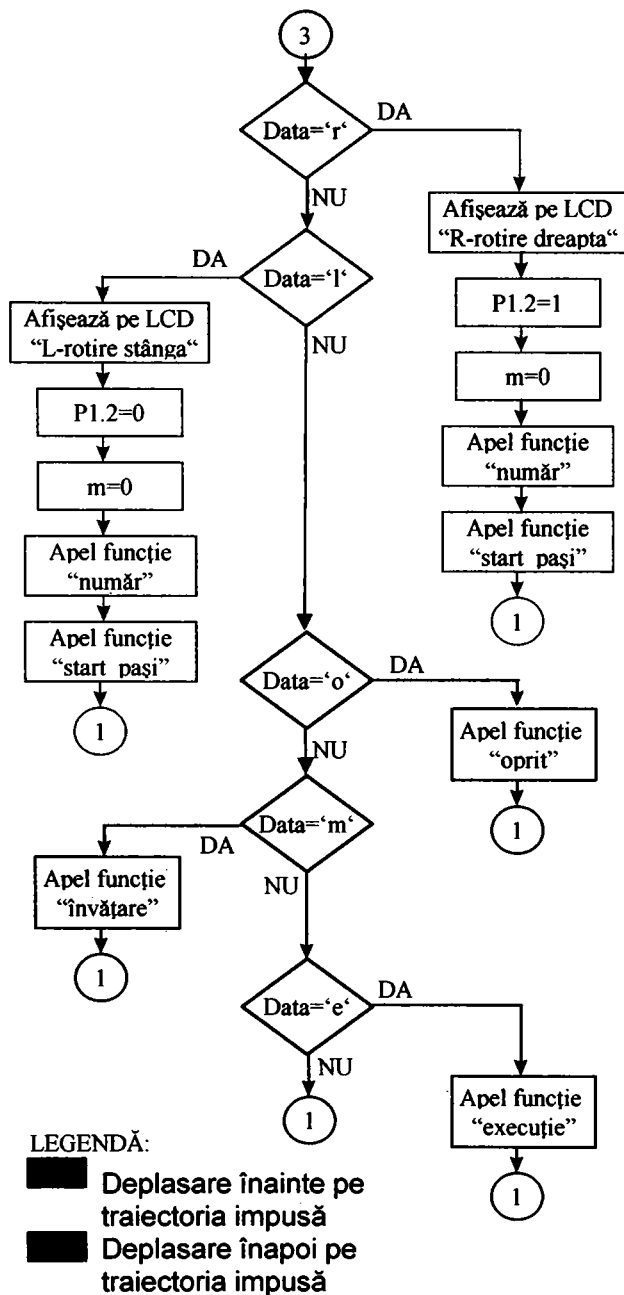


Figura 3.9.

Se apelează funcția **start\_pași()** cu parametrul numărul de grade. Ea returnează un pointer spre zona de memorie care conține numărul de pași (numărul de impulsuri care trebuie furnizate de către TIRO al sistemului de virare, astfel încât să se obțină rotirea cu numărul de grade dorit).

În cazul în care comanda este **L**, ceea ce semnifică rotire la stânga cu un număr prescris de grade, schema logică a programului este cea din figura 3.9, traseul marcat cu culoare verde.

*În concluzie, deplasarea, respectiv virarea pe o traiectorie impusă se bazează pe informațiile primite de la senzorii TIRO, montați în sistemul de tracțiune, respectiv de virare. Utilizând aceste informații se construiește profilul de viteză la deplasarea pe traiectoria impusă.*

### 3.2.4. Învățarea unei traiectorii.

Pentru conceperea și implementarea algoritmului de învățare, se va avea în vedere faptul că spațiul de memorie RAM al sistemului cu microcontroler 80C552 este restrâns.

O metodă de realizare a învățării este memorarea informațiilor, primite de la traductoarele TIRO montate pe axele robocarului, la anumite intervale de timp constante (egale cu durata de eșantionare).

Această metodă de învățare, presupune alegerea unei perioade de eșantionare cât mai mici, pentru ca traiectoria **învățată** să reproducă cât mai precis traiectoria parcursă în timpul instruirii. În fiecare perioadă de eșantionare are loc achiziționarea și memorarea informațiilor furnizate de către traductoare. Metoda necesită un spațiu larg de memorie RAM și datorită acestui motiv nu este aplicată în cazul de față.

*În cadrul algoritmului de învățare proiectat, se realizează memorarea informațiilor numai atunci când se primește o nouă comandă de deplasare sau de virare. Acest mod de lucru micșorează semnificativ spațiul de memorie ocupat. Pentru ca sistemul de conducere al vehiculului să nu execute și comenzile eronate, acestea nu sunt luate în considerare.*

**Algoritmul de învățare realizează o comprimare maximă a comenzilor primite, fiind salvate în memorie numai comenzile utile.**

Logica utilizată pentru compresia comenzilor primite este următoarea:

**a.)** În cazul în care se primește comanda pentru **deplasare manuală înainte** (se apasă tasta **8**) pot exista două situații:

- dacă anterior primirii comenzii robocarul a fost oprit sau a efectuat o deplasare înapoi, atunci se va salva în memorie comanda pentru **deplasare înainte** (codul ASCII al caracterului **8**). Salvarea în memorie se va face o singură dată, indiferent de numărul comenzilor pentru **deplasare înainte** pe care OU le transmite sistemului de conducere (în cazul în care OU ține tasta **8** apăsată continuu se ia în considerare numai o singură comandă). Nu se va face o nouă salvare în memorie a comenzii pentru **deplasare înainte** decât după ce MCC se va opri (comanda **O**) sau după ce MCC își va schimba sensul de rotire (comanda **2**);
- dacă anterior primirii comenzii, robocarul efectua o deplasare înainte, nu se va salva în memorie noua comandă.

**b.)** În cazul în care se primește comanda pentru **deplasare manuală înapoi** (se apasă tasta **2**), algoritmul folosit pentru salvarea în memorie a acestei comenzi este similar cu cel prezentat la punctul **a**).

**c.)** Dacă se primesc comenzile pentru **virare manuală la stânga** sau **la dreapta** (comanda de rotire la stânga un grad este **4**, iar comanda rotire la dreapta un grad este **6**), există două posibilități:

- dacă robocarul se deplasează înainte sau înapoi, comanda pentru virare este salvată în memorie necondiționat;
- dacă robocarul este oprit și sistemul de conducere primește comanda pentru **virare un anumit număr de grade la dreapta** și apoi comanda pentru **virare un anumit număr de grade la stânga**. Pe baza algoritmului de învățare se va calcula diferența dintre cele două numere de impulsuri furnizate de către traductor, stabilindu-se sensul de virare. Apoi se va salva în memorie comanda pentru **virare** în sensul rezultat, împreună cu diferența numerelor de impulsuri furnizate de către TIRO, corespunzătoare celor două rotații. Dacă cele două numere sunt egale, atunci nu va avea loc salvarea în memorie.

Această metodă de tratare a comenzilor pentru **virare la stânga sau la dreapta** are următoarele avantaje:

- ◆ utilizarea mai eficientă a memoriei, prin salvarea doar a unei singure comenzi de virare sau poate chiar a nici uneia în cazul în care unghiul de virare spre stânga este egal cu unghiul de virare spre dreapta (pentru că cele mai mari consumatoare de spațiu de memorie sunt comenzile de rotire);
- ◆ la redare nu se vor efectua o serie de mișcări suplimentare de virare la stânga sau la dreapta, ci mecanismul de virare se va poziționa pe direcția dorită.

**d.)** În cazul în care se primește comanda pentru **oprire** (se apasă tasta **O**), iar robocarul se deplasa într-un sens sau în celălalt anterior primirii comenzii, această comandă este salvată în memorie. Salvarea în memorie se realizează o singură dată, indiferent de numărul comenzilor de oprire care se primesc.

În momentul în care se primește o comandă validă (care se salvează în memorie), se vor citi numerele de impulsuri furnizate de către TIRO montate pe cele două axe, iar aceste numere se vor salva în locațiile de memorie corespunzătoare.

În program se utilizează trei pointeri care indică spre zonele de memorie în care se vor salva comenzile, impulsurile de la TIRO al MCC și impulsurile de la TIRO al MPP:

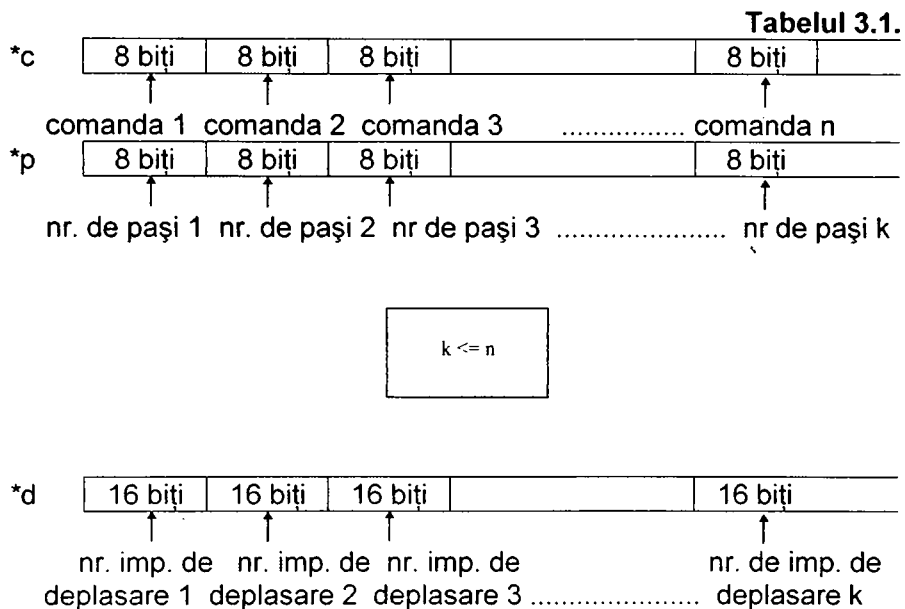
- ◆ **\*c** - pointer care indică spre zona de memorie în care se salvează comenzile valide;
- ◆ **\*p** - pointer care indică spre zona de memorie în care se salvează impulsurile provenite de la TIRO al MPP;
- ◆ **\*d** - pointer care indică spre zona de memorie în care se salvează impulsurile provenite de la TIRO al MCC.

Intrarea în bucla algoritmului de învățare se realizează apăsând tasta **M**.

Zonele de memorie alocate pointerilor sunt:

- ◆ pentru **c** zona de memorie începe de la adresa 2300H;
- ◆ pentru **p** zona de memorie începe de la adresa 3300H;
- ◆ pentru **d** zona de memorie începe de la adresa 5300H.

În tabelul 3.1 sunt prezentate schematic zonele de memorie corespunzătoare pointerilor: \*c, \*p și \*d:



Pentru comenzi se alocă câte un singur octet, deoarece codurile comenzilor sunt reprezentate pe un singur octet. Pentru numărul de pași (numărul de impulsuri furnizate de către TIRO al MPP) se alocă de asemenea un octet, ceea ce corespunde unei rotiri la stânga sau la dreapta de 60 grade ( $2 \times 60 \text{ grade} \times 2 \text{ imp} / \text{grad} = 240 \text{ imp} < 255$  adică un octet). Pentru deplasare (numărul de impulsuri furnizate de TIRO al MCC) se alocă o zonă de memorie de doi octeți, deoarece numărul de impulsuri/centimetru este 53 ( $100 \text{ cm} \times 53 \text{ imp} / \text{cm} = 530 \text{ imp} > 255$ ).

Salvarea în memorie a numerelor de impulsuri furnizate de către cele două traductoare se realizează conform tabelului 3.2:

**Tabelul 3.2.**

*c		c1	c2	c3	
*p	p0	p1	p2	p3	
*d	d0	d1	d2	d3	

Salvarea în memorie a acestor numere se va realiza numai atunci când se primește următoarea comandă validă.

Algoritmul de învățare descris mai sus (care conține compresia comenzilor și eliminarea comenzilor false) este implementat conform schemelor logice din figurile 3.10, 3.11, 3.12.

În continuare se va prezenta algoritmul de învățare, urmărind schema logică din figura 3.10. Inițializările se referă la ștergerea zonelor de memorie alocate pointerilor **c**, **p** și **d**. Ștergerea se realizează prin apelarea funcției **șterge()**.

```

șterge()
{
  c=0x2300;
  p=0x3300;
  d=0x5300;
  for(i=0;i<2000;i++)
  {
    *p++=0;
    *d++=0;
    *c++=0;
  }
  c=0x2300;
  p=0x3300;
  d=0x5300;
}

```

Din momentul în care s-a intrat în bucla de învățare, se așteaptă introducerea comenzilor pentru deplasare sau virare manuală. Din această buclă se poate ieși doar dacă se apasă tasta **Q**.

Astfel, dacă se apasă tasta **2** (deplasare înapoi), se verifică dacă bitul **stop** este "1" logic (dacă robocarul nu a efectuat anterior primirii acestei comenzi o deplasare).

Dacă bitul **stop** este "0" logic (vezi organigrama din figura 3.10), va fi salvată în memorie comanda anterioară (care ar putea fi **virare**), numai dacă variabila **pas** (impulsuri provenite de la traductorul TIRO corespunzător MPP) este diferită de 0.

Dacă variabila **pas** este diferită de 0, se determină sensul în care a avut loc rotirea anterioară prin compararea variabilei **pas** cu valoarea 0.

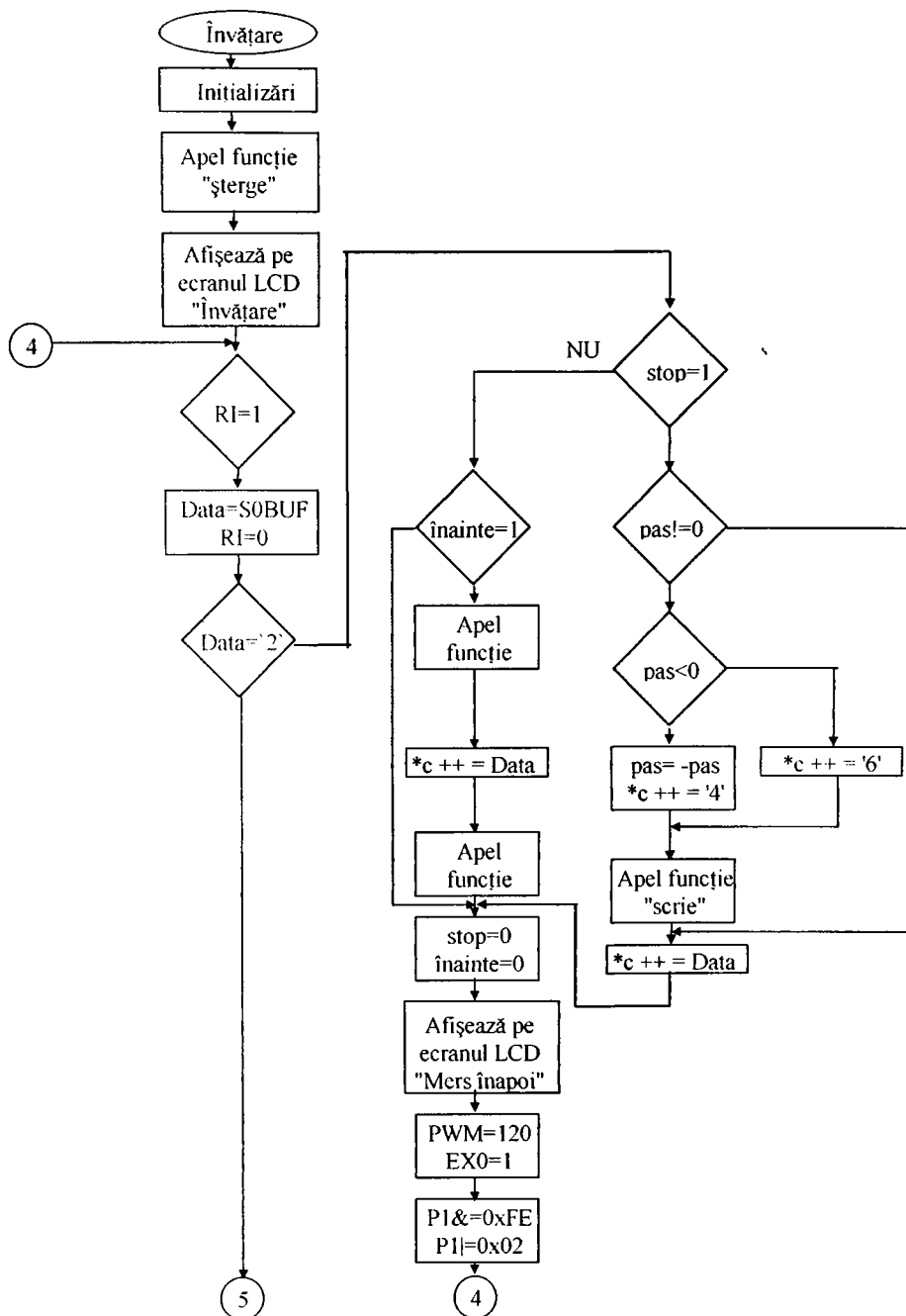


Figura 3.10.



Dacă variabila **pas** < 0, comanda de virare a fost spre **stânga** și se salvează în memorie la adresa indicată de către pointerul **c**, codul ASCII al caracterului **4** și la adresa indicată de pointerul **p** numărul de impulsuri furnizate de către TIRO al MPP. Dacă variabila **pas** > 0, comanda de virare a fost spre **dreapta** și se salvează codul ASCII al caracterului **6**. După aceasta se comandă elementul de execuție al MCC la fel ca în modul **deplasare manuală**. Dacă bitul **stop** nu este "0" logic, rezultă că, anterior primirii comenzii **deplasare înapoi**, robocarul se deplasa **înainte**. Are loc detecția sensului de deplasare anterior primirii comenzii, prin testarea bitului **înainte**. Dacă bitul **înainte** este "0" logic, robocarul se deplasa **înapoi** anterior primirii comenzii, ceea ce înseamnă că actuala comandă este una redundantă și nu va fi salvată în memorie, ci se va face un salt direct la comanda manuală a robocarului **înapoi**. Dacă bitul **înainte** este "1" logic, rezultă că robocarul se deplasa **înainte** (are loc o schimbare a sensului de deplasare) și se va salva în memorie comanda anterioară, care era de deplasare **înainte**. Deci se va salva în memorie la adresa indicată de pointerul **c**, codul ASCII al caracterului **8**. De asemenea vor fi salvate în memorie informațiile primite de la traductoarele TIRO.

Dacă se primește comanda pentru **virare**, de exemplu **4** (virare la stânga), se verifică dacă robocarul a fost oprit anterior, prin testarea bitului **stop**. Dacă acesta nu este oprit (robocarul efectuează o deplasare), se salvează comanda corespunzătoare în memorie la adresa indicată de pointerul **c**. De asemenea se salvează în memorie informațiile primite de la traductoarele corespunzătoare MCC și MPP la adresele indicate de pointerii **d** și **p**. Dacă robocarul este oprit, se poziționează bitul **învățare** pe "1" logic, iar numărul de impulsuri furnizate de către TIRO nu este salvat în memorie. În momentul în care se va primi o comandă pentru deplasare, se va stabili dacă s-a efectuat o rotire. Dacă rotirea a avut loc, se salvează în memorie la adresa indicată de pointerul **c** comanda de virare respectivă, iar numărul de impulsuri furnizate de către TIRO al MPP se salvează la adresa de memorie indicată de pointerul **p**. La adresa de memorie indicată de pointerul **d** este salvată valoarea 0, deoarece nu s-a efectuat deplasarea înainte sau înapoi.

Dacă comanda transmisă este **O** (schema logică din figura 3.10.), atunci se verifică valoarea bitului **stop**. Dacă robocarul a fost oprit (**stop** este "1" logic), nu se salvează această comandă în memorie, deoarece este redundantă (anterior a fost dată încă o comandă de oprire). Dacă robocarul nu a fost oprit, se salvează în memorie comanda de oprire și apoi este apelată funcția **oprit()**.

În cazul comenzii **Q**, se verifică din nou bitul **stop**. Dacă robocarul nu a fost oprit (valoarea bitului de **stop** este "1" logic), atunci se citesc din memorie numerele de impulsuri provenite de la TIRO și se salvează la adresele corespunzătoare de memorie, iar apoi se iese din bucla de învățare.

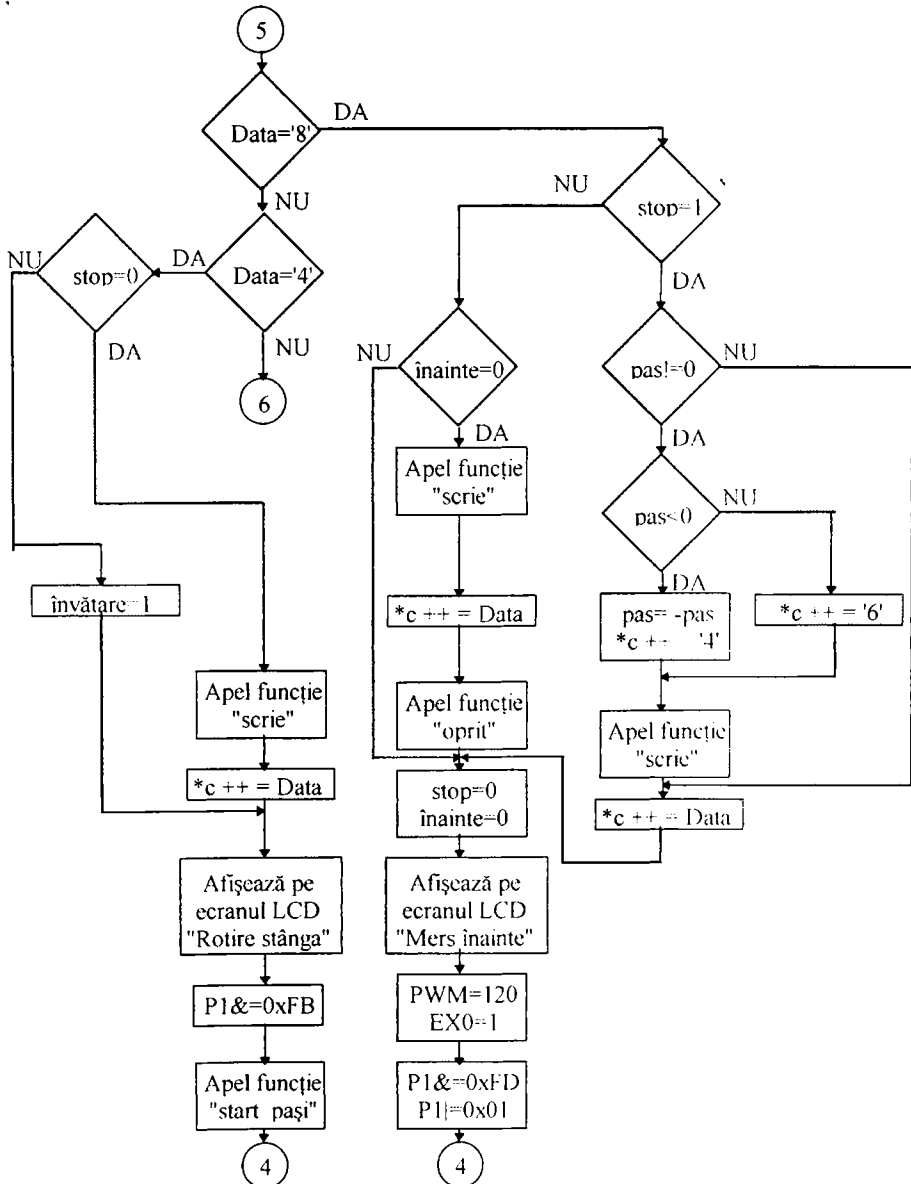


Figura 3.11.

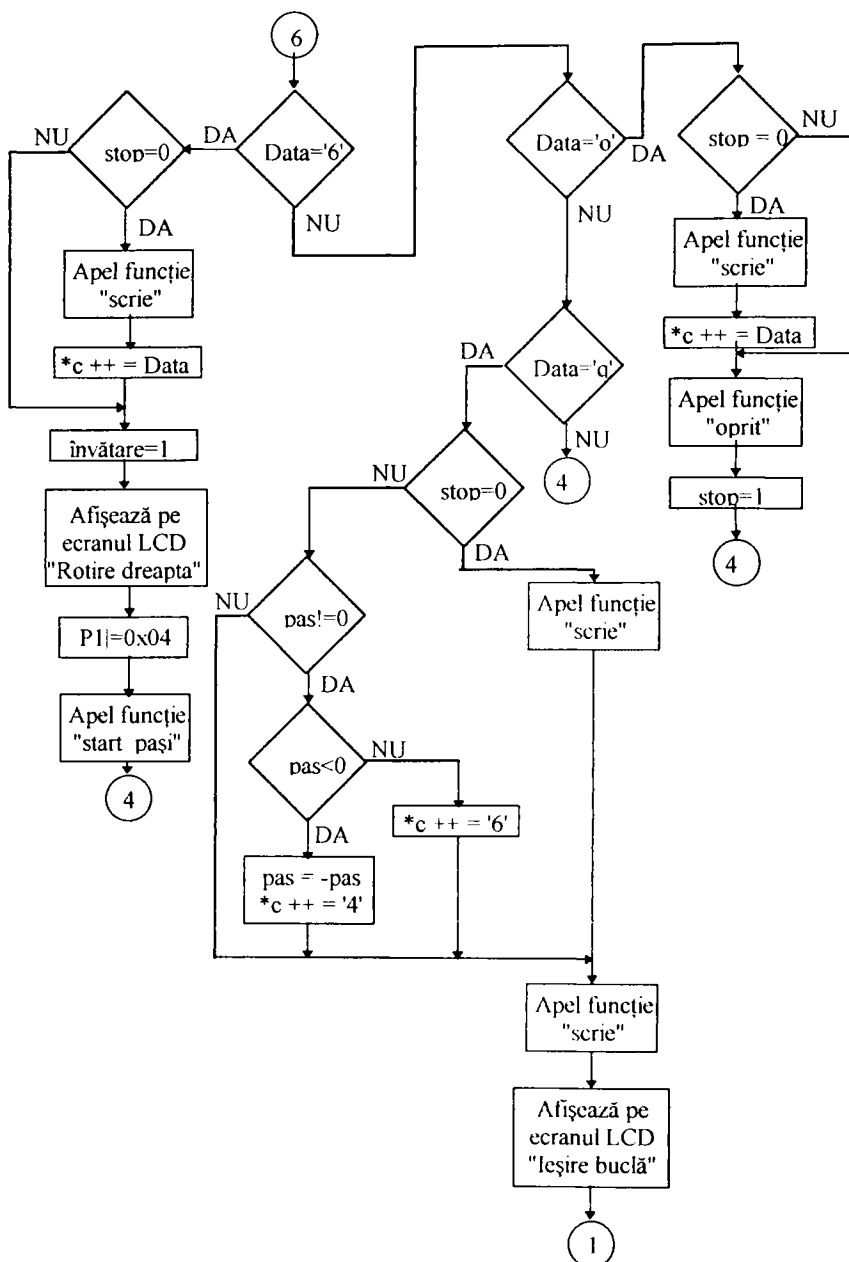


Figura 3.12.

Dacă robocarul a fost oprit, se compară variabila **pas** (care reprezintă numărul de impulsuri provenite de la TIRO corespunzător MPP) cu 0.

Dacă aceasta este 0, rezultă că nu se va salva nimic în memorie și se iese din bucla de învățare. Dacă variabila **pas** este însă diferită de 0, are loc detecția sensului în care s-a rotit MPP. Această procedură de detecție a fost descrisă anterior. Apoi se iese din bucla de învățare.

### 3.2.5. Deplasarea pe o traiectorie *învățată*.

La redarea traiectoriei se citește din memorie comanda aflată la adresa indicată de pointerul **c**.

Corespunzător comenzii citite se vor realiza următoarele operații:

- ◆ dacă în memorie se află comanda **oprit**, codul caracterului **O**, se va apela funcția **oprit()** și nu se va efectua altă operație;
- ◆ dacă în memorie se află una dintre celelalte comenzi (**2** - deplasare înapoi, **8** - deplasare înainte, **4** - rotire stânga un grad, **6** - rotire dreapta un grad), se vor citi din memorie numărul de impulsuri prescris pentru deplasare (de la adresa indicată de către pointerul **d**) și numărul de impulsuri prescris pentru virare (de la adresa indicată de către pointerul **p**).

După citirea acestor informații din memorie, se vor comanda corespunzător elementele de execuție ale MPP sau MCC, până la epuizarea numărului de impulsuri prescrise, moment în care se va citi din memorie noua comandă care urmează a fi executată. Această procedură se repetă până când sunt epuizate toate comenzile memorate (până la prima comandă nulă, adică codul 0). Comenzile sunt citite din memorie de la adresele prezentate în tabelul 3.3.:

**Tabelul 3.3.**

*c	c1	c2	c3	c4	
*p	p1	p2	p3	p4	
*d	d1	d2	d3	d4	

Algoritmul de redare este ilustrat în schemele logice din figurile 3.13, 3.14.

Dacă există o comandă memorată (**\*c** ≠ **0**), se va compara această comandă cu comenzile cunoscute **2**, **4**, **6**, **8**, **O**, pentru a se determina tipul comenzii, iar apoi aceasta va fi executată corespunzător.

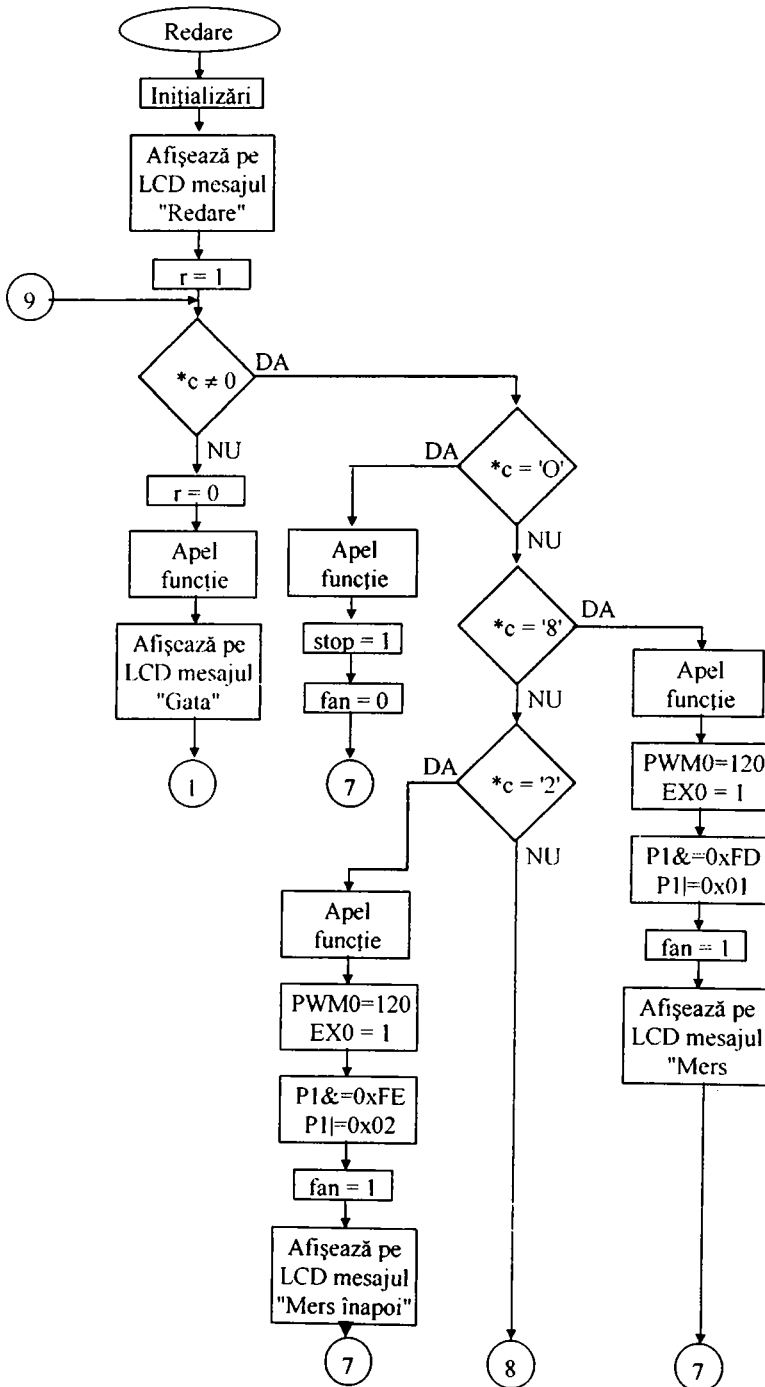


Figura 3.13.

În momentul în care s-au epuizat comenzile din memorie (se întâlnește prima comandă nulă), se întrerupe bucla algoritmului de redare și se revine în programul principal.

Redarea unui traseu învățat depinde foarte mult de precizia cu care a fost poziționat robocarul în punctul din care a început învățarea.

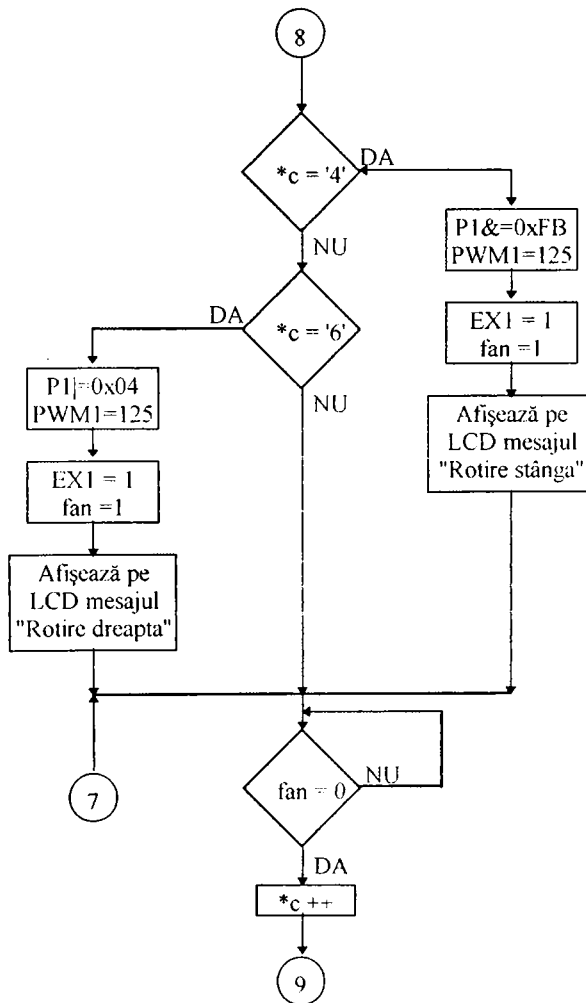


Figura 3.14.

### 3.3. Concluzii

Traductoarele incrementale rotative optice (TIRO) sunt elementele senzoriale care echipează în majoritatea cazurilor axele unui robot industrial. Informațiile achiziționate de către sistemul de conducere al RI cu ajutorul acestor traductoare sunt utilizate pentru obținerea mărimilor de reacție în cadrul buclelor de reglare a vitezei și poziției. Robocarele fiind vehicule care se deplasează într-un spațiu de lucru de dimensiuni mari, necesită sisteme senzoriale pentru determinarea situații. O metodă utilizată curent în acest scop este **odometria**. Aceasta se bazează pe măsurarea spațiului parcurs și a unghiurilor de virare prin utilizarea informațiilor prelevate cu ajutorul TIRO. Dacă în sistemul de virare nu apar erori datorate alunecărilor, în sistemul de tracțiune, în schimb, erorile sunt inevitabile, fiind o consecință a uzurilor mecanice ale roților, a alunecărilor sau a variațiilor sarcinii transportate.

**Situarea robocarului într-un punct al traiectoriei se determină în raport cu situarea acestuia în punctul precedent.** Deoarece fiecare determinare se face eronat din motivele prezentate mai sus, rezultă că erorile au un efect cumulativ și cresc odată cu creșterea spațiului parcurs.

Odometria fiind o metodă larg răspândită, **autorul a considerat necesară elaborarea unui program pentru măsurarea distanțelor parcurse de către robocar și a unghiurilor de virare ale acestuia**, mărimi folosite în calculul coordonatelor robocarului și a orientării axei vehiculului la un moment dat.

Utilizând **programul conceput și realizat în întregime de către autor**, se poate opta pentru unul dintre cele patru moduri de deplasare a vehiculului și anume:

- ◆ deplasare prin comandă manuală;
- ◆ deplasare pe traiectorie impusă;
- ◆ învățarea unei traiectorii;
- ◆ deplasarea pe o traiectorie învățată.

În continuare se vor prezenta în rezumat cele patru moduri de funcționare.

**Deplasarea prin comandă manuală.** Acest mod de funcționare are loc prin transmiterea unor comenzi sistemului de conducere de către

operatorul uman prin intermediul tastaturii. Dacă se apasă una dintre tastele care transmit comenzile pentru deplasarea înainte sau înapoi, robocarul se va deplasa conform acestor comenzi până la primirea comenzii de oprire. Comanda de oprire este aceeași în orice mod de funcționare. Nu este necesară numărarea impulsurilor obținute cu ajutorul TIRO al MCC.

Rotirea sistemului de virare are loc tot pe baza comenzilor manuale, dar impulsurile furnizate de către TIRO al MPP sunt numărate de către sistemul de conducere pentru a se evita depășirea limitelor impuse mecanismului de virare.

**Deplasarea pe traiectoria impusă.** Se introduc în memoria sistemului de conducere următoarele date: sensul deplasării, distanța care se va parcurge (exprimată în cm), sensul de rotire al mecanismului de virare și unghiul de virare (exprimat în grade). Cu ajutorul funcției **profil** se stabilește care dintre cele două profile de viteză se va utiliza. Această decizie se ia pe baza mărimii traiectoriei care urmează a fi parcursă. În continuare se generează semnalul PWM pentru comanda elementului de execuție al MCC și se numără impulsurile furnizate de către TIRO, iar deplasarea are loc până la atingerea numărului de impulsuri prescris.

Sensului de virare îi este asociată mărimea unghiului de virare. Pe baza acestor mărimi are loc virarea, pe durata căreia se numără impulsurile furnizate de către TIRO al MPP. Unghiul de virare nu poate depăși o limită maximă admisă stabilită prin soft.

**Învățarea unei traiectorii.** În acest mod de lucru, comenzile pentru virare sau deplasare sunt introduse manual de către operatorul uman. Ele sunt memorate de către sistemul de conducere. Totodată se memorează și numărul de impulsuri obținute cu ajutorul celor două traductoare. Se face mențiunea că **toate comenzile sunt executate, dar sunt învățate numai cele utile.**

**Deplasarea pe o traiectorie învățată.** Funcționarea decurge ca în modul **deplasare pe o traiectorie impusă**, cu mențiunea că parametri mișcărilor nu sunt prescrși de către operatorul uman, ci sunt obținuți prin învățare.

**Fiecare dintre aceste moduri a fost studiat în prealabil cu minuțiozitate, au fost analizate soluțiile existente și în final s-a elaborat o variantă care corespunde scopului propus.** Pe baza



experimentărilor s-au optimizat toate programele, astfel încât, varianta propusă constituie, după părerea autorului, soluția optimă.

**Elementele de originalitate sunt prezentate în cele ce urmează.**

Pentru generarea semnalelor de comandă necesare elementelor de execuție care alimentează cele două motoare (MCC și MPP) s-a ales microcontrolerul 80C552, datorită faptului că dispune de două ieșiri PWM. Semnalele de comandă pot fi generate simultan și nu sunt necesare două microcontrolere.

Un alt element de originalitate îl constituie faptul că programul funcționează pe întreruperi (INT0 impulsurile provenite de la TIRO al MCC, INT1 impulsurile provenite de la TIRO al MPP) și astfel se poate prelua în orice moment controlul asupra robocarului, dacă acesta efectuează o operație neprevăzută.

Inițializarea sistemului de virare al robocarului pe direcția de deplasare rectilinie se face pe baza unui algoritm care reprezintă de asemenea un element de noutate. Limitele de virare la stânga, respectiv la dreapta ale mecanismului de virare al robocarului sunt stabilite prin soft ( $\pm 60^\circ$ ), pentru a se evita eventualele deteriorări ale acestui mecanism.

Originalitatea algoritmului de învățare proiectat constă în faptul că memorarea datelor are loc numai atunci când se primește o nouă comandă de deplasare sau de virare. Această soluție a fost aleasă având în vedere necesitatea reducerii spațiului de memorie RAM utilizat.

În timpul instruirii, anumite comenzi pot fi eronate. Pentru evitarea învățării acestora, cu o ocupare inutilă a memoriei și pentru ca robocarul să nu execute manevre care nu sunt necesare, a fost elaborat un algoritm special.

Prin utilizarea algoritmului de învățare prezentat se realizează o comprimare maximă a comenzilor primite, fiind salvate în memorie doar comenzile utile.

**Verificarea programului s-a efectuat cu ajutorul sistemului de comandă și al robocarului construit în întregime de către autor. Experimentele realizate au confirmat corectitudinea programului conceput și sunt prezentate în cadrul tezei.**

## Capitolul 4

### CONTRIBUȚII ADUSE METODELOR UTILIZATE PENTRU DETERMINAREA SITUAȚII ABSOLUTE A ROBOCARELOR.

Metodele utilizate pentru determinarea situații absolute a roboților mobili și a vehiculelor ghidate automat asigură instrumentul necesar determinării coordonatelor și a orientării acestora în raport cu un sistem de referință care poate fi detectat cu ajutorul sistemelor senzoriale disponibile. Pentru determinarea situații robocarul va fi echipat cu:

- ◆ un sistem de conducere cu performanțe foarte bune;
- ◆ un sistem senzorial evoluat;
- ◆ echipamentul necesar prelucrării informațiilor obținute cu ajutorul senzorilor;
- ◆ setul de programe adecvate;

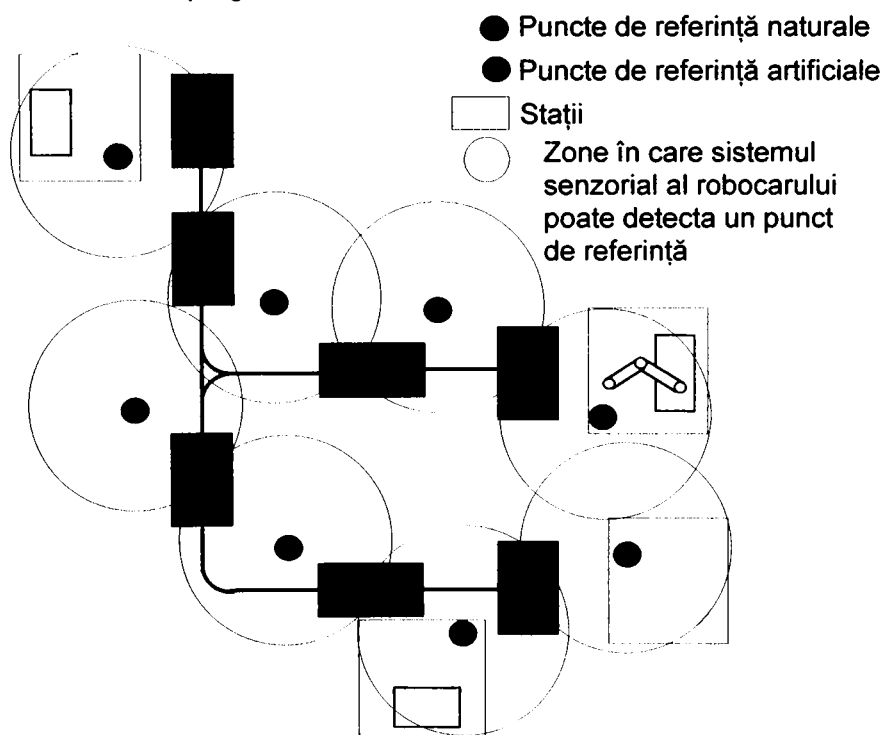


Figura 4.1.

Sistemele de referință sunt construite prin atribuirea unor axe de coordonate fiecărui punct de referință. Aceste puncte aparțin obiectelor aflate în imediata vecinătate a traiectoriei parcurse. În lipsa obiectelor, sau dacă acestea nu sunt suficiente, punctele de referință pot fi amplasate în spațiul de lucru de către proiectantul sistemului de transport sau de către utilizator.

Determinarea situații absolute a vehiculului poate avea loc în mod continuu. În acest caz, sistemul senzorial al robocarului care parcurge traiectoria, poate detecta în orice moment un punct sau un obiect care are coordonatele cunoscute în raport cu sistemul propriu de referință și implicit în raport cu sistemul de referință universal. Față de acestea, sistemul de conducere determină situarea vehiculului. În figura 4.1 se prezintă dispunerea sistemelor de referință de-a lungul traiectoriei [HEL 96]. Deoarece amplasarea unor marcaje artificiale este costisitoare, iar recunoașterea obiectelor aflate în spațiul de lucru este dificilă, se poate utiliza o metodă rezultată prin combinarea determinării odometrice a situații cu determinarea situații absolute. În acest caz sunt necesare mai puține repere. Metoda este denumită în literatura de specialitate **referențiere discretă** [SCH 96] , [HEL 96].

*În capitolul de față se prezintă principiul referențierii discrete, câteva metode cunoscute [HEL 96] utilizate pentru determinarea situații și câteva metode originale, care constituie contribuții ale autorului. În încheierea capitolului se vor compara metodele prezentate pe baza unor criterii stabilite de către autor, indicându-se avantajele și dezavantajele aplicării lor.*

#### **4.1.Principiul referențierii discrete.**

Robocarul se deplasează între cele patru stații (figura 4.1). Dacă deplasarea ar avea loc numai pe baza odometriei, ar rezulta abateri ale vehiculului de la traiectoria prescrisă. Aceste abateri sunt cu atât mai mari, cu cât distanța parcursă este mai mare. În stații, vehiculul trebuie să ajungă în poziția stabilită, altfel nu poate avea loc procesul de transfer a sarcinii transportate. De asemeni, dacă în stație are loc încărcarea acumulatorilor, nu se poate realiza cuplarea vehiculului cu echipamentul de încărcare. Pentru evitarea acestor situații, nu este suficientă numai determinarea cu precizie a situații, ci și corectarea permanentă a traiectoriei. Corecția se face pe baza punctelor de referință, amplasate la intervale determinate pe baza criteriului abaterii minime. Punctele de referință sunt detectate cu ajutorul senzorilor, iar

sistemul de conducere generează o **trajectorie de corecție**, astfel încât în stație să se obțină situaarea dorită cu o eroare cât mai mică. Dacă precizia de situaare necesară nu este obținută, are loc o indexare mecanică în cadrul stației.

La baza acestei metode stă atribuirea unor sisteme de axe de coordonate fiecărui punct de referință. Vehiculul se deplasează pe rând în fiecare sistem de axe de coordonate, nefiind necesar un sistem de referință absolut. Punctele de referință se află de-a lungul trajectoriei parcurse și pot fi naturale sau artificiale. Se definește un sistem de axe de coordonate fix  $R_i$  și unul solidar cu vehiculul  $F_p$  (figura 4.2).

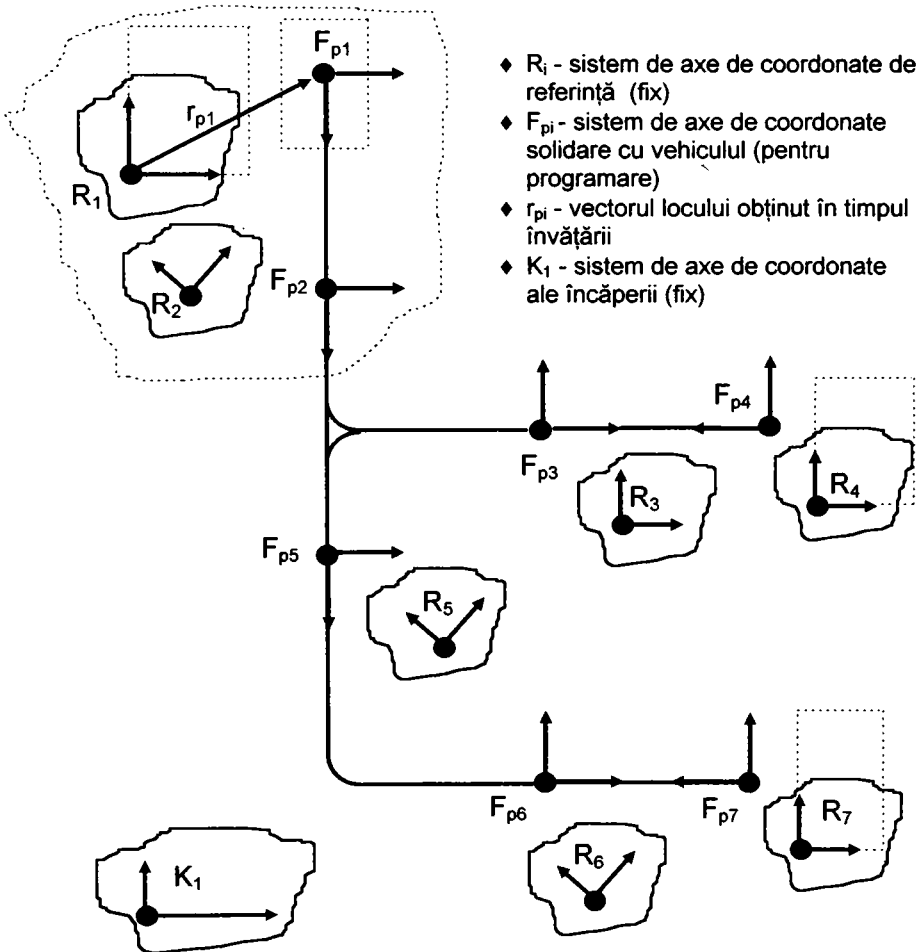


Figura 4.2.

Programatorul stabilește în timpul fazei de instruire puncte singulare ajutoare de-a lungul traiectoriei, între care va avea loc interpolarea pe durata deplasării automate a vehiculului. Cu ajutorul sistemului senzorial se determină în fiecare punct de referință vectorul  $r_{pi}$  și orientarea sa de la  $R_i$  la  $F_{pi}$ . Prin măsurarea cu ajutorul traductoarelor incrementale a spațiilor parcurse în timpul deplasării spre  $R_{i+1}$ , se determină coordonatele punctelor care aparțin sistemului de referință  $R_i$  și se memorează. Atunci când vehiculul a ajuns în  $R_{i+1}$ , urmează o nouă măsurare cu ajutorul senzorilor și se determină un nou vector  $r_{pi+1}$ . Această procedură se repetă până când vehiculul ajunge în punctul final.

Avantajul metodei prezentate față de alte metode [KUG 90], [WIK 88], [FOJ 91], [VOG 92] constă în faptul că nu este necesar ca marcajele (punctele de referință) să fie amplasate în mediu cu ajutorul unor instalații de măsură complexe (teodolit), deoarece ele vor fi percepute în faza de instruire. Alt avantaj este acela că erorile care apar în timpul măsurărilor odometrice nu se cumulează. Ele au efect numai în timpul deplasării între punctele de referință. Pe baza programului de generare a traiectoriei vehiculului și pe baza datelor obținute cu ajutorul sistemelor senzoriale se calculează mărimile de conducere necesare deplasării pe traiectorie. Generarea valorilor prescrise ale coordonatelor punctelor care alcătuiesc traiectoria are loc în modul prezentat anterior. În figura 4.3 se prezintă traiectoria prescrisă și traiectoria reală, precum și modalitatea de corectare a ei.

- ◆  $F_{li+1}$  - sistem de axe de coordonate al vehiculului
- ◆  $r_{li+1}$  - vectorul locului de la  $R_{i+1}$  la  $F_{li+1}$  obținut on line
- ◆ ——— traiectoria reală
- ◆ - - - - - traiectoria prescrisă

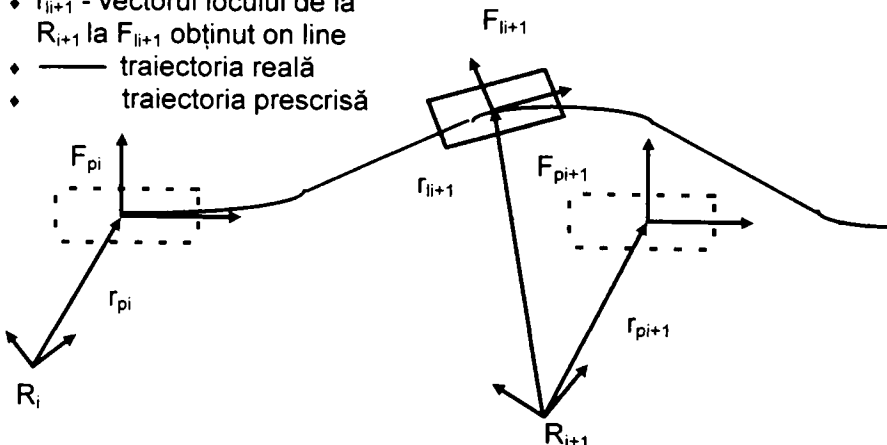


Figura 4.3.

Modul în care are loc generarea traiectoriei de corecție este prezentat schematic în figura 4.4.

Sarcina sistemului de conducere este înlăturarea abaterii vehiculului de la traiectoria prescrisă cu ajutorul unei traiectorii de corecție, adaptată la condițiile limită cinematice și dinamice ale vehiculului.

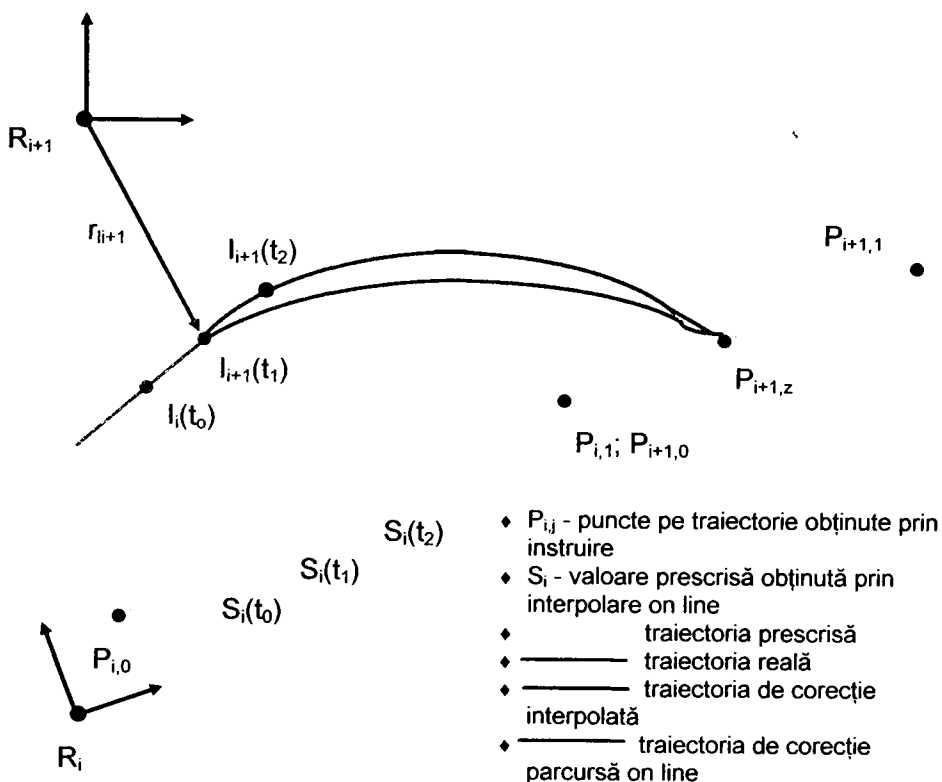


Figura 4.4.

Se notează cu  $P_{i,j}$  un punct oarecare utilizat în procesul de instruire, aflat pe traiectoria prescrisă, cu  $S_i(t_k)$ ,  $k=0..n$  un punct corespunzător momentului  $t_k$ , aflat pe aceeași traiectorie și cu  $l_i(t_k)$  un punct aflat pe traiectoria reală în același moment. Toate punctele menționate se află în sistemul de referință  $R_i$ . Referențierea discretă va decurge după cum urmează.

Pentru măsurarea coordonatelor robocarului raportate la sistemul de referință, se activează senzorii la momentul  $t_0$ , corespunzător unei anumite distanțe dintre robocar și punctul  $R_{i+1}$ . Interpolatorul care

generează valoarea prescrisă pentru segmentul de traiectorie  $P_{i,0}$ ,  $P_{i,1}$  se află în punctul  $S_i(t_0)$ . Datorită abaterii de la traiectoria prescrisă, vehiculul se află în punctul  $I_i(t_0)$ , aflat pe traiectoria reală, ale cărui coordonate nu sunt cunoscute.

Dacă măsurarea coordonatelor vehiculului aflat pe traiectoria reală în punctul  $I_{i+1}(t_1)$ , în raport cu sistemul de referință  $R_{i+1}$ , are loc în momentul  $t_1$ , coordonatele vectorului sunt  $r_{i+1} = (x_F, y_F, \gamma_F)$ , iar interpolatorul valorii prescrise a ajuns în punctul  $S_i(t_1)$ .

Alături de coordonatele punctului  $I_{i+1}(t_1)$ , aflat pe traiectoria reală, obținute on line, sunt cunoscute coordonatele punctului de instruire  $P_{i+1,0}$ , care corespunde fizic punctului  $P_{i,1}$ , aflat pe traiectoria prescrisă, cât și coordonatele punctului de instruire  $P_{i+1,1}$  în raport cu sistemul de referință  $R_{i+1}$ .

Prin deplasarea pe traiectoria de corecție, sistemul de conducere al robocarului va aduce vehiculul pe traiectoria prescrisă. Pentru aceasta se definește punctul  $P_{i+1,z}$ , care poate fi indicat în prealabil de către utilizator sau este ales automat de către sistemul de conducere în funcție de distanța dintre punctele  $P_{i+1,0}$ ,  $P_{i+1,1}$ , de condițiile limită dinamice ale acționărilor și de spațiul disponibil pentru deplasare.

Nu trebuiește neglijat nici timpul necesar pentru procesarea datelor furnizate de către senzori și utilizate în calculul mărimilor de conducere pe traiectoria de corecție. Deoarece vehiculul se află în mișcare, el parcurge un anumit spațiu în acest interval de timp. Dacă timpul de procesare este  $\Delta t = t_2 - t_1$ , vehiculul începe deplasarea pe traiectoria de corecție atunci când se află în punctul  $I_{i+1}(t_2)$ . Mărimea de conducere utilizată în acest moment corespunde însă punctului  $I_{i+1}(t_1)$ , astfel încât, după parcurgerea traiectoriei de corecție, rămâne o abatere a traiectoriei reale față de traiectoria prescrisă corespunzătoare diferenței dintre mărimile de conducere calculate pentru punctele  $I_{i+1}(t_1)$  și  $I_{i+1}(t_2)$ . Din momentul  $t_0$  se memorează de către sistemul de conducere coordonatele punctelor aflate pe traiectoria prescrisă și pe cea reală. Valorile prescrise în intervalul de timp  $t_1 \div t_2$  se transferă în sistemul de coordonate  $R_{i+1}$  și sunt utilizate pentru planificarea traiectoriei de corecție. Astfel sfârșitul acestei traiectorii va fi în punctul  $P_{i+1,z}$ .

În modul automat de funcționare, prin utilizarea sistemelor de referință pentru determinarea situații absolute, nu apar erori cumulative.

Intervalele dintre două sisteme de referință se vor putea determina ținând cont de precizia determinării situații în aceste sisteme de referință și de precizia măsurării deplasării și a orientării (odometria) până la următorul sistem de referință.

## 4.2. Metode utilizate pentru determinarea situații absolute a vehiculelor ghidate automat

Așa cum s-a arătat, determinarea situații unui vehicul în raport cu sistemele de referință asociate unor repere naturale sau artificiale are o importanță deosebită în cadrul procesului de conducere a acestuia. În continuare se vor prezenta câteva metode care utilizează marcaje artificiale și senzorii adecvați acestor metode.

Distanța maximă dintre punctele de referință depinde de precizia detectării marcajelor și de abaterea maximă admisă a traiectoriei. Dar sunt și alți factori care influențează această distanță, cum ar fi: calitatea pardoselii, variațiile sarcinii transportate, etc.

Eroarea de măsurare a orientării vehiculului este preponderentă față de eroarea de măsurare a spațiului parcurs, ducând la abateri ale traiectoriei care nu pot fi compensate numai prin odometrie.

În analiza care urmează se va obține expresia pentru orientarea vehiculului (unghiul  $\gamma_F$ ) și se va calcula eroarea de măsurare a unghiului  $\gamma_F$  ( $\Delta\gamma_F$ ). Pe baza acestui rezultat se stabilește distanța maximă dintre punctele de referință.

### 4.2.1. Determinarea situații vehiculelor cu ajutorul senzorilor ultrasonici

Senzorii ultrasonici echipează în mod curent mijloacele de transport automat. Utilizarea pe scară largă a acestor senzori se datorează prețului de cost scăzut, simplității măsurării distanțelor și complexității reduse a echipamentelor electronice necesare procesării informațiilor achiziționate cu ajutorul lor.

Determinarea coordonatelor unui punct mobil oarecare în raport cu un sistem de referință prin utilizarea senzorilor ultrasonici are loc prin calculul distanțelor dintre acest punct și două puncte fixe care au coordonatele cunoscute. Dacă punctul mobil este tocmai punctul



caracteristic al robocarului, coordonatele vehiculului sunt cunoscute. Nu se cunoaște însă unghiul sub care este orientată axa robocarului față de o axă a sistemului de referință (de obicei axa  $x$ ). Pentru determinarea acestui unghi sunt necesare două puncte situate pe robocar sau un sistem senzorial foarte directiv. Deoarece senzorii ultrasonici nu au directivitate foarte bună, metodele utilizate pentru determinarea situații cu ajutorul lor trebuie să elimine acest inconvenient.

#### 4.2.1.1. Senzori ultrasonici și panouri.

Metoda analizată se bazează pe determinarea distanțelor dintre anumite puncte aparținând robocarului și reperate care au coordonatele cunoscute, aflate în spațiul de lucru [HEL96]. În figura 4.5 se prezintă modul în care pot fi determinate, prin măsurarea a trei distanțe, coordonatele vehiculului ( $x_F$ ,  $y_F$ ) și orientarea sa ( $\gamma_F$ ). Panourile sunt amplasate în diverse puncte, aflate în spațiul de lucru și reprezintă sistemele de referință  $R_i$ .

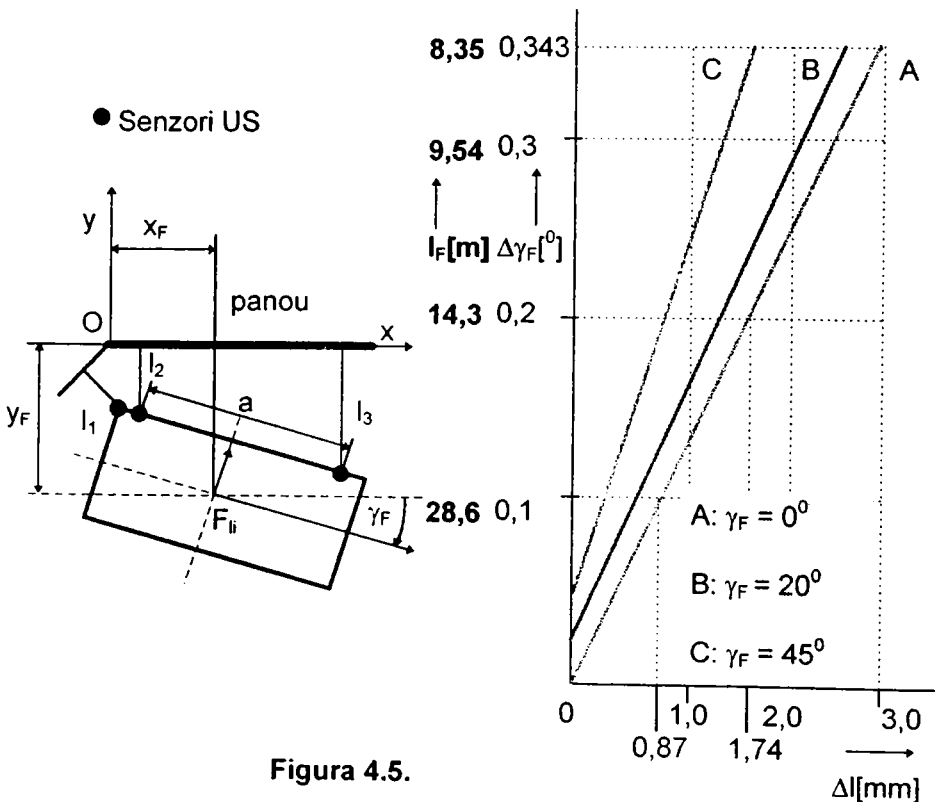


Figura 4.5.

Nu este suficientă numai măsurarea distanțelor  $l_2$ ,  $l_3$ , deoarece acestea determină orientarea și coordonata  $y_F$ . Pentru determinarea coordonatei  $x_F$  este necesară și măsurarea distanței  $l_1$ . Din acest motiv panoul are forma din figura 4.5. Orientarea vehiculului  $\gamma_F$  rezultă din relațiile următoare:

$$\sin \gamma_F = \frac{l_3 - l_2}{a} \quad (4.1)$$

$$\gamma_F = \arcsin \frac{l_3 - l_2}{a} \quad (4.2)$$

Utilizând pentru eroarea maximă relația:

$$\Delta f = \sum_{i=1}^k \left| \frac{\delta f}{\delta x_i} \right| \Delta x_i \quad (4.3)$$

rezultă pentru eroarea de măsurare a deviației maxime:

$$\Delta \gamma_F = \frac{1}{\sqrt{1 - \left( \frac{l_3 - l_2}{a} \right)^2}} \left[ \frac{l_3 - l_2}{a^2} \Delta a + \frac{1}{a} (\Delta l_3 + \Delta l_2) \right] \quad (4.4)$$

Dacă erorile de măsurare ale distanțelor  $l_2$  și  $l_3$  se consideră egale  $\Delta l_2 = \Delta l_3 = \Delta l$ , la o distanță  $a = 1$  m între senzori (eroarea de montare  $\Delta a = 1$  mm), în funcție de  $\Delta l$  se obțin valorile din figură pentru  $\Delta \gamma_F$  și pentru deplasarea autonomă,  $l_F$ . Abaterea admisă de la traiectoria prescrisă, din relația (5.1) se consideră  $\Delta S = \pm 50$  mm. *Distanța dintre senzori considerată în calcul,  $a = 1$  m, corespunde robocarului construit de către autor.*

Se observă din figura 4.5 că pentru o eroare de măsurare a distanțelor  $\Delta l \leq 3$  mm se pot obține deplasări autonome ale robocarului de până la 8,35 m în funcție de orientarea acestuia ( $\gamma_F$ ). Măsurările trebuie efectuate pe cât posibil atunci când vehiculul este orientat paralel cu panoul ( $\gamma_F \cong 0^\circ$ ), altfel eroarea de măsurare crește foarte mult. Deoarece măsurarea se face cu ultrasunete și acestea au viteza de propagare în aer redusă, aplicarea acestui principiu este indicată numai atunci când vehiculul staționează. Costurile pentru instalarea panourilor sunt mari, dar echipamentul necesar măsurării, aflat pe robocar, este ieftin.

#### 4.2.1.2. Determinarea situații robocarului prin utilizarea a doi emițători și doi receptori ultrasonici.

În cadrul acestei metode, prezentată în literatură [PRU 89], se utilizează un singur emițător montat pe vehicul și doi receptori ultrasonici fiși. Prin măsurarea distanțelor dintre emițător și cei doi receptori, fiind

cunoscută distanța dintre receptori, se pot calcula coordonatele vehiculului. Dezavantajul metodei este acela că orientarea vehiculului nu poate fi determinată. **Autorul propune înlăturarea acestui dezavantaj prin completarea echipamentului cu încă un emițător ultrasonic.** Amplasarea elementelor este prezentată în figura 4.6. Cei doi emițători sunt amplasați pe robocar în punctul A, respectiv B, iar receptorii se află în punctul M, respectiv N, care aparțin spațiului de lucru. Procedeeul de calcul este asemănător cu cel descris în paragraful 4.2.6, cu mențiunea că nu are loc măsurarea unghiurilor, ci numai a distanțelor.

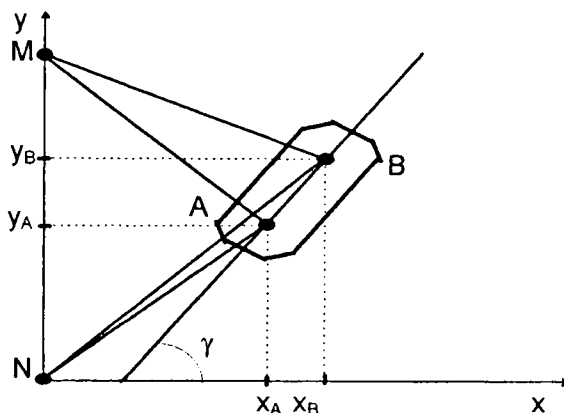


Figura 4.6.

Fiecare emițător generează unde sonore ultraacustice care au frecvențele diferite, corespunzător fiecărui receptor. Astfel este posibilă identificarea sursei din care acestea provin. În momentul emisiei începe măsurarea timpului în care unda ultrasonoră parcurge distanța MA. Când unda a fost recepționată, se încheie procesul de măsurare. Apoi se determină pe rând timpii corespunzători distanțelor MB, NA, NB. După calculul acestor distanțe se determină coordonatele și orientarea vehiculului.

$$\begin{cases} y_A = \frac{d^2 + NA^2 - MA^2}{2d} \\ x_A^2 = NA^2 - y_A^2 \end{cases} \quad (4.5)$$

$$\begin{cases} y_B = \frac{d^2 + NB^2 - MB^2}{2d} \\ x_B^2 = NB^2 - y_B^2 \end{cases} \quad (4.6)$$

$$\gamma = \frac{x_B - x_A}{y_B - y_A} \quad (4.7)$$

Dezavantajul metodei descrise este, ca și în cazul altor metode bazate pe măsurarea distanțelor cu ultrasunete, durata mare necesară parcurgerii de către undele US a distanțelor care se măsoară. Prin procedeul amintit, timpul acesta este redus la jumătate față de alte metode [AND85], deoarece nu se așteaptă recepționarea unei reflectate. Spațiul măsurat în cazul de față este parcurs de către unda ultrasonoră într-un singur sens.

#### 4.2.2. Determinarea situației cu ajutorul fasciculelor laser.

Utilizarea fasciculelor laser oferă metodelor de determinare a situației avantajul unei precizii de determinare a coordonatelor mai bune decât în cazul utilizării senzorilor ultrasonici. Cu ajutorul acestor fascicule se pot determina distanțele dintre emițător și obiectul care le reflectă, prin măsurarea defazajului dintre unda directă și unda reflectată. Echipamentul utilizat pentru această măsurare are o complexitate mare și un preț ridicat.

Având în vedere posibilitatea realizării, prin utilizarea unor sisteme optice, a unui fascicul îngust și foarte directiv, se pot măsura cu precizie și valorile anumitor unghiuri. Acestea sunt folosite în determinarea situației vehiculelor. Devierea fasciculului în plan vertical și orizontal trebuie să se realizeze cu mijloace care asigură o viteză de baleiere ridicată, pentru ca achiziționarea informațiilor prelevate cu ajutorul acestui sistem să nu necesite o durată mare.

##### 4.2.2.1. Scaner laser cu un reflector.

Principiul metodei rezultă din figura 4.7 [ERW 95]. Scanerul laser instalat pe vehicul baleiază, prin intermediul unei oglinzi rotitoare, un domeniu de  $60^\circ$ . Spotul este mai lat pe verticală, astfel încât denivelările pardoselii și eventualele oscilații ale vehiculului să nu influențeze măsurarea.

Reflectorului, montat lateral față de traiectorie, i se asociază sistemul de coordonate  $R_i$ . Reflectorul prismatic are în componența sa trei prisme, care au o construcție specială. Prismele  $pr_1$  și  $pr_2$  reflectă fasciculele paralel cu direcția fasciculelor incidente, pe când prisma  $pr_3$  reflectă fasciculul numai dacă incidența are loc sub un unghi de  $90^\circ$ . Astfel sunt disponibile, pentru determinarea coordonatelor vehiculului, trei unghiuri:  $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$ . Valorile acestor unghiuri se determină prin memorarea valorilor unghiurilor cu care se rotește oglinda, măsurate în momentul în care se recepționează fasciculul reflectat.

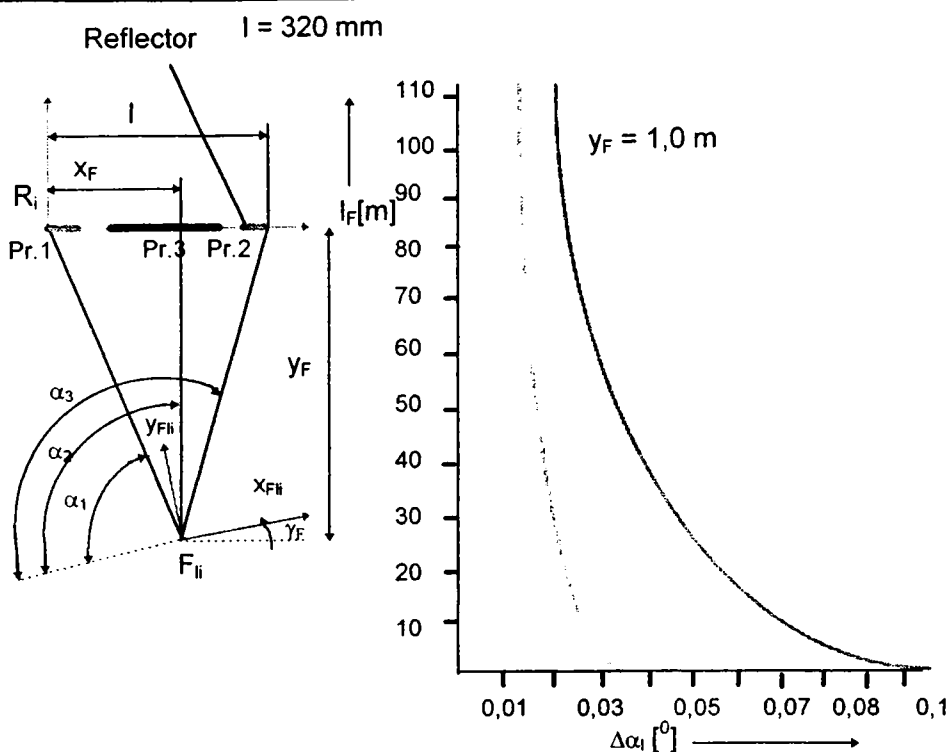


Figura 4.7.

$$\gamma_F = \frac{\pi}{2} - \alpha_2 \quad (4.8)$$

$$x_F = l \frac{\operatorname{tg}(\alpha_2 - \alpha_1)}{\operatorname{tg}(\alpha_3 - \alpha_2) + \operatorname{tg}(\alpha_2 - \alpha_1)} \quad (4.9)$$

$$y_F = \frac{x_F}{\operatorname{tg}(\alpha_2 - \alpha_1)} \quad (4.10)$$

Cu ajutorul relației (4.4), aplicată mărimilor de mai sus se obține:

$$\Delta \gamma_F = \Delta \alpha_2 \quad (4.11)$$

$$\Delta x_F = l \left( \frac{\operatorname{tg}(\alpha_3 - \alpha_2)}{\cos^2(\alpha_2 - \alpha_1) [\operatorname{tg}(\alpha_3 - \alpha_2) + \operatorname{tg}(\alpha_2 - \alpha_1)]^2} \Delta(\alpha_2 - \alpha_1) + \frac{\operatorname{tg}(\alpha_2 - \alpha_1)}{\cos^2(\alpha_3 - \alpha_2) [\operatorname{tg}(\alpha_3 - \alpha_2) + \operatorname{tg}(\alpha_2 - \alpha_1)]^2} \Delta(\alpha_3 - \alpha_2) \right) \quad (4.12)$$

$$\Delta y_F = \frac{1}{\operatorname{tg}(\alpha_2 - \alpha_1)} \Delta x_F + \frac{x_F}{\sin^2(\alpha_2 - \alpha_1)} \Delta(\alpha_2 - \alpha_1) \quad (4.13)$$

Pentru deplasarea autonomă se obține:

$$I_F = \frac{\Delta I_s - \Delta y_F}{\operatorname{tg} \Delta \gamma_F} \quad (4.14)$$

Influența preponderentă asupra acestei mărimi o are  $y_F$ .

Instalațiile fixe necesare acestei metode sunt mai costisitoare decât cele utilizate în cazul metodelor prezentate anterior. Avantajul este însă faptul că măsurarea poate avea loc atât în mers, cât și în timpul staționării, ceea ce face ca metoda să poată fi utilizată pentru poziționări care necesită o precizie ridicată. Sistemul este însă sensibil la denivelările pardoselii, care pot face ca raza laser să nu atingă reflectorul. De aceea se iau măsuri pentru menținerea fixă a poziției scanerului, utilizând în acest scop echipamente speciale.

#### 4.2.2.2. Scanner laser cu trei reflectoare.

În sistemul de referință al robocarului este amplasată o sursă de radiație laser, care poate fi rotită cu  $360^\circ$  în planul  $xOy$ , astfel încât fasciculul baleiază spațiul de lucru din jurul robocarului (figura 4.8) [ PRU 89 ].

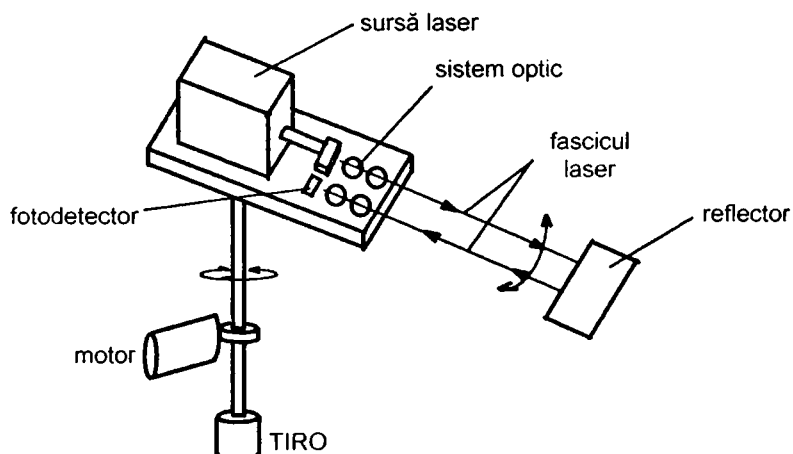


Figura 4.8.

În apropierea emițătorului laser se află un fotoreceptor, care poate recepționa fasciculul laser reflectat de către cele trei reflectoare fixe aflate în jurul robocarului. Robocarul se deplasează pe traiectorie, iar acolo unde este necesară determinarea situații sale (sistemul de referință  $R_i$ ) se amplasează cele trei reflectoare. Ele au o construcție specială și reflectă fasciculul paralel cu el însăși. Cu ajutorul acestui echipament se determină trei unghiuri, pe baza cărora se pot calcula coordonatele și orientarea axei robocarului. Amplasarea celor trei reflectoare (1, 2, 3) și a sursei laser este prezentată în figura 4.9.

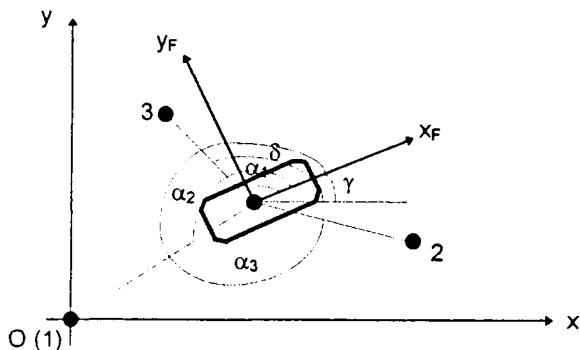


Figura 4.9.

În figura 4.10 se prezintă triunghiul format de către cele trei reflectoare, având în interior sursa laser. Următoarele elemente sunt cunoscute:

- laturile triunghiului  $l_{12}$ ,  $l_{13}$ ,  $l_{23}$  (prin amplasarea reflectoarelor);
- unghiurile formate de fasciculele laser (când intersectează fotoreceptorul) cu axa robocarului ( $\alpha_{11}$ ,  $\alpha_{12}$ ,  $\alpha_{23}$ ) (prin măsurare);
- unghiurile formate de  $l_{12}$  și  $l_{13}$  cu axa  $Ox$  ( $\gamma_1$ ,  $\gamma_2$ ) (din construcție);
- unghiurile de la vârfurile triunghiului 123 (din construcție).

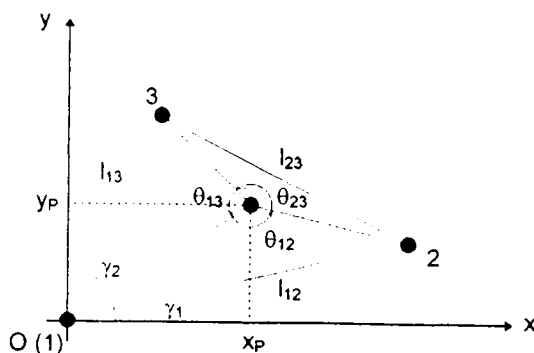


Figura 4.10.

Se vor determina unghiurile 1P2, 1P3, 2P3, iar apoi va rezulta orientarea robocarului față de axa Ox și coordonatele acestuia.

### 4.2.3. Determinarea situații vehiculului cu ajutorul camerei video.

Utilizarea camerei video prezintă avantajul că nu este necesară o baleiere mecanică a spațiului cu ajutorul senzorului (ca și în cazul metodelor care utilizează scannerul laser). Dacă se pot determina prin diverse metode unghiurile **sub care sunt văzute** de către cameră anumite repere, se pot determina coordonatele acestor repere și implicit se poate determina situația robocarului.

#### 4.2.3.1. Camera video și marcaje pe pardoseală.

Acest procedeu se bazează pe detectarea unor marcaje realizate pe pardoseală, cu ajutorul unei camere video, amplasată pe robocar, în timpul mișcării vehiculului. În figura 4.11 se prezintă proiecția marcajului în planul CCD-ului. În funcție de precizia de determinare  $\Delta l$  a mărimilor  $l_x$  și  $l_y$  cu ajutorul camerei video, se poate determina deplasarea autonomă a vehiculului.

Orientarea vehiculului este:

$$\gamma_F = \arcsin \frac{l_x}{l_y} \quad (4.15)$$

Înlocuită în (4.3) se obține:

$$\Delta \gamma_F = \frac{1}{\sqrt{1 - \left(\frac{l_x}{l_y}\right)^2}} \left[ \frac{1}{l_y} \Delta l_x + \frac{l_x}{l_y^2} \Delta l_y \right] \quad (4.16)$$

După cum rezultă din această relație,  $\Delta \gamma_F$  depinde de lungimea marcajului  $l_y$ , de mărimea  $l_x$ , precum și de precizia măsurării segmentelor  $\Delta l_x$  și  $\Delta l_y$ . Precizia de măsurare este determinată de numărul de pixeli ai camerei. Pentru un număr de 512 pixeli și o lungime  $l_y = 250\text{mm}$ , rezultă

$$\Delta l_y = \frac{250}{512} = 0,48 \text{ mm/pixel.}$$

Pentru alegerea lungimii marcajului  $l_y$ , trebuie realizat un compromis între faptul că lungimea aceasta trebuie să fie suficient de mare pentru o determinare cât mai precisă a mărimilor  $l_x$  și  $\gamma_F$ , și ca linia să fie **văzută** în întregime de către camera video ( $l_y$  mic). Dacă se ia  $\Delta l_x = \Delta l_y = \Delta l = 0,48$  mm/pixel, în funcție de orientarea vehiculului se pot obține deplasări



autonome cuprinse între 9m și 28m. Deoarece măsurarea are loc în timpul mersului, ea nu este indicată ca aceasta să fie efectuată atunci când vehiculul staționează, la pornirea sau oprirea acestuia.

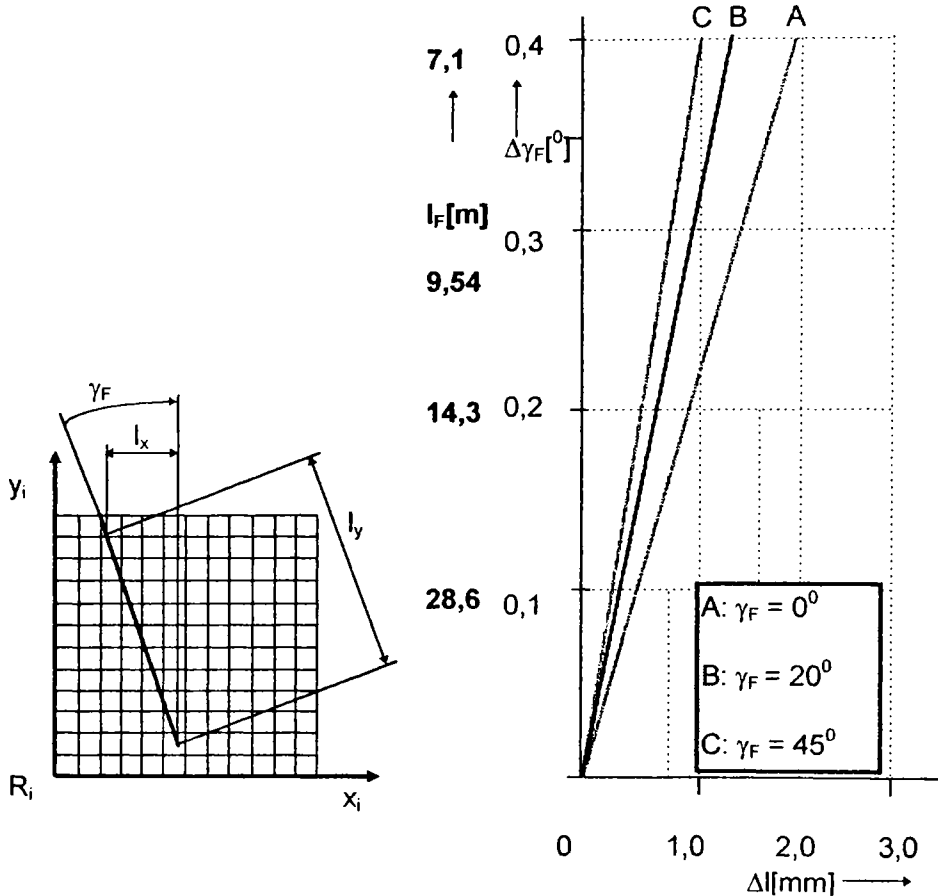


Figura 4.11.

Avantajul acestei metode este costul redus al realizării marcajelor. Dezavantajul utilizării ei pentru o poziționare precisă la oprire se înlătură printr-o combinație cu prima metodă.

#### 4.2.4. Alte metode și tipuri de senzori utilizați pentru determinarea situații.

Variatatea senzorilor cunoscuți este foarte mare. Oricare ar fi însă tipul acestora, determinarea situații se bazează pe măsurarea unor distanțe

și/sau a unor unghiuri. În continuare se vor prezenta două metode care utilizează senzori mai puțini uzuali.

#### 4.2.4.1. Senzor pentru măsurarea lungimilor și marcaje pe pardoseală.

Senzorul utilizat în acest caz este un senzor special, folosit pentru măsurarea lungimilor. El este instalat transversal pe vehicul și măsoară distanțele  $l_n$  și  $l_{n+1}$  dintre vehicul și punctele marcate pe pardoseală, pe o direcție perpendiculară pe axa vehiculului. Fiind cunoscută distanța dintre marcaje  $L_m$ , prin utilizarea mișcării vehiculului se transformă coordonatele acestuia în coordonatele sistemului  $R_i$ .

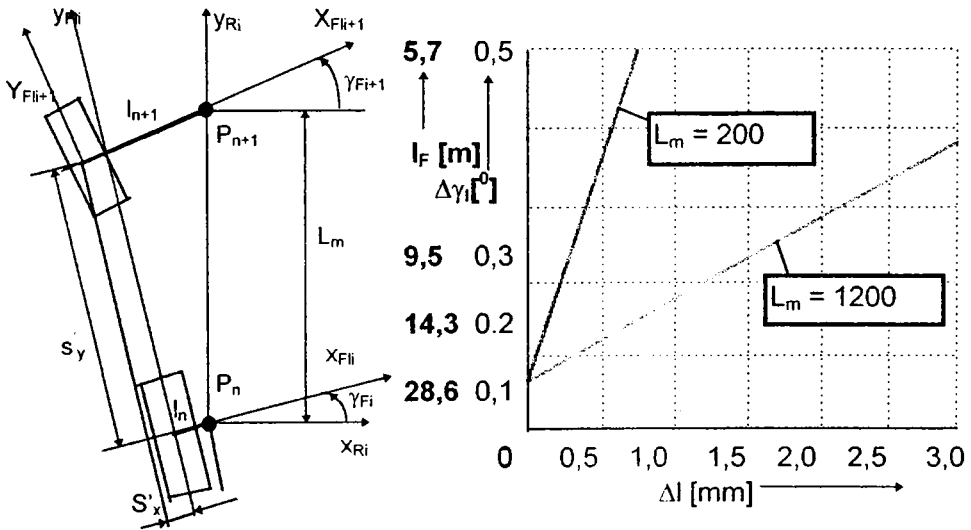


Figura 4.12.

Din figura 4.12 rezultă:

$$\Delta\gamma_{Fi} = \arccos\left(\frac{s'_y + l_{n+1} \sin d\gamma_F}{L_m}\right) = f_1(s'_y, l_{n+1}, d\gamma_F, L_m) \quad (4.17)$$

$$\Delta\gamma_{Fi} = \arcsin\left(\frac{l_{n+1} \cos d\gamma_F - l_n - s'_x}{L_m}\right) = f_2(s'_x, l_{n+1}, l_n, d\gamma_F, L_m) \quad (4.18)$$

Diferența unghiurilor  $d\gamma_F = \gamma_{Fi+1} - \gamma_{Fi}$  și segmentele  $S'_x$  și  $S'_y$  se determină prin utilizarea odometriei, vehiculul deplasându-se din punctul  $P_n$  în punctul  $P_{n+1}$ .

Aplicând relația (4.3) relațiilor (4.17) și (4.18), rezultă:

$$\Delta\gamma_{Fi} = -\frac{1}{\sqrt{1 - \left(\frac{s'_y + l_{n+1} \sin d\gamma_F}{L_m}\right)^2}} \left( \left| \frac{\delta f_1}{\delta s'_y} \right| \Delta s'_y + \left| \frac{\delta f_1}{\delta l_{n+1}} \right| \Delta l_{n+1} + \left| \frac{\delta f_1}{\delta d\gamma_F} \right| \Delta d\gamma_F + \left| \frac{\delta f_1}{\delta L_m} \right| \Delta L_m \right) \quad (4.19)$$

$$\Delta\gamma_{Fi} = \frac{1}{\sqrt{1 - \left(\frac{l_{n+1} \cos d\gamma_F - l_n - s'_x}{L_m}\right)^2}} + \left( \left| \frac{\delta f_2}{\delta s'_x} \right| \Delta s'_x + \left| \frac{\delta f_2}{\delta l_{n+1}} \right| \Delta l_{n+1} + \left| \frac{\delta f_2}{\delta d\gamma_F} \right| \Delta d\gamma_F + \left| \frac{\delta f_2}{\delta l_n} \right| \Delta l_n + \left| \frac{\delta f_2}{\delta L_m} \right| \Delta L_m \right) \quad (4.20)$$

Deoarece marcajele sunt amplasate de-a lungul direcției de mers, se consideră  $\Delta\gamma_{Fi} \cong 0^0$ , deci  $\cos \Delta\gamma_{Fi} \cong 1$ . Expresia de sub radical din relația (4.19) tinde spre 0, ceea ce are ca efect o creștere a erorii pentru  $\Delta\gamma_{Fi} \cong 0$ .

De aceea relația (4.21) se poate utiliza mai degrabă pentru determinarea orientării vehiculului. Se mai poate scrie :

$$\Delta\gamma_{Fi} = \frac{1}{\sqrt{1 - \left(\frac{l_{n+1} \cos d\gamma_F - l_n - s'_x}{L_m}\right)^2}} \left( \frac{1}{L_m} (\Delta s'_x + \Delta l_n) + \frac{\cos d\gamma_F}{L_m} \Delta l_{n+1} + \frac{l_{n+1} \sin d\gamma_F}{L_m} \Delta d\gamma_F + \frac{l_{n+1} \cos d\gamma_F - l_n - s'_x}{L_m^2} \Delta L_m \right) \quad (4.21)$$

Se pornește de la  $\gamma_{Fi} = 0^0$  și  $d\gamma_F = 0^0$ , deci de la o deplasare pe direcția axei  $y_{Fi}$  fără modificarea orientării. Metoda dă rezultate foarte bune dacă preciziile de măsurare a distanțelor cu ajutorul senzorilor sunt ridicate. Aceasta implică utilizarea unor senzori având rezoluția foarte bună. Aplicarea în timpul staționării a metodei descrise nu este posibilă.

Dacă se utilizează acest procedeu de măsurare, trebuie realizat un compromis între o distanță mare  $L_m$ , pentru obținerea unei precizii ridicate (4.20) și o distanță redusă, pentru ca erorile de măsurare prin odometrie ale lui  $S'_y$  și  $S'_x$  să fie mici.

#### 4.2.4.2. Receptor cu senzor PSD și scaner laser.

Senzorul PSD poate fi utilizat pentru măsurarea distanțelor dintre emițătorul laser și obiectul care reflectă fasciculul [LEU 97]. Distanța

măsurată este utilizată pentru determinarea situației. Principiul de măsurare rezultă din figura 4.13.

Dioda laser emite un fascicul de lumină care, prin intermediul unui sistem optic este proiectat spre obiectul până la care se determină distanța. Obiectul reflectă **difuz** fasciculul. Unul dintre fasciculele reflectate va ajunge în obiectivul receptorului. Senzorul utilizat este un senzor **PSD**. Acesta este un element analogic liniar, construit dintr-un substrat din siliciu și două straturi semiconductoare de tip **p**, respectiv **n**. Fasciculul laser este focalizat de către sistemul optic al receptorului pe suprafața senzorului. Poziția în care are loc incidența este diferită pentru distanțe diferite și poate fi detectată cu ajutorul unui echipament electronic relativ simplu.

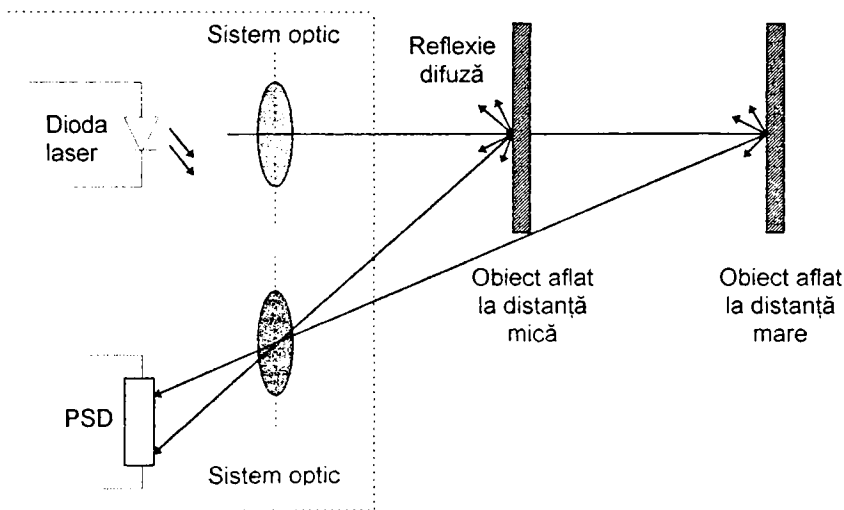


Figura 4.13.

Prin măsurarea distanței dintre emițătorul amplasat pe robocar și două obiecte care au coordonatele cunoscute, se poate determina poziția vehiculului. Cunoscând unghiul dintre fasciculul emis și axa robocarului, se poate determina orientarea acestuia. Măsurarea distanțelor cu ajutorul acestui senzor este mai rapidă decât dacă se utilizează ultrasunete.

\*\*\*\*\*

În cadrul metodelor descrise în continuare se poate utiliza pentru determinarea unghiurilor un dispozitiv de scanare cu fascicul laser. Autorul consideră însă că măsurarea unghiurilor cu ajutorul camerei video CCD oferă anumite avantaje. Aceste avantaje vor fi prezentate în cadrul concluziilor și vor fi susținute prin experimentele realizate.

### 4.2.5. Determinarea situații unui robocar în spațiu prin vizarea a patru puncte țintă.

Metoda prezentată în continuare se poate aplica pentru determinarea situații unui corp oarecare în spațiul 3D. În acest scop sunt necesare patru puncte, având distanțele dintre ele cunoscute. Deoarece acestea nu pot fi percepute pentru orice situație a corpului, este necesar ca ele să fie amplasate pe toate fețele acestuia. În cele ce urmează se va prezenta modul de determinare a situații unui robocar, punctele fiind amplasate pe o singură față.

#### 4.2.5.1. Principiul metodei.

Prin metoda propusă se determină coordonatele robocarului și orientarea sa, făcându-se apel numai la unghiurile pe care le formează dreptele care trec prin origine și prin fiecare dintre cele patru puncte țintă și la distanțele dintre puncte. Nu sunt cunoscute distanțele de la sursa luminoasă până la receptoare (de la origine până la cele patru puncte).

Se consideră un corp de formă paralelipipedică aflat în spațiul 3D. O sursă de lumină laser, care poate baleia spațiul pe orizontală (azimut) și pe verticală (elevație), este amplasată în originea sistemului de axe de coordonate  $(x, y, z)$ .

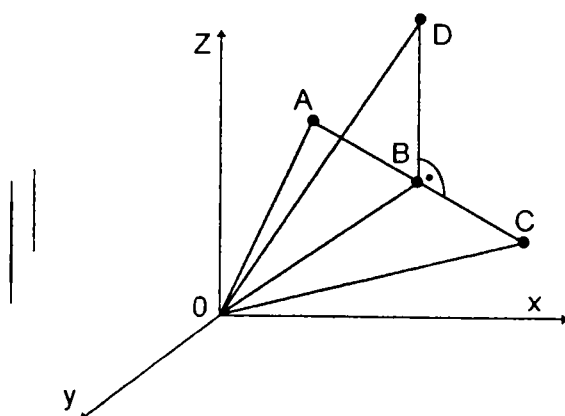


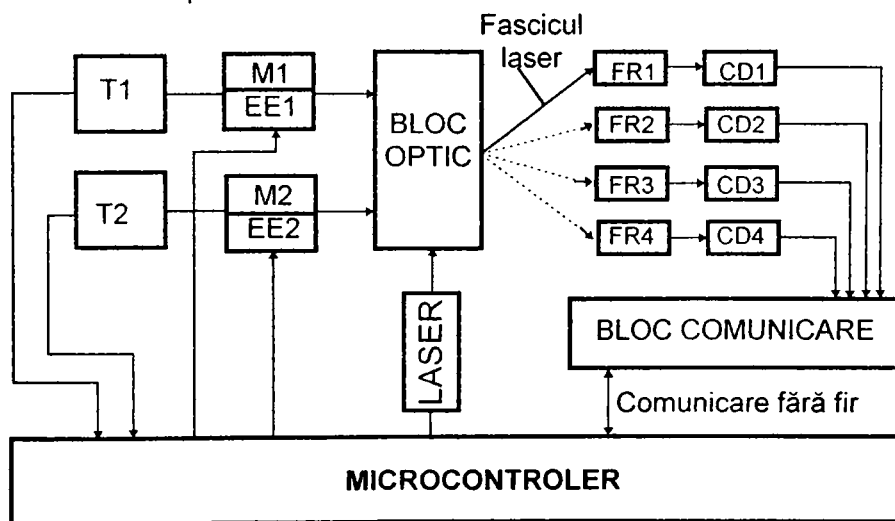
Figura 4.14.

În fiecare moment, unghiurile pe care fasciculul laser le formează cu axele  $Ox$ ,  $Oy$ ,  $Oz$  se pot măsura cu ajutorul unor traductoare, care echepează fiecare axă de rotație (figura 4.14).

Măsurarea unghiurilor se realizează prin utilizarea unui echipament, a cărui schemă bloc este prezentată în figura 4.15. Blocul optic conține, pe lângă obiectiv, două oglinzi pentagonale, rotite cu ajutorul unor motoare fără perii (M1, M2). Fiecare sistem de rotire este prevăzut cu un traductor incremental (T1, T2), astfel încât se poate determina poziția oglinzii și implicit unghiul pe care-l formează fasciculul cu una dintre axele sistemului de coordonate. Una dintre oglinzi deviază fasciculul în plan orizontal, iar cealaltă în plan vertical.

În patru puncte, având coordonatele cunoscute față de sistemul de referință atașat robocarului, sunt amplasate patru fotoreceptoare (FR1, FR2, FR3, FR4), care au sensibilitatea maximă pentru lungimea de undă a luminii emise de dioda laser.

Fotoreceptoarele FR1÷FR4 sesizează incidența fasciculului laser, iar prin intermediul circuitului de detecție, transmit informația sistemului de conducere echipat cu microcontroler.



T1, T2 - traductoare incrementale  
M1, M2 - motoare pentru rotirea oglinzilor  
EE1, EE2 - elemente de execuție  
BLOC OPTIC - blocul oglinzilor  
FR1, FR2 - fotoreceptoare (fotodiode)  
CD1, CD2 - circuite de detecție  
BLOC COMUNICARE - asigură legătura între robocar și sistemul fix.

Figura 4.15.

Valorile obținute prin măsurarea unghiurilor sub care sunt orientate oglinzile în momentul sesizării de către fotodetectoare a fascicului laser sunt memorate de către sistemul de conducere și stocate în fișiere corespunzătoare fiecărui punct.

Deoarece emițătorul laser este fix, iar receptoarele se află pe robocar, între punctul fix și robocar este necesar să existe o comunicare permanentă, iar informația trebuie să fie transmisă în timp real.

În figura 4.14 se cunosc distanțele dintre cele patru puncte A, B, C, D și unghiurile formate de către razele vectoriale ale punctelor țintă cu axele sistemului de referință.

Se consideră:

$AB = BC = BD = d$  și  $BD \perp AC$  (cunoscute)

$\alpha_A, \beta_A, \alpha_B, \beta_B, \alpha_C, \beta_C, \alpha_D, \beta_D$ , (măsurate prin sistemul descris mai sus). Semnificația unghiurilor rezultă din figura 4.16.

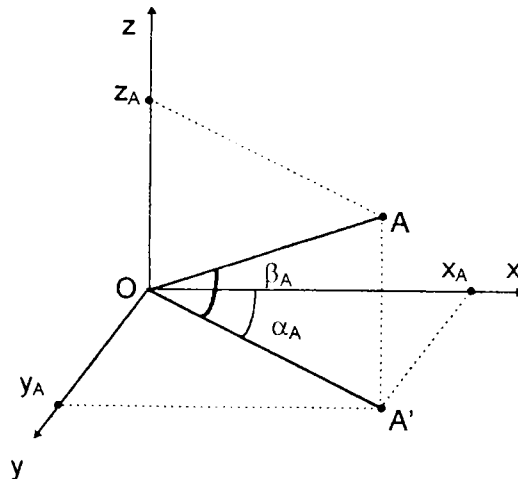


Figura 4.16.

S-a notat cu  $\alpha$  azimutul și cu  $\beta$  elevația.

Se vor determina coordonatele fiecărui punct în spațiu ( $x_A, y_A, z_A, x_B, y_B, z_B, \dots$ ) și orientarea robocarului raportată la una dintre axe.

### 4.2.5.2. Determinarea situații robocarului.

Se consideră cele două puncte A și B ca în figura 4.17.

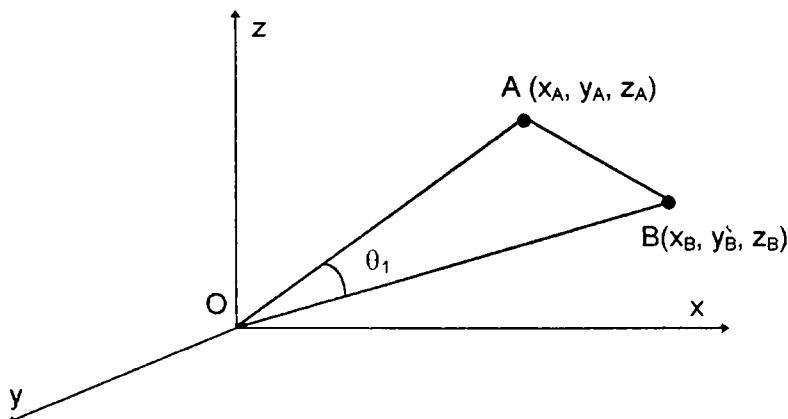


Figura 4.17.

Se notează unghiurile  $\text{AOB} = \theta_1$ ,  $\text{BOC} = \theta_2$ ,  $\text{BOD} = \theta_3$ ,  $\text{AOC} = \theta = \theta_1 + \theta_2$ . Se observă că în triunghiul AOB trebuie determinat unghiul  $\theta_1$  pentru a se putea calcula coordonatele punctelor.

Segmentul de dreaptă  $\overline{OA}$  este determinat de punctele  $O(0, 0, 0)$  și  $A(x_A, y_A, z_A)$  și are vectorul director  $\vec{U}_1 = \overline{OA}$ .

Parametrii vectorului  $U_1$  sunt:

$$\begin{cases} l = x_A - x_0 & l_1 = x_A \\ m_1 = y_A - y_0 & \Rightarrow m_1 = y_A \\ n_1 = z_A - z_0 & n_1 = z_A \end{cases} \quad (4.22)$$

pentru segmentul  $\overline{OB}$ :

$$\begin{aligned} \vec{U}_2: & \quad l_2 = x_B \quad m_2 = y_B \quad n_2 = z_B \\ \overline{OC}, \vec{U}_3: & \quad l_3 = x_C \quad m_3 = y_C \quad n_3 = z_C \\ \overline{OD}, \vec{U}_4: & \quad l_4 = x_D \quad m_4 = y_D \quad n_4 = z_D \end{aligned}$$

unghiul  $\text{AOB} = \theta_1$  dintre  $\vec{U}_1$  și  $\vec{U}_2$  este dat de relația:

$$\cos \theta_1 = \frac{\vec{U}_1 \cdot \vec{U}_2}{|\vec{U}_1| \cdot |\vec{U}_2|} \quad (4.23)$$

$$\cos \theta_1 = \frac{l_1 \cdot l_2 + m_1 \cdot m_2 + n_1 \cdot n_2}{\sqrt{l_1^2 + m_1^2 + n_1^2} \cdot \sqrt{l_2^2 + m_2^2 + n_2^2}} \quad (4.24)$$



$$\cos \theta_2 = \frac{l_2 \cdot l_3 + m_2 \cdot m_3 + n_2 \cdot n_3}{\sqrt{l_2^2 + m_2^2 + n_2^2} \cdot \sqrt{l_3^2 + m_3^2 + n_3^2}} \quad (4.25)$$

$$\cos \theta_3 = \frac{l_2 \cdot l_4 + m_2 \cdot m_4 + n_2 \cdot n_4}{\sqrt{l_2^2 + m_2^2 + n_2^2} \cdot \sqrt{l_4^2 + m_4^2 + n_4^2}} \quad (4.26)$$

Din figura 4.16 rezultă coordonatele unui punct (de exemplu A) în funcție de unghiurile de azimut și elevație ( $\alpha_A$  - azimut,  $\beta_A$  - elevație) și de distanța dintre origine și punctul A.

$$\begin{cases} x_A = l_A \cdot \cos \beta_A \cdot \cos \alpha_A \\ y_A = l_A \cdot \cos \beta_A \cdot \sin \alpha_A \\ z_A = l_A \cdot \sin \beta_A \end{cases} \quad (4.27)$$

cu

$$\begin{aligned} l_1 &= x_A & l_{2,3,4} &= m_{2,3,4} & n_{2,3,4} \\ m_1 &= y_A \\ n_1 &= z_A \end{aligned}$$

se scrie:

$$\begin{aligned} \cos \theta_1 &= (l_A l_B \cos \beta_A \cos \alpha_A \cos \beta_B \cos \alpha_B + l_A l_B \cos \beta_A \sin \alpha_A \cos \beta_B \sin \alpha_B + \\ &+ l_A l_B \sin \beta_A \sin \beta_B) \cdot \left( \sqrt{l_A^2 \cos^2 \beta_A + l_A^2 \cos^2 \beta_A \sin^2 \alpha_A + l_A^2 \sin^2 \beta_A} \cdot \right. \\ &\left. \cdot \sqrt{l_B^2 \cos^2 \beta_B \cos^2 \alpha_B + l_B^2 \cos^2 \beta_B \sin^2 \alpha_B + l_B^2 \sin^2 \beta_B} \right)^{-1} \end{aligned} \quad (4.28)$$

rezultă:

$$\cos \theta_1 = \frac{\cos \beta_A \cos \alpha_A \cos \beta_B \cos \alpha_B + \cos \beta_A \sin \alpha_A \cos \beta_B \sin \alpha_B + \sin \beta_A \sin \beta_B}{\sqrt{\cos^2 \beta_A \cos^2 \alpha_A + \cos^2 \beta_A \sin^2 \alpha_A + \sin^2 \beta_A} \sqrt{\cos^2 \beta_B \cos^2 \alpha_B + \cos^2 \beta_B \sin^2 \alpha_B + \sin^2 \beta_B}}$$

și (4.28')

$$\cos \theta_2 = \frac{\cos \beta_B \cos \alpha_B \cos \beta_C \cos \alpha_C + \cos \beta_B \sin \alpha_B \cos \beta_C \sin \alpha_C + \sin \beta_B \sin \beta_C}{\sqrt{\cos^2 \beta_B \cos^2 \alpha_B + \cos^2 \beta_B \sin^2 \alpha_B + \sin^2 \beta_B} \sqrt{\cos^2 \beta_C \cos^2 \alpha_C + \cos^2 \beta_C \sin^2 \alpha_C + \sin^2 \beta_C}}$$

și (4.29)

$$\cos \theta_3 = \frac{\cos \beta_B \cos \alpha_B \cos \beta_D \cos \alpha_D + \cos \beta_B \sin \alpha_B \cos \beta_D \sin \alpha_D + \sin \beta_B \sin \beta_D}{\sqrt{\cos^2 \beta_B \cos^2 \alpha_B + \cos^2 \beta_B \sin^2 \alpha_B + \sin^2 \beta_B} \sqrt{\cos^2 \beta_D \cos^2 \alpha_D + \cos^2 \beta_D \sin^2 \alpha_D + \sin^2 \beta_D}}$$

(4.30)

În continuare,  $\theta_1$  și  $\theta_2$  fiind cunoscute, se determină  $l_A$ ,  $l_B$ ,  $l_C$ ,  $l_D$  (figura 4.18).

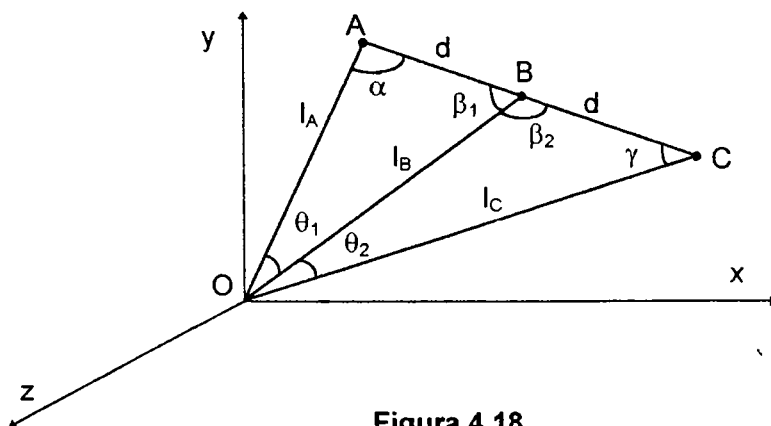


Figura 4.18.

Aplicând teorema sinusurilor în triunghiurile AOB, BOC și AOC rezultă:

$$\begin{cases} \frac{l_A}{\sin \beta_1} = \frac{l_B}{\sin \alpha} = \frac{d}{\sin \theta_1} \\ \frac{l_C}{\sin \beta_2} = \frac{l_B}{\sin \gamma} = \frac{d}{\sin \theta_2} \\ \frac{l_A}{\sin \gamma} = \frac{l_C}{\sin \alpha} = \frac{2d}{\sin \theta} \end{cases} \quad (4.31)$$

și

$$\begin{cases} \beta_1 + \beta_2 = \pi \\ \alpha + \gamma + \theta = \pi \\ \theta_1 + \theta_2 = \theta \end{cases} \quad (4.32)$$

deci:

$$\frac{l_B}{l_C} = \frac{\sin \theta}{2 \sin \theta_1} \quad (4.33)$$

dacă se notează:

$$\frac{l_C}{l_B} = p_1 \quad (4.34)$$

Aplicând teorema cosinusurilor în triunghiul BOC:

$$l_B^2 + l_C^2 - 2l_B l_C \cos \theta_2 = d^2 \quad (4.35)$$

dar  $l_C = l_B p_1$

$$l_B^2 + p_1^2 l_B^2 - 2p_1 l_B^2 \cos \theta_2 = d^2 \quad (4.35')$$

$$l_B = \pm \sqrt{\frac{d^2}{p_1^2 - 2p \cos \theta_2 + 1}} = \pm \frac{d}{\sqrt{p_1^2 - 2p \cos \theta_2 + 1}} \quad (4.36)$$

cu condiția  $p_1^2 - 2p \cos \theta_2 + 1 > 0$

$$l_C = \frac{p_1 d}{\sqrt{p_1^2 - 2p_1 \cos \theta_2 + 1}} \quad (4.37)$$

$$l_A = 2 \frac{\sin \theta_2}{\sin \theta} \cdot l_B = 2 \frac{\sin \theta_2}{\sin \theta} \cdot \frac{d}{\sqrt{p_1^2 - 2p_1 \cos \theta_2 + 1}} \quad (4.38)$$

$$2 \frac{\sin \theta_2}{\sin \theta} = p_2 \quad (4.39)$$

$$l_A = \frac{dp_2}{\sqrt{p_1^2 - 2p_1 \cos \theta_2 + 1}} \quad (4.40)$$

Aplicând teorema sinusurilor în triunghiul BOD:

$$\frac{d}{\sin \theta_3} = \frac{l_B}{\sin \delta} \quad (4.41)$$

$$\delta = \arcsin \left( \frac{l_B \cdot \sin \theta_3}{d} \right) \quad (4.41')$$

$$l_D = \frac{d \sin(\theta_3 + \delta)}{\sin \theta_3} = d \cos \delta + d \sin \delta \operatorname{ctg} \theta_3 \quad (4.42)$$

$$\sin \delta = \frac{l_B \cdot \sin \theta_3}{d} \quad (4.43)$$

$$\cos \delta = \sqrt{1 - \sin^2 \delta} = \sqrt{\frac{d^2 - l_B^2 \sin^2 \theta_3}{d^2}} \quad (4.44)$$

$$l_D = \sqrt{d^2 - l_B^2 \sin^2 \theta_3} + l_B \cos \theta_3 \quad (4.45)$$

În această fază se cunosc  $l_A$ ,  $l_B$ ,  $l_C$ ,  $l_D$  și se vor determina coordonatele  $x$ ,  $y$ ,  $z$  ale punctelor A, B, C, D folosind relațiile:

$$x_A = l_A \cos \beta_A \cos \alpha_A$$

$$x_B = l_B \cos \beta_B \cos \alpha_B$$

$$x_C = l_C \cos \beta_C \cos \alpha_C$$

$$x_D = l_D \cos \beta_D \cos \alpha_D$$

$$x_A = \frac{dp_2}{\sqrt{p_1^2 - 2p_1 \cos \theta_2 + 1}} \cdot \cos \beta_A \cos \alpha_A \quad (4.46)$$

$$y_A = \frac{dp_2}{\sqrt{p_1^2 - 2p_1 \cos \theta_2 + 1}} \cdot \cos \beta_A \sin \alpha_A \quad (4.47)$$

$$z_A = \frac{dp_2}{\sqrt{p_1^2 - 2p_1 \cos \theta_2 + 1}} \cdot \sin \alpha_A \quad (4.48)$$

La fel se vor calcula și coordonatele  $x$ ,  $y$  și  $z$  ale punctelor B, C și D. Pentru stabilirea orientării robocarului, se scrie ecuația planului feței care conține punctele A, B, C, D.

$$\begin{vmatrix} x & y & z & 1 \\ x_A & y_A & z_A & 1 \\ x_B & y_B & z_B & 1 \\ x_C & y_C & z_C & 1 \end{vmatrix} = 0 \quad (4.49)$$

Parametrii directori ai normalei pe planul care conține punctele A, B, C, D sunt:

$$A^* = \begin{vmatrix} y_A & z_A & 1 \\ y_B & z_B & 1 \\ y_C & z_C & 1 \end{vmatrix} \quad (4.50)$$

$$B^* = \begin{vmatrix} x_A & z_A & 1 \\ x_B & z_B & 1 \\ x_C & z_C & 1 \end{vmatrix} \quad (4.51)$$

$$C^* = \begin{vmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_C & y_C & 1 \end{vmatrix} \quad (4.52)$$

Astfel s-au determinat coordonatele punctelor A, B, C, D și direcțiile normalei la planul format de către aceste puncte.

#### 4.2.6. Determinarea situații unui robocar în plan, prin vizarea a două puncte țintă.

Această metodă, cunoscută în literatura de specialitate [HAM 86b] sub denumirea **metoda Beacon**, se bazează pe măsurarea unghiurilor formate de către patru drepte (care unesc două puncte fixe cu două puncte mobile aflate pe robocar) cu axa Ox. Distanța dintre cele două puncte fixe fiind cunoscută, utilizând unghiurile măsurate, se pot determina coordonatele celor două puncte mobile și orientarea dreptei care le unește.

##### 4.2.6.1. Principiul metodei.

Prin metoda descrisă în paragraful anterior se determină situarea unui obiect aflat în spațiul 3D. Considerând însă că robocarul, care se deplasează în mediu industrial, se află în spațiul 2D, deoarece nu există

variații de altitudine în timpul mișcării, problema determinării coordonatelor și a orientării se simplifică. O variantă posibilă de măsurare este cea prezentată în continuare.

În două puncte aparținând sistemului de axe de coordonate de referință și având coordonatele cunoscute  $O(0,0)$  și  $O'(d,0)$  (figura 4.20), se amplasează două surse laser, L1, L2 (vezi și figura 4.19). Fasciculul fiecărei surse laser este deviat de către o oglindă care face parte din blocul optic BO1, respectiv BO2 și este rotită cu ajutorul unui motor fără perii (M1, respectiv M2). Sistemele de rotire sunt prevăzute cu traductoare incrementale (T1, T2). Fasciculele laser sunt deviate în plan orizontal și pot fi recepționate de către cele două fotoreceptoare FR1, FR2, aflate pe robocar. Poziția fotoreceptoarelor în sistemul de referință al robocarului este cunoscută.

Momentul detectării fasciculului de către fotoreceptoare este marcat prin semnalul obținut la ieșirea fotodetectorului. Cu ajutorul acestui semnal, sistemul de conducere al vehiculului determină valoarea unghiului dintre fasciculul care activează fotodetectorul respectiv și axa  $Ox$  ( $\alpha_A$ ,  $\alpha_B$ ,  $\alpha'_A$ ,  $\alpha'_B$ ).

Schema bloc a echipamentului care comandă rotirea oglinzilor și a echipamentului pentru măsurarea unghiurilor este prezentată în figura 4.19.

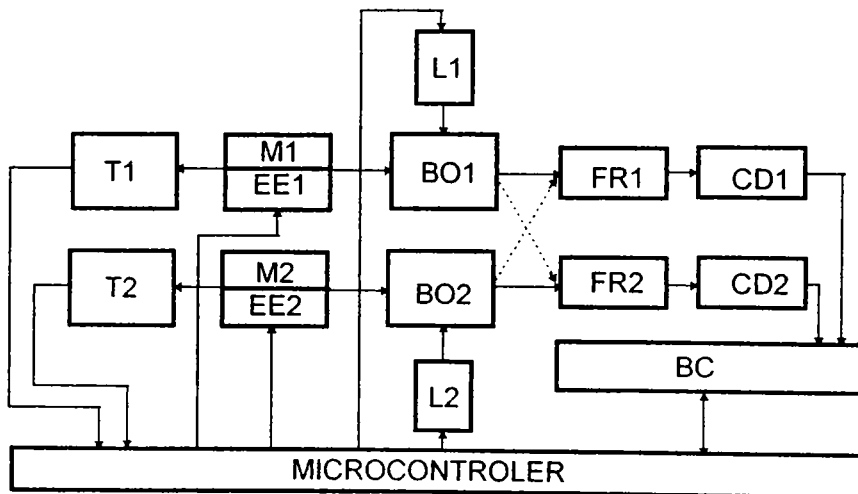


Figura 4.19.

### 4.2.6.2. Determinarea situații robocarului.

Situarea robocarului în planul  $xOy$ , modul de amplasare a punctelor fixe ( $O$  și  $O'$ ) și a punctelor mobile ( $A$  și  $B$ ) sunt conform figurii 4.20.

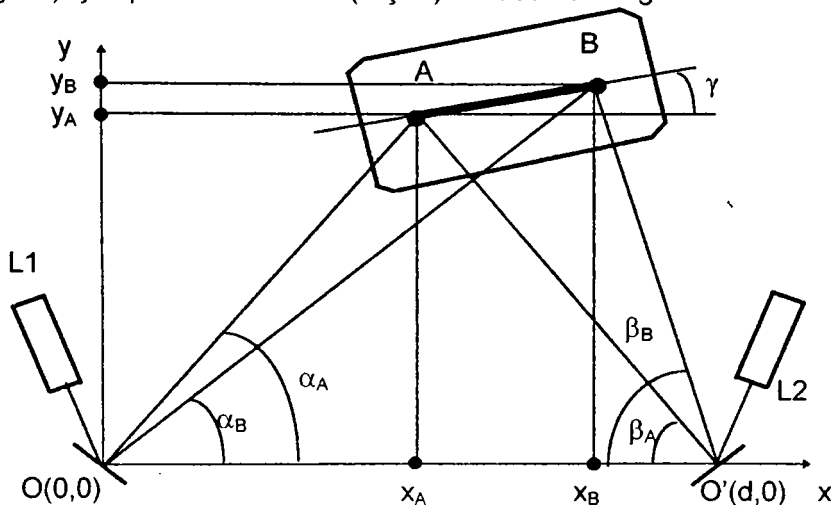


Figura 4.20.

Considerând un singur punct și folosind notațiile din figura 4.20, rezultă figura 4.21. Dacă se cunosc unghiurile  $\alpha_A$  și  $\beta_A$  (măsurate cu ajutorul tractoarelor de rotație), atunci când raza laser este recepționată de către fotodetectorul amplasat în punctul  $A$  se poate scrie:

$$\operatorname{tg} \alpha_A = \frac{x_A}{y_A} \quad (4.53)$$

$$\operatorname{tg} \beta_A = \frac{d - x_A}{y_A} \quad (4.54)$$

$$x_A = y_A \cdot \operatorname{tg} \alpha_A \quad (4.55)$$

$$a - x_A = y_A \cdot \operatorname{tg} \beta_A \quad (4.56)$$

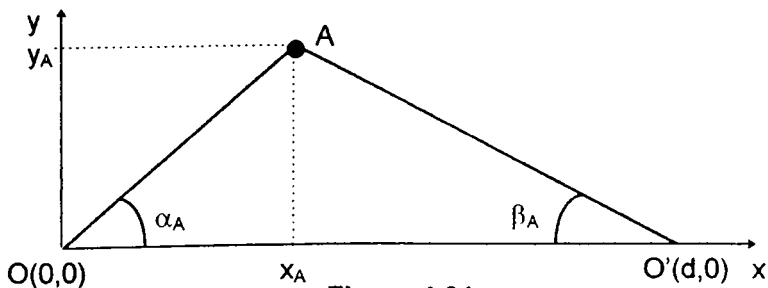


Figura 4.21.

Din raportul relațiilor 4.55 și 4.56 rezultă:

$$\frac{x_A}{d - x_A} = \frac{\operatorname{tg}\alpha_A}{\operatorname{tg}\beta_A} \quad (4.57)$$

$$x_A \cdot \operatorname{tg}\beta_A = (d - x_A) \operatorname{tg}\alpha_A \quad (4.58)$$

$$x_A (\operatorname{tg}\alpha_A + \operatorname{tg}\beta_A) = d \cdot \operatorname{tg}\alpha_A \quad (4.59)$$

$$x_A = d \cdot \frac{\operatorname{tg}\alpha_A}{\operatorname{tg}\alpha_A + \operatorname{tg}\beta_A} \quad (4.60)$$

iar 
$$y_A = \frac{x_A}{\operatorname{tg}\alpha_A} = \frac{d}{\operatorname{tg}\alpha_A + \operatorname{tg}\beta_A} \quad (4.61)$$

Pentru coordonatele punctului B,  $x_B$  și  $y_B$  se obțin similar:

$$x_B = d \cdot \frac{\operatorname{tg}\alpha_B}{\operatorname{tg}\alpha_B + \operatorname{tg}\beta_B} \quad (4.62)$$

$$y_B = \frac{d}{\operatorname{tg}\alpha_B + \operatorname{tg}\beta_B} \quad (4.63)$$

Pentru determinarea orientării vehiculului, din figura 4.22 rezultă:

$$\operatorname{tg}\gamma = \frac{y_B - y_A}{x_B - x_A} \quad (4.64)$$

$$\gamma = \operatorname{arctg} \frac{y_B - y_A}{x_B - x_A} \quad (4.65)$$

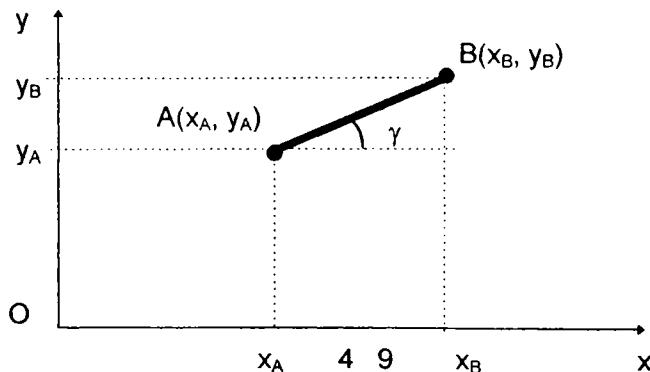


Figura 4.22.

Deci, poziția și orientarea vehiculului se pot determina măsurând unghiurile pe care fasciculele laser le formează cu axa Ox în momentul detectării acestora de către fotoreceptoarele amplasate în punctele A, respectiv B și cunoscând distanța dintre cele două surse de lumină. Între fotodetectoare și dispozitivele de măsură ale unghiurilor trebuie să existe

un sistem de comunicare în timp real, pentru o determinare precisă a situații.

#### **4.2.7. Determinarea situații unui robocar aflat în plan, prin vizarea a două puncte țintă.**

În continuare se va prezenta o metodă care utilizează pentru determinarea coordonatelor și a orientării vehiculului o singură sursă laser și două fotoreceptoare.

##### **4.2.7.1. Principiul metodei.**

Din analiza primelor două metode se pot desprinde următoarele concluzii:

- prima metodă utilizează o singură sursă laser și patru fotoreceptoare, deoarece trebuie calculate atât coordonatele robocarului, cât și orientarea sa. Aceste calcule se efectuează fără măsurarea distanței de la cele patru puncte mobile la punctul fix și a distanței dintre acestea. Este necesară măsurarea unor unghiuri. Metoda se poate utiliza pentru determinarea situații oricărui corp în spațiul 3D;
- a doua metodă utilizează două puncte situate pe robocar, dar pentru calculul coordonatelor și a orientării sunt necesare în acest caz două surse laser. Fotodectoarele trebuie să discearnă care dintre fasciculele laser este cel care a fost detectat. Prin acest procedeu se determină situația în spațiul 2D.

**Metoda propusă în continuare, imaginată de către autor, este rezultatul analizei mai multor metode și are ca scop determinarea situații unui robocar aflat într-un sistem de referință oarecare.**

Schema bloc a circuitului este prezentată în figura 4.23. Ca și în cazul primei metode prezentate, blocul optic conține două oglinzi care se rotesc și deviază fasciculul în plan orizontal și vertical. Rotirea oglinzilor se face cu ajutorul motoarelor M1 și M2, iar azimutul și elevația se determină prin informațiile obținute cu ajutorul traductoarelor T1 și T2. Cele două fotodectoare FR1-CD1 respectiv FR2-CD2, sunt amplasate în două puncte A și B aflate pe robocar. Ele transmit spre microcontroler un semnal atunci când are loc detecția fasciculului laser.



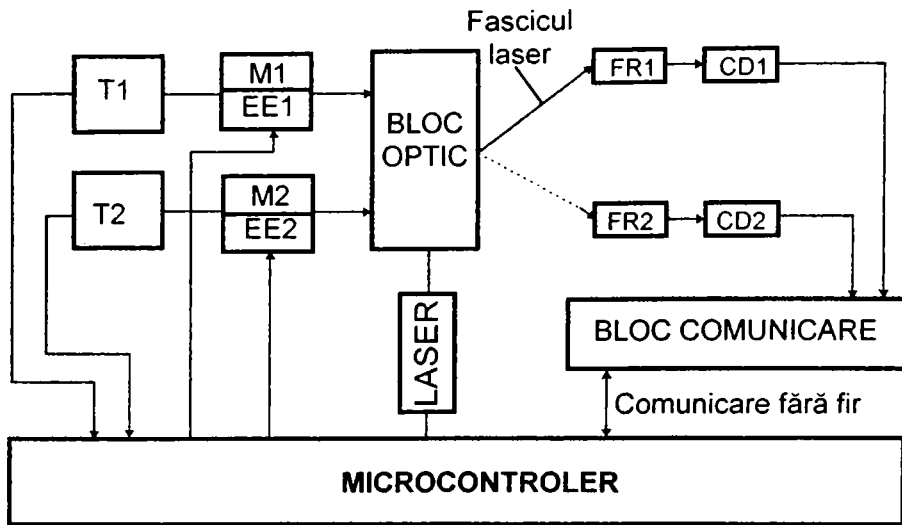


Figura 4.23.

Unghiul măsurat în acest moment cu ajutorul traductorului corespunzător este memorat de către microcontroler și este utilizat în calculul situații robocarului, după cum se va arăta în cele ce urmează.

Între blocul de comandă al emițătorului laser (al dispozitivului de scanare) și receptoare trebuie să existe o comunicație în timp real pentru determinarea corectă a unghiurilor.

#### 4.2.7.2. Determinarea situații robocarului.

Pentru determinarea situații robocarului s-a considerat sistemul de axe de coordonate  $(x,y,z)$ . Robocarul se află în planul  $(xOy)$ , iar sursa laser pe axa  $Oz$  la înălțimea  $h$  (figura 4.24).

Două fotoreceptoare sunt amplasate pe axa longitudinală a robocarului, în punctele A și B, fără ca distanța dintre acestea să fie cunoscută. Orientarea robocarului (unghiul format de către dreapta AB cu axa Ox) este dată de relația 4.66.

$$\operatorname{tg} \gamma = \frac{y_B - y_A}{x_B - x_A} \quad (4.66)$$

$$\gamma = \operatorname{arctg} \frac{y_B - y_A}{x_B - x_A} \quad (4.66')$$

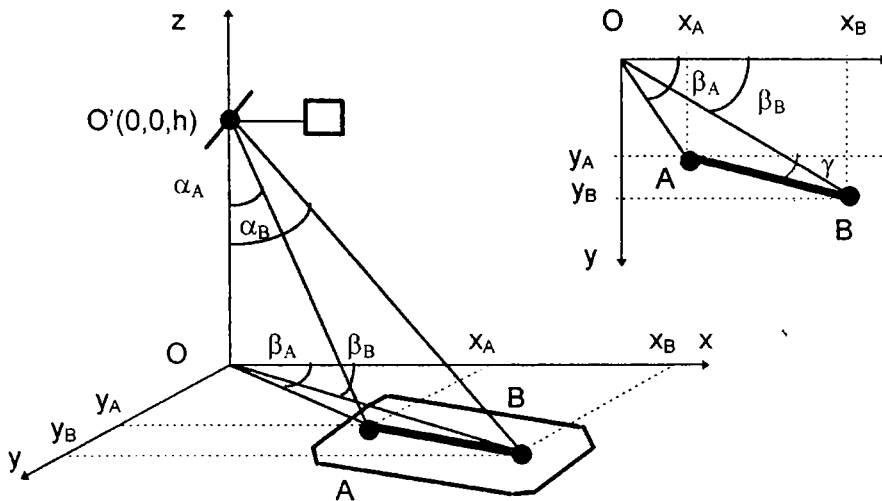


Figura 4.24.

Pentru calculul coordonatelor punctelor A și B, este necesară cunoașterea înălțimii  $h$  și a unghiurilor  $\alpha$  și  $\beta$ , după cum rezultă din cele ce urmează:

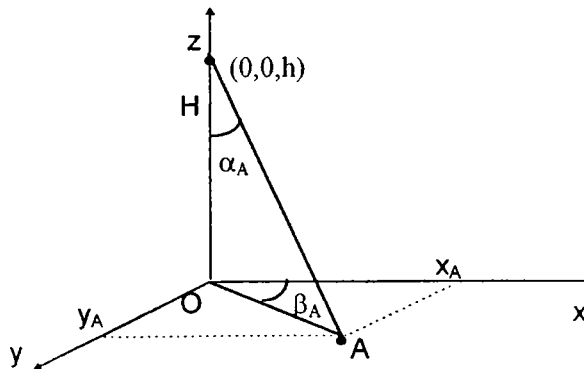


Figura 4.25.

Din figura 4.25, considerând triunghiul AOH, rezultă:

$$\frac{OA}{h} = \operatorname{tg} \alpha_A \quad (4.67)$$

$$OA = h \cdot \operatorname{tg} \alpha_A \quad (4.67')$$

iar  $x_A = OA \cdot \sin \beta_A = h \cdot \operatorname{tg} \alpha_A \cdot \sin \beta_A \quad (4.68)$

$$y_A = OA \cdot \cos \beta_A = h \cdot \operatorname{tg} \alpha_A \cdot \cos \beta_A \quad (4.69)$$

Similar, pentru punctul B rezultă:

$$x_B = OA \cdot \sin \beta_B = h \cdot \operatorname{tg} \alpha_B \cdot \sin \beta_B \quad (4.70)$$

$$y_B = OA \cdot \cos \beta_B = h \cdot \operatorname{tg} \alpha_B \cdot \cos \beta_B \quad (4.71)$$

Se observă că, atât coordonatele punctelor care determină coordonatele robocarului ( $x_A, y_A$ , respectiv  $x_B, y_B$ ), cât și orientarea ( $\gamma$ ) robocarului nu depind de distanța dintre sursa de lumină și punct și nici de distanța dintre cele două puncte. Ele depind numai de înălțimea  $h$  și de unghiurile de azimut și elevație. Precizia determinării situației depinde de precizia determinării unghiurilor (precizia detecției fasciculului laser și precizia traductoarelor TIRO) și de precizia amplasării sursei de lumină la înălțimea  $h$ . Experimentele care confirmă performanțele metodei vor fi prezentate în capitolul 7.

#### 4.2.8. Determinarea situației unui robocar cu cameră video staționară.

Metoda prezentată în continuare este avantajoasă deoarece nu necesită echipamente speciale pentru calibrarea dispozitivelor și pentru măsurare.

##### 4.2.8.1. Principiul metodei.

La înălțimea  $z_1=h$  cunoscută se amplasează o cameră video (figura 4.26). Axa sistemului optic al camerei este perpendiculară pe traiectoria robocarului, iar coordonatele punctului D, care reprezintă intersecția acestei axe cu planul  $xOy$ , ( $x_1, y_1$ ), sunt cunoscute.

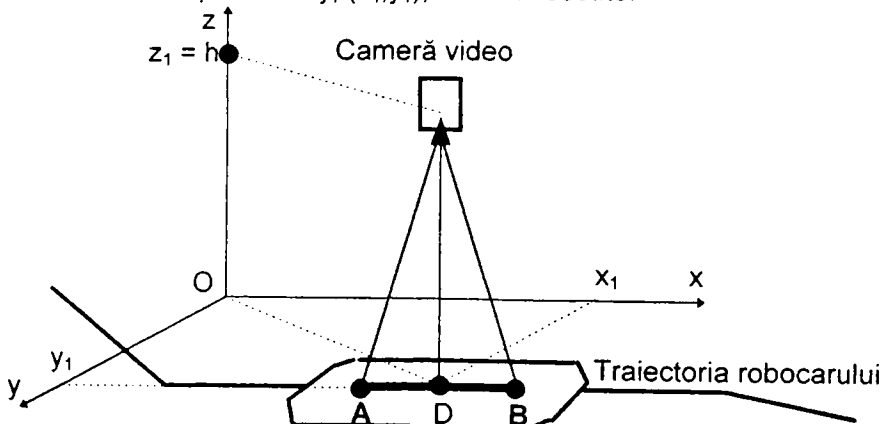


Figura 4.26.

Pe vehicul se dispun două marcaje. Aceste marcaje pot fi, pentru o recunoaștere mai ușoară, două leduri care funcționează în infraroșu, alimentate cu impulsuri de curent, care au o anumită frecvență.

Cunoscând numărul de milimetri din planul robocarului care corespunde unui pixel, se pot calcula coordonatele punctelor și apoi se poate determina situația vehiculului.

#### 4.2.8.2. Determinarea situații vehiculului.

Proiecția celor două puncte **A**, **B** și a originii **O** în planul CCD-ului este reprezentată în figura 4.27 ( $x_c, y_c$  este sistemul de referință al camerei, iar **a** și **b** sunt imaginile punctelor aflate pe robocar).

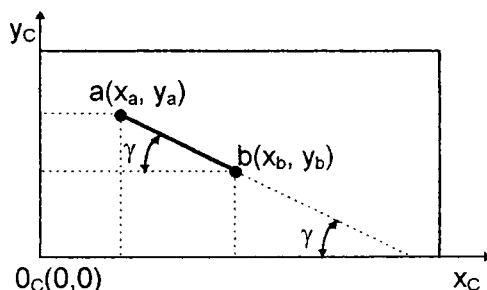


Figura 4.27.

Se măsoară, folosind un program realizat de către autor, coordonatele punctelor  $a(x_a, y_a)$  și  $b(x_b, y_b)$ , în pixeli. Se cunoaște, printr-o calibrare precedentă, numărul de milimetri corespunzător unui pixel  $k_x$  (pentru axa  $Ox$ ), respectiv  $k_y$  (pentru axa  $Oy$ ).

Coordonatele punctelor A și B în mm rezultă:

$$x_A \text{ [mm]} = x_a \text{ [pixeli]} \cdot k_x \text{ [mm/pixel]} \text{ și} \quad (4.72)$$

$$x_B \text{ [mm]} = x_b \text{ [pixeli]} \cdot k_x \text{ [mm/pixel]}$$

la fel:

$$y_A \text{ [mm]} = y_a \text{ [pixeli]} \cdot k_y \text{ [mm/pixel]} \text{ și} \quad (4.73)$$

$$y_B \text{ [mm]} = y_b \text{ [pixeli]} \cdot k_y \text{ [mm/pixel]}$$

Orientarea este dată de expresia  $\text{tg} \gamma = \frac{y_A - y_B}{x_A - x_B}$  (4.74)

Aceste rezultate sunt valabile numai dacă distanța de la cameră până la planul în care are loc mișcarea rămâne constantă.

În faza de instruire se memorează, la anumite intervale de timp, coordonatele punctelor aflate pe robocar. Atunci când deplasarea are loc în mod automat, pe traiectoria reală, se pot determina în fiecare moment diferențele dintre cele două traiectorii prin diferențele dintre coordonatele memorate și cele reale.

Urmează generarea traiectoriei de corecție, astfel încât la ieșirea din câmpul vizual al camerei, traiectoria reală va coincide cu cea prescrisă (învățată).

#### 4.2.9. Determinarea situației unui robocar cu cameră video amplasată pe vehicul.

Aplasarea camerei CCD pe vehicul prezintă un avantaj esențial și anume că nu este necesară existența unui sistem special (radio) de comunicare între sistemul de conducere al robocarului și echipamentul utilizat pentru achiziția imaginii.

##### 4.2.9.1. Principiul metodei.

Camera video se amplasează pe vehicul în punctul A (figura 4.28). De-a lungul traiectoriei pe care se deplasează vehiculul sunt amplasate două marcaje la înălțimea  $h$  și distanța  $d$  unul față de celălalt. Cele două marcaje pot fi două surse luminoase, așa cum s-a arătat în paragraful 4.2.8. Punctele fixe și punctul în care este amplasată camera pe robocar formează un triunghi oarecare, care are laturile și unghiurile cunoscute. Deci, poziția și orientarea vehiculului sunt unic determinate, prin măsurarea laturilor și a unghiurilor.

##### 4.2.9.2. Determinarea situației vehiculului.

În sistemul de axe de coordonate  $(x,y,z)$ , robocarul se află, ca și în celelalte cazuri în planul  $(xOy)$ . Pe axa  $Oz$  și în planul  $yOz$  sunt amplasate două puncte M și N, la înălțimile  $h_M$ , respectiv  $h_N$  și la distanța  $MM' = d$  unul față de celălalt (figura 4.28).

Cunoscând înălțimile  $h_M$ ,  $h_N$  și elevațiile celor două puncte  $\beta_M$  și  $\beta_N$ , se pot determina cele două laturi necunoscute ale triunghiului oarecare OAN':  $l_M$  respectiv  $l_N$ , după cum urmează:

$$\frac{h_M}{l_M} = \operatorname{tg} \beta_M \text{ și } \frac{h_N}{l_N} = \operatorname{tg} \beta_N \quad (4.75)$$

de unde:



$$\text{Deci: } x_A = \frac{l_M \cdot l_N}{d} \sin(\alpha_N - \alpha_M) \quad (4.80)$$

Din relația 4.77 mai rezultă:

$$\frac{d}{\sin(\alpha_N - \alpha_M)} = \frac{l_M}{\sin \delta_1} = \frac{l_N}{\sqrt{1 - \cos^2 \delta_1}} = \frac{l_M}{\sqrt{1 - \frac{y_A^2}{l_M^2}}} \quad (4.81)$$

$$\text{Deci: } \frac{d}{\sin(\alpha_N - \alpha_M)} = \frac{l_N \cdot l_M}{\sqrt{(l_M^2 - y_A^2)}} \quad (4.82)$$

$$d \sqrt{l_M^2 - y_A^2} = l_M \cdot l_N \cdot \sin^2(\alpha_N - \alpha_M)$$

$$d^2 \cdot l_M^2 - d^2 \cdot y_A^2 = l_M^2 \cdot l_N^2 \cdot \sin^2(\alpha_N - \alpha_M)$$

$$y_A^2 = l_M^2 - \frac{l_M \cdot l_N^2}{d^2} \cdot \sin^2(\alpha_N - \alpha_M)$$

$$y_A = \frac{l_M}{d} \cdot \sqrt{d^2 - l_N^2 \cdot \sin^2(\alpha_N - \alpha_M)} \quad (4.83)$$

$$\text{iar orientarea } \gamma = 180^\circ - (\alpha_N + \theta_1) \quad (4.84)$$

dar în triunghiul AA'N'

$$\operatorname{tg} \theta_1 = \frac{y_A - d}{x_A}$$

deci:

$$\theta_1 = \operatorname{arctg} \frac{y_A - d}{x_A} \quad (4.85)$$

Rezultă:

$$\gamma = 180^\circ - \left( \alpha_N + \operatorname{arctg} \frac{y_A - d}{x_A} \right) \quad (4.86)$$

Din relațiile (4.80) și (4.83) rezultă că, pentru determinarea coordonatelor robocarului sunt necesare unghiurile  $\alpha_M, \beta_M, \alpha_N, \beta_N$ . **În continuare se vor propune de către autor două soluții originale pentru măsurarea acestor unghiuri.**

#### 4.2.9.3. Determinarea situații cu camera video orientabilă, amplasată pe robocar.

Camera video este situată pe robocar. Un sistem de baleiere, realizat cu motoare pas cu pas, permite, prin rotirea camerei în plan vertical și

orizontal, modificarea azimutului și a elevației axei optice a acesteia. Când se dorește determinarea poziției robocarului, camera este rotită astfel încât punctul vizat se va afla pe axa optică a camerei, iar proiecția sa se va găsi în centrul elementului CCD. În acest moment unghiurile măsurate cu ajutorul traductoarelor sunt memorate și utilizate în calculul coordonatelor robocarului.

Dezavantajul principal al acestei metode este că centrarea în imagine a punctului vizat, descrisă mai sus, se face într-un interval mare de timp. Pentru scanare este necesară o deplasare mecanică a camerei, iar în acest timp robocarul se află în mișcare. Generarea mărimilor de conducere pe traiectoria de corecție începe în momentul în care s-a încheiat calculul coordonatelor și corespunde punctului pentru care acestea s-au determinat. Dar, robocarul se află acum într-un punct care are alte coordonate decât acela pentru care s-au generat mărimile de conducere, ceea ce face ca traiectoria de corecție reală să fie diferită de traiectoria de corecție planificată.

#### 4.2.9.4. Determinarea situației cu camera video fixă amplasată pe robocar.

Dezavantajul duratei mari necesare măsurării unghiurilor este înlăturat prin utilizarea **metodei propuse în continuare de către autor.**

Camera video este amplasată pe robocar. Unghiurile pe care axa optică a obiectivului camerei le formează cu axele  $O_V X_V$ , respectiv  $O_V Z_V$  ale sistemului de referință  $x_V, y_V, z_V$  solidar cu robocarul sunt  $\alpha_{cam}$  (azimutul) și  $\beta_{cam}$  (elevația). Aceste unghiuri sunt astfel alese încât vehiculul, deplasându-se pe traiectorie, camera video să achiziționeze imaginea unor puncte aflate în spațiul de lucru. Se utilizează câte două puncte amplasate de-a lungul traiectoriei, la intervale determinate pe baza criteriului abaterii minime admise. Distanța dintre ele și înălțimea la care sunt amplasate se cunosc. Atunci când aceste două puncte se află în **câmpul de vizibilitate** al camerei, se măsoară unghiurile  $\alpha$  și  $\beta$ . Aceste unghiuri sunt formate de către dreptele care leagă centrul optic al camerei cu fiecare punct și axele sistemului de referință în care are loc determinarea situației robocarului.

Metoda prezintă avantajul că determinarea situației se face în timp real, iar implicit corecția traiectoriei se face fără erorile care apar, prin utilizarea altor metode, datorită întârzierilor. Pentru o determinare corectă a unghiurilor necesare metodei de mai sus trebuie efectuate operații



de calibrare a camerei. Modul de determinare a unghiurilor este propus de către autor și va fi descris în paragraful 4.2.11.

#### 4.2.10. Metodă de calibrare a camerei CCD.

Pentru determinarea coordonatelor punctelor țintă (reperele staționare amplasate în spațiul de lucru sau cele mobile amplasate pe robocar), este necesară cunoașterea distanței dintre planul imaginii și centrul optic al camerei CCD. Modelul camerei considerat în descrierea următoare este *modelul Ito Ishii* [IAF 98] și se reprezintă ca în figura 4.30. Circuitul integrat CCD se află între obiectiv și centrul optic  $O_c$ , cât mai aproape de acesta, astfel încât imaginea care se formează să fie corect focalizată.

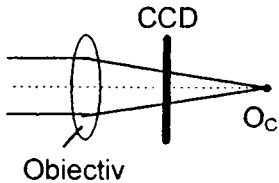


Figura 4.30.

Pentru calibrarea camerei se propune următoarea metodă de lucru: pe un banc optic prevăzut cu o masă de poziționare având rezoluția 0.01 mm se fixează camera CCD și corpul de referință pe care se află un punct țintă A (figura 4.31). Camera poate fi deplasată pe axele  $O_x$  și  $O_y$ , iar corpul de referință pe axa  $O_z$ .

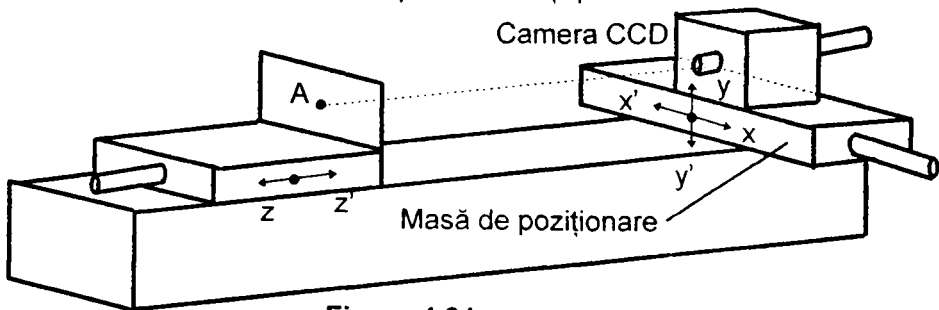


Figura 4.31.

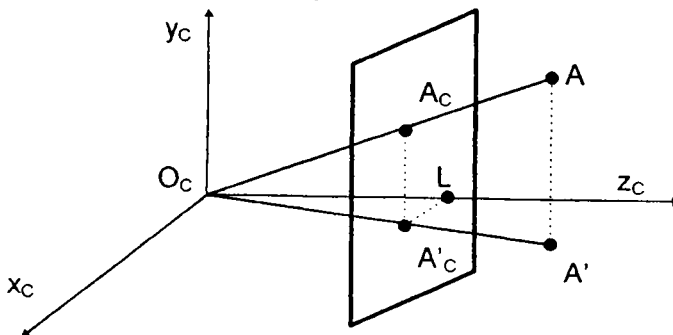


Figura 4.32.

În sistemul de referință al camerei CCD ( $x_c, y_c, z_c$ ) se formează imaginea punctului țintă A ( $A'_c$ ).  $O_c$  este centrul optic al camerei,  $A'$  este proiecția punctului A în planul  $x_c O_c z_c$ , iar L este intersecția axei optice a obiectivului camerei cu planul matricei CCD (figura 4.32). Se urmărește determinarea segmentului  $\overline{O_c L}$ . Pentru aceasta se va opera cu proiecțiile punctelor implicate pe planul ( $x_c, y_c, z_c$ ) (figura 4.33).

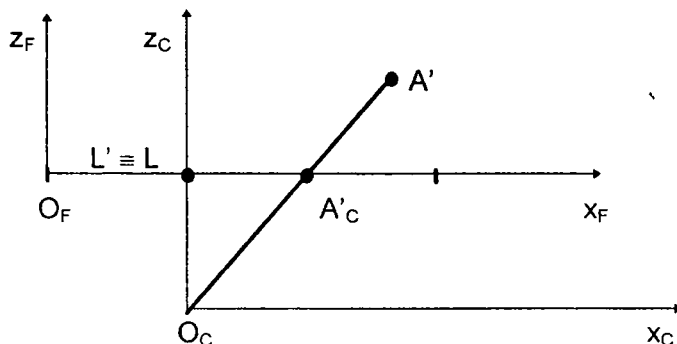


Figura 4.33.

Punctele  $A_c$ , L din planul CCD sunt localizate prin numărul de pixeli pe direcția x și z notat cu  $n_x$ , respectiv  $n_z$ . Pentru a determina numărul de pixeli pe axa  $O_F x_F$  corespunzător punctului L, se deplasează camera în direcția  $x x'$  și  $z z'$  până în momentul în care deplasarea pe  $z z'$  nu mai produce modificarea numărului de pixeli pentru punctul A. În acest moment  $n_{x_A} = n_{x_L}$ .

În continuare se execută o deplasare pe direcția  $x x'$  și apoi o singură deplasare pe direcția  $z z'$  cu distanța cunoscută  $\Delta z$ . Se obține configurația din figura 4.34.

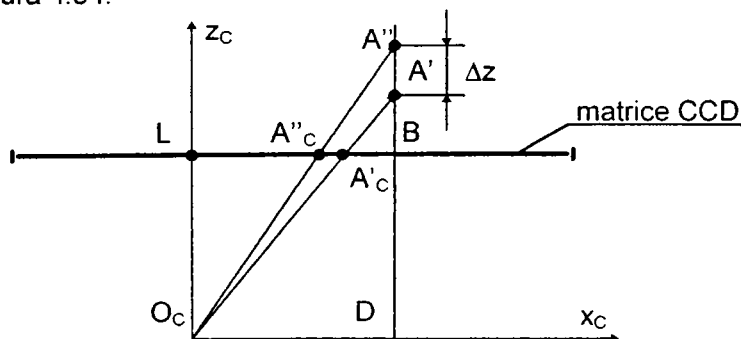


Figura 4.34.

Pentru elementele din figura 4.34 se cunosc:

$n_{x_1}, n_{x_2}, n_{x_c}$  - număr de pixeli

$O_cD$  - realizat cu masa de poziționare

$A'A'' = \Delta z$  - realizat cu masa de poziționare.

În triunghiul  $A''O_cD$ , teorema asemănării este:

$$\frac{\Delta z + A'B}{\Delta z + A'B + BD} = \frac{A_cB}{O_cD} \quad (4.87)$$

În triunghiul  $A'A_cB$ , teorema asemănării este:

$$\frac{A'B}{A'B + BD} = \frac{A_cB}{O_cD} \quad (4.88)$$

pentru simplificare se notează:

$$A_cB = x_1, A''B = x_2, A'B = a, BD = b, O_cD = c$$

și deci:

$$\begin{cases} \frac{\Delta z + a}{\Delta z + a + b} = \frac{x_2}{c} & (4.89) \\ \frac{a}{a + b} = \frac{x_1}{c} & (4.91) \end{cases} \Rightarrow b = a \cdot \frac{c - x_1}{x_1} \quad (4.90)$$

$$a = \frac{\Delta y \cdot x_1 \cdot (c - x_2)}{c \cdot (x_2 - x_1)} \quad (4.92)$$

$$b = \frac{\Delta z \cdot x_1 \cdot (c - x_2)}{c \cdot (x_2 - x_1)} \cdot \frac{c - x_1}{x_1} = \Delta z \cdot \frac{(c - x_1) \cdot (c - x_2)}{c \cdot (x_2 - x_1)} \quad (4.93)$$

Convertirea numărului de pixeli în milimetri se face prin cunoașterea dimensiunilor unei celule fotodectoare, de exemplu pentru camera CCD utilizată, TAC 43C celula are dimensiunile  $9,6 \cdot 10^{-3} \text{ mm} \times 8,4 \cdot 10^{-3} \text{ mm}$ .

#### 4.2.11. Procedeu pentru măsurarea unghiurilor cu camera video.

Cu ajutorul camerei video se determină unghiurile  $\alpha$  (azimut),  $\beta$  (elevație), care sunt folosite pentru determinarea situații VGA.

Se consideră un punct M, care are corespondentul  $M_c$  în planul fotodectoarelor camerei CCD (figura 4.35). Punctul  $M_c$  va avea coordonatele  $x_c, y_c, z_c$  după cum urmează:

$x_c$  - numărul de pixeli pe orizontală (determinat prin măsurare);

- $z_c$  - distanța de la planul de fotodectoare la centrul optic, notată cu  $b$  (cunoscută);
- $y_c$  - numărul de pixeli pe verticală (determinat prin măsurare).

Cu ajutorul coordonatelor  $x_c, y_c, z_c$  se determină azimutul și elevația ( $\alpha$  și  $\beta$ ) razei vectoriale a punctului țintă, care servesc în continuare pentru determinarea coordonatelor punctului vizat.

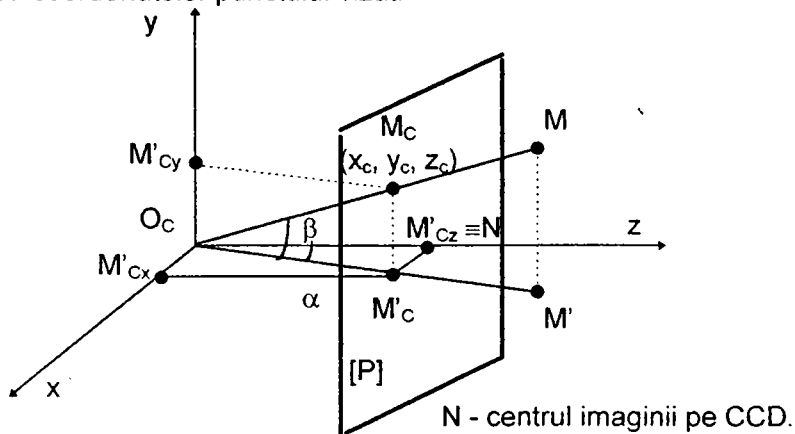


Figura 4.35.

De la acest sistem de referință se trece la orice alt sistem de referință, utilizat pentru corecția traiectoriei robocarului.

Dacă se reprezintă numai punctul  $M_c$  (proiecția punctului  $M$  pe planul CCD) se obțin relativ simplu unghiurile  $\alpha$  și  $\beta$  (figura 4.36).

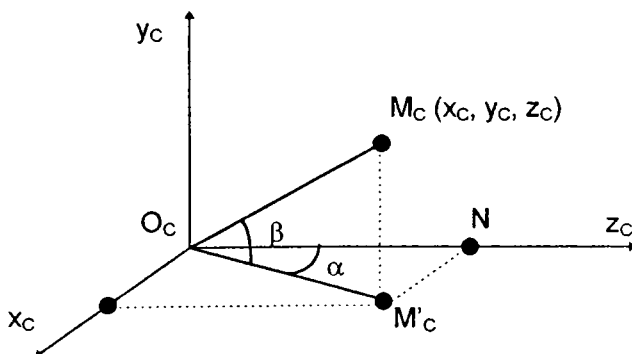


Figura 4.36.

Dacă se cunosc  $x_c = n_x$  - numărul de pixeli pe orizontală;  
 $y_c = n_y$  - numărul de pixeli pe verticală;  
 $p_x$  - dimensiunea pe orizontală a pixelilor camerei;  
 $p_y$  - dimensiunea pe verticală a pixelilor camerei;  
 $z_c = b$  - distanța dintre matricea CCD și centrul optic  
 $O_c$ .

Coordonatele punctului  $M_c$  (exprimate în mm) devin  $(X_c, Y_c, Z_c)$

$$X_c = x_c \cdot p_x = n_x \cdot p_x \quad [mm]$$

$$Y_c = y_c \cdot p_y = n_y \cdot p_y \quad [mm]$$

$$Z_c = b \quad [mm]$$

atunci:

$$\operatorname{tg} \beta = \frac{M_c \cdot M'_c}{OO_c \cdot M'_c} = \frac{Y_c}{\sqrt{X_c^2 + Z_c^2}} = \frac{n_y \cdot p_y}{\sqrt{(n_x \cdot p_x)^2 + b^2}} \quad (4.94)$$

$$\beta = \operatorname{arctg} \frac{n_y \cdot p_y}{\sqrt{n_x^2 \cdot p_x^2 + b^2}} \quad (4.95)$$

iar

$$\operatorname{tg} \alpha = \frac{M_c \cdot M}{OO_c \cdot N} = \frac{x_c}{y_c} = \frac{n_x \cdot p_x}{b} \quad (4.96)$$

$$\alpha = \operatorname{arctg} \frac{n_x \cdot p_x}{b} \quad (4.97)$$

Expresiile (4.95) și (4.97) sunt utilizate pentru determinarea unghiurilor cu ajutorul camerei video.

#### 4.2.12. Rutină pentru măsurarea distanței în pixeli, între două puncte, într-o imagine digitală

Elaborarea unei proceduri pentru măsurarea distanței în pixeli și, pe baza acesteia, a distanței în milimetri, este necesară în primul rând pentru efectuarea calibrării camerei video (paragraful 4.2.10), iar apoi pentru determinarea unghiurilor (paragraful 4.2.11). **Programul prezentat în cele ce urmează a fost realizat și experimentat de către autor.** Cu ajutorul acestuia s-a efectuat calibrarea camerei video care a fost utilizată în cadrul experimentelor prezentate în capitolul 7.

Măsurarea distanței decurge în câteva etape, necesare achiziționării și prelucrării unei imagini.

Prima etapă este achiziția imaginii și constă din preluarea unei imagini obținute cu ajutorul camerei video prin intermediul unei plăci de achiziție video și transpunerea ei într-un format care poate fi ușor prelucrat, în cazul de față într-un format bitmap.

Cea de-a doua etapă reprezintă procedura de filtrare și binarizare a imaginii obținute în prima fază, în vederea facilitării operației de măsurare propriu-zisă.

#### 4.2.12.1. Achiziția imaginii.

În vederea achiziționării imaginii video s-au folosit două componente principale:

- Componenta hardware alcătuită dintr-o cameră video CCD și o placă de achiziție, în cazul de față o placă All In Wonder Pro produsă de firma canadiană ATI. Această placă poate achiziționa imagini statice cu rezoluții de până la 1024 x 768 pixeli.
- Componenta software conține câteva proceduri care folosesc ca elemente de bază funcțiile oferite de biblioteca Video for Windows de la Microsoft.

Având în vedere faptul că echipamentele hardware pot fi foarte diferite, se vor prezenta numai procedurile soft, care sunt universal valabile indiferent de hardware-ul utilizat.

Prima procedură necesară în vederea achiziționării unei imaginii este rutina pentru inițializarea sistemului video și pentru pornirea modului de vizionare în timp real a imaginilor care vor fi achiziționate. Acest mod de vizionare permite aranjarea mediului de lucru și a **centrării** camerei video pe obiectele de interes. Etapa aceasta începe prin crearea unei ferestre în care este vizionat filmul procesului de achiziție (figura 4.37 a). În acest scop se creează o fereastră Windows de dimensiunile dorite (aceste dimensiuni nu influențează rezoluția de achiziție statică și din acest motiv se alege în general de dimensiuni mici, pentru a nu ocupa procesorul cu afișarea unor imagini mari în timp real; în exemplul prezentat ele se alege 146x110 din motive de aspect și de spațiu pe ecran).

Având fereastra Windows creată, urmează atașarea de această fereastră a unei ferestre de captură specifică lui Video for Windows, folosind funcția **capCreateCaptureWindow()**.

Următorul pas este determinarea prezenței diverselor plăci de achiziție care se află în sistem și alegerea uneia dintre ele, folosind funcția **capGetDriverDescription()**. Biblioteca Video for Windows suportă până la 10 plăci de achiziție diferite într-un sistem. În cazul unui sistem cu o singură placă de achiziție, numărul de ordine al driverului va fi 0. Dacă există o placă de achiziție în sistem, urmează inițializarea acesteia și pornirea achiziției în timp real cu vizionare în fereastra creată anterior.

Conectarea driverului la fereastra creată se realizează prin funcția **capDriverConnect()**. Următoarele faze de inițializare setează rata de achiziție a imaginii în milisecunde cu ajutorul funcției **capPreviewRate()**, scalarea imaginii la dimensiunile ferestrei cu funcția **capPreviewScale()** și pornirea vizionării propriu-zise prin **capPreview()**. În acest moment se va putea urmări în fereastra creată imaginea achiziționată cu ajutorul camerei video (figura 4.37 a).

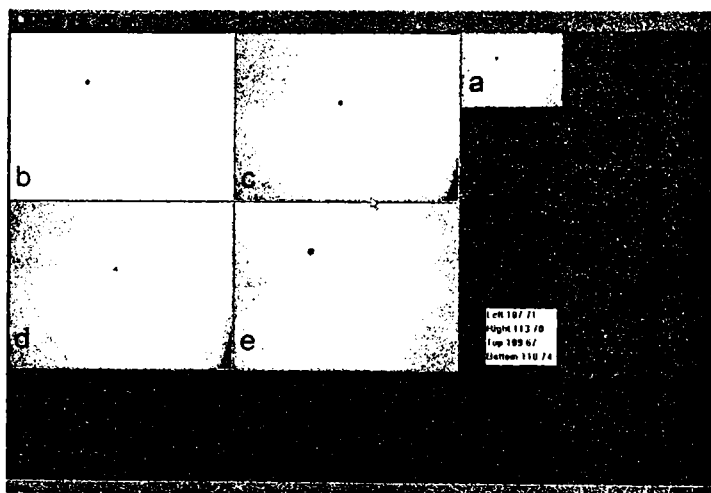


Figura 4.37.

Etapa următoare constă în achiziționarea unei imagini statice cu ajutorul funcției **capEditCopy()**, care salvează imaginea în format bitmap în clipboard-ul Windows-ului. Folosind funcții specifice clipboard-ului, cum ar fi **OpenClipboard()**, **GetClipboardData()** și **CloseClipboard()**, se încarcă în structura de document a programului imaginea achiziționată. Se utilizează clipboard-ul în locul salvării și citirii din fișier pentru creșterea vitezei și pentru ușurința utilizării acestuia cu formate bitmap.

### 4.2.12.2. Măsurarea distanței.

Camera video utilizată pentru achiziția imaginii poate fi color sau alb-negru. De aceea este necesară o procedură de conversie la 256 niveluri de gri în cazul în care imaginea se achiziționează cu ajutorul unei camere color. Faptul că imaginea este color, se determină prin compararea celor trei componente RGB ale unui pixel. Dacă, pentru cel puțin un pixel din imagine, una dintre cele trei culori RGB este diferită de celelalte două, rezultă că imaginea este color. În cazul imaginii color formula de conversie la niveluri de gri utilizată este următoarea:

$$\text{GrayValue} = 0.3 * R + 0.59 * G + 0.11 * B$$

După conversia la niveluri de gri urmează afișarea propriu-zisă a bitmap-ului în cadrul ferestrei prin refresh-ul aplicat acesteia cu ajutorul funcției **UpdateAllViews()**. Deoarece aplicația utilizată pentru măsurarea distanței se poate limita doar la două niveluri de culoare, alb și negru, urmează binarizarea imaginii prin stabilirea automată a unui prag de binarizare. Acesta este ales ca fiind egal cu intensitatea celui mai negru punct din imaginea cu 256 de niveluri de gri plus 10.

Având pragul de binarizare selectat, urmează crearea unei noi imagini binare din imaginea cu niveluri de gri, astfel încât orice pixel care are o intensitate mai mare decât pragul va fi alb, iar ceilalți vor fi negri. În figura 4.37 b se prezintă imaginea obținută în urma acestei procesări.

În vederea obținerii unei imagini clare și cu contururi precise, trebuie eliminate umbrele printr-o iluminare adecvată a mediului de lucru și prin crearea unui contrast puternic al imaginii.

Având imaginea binară creată și afișată pe ecran, printr-o simplă baleiere a tuturor pixelilor, se determină coordonatele pixelilor negri extremi din partea stângă, respectiv dreaptă, din partea de sus, respectiv de jos. Coordonatele astfel determinate se afișează pe ecran (figura 4.37 f).

Distanța va fi calculată simplu prin diferența acestor coordonate. Programele utilizate pentru achiziția imaginii și pentru calculul distanței vor fi prezentate în continuare.

Cu cât rezoluția achiziției este mai mare, cu atât vor rezulta mai mulți pixeli pe centimetru. Creșterea rezoluției se poate obține în două moduri:



- prin utilizarea unui program care dublează numărul de pixeli;
- prin utilizarea unei camere video speciale, cu o rezoluție foarte mare [EDS 98].

**În acest mod se va obține o precizie de măsurare ridicată.**

Procedura se încheie cu afișarea rezultatelor, după care se poate efectua o nouă măsurare.

### 4.2.12.3. Programul utilizat pentru măsurarea distanței dintre două puncte.

Programul utilizat pentru măsurarea distanței dintre două puncte este prezentat în continuare.

```
void CMainFrame::OnToolsMeasure()
{
    CROBOVIEWDoc *SourceDoc;
    CROBOVIEWDoc *DestinDoc;
    unsigned char *ClipboardBuffer;
    short x,y,TresholdValue;
    POINT PixelXmin,PixelXmax,PixelYmin,PixelYmax;
    short TypeOfCapture;
    char sir[40];
    CDC *ActiveFrameDC;
    RECT rect = {690,430,790,510};
    extern BOOL VideoCaptureInit;
    BYTE GrayValue;
    long IntensityNumber[256];

    //--> Realtime capture
    if(!VideoCaptureInit)
        VideoCapture();
    //<-- Realtime capture
    //--> Capturarea unei singure imagini si determinare carcateristici
    SourceDoc = GetActiveSourceDocument();
    if(SourceDoc->BitmapBuffer == NULL)
        SourceDoc->BitmapBuffer= (BYTE *)malloc(DIBSIZE);
    MyEditCopy();
    if(!MyOpenClipboard(NULL))
    {
        MessageBox("Error opening clipboard","Roboview",MB_OK);
    }
}
```

```

    return;
}
if(!(ClipboardBuffer=(unsigned char *)MyGetClipboardData(CF_DIB)))
{
    MessageBox("Error reading clipboard data","Roboview",MB_OK);
    return;
}
memcpy(SourceDoc->BitmapBuffer,ClipboardBuffer,DIBSIZE);
CloseClipboard();
if(SourceDoc->BitmapBuffer != NULL)
{
    SourceDoc->BitmapBits = GetDibBitsAddr(SourceDoc->BitmapBuffer);
    Xmax=GlobalBitmapInformation.XDimension=
        GetDibWidth(SourceDoc->BitmapBuffer);
    Ymax=GlobalBitmapInformation.YDimension=
        GetDibHeight(SourceDoc->BitmapBuffer);
}
for(y=Ymax-1;y>=0;y--)
    for(x=0;x<Xmax;x++)
    {
        SourceDoc->BitmapMatrix[x][y][0]=
            *(SourceDoc->BitmapBits+3*Xmax*(Ymax-1-y)+3*x);
        SourceDoc->BitmapMatrix[x][y][1]=
            *(SourceDoc->BitmapBits+3*Xmax*(Ymax-1-y)+3*x+1);
        SourceDoc->BitmapMatrix[x][y][2]=
            *(SourceDoc->BitmapBits+3*Xmax*(Ymax-1-y)+3*x+2);
    }
for(x=0;x<Xmax;x++)
    for(y=REALYMAX;y<Ymax;y++)
        SourceDoc->BitmapMatrix[x][y][0]=
        SourceDoc->BitmapMatrix[x][y][1]=
        SourceDoc->BitmapMatrix[x][y][2]= 0xFF;

TypeOfCapture = GRAYSCALE;
for(x=0;x<Xmax;x++)
    for(y=0;y<REALYMAX;y++)
        if((SourceDoc->BitmapMatrix[x][y][0]!=
            SourceDoc->BitmapMatrix[x][y][1])||
            (SourceDoc->BitmapMatrix[x][y][0]!=
            SourceDoc->BitmapMatrix[x][y][2])||
            (SourceDoc->BitmapMatrix[x][y][1]!=
            SourceDoc->BitmapMatrix[x][y][2]))
            TypeOfCapture = TRUECOLOR;

```

```

if(.TypeOfCapture = TRUECOLOR)
  for(x=0;x<Xmax;x++)
    for(y=0;y<REALYMAX;y++)
      {
        GrayValue=(BYTE)(0.3*SourceDoc->BitmapMatrix[x][y][0]+
          0.59*SourceDoc->BitmapMatrix[x][y][1]+
          0.11*SourceDoc->BitmapMatrix[x][y][2]);
        SourceDoc->BitmapMatrix[x][y][0]=
        SourceDoc->BitmapMatrix[x][y][1]=
        SourceDoc->BitmapMatrix[x][y][2]= GrayValue;
      }

SourceDoc->UpdateAllViews(NULL);
//<- Capturarea unei singure imagini si determinare carcateristici
//-> Binarizarea imaginii
memset(IntensityNumber,0,sizeof(IntensityNumber));
for(x=0;x<Xmax;x++)
  for(y=0;y<REALYMAX;y++)
    IntensityNumber[SourceDoc->BitmapMatrix[x][y][0]]++;
for(x=0;x<256;x++)
  if(IntensityNumber[x])
    {
      TresholdValue = IntensityNumber[x]+10;
      break;
    }
DestinDoc      =      OpenNewDestinationDocument(SourceDoc-
>BitmapColorType);
for(x=0;x<Xmax;x++)
  for(y=0;y<REALYMAX;y++)
    if(SourceDoc->BitmapMatrix[x][y][0] <= TresholdValue)
      DestinDoc->BitmapMatrix[x][y][0]=
      DestinDoc->BitmapMatrix[x][y][1]=
      DestinDoc->BitmapMatrix[x][y][2]= 0;
    else
      DestinDoc->BitmapMatrix[x][y][0]=
      DestinDoc->BitmapMatrix[x][y][1]=
      DestinDoc->BitmapMatrix[x][y][2]= 255;
DestinDoc->BitmapColorType = BINARY;
DestinDoc->UpdateAllViews(NULL);
//<- Binarizarea imaginii
PixelXmin.x = 10000;
PixelXmin.y = 0;
PixelXmax.x = 0;

```

```

PixelXmax.y = 0;
PixelYmin.x = 0;
PixelYmin.y = 10000;
PixelYmax.x = 0;
PixelYmax.y = 0;
for(y=Ymax-1;y>=0;y--)
  for(x=0;x<Xmax;x++)
    {
      if(DestinDoc->BitmapMatrix[x][y][0] == 0)
        {
          if(x < PixelXmin.x)
            {
              PixelXmin.x = x;
              PixelXmin.y = y;
            }
          if(x > PixelXmax.x)
            {
              PixelXmax.x = x;
              PixelXmax.y = y;
            }
          if(y < PixelYmin.y)
            {
              PixelYmin.x = x;
              PixelYmin.y = y;
            }
          if(y > PixelYmax.y)
            {
              PixelYmax.x = x;
              PixelYmax.y = y;
            }
        }
    }
ActiveFrameDC = AfxGetMainWnd()->GetDC();
ActiveFrameDC->SetTextColor(RGB(255,0,0));
ActiveFrameDC->SetBkMode(OPAQUE);
ActiveFrameDC->ExtTextOut(690,430,ETO_OPAQUE,&rect,"
",1,NULL);
sprintf(sir,"Left %d %d",PixelXmin.x,PixelXmin.y);
ActiveFrameDC->TextOut(690,430,sir,strlen(sir));
sprintf(sir,"Right %d %d",PixelXmax.x,PixelXmax.y);
ActiveFrameDC->TextOut(690,450,sir,strlen(sir));
sprintf(sir,"Top %d %d",PixelYmin.x,PixelYmin.y);
ActiveFrameDC->TextOut(690,470,sir,strlen(sir));

```

```

printf(sir,"Bottom %d %d",PixelYmax.x,PixelYmax.y);
ActiveFrameDC->TextOut(690,490,sir,strlen(sir));
}

```

### 4.3. Concluzii.

După cum s-a arătat, unul dintre procedeele de navigare se bazează pe determinarea situații unui robocar, aflat în spațiul de lucru, pe baza măsurărilor odometrice. Acest mod de determinare presupune măsurarea spațiilor parcurse și a unghiurilor de virare cu ajutorul traductoarelor incrementale. Datorită acumulării erorilor în determinarea situații (paragraful 3.1), este necesară realizarea unei corecții a traiectoriei. Corecția traiectoriei are loc pe baza actualizării situații robocarului, utilizând sisteme de referință atașate unor puncte amplasate de-a lungul traiectoriei. Deplasarea robocarului decurge ca în figura 4.38.

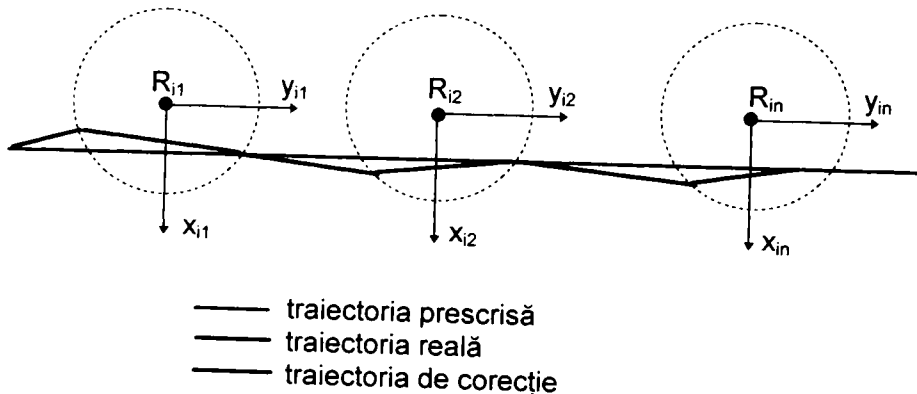


Figura 4.38.

Amplasarea sistemelor de referință se face astfel încât, la intrarea într-un sistem de referință, robocarul să se afle pe o traiectorie a cărei abatere de la traiectoria prescrisă să nu fie mai mare decât o abatere maximă prescrisă. Pentru a determina situația robocarului (coordonatele  $x$ ,  $y$  și orientarea  $\gamma$ ), se folosesc mai multe procedee, bazate pe diverse tipuri de senzori.

**Obiectul prezentului capitol l-a constituit prezentarea și analiza unor procedee pentru determinarea situații, care sunt contribuții originale ale autorului.**

Deoarece determinarea situației reprezintă, în primul rând, determinarea unor coordonate, iar aceste coordonate sunt de fapt **distanțele** măsurate pe axele  $O_x$  și  $O_y$ , rezultă că, oricare ar fi procedeul utilizat, acesta se rezumă la măsurarea unor distanțe. De aici se poate desprinde concluzia că, pentru determinarea situației, sunt necesare sisteme senzoriale cu ajutorul cărora să fie posibilă măsurarea unor distanțe, sau în anumite cazuri, măsurarea unor unghiuri.

În continuare sunt analizate metodele în cadrul cărora se utilizează pentru determinarea situației fascicule laser și camere video. Cu ajutorul ultrasunetelor se măsoară distanțe, iar procedeele de măsurare sunt în general cunoscute. Măsurarea poate avea loc în timpul deplasării vehiculului numai cu acceptarea unor erori, cauzate și dependente de viteza de deplasare a acestuia.

Toate metodele prezentate, care utilizează pentru determinarea situației fasciculele laser și camera video, se bazează pe existența a cel puțin trei puncte. Aceste puncte formează un triunghi ale cărui laturi și unghiuri pot fi determinate. Punctele pot fi amplasate astfel:

- ◆ două puncte pe robocar (mobile) și unul în spațiul de lucru (fix);
- ◆ un punct pe robocar (mobil) și două puncte în spațiul de lucru (fixe).

În punctele respective se amplasează diferite elemente, cum ar fi:

- ◆ marcaje pasive;
- ◆ marcaje active (surse de lumină IR);
- ◆ fotoreceptoare;
- ◆ surse de lumină laser;
- ◆ camere video.

Între elementele amplasate se pot realiza combinațiile:

- ◆ laser și fotoreceptoare;
- ◆ cameră video și marcaje pasive;
- ◆ cameră video și marcaje active.

Pentru o apreciere a calității metodelor prezentate, se vor avea în vedere următoarele considerente:

- ◆ **numărul surselor laser.** În cadrul unor metode se folosesc două surse laser și unul sau mai multe fotoreceptoare (4.2.6), sau o singură sursă laser și mai multe fotoreceptoare (4.2.5; 4.2.7).
- ◆ **numărul camerelor video.** Dacă se amplasează camere video fixe în fiecare sistem de referință, sunt necesare atâtea camere

video câte sisteme de referință sunt (4.2.8). Dacă se amplasează camera video pe robocar, este necesară o singură cameră (4.2.9.3; 4.2.9.4)

- ◆ **numărul punctelor active (pasive).** Dacă se utilizează camera video, aceste puncte pot fi pasive, dar pentru ușurința detecției se pot utiliza surse pulsatorii de radiație infraroșie. Punctele amplasate pe robocar sunt în număr de două. Dacă sunt amplasate în sistemele de referință, au numărul egal cu dublul numărului sistemelor de referință.
  - ◆ **numărul fotoreceptoarelor.** Pentru determinarea situații în spațiul 3D sunt necesare cel puțin patru fotoreceptoare amplasate pe robocar, iar în spațiul 2D sunt necesare două fotoreceptoare.
  - ◆ **comunicarea.** Atunci când se utilizează fotoreceptoare și scanere cu fascicule laser, trebuie realizată comunicarea între unitățile de comandă ale echipamentelor fixe și ale celor mobile.
- Observație:** prin utilizarea unui sistem optic adecvat (reflector), fotoreceptorul poate fi amplasat lângă emițător.
- ◆  **timpul necesar procesării informațiilor** achiziționate cu ajutorul sistemelor senzoriale este timpul în care se obține informația despre situație. Pentru scanare este necesar un anumit timp (mai redus atunci când scanarea are loc cu fascicule laser și mai mare atunci când scanarea are loc cu ajutorul camerei video). De asemenea, pentru procesarea informației este necesar un timp mai lung în cazul utilizării camerei video și mai scurt atunci când se utilizează fascicule laser. În tot acest interval de timp necesar procesării informației, robocarul se deplasează.
  - ◆ **precizia de măsurare** a unghiurilor cu ajutorul sistemelor utilizate. Precizia de măsurare este foarte bună în cazul utilizării unei surse laser cu fascicul bine focalizat și fotoreceptoare directive. În cazul camerelor video, precizia depinde de rezoluția acestora. Ea crește mult prin utilizarea camerelor video speciale, care au rezoluția ridicată [EDS 98].

Pe baza considerentelor prezentate mai sus se poate alege o metodă care are performanțe foarte bune, ieftină și ușor aplicabilă.

**Metodele descrise în paragrafele 4.2.5, 4.2.7, 4.2.8, 4.2.9.3, 4.2.9.4 sunt originale și sunt propuse de către autor. Atât principiul, cât și calculul coordonatelor și a orientării sunt de asemeni contribuții ale autorului.**

Metodele propuse și criteriile de apreciere a acestora pot fi reunite într-un tabelul 4.1.

Tabelul 4.1.

Metoda \ Criteriul	Nr. de surse laser a / b	Nr. de puncte a / b	Nr. de detect. a / b	Nr. de cam. Video a / b	Comunicare	Timp prelucrare	Prezisie
Scanner laser în spațiul 3D (4.2.5)	1/1×n	4/4	4/4	-	da	mare	f.bună
Scanner laser în spațiul 2D (4.2.6)	2/2×n	2/2	2/2	-	da	mare	f.bună
Scanner laser în spațiul 3D (4.2.7)	1/1×n	2/2	2/2	-	da	mare	f.bună
Camăra video staționară (4.2.8)	-	2/2	-	1/1×n	da	mic	bună/ f.bună
Camăra video orientabilă pe vehicul (4.2.9.3)	-	2/2×n	-	1/1	nu	f.mare	bună/ f.bună
Camăra video fixă pe vehicul (4.2.9.4)	-	2/2×n	-	1/1	nu	mic	bună/ f.bună

a - reprezintă un sistem de referință

b - reprezintă n sisteme de referință.

Din tabelul 4.1 se desprinde concluzia că metoda de determinare a situații cu camera video fixă amplasată pe vehicul (4.2.9.4) necesită o singură cameră video, fără sistem mecanic și electronic de orientare. Punctele vizate sunt câte două pentru fiecare sistem de referință în parte, dar amplasarea lor este simplă și nu necesită cheltuieli mari.

În ceea ce privește comunicarea, aceasta nu este necesară deoarece camera este amplasată pe robocar, acolo unde se află sistemul de conducere.

Timpul necesar pentru determinarea situații vehiculului este redus deoarece camera este fixă și nu este necesară scanarea mecanică, iar



durata prelucrării informației preluate prin intermediul camerei este redusă prin utilizarea unui program special destinat acestui scop (paragraful 4.2.12).

Se poate deci considera că situarea se determină în timp real și poate avea loc în regim dinamic.

Precizia determinării coordonatelor și a orientării este bună și este determinată cu ajutorul experimentelor prezentate în capitolul 7.

**În concluzie se apreciază că, pe baza cercetărilor efectuate se pot propune câteva metode pentru determinarea situații robocarului, care, împreună cu odometria, se pot aplica cu succes în mediul de lucru industrial.**

## Capitolul 5

### CONCEPȚIA ȘI REALIZAREA SISTEMULUI MECANIC AL ROBOCARULUI.

În nenumărate cazuri, scopul final spre care se îndreaptă acțiunile realizate cu ajutorul unui echipament electronic și/sau ale unui sistem de comandă numeric, împreună cu programele aferente, este **efectuarea unui lucru mecanic**. Din multitudinea exemplurilor, unele dintre ele foarte des întâlnite și binecunoscute, face parte și cel de față, **robocarul**.

Fiind amintit **lucrul mecanic** efectuat, nu se poate exclude din teza de față prezentarea elementelor constructive mecanice, a echipamentelor electrice și electronice de execuție și a sistemului de conducere, ca părți componente esențiale ale **robocarului construit de către autor**. Cu alte cuvinte, se consideră ca având o importanță egală cu a celorlalte probleme tratate și realizarea sistemului mecanic al robocarului. Mai mult decât atât, **această realizare a constituit suportul pentru experimentarea unor soluții, a unor echipamente electronice și a unor programe originale, care pot fi elemente constitutive ale unui vehicul modern, destinat transportului industrial automat**.

În capitolul de față se va prezenta modul de realizare a sistemului mecanic și elementele care aparțin acestei construcții. De asemenea se vor prezenta parametrii motoarelor, reductoarelor și traductoarelor de poziție folosite. Acești parametri vor fi utilizați pentru dimensionarea elementelor de execuție și pentru conceperea programelor necesare sistemului de conducere, ale căror prezentare se va face în capitolul următor.

Elementele sistemului mecanic pot avea diverse forme constructive. Aceste variante sunt în general cunoscute. **Sarcina autorului a fost găsirea unei soluții cu performanțe foarte bune și care poate fi adaptată condițiilor de realizare disponibile.**

Sistemul mecanic prezentat în continuare a fost ales avându-se în vedere câteva criterii, și anume:

- ◆ soluțiile adoptate să fie cât mai simple și fiabile;
- ◆ costul materialelor și al manoperei să fie cât mai redus;
- ◆ să asigure performanțe dinamice bune;
- ◆ să posede o greutate redusă;

- ◆ să existe posibilitatea modificării sau extinderii simple a construcției;
- ◆ să ofere posibilitatea montării unui sistem de încărcare-descărcare.

Din studiul multitudinii de soluții, avându-se în vedere posibilitățile de realizare, s-a optat pentru o variantă care satisface toate criteriile prezentate. A rezultat o construcție compactă, originală, care a fost ușor pusă în funcțiune și care înglobează majoritatea elementelor constructive ale unui vehicul ghidat automat actual.

**Contribuția autorului la realizarea acestui vehicul este, în primul rând, stabilirea cerințelor impuse construcției mecanice, studiul variantelor cunoscute, alegerea soluției care să satisfacă criteriile enunțate și realizarea fizică a robocarului. Robocarul satisface exigențele tehnice actuale, cu toate că este realizat cu mijloace simple și cu cheltuieli reduse.**

Sistemul mecanic al robocarului realizat conține următoarele elemente constructive:

- șasiul;
- mecanismul de tracțiune;
- mecanismul de virare;
- dispozitivul mecanic de protecție (bare antișoc);
- dispozitivul de încărcare / descărcare.

Soluțiile adoptate pentru sistemul mecanic și motivele alegerii acestor variante sunt:

- ◆ **șasiul** este de forma paralelipipedică. El poate fi ușor realizat și oferă utilizatorului o structură rezistentă, făcând posibilă totodată montarea simplă a tuturor subansamblurilor;
- ◆ **sistemul de rulare** este echipat cu trei roți. El poate fi folosit atât pentru mersul înainte, cât și pentru mersul înapoi și prezintă o soluție simplă în ceea ce privește construcția mecanică. Pentru acest sistem nu se utilizează suspensia elastică.
- ◆ **mecanismul de virare** acționează asupra roții anterioare. El este simplu, iar prin extinderea sa spre roțile posterioare, face posibilă obținerea unei raze mici de virare, atât atunci când are loc deplasarea înainte, cât și înapoi. Nu este necesar un mecanism cu cremalieră.

- ◆ **sistemul de tracțiune** acționează asupra unei singure roți. Aceasta este o soluție constructivă care nu necesită antrenarea diferențială (electrică sau mecanică). Motorul antrenează mecanismul de tracțiune prin intermediul reductorului.

## 5.1. Șasiul.

Șasiul reprezintă structura de rezistență a robocarului. Destinația sa este, așa cum s-a arătat în paragraful 1.1.1.1, susținerea sistemului de rulare, a echipamentelor electronice și a surselor de alimentare. De asemenea el constituie suportul pentru dispozitivele de transfer a sarcinii. Forma constructivă și dimensiunile șasiului depind atât de forma și dimensiunile elementelor componente, cât și de destinația vehiculului. Dacă robocarele sunt utilizate în sfera serviciilor, se aleg pentru șasiu variante constructive **prietenoase**, având un design cât mai atractiv. Dacă vehiculul are ca destinație transportul automat industrial, construcția sa se adaptează condițiilor de lucru. Șasiul robocarului **construit de către autor** este realizat în formă paralelipipedică, din țevă de oțel cu secțiunea dreptunghiulară (20×40)mm, sudată (figura 5.1).

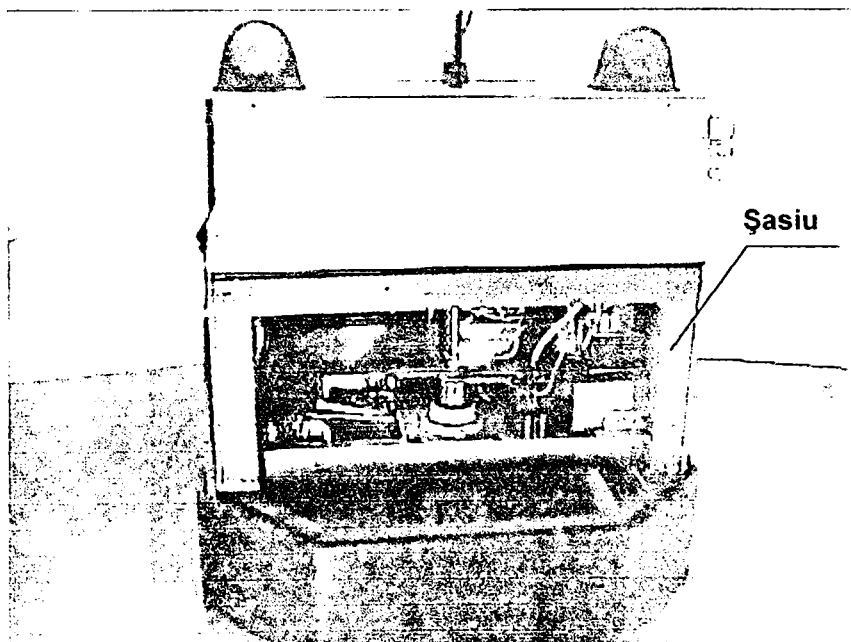


Figura 5.1.

Șasiul este acoperit cu tablă din aluminiu cu grosimea de 0,8 mm, iar platforma posterioară este realizată din tablă de oțel cu grosimea de 4 mm și constituie suportul pentru robotul **SCORBOT**. Acesta este dispozitivul de încărcare-descărcare al vehiculului. În partea frontală a șasiului se află un compartiment care conține echipamentul electronic și un monitor. Deasupra compartimentului se află două lămpi, utilizate pentru indicarea stării normale de funcționare sau a apariției unei avarii și camerele video necesare pentru determinarea situației robocarului. În partea posterioară, sub platformă, se montează modulul de comandă al robotului **SCORBOT**.

Pentru alimentarea robocarului se utilizează două acumulatori cu plumb conectate în serie, care au tensiunea nominală 12V și capacitatea nominală 45Ah fiecare. Ele asigură o autonomie în funcționare de cel puțin 8ore, dacă au fost încărcate la capacitatea maximă. Acumulatorii sunt amplasați în zona centrală a robocarului, la o înălțime de 5cm față de sol, **pentru a coborî** centrul de greutate al vehiculului. Încărcarea acumulatorilor se face cu ajutorul unui echipament de încărcare, cuplat cu un conector amplasat în partea stângă a robocarului.

## 5.2. Sistemul de rulare.

Sistemul de rulare cuprinde toate elementele prin intermediul cărora vehiculul intră în contact cu calea de rulare și care sunt utilizate pentru deplasare. Aceste elemente sunt în cazul de față roțile. Amplasarea roților și tipurile acestora (active/pasive) au fost prezentate în capitolul 1 (figura 1.4). Pentru sistemul de rulare al robocarului realizat s-a ales soluția constructivă care utilizează trei roți, amplasate ca în figura 5.2, dintre care două sunt pasive (cele posterioare), iar una este activă.

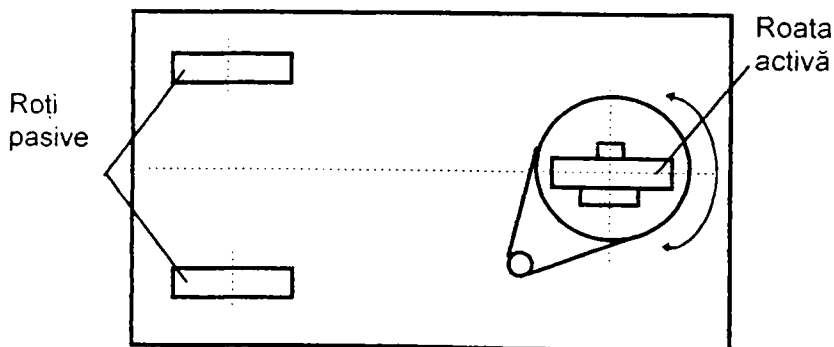


Figura 5.2.

Roata activă, aflată în partea anterioară a vehiculului, este antrenată de către mecanismul de tracțiune și este rotită în jurul axei verticale de către mecanismul de virare.

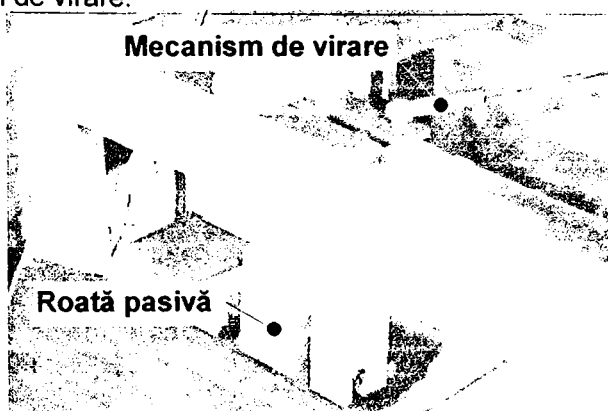


Figura 5.3.

Diametrul roților este  $D=100$  mm. Ele sunt fixate pe ax prin intermediul unui rulment. Axul este fixat într-o furcă care se poate roti în jurul axei verticale. Prinderea furcii de șasiu se face cu ajutorul unor rulmenți. Ansamblul roții pasive este prezentat în figura 5.3., iar cel al roții active în figura 5.4.

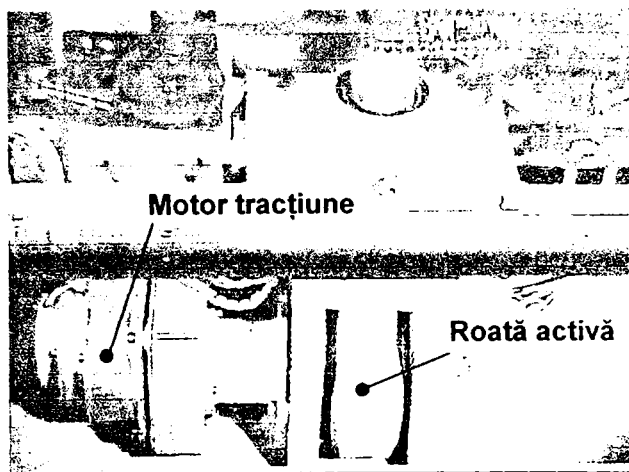


Figura 5.4.

### 5.2.1. Mecanismul de tracțiune.

Varianta constructivă aleasă pentru mecanismul de tracțiune este amplasarea motorului și a reductorului pe ansamblul roții față, care face parte și din mecanismul de virare. Considerentul principal care a stat la

baza acestei opțiuni este simplitatea construcției mecanice și ușurința realizării elementelor de execuție și a circuitelor de comandă ale acestora, însoțită de performanțe foarte bune.

Dacă s-ar fi ales o variantă constructivă cu două roți de tracțiune, ar fi fost necesar un sistem diferențial mecanic (pentru varianta cu un motor de tracțiune) [HAM 86] sau o comandă diferențială a elementelor de execuție (pentru varianta cu două motoare de tracțiune) [HEL 96], [LEU 97].

Motorul ales pentru tracțiune este un motor de curent continuu cu rotor disc, având dimensiuni reduse și randament ridicat. Tensiunea de alimentare nominală a motorului este 24V, iar curentul consumat în sarcină este 1,5A. Turația maximă este  $n_{\max} = 3000 \text{ rot/min}$ . S-a ales acest motor din considerente energetice. Este preferabil un motor cu randament foarte bun și caracteristici dinamice bune, pentru a asigura un consum redus și o eficiență ridicată la accelerare și frânare.

Reductorul are gabarit redus și un raport de transmitere  $i = 1 : 34,6$ . El este cuplat cu axul roții motoare.

Prin intermediul acestui ansamblu se asigură o viteză de deplasare :

$$v_{\max} [m / s] = \frac{n_{\max} [rot / min]}{60} \cdot i \cdot \frac{\pi \cdot D [mm]}{1000}$$

$$v_{\max} = 0,46 m / s$$

Pe axul motorului este fixat un traductor inductiv (figura 5.4). Acesta furnizează o tensiune sinusoidală alternativă, a cărei frecvență și amplitudine ( $e = -n \cdot \frac{d\phi}{dt}$ ) este dependentă de turație. Numărul de perioade ale semnalului generat de traductor, corespunzător unei rotații este  $n_1 = 50$  perioade/rot. Deoarece semnalul este sinusoidal și poate avea amplitudinea redusă la o turație scăzută, acesta este procesat cu ajutorul unui bloc electronic, care este prezentat în detaliu în capitolul 6.

Considerând că numărul de impulsuri furnizate de către blocul de procesare al semnalului generat de către traductor în timpul unei rotații complete a motorului este  $n_1 = 50$ , rezultă că numărul de impulsuri furnizate de către acest bloc, dacă robocarul parcurge un centimetru este:

$$n_i = \frac{\frac{n_1}{i_1}}{\pi \cdot D[\text{cm}]} = \frac{\frac{50}{1}}{\pi \cdot 10} = 55 \text{ imp / cm}$$

Aceste mărimi sunt necesare pentru determinarea prin soft a parametrilor mișcării.

### 5.2.2. Mecanismul de virare.

Structura mecanică aleasă pentru rulare și pentru virare este varianta constructivă cu trei roți (una activă și două pasive), dintre care una este și roată directoare. Asupra furcii roții directoare acționează, prin intermediul unui reductor și a unei transmisii cu lanț, motorul pas cu pas al sistemului de virare. S-a ales această soluție datorită simplității ei și totodată a manevrabilității sale ușoare. Utilizarea motorului pas cu pas s-a impus ca necesară din două motive:

- este posibilă folosirea unui sistem de reglare a poziției în buclă deschisă (fără traductor de poziție);
- se poate realiza blocarea mecanismului de virare prin menținerea alimentată a cel puțin unei faze a motorului.

Mecanismul de virare se menține în stare blocată și datorită reductorului folosit, care realizează un raport de transmitere de 1 : 125, dar pentru siguranță se preferă și blocarea electrică (descrisă în capitolul următor).

Pentru reducerea razei de virare, sistemul de virare a fost îmbunătățit, fiind completat cu un mecanism care transmite mișcarea de rotație celor două roți pasive. Structura și analiza matematică a mecanismului de virare vor fi prezentate în paragraful 5.2.3.

Motorul utilizat pentru rotirea mecanismului de virare este un motor pas cu pas cu patru faze alimentate unipolar și are următorii parametri:

- tensiune nominală  $U_N = 12 \text{ V}$
- rezistența înfășurării  $R_m = 30 \Omega$
- inductivitatea înfășurării  $L_m = 7 \text{ mH}$
- unghiul de pas  $\varphi_{\text{pas}} = 7,5^\circ$  (48 pași / rotație)
- cuplul nominal  $C_N = 0,25 \text{ Nm}$

Motorul antrenează axul vertical al roții directoare prin intermediul unui reductor și al unui sistem de transmisie cu lanț, așa cum rezultă din figura 5.5.



Utilizând parametrii sistemului prezentat mai sus se poate deduce numărul de impulsuri de tact  $N_1$ , necesar pentru ca motorul pas cu pas să rotească mecanismul de virare cu 360 grade.

$$N_1 = 48 \times 125 = 6000 \text{ imp} / 360^\circ$$

La o deplasare unghiulară cu un grad corespund:

$$n_1 = \frac{6000}{360} = 16,66 \text{ imp} / \text{grad}$$

Mișcarea se transmite de la roata directoare unui traductor incremental printr-o curea. Traductorul furnizează  $n_4 = 96 \text{ imp/rotație}$ . Rezultă că, pentru o rotație completă a sistemului de virare se vor genera:

$$N_2 = \frac{32}{25} \cdot \frac{70}{10} \cdot 96 = 860,16 \text{ imp} / \text{rot} \quad \text{sau}$$

$$n_2 = \frac{860,16}{360} = 2,38 \text{ imp} / \text{grad}$$

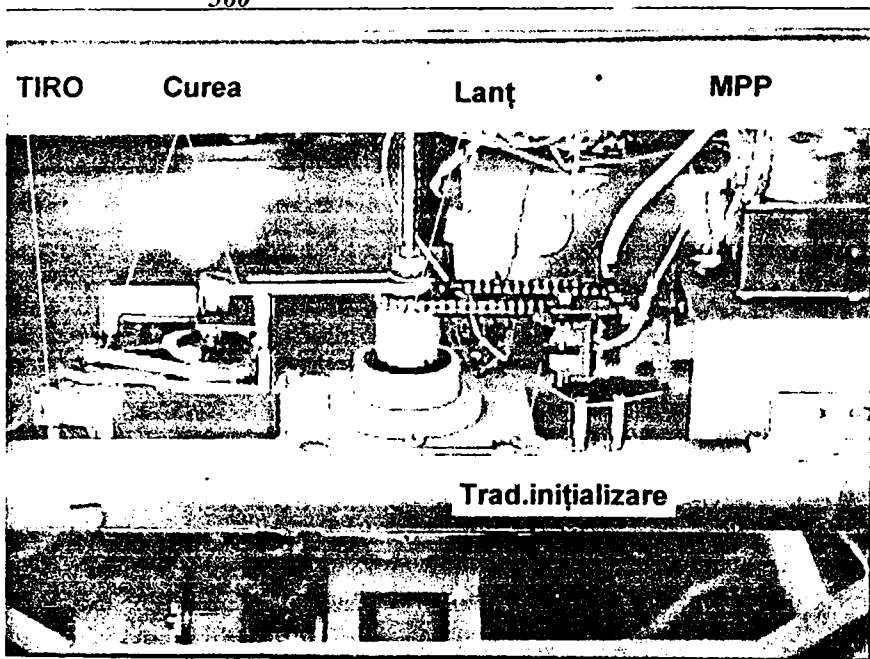


Figura 5.5.

Sistemul de virare necesită o inițializare prealabilă, care se realizează prin utilizarea unui optocuplor, amplasat astfel încât să fie obturat atunci când sistemul de virare a fost rotit spre stânga, depășind unghiul maxim admis. Acest unghi poate fi stabilit printr-un reglaj mecanic. În figura 5.5. se prezintă întregul mecanism descris mai sus.

### 5.2.3. Calculul mecanismului de virare.

În cele ce urmează se prezintă structura și calculul unui mecanism de virare. Acesta transmite mișcarea de rotație de la sistemul de virare, care acționează asupra roții de tracțiune, celor două roți pasive. Unghiurile de virare ale celor trei roți sunt diferite, după cum și razele traiectoriilor pe care se deplasează sunt diferite [KOV 98].

#### 5.2.3.1. Calculul unghiurilor de virare $\theta = \theta ( R )$ .

a). Roțile din spate nedirectoare, roata din față directoare.

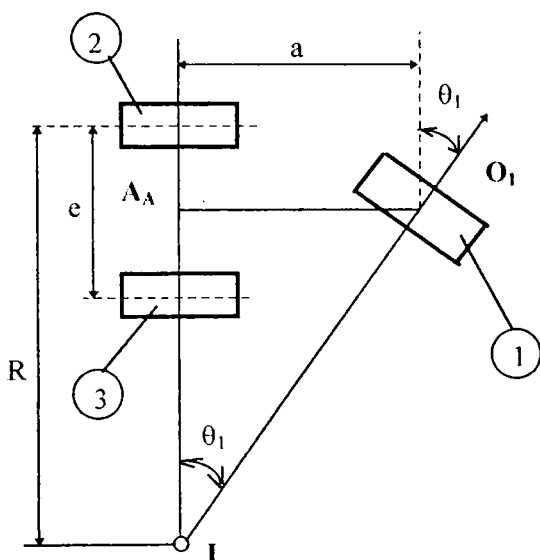


Figura 5.6.

Din triunghiul  $\Delta A_A O_1 I$  rezultă:

$$\theta_1 = \arctg \frac{a}{R} \quad (5.1)$$

R	$\theta_1$
$\infty$	0
$R_{min}$	$\theta_{1max}$

$R = \infty$ , mersul în linie dreaptă;

$a \rightarrow$  ampatament;

$e \rightarrow$  ecartament.

Stabilirea corelației unghi de virare - rază de virare se face ținând seama de faptul că centrul instantaneu de rotație al vehiculului (centrul de virare) se găsește la intersecția

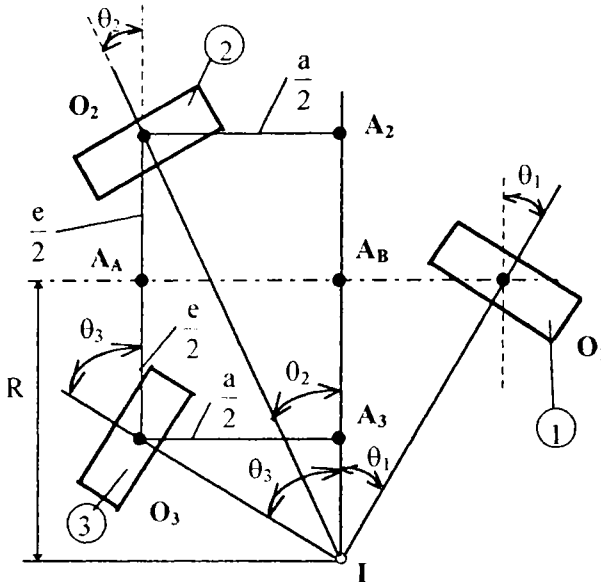
axelor roții directoare și a celor nedirectoare. Se alege  $R_{min}$  și se calculează  $\theta_{1max}$  (figura 5.6)

b). Roțile din spate directoare, roata din față directoare.

Se alege centrul instantaneu de rotație I pe dreapta mediatoare a segmentului  $A_1 O_1$ . În acest caz raza de virare va fi  $R = \overline{A_1 I}$ .

Din triunghiul  $\Delta O_1A_BI$ ,  $\Delta O_2A_2I$  și  $\Delta O_3A_3I$  rezultă :

$$\theta_3 = \arctg \frac{a}{R - \frac{e}{2}} \quad (5.2); \quad \theta_3 = \arctg \frac{a}{R + \frac{e}{2}} \quad (5.3); \quad \theta_1 = \arctg \frac{a}{R} \quad (5.4).$$



R	$\theta_1$	$\theta_2$	$\theta_3$
$\infty$	0	0	0
$R_{\min}$	$\theta_{1\min}$	$\theta_{2\min}$	$\theta_{3\min}$

$\theta_1 > 0$   
 $\theta_2 < 0$   
 $\theta_3 < 0$   
 când  $\theta_1$  crește  $\theta_2$ ,  
 $\theta_3$  se micșorează.

Figura 5.7.

### 5.2.3.2. Sinteza mecanismelor de virare prin metoda analitică.

a). Roțile din spate nedirectoare, roata din față directoare.  
 Nu este necesar mecanism de virare.

b). Roțile din spate directoare, roata din față directoare.

Se vor folosi două mecanisme patrulater articulat. Primul va corela mișcările de virare ale roților ① și ②, cel de-al doilea mișcările de virare ale roților ② și ③, în conformitate cu relațiile (5.3) și (5.4), respectiv (5.2) și (5.3).

**Mecanismul care corelează mișcările roților ① și ②.**

$$a = -l_{0,c_{12}} \cos \beta_2 - l_{0,d_{12}} \cos \beta_1 + l_{c_{12},d_{12}} \cos \varphi_0 \quad (5.5)$$

$$\frac{e}{2} = +l_{0,c_{12}} \sin \beta_2 + l_{0,d_{12}} \sin \beta_1 - l_{c_{12},d_{12}} \sin \varphi_0 \quad (5.6)$$

$$a = -l_{0,c_{12}} \cos(\beta_2 + \theta_{2i}) - l_{0,d_{12}} \cos(\beta_1 + \theta_{1i}) + l_{c_{12},d_{12}} \cos(\varphi_{2i}) \quad (i = 1 \dots n) \quad (5.7)$$

$$\frac{e}{2} = +l_{0,c_{12}} \sin(\beta_2 + \theta_{2i}) + l_{0,d_{12}} \sin(\beta_1 + \theta_{1i}) - l_{c_{12},d_{12}} \sin(\varphi_{2i}) \quad (i = 1 \dots n) \quad (5.8)$$

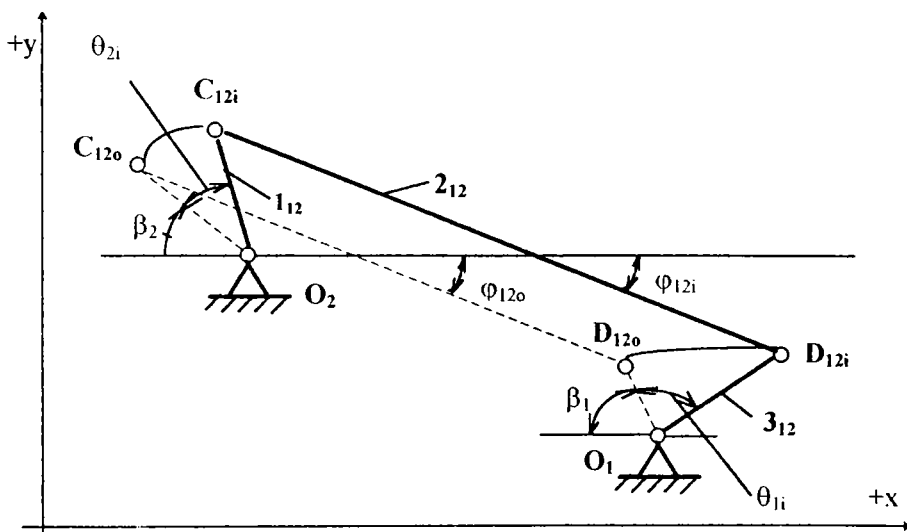


Figura 5.8.

Se formează sistemul de ecuații (5.5), (5.6), (5.7), (5.8). Necunoscutele sunt  $l_{0,c_{12}}$ ,  $l_{0,d_{12}}$ ,  $l_{c_{12},d_{12}}$ ,  $\beta_1$ ,  $\beta_2$ ,  $\varphi_{120}$  și  $\varphi_{12i}$ . Se alege, spre exemplu,  $\beta_1$  (unghiul de montaj al manivelei  $O_1D_{12}$ ) constructiv, ca și trei perechi de valori  $\theta_{1i} - \theta_{2i}$ , în conformitate cu paragraful I.B. (spre exemplu  $\theta_1=0$ ,  $\theta_2=0$ ,  $\theta_{1min}-\theta_{2min}$  și o pereche de valori intermediare între  $(\theta_1=0) - (\theta_2=0)$  și  $\theta_{1max}-\theta_{2max}$ . Se formează un sistem de 8 ecuații tip (5.5), (5.6), (5.7), (5.8), se rezolvă în raport cu necunoscutele  $l_{0,c_{12}}$ ,  $l_{0,d_{12}}$ ,  $l_{c_{12},d_{12}}$  (lungimile elementelor),  $\beta_2$  (unghiul de montaj al manivelei  $O_2C_{12}$ ) și  $\varphi_{120}$ ,  $\varphi_{12i}$  (valorile pentru cele trei poziții ale mecanismului).

### Mecanismul care corelează mișcările roților ② și ③.

$$a = l_{0,C_{23}} \sin \gamma_2 - l_{0,D_{23}} \sin \gamma_3 + l_{C_{23},D_{23}} \cos \varphi_{23} \quad (5.9)$$

$$o = -l_{0,C_{23}} \cos \gamma_2 + l_{0,D_{23}} \cos \gamma_3 - l_{C_{23},D_{23}} \sin \varphi_{23} \quad (5.10)$$

$$a = l_{0,C_{23}} \sin(\gamma_2 + \theta_{2i}) - l_{0,D_{23}} \sin(\gamma_3 + \theta_{3i}) + l_{C_{23},D_{23}} \cos \varphi_{23i} \quad (i = 1 \dots n) \quad (5.11)$$

$$o = -l_{0,C_{23}} \cos(\gamma_2 + \theta_{2i}) - l_{0,D_{23}} \cos(\gamma_3 + \theta_{3i}) + l_{C_{23},D_{23}} \sin \varphi_{23i} \quad (i = 1 \dots n) \quad (5.12)$$

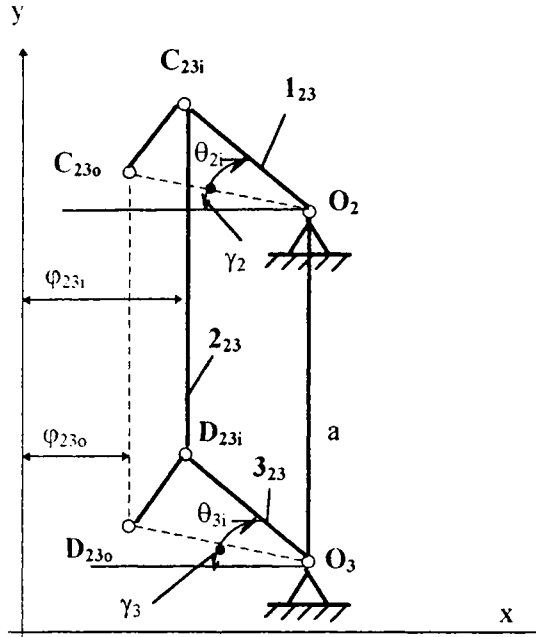


Figura 5.9.

Se formează sistemul (5.9), (5.10), (5.11), (5.12).

Necunoscutele sunt  $l_{0,C_{23}}$ ,  $l_{0,D_{23}}$ ,  $l_{0,D_{23}}$ ,  $\gamma_2$ ,  $\gamma_3$ ,  $\varphi_{23i}$ ,  $\varphi_{23o}$ .

Se alege constructiv spre exemplu  $\gamma_2$  (unghiul de montaj al manivelei  $O_2C_{23}$ ) și trei perechi de valori  $\theta_{2i} - \theta_{3i}$  conform paragrafelor I<sub>B</sub> și II<sub>B2</sub>. Se formează 8 ecuații de tip (5.9), (5.10), (5.11), (5.12) care se rezolvă în raport cu necunoscutele  $l_{0,C_{23}}$ ,  $l_{0,D_{23}}$ ,  $l_{0,D_{23}}$  (lungimile elementelor),  $\gamma_2$ , unghiul de montaj al manivelei  $O_3D_{23}$ ,  $\varphi_{23}$  și trei valori  $\varphi_{23i}$  ( $i = 1, 2, 3$ ) pentru cele trei poziții alese ale mecanismului.

### 5.2.3.3. Observații.

Sinteza de mai sus asigură faptul că vehiculul virează într-un sens. Pentru ca vehiculul să vireze în ambele sensuri, perechile de valori  $\theta_{1i} - \theta_{2i}$ ,  $\theta_{2i} - \theta_{3i}$  se vor lua pentru poziții ale lui I în ambele sensuri de virare a vehiculului.

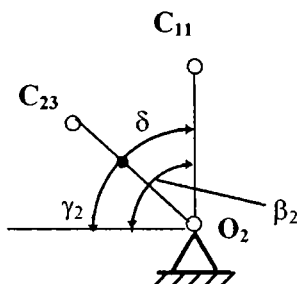


Figura 5.10.

Manivelele  $O_2C_{12}$  și  $O_2C_{23}$  se vor monta solidarizat, decalate cu unghiul  $\delta = |\beta_2 - \gamma_2|$

### 5.2.3.4. Rezolvarea grafică.

Sinteza dimensională grafică a celor două mecanisme patrulare se realizează prin metoda **pozițiilor asociate**.

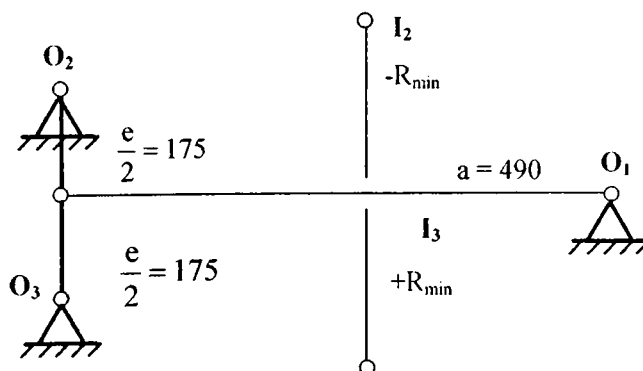


Figura 5.11.

Se iau în considerare următoarele poziții ale mecanismelor (R, raza de virare):

1 - mers în linie dreaptă,  $R_1 = \infty$ ,  $\theta_{1i} = \theta_{2i} = \theta_{3i} = 0$

2 - viraj maxim dreapta,  $R_2 = -R_{\min} = -\frac{a}{2} = -245$ ,  $\theta_{1_2} = -45^\circ$ ,

$$\theta_{2_2} = +30^\circ 15', \theta_{3_2} = +74^\circ 20'.$$

3 - viraj maxim stânga,  $R_3 = +R_{\min} = +\frac{a}{2} = +245$ ,  $\theta_{1_3} = +45^\circ$ ,

$$\theta_{2_3} = -30^\circ 15', \theta_{3_3} = -74^\circ 20'.$$

$$\theta_{2_2} = +\operatorname{arctg} \frac{|R_2| + \frac{e}{2}}{\frac{a}{2}} = +\operatorname{arctg} \frac{245 + 175}{245} = +30^\circ 15'$$

$$\theta_{3_2} = -\operatorname{arctg} \frac{R_2 + \frac{e}{2}}{\frac{a}{2}} = -\operatorname{arctg} \frac{245 + 175}{245} = -30^\circ 15'$$

$$\theta_{3_3} = +\operatorname{arctg} \frac{R_3 + \frac{e}{2}}{\frac{a}{2}} = +\operatorname{arctg} \frac{245 - 175}{245} = +74^\circ 20'$$

$$\theta_{2_3} = -\operatorname{arctg} \frac{R_3 - \frac{e}{2}}{\frac{a}{2}} = -\operatorname{arctg} \frac{245 - 175}{245} = -74^\circ 20'$$

Se aleg cele trei poziții ale manivelor  $O_1D_{12}$  și  $O_3D_{23}$ , decalate conform unghiurilor calculate mai sus.

Se aleg arbitrar direcțiile  $O_2\alpha_1$ ,  $O_2\alpha_2$ ,  $O_2\alpha_3$ , decalate între ele cu  $\theta_2$ , respectiv  $\theta_3$  (figura 5.7).

Se construiesc triunghiurile

$$\Delta O_2\alpha_1 D'_{123} = \Delta O_2\alpha_3 D_{123}$$

$$\Delta O_2\alpha_1 D'_{122} = \Delta O_2\alpha_2 D_{122}$$

Pe această cale s-au determinat punctele  $D'_{12}$  și  $D'_{12}$ . Se ridică mediatoarele segmentelor  $\overline{D'_{12} D'_{12}}$  și  $\overline{D'_{12} D'_{12}}$ . La intersecția lor se află punctul  $C_{12}$ .

Se construiesc triunghiurile:

$$\Delta O_2 \alpha_1 D'_{23} = \Delta O_2 \alpha_3 D_{23},$$

$$\Delta O_2 \alpha_1 D'_{232} = \Delta O_2 \alpha_2 D_{232}$$

Pe această cale s-au determinat punctele  $D'_{23}$  și  $D'_{23}$ . Se ridică mediatoarele segmentelor  $\overline{D'_{23} D'_{23}}$  și  $\overline{D'_{23} D'_{23}}$ . La intersecția lor se găsește punctul  $C_{23}$ .

Punctele  $C_{12}$ ,  $C_{23}$  și  $O_2$  formează un element rigid.  $O_1$ ,  $O_2$ ,  $O_3$  sunt axele furcilor roților directe. Mecanismul rezultat este prezentat în anexa 3.

### 5.3. Elemente mecanice de protecție.

După cum s-a arătat în capitolul 1, protecția la coliziune a vehiculelor ghidate automat și a celorlalți participanți la trafic se poate realiza pe două căi:

- cu contact mecanic. Se utilizează dispozitive care se deformează mecanic;
- fără contact mecanic. Se utilizează senzori ultrasonici, senzori optici sau camere video.

Din prima categorie, cele mai des întâlnite sunt tampoanele mecanice, realizate din materiale ușor deformabile (burete), sau prevăzute cu sisteme elastice care se pot deforma la atingerea unui obstacol (figura 5.12.).

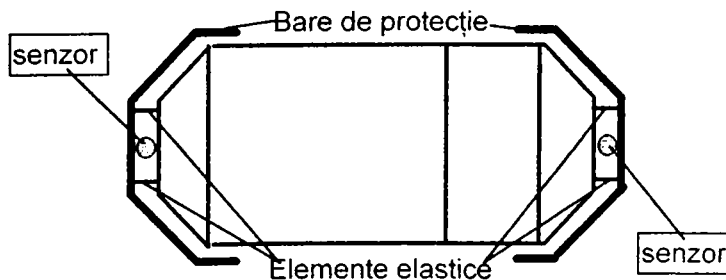


Figura 5.12.



De obicei, protejarea împotriva coliziunii a obiectelor cu care intră în contact vehiculul și a vehiculului în sine nu se bazează numai pe deformarea mecanică a barelor de protecție. În interiorul acestora se montează senzori care au sensibilitatea ridicată. Cu ajutorul acestora, la cea mai ușoară atingere a barelor, se sesizează deformarea lor și este furnizat un semnal electric sistemului de conducere.

Efectul sesizării contactului mecanic al robocarului cu un obiect aflat pe traiectoria sa este frânarea motorului de tracțiune. Modul de protecție împotriva coliziunii descris mai sus este o soluție extremă, utilizată numai atunci când sistemul senzorial evoluat al robocarului este avariat.

## 5.4. Concluzii.

Din partea introductivă a capitolului 5 a rezultat importanța pe care autorul o acordă construcției sistemului mecanic al robocarului, fără ca aceasta să constituie un obiectiv rezultat direct din tematica abordată. Dacă unul dintre obiectivele tezei ar fi fost verificarea prin simulare a rezolvărilor teoretice ale unor probleme, această construcție nu ar fi fost necesară. **Deoarece autorul și-a propus de la început experimentarea atât a blocurilor electronice și a programelor realizate, cât și a soluțiilor originale propuse, construirea robocarului a fost strict necesară.** Alegerea soluțiilor constructive a fost impusă în primul rând de limitările de ordin material existente. Cu toate acestea s-a realizat o variantă care oferă condiții foarte bune desfășurării experimentelor.

Structura șasiului nu a fost aleasă din motive estetice sau pentru îndeplinirea unor anumite sarcini de producție. Ea a fost aleasă din necesitatea de a oferi utilizatorului posibilitatea montării simple a subansamblurilor electronice, a senzorilor și a traductoarelor și pentru a face posibil un acces rapid la fiecare echipament.

Sintetizând problemele referitoare la sistemul de rulare al **VGA**, prezentate pe parcursul tezei și abordate de către numeroși autori [HOL 87], [HAM 87], se obțin trei variante constructive ale acestui sistem. Aceste variante pun în evidență poziția roților motoare și modul de realizare a sistemului de virare [NIT 97]. Ele sunt următoarele:

- a) vehicul cu direcție de tip osie - frântă (figura 5.13 a);
- b) vehicul cu roți diferențiale (figura 5.13 b);
- c) vehicul cu roată motrico-directoare (figura 5.13 c).



În concluzie, a fost realizat un vehicul cu roată motrico-directoare (cu trei roți) și mecanism de virare extins asupra roților posterioare.

Considerentele care au stat la baza alegerii acestei variante sunt:

- este varianta cea mai simplă, atât din punct de vedere mecanic, cât și al acționării;
- permite o îmbunătățire simplă, prin utilizarea mecanismului prezentat în paragraful 5.2.3;
- o singură roată de tracțiune necesită un singur motor, consumul de energie electrică este redus, iar echipamentul de comanda este relativ simplu;
- o roată de virare activă presupune un singur motor, deci consum de energie redus și un echipament de comandă de complexitate mai scăzută.

Componentele utilizate sunt următoarele:

- mecanismul de tracțiune este echipat cu un motor de curent continuu cu rotor disc. Deoarece acesta are randament ridicat, consumul de energie electrică este redus, la o putere activă ridicată. Totodată greutatea și gabaritul sunt mici, cerințe necesare pentru a nu mări inutil greutatea și dimensiunile vehiculului.
- reductorul motorului de c.c., având raportul de transmitere 1:34,6, asigură cuplul necesar deplasării robocarului și totodată menținerea sa frânată în stare de repaus. Volumul său este mic, greutatea redusă și poate fi montat simplu;
- mecanismul de virare este echipat cu un motor pas cu pas, care are consumul de energie electrică mic, iar prin menținerea mecanismului de virare blocat cu ajutorul reductorului, chiar în lipsa alimentării se asigură reducerea acestui consum. De asemenea gabaritul și volumul întregului ansamblu sunt reduse.
- reductorul motorului pas cu pas, având raportul de transmitere 1:125, asigură cuplul necesar rotirii mecanismului de virare, iar în repaus este posibilă menținerea sa în poziția dorită, fără alimentarea motorului pas cu pas.

Soluțiile adoptate au concurat la finalizarea construcției unui vehicul care are performanțe mecanice și electrice ridicate și anume:

- greutate redusă;
- posibilități dinamice foarte bune;
- consum redus de energie electrică;

- 
- randament ridicat;
  - simplitate constructivă;
  - posibilitatea realizării unor circuite electronice de alimentare simple și fiabile.

Aceste caracteristici fac ca robocarul să se încadreze în standardele internaționale și să poată fi considerat ca o realizare tehnică apreciabilă.

Mecanismul de virare a fost ales atât din considerentele prezentate mai sus, cât și din necesitatea de a se realiza raze de virare cât mai reduse, datorită spațiului de lucru restrâns din Laboratorul de Roboți Industriali. A rezultat un mecanism complex, necesar din punct de vedere practic, a cărui originalitate constă în alegerea, conceperea, proiectarea și realizarea efectivă.

Mecanismul de tracțiune a fost și el ales, pe lângă celelalte considerente și din necesități practice. Robocarul trebuie să facă față unor sarcini efective de transport, fiind necesar un cuplu care să permită transportarea unor semifabricate în laborator. Alegerea sa, precum și construirea în întregime au fost sarcina autorului.

**Se poate deci afirma că robocarul construit prezintă un grad ridicat de originalitate datorită soluțiilor adoptate, dar și al modului de realizare a sa.**



## Capitolul 6

# CONCEPȚIA ȘI REALIZAREA SISTEMULUI DE ACȚIONARE ȘI DE CONDUCERE AL ROBOCARULUI

În capitolul 5 a fost prezentată varianta constructivă a robocarului realizat de către autor. De asemenea s-a justificat importanța construirii sistemului mecanic al unui vehicul ghidat automat.

Cel puțin la fel de necesară pentru desfășurarea experimentelor ca și construcția mecanică, este realizarea unui sistem de acționare și de conducere al vehiculului. Sistemele de acționare utilizate în robotică sunt dintre cele mai diverse. Alegerea acestora se bazează atât pe criteriile economice, cât și pe criteriile tehnice.

Criteriile economice se referă în primul rând la costul echipamentelor utilizate.

Criteriile de ordin tehnic sunt multiple și vor fi enumerate în continuare:

- fiabilitatea ridicată a motoarelor și a elementelor de execuție;
- consum energetic cât mai redus;
- funcționare sigură în condițiile mediului industrial;
- caracteristici dinamice foarte bune;
- gabarit redus;
- structură modulară și cât mai compactă;
- posibilitatea înlocuirii cu alte module care îndeplinesc aceleași funcții, fără modificarea echipamentului electric al robocarului.

**Aceste criterii au stat la baza opțiunii autorului pentru varianta constructivă realizată.** Desigur că există și alte soluții care satisfac aceste cerințe, dar limitările de ordin material au dus la adoptarea unei soluții, care va fi prezentată în acest capitol, fără ca ea să fie în contradicție cu vreunul dintre criteriile enunțate.

În ceea ce privește traductoarele și sistemul senzorial utilizat, este necesar ca acestea să îndeplinească următoarele cerințe:

- informațiile prelevate cu ajutorul lor să nu fie eronate datorită perturbațiilor existente în mediul de lucru industrial;
- procesarea acestor informații să fie cât mai simplă și mai rapidă;

- să fie posibilă utilizarea aceluiași senzori pentru scopuri diverse (detectarea obstacolelor și determinarea situației);
- montarea acestora să fie relativ simplă.

Traductoarele utilizate de către autor și caracteristicile lor au fost prezentate în capitolul 5.

Sistemul senzorial al robocarului construit a fost adoptat datorită performanțelor pe care le oferă. S-a preferat camera video CCD, deoarece aceasta poate fi utilizată atât pentru determinarea situației, cât și pentru detectarea obstacolelor. În plus, nu este necesar un sistem de achiziție a imaginii special construit pentru efectuarea experimentelor. Se poate utiliza cu rezultate foarte bune o placă de achiziție uzuală. În cazul în care robocarul devine o entitate de sine stătătoare, este necesară construirea unui sistem pentru achiziția imaginilor. Pentru detectarea obstacolelor s-au utilizat senzori ultrasonici, deoarece s-a preferat protejarea împotriva coliziunilor **fără atingere**.

Sistemul de conducere trebuie să dispună de cât mai multe resurse, pentru a putea îndeplini funcțiile prezentate în capitolul 2. Varianta constructivă a sistemului de conducere utilizat de către autor este prezentată în acest capitol. Pentru experimentări, unele funcții pe care trebuie să le îndeplinească acest sistem au fost preluate de către calculatorul PC utilizat (sistemul de conducere ierarhic superior).

**În continuare vor fi prezentate circuitele electronice realizate și programele utilizate, concepute în totalitate de către autor.**

## **6.1. Sistemul de acționare.**

În acest paragraf se vor prezenta elementele de execuție, prin intermediul cărora se realizează alimentarea motoarelor care fac parte din sistemul de acționare al robocarului. Rotirea mecanismului de virare se realizează cu ajutorul unui motor pas cu pas. Pentru realizarea deplasării se utilizează un motor de curent continuu, iar pentru baleierea întregului spațiu de lucru cu ajutorul camerelor video se utilizează două motoare pas cu pas. Fiecare motor este prevăzut cu câte un circuit propriu pentru alimentarea cu energie electrică. Aceste circuite sunt comandate cu ajutorul unor semnale generate de către sistemul de conducere, pe baza unor programe concepute în acest scop. Informațiile prelevate cu ajutorul sistemului senzorial sunt procesate cu ajutorul unor circuite electronice adecvate. Circuitele electronice utilizate pentru

alimentarea motoarelor, împreună cu circuitele folosite pentru procesarea semnalelor furnizate de către sistemul senzorial și circuitele de protecție aferente motoarelor au fost grupate într-un singur bloc, sub denumirea de **elemente de execuție**. Pentru simplitatea exprimării se vor folosi următoarele abrevieri:

**EEMCC** - element de execuție pentru motorul de curent continuu;

**EEMPP** - element de execuție pentru motorul pas cu pas.

Toate aceste echipamente vor fi prezentate în continuare.

### 6.1.1. Sistemul de acționare a mecanismului de tracțiune.

Pentru antrenarea roții active s-a utilizat, așa cum s-a mai arătat, un motor de curent continuu cu rotor disc. Acest motor are gabaritul redus și un randament ridicat. El asigură un cuplu satisfăcător pentru deplasarea robocarului încărcat cu o sarcină utilă de 150kg. Motorul se alimentează de la bateriile de acumuloare care constituie sursa de energie electrică a vehiculului. Curentul absorbit în sarcină de către motor este 1,5A, iar tensiunea sa de alimentare maximă este 24V.

Cerințele care se impun elementului de execuție al motorului de curent continuu (**EEMCC**) sunt următoarele:

- să permită schimbarea sensului de rotație al motorului;
- să permită rotirea motorului cu turație variabilă în ambele sensuri;
- să accepte semnalele de comandă furnizate de către sistemul de conducere (semnale logice);
- să fie prevăzut cu un circuit de protecție la suprasarcină;
- să realizeze separarea galvanică între circuitul de putere, inclus în **EEMCC** și sistemul de conducere;
- consumul de energie electrică să fie redus;
- să permită frânarea și menținerea în stare frânată a motorului.

Aceste cerințe au fost luate în considerare în proiectarea elementului de execuție.

#### 6.1.1.1. Schema bloc și schema de principiu a EEMCC.

Alimentarea motorului de curent continuu se realizează de la bateria de acumuloare (care echează robocarul) cu ajutorul unor circuite electronice comandate cu semnalele furnizate de către sistemul de



conducere. Aceste circuite fac parte din elementul de execuție **EEMCC**, a cărui schemă bloc este prezentată în figura 6.1.

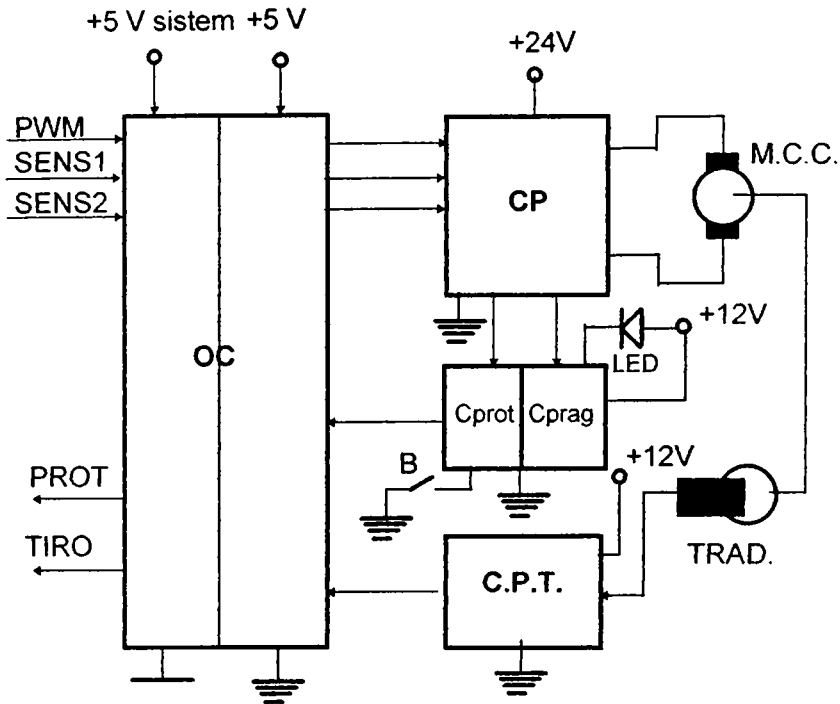


Figura 6.1.

Blocurile care compun elementul de execuție vor fi descrise în cele ce urmează.

Circuitul de putere (**CP**) este realizat cu circuitul integrat specializat L298, produs de firma SGS Thompson [SGS 94]. Acesta este comandat, prin intermediul unor optocuploare, de către sistemul de conducere. Cu ajutorul semnalelor de comandă se prescrie sensul de rotație al motorului (semnalele **SENS1** și **SENS2**) și viteza unghiulară (semnalul **PWM**).

Traductoarele de curent furnizează o tensiune proporțională cu valoarea curentului care parcurge înfășurările motorului. Această tensiune se aplică unui comparator cu prag reglabil (**Cprag**). Dacă a fost depășită limita maximă admisă admisă a curentului, comparatorul va sesiza acest fapt, iar semnalul furnizat la ieșirea sa va declanșa circuitul de protecție (**Cprot**), care la rândul său blochează alimentarea motorului. În același

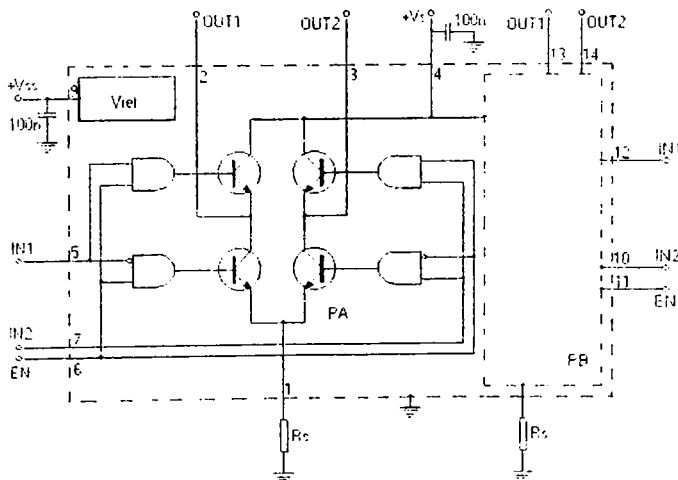
timp este informat sistemul de conducere al vehiculului despre faptul că în cadrul sistemului de acționare al mecanismului de tracțiune a avut loc o avarie.

Semnalul sinusoidal furnizat de către traductorul inductiv de turație este amplificat și limitat cu ajutorul unui circuit de procesare (CPT), pentru a se obține impulsuri dreptunghiulare. Aceste impulsuri constituie un semnal care poate fi folosit de către sistemul de conducere drept semnal de intrare.

Toate semnalele de comandă se transmit prin blocul cu optocuploare (OC), pentru a se asigura separarea galvanică între circuitul de putere și circuitul de comandă.

Schema electronică detaliată a EEMCC nu va fi prezentată în acest capitol. Ea se află în anexa 4. În cele ce urmează va fi evidențiată numai soluția originală propusă pentru utilizarea circuitului L298, **imaginată de autor**.

Circuitul integrat L298 conține două punți realizate cu tranzistoare bipolare (figura 6.2).



**Figura 6.2.**

El este destinat alimentării bipolare a înfășurărilor motoarelor pas cu pas sau a motoarelor de curent continuu în regim start - stop, fără posibilitatea modificării turației prin modulație în durată a impulsurilor (PWM). Modul în care s-ar putea realiza modificarea turației motorului ar

fi alimentarea circuitului cu tensiune variabilă. Randamentul acționării, prin utilizarea unei astfel de proceduri fiind scăzut, ceea ce nu este convenabil în cazul de față, s-a căutat un mod de rezolvare a acestei probleme, care va fi prezentat în continuare.

Sensul de rotație al motorului se stabilește cu ajutorul semnalelor aplicate pe cele două intrări **IN1**, **IN2**, iar blocarea tuturor tranzistoarelor se face prin semnalul aplicat pe intrarea **EN**. Din tabelul de adevăr prezentat în figura 6.3. se observă că, dacă impulsurile modulate în durată s-ar aplica pe intrările **IN1** sau **IN2**, ar fi posibilă modificarea turației, iar frânarea ar putea fi făcută acționând asupra intrării **EN**.

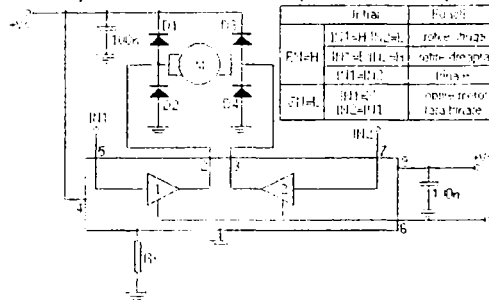


Figura 6.3.

În anumite condiții însă (**EN=H, IN1=IN2**) motorul se va afla în regim de frână, ceea ce duce la reducerea puterii și creșterea consumului (în funcționare normală). Pentru evitarea acestei situații s-a ales o altă variantă și anume, aplicarea unor niveluri logice pe intrările **IN1**, **IN2** pentru stabilirea sensului de rotație, și a impulsurilor modulate în durată pe intrarea **EN** pentru modificarea turației (figura 6.4)

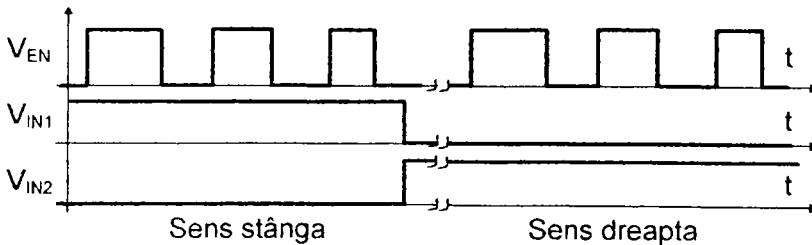


Figura 6.4.

Frânarea are loc atunci când sunt îndeplinite condițiile **EN = H, IN1 = IN2**, ceea ce se realizează numai pentru oprirea vehiculului. Menținând în continuare stările **EN=H, IN1=IN2**, se ține rotorul în scurtcircuit, ceea ce face ca robocarul să fie blocat în poziția de repaus.

Utilizând circuitul în acest mod, se poate modifica turația în intervalul  $0 \div n_{\max}$ , iar motorul funcționează fără reducerea puterii nominale. Dacă operațiile pe care le efectuează robocarul necesită o menținere strictă a acestuia în poziția de lucru, atunci este necesară o frânare electromagnetică sau o indexare mecanică.

În concluzie, semnalele de comandă necesare **EEMCC** sunt:

- două semnale logice utilizate pentru prescrierea sensului de rotație al motorului. Acestea sunt aplicate pe intrările **IN1** și **IN2** în antifază (**SENS1** și **SENS2**);
- impulsuri modulate în durată (**PWM**), cu ajutorul cărora se prescrie viteza de rotație a motorului, aplicate pe intrarea **EN**.

La ieșirea **TIRO** sunt furnizate impulsuri obținute prin procesarea semnalului sinusoidal generat de către traductorul inductiv, a căror frecvență de repetiție este proporțională cu turația motorului. Numărul de impulsuri generate la o rotație este 50 imp/rot.

La ieșirea **PROT** se obține un nivel logic ridicat dacă a avut loc o depășire a curentului maxim admis prin înfășurările motorului. Acest semnal blochează punțile cu tranzistoare ale circuitului integrat L298 și are loc întreruperea alimentării motorului. Același semnal este transmis, prin intermediul optocuploarelor, sistemului de conducere, pentru a fi posibilă localizarea defecțiunilor survenite în sistemul de acționare.

### 6.1.1.2. Generarea prin soft a semnalelor de comandă .

Deoarece sistemul de conducere generează semnalele de comandă necesare elementelor de execuție **EEMCC** și **EEMPP**, anumite noțiuni prezentate în capitolul 3 se vor relua sumar, pentru o mai bună înțelegere a modului de generare a acestor semnale.

În continuare va fi prezentat programul utilizat pentru generarea de către sistemul de conducere al vehiculului a semnalelor de comandă necesare **EEMCC**.

Sistemul de conducere generează aceste semnale de comandă pe baza semnalului de reacție **TIRO** furnizat la ieșirea cu același nume a **EEMCC**. Semnalele de comandă sunt:

- impulsuri modulate în durată (semnalul **PWM**). Acest semnal este necesar pentru modificarea turației motorului și este furnizat la ieșirea **PWM0** a microcontrolerului 80C552 (pinul 4);

- două semnale în antifază (semnalele **SENS1**, respectiv **SENS2**), utilizate pentru schimbarea sensului de rotație al motorului. Aceste semnale sunt furnizate de către microcontroler (P1.0, respectiv P1.1).

Formele de undă ale semnalelor de comandă sunt cele prezentate în figura 6.4.

Nivelurile logice ale semnalelor **SENS1**, respectiv **SENS2** sunt cele menționate în tabelul 6.1.

Tabelul 6.1.

P1.0	P1.1	Sens deplasare
1	0	înainte
0	1	înapoi

Semnalele de mai sus reprezintă semnale de ieșire ale sistemului de dezvoltare realizat cu microcontrolerul 80C552.

După cum s-a arătat, motorul de curent continuu se rotește pe baza semnalelor de comandă generate de către sistemul de conducere în concordanță cu semnalul **TIRO**. Acesta este semnal de intrare în sistemul de dezvoltare cu microcontroler pe linia INT0 (pinul 26) și generează întreruperi în microcontroler. Întreruperile sunt deservite de către acesta prin executarea subrutinei: **MOTOR-CC**.

Se menționează că toate comenzile transmise de la calculator la sistemul de conducere sunt afișate pe ecranul LCD pentru ca mișcarea care se efectuează să fie cunoscută de către operator.

Pentru a se evita solicitarea mecanismelor de transmisie în cazul schimbării sensului de rotire, este realizată o întârziere soft de 0,5 secunde între momentul aplicării comenzii de schimbare a sensului și momentul execuției acesteia. Atunci când robocarul este oprit, **MCC** este frânat prin poziționarea liniei **PWM0** pe "1" logic, iar liniile de port P1.0 și P1.1 vor fi poziționate pe "0" logic. Oprirea și frânarea se realizează cu ajutorul funcției **opritMCC()**.

```
opritMCC()
{
P1 = 0x00;
PWM0=0; //blochează MCC 1 pe validare
```

```
EX0=0;  
delay(700);  
PWM0=165;  
}
```

Instrucțiunea `P1 = 0x00` poziționează liniile portului P1 pe "0" logic (ceea ce înseamnă că și P1.0, P1.1 sunt poziționate pe "0" logic). Poziționarea liniei **PWM0** pe "0" logic realizează frânarea motorului de curent continuu.

Funcția `delay(700)` realizează o întârziere soft de 700 ms, după care este generat din nou un semnal pe linia **PWM0**.

Cu ajutorul funcției **start()**, sistemul de conducere generează semnalele de comandă ale **EEMCC**. Această funcție este prezentată în continuare.

```
if(sens)  
{  
    if(!înainte)  
        opritMCC();  
    înainte=1;  
    P1 &= 0xFD;  
    P1 |= 0x01;  
}  
else  
{  
    if(înainte)  
        opritMCC();  
    înainte=0;  
    P1 &= 0xFE;  
    P1 |= 0x02;  
}  
PWM0 = 165;  
EX0=1;
```

Nivelul logic al bitului **sens** stabilește sensul de rotație al motorului astfel:

- "0" logic - deplasarea înainte a robocarului;
- "1" logic - deplasarea înapoi a robocarului.

Sensul de deplasare precedent este memorat în bitul **înainte**. Acest bit suplimentar este necesar pentru oprirea robocarului la schimbarea

sensului de deplasare. Funcția **opritMCC()** este apelată numai dacă se primește o comandă de deplasare în sens opus.

Semnalele de comandă a **EEMCC** corespunzătoare deplasării înainte sunt generate pe baza instrucțiunilor:

**P1 &= 0xFD;**

**P1 |= 0x01;**

ceea ce înseamnă că linia P1.0 este poziționată pe "1" logic, iar linia P1.1 este poziționată pe "0" logic.

Semnalele de comandă a **EEMCC** corespunzătoare deplasării înapoi sunt generate pe baza instrucțiunilor:

**P1 &= 0xFE;**

**P1 |= 0x02;**

ceea ce înseamnă că linia P1.1 este poziționată pe "1" logic, iar linia P1.0 este poziționată pe "0" logic.

Semnalul de tact este generat pe linia **PWM0**, prin încărcarea registrului **PWM0** cu valoarea 165.

### 6.1.2. Sistemul de acționare a mecanismului de virare.

Acționarea mecanismului de virare se realizează prin utilizarea unui motor pas cu pas care are următorii parametri: cuplul nominal  $C = 0,25\text{Nm}$ , tensiunea de alimentare nominală 12V. Rezistența unei înfășurări este  $R_m = 30\Omega$ , iar inductivitatea acesteia are valoarea  $L_m = 7\text{mH}$ .

Cerințele impuse elementului de execuție al motorului pas cu pas (**EEMPP**) sunt următoarele:

- să permită schimbarea sensului de rotație al motorului;
- să permită rotirea motorului cu turație variabilă, prin modificarea frecvenței de tact;
- să accepte semnalele de comandă furnizate de către sistemul de conducere;
- să fie prevăzut cu circuite de protecție la suprasarcină;
- să realizeze separarea galvanică între circuitul de putere inclus în **EEMPP** și sistemul de conducere;
- consumul de energie electrică să fie redus;
- să permită blocarea motorului într-o anumită poziție (și implicit a mecanismului de virare), cu un consum de energie electrică redus;
- să permită inițializarea mișcării mecanismului de virare.

### 6.1.2.1. Schema bloc și schema de principiu a EEMPP.

Pentru circuitul de putere (CP) necesar alimentării motorului pas cu pas s-a utilizat același tip de circuit integrat, L298. Fiind o punte cu tranzistoare comandată cu semnale logice, cu ajutorul acestui circuit se pot alimenta bipolar relativ simplu motoare pas cu pas. Semnalele de comandă (patru la număr), corespunzătoare fiecărei faze, sunt generate de către un distribuitor (translator) de impulsuri realizat cu circuitul integrat L297. Funcționarea are loc în secvență dublă, iar comanda **EEMPP** se face cu ajutorul semnalelor **TACT** și **SENS**, furnizate de către sistemul de conducere. Schema bloc a **EEMPP** este prezentată în figura 6.5, iar schema detaliată se află în anexa 5.

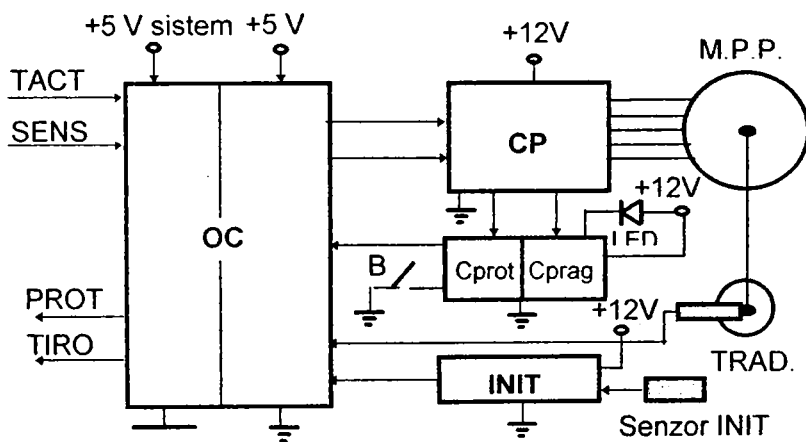


Figura 6.5.

Elementul de execuție are în componența sa un circuit de protecție (**Cprot-Cprag**) care, în același mod cu acela al **EEMCC**, sesizează depășirea limitei maxime admise a curentului care parcurge înfășurările motorului pas cu pas, blocând alimentarea acestuia. Totodată este informat sistemul de conducere despre avaria apărută în cadrul sistemului de acționare al mecanismului de virare. **EEMPP** cuprinde și circuitul electronic necesar procesării semnalului furnizat de către traductorul optic **INIT** (blocul **INIT**). Acest traductor este utilizat pentru sesizarea poziției în care are loc inițializarea sistemului de virare. Semnalul furnizat de către **EEMPP** la ieșirea **INIT** este interpretat de către sistemul de conducere în cadrul procedurii de inițializare, care a fost descrisă în paragraful 3.2.1.



Pentru realizarea circuitului de putere (CP) din cadrul **EEMPP** s-au utilizat componente uzuale (L297, L298). Circuitul integrat L298 este destinat alimentării bipolare a motoarelor pas cu pas, dar poate fi folosit și pentru alimentarea unipolară a acestora.

Deoarece motorul trebuie alimentat unipolar și s-a impus utilizarea acelorași circuite integrate pentru toate elementele de execuție, **se propune de către autor o modificare a schemei de aplicație existente în catalog** [SGS 94].

Pornind de la observația că tranzistoarele de putere ale circuitului integrat L298 pot fi alimentate de la o sursă de tensiune diferită de sursa de alimentare a circuitului de comandă, s-a renunțat la alimentarea tranzistoarelor de putere prin pinul 4. În această situație, alimentarea înfășurării motorului se realizează prin punctul median, accesibil la bornele sale, iar tranzistoarele din partea superioară a punții nu sunt utilizate (figura 3.6).

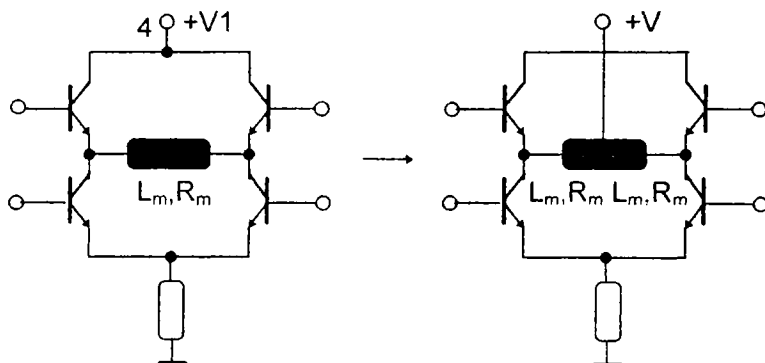


Figura 6.6.

Sistemul de conducere generează două semnale, **TACT** și **SENS**, care se aplică distribuitorului de impulsuri realizat cu L297. Acesta generează, la rândul său, semnalele de comandă în **secvență dublă**, care se vor aplica intrărilor **IN1**, respectiv **IN2** ale celor patru circuite de putere corespunzătoare celor patru faze. Semnalele de comandă se aplică pe intrările **IN1**, respectiv **IN2** pentru obținerea sensului de rotire dorit.

**În concluzie se poate spune că, prin utilizarea circuitului integrat L297, numărul de semnale de comandă generate de către microcontroler s-a redus de la patru la două.**

### 6.1.2.2. Generarea prin soft a semnalelor de comandă .

Motorul utilizat pentru rotirea sistemului de virare este un motor pas cu pas. Semnalele de comandă aplicate **EEMPP** sunt:

- un semnal de tact, generat de către sistemul de comandă pe linia **PWM1** (linia 5 a microcontrolerului 80C552);
- un semnal pentru impunerea sensului de rotație, generat pe linia de port P1.2 (pinul 18 al microcontrolerului 80C552).

La fiecare impuls de tact generat pe linia de comandă **PWM1**, motorul pas cu pas se rotește cu un pas în sensul stabilit de către semnalul generat pe linia de port P1.2. Astfel, dacă P1.2 este "0" logic, atunci virarea are loc spre dreapta, iar dacă P1.2 este "1" logic, atunci virarea are loc spre stânga.

Deși nu era necesară utilizarea unui traductor de unghi pentru motorul pas cu pas, pentru obținerea unei poziționări precise, în cazul în care se pierde pași, se utilizează totuși un **TIRO**.

Secvența de program cu ajutorul căreia se stabilesc limitele pentru rotirea spre stânga și spre dreapta a mecanismului de virare este prezentată în continuare:

```
pasi=2;
if(sens & !limd)
{
  dreapta=1;
  fan=1;
  limleft=0;
  gata=0;
  P1 |= 0x04;
  PWM1=125;
  EX1=1;
}
if(!sens & !limleft)
{
  dreapta=0;
  limd=0;
  fan=1;
  gata=0;
  P1 &= 0xFB;
  PWM1=125;
}
```

```

EX1=1;
}

```

Dacă comanda primită de către **EEMPP** este de virare la dreapta și nu a fost atinsă limita soft dreapta:

***if(sens & !limd)***

atunci este anulat bitul corespunzător limitării la stânga a rotirii **limleft** și este poziționată linia P1.2 pe "1" logic. Registrul **PWM1** este încărcat cu valoarea 125. Variabila **pași** memorează numărul de pași care vor fi efectuați de către motorul pas cu pas. Unei rotiri a mecanismului de virare cu un grad îi corespund 2,38 pași. Apoi este validată întreruperea **INT1** prin setarea bitului **EX1** pe "1" logic. Semnalul furnizat de către **TIRO** este semnal de intrare în sistem pe linia de întrerupere **INT1** (pinul 27 al microcontrolerului 80C552). Din acest moment motorul pas cu pas se rotește spre dreapta și după efectuarea numărului de pași prescriși (pași care sunt numărați în interiorul subrutinei de întrerupere **INT1**), registrul **PWM1** se va încărca cu valoarea maximă 255, ceea ce are drept consecință poziționarea liniei pe care se generează semnalul de tact al **EEMPP** pe "0" logic.

Dacă comanda primită de către **EEMPP** este de virare la stânga și nu a fost atinsă limita soft stânga:

***if(!sens & !limleft)***

atunci este anulat bitul corespunzător limitării la dreapta a rotirii **limd** și este poziționată linia de port P1.2 pe "0" logic. În continuare funcționarea decurge ca și în cazul rotirii spre dreapta. În funcție de nivelurile logice ale biților **limd** și **limleft** are loc limitarea unghiului de virare al robocarului la  $\pm 60^\circ$  față de poziția  $0^\circ$ , stabilită la inițializarea sistemului de virare.

### 6.1.3. Sistemul de orientare a camerei video.

În capitolul 4 au fost prezentate metodele propuse de către autor pentru determinarea situației. Pentru aplicarea metodei descrise în paragraful 4.2.9.3. este necesară o cameră video prevăzută cu un dispozitiv cu ajutorul căruia să fie posibilă modificarea azimutului și a elevației acesteia. Acest dispozitiv va fi numit pe scurt **dispozitiv de orientare**.

Pentru orientarea camerei video s-au utilizat două motoare pas cu pas care au cuplul nominal  $C = 0,1\text{Nm}$  și tensiunea de alimentare 12V. Înfășurările au rezistența  $R_m = 45\Omega$  și inductivitatea  $L_m = 20\text{mH}$ . Motoarele sunt alimentate cu tensiunea obținută de la același acumulator

de la care se obține tensiunea de alimentare a motorului pas cu pas al sistemului de virare. Orientarea camerei se realizează cu ajutorul unui motor în plan orizontal și cu ajutorul celuilalt motor în plan vertical. Elementele de execuție sunt identice. Sistemul este astfel conceput încât să permită și orietarea în câte un singur plan a două camere video.

Cerințele impuse elementelor de execuție sunt:

- să ofere posibilitatea schimbării sensului de rotație;
- să ofere posibilitatea modificării vitezei unghiulare a motorului pas cu pas;
- să accepte semnalele de comandă furnizate de către sistemul de conducere;
- să realizeze protecția la supracurent a elementului de execuție;
- să realizeze separarea galvanică între elementul de execuție și sistemul de comandă;
- consumul de energie electrică să fie redus;
- să permită inițializarea mișcării dispozitivului de orientare cu ajutorul semnalului furnizat de către un traductor optic.

### 6.1.3.1. Schema bloc și schema de principiu a elementului de execuție utilizat pentru orientarea camerei video.

În cele ce urmează se prezintă schema bloc a ansamblului format din cele două elemente de execuție (figura 6.7), iar descrierea blocurilor componente se va face pentru un singur element de execuție, deoarece acestea sunt identice. Schema detaliată va fi prezentată în anexa 6.

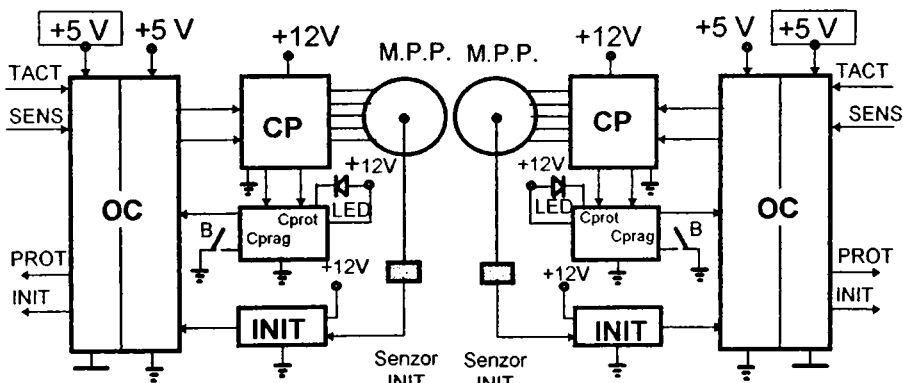


Figura 6.7.



Pinii 0-3 ai portului de ieșire sunt utilizați atât pentru furnizarea semnalului de tact **PWM** (**EPO.0** și respectiv **EPO.2**), cât și pentru furnizarea semnalului pentru stabilirea sensului de rotație (**EPO.1** și, respectiv **EPO.3**). Pinii 0 și 1 ai portului de intrare sunt utilizați pentru achiziția informației furnizate de către traductorul de poziție (**INIT**).

La conectarea tensiunii de alimentare sau după reset, executarea programului începe prin apelarea subrutinei utilizate pentru inițializarea mișcării pe cele două axe (**void inițializare()**).

Inițializarea celor două mișcări se face succesiv. Motorul 1 este avansat pas cu pas până în momentul obturării traductorului **INIT** corespunzător (cu element fotoemisiv-fotosenzitiv). După atingerea acestui punct, motorul este rotit cu  $198^\circ$  (110 pași), ceea ce corespunde aducerii în poziția extremă opusă. Același algoritm se aplică în continuare pentru motorul 2. Prin urmare, când are loc revenirea din subrutina de inițializare, ambele motoare se află în pozițiile extreme.

Din acest moment, programul oferă două modalități de funcționare:

- a) baleierea, cu un unghi de deschidere de  $180^\circ$ , cu ajutorul celor două motoare, a spațiului din fața robocarului;
- b) rotirea unui motor, respectiv a celor două motoare atunci când se utilizează o cameră, respectiv două camere, simultan cu mecanismul de virare.

În cazul (a) se intră într-o buclă infinită, în care generarea unui impuls de comandă pentru elementele de execuție ale celor două motoare este urmată de generarea a patru impulsuri de comandă pentru elementul de execuție al motorului mecanismului de virare. Cele patru impulsuri sunt necesare deoarece, pentru un pas efectuat de către motoarele sistemelor de orientare, se realizează o deplasare unghiulară de  $1,8^\circ$  a camerelor video, iar pentru un pas efectuat de către motorul sistemului de virare se realizează o deplasare unghiulară de  $0,5^\circ$  a roții directoare.

Secvența de program pe baza căreia se efectuează rotirea motoarelor este următoarea:

```
ch|=0x01;  
ch&=0xFD;  
outp_WritePort(ch);  
ch|=0x12;  
outp_WritePort(ch);  
delay(20);
```

```
ch&=0xED;  
outp_WritePort(ch);  
delay(20);  
pasi=4;  
dreapta=0;  
limd=0;  
fan=1;  
gata=0;  
P1&= 0xFB;  
PWM1=125;  
EX1=1;
```

pentru rotirea spre stânga, respectiv

```
ch&=0xFE;  
ch|=0x04;  
outp_WritePort(ch)  
ch|=0x12;  
outp_WritePort(ch);  
delay(20);  
ch&=0xED;  
outp_WritePort(ch);  
delay(20);  
pasi=4;  
dreapta=1;  
fan=1;  
limleft=0;  
gata=0;  
P1 |= 0x04;  
PWM1=125;  
EX1=1;
```

pentru rotirea spre dreapta.

Așadar, prin biții 1 și 3 ai extensiei de port se stabilesc sensurile de rotație, apoi pe liniile 0 și 2 ale extensiei de port de ieșire se generează, la interval de 20ms, un nivel logic HIGH, respectiv un nivel logic LOW, care realizează avansarea cu un pas a celor două motoare. În continuare, prin portul **PWM1** se furnizează patru impulsuri cu factorul de umplere 0,5.

Sensul de rotație este prescris în funcție de codul caracterului generat prin programul de conducere care rulează pe PC. Astfel, la recepționarea codului corespunzător caracterului **a**, rotirea se va face spre stânga, iar la recepționarea codului caracterului **s**, rotirea se va face spre dreapta.

Rutinele de mai sus sunt prezentate în figura 6.9.

**În cazul (b)**, următorul pas în parcurgerea programului este intrarea într-o buclă infinită.

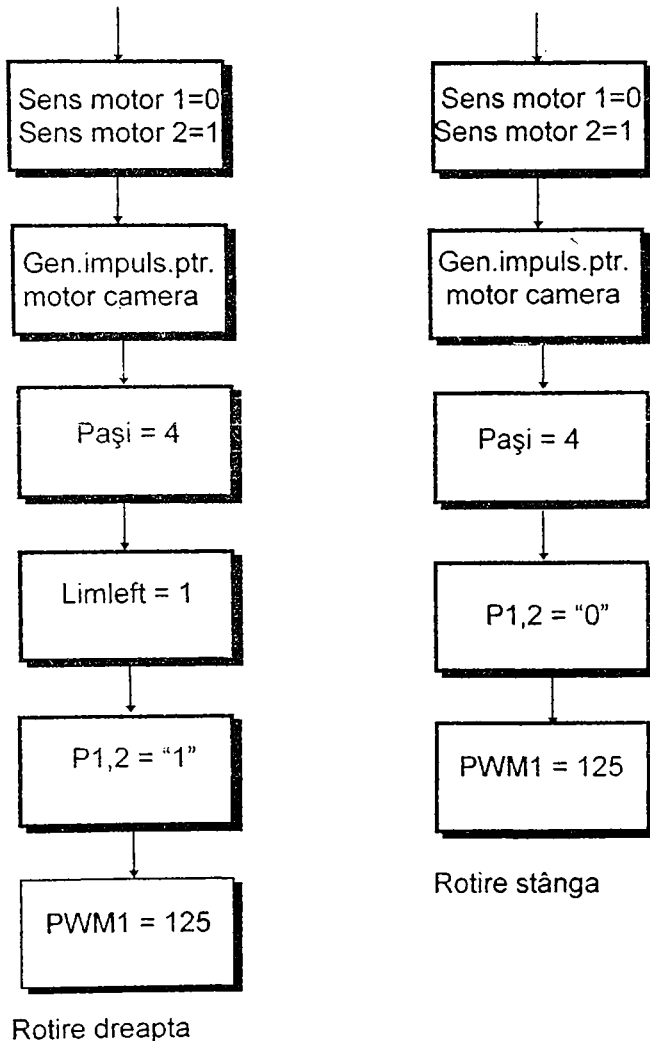


Figura 6.9.



Se inițializează un ciclu **for**, în cadrul căruia se generează câte un impuls pentru comanda elementelor de execuție ale motoarelor. Acest ciclu permite realizarea a 100 de pași, ceea ce echivalează cu un unghi de baleiere de 180°.

După parcurgerea celor 100 de pași se schimbă nivelul logic al semnalului **SENS** (se inversează sensul de rotație) și se reia bucla **for**.

```
for (i=0;i<100;i++)
{
    ch|=0x12;
    outp_WritePort(ch);
    delay(20);
    ch&=0xED;
    outp_WritePort(ch);
    delay(20);
}
```

Sensul de rotație este stabilit prin nivelurile logice generate pe liniile de port 1 și 3 astfel:

```
ch|=0x01;    //sens1=1
ch&=0xFD;   //sens2=0
outp_WritePort(ch);
```

respectiv

```
ch&=0xFE;   //sens1=0
ch|=0x04;   //sens2=1
outp_WritePort(ch);
```

Modul de amplasare a traductoarelor (**INIT**), utilizate pentru inițializarea mișcării de rotație a celor două motoare, este prezentat în figura 6.10.

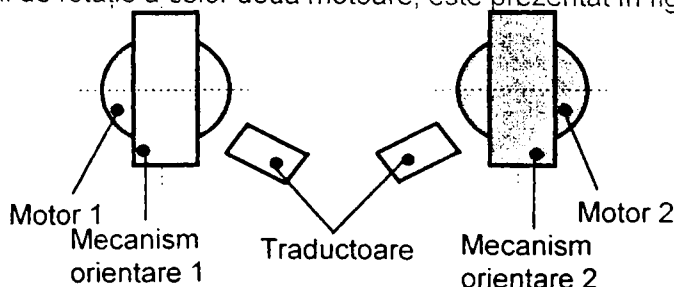


Figura 6.10.

Organigrama programului descris, corespunzător cazului (a) este prezentat în figura 6.11.

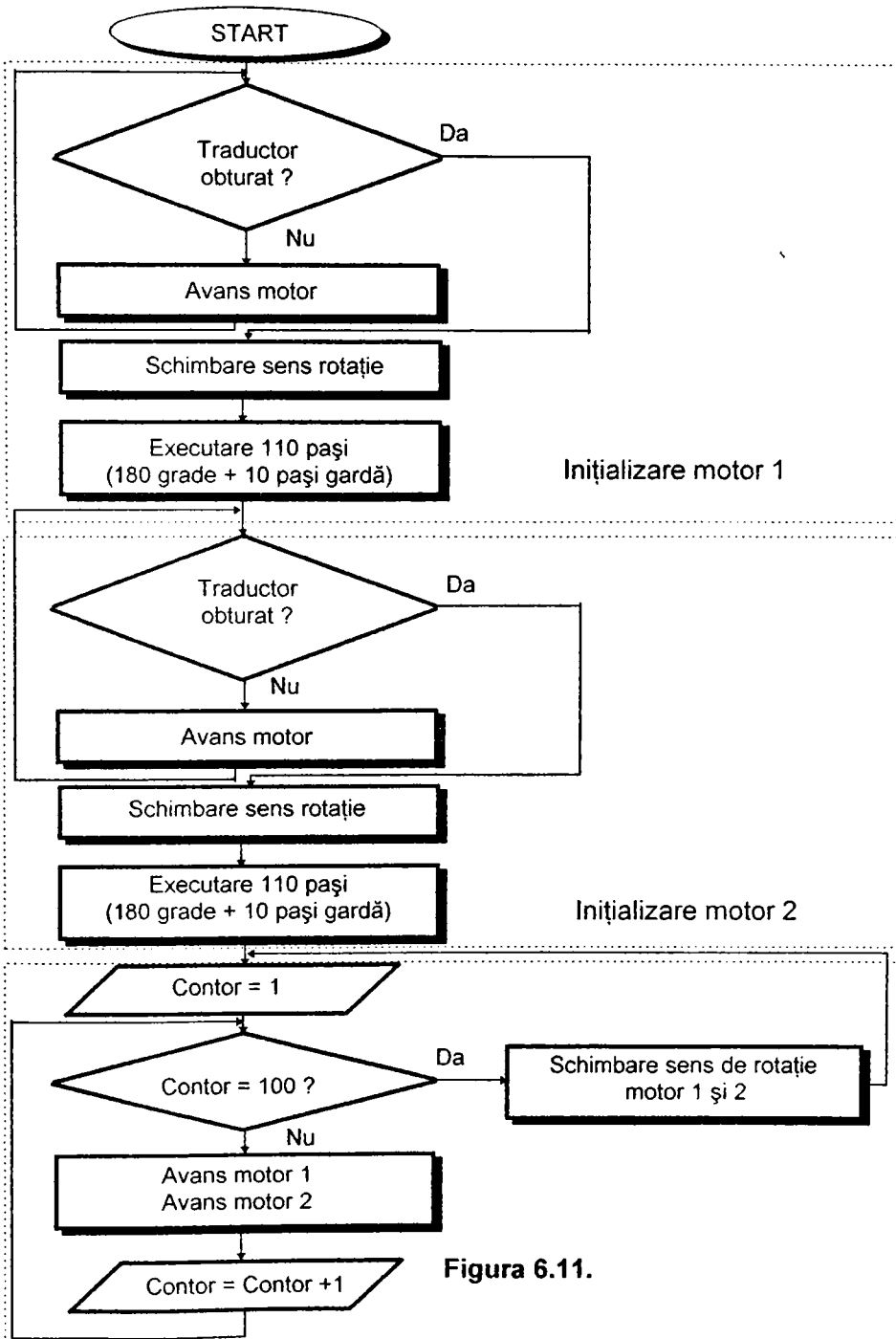


Figura 6.11.

## 6.2. Sistemul de conducere al vehiculului.

Sistemul de conducere al robocarului a fost astfel conceput, încât punerea în aplicație a oricărui program să fie ușor realizabilă. El comunică cu un calculator PC, care reprezintă sistemul ierarhic superior. În cazul în care programele pe care acesta le execută nu mai necesită modificări, se poate construi un sistem de conducere special destinat acestui scop. Oricare ar fi însă structura acestuia, dialogul cu sistemul de conducere ierarhic superior trebuie păstrat.

Sistemul de conducere al robocarului are ca element de bază microcontrolerul 80C552. El este alcătuit din două module hard:

- placa de bază **SDM-PB** (figura 6.12.);
- placa de extensie **SDM-EX01** (figura 6.13.).

De asemenea, sistemul de conducere conține și un pachet de programe dedicat, care rulează atât pe PC cât și pe sistemul de dezvoltare **SDM**, conectate prin interfața serială **RS-232**.

În continuare vor fi prezentate cele două module hard și pachetul de programe utilizat.

### 6.2.1. Placa de bază SDM-PB

Placa de bază este modulul esențial al sistemului de dezvoltare (figura 6.12). Elementele care compun acest modul și caracteristicile lor esențiale sunt următoarele:

- **Microcontrolerul 80C552**, care funcționează cu un semnal de tact având frecvența 11,0592 MHz;
- **Memoria EPROM** are capacitatea 32K și este adresată ca memorie de program;
- **Memoria RAM** are capacitatea 32K, și este adresată ca memorie de program și de date;
- **Memoria nevolatilă EEPROM** are capacitatea 512 octeți și este adresată pe magistrala internă I2C;
- **Interfața serială RS-232** realizează comunicarea cu calculatorul PC;

Microcontrolerul poate adresa o zonă de memorie externă cu capacitatea 64K, împărțită fizic și logic în două blocuri de câte 32K fiecare. Blocul care conține memoria de tip EPROM poate fi adresat numai ca memorie

de program, iar blocul care conține memoria de tip RAM poate fi adresat atât ca memorie de program, cât și ca memorie de date. Cele două zone astfel obținute nu se suprapun.

Un circuit de selecție existent pe placă permite alocarea adreselor pentru cele două zone astfel:

- memoria EPROM este adresată în zona de adrese (0000H-7FFFH) și memoria RAM în zona de adrese (8000H-FFFFH), situație indicată prin aprinderea LED-ului roșu aflat pe placă;
- memoria RAM este adresată în zona de adrese (0000H-7FFFH), iar memoria EPROM este adresată în zona de adrese (8000H-FFFFH). În acest caz LED-ul nu este alimentat.

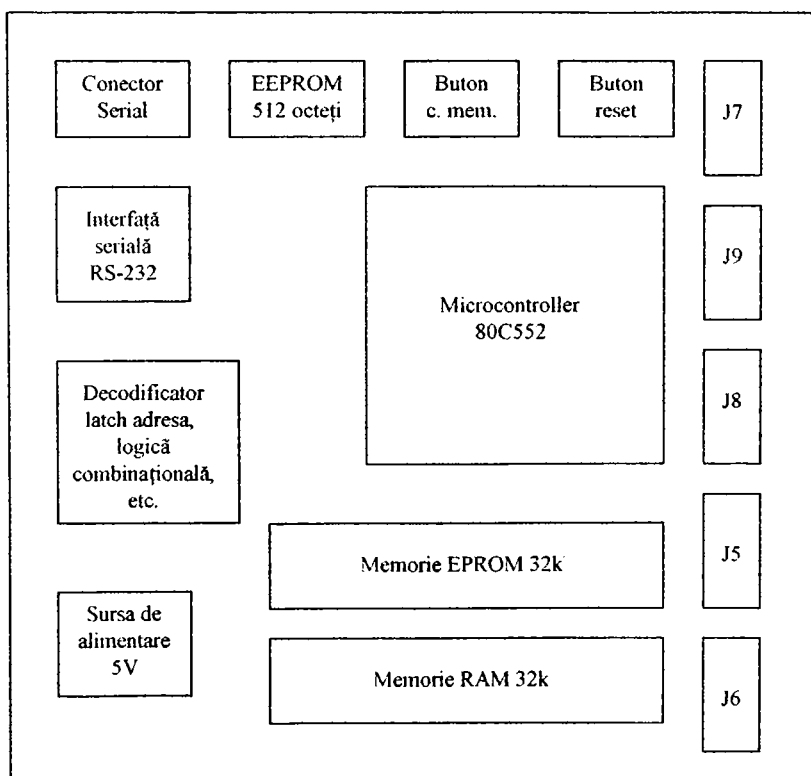


Figura 6.12.

Comutarea adreselor poate avea loc pe baza comenzilor primite de la calculator sau prin utilizarea tastei **Comutare memorie**, aflată pe placă. În momentul comutării adreselor microcontrolerul va fi resetat. Sistemul de dezvoltare poate fi resetat și cu ajutorul tastei **Reset**, aflată pe placa sistemului de dezvoltare.

## 6.2.2. Placa de extensie SDM-EX01

Următorul modul hard este placa de extensie **SDM-EX01** (figura 6.13) și are în componența sa elementele prezentate în continuare:

- afișajul cu cristale lichide LCD;
- tastatura cu 12 taste;
- un port de intrare, realizat cu circuite integrate MOS;
- un port de ieșire, realizat cu circuite integrate MOS;
- un port de ieșire de putere de tip colector în gol, realizat cu tranzistoare npn care au  $I_{cmax}=1A$ ;

Afișajul are 1x16 caractere, iar contrastul acestuia poate fi reglat de către utilizator.

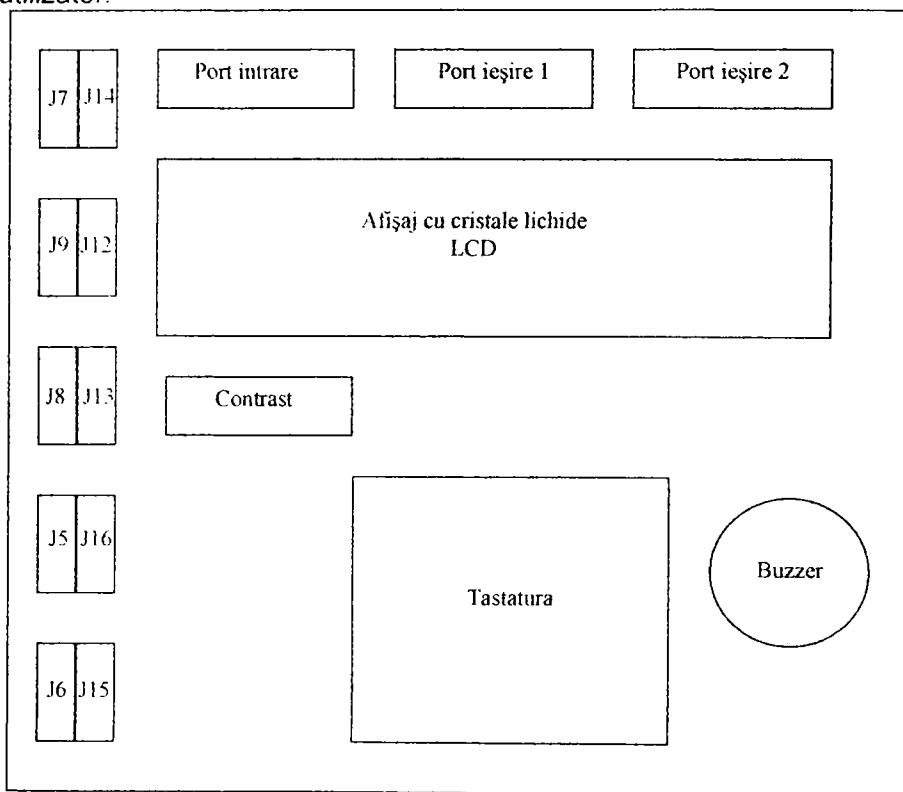


Figura 6.13.

Portul de intrare este realizat cu un circuit integrat 74HC244 și poate fi utilizat pentru achiziționarea unor semnale pe un număr de opt linii.

Portul de ieșire este realizat cu un circuit integrat 74HC374 și poate fi utilizat pentru generarea unor semnale pe un număr de opt linii. Portul de ieșire de putere este realizat cu un circuit integrat 74HC374 și cu opt tranzistoare npn de putere. El poate fi folosit pentru comanda unor echipamente alimentate de la o sursă de tensiune aflată în exteriorul sistemului.

Afișajul, tastatura și porturile de intrare/ieșire sunt adresate în zona memoriei de date, neocupată de memoria de date a sistemului.

După cum se observă din figura 6.12 și din figura 6.13, este posibil accesul din exterior la toate porturile microcontrolerului datorită cuplei duble J5-J9, J12-J16 (2x5x16 pini). Prin această cuplă sunt disponibile toate semnalele generate de către circuitele electronice aflate placă.

### 6.2.3. Mediul integrat de dezvoltare PCMON

Programul principal al **SDM** este un pachet integrat care a fost conceput astfel încât să realizeze optimizarea procesului de dezvoltare a programelor utilizate pentru anumite aplicații. Acest program oferă utilizatorului următoarele facilități:

- încărcarea, editarea și salvarea programelor sursă, folosind editorul de text încorporat;
- apelarea programelor utilitare pentru conversia programului în format hexazecimal;
- încărcarea programului în format hexazecimal în **SDM** și lansarea în execuție a programului;
- depanarea programelor, folosind Debugger-ul încorporat;
- editarea programelor utilizator;
- lansarea în execuție direct din mediul **PCMON** a unui program încărcat în prealabil pe **SDM**;
- apelarea programului care editează o bibliotecă.

Utilizarea mediului **PCMON** este simplă și se realizează prin intermediul ferestrelor, meniurilor, butoanelor și mouse-ului. Sunt disponibile 6 meniuri principale: **FILE**, **EDIT**, **SEARCH**, **TOOLS**, **OPTIONS**, **WINDOW**. Opțiunile aflate în aceste meniuri determină executarea unor acțiuni și afișarea unor submeniuri sau ferestre de dialog. Operarea în cadrul interfeței se poate realiza utilizând tastatura sau mouse-ul. De asemenea sunt disponibile o serie de taste rapide, pentru selecția opțiunilor utilizate mai des.

a.) Meniul **FILE** este meniul utilizat pentru operațiile cu fișiere și este prezentat în figura 6.14. Operațiile care se pot realiza din acest meniu sunt: crearea unui fișier nou, deschiderea pentru editare a unui fișier existent, salvarea unui fișier editat și salvarea fișierului cu un alt nume. Opțiunea **Save all** permite salvarea tuturor fișierelor care sunt deschise în acel moment. Opțiunea **Change dir** se utilizează pentru schimbarea directorului curent în care se lucrează. Cu ajutorul comenzii **Exit** se părăsește mediul de dezvoltare **PCMON**.

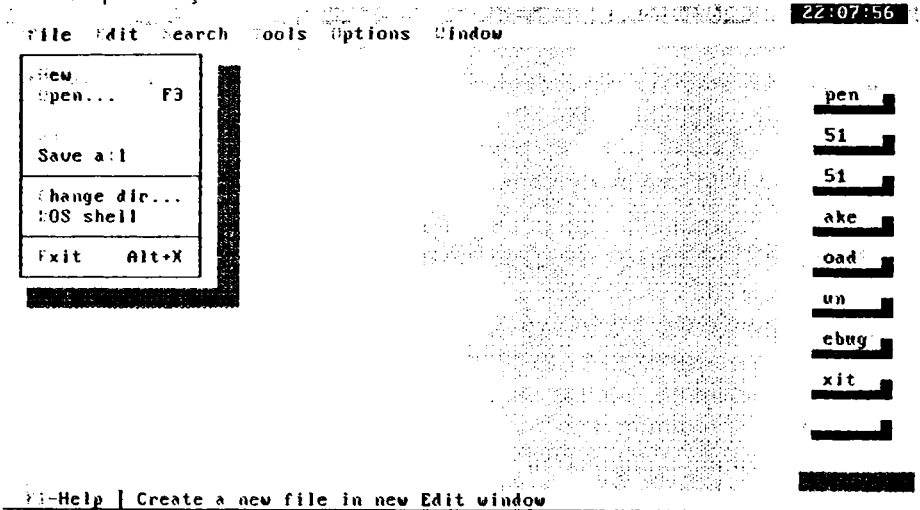


Figura 6.14.

b.) Meniul **EDIT** conține comenzile de editare specifice, prezentate în figura 6.15.

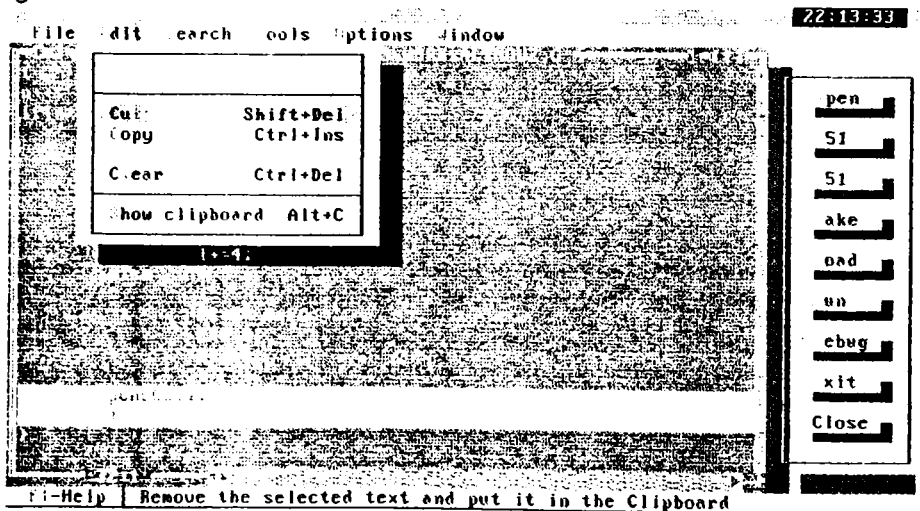


Figura 6.15.

c.) Meniul **SEARCH** conține comenzile necesare căutării și înlocuirii unor șiruri de caractere (figura 6.16). Înlocuirea se poate face automat sau cu prin confirmare de către utilizator.

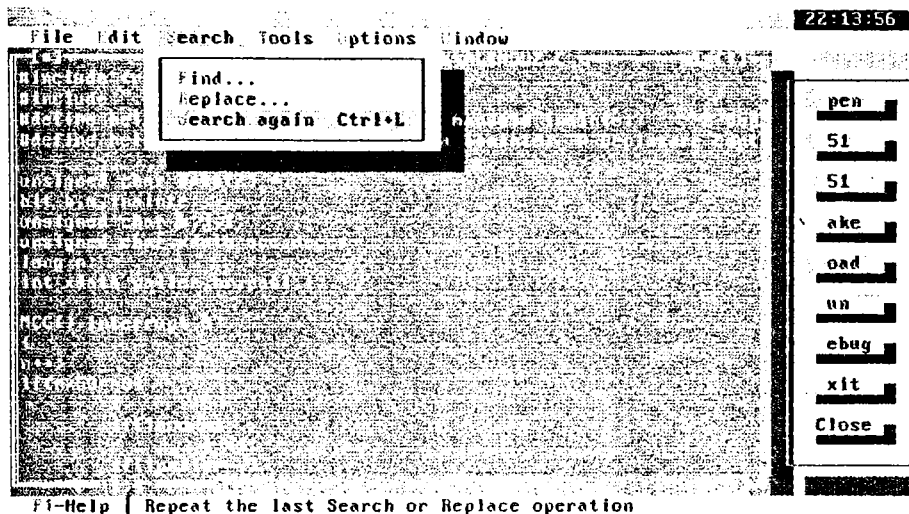


Figura 6.16

d.) Meniul **TOOLS** conține comenzile necesare încărcării programului în sistemul de dezvoltare, lansarea în execuție a programelor utilitare și lansarea în execuție a programului aflat în sistemul de dezvoltare (figura 6.17). Pentru fiecare dintre aceste programe se poate edita linia de comandă corespunzătoare utilizând meniul **OPTIONS**. Opțiunea **Make** permite automatizarea procesului de generare a fișierului program în format hexazecimal (se apelează succesiv compilatorul, asamblorul, link-editorul și programul de conversie în format hexazecimal). Meniul **OPTIONS** permite setarea unor opțiuni ale utilizatorului (figura 6.18). Se poate seta portul serial prin care se va realiza comunicația între calculator și **SDM** (COM1 sau COM2) și rata de comunicație a calculatorului cu **SDM** (în cazul acestui sistem comunicația se realizează cu rata de transfer de 9600 baud). De asemenea se pot selecta programele care vor fi apelate în cadrul opțiunii **Make**. Aceste opțiuni sunt salvate în fișierul **PCMON.INI** (din directorul curent) la ieșirea din program. Când are loc următoarea execuție a programului, opțiunile vor fi citite din acest fișier sau vor primi valorile implicite în cazul în care fișierul nu există în directorul curent.



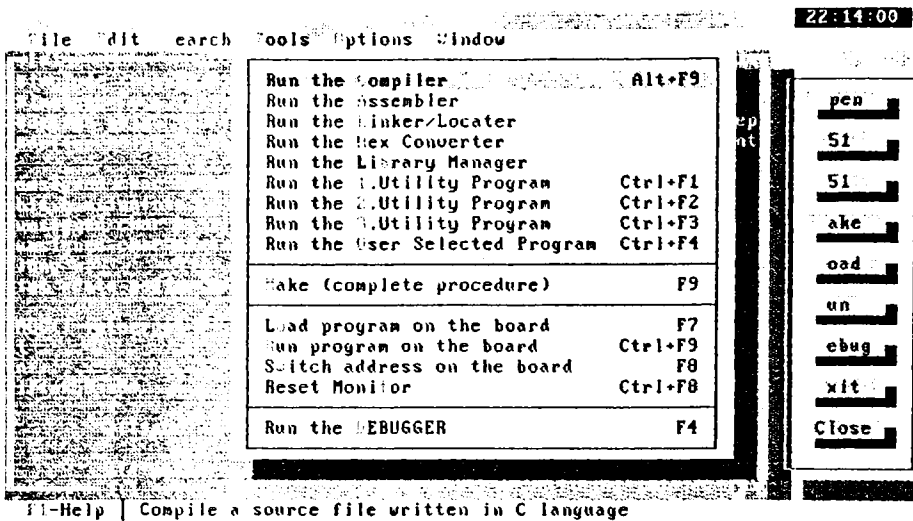


Figura 6.17.

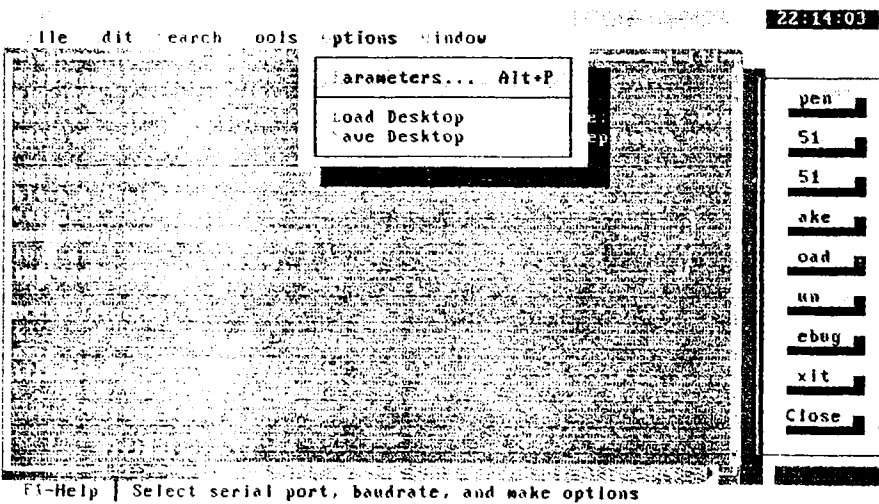


Figura 6.18.

În acest capitol au fost prezentate cele mai importante meniuri. Meniul **WINDOW**, precum și **debuger-ul** și **biblioteca de funcții** se află în anexa7.

### 6.3. Concluzii.

Concluziile prezentate în continuare se referă la sistemul de acționare și la sistemul de conducere, iar în încheierea acestora se va analiza schema bloc a echipamentului electric și electronic al robocarului construit.

#### Sistemul de acționare.

În capitolul 1 s-a arătat că sistemul electric și electronic al unui VGA cuprinde, pe de o parte echipamente electrice și pe de altă parte echipamente electronice. În cadrul tezei nu au fost analizate toate elementele constructive ale sistemului electric și electronic, cu toate că **preocupările autorului s-au extins asupra fiecăruia dintre ele.** Sistemul de acționare, după cum rezultă din schema bloc prezentată în figura 1.2 cuprinde, atât o parte a echipamentelor electrice, cât și o parte a echipamentelor electronice. Aceste echipamente sunt cele din figura 6.19 și anume:

- ◆ elementele de execuție;
- ◆ motoarele;
- ◆ traductoarele.

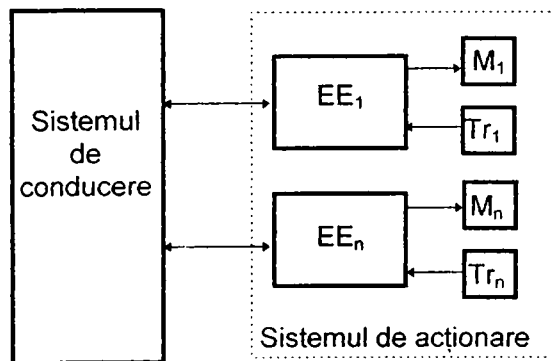


Figura 6.19.

Structura elementelor de execuție, tipul motoarelor și al traductoarelor folosite au fost prezentate pe parcursul capitolului 6.

Ele pot fi realizate în nenumărate forme constructive. Aceste forme se referă în primul rând la componente electronice utilizate (componente discrete sau circuite integrate). Avantajele utilizării tranzistoarelor sau a circuitelor integrate MOS sunt evidente și cunoscute.

Cerințele care se impun fiecărui element de execuție în parte au fost analizate odată cu prezentarea acestora. Se subliniază însă trei cerințe pe care elementele de execuție trebuie să le îndeplinească:

- ◆ să fie realizate sub formă modulară;
- ◆ consumul de energie electrică al acestora să fie redus;
- ◆ să necesite un număr cât mai mic de semnale de comandă.

Circuitul de putere din cadrul fiecărui element de execuție este realizat cu un circuit integrat de tipul L298. Acesta este o punte dublă formată din tranzistoare bipolare, completată cu circuitele logice necesare comandării acestora. Funcțiile circuitului sunt selectate cu ajutorul celor trei semnale, **IN1**, **IN2**, **EN**.

Circuite integrate similare cu L298 sunt:

- a. L6202; L6203, care au aceeași schemă bloc, dar tranzistoarele finale sunt MOS;
- b. IR8200, care are tranzistoare MOS și trei intrări, **DIRECTION**; **PWM** și **BRAKE**. Semnalele de intrare necesare respectă aceeași logică cu a altor circuite [PCIM 90];
- c. A3952SB este o punte cu tranzistoare bipolare, incluzând și circuitul de comandă. Are trei intrări: **BRAKE**, **PHASE**; **ENABLE**, cu semnificații asemănătoare cu a celorlate circuite.

Pot fi utilizate și circuite integrate de tipul A3955SB, A3966SLB, A3968, L6219DS [ALG 99]. Deoarece circuitul integrat L298 este un circuit integrat uzual, autorul a optat pentru realizarea elementului de execuție cu ajutorul acestuia.

Construcția sub formă modulară este necesară pentru a fi posibile anumite îmbunătățiri ulterioare în ceea ce privește echipamentele electronice ale robocarului sau setul de programe utilizat. Consumul de energie redus al echipamentelor amintite mărește durata de utilizare a acumulatorilor (intervalul dintre două încărcări succesive).

Minimizarea numărului semnalelor de comandă este importantă din două puncte de vedere:

- ◆ face mai ușoară realizarea modulară;
- ◆ degrevează sistemul de conducere de anumite sarcini referitoare la sistemul de acționare, astfel încât resursele acestuia pot fi utilizate în alte scopuri.

Dacă semnalele de comandă sunt unificate (de exemplu, se pot utiliza două semnale: unul pentru prescrierea vitezei și unul pentru prescrierea sensului de rotație al motoarelor) și numărul lor este cât mai redus, realizarea modulelor elementelor de execuție se simplifică.

**Elementele de execuție realizate îndeplinesc toate condițiile prezentate pe parcursul capitolului și în concluzii.** Ele au fost construite sub formă modulară și necesită un număr minim de semnale de comandă. Concepția tipizată se referă și la conexiunile externe ale modulelor, ele fiind interschimbabile (bineînțeles, acest lucru presupune schimbarea motoarelor și a modulului de generare a semnalelor de comandă).

### **Sistemul de conducere.**

Funcțiile sistemului de conducere au fost prezentate în capitolul 2, paragraful 2.4. Din cele arătate până în prezent, se poate deduce că sistemul de conducere al vehiculului poate îndeplini toate aceste funcții, mai puțin determinarea situației pe baza imaginii achiziționate cu ajutorul camerei video. Pentru realizarea acestei funcții este necesară o placă specială pentru achiziția imaginilor.

Alegerea microcontrolerului 80C552 ca element esențial al sistemului de dezvoltare (care are rolul sistemului de conducere al vehiculului) a avut loc pe baza unei analize temeinice a modului de lucru, pentru rezolvarea problemelor din cadrul tezei și din viitor. Conceperea și experimentarea rapidă a unor programe nu ar fi fost posibilă decât prin utilizarea unui sistem de dezvoltare.

Utilizarea registrelor **PWM** ale microcontrolerului 80C552 face posibilă generarea semnalelor de comandă a elementelor de execuție ale celor două motoare simultan și independent, fără ocuparea inutilă a resurselor microcontrolerului (timere și linii de port). Aceste resurse pot fi utilizate pentru implementarea unor algoritmi de reglare sau pentru execuția altor funcții cum ar fi:

- ◆ stabilirea limitelor mecanismului de virare;
- ◆ învățarea unei traiectorii;
- ◆ orientarea camerei video;
- ◆ procesarea informațiilor achiziționate cu ajutorul US, etc.

Sistemul de dezvoltare se află în comunicare continuă cu un calculator PC. Comunicarea este utilizată pentru atribuirea unor sarcini sistemului

de conducere al vehiculului de către sistemul de conducere superior ierarhic.

**Elementele de originalitate aduse de către autor** constau în primul rând în alegerea soluțiilor pentru realizarea elementelor de execuție și a sistemului de conducere, iar apoi în conceperea, proiectarea și construirea efectivă a acestora. Elementele de execuție construite au fost optimizate. S-a folosit același tip de circuit integrat pentru toate elementele de execuție, prin aducerea unor modificări în schema de aplicație recomandată de către producător. De asemenea au fost concepute programe originale utilizate pentru generarea semnalelor de comandă. Modularizarea și tipizarea elementelor de execuție și a sistemului de comandă sunt de asemenea contribuții ale autorului. A rezultat un sistem de acționare și conducere modern, cu numeroase facilități, a cărui schemă bloc este prezentată în figura 6.20.

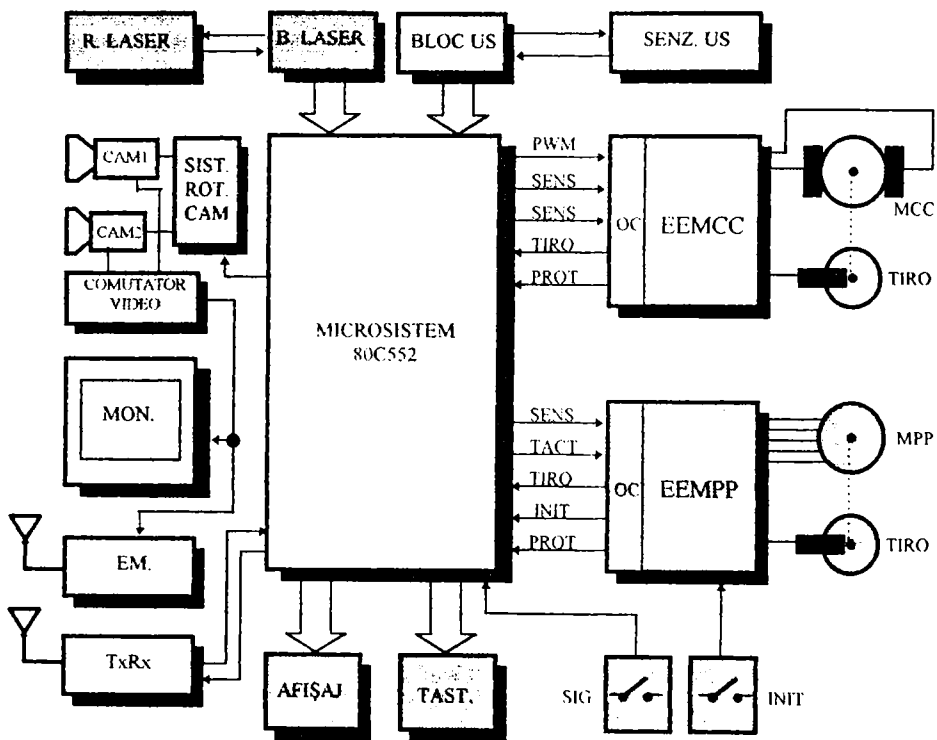


Figura 6.20.

Sistemul de conducere (sistemul de dezvoltare realizat cu microcontrolerul 80C552) îndeplinește următoarele funcții:

- comandă EEMCC; semnale PWM, SENS1, SENS2;

- comandă **EEMPP**; semnale **SENS, TACT**;
- comandă elementele de execuție ale sistemului de orientare a camerei video ;
- interpretează semnalul de reacție furnizat de către traductorul de impulsuri atașat motorului de curent continuu ; semnal **TIRO**;
- interpretează semnalul de reacție furnizat de către traductorul de impulsuri atașat motorului pas cu pas (semnal **TIRO**);
- interpretează impulsul furnizat de către traductorul utilizat pentru inițializare (semnal **INIT**);
- interpretează informațiile furnizate de către senzorii ultrasonici (**BLOC US**);
- interpretează informațiile furnizate de către receptoarele laser (**BLOC LASER**);
- implementarea programelor de navigație;
- realizează protecția împotriva coliziunii (semnal **SIG**);
- comandă circuitele de comutație video (**BLOC COM. VIDEO**).
- comandă sistemul de rotire a camerelor video;
- realizează dialogul cu operatorul uman prin intermediul afișajului și a tastaturii (**AFIȘAJ, TAST**);
- realizează comunicarea cu sistemul de comandă aflat pe un nivel superior.

**Cu ajutorul acestui echipament, amplasat în cadrul construcției mecanice a robocarului, s-au efectuat numeroase experimente, sintetizate în capitolul 7. Aceste experimente sunt originale și au confirmat ipotezele teoretice ale autorului.**



## Capitolul 7

### REZULTATE EXPERIMENTALE

Robocarul prezentat a fost construit cu scopul realizării unui suport material necesar verificării soluțiilor propuse de către autor pentru sistemul mecanic, de acționare și de conducere. De asemenea, pe baza construcției existente, au fost verificate programele și procedeele utilizate pentru determinarea situației.

**Toate echipamentele și metodele prezentate în cadrul tezei au fost verificate experimental de către autor.**

Experimentele au decurs în mai multe etape. Ordinea acestora este însă diferită de ordinea prezentării capitolelor tezei. Ele s-au desfășurat după cum urmează.

#### **7.1. Verificarea sistemului mecanic.**

În primul rând a fost verificată funcționarea mecanismului de tracțiune. Motorul de curent continuu a fost alimentat de la o sursă de tensiune de 24V și s-a măsurat curentul absorbit de către acesta, robocarul fiind încărcat cu o sarcină utilă de 150kg. Valoarea măsurată a curentului a fost 1,45A.

În continuare s-a verificat mecanismul de virare prin simularea acestuia cu ajutorul programului de simulare a mecanismelor MECAPLAN.

#### **7.2. Experimente efectuate cu elementele sistemului de acționare.**

În figura 7.1 se prezintă standul cu ajutorul căruia s-a verificat funcționarea elementelor de execuție. S-a utilizat un osciloscop PHILLIPS PM 3350 A și un generator de semnal PGP 5. Cu ajutorul osciloscopului se pot vizualiza simultan două semnale, care sunt apoi digitalizate și transmise unui calculator PC prin interfața serială. Generatorul de semnal poate furniza impulsuri cu factorul de umplere variabil, care sunt utilizate drept semnale de comandă ale elementelor de execuție. Toate oscilogramele prezentate în capitolul de față au fost ridicate în acest mod.

S-a conectat la ieșirea fiecărui element de execuție motorul corespunzător, iar la intrare s-au aplicat semnale de comandă obținute cu ajutorul generatoarelor de semnal. Funcționarea a fost verificată prin

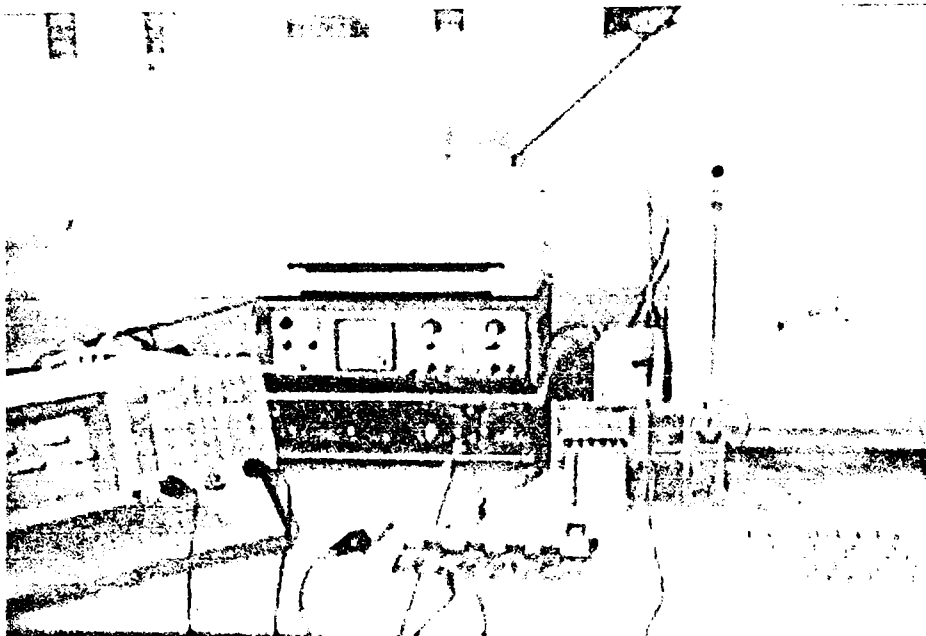


Figura 7.1.

oscilografiera semnalelor în diverse puncte ale circuitului (semnalele aplicate la intrare., semnalele obținute la ieșire, etc.) și prin măsurarea unor tensiuni și curenți (curentul prin sarcină, tensiunea la bornele sarcinii, etc.). Aceste mărimi au fost estimate în prealabil prin analiza teoretică a circuitelor respective.

### 7.2.1. Verificarea EEMCC.

Elementul de execuție al motorului de curent continuu a fost realizat în varianta definitivă sub forma unui modul ale cărui dimensiuni fac posibilă montarea în compartimentul aflat pe robocar și destinat acestui scop.

Înainte de montarea pe robocar, elementul de execuție a fost verificat cu ajutorul aparatelor care fac parte din standul prezentat.

Deoarece modulele sunt prevăzute cu cuple pentru intrări și ieșiri, conectarea generatoarelor de semnal la intrare și a motoarelor la ieșire



poate fi realizată foarte ușor. În figura 7.2 se prezintă modul în care a fost construit elementul de execuție al motorului de curent continuu.

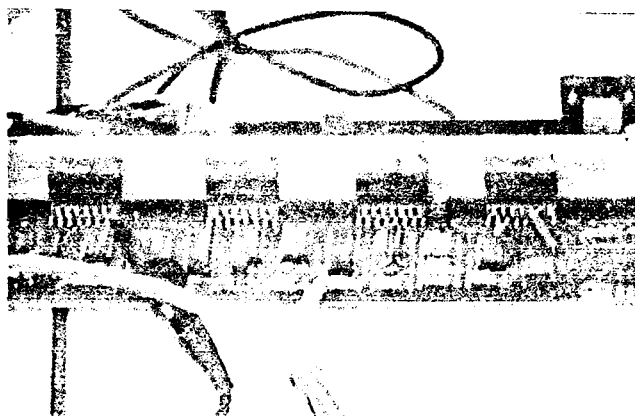


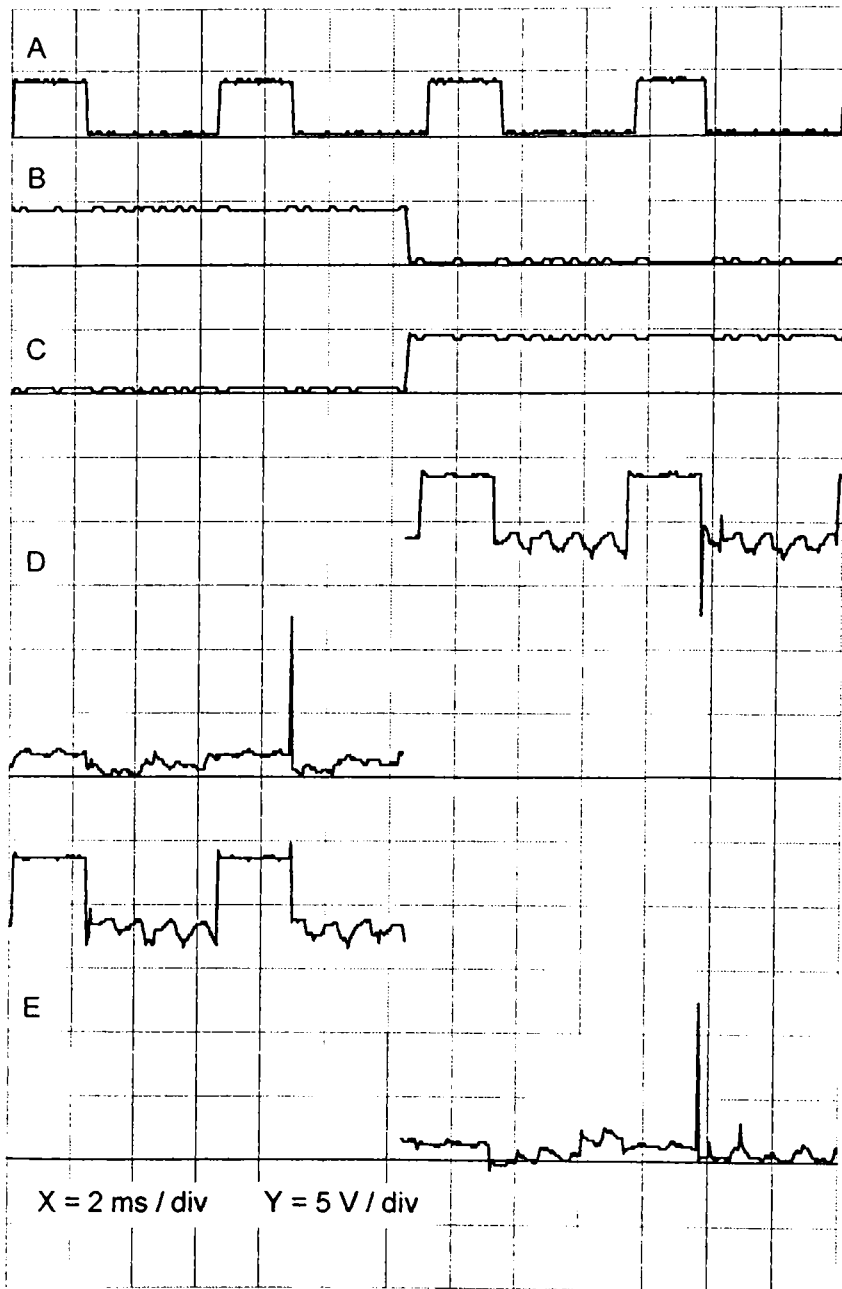
Figura 7.2.

EEMCC a fost verificat prin oscilografiera semnalelor care caracterizează funcționarea circuitului. Ele sunt reprezentate în figura 7.3: **A** - PWM; **B** - SENS1; **C** - SENS2; **D** - OUT1; **E** - OUT2. Prin modificarea tensiunii aplicate motorului între 0V și 24V s-au obținut turații cuprinse între 0 și 3000 rot/min. Modificarea tensiunii s-a realizat prin modificarea factorului de umplere între 0 și 1.

Până în momentul schimbării sensului de rotație, motorul este alimentat între ieșirea **OUT2**, aflată la +24V și ieșirea **OUT1**, aflată la 0V. După schimbarea sensului, **OUT2** devine 0V, iar **OUT1** se află la 24V.

Impulsurile modulate în durată, obținute cu ajutorul generatorului de semnal sau, în cazul funcționării automate, de la sistemul de conducere, sunt aplicate intrării **EN** (diagrama A). Semnalele în antifază necesare schimbării sensului de rotație se aplică intrărilor **IN1** și **IN2** (diagramele B și C).

Dacă nivelul logic aplicat intrării **EN** este ridicat, vor conduce două dintre tranzistoarele de putere aflate pe laturi diferite ale punții, în funcție de nivelurile semnalelor aplicate intrărilor **IN1** și **IN2**. Nivelul ridicat al semnalului reprezentat în diagramele D, respectiv E (24V) se obține prin saturarea tranzistorului corespunzător aflat în partea superioară a punții. Atunci când tranzistoarele sunt blocate, tensiunea obținută la bornele motorului este tensiunea generată de către acesta în regim de generator.



**Figura 7.3.**

În continuare s-a reglat valoarea maximă admisă a curentului la  $I_{\max} = 1,7\text{A}$ . Aceasta s-a realizat prin conectarea între ieșirile OUT1 și OUT2 a unei rezistențe variabile cu valoarea de  $25 \Omega$ . S-a stabilit factorul de

umplere al impulsurilor de comandă (semnalul **PWM**) la valoarea 0,5 și s-a modificat rezistența până când curentul a devenit  $I_{max} = 1,7A$ . Pentru această valoare a fost reglat pragul la care acționează protecția la supracurent.

Apoi s-a verificat funcționarea traductorului inductiv și a circuitului utilizat pentru procesarea semnalului furnizat de către acesta, formele de undă obținute fiind reprezentate în figura 7.4.

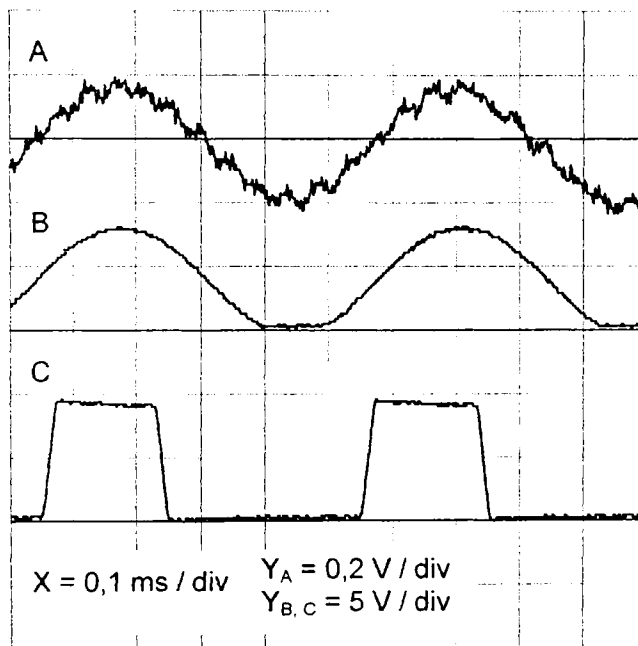


Figura 7.4.

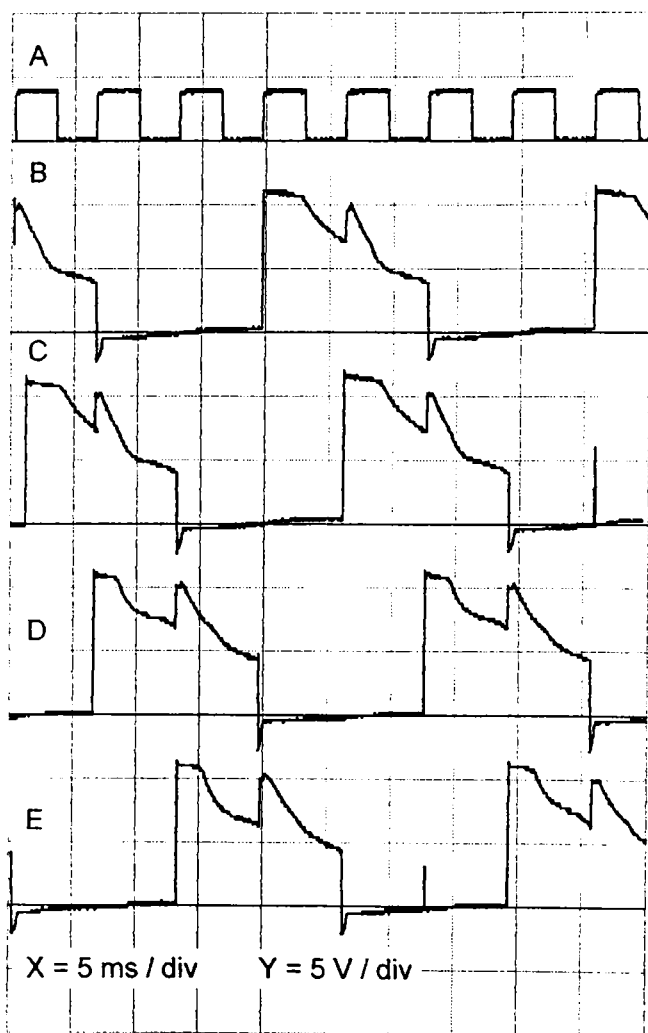
Diagrama **A** reprezintă tensiunea alternativă obținută la ieșirea traductorului inductiv. Diagrama **B** este tensiunea obținută la ieșirea amplificatorului prin amplificarea semnalului furnizat de către traductor, iar diagrama **C** reprezintă semnalul obținut la ieșirea comparatorului.

### 7.2.2. Verificarea EEMPP.

Forma constructivă a **EEMPP** este asemănătoare cu aceea a **EEMCC**. Funcționarea sa a fost verificată utilizând standul descris anterior. În figurile 7.5 și 7.6 sunt reprezentate principalele forme de undă ale

semnalelor care caracterizează funcționarea elementului de execuție. Aceste sunt: **A** - TACT; **B** - OUT1; **C** - OUT2; **D** - OUT3; **E** - OUT4.

În figura 7.5 sunt reprezentate formele de undă ale semnalelor furnizate la cele patru ieșiri, corespunzătoare celor patru faze ale motorului pas cu pas.



**Figura 7.5.**

În figura 7.6 se observă schimbarea succesiunii acestor semnale, deci schimbarea sensului de rotație.

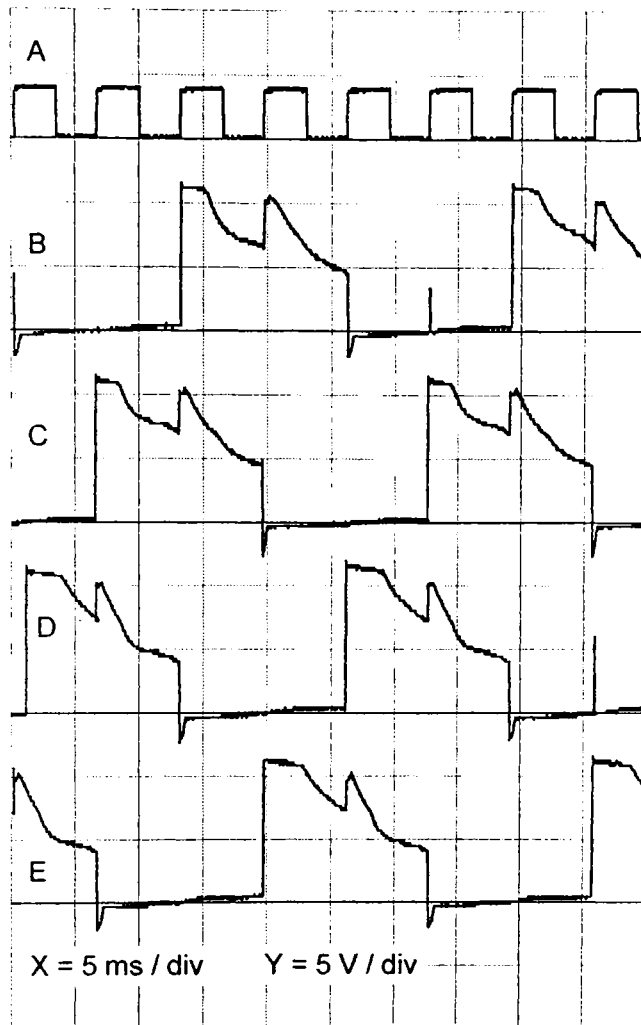
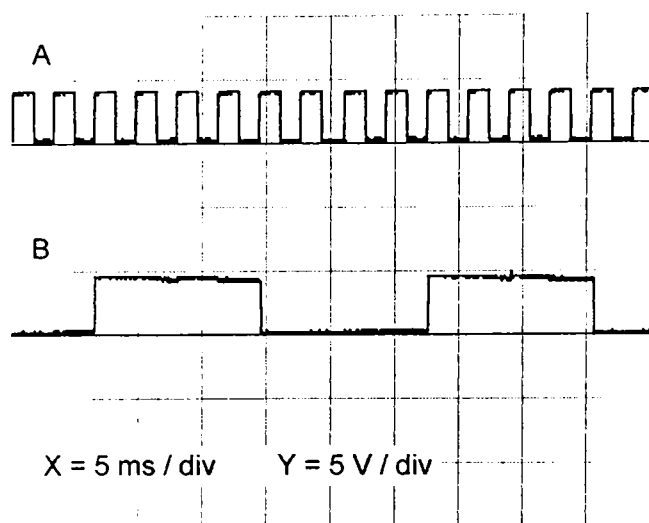


Figura 7.6.

În continuare s-a măsurat frecvența maximă a impulsurilor de tact, astfel încât motorul să nu piardă pași. Aceasta a fost  $f_{\max}=250\text{Hz}$ .

Curentul absorbit de către motor are valoarea  $I = 0.4\text{A}$ . Apoi s-a reglat valoarea maximă admisă a acestui curent  $I_{\max} = 0.5\text{A}$ , procedându-se la fel ca și pentru **EEMCC**.

În încheiere s-a testat funcționarea traductorului incremental (figura 7.7, diagrama **B**) și a traductorului utilizat pentru inițializare.



**Figura 7.7.**

Analizând diagramele **A** și **B**, se observă că, pentru ca sistemul de virare să se rotească cu un grad, sunt necesare 16 impulsuri de tact (**A**), iar traductorul incremental va genera, în acest interval de timp, două impulsuri. Motorul pas cu pas a efectuat 16 pași (a se vedea mențiunea din pagina 5.8).

### **7.2.3. Verificarea elementelor de execuție ale sistemului de orientare a camerei video.**

S-au efectuat aceleași măsurători ca și în paragraful 7.2.2, cu mențiunea că frecvența maximă a impulsurilor de tact este  $f_{\max} = 200\text{Hz}$ , iar curentul absorbit de către o fază a motorului are valoarea  $I = 0.25\text{A}$ .

### **7.3. Testarea programelor utilizate pentru deplasarea robocarului.**

S-au generat cu ajutorul sistemului de dezvoltare semnalele de comandă necesare fiecărui element de execuție și s-a verificat funcționarea acestora.

În figura 7.8 este prezentat sistemul de dezvoltare utilizat, care are rolul sistemului de conducere a robocarului.

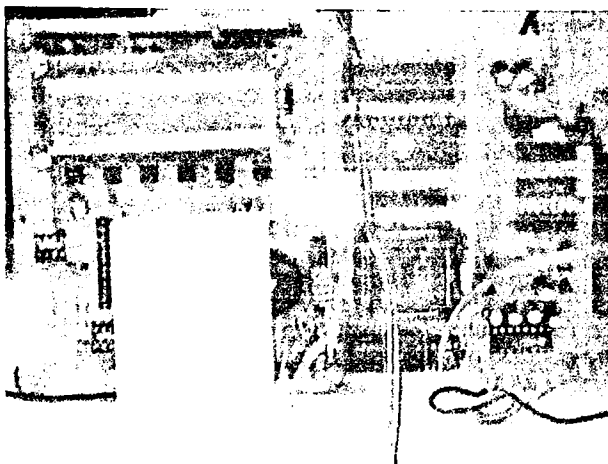


Figura 7.8.

S-a verificat funcționarea programului cu ajutorul căruia se realizează deplasarea pe traiectoria prescrisă (prezentat în capitolul 3, pg. 3.13). Precizia de poziționare a robocarului obținută este mai bună de  $\pm 0,15\text{cm}$  pentru o deplasare de 1m. În figura 7.9 se prezintă modul în care a avut loc poziționarea robocarului după parcurgerea acestei distanțe.

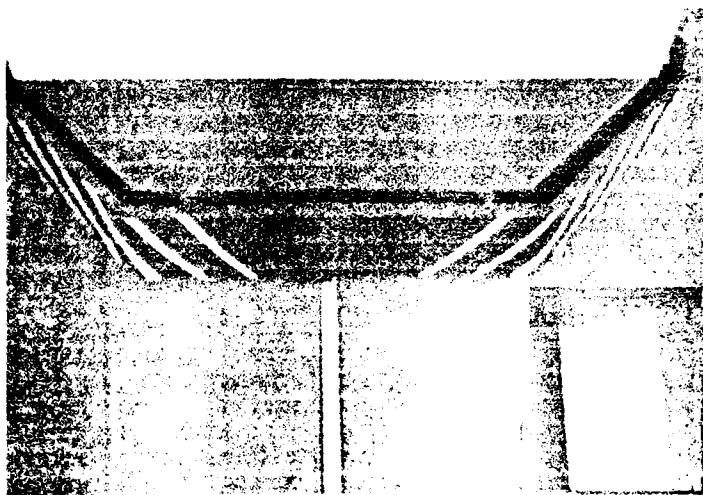


Figura 7.9.

În partea dreaptă jos a imaginii se observă în detaliu poziția indicatorului fixat de robocar în raport cu cifra 100 (cm), aflată pe rigla gradată utilizată pentru măsurarea spațiului parcurs.

Un alt experiment pe baza căruia poate fi stabilită precizia și repetabilitatea poziționării robocarului este descris în continuare.

Vehiculul a fost deplasat pe o suprafață orizontală netedă, spațiul parcurs de către acesta fiind 2000mm. Deplasarea a avut loc automat, prin prescrierea distanței. Poziționarea robocarului în punctul de pornire s-a realizat cu ajutorul unui comparator de rezoluție 0.01mm. Punctul final s-a stabilit la o distanță de 2000mm de punctul inițial, cu ajutorul unui comparator identic cu cel folosit pentru stabilirea punctului de plecare (figura 7.10). Robocarul a fost adus în poziția de pornire manual, iar în poziția finală, automat.

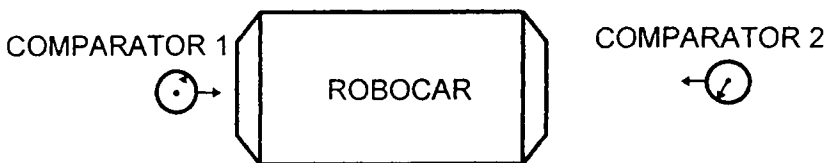


Figura 7.10.

S-au efectuat câte  $n=42$  măsurări, pentru două calități ale pardoselii (uscat și umed), condițiile fiind menținute identice pentru fiecare caz. Rezultatele măsurărilor sunt prezentate în tabelul 7.1.

Tabelul 7.1.

Nr. măs.	x		Nr. măs.	x		Nr. măs.	X	
	uscat	umed		uscat	umed		uscat	umed
1	1999,37	1999,11	15	2000,11	1995,05	29	2001,04	1999,21
2	2000,28	2000,02	16	2001,08	1997,25	30	2000,43	1998,31
3	2000,09	2000,22	17	2002,12	1999,55	31	2001,87	1997,03
4	2001,05	2001,13	18	1999,95	1999,88	32	2000,36	1998,12
5	2002,21	1996,22	19	2000,18	2000,74	33	1999,98	1999,68
6	1999,68	1998,52	20	2000,02	2001,02	34	2000,04	1998,85
7	1995,35	1998,91	21	2003,24	1999,14	35	1999,87	1998,77
8	1997,52	1999,41	22	1997,81	1995,81	36	2001,06	1999,47
9	2003,83	2000,26	23	2000,06	1996,83	37	2000,14	1998,77
10	2001,01	1998,77	24	1999,93	1998,44	38	1998,51	1999,37
11	2001,12	1998,99	25	1999,25	2002,09	39	2003,41	2000,15
12	1999,74	1999,25	26	2001,13	2001,98	40	2000,28	2000,43
13	1998,22	2000,13	27	2002,34	1996,15	41	1997,14	2000,77
14	1999,43	1996,12	28	2002,09	1998,15	42	1996,12	2000,98

Utilizând programul MATLAB, valorile măsurate pentru cele 2 cazuri au fost aranjate în ordine crescătoare ( $x_{\min} \div x_{\max}$ ) și au fost calculate:



- intervalul  $\delta$  (cu formula lui Sturges):

$$\delta = \frac{x_{\max} - x_{\min}}{1 + 3,22 \log n} \quad (7.a)$$

- media aritmetică:

$$\bar{x} = \sum_{i=1}^k f_i x_i = \frac{1}{n} \sum_{i=1}^n x_i \quad (7.b)$$

unde  $f_i = \frac{n_i}{n}$  este frecvența relativă

- eroarea medie pătratică:

$$\sigma = \sqrt{\frac{1}{n-1} \sum (x_i - \bar{x})^2} \quad (7.c)$$

Programul utilizat pentru prelucrarea valorilor măsurate este prezentat în continuare:

```
clear
n=input('Numarul de masurari=');
for i=1:n
    x(i)=input("");
end
sort(x)
x_max=max(x)
x_min=min(x)
delta=(x_max-x_min)/(1+3.22*log10(n));
m=1+3.22*log10(n);
for i=1:m+1
    f(i)=0;
end
for i=1:n
    for j=1:m+1
        if x(i)>x_min+(j-1)*delta
            if x(i)<x_min+j*delta
                f(j)=f(j)+1;
            end
        end
    end
end
end
for i=1:m+1
    newx(i)=x_min+i*delta;
end
bar(newx,f,'r'); hold on;
```

```

plot(newx,f,'k')
xmed=0;
xx=0;
for i=1:n
    xmed=xmed+x(i);
end
xmed=xmed/n
for i=1:n
    xx=xx+(x(i)-xmed)^2;
end
sigma=sqrt(xx/(n-1))
%axis([0 x_max -1 max(f)])

```

În urma prelucrării, pentru măsurările efectuate în timpul deplasării robocarului pe suprafața uscată, valorile ordonate sunt:

Columns 1 through 7

1.9954 1.9961 1.9971 1.9975 1.9978 1.9982 1.9985

Columns 8 through 14

1.9993 1.9994 1.9994 1.9997 1.9997 1.9999 1.9999

Columns 15 through 21

2.0000 2.0000 2.0000 2.0000 2.0001 2.0001 2.0001

Columns 22 through 28

2.0001 2.0002 2.0003 2.0003 2.0004 2.0004 2.0010

Columns 29 through 35

2.0010 2.0010 2.0011 2.0011 2.0011 2.0011 2.0011

Columns 36 through 42

2.0021 2.0021 2.0022 2.0023 2.0030 2.0032 2.0034

iar:

**x\_max = 2.0034e+003**

**x\_min = 1.9954e+003**

**xmed = 2.0002e+003**

**sigma = 1.7236**

Pentru măsurările efectuate în timpul deplasării robocarului pe suprafața uscată valorile ordonate sunt:

Columns 1 through 7

1.9951 1.9958 1.9961 1.9962 1.9962 1.9968 1.9970

Columns 8 through 14

1.9973 1.9981 1.9982 1.9983 1.9983 1.9984 1.9985

Columns 15 through 21

1.9988 1.9988 1.9988 1.9989 1.9989 1.9990 1.9991

Columns 22 through 28

1.9991	1.9992	1.9993	1.9994	1.9994	1.9995	1.9997
Columns 29 through 35						
1.9999	2.0000	2.0001	2.0002	2.0002	2.0003	2.0004
Columns 36 through 42						
2.0007	2.0008	2.0010	2.0010	2.0011	2.0020	2.0021

iar:

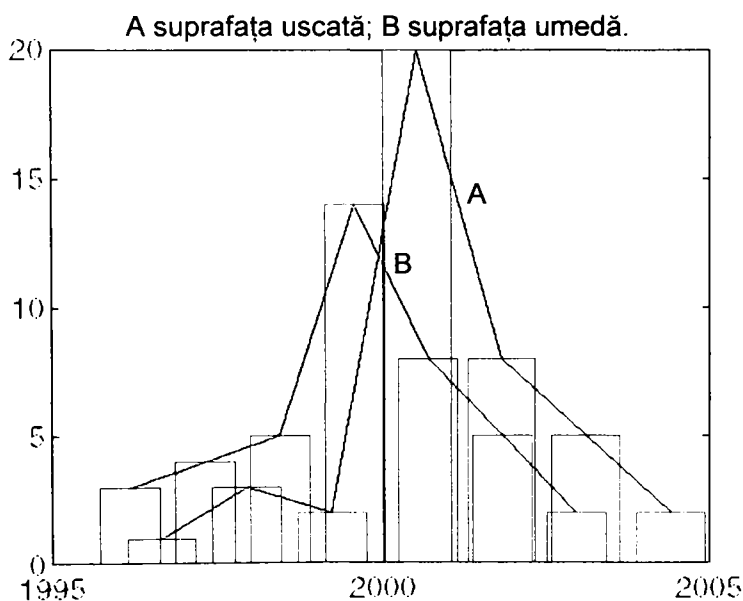
**x\_max = 2.0021e+003**

**x\_min = 1.9951e+003**

**xmed = 1.9990e+003**

**sigma = 1.6566**

În continuare s-a construit histograma pentru cele două cazuri. Aceasta este reprezentată în figura 7.11.

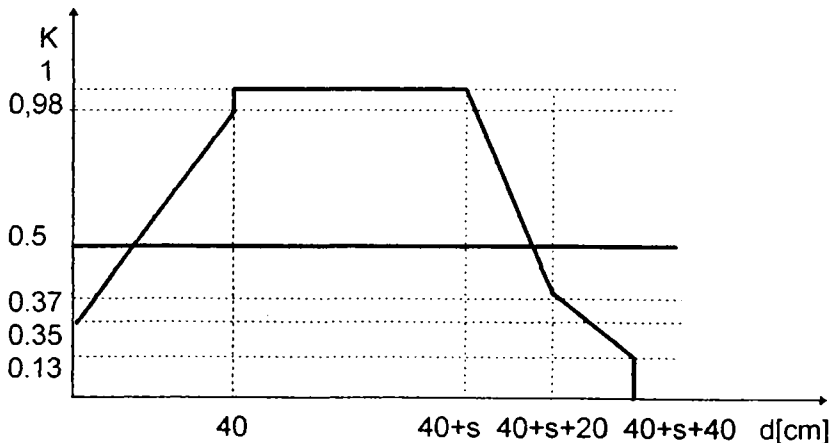


**Figura 7.11.**

Analizând histograma, se observă că distribuția este normală (de tip Gaussian) și poate fi calculată media tuturor rezultatelor conform relației 7.a. Pentru a elimina eventualele greșeli se aplică criteriul  $3\sigma$ . În cazul aplicării acestuia se calculează  $\sigma$  corectat la un număr mic de rezultate conform relației 7.c. Dintre valorile aflate în afara intervalului  $\pm 3\sigma$  se elimină cea mai îndepărtată (dacă este cazul), după care se reia

procedura. În final se obține valoarea cea mai probabilă a spațiului parcurs  $\bar{x}=2000,20$  (pentru pardoseală uscată) și  $\bar{x}=1999,00$  (pentru pardoseală umedă). Din tabelul 7.1 și din histogramă (figura 7.11) se observă că majoritatea rezultatelor se află în jurul valorii 2000,00mm pentru pardoseală uscată, iar pentru pardoseală alunecoasă histograma este decalată spre stânga (1999,00mm). Oprirea mai rapidă are loc datorită faptului că, în timpul demarajului (primii 40cm parcurși) roata alunecă și sunt generate mai multe impulsuri de către TIRO. La frânare, datorită celor două pante diferite (figura 7.12.), nu are loc alunecarea roții motoare. Astfel impulsurile generate de către traductor corespund unui spațiu parcurs de 2000mm, în realitate acesta nefiind parcurs în întregime.

În continuare s-a verificat modul în care variază viteza în funcție de spațiul parcurs. După cum s-a arătat în capitolul 3, pg. 3.16, 3.17, această variație se prezintă ca în figurile 3.7 și 3.8. Experimental s-a constatat necesitatea modificării acestei caracteristici, pentru a se asigura oprirea vehiculului și în condiții defavorabile, așa cum s-a arătat mai sus. Factorul de umplere trebuie modificat conform cu diagrama din figura 7.12.



**Figura 7.12.**

Pornirea motorului și a robocarului are loc atunci când factorul de umplere este  $k_{\text{inițial}} = 1 - 165 / 255 = 0,35$ . După parcurgerea unui spațiu de 40cm, factorul de umplere, modificat la fiecare cm cu valoarea  $\Delta k_1 = 4 / 255$  atinge valoarea  $k_{40} = 1 - (165 / 255 - 40 \times 4 / 255) = 0,98$ . De la această valoare, factorul de umplere devine 1, după care este menținut constant până în momentul în care robocarul mai are de parcurs 40cm până la punctul de oprire. Din acest moment factorul de umplere se modifică în două etape. În prima etapă (primii 20cm) modificarea are loc

cu valoarea  $\Delta k_2 = 8 / 255$  pentru fiecare cm parcurs, iar în a doua etapă cu valoarea  $\Delta k_3 = 3 / 255$ . Astfel, după parcurgerea primilor 20cm s-a atins valoarea  $k_{20} = 0,37$  iar după parcurgerea ultimilor 18cm se obține  $k_{final} = 0,16$ . În acest moment are loc frânarea robocarului prin scurtcircuitarea bornelor motorului (vezi tabelul de adevăr din figura 6.3, pg. 6.6: **EN = H, IN1 = IN2**). Valorile impuse factorului de umplere prin program au fost reținute la fiecare 2cm parcurși de către robocar, într-un spațiu total de 2m. Aceste valori se află în tabelul 7.2, coloana 3.

Tabelul 7.2.

Nr.crt.	d[cm]	k	T[ms]	f[KHz]	v[m/s]	a[m/s <sup>2</sup> ]
1	0	0,3529	-	0	0	0,64
2	2	0,3842	1,13	0,88	0,16	0,64
3	4	0,4156	1,03	0,96	0,17	0,21
4	6	0,4470	0,95	1,04	0,19	0,21
5	8	0,4784	0,88	1,12	0,20	0,21
6	10	0,5098	0,82	1,21	0,22	0,21
7	12	0,5411	0,77	1,29	0,23	0,21
8	14	0,5725	0,72	1,37	0,25	0,21
9	16	0,6039	0,68	1,46	0,26	0,21
10	18	0,6352	0,64	1,54	0,28	0,21
11	20	0,6666	0,61	1,62	0,29	0,21
12	22	0,6980	0,58	1,70	0,31	0,21
13	24	0,7294	0,55	1,79	0,32	0,21
14	26	0,7607	0,53	1,87	0,34	0,21
15	28	0,7921	0,51	1,95	0,36	0,21
16	30	0,8235	0,49	2,03	0,37	0,21
17	32	0,8549	0,47	2,12	0,39	0,21
18	34	0,8862	0,45	2,20	0,40	0,21
19	36	0,9176	0,43	2,28	0,42	0,21
20	38	0,9490	0,42	2,37	0,43	0,21
21	39	0,9647	0,41	2,41	0,44	0,89
22	40	1,0000	0,40	2,45	0,45	0,89
23	40,5	1,0000	0,40	2,49	0,46	0,89
24	160	1,0000	0,40	2,49	0,46	0
25	160,5	0,9843	0,40	2,49	0,46	-1,76
26	161	0,9529	0,40	2,46	0,45	-1,76
27	162	0,9372	0,42	2,38	0,43	-0,48
28	164	0,8745	0,44	2,22	0,40	-0,48
29	166	0,8117	0,48	2,06	0,37	-0,48
30	168	0,7490	0,52	1,90	0,35	-0,48
31	170	0,6862	0,57	1,74	0,32	-0,48

Nr.crt.	d[cm]	k	T[ms]	f[KHz]	v[m/s]	a[m/s <sup>2</sup> ]
32	172	0,6235	0,62	1,58	0,29	-0,48
33	174	0,5607	0,69	1,42	0,26	-0,48
34	176	0,4980	0,78	1,27	0,23	-0,48
35	178	0,4352	0,89	1,11	0,20	-0,48
36	180	0,3725	1,04	0,95	0,17	-0,48
37	181	0,3607	1,07	0,93	0,17	-0,082
38	182	0,3490	1,10	0,90	0,16	-0,064
39	184	0,3254	1,18	0,84	0,15	-0,064
40	186	0,3019	1,27	0,78	0,14	-0,064
41	188	0,2784	1,37	0,72	0,13	-0,064
42	190	0,2549	1,49	0,66	0,12	-0,064
43	192	0,2313	1,64	0,60	0,11	-0,064
44	194	0,2078	1,82	0,54	0,10	-0,064
45	196	0,1843	2,03	0,49	0,09	-0,064
46	198	0,1607	2,31	0,43	0,07	-0,064
47	199	0,0803	2,48	0,40	0,07	-0,18
48	200	0	2,91	0,34	0,06	-0,18
49	200,3	0	-	0	0	-0,18

Prin utilizarea unui programului MATLAB, a fost reprezentată în figura 7.13 dependența dintre factorul de umplere a impulsurilor de comandă a EEMCC,  $k$  și spațiul parcurs de către robocar

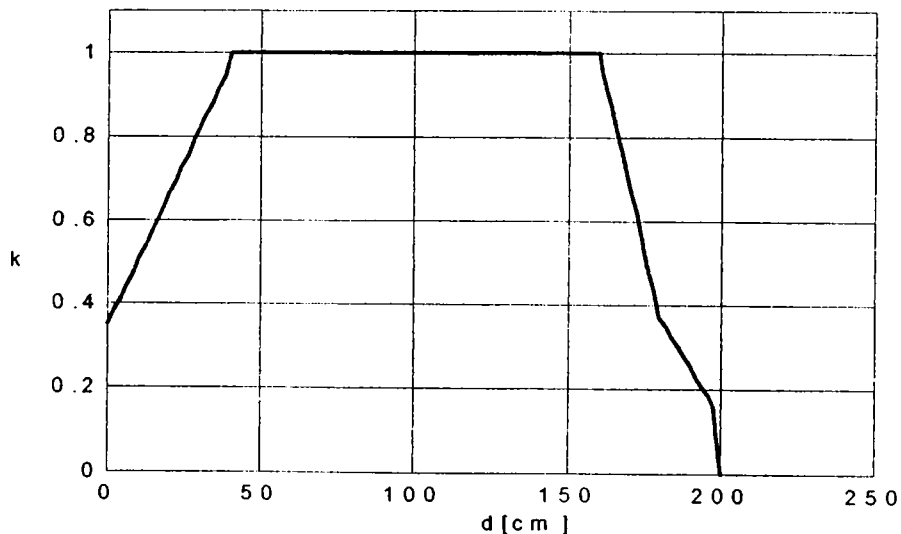


Figura 7.13.

Din calculul prezentat în capitolul 5, pg. 5.6, rezultă că valoarea maximă a vitezei este  $v_{\max}=0,46$  m/s. Pentru a se evidenția modul în care variază viteza în funcție de spațiul parcurs, este necesar ca aceasta să fie măsurată în regim dinamic. Măsurarea s-a efectuat indirect, prin măsurarea perioadei impulsurilor furnizate de către TIRO. S-a preferat măsurarea perioadei și nu a frecvenței pentru că aceasta se desfășoară într-un timp redus. Este necesară o măsurare rapidă deoarece robocarul se deplasează, având loc o modificare continuă a vitezei, iar pentru ridicarea experimentală a caracteristicii trebuie prelevate cât mai multe eșantioane. Eșantioanele s-au obținut prin măsurarea perioadei la fiecare 2cm parcurși de către robocar, iar pe anumite segmente de traiectorie măsurarea s-a efectuat și la intervale de 0.5cm. Astfel s-a obținut o caracteristică care reflectă fidel dinamica mișcării.

Programele prezentate în capitolul 3 realizează măsurarea spațiului parcurs. Pentru efectuarea experimentelor, acestea au fost completate cu următoarea secvență de program, fiind astfel posibilă măsurarea perioadei.

Programul începe prin declarea unei constante, **nr\_imp**. Ea va defini numărul de impulsuri care determină intervalul dintre două măsurări ale perioadei acestora. În continuare se inițializează întreruperea folosită (întreruperea externă **EX0**) și se determină nivelulul de prioritate (nivel mare). Programul intră într-o buclă infinită (**loop**), din care va ieși doar în momentul setării fanionului **F0**.

Impulsurile furnizate de către **TIRO** sunt interpretate drept cereri de întrerupere externă. Prin tratarea acestei întreruperi se va incrementa un contor (implementat cu ajutorul registrului **R7**).

La atingerea valorii fixate prin **nr\_imp** se setează fanionul **F0**. La revenirea în programul principal se va ieși din bucla **loop** (dacă **F0** este setat) și se incrementează conținutul registrului **R0** până la apariția următorului impuls furnizat de către **TIRO**. În acest moment **F0** va fi din nou șters, iar programul va reintra în bucla **loop**.

Durata de execuție a instrucțiunilor cuprinse între eticheta **mas2** și instrucțiunea **jmp mas2** inclusiv, este  $10\mu\text{s}$ , ceea ce duce la o perioadă maximă a semnalului furnizat de către **TIRO**, de  $2560\mu\text{s}$ .

Înainte de revenirea din subrutina pentru contorizarea duratei impulsurilor, conținutul registrului **R0** este transferat portului **P1**.

Programul utilizat pentru măsurarea perioadei este prezentat în continuare.

```
nr_imp equ 64h
org 0200h
    mov r7,#00h
    clr f0
    mov ie,#81h
    mov ip,#01h
    mov tcon,#01
    mov p1,#00h
loop: jnb f0,loop

mas2:    jnb f0,mas3
        inc r0
        nop
        nop
        cjne r0,#00h,mas2
        jmp mas2

mas3:    mov p1,r0
        jmp loop

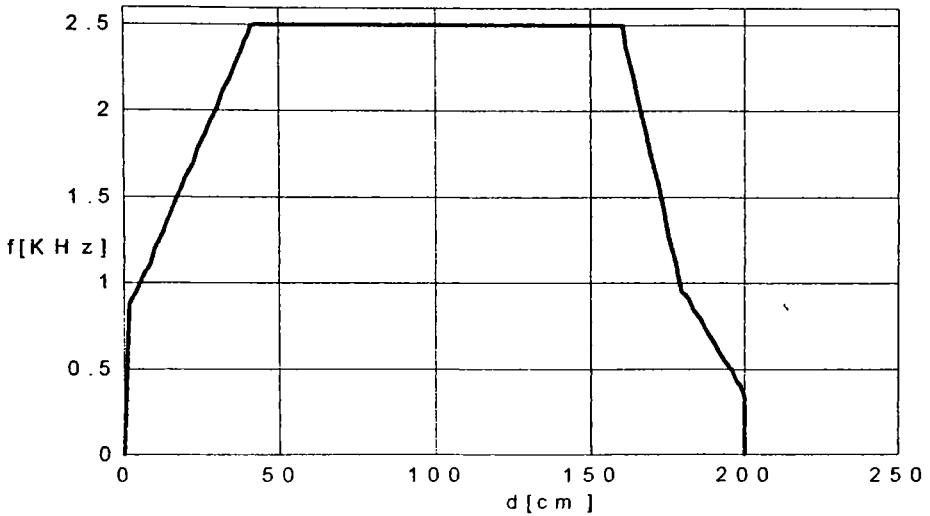
org 0003h
    clr f0
    inc r7
    cjne r7,#nr_imp,mas1
    mov r7,#00h
    setb f0
mas1:    reti
end
```

Rezultatele măsurării perioadei impulsurilor, efectuate cu ajutorul programului descris se află în tabelul 7.2, coloana 4.

În coloana 5 a aceluiași tabel se află valorile corespunzătoare ale frecvenței impulsurilor, obținute prin calcul, utilizând valoarea măsurată a perioadei, iar caracteristica obținută pe baza acestor rezultate este prezentată în figura 7.14.

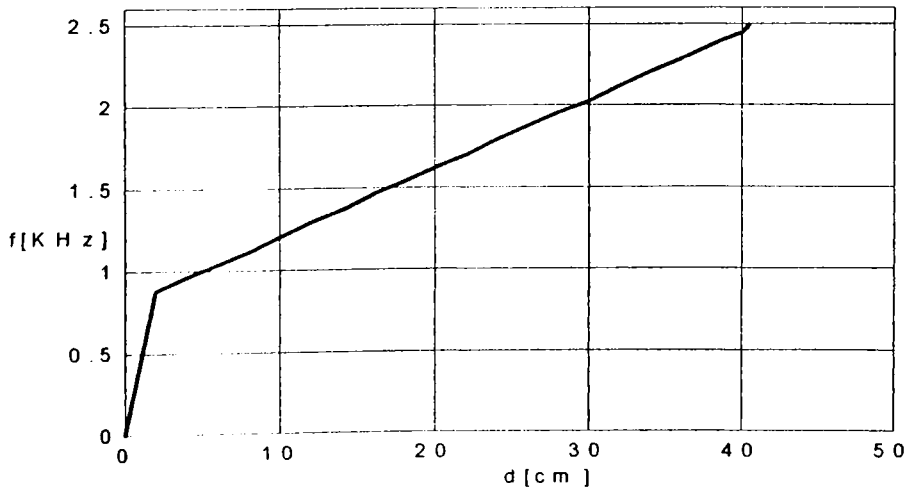
S-a reprezentat variația frecvenței, cu toate că a fost măsurată perioada impulsurilor, deoarece frecvența este direct proporțională cu viteza, iar caracteristica obținută este asemănătoare cu cea care reflectă variația vitezei.



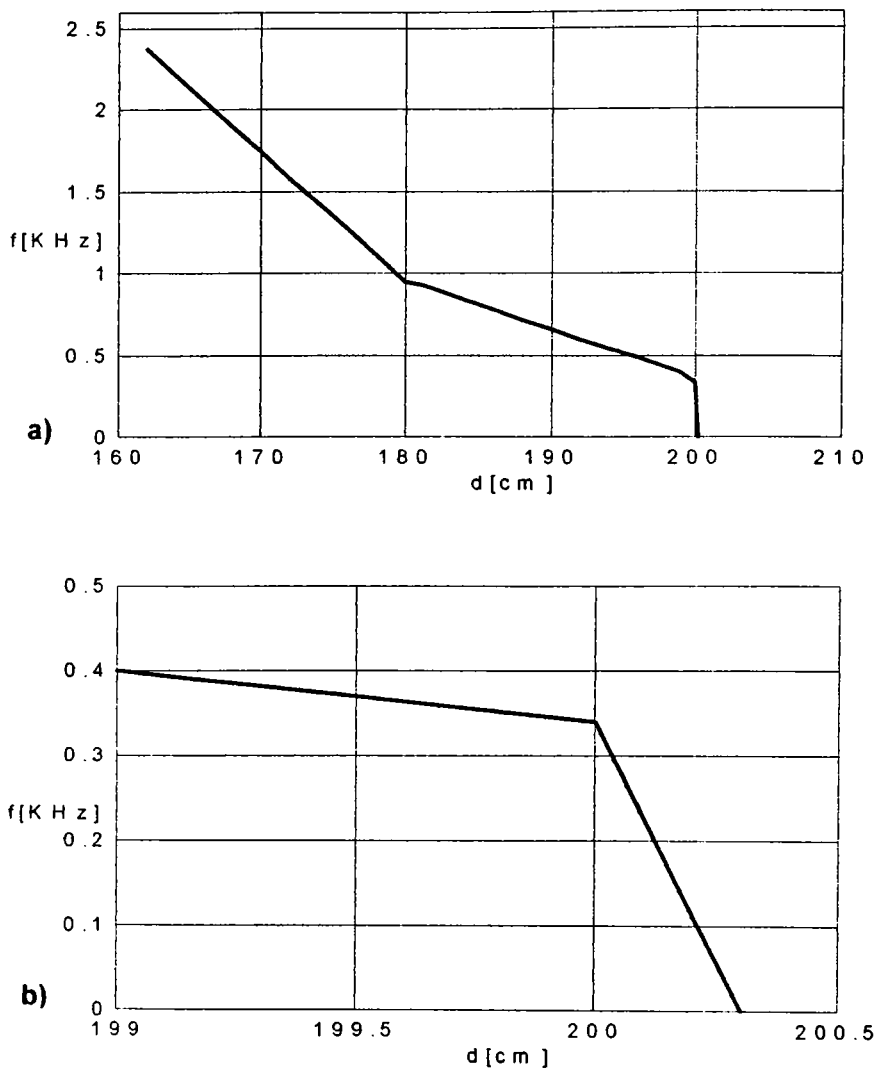


**Figura 7.14.**

În figura 7.15 este prezentată în detaliu prima parte a caracteristicii (primii 40cm parcurși), iar în figura 7.16 se poate observa detaliat modul de variație a frecvenței în ultimii 40cm parcurși (a) și până la oprirea robocarului (b).



**Figura 7.15.**

**Figura 7.16.**

Fiind măsurată perioada impulsurilor, se poate calcula viteza vehiculului. De asemenea se poate reprezenta variația acesteia în funcție de spațiul parcurs de către robocar.

Viteza de deplasare a robocarului este:

$$v[m/s] = \frac{2\pi}{T_R[s]} \cdot \frac{D[mm]}{2 \cdot 1000} = \frac{2\pi}{T_m} \cdot i \cdot \frac{D}{2 \cdot 1000} = \frac{2\pi}{n_1 \cdot T_T} \cdot i \cdot \frac{D}{2 \cdot 1000} =$$

$$= \frac{2\pi \cdot i}{n_1 \cdot T_T} \cdot \frac{D[mm]}{2}$$

rezultă:

$$v[m/s] = \frac{\pi \cdot i \cdot D[mm]}{n_1 \cdot T_T[ms]}$$

S-a notat cu:

- v** - viteza de deplasare a robocarului;
- T<sub>R</sub>** - perioada mișcării de rotație a roții motoare;
- T<sub>m</sub>** - perioada mișcării de rotație a motorului;
- T<sub>T</sub>** - perioada impulsurilor furnizate de către traductorul inductiv;
- D** - diametrul roții;
- n<sub>1</sub>** - numărul de impulsuri furnizate de către traductor la o rotație completă;
- i** - raportul de transmitere.

Cu ajutorul relației de mai sus s-a calculat viteza de deplasare a robocarului în fiecare punct în care s-a măsurat perioada impulsurilor. Rezultatele calculelor se află în tabelul 7.2, coloana 6. Pe baza acestor rezultate au fost ridicate caracteristicile din figurile 7.17, 7.18, 7.19 și 7.20.

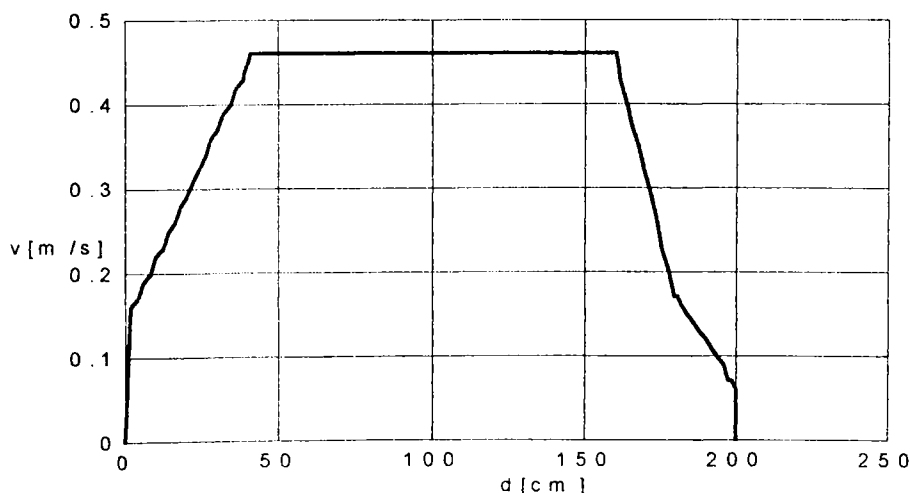


Figura 7.17.

În figura 7.17 este prezentată caracteristica pentru un spațiu total parcurs de 200cm. Variația vitezei pentru primul interval este ilustrată în detaliu în figura 7.18.

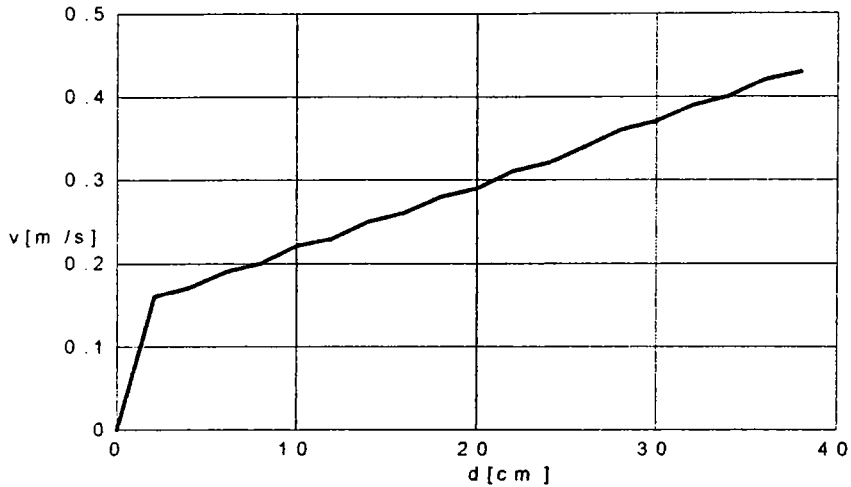


Figura 7.18.

Pentru ultimul interval parcurs s-a reprezentat caracteristica din figura 7.19, iar pentru spațiul din vecinătatea punctului final reprezentarea este aceea din figura 7.20.

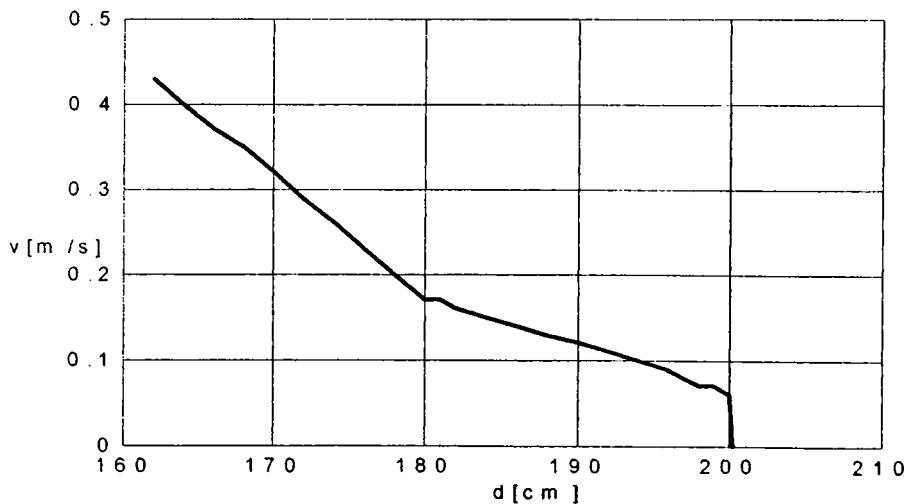


Figura 7.19.

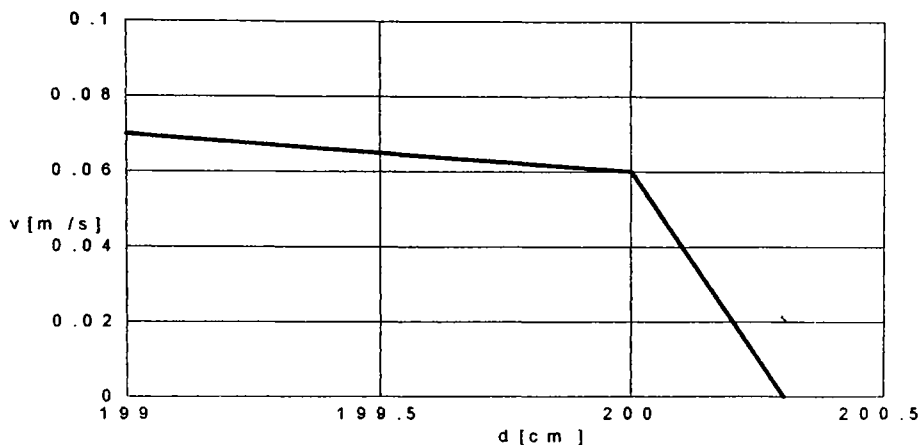


Figura 7.20.

Din tabelul 7.2 și din figura 7.13 se observă că factorul de umplere a impulsurilor care constituie semnalul de comandă al EEMCC,  $k$ , devine egal cu 0 în momentul în care robocarul a atins punctul final (200cm în exemplul considerat). În acest moment sunt scurtcircuitate bornele motorului. Robocarul s-a mai deplasat însă din inerție 0,3cm, cu toate că motorul este frânat (figura 7.20).

Pe baza valorilor calculate ale vitezei a fost calculată accelerația vehiculului pentru opt intervale (tabelul 7.2, coloana 7). Pentru calculul accelerației s-a utilizat formula lui Galilei ( $v^2 = v_0^2 + 2as$ ). În figura 7.21 este reprezentată variația accelerației în primul interval, iar în figura 7.22 se prezintă variația acesteia în ultimul interval.

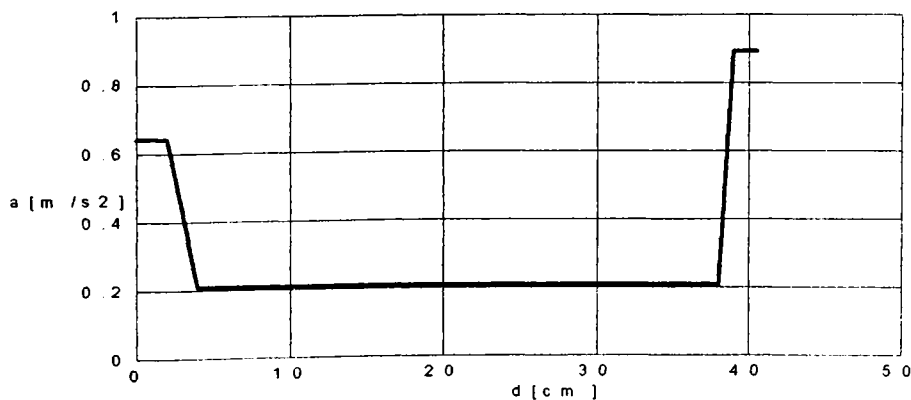


Figura 7.21.

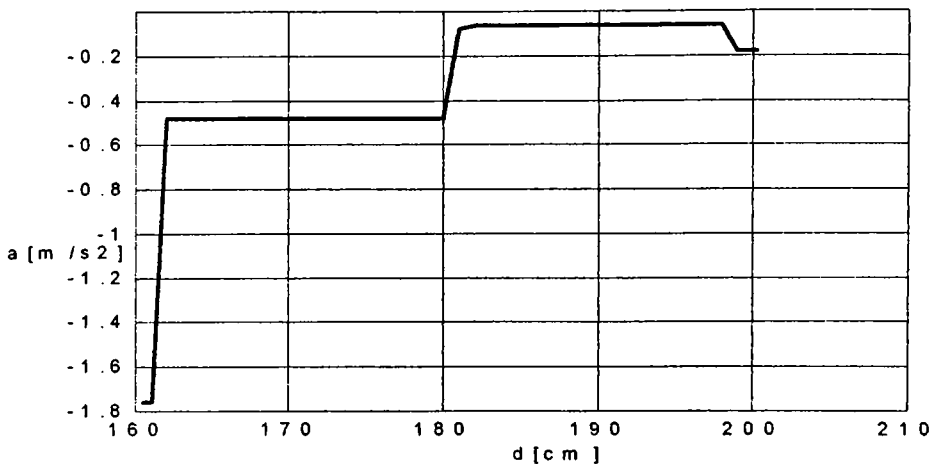


Figura 7.22.

În figura 7.23 sunt reprezentate toate cele patru caracteristici împreună, pentru a fi posibilă compararea lor. Se poate observa existența unui decalaj între caracteristica care reprezintă variația vitezei ( $v$ ) și aceea care reprezintă variația factorului de umplere ( $k$ ), datorită răspunsului întregului sistem.

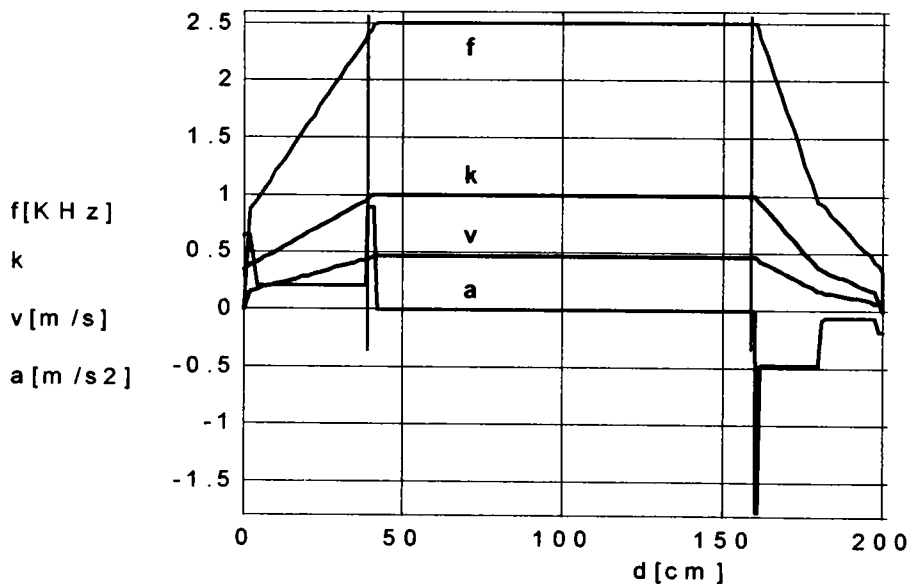


Figura 7.23.

---

Programul utilizat pentru trasarea caracteristicilor prezentate a fost scris în MATLAB și este redat în continuare.

```
dist1=0:2:38;
dist2=[39 40 40.5];
dist3=42:2:160;
dist4=[160.5 161];
dist5=162:2:180;
dist6=[181];
dist7=182:2:198;
dist8=[199 200 200.3];

dist=[dist1 dist2 dist3 dist4 dist5 dist6 dist7 dist8];

kappa1=[3529 3843 4156 4470 4784 5098 5411 5725 6039 6352 6666
6980 7294 7607 7921 8235 8549 8862 9176 9490];
kappa2=[9647 10000 10000];
kappa3=ones(1,60)*10000;
kappa4=[9843 9529];
kappa5=[9372 8745 8117 7490 6862 6235 5607 4980 4352 3725];
kappa6=[3607];
kappa7=[3490 3254 3019 2784 2549 2313 2078 1843 1607];
kappa8=[803 0 0];

kappa=[kappa1 kappa2 kappa3 kappa4 kappa5 kappa6 kappa7 kappa8];
kappagr=kappa/10000;
kappagr1=kappa1/10000;

f1=[0 88 96 104 112 121 129 137 146 154 162 170 179 187 195 203 212
220 228 237];
f2=[241 245 249];
f3=ones(1,60)*250;
f4=[250 246];
f5=[238 222 206 190 174 158 142 127 111 95];
f6=[93];
f7=[90 84 78 72 66 60 54 49 43];
f8=[40 34 0];

f=[f1 f2 f3 f4 f5 f6 f7 f8];
fgr=f/100;

v1=[0 16 17 19 20 22 23 25 26 28 29 31 32 34 36 37 39 40 42 43];
v2=[44 45 46];
```

---

```
v3=ones(1,60)*46;
v4=[46 45];
v5=[43 40 37 35 32 29 26 23 20 17];
v6=[17];
v7=[16 15 14 13 12 11 10 9 7];
v8=[7 6 0];

v=[v1 v2 v3 v4 v5 v6 v7 v8];
vgr=v/100;

a11=ones(1,2)*0.64;
a12=ones(1,18)*0.21;
a2=ones(1,3)*0.89;
a3=ones(1,60)*0;
a4=ones(1,2)*(-1.76);
a5=ones(1,10)*(-0.48);
a6=ones(1,1)*(-0.082);
a7=ones(1,9)*(-0.064);
a8=ones(1,3)*(-0.18);

a=[a11 a12 a2 a3 a4 a5 a6 a7 a8];

h1=plot(dist,kappagr,'k');
hold on
h2=plot(dist,fgr,'k');
hold on
h3=plot(dist,vgr,'k');
hold on
h4=plot(dist,a,'k');
hold on
grid
set(h1,'LineWidth',2)
set(h2,'LineWidth',2)
set(h3,'LineWidth',2)
set(h4,'LineWidth',2)
set(gca,'YLimMode','manual','YLim',[-1.8 2.6])

xlabel('d[cm]');
ylabel('f[KHz], k, v[m/s],a[m/s2]')
```



Apoi s-a verificat funcționarea programului pentru comanda elementului de execuție al mecanismului de virare. Poziționarea roții directoare a avut loc cu o precizie de  $\pm 0,15^\circ$ , atunci când mecanismul de virare a fost rotit cu  $120^\circ$  (figura 7.24). Din detaliul aflat în partea dreaptă jos a figurii se observă poziția indicatorului fixat de axul roții directoare, în raport cu cifra 60 (grade), aflată pe raportorul utilizat pentru măsurarea unghiului de virare.

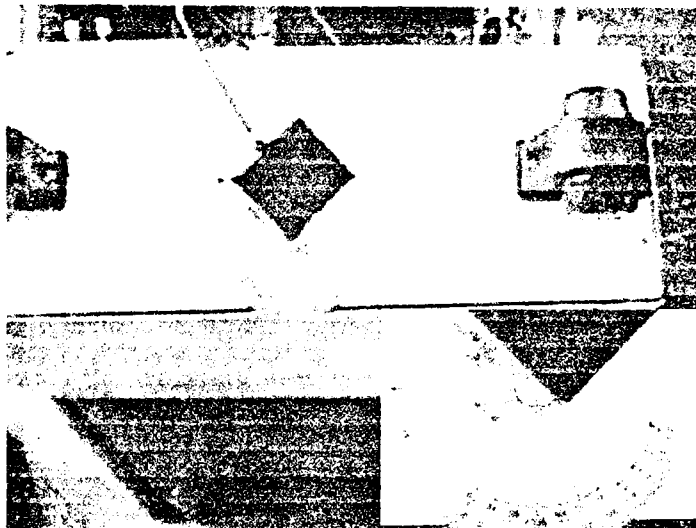


Figura 7.24.

S-a verificat funcționarea prin comandă manuală. Sistemul de dezvoltare a generat toate comenzile impuse, iar acestea au fost transmise corect elementelor de execuție corespunzătoare.

Apoi a fost verificat programul pentru inițializarea sistemului de virare și s-au stabilit limitele de rotire a mecanismului de virare la  $\pm 60^\circ$ .

Procedeul de determinare a poziției inițiale și finale a mecanismului de virare în cazul inițializării acestuia este asemănător cu cel descris la mecanismul de tracțiune. Rezultatele măsurărilor efectuate sunt concludente pentru aprecierea performanțelor acestui mecanism și în cazul virării cu un unghi oarecare.

Pentru efectuarea măsurării s-au amplasat 2 comparatoare: unul în poziția inițială a mecanismului de virare și unul în punctul în care are loc oprirea acestuia (direcția de mers rectiliniu). De axul vertical al roții

directoare s-a fixat o tijă de lungime  $R=100\text{mm}$ , care acționează asupra comparatoarelor.

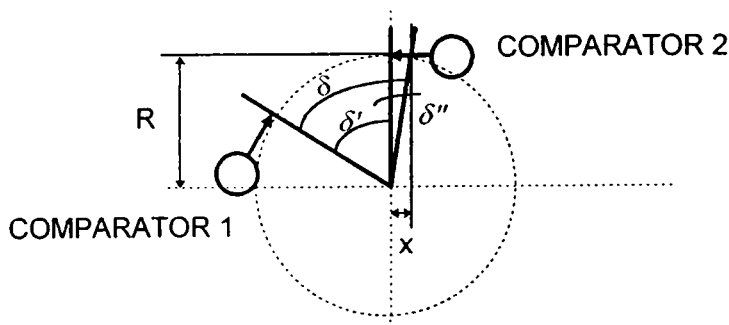


Figura 7.25.

Din figura 7.25 se observă că, dacă se măsoară deplasarea  $x$  cu ajutorul comparatorului și se cunoaște  $R$  se poate determina unghiul  $\delta''$ :

$$\operatorname{tg} \delta'' = \frac{x}{R} \quad \delta'' = \operatorname{arctg} \frac{x}{R}$$

Unghiul total  $\delta = \delta' + \delta''$ , dacă se depășește poziția prescrisă, iar  $\delta = \delta' - \delta''$  dacă oprirea are loc înaintea acestei poziții.

Valorile unghiului  $\delta$ , rezultate în urma unui număr  $n=42$  măsurări, sunt prezentate în tabelul 7.3. Acestea s-au calculat în urma măsurării, utilizând relația de mai sus.

Tabelul 7.3.

Nr.	$\delta$	Nr.	$\delta$	Nr.	$\delta$
1	74,9405	12	75,6506	23	75,0111
2	74,9302	13	74,8702	24	75,0252
3	74,8901	14	74,8918	25	75,0316
4	75,0002	15	75,0020	26	74,1315
5	75,1585	16	75,7503	27	74,7229
6	74,9928	17	74,9612	28	74,9213
7	75,2131	18	74,9990	29	74,8707
8	74,8347	19	75,0302	30	74,6336
9	75,4644	20	75,0605	31	75,5265
10	74,9403	21	74,9636	32	74,6133
11	75,0313	22	74,9876	33	74,9517

Nr.	$\delta$	Nr.	$\delta$	Nr.	$\delta$
34	74,8117	37	75,0101	40	75,1213
35	74,8805	38	75,0010	41	75,3434
36	75,0515	39	75,1761	42	75,3938

Valorile din tabelul 7.3 au fost ordonate crescător:

Columns 1 through 7

74.1315 74.6133 74.6336 74.7229 74.8117 74.8347 74.8702

Columns 8 through 14

74.8707 74.8805 74.8901 74.9213 74.9302 74.9403 74.9405

Columns 15 through 21

74.9517 74.9612 74.9636 74.9818 74.9876 74.9928 74.9990

Columns 22 through 28

75.0002 75.0010 75.0020 75.0101 75.0111 75.0252 75.0302

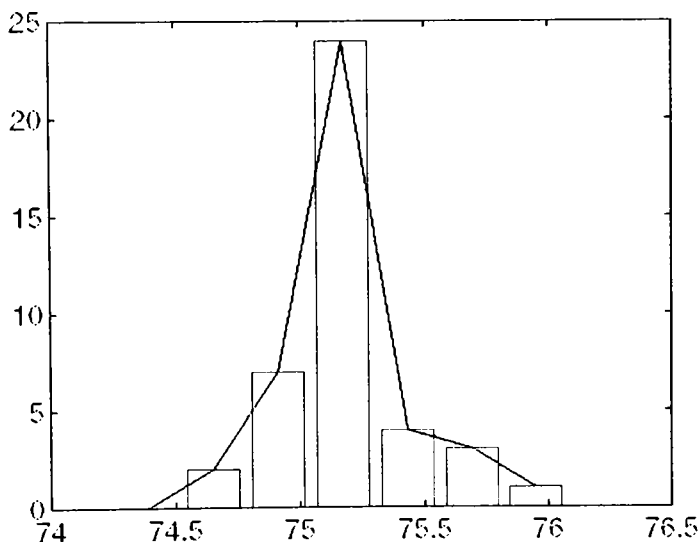
Columns 29 through 35

75.0313 75.0316 75.0515 75.0605 75.1213 75.1585 75.1761

Columns 36 through 42

75.2131 75.3434 75.3938 75.4644 75.5265 75.6506 75.7503

În continuare s-a construit histograma, prezentată în figura 7.26.



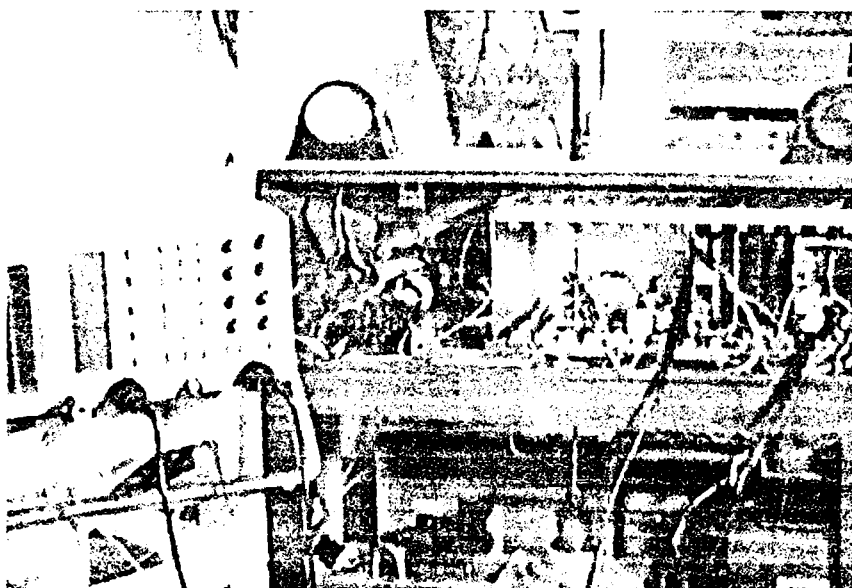
**Figura 7.26.**

Prin aplicarea procedurii prezentate la mecanismul de tracțiune s-au calculat:

**x\_max = 75.7503**  
**x\_min = 74.1315**  
**xmed = 75.0210**  
**sigma = 0.2759**

Se observă că o singură valoare depășește intervalul  $\pm 3\sigma$ . Se poate considera că aceasta este datorată unei comenzi greșite, unui impuls parazit, etc. Nefiind alte greșeli, rezultă că motorul pas cu pas nu pierde pași și mecanismul de virare nu are jocuri. Precizia poziționării este foarte bună.

Ultimele verificări ale echipamentelor electronice au fost efectuate după amplasarea acestora într-un compartiment special destinat, care face parte din construcția mecanică a robocarului (figura 7.27).



**Figura 7.27.**

În urma verificărilor și a măsurărilor efectuate s-a constatat că, atât elementele de execuție, cât și sistemul de conducere funcționează corect, iar cu ajutorul programelor realizate pot fi executate mișcările dorite.

Pe baza acestor considerente se poate afirma că robocarul construit corespunde obiectivului propus și poate fi încadrat într-un sistem de transport automat care face parte dintr-un hipersistem **CIM**.

## 7.4. Calibrarea camerei video.

În primul rând s-a efectuat calibrarea camerei video CCD utilizate în cadrul experimentelor. Camera are, conform datelor de catalog dimensiunile unei celule  $p_x=9,6 \cdot 10^{-3}$  mm și  $p_y=8,4 \cdot 10^{-3}$  mm. Aceasta s-a fixat pe o masă de poziționare (figura 7.28). Masa oferă posibilitatea deplasării pe axa z a camerei și pe axa x, respectiv y, a punctului vizat. Punctul vizat, marcat în sistemul de axe de coordonate xOy, poate fi deplasat cu o precizie de 0,01mm.

Cunoscând datele camerei și utilizând programul prezentat în paragraful 4.2.12, determinările s-au efectuat prin parcurgerea următoarelor etape:

- s-a limitat domeniul în care este prelucrată imaginea la  $x_{\max}=320$  pixeli și  $y_{\max}= 240$  pixeli. Orice element de imagine care nu se încadrează în acest domeniu nu este luat în considerare, deoarece este necesară eliminarea erorilor introduse de către sistemul optic.

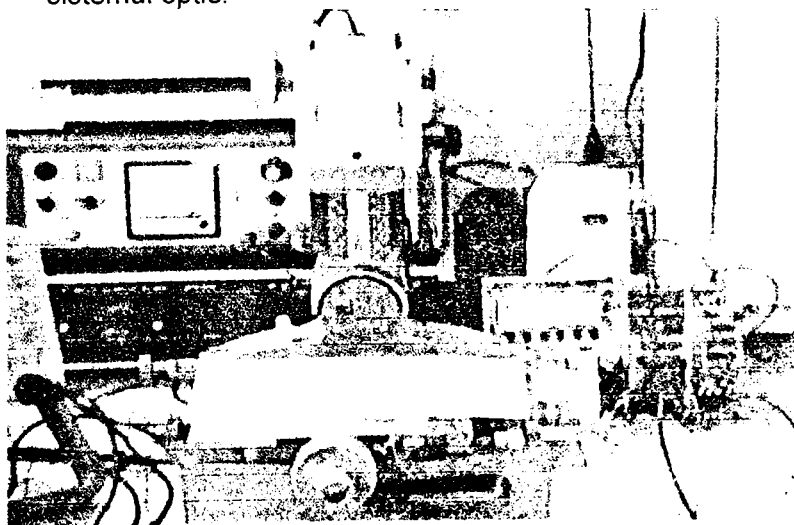


Figura 7.28.

- s-a deplasat punctul M până când coordonatele sale, citite pe ecranul monitorului au fost  $x_{\max}/2$  și  $y_{\max}/2$ .
- s-a efectuat o deplasare  $\Delta x = 5\text{mm}$  pe axa x, iar apoi  $\Delta z = 5\text{mm}$  pe axa z (figura 7.29).
- s-a citit  $n_x = 10$  pixeli și  $n'_x = 9$  pixeli. Folosind aceste valori se pot scrie următoarele relații:

$$\frac{b}{d} = \frac{n_x \cdot p_x}{\Delta x} \quad (7.1)$$

$$\frac{b}{d + \Delta z} = \frac{n'_x \cdot p_x}{\Delta x} \quad (7.2)$$

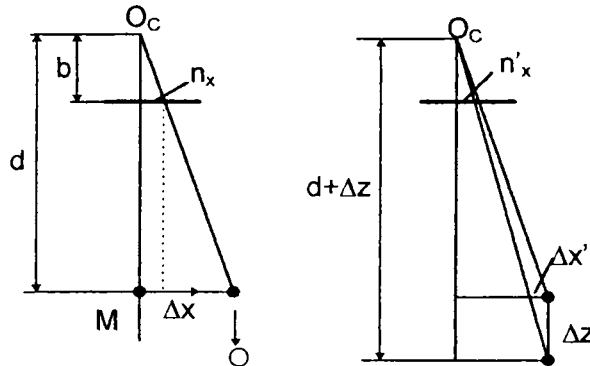


Figura 7.29.

$$\frac{b}{d} = \frac{n'_x \cdot p_x}{\Delta x - \Delta x'} \quad (7.3)$$

$$\frac{p_x}{\Delta x'} = \frac{d}{d} \quad (7.4)$$

$$\frac{d}{d + \Delta z} = \frac{\Delta x - \Delta x'}{\Delta x} \quad (7.5)$$

din (7.1) și (7.4) rezultă:

$$\frac{10 \cdot p_x}{5} = \frac{p_x}{\Delta x'}$$

$$2 \cdot p_x = \frac{p_x}{\Delta x'} \Rightarrow \Delta x' = \frac{1}{2} = 0,5 \text{ mm}$$

în (7.5)

$$\frac{d}{d + 5} = \frac{5 - 0,5}{5} \Rightarrow d = 45 \text{ mm}$$

Înlocuind în (7.1) rezultă:

$$\frac{b}{45} = 2 \cdot p_x \Rightarrow b = 90 p_x = 90 \cdot 96 \cdot 10^{-3} \text{ mm} = 0,864 \text{ mm}$$

Deci, distanța de la centrul optic  $O_C$  la planul circuitului integrat CCD este:

$$b = 0,864 \text{ mm},$$

Această valoare este necesară pentru calibrarea camerei video și se determină o singură dată. În continuare s-au realizat deplasări pe axele Ox și Oy ale punctelor marcate, conform datelor din tabelele 7.4 și 7.5 și s-au citit valorile corespunzătoare ale coordonatelor în pixeli. Aceste valori sunt reprezentate în tabelele 7.4 și 7.5.

**Tabelul 7.4.**

Nr. crt.	Deplas. pct. țintă pe Y[mm]	X*	Y*		Deplas. pct. țintă pe X[mm]	X*		Y*
			Stânga	Dreapta		Stânga	Dreapta	
0	1	2	3	4	5	6	7	8
1	0,5	159/160	135	134	0,5	160	159	120/121
2	1	159/160	134	133	1	159	158	120/121
3	1,5	159/160	133	132	1,5	158	157	120/121
4	2	159/160	132	131	2	157	156	120/121
5	2,5	159/160	131	130	2,5	156	155	120/121
6	3	159/160	130	129	3	155	154	120/121
7	3,5	159/160	129	128	3,5	154	153	120/121
8	4	159/160	128	127	4	153	152	120/121
9	4,5	159/160	127	126	4,5	152	151	120/121
10	5	159/160	126	125	5	150	149	120/121
11	5,5	159/160	125	124	5,5	149	148	120/121
12	6	159/160	124	123	6	148	147	120/121
13	6,5	159/160	123	122	6,5	147	146	120/121
14	7	159/160	122	121	7	146	145	120/121
15	7,5	159/160	121	120	7,5	145	144	120/121
16	8	159/160	120	119	8	144	143	120/121
17	8,5	159/160	119	118	8,5	143	142	120/121
18	9	159/160	118	117	9	142	141	120/121
19	9,5	159/160	117	116	9,5	141	140	120/121
20	10	159/160	116	115	10	140	139	120/121

- X\*, Y\* = numărul de pixeli corespunzător extremităților punctului vizat pe axa X, respectiv Y.

**Tabelul 7.5.**

Nr. crt.	Nr. pixeli pe X	Nr. pixeli pe Y	Deplas. pct. țintă pe X[mm]	Nr. pixeli pe X	Nr. pixeli pe Y	Deplas. pct. țintă pe Y[mm]
0	1	2	3	4	5	6
1	133/132	164/167	START	133/132	164/163	START
2	136/135	164/167	1,5	133/132	161/160	1,5
3	139/138	164/167	3	133/132	158/157	3
4	142/141	164/167	4,5	133/132	155/154	4,5
5	145/144	164/167	6	133/132	152/151	6

6	148/147	164/167	7,5	133/132	149/148	7,5
7	151/150	164/167	9	133/132	146/145	9
8	154/153	164/167	10,5	133/132	143/142	10,5
9	157/156	164/167	12	133/132	140/139	12
10	160/159	164/167	13,5	133/132	137/136	13,5
11	163/162	164/167	15	133/132	134/133	15
12	166/165	164/167	16,5	133/132	131/130	16,5
13	169/168	164/167	18	133/132	128/127	18
14	172/171	164/167	19,5	133/132	125/124	19,5
15	175/174	164/167	21	133/132	122/121	21
16	178/177	164/167	22,5	133/132	119/118	22,5
17	181/180	164/167	24	133/132	116/115	24
18	184/183	164/167	25,5	133/132	113/112	25,5
19	187/186	164/167	27	133/132	110/109	27
20	190/189	164/167	28,5	133/132	107/106	28,5
21	193/192	164/167	30	133/132	104/103	30
22	196/195	164/167	31,5	133/132	101/100	31,5
23	199/198	164/167	33	133/132	98/97	33
24	202/201	164/167	34,5	133/132	95/94	34,5
25	205/204	164/167	36	133/132	92/91	36
26	208/207	164/167	37,5	133/132	89/88	37,5

Pe baza determinărilor efectuate se observă că, menținând intervalul de deplasare, pentru un pas constant se obține o variație constantă a numărului de pixeli corespunzător punctului țintă.

**Rezultatele măsurărilor efectuate demonstrează că determinarea situației țintei poate avea loc oriunde s-ar afla aceasta în câmpul vizual al camerei video.**

### **7.5. Determinarea poziției unui punct în spațiul 2D cu ajutorul camerei video și simularea deplasării robocarului pe o traiectorie oarecare.**

Experimentele descrise în continuare au fost realizate cu scopul de a verifica corectitudinea modelului considerat pentru camera CCD și a metodei de calibrare popuse.

A fost simulată deplasarea unui robocar pe o traiectorie oarecare și s-a verificat posibilitatea determinării situației acestuia prin calculul coordonatelor a două puncte. Experimentul s-a desfășurat după cum urmează.

S-a trasat pe un fond alb o traiectorie oarecare, reprezentată în figura 7.30.



• •  
A B

**Figura 7.30.**

Nivelul de negru al acesteia și al inscripționărilor a fost astfel ales încât să se afle deasupra pragului de discriminare (capitolul 4, pag. 4.47) și să fie interpretat drept alb, pentru a nu perturba măsurarea. Nivelul de negru al punctelor A și B se află sub prag, astfel încât acestea sunt considerate negre. Poziția acestor puncte se va determina în continuare.

Robocarul, simulat prin punctele A și B, a fost deplasat de-a lungul traiectoriei, la intervale aleatorii și au fost citite coordonatele acestor puncte. Rezultatele citirii sunt prezentate în tabelul 7.6.

**Tabelul 7.6.**

Numărul punctului de măsură	Numărul de pixeli corespunzător punctelor țintă A și B				
	Stânga	Dreapta	Stânga	Dreapta	
0	1	2	3	4	5
1	121/132	188/189			
2	98/109	185/187			
3	81/91	177/183			
4	74/81	168/176			
5	69/70	151/152			
6	68/69	126/126	68/115	68/126	
7	66/68	107/107	66/67	96/108	
8	69/74	82/72	73/71	69/82	
9	79/88	64/56	88/55	79/64	
10	103/115	51/50			
11	125/137	49/48			
12	151/163	50/54			
13	171/177	65/75	179/91	179/103	

14	179/180	103/103			
15	184/191	120/129			
16	204/217	137/137			
17	227/237	140/146			
18	239/241	169/159			
19	228/236	184/177			
20	192/203	188/188			
21	157/169	189/189			
22	121/133	188/188			STOP=START

Pentru interpretarea rezultatelor s-a realizat un program, cu ajutorul căruia se reprezintă coordonatele determinate ale punctelor și se trasează traiectoria robocarului.

Programul **repréz.cpp** realizează reprezentarea în plan a punctelor ale căror coordonate au fost stabilite de către programul de prelucrare a imaginii. Organigrama programului este prezentată în figura 7.31.

Coordonatele acestor puncte sunt măsurate în pixeli și sunt stocate în fișierul **coord.num**.

Fișierul începe printr-un caracter **SPACE** urmat de coordonata pe axa Ox a primului punct, apoi coordonata pe axa Oy a aceluiași punct, continuând prin coordonatele celorlalte puncte în aceeași ordine (coordonata x, coordonata y). Coordonatele sunt separate între ele prin două caractere **SPACE**. Ultimul șir de caractere din fișier este urmat de către un singur caracter **SPACE**, respectiv **EOF** (End Of File). Fișierul **coord.num** se va afla în același director cu executabilul (**repréz.exe**). Fișierul **coord.num** se deschide prin comanda **fopen** în mod **read only**. În cazul în care fișierul nu există în director, se va afișa un mesaj de eroare (*Eroare la deschiderea fișierului coord.num*).

Coordonatele sunt citite ca șir de caractere într-un tabel cu dimensiunea maximă 3, **carcit**. În continuare fiecare caracter din șirul de caractere este convertit într-o constantă de tip întreg și este memorat într-un tabel cu dimensiunea maximă 3, **cartran**. Pornind de la aceste constante, se calculează coordonatele care au fost transformate astfel, din șir de caractere în constante de tip întreg. Aceste constante sunt stocate într-un al treilea tabel cu dimensiunea maximă 200, **res**.

După terminarea operațiilor de citire din fișier, acesta este închis, utilizând comanda **fclose**. În continuare se inițializează modul de lucru grafic prin apelarea funcției **initgraph**. În cazul în care modul grafic nu poate fi inițializat, se afișează un mesaj de eroare (*Eroare la inițializarea modului grafic*).

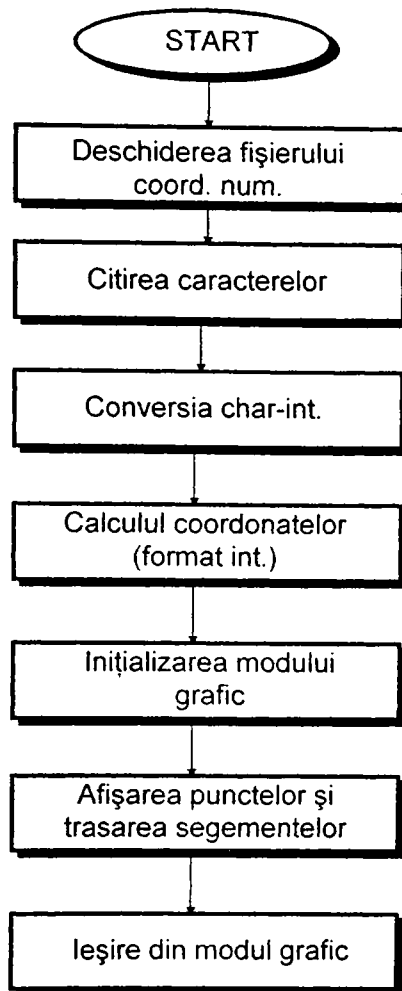
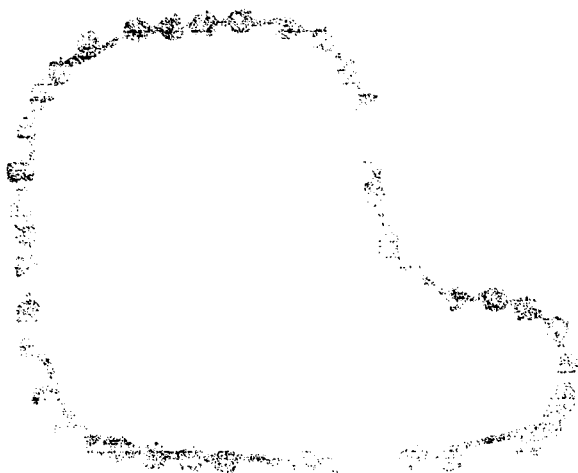


Figura 7.31.

Modul grafic odată inițializat, se trece la afișarea punctelor ale căror coordonate au fost citite din fișierul **coord.num**. Reprezentarea punctelor se face prin trasarea unor cercuri care au raza trei pixeli, apoi se unesc punctele prin segmente de dreaptă, pentru a reprezenta traiectoria parcursă de către robocar. Ultima operație este ieșirea din modul grafic prin comanda **closegraph**.

Prin trasarea cu ajutorul acestui program a traiectoriei robocarului, s-a obținut reprezentarea din figura 7.32.



**Figura 7.32.**

**Se observă coincidența celor două traiectorii (figurile 7.30 și 7.32), aceasta fiind o verificare suplimentară a corectitudinii calibrării camerei video.**

Pe baza experimentelor de mai sus se poate afirma că metoda de măsurare a unghiurilor cu ajutorul camerei video propusă de către autor, pe baza relațiilor 4.97 și 4.95, poate fi utilizată pentru determinarea situării unui obiect în spațiu și în particular a unui *robocar*.

## **7.6. Planificarea traiectoriei robocarului în cadrul hipersistemului CIM.**

Pentru încadrarea robocarului în sistemul **CIM** existent în Laboratorul de robotică de la Facultatea de Mecanică a Universității Politehnica din Timișoara, a fost necesară reprezentarea hărții spațiului de lucru și planificarea traiectoriei vehiculului în acest spațiu. Harta a fost realizată pe baza măsurărilor efectuate de către autor și este prezentată în figura 7.33.

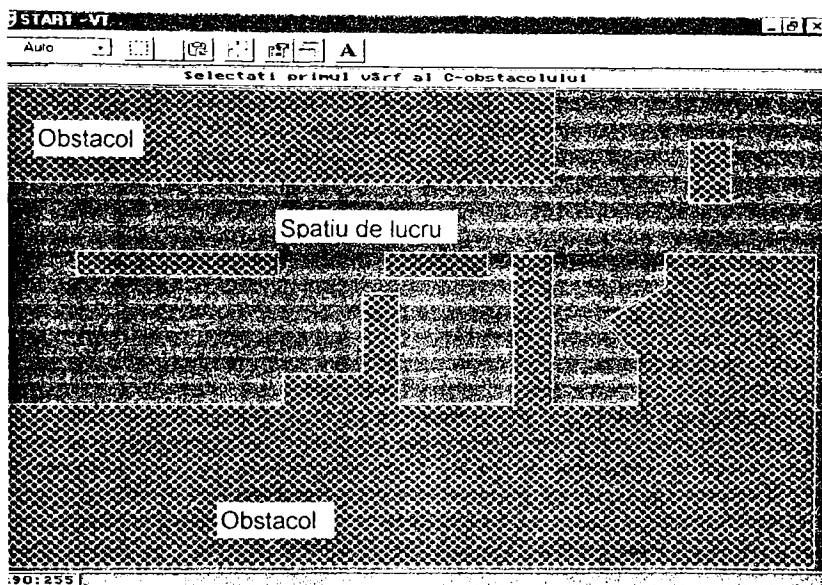


Figura 7.33.

Planificarea traiectoriei robocarului a avut loc prin utilizarea unor programe care implementează metoda **grafului vizibilității** și metoda **grilei neomogene**. În figura 7.34 este prezentată o etapă intermediară a planificării traiectoriei, utilizând metoda **grafului vizibilității**.

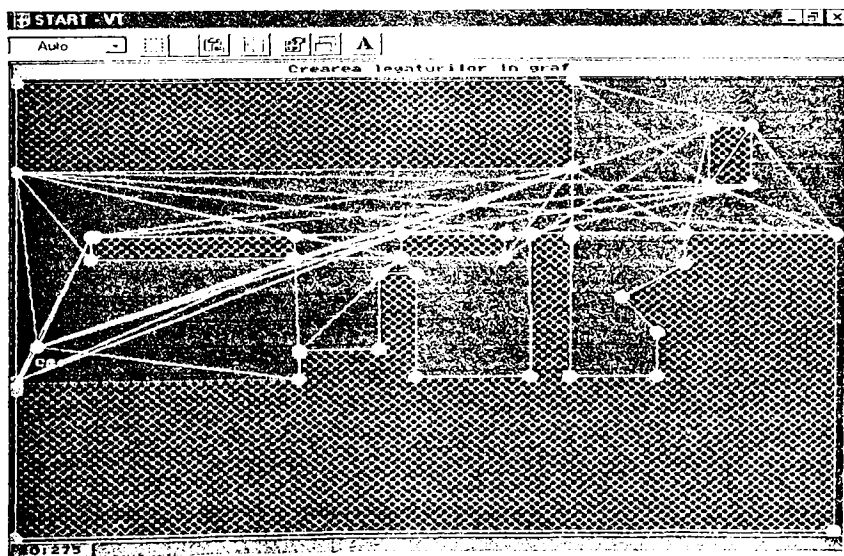


Figura 7.34.

Figura 7.35 ilustrează o etapă intermediară obținută în timpul planificării, prin utilizarea metodei grilei neomogene.

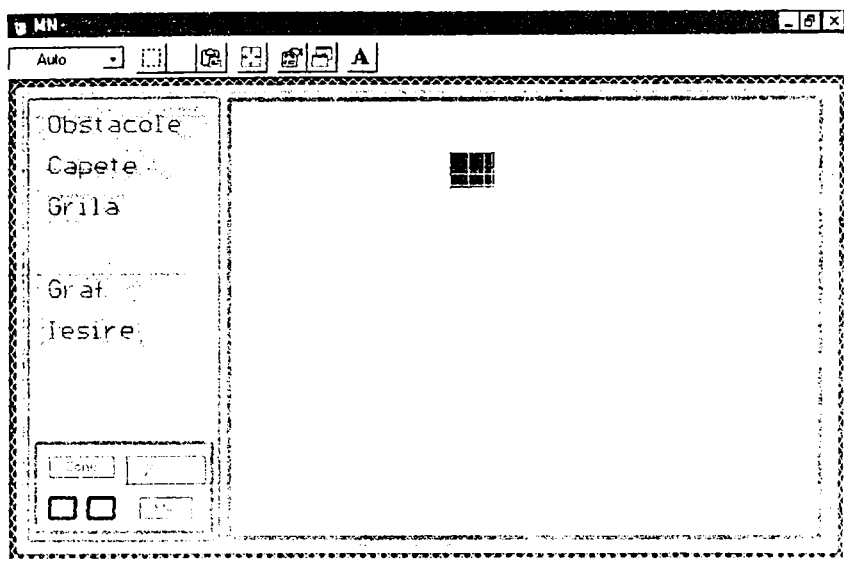


Figura 7.35.

Varianta finală a traiectoriei planificate este reprezentată în figura 7.36. În colțul din stânga sus se află originea sistemului de axe de coordonate ale spațiului de lucru, **SW**, iar în colțul din stânga jos este marcat punctul de plecare a robocarului, **SA**. Robocarul a fost experimentat în cadrul sistemului **CIM**, în figura 7.37 fiind prezentat un aspect din timpul acestor experimentări. Se observă că pe vehicul s-a montat manipulatorul **SCORBOT**, acesta îndeplinind rolul de dispozitiv de încărcare-descărcare. Cu ajutorul lui au fost executate operații simple, cum ar fi: preluarea unor obiecte și transportul acestora la o destinație dinaintea stabilită, deschiderea unor uși pentru a face posibilă trecerea robocarului, etc. Rezultatele experimentale obținute au confirmat rezolvarea în totalitate a obiectivelor propuse și au deschis o cale pentru cercetările ulterioare în acest domeniu.

În capitolul 8, pg.6 sunt enumerate și alte probleme aparținând acestui domeniu, care au fost rezolvate de către autor, dar nu au fost prezentate în teza de față. Pentru fiecare problemă rezolvată a fost efectuat un număr însemnat de experimente. Atât elementele prezentate în teză, cât și cele care nu au fost descrise sunt funcționale și fac parte din robocarul construit de către autor, care poate fi utilizat și în cadrul unor cercetări ulterioare.

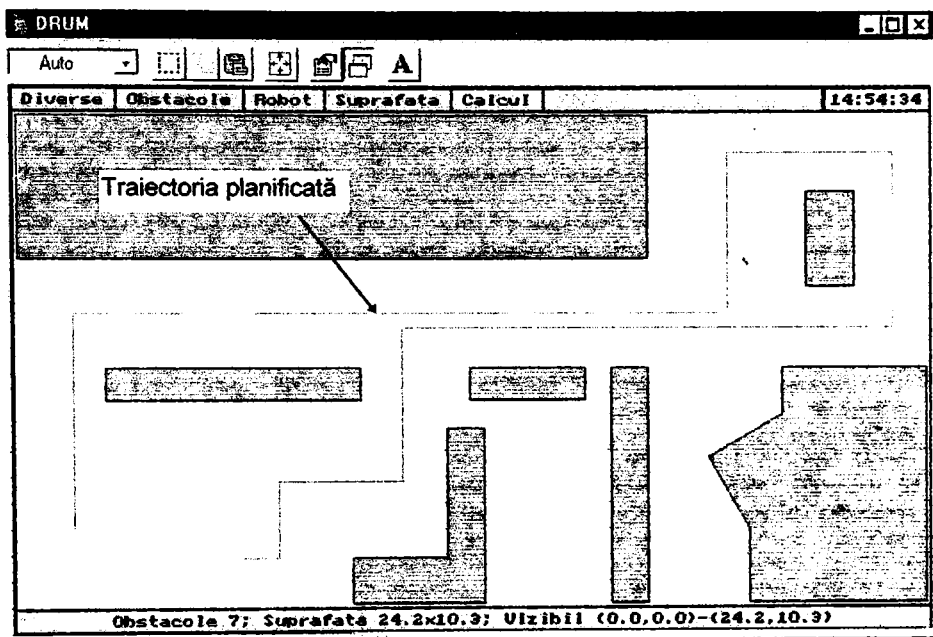


Figura 7.36.

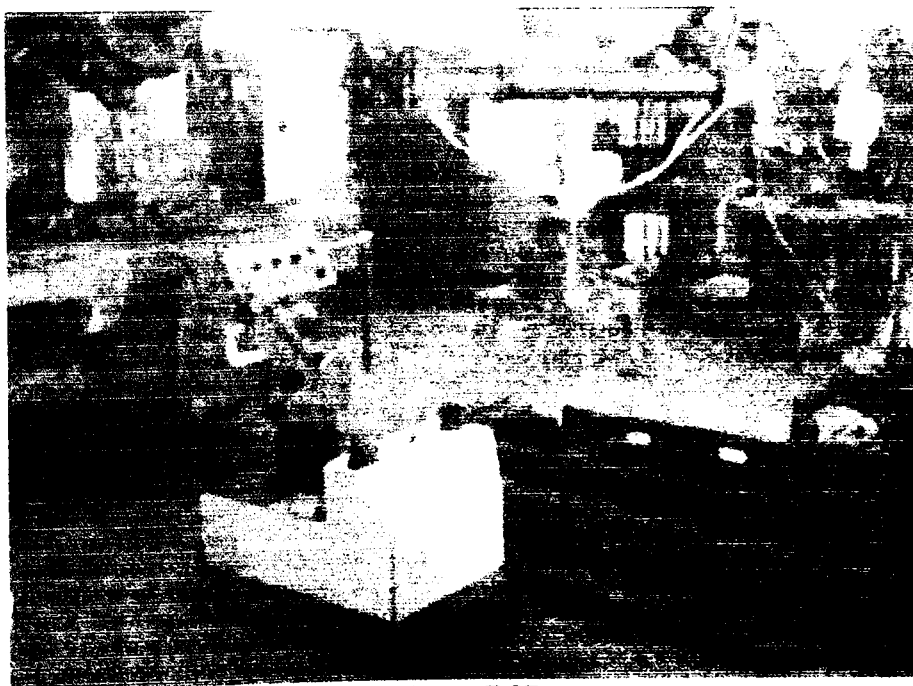


Figura 7.37.





## Capitolul 8

### CONCLUZII FINALE ȘI CONTRIBUȚII

Prezenta teză este rezultatul activității de perfecționare prin doctorat, activitate corelată cu preocupările profesionale ale autorului. Ea reflectă contribuțiile originale și realizările concrete în domeniul **vehiculelor ghidate automat**, elaborate pe parcursul ultimilor ani.

Activitatea de cercetare a autorului în domeniul roboților industriali a început cu aproape 20 de ani în urmă, în cadrul unui colectiv de cercetare multidisciplinar. În anul 1898, domeniul de interes al acestuia devine cel al **sistemelor de transport automate** și ca parte componentă a acestora, **vehiculele ghidate automat**. Cursul **Bazele roboticii** și cartea care urmează a fi publicată, abordează **vehiculele ghidate automat**, atât sub aspect teoretic, cât și practic.

#### 8.1. Considerații referitoare la modul de realizare a unui robocar.

**Robocarul construit în întregime de către autor este un vehicul funcțional, capabil să se deplaseze în mediul interior condus de către un sistem propriu de conducere.** El este echipat cu sisteme senzoriale moderne, cu ajutorul cărora poate fi oricând determinată situația vehiculului. Între sistemul său de conducere și sistemul de conducere aflat pe un nivel superior ierarhic are loc o comunicare permanentă, necesară schimbului de informații referitoare la deplasarea vehiculului. Figura 8 prezintă modul de realizare a comunicării dintre cele două sisteme utilizate de către autor.

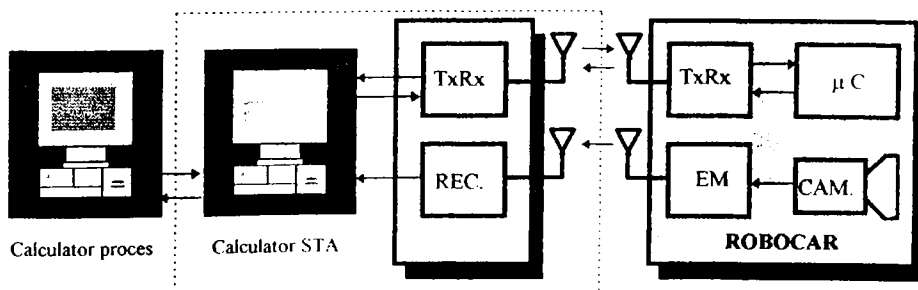


Figura 8.1.

Deoarece nu s-a construit un echipament special pentru achiziția imaginilor și a fost utilizată o placă uzuală, imaginile sunt transmise între robocar și sistemul de conducere al procesului de transport prin unde radio. În același mod se realizează și transmiterea datelor prin interfața serială RS 232.

Funcționarea corectă a tuturor elementelor constructive ale vehiculului a fost verificată experimental. În figura 8.2 este prezentat robocarul realizat de către autor.

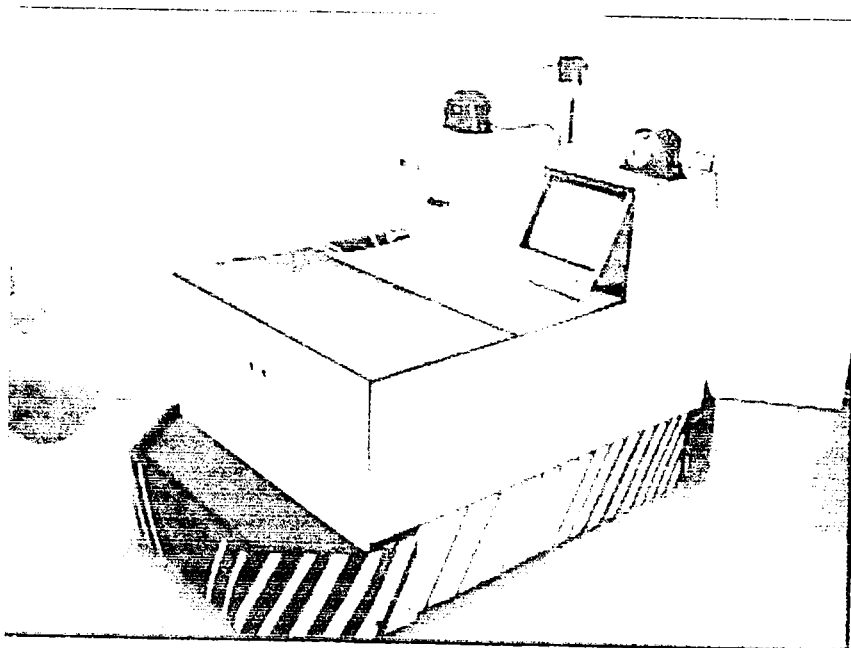


Figura 8.2.

Pe baza celor prezentate pe parcursul tezei, se pot emite câteva **părerii personale**, care vor fi formulate în continuare.

Construcția mecanică este subansamblul care realizează primul contact cu mediul înconjurător. Ea constituie suportul pentru sistemul locomotor al vehiculului și pentru dispozitivele cu ajutorul cărora robocarul efectuează transferul sarcinii spre și dinspre mediul înconjurător. De aceea este importantă, în primul rând, asigurarea funcționalității vehiculului din punct de vedere mecanic. Sistemul de rulare (locomotor) al robocarului trebuie conceput în strânsă legătură cu celelalte sisteme (de acționare, senzorial și de conducere).

Modul de realizare a sistemului de rulare determină varianta aleasă pentru acționare. Aceasta, la rândul determină anumite funcții ale sistemului de conducere. Pe de altă parte, sistemul senzorial și traductoarele utilizate, determină modul de realizare al sistemului de rulare. Dacă deplasarea are loc pe baza odometriei, măsurarea diferențială a spațiului parcurs necesită algoritmi mai simpli pentru calculul coordonatelor. Din acest motiv este necesară măsurarea deplasării cu ajutorul a două roți simetrice. Utilizarea unui sistem unic de tracțiune și de virare cu două roți motoare antrenate diferențial și neorientabile este soluția cea mai indicată pentru aplicarea odometriei diferențiale. Aceasta necesită însă o comandă independentă a elementelor de execuție ale celor două motoare de tracțiune. Dacă se dorește o simplificare a modurilor de comandă, se utilizează soluția cu roată de tracțiune activă. Măsurarea odometrică în această situație este însă mai complexă decât în primul caz.

Sistemul senzorial utilizat se poate baza pe ultrasunete, care oferă modul cel mai simplu pentru măsurarea distanțelor. Măsurările sunt însă perturbate atunci când se efectuează în mediul industrial, iar precizia de determinare a distanței este scăzută. Măsurarea nu poate avea loc în timpul mersului, decât prin acceptarea unor erori datorate deplasării robocarului.

Camera video reprezintă, după părerea autorului, soluția optimă pentru realizarea sistemului senzorial al vehiculului atât în determinarea situației, cât și pentru recunoașterea obstacolelor. Mai mult decât atât, deplasarea vehiculului poate avea loc și prin recunoașterea și urmărirea unui obiect aflat în mișcare. Dacă se utilizează camera video, trebuie avută în vedere necesitatea reducerii timpului afectat procesării informației.

Sistemul de conducere trebuie să fie capabil să îndeplinească cât mai multe funcții, astfel încât să fie posibilă aplicarea modului de conducere descentralizată întregului proces de transport sau de fabricație.

## **8.2. Contribuții**

Concluziile referitoare la conținutul fiecărui capitol al tezei sunt detaliate în cadrul ultimei secțiuni a capitolului respectiv, motiv pentru care nu mai este necesară o reluare a lor.

Contribuțiile autorului sunt, de asemenea, detaliate în aceste ultime secțiuni.

În continuare se vor prezenta în rezumat contribuțiile autorului, sub forma unei înșiriri sintetice. Informații suplimentare se pot obține din concluziile fiecărui capitol sau, în amănunt, din conținutul acestora.

### 8.2.1. Obiectivele cercetării au fost următoarele:

- a. **Stabilirea numărului minim de elemente** care compun un vehicul ghidat automat. Pe baza concluziilor desprinse din analiza unor structuri cunoscute din literatură s-a enunțat tema de cercetare și proiectare, care a fost rezolvată pe parcursul tezei. Structura robocarului care a fost construit s-a propus după parcurgerea următoarelor etape:
  - a1. *Precizarea elementelor componente* ale sistemului de transport automat (**STA**), care include, după părerea autorului, atât robocarul propriu zis, cât și elementele din mediu cu care acesta intră în contact nemijlocit (infrastructura). Aceste elemente au fost prezentate sintetic, fiind abordate și problemele ridicate de construcția și exploatarea acestora;
  - a2. *Precizarea avantajelor și dezavantajelor* pe care le oferă mediul de lucru industrial sistemului de transport automat. Cunoașterea acestora este necesară proiectantului vehiculului sau al sistemului, pentru a lua măsuri de eliminare a dezavantajelor și pentru a folosi constructiv avantajele oferite;
- b. **Determinarea funcțiilor sistemului de conducere** s-a impus atât pentru a avea disponibile datele necesare studiului acestuia, cât și pentru proiectarea sistemului de acționare. Funcțiile sistemului de conducere au fost sintetizate prin:
  - b1. *Analiza hipersistemului CIM*, fiind prezentate toate elementele sale componente și funcțiile acestora;
  - b2. *Integrarea robocarului în sistemul de transport automat*, cuprins la rândul său în hipersistemul **CIM**;
  - b3. *Definirea unor termeni* referitori la anumite funcții ale sistemului de conducere, care nu sunt tratați explicit în literatura de specialitate;
- c. **Evidențierea importanței cercetării și perfecționării metodelor de determinare a situației** a avut loc prin:
  - c1. *Analiza funcțiilor* care sunt abordate în timpul și în scopul deplasării vehiculului (navigația), care fac posibilă deplasarea robocarului pe traiectoria dorită;

- 
- c2. *Analiza strategiilor de navigare*, care se referă la modul în care are loc deplasarea vehiculului și sunt impuse de către operatorul uman sau de către sistemul de conducere aflat pe un nivel ierarhic superior;
  - c3. *Analiza procedurilor de navigare*, bazate pe modurile de determinare a situației;
  - c4. *Analiza metodelor de determinare a situației*, acestea fiind mult mai complexe decât cele utilizate de către sistemele de conducere ale roboților industriali;
  - c5. *Analiza senzorilor și a sistemelor senzoriale utilizate*, pentru luare a unei decizii în ceea ce privește alegerea acestora.
- d. **Conceperea și dezvoltarea unor metode pentru determinarea situației** unui robocar care funcționează în mediul industrial. Pentru a fi atins acest obiectiv au fost parcurse următoarele etape:
- d1. *S-au studiat și sintetizat numeroase metode utilizate pentru determinarea situației*, prezentate în bibliografia studiată;
  - d2. *S-au analizat avantajele și dezavantajele* senzorilor uzuali disponibili;
  - d3. *Au fost concepuți algoritmi și programe* pentru prelevarea și procesarea informațiilor necesare;
  - d4. *Au fost efectuate experimente* cu anumiți senzori, fiind utilizate programele concepute.
- e. **Realizarea unui suport material** necesar efectuării experimentelor a constat din următoarele:
- e1. *Concepția, proiectarea și realizarea unui sistem mecanic complet*, care poate fi echipat cu toate componentele necesare funcționării robocarului;
  - e2. *Concepția, proiectarea și realizarea sistemului de conducere* capabil să gestioneze echipamentele electrice și electronice din componența robocarului;
  - e3. *Concepția, proiectarea și realizarea sistemului de acționare*, compus din motoare, elemente de execuție și programe special concepute pentru generarea semnalelor de comandă necesare elementelor de execuție;
  - e4. *Implementarea algoritmilor, conceperea și realizarea programelor* necesare determinării situației robocarului.

f. **Confirmarea rezultatelor teoretice** prin experimentarea cu ajutorul:

- f1. *Robocarului* construit în întregime de către autor;
- f2. *Echipamentelor și instalațiilor* de măsură speciale.

Aceste obiective au fost atinse în **totalitate**, pe parcursul elaborării tezei fiind abordate și rezolvate numeroase alte probleme, care sunt în strânsă legătură cu tematica de față, dar au fost considerate importante numai pentru perfecționarea autorului în domeniul roboticii și pentru activitatea sa de cercetare din acest domeniu.

Dintre obiectivele suplimentare propuse și rezolvate pot fi menționate:

- realizarea unui sistem senzorial pentru determinarea situației bazat pe ultrasunete;
- realizarea unui echipament pentru ghidarea inductivă și optică;
- realizarea mai multor variante de elemente de execuție;
- realizarea unui sistem pentru transmiterea datelor prin radiație infraroșie. De asemenea se află în stadiu de experiment un sistem de transmitere radio;
- realizarea unui sistem pentru transmiterea imaginilor furnizate de către camera video;
- realizarea unui dispozitiv pentru încărcarea acumulatorilor, echipat cu microcontroler.

### 8.2.2. Rezultatele obținute referitor la obiectivele propuse.

În concordanță cu obiectivele menționate mai sus, au fost obținute următoarele rezultate:

1. S-a prezentat sintetic și unitar fiecare element care face parte din structura robocarului și a **STA** (1.1+1.35, pg.1.5+1.23);
2. S-a elaborat o structură minimală a unui **VGA** (1.4, pg.1.25);
3. Prin analiza hipersistemului **CIM** a fost integrat **STA** în cadrul acestuia (2.2, pg. 2.4+2.6);
4. Determinarea funcțiilor de conducere a fost posibilă prin studiul elementelor care compun **STA** și **CIM** (2.1, pg. 2.1+2.4; 2.4, pg. 2.11+2.13);
5. Au fost analizate și definite noțiunile:
  - **navigație** (2.5, pg. 2.13);
  - **strategie de navigare** (2.5, pg. 2.14);
  - **procedee de navigare** (2.5, pg. 2.14).

6. Analiza metodelor de determinare a situației a pus în evidență cele două moduri de determinare a situației:
  - determinarea situației relative (3.1, pg. 3.2+3.3);
  - determinarea situației absolute (capitolul 4).și a evidențiat complexitatea și importanța pe care determinarea situației o are în cadrul funcționării sistemului de conducere;
7. Au fost elaborate metode originale pentru determinarea situației și s-au conceput și realizat algoritmi speciali necesari pentru experimentarea metodelor elaborate (4.2.5, 4.2.7, 4.2.8, 4.2.9, 4.2.12, pg. 4.44+4.52);
8. S-au conceput programe pentru determinarea situației prin măsurarea distanței parcurse (3.2, pg. 3.3+3.31), a unghiurilor de virare și a unghiurilor formate de către axa optică a camerei video cu axele sistemului de coordonate (4.2.11, pg. 4.42+4.45);
9. Au fost verificate pe cale experimentală programele și metodele propuse (7.5, pg. 7.8+7.12);
10. Elaborarea procedurii de calibrare a camerei video (4.2.10, pg. 4.40+4.42) a fost necesară pentru verificarea experimentelor efectuate mai sus;
11. Pe baza procedurii de calibrare propus s-a efectuat calibrarea camerei video utilizate (7.4, pg. 7.5+7.8);
12. Sistemul mecanic a fost proiectat și realizat în întregime (capitolul 5).
13. A fost realizat sistemul de acționare care constă din:
  - motoarele electrice (5.2.1, pg. 5.6; 5.2.2, pg. 5.7);
  - elementele de execuție corespunzătoare (6.1.1.1, pg. 6.3+6.7; 6.1.2.1, pg. 6.11+6.13).
14. Sistemul de conducere a fost proiectat și realizat integral (capitolul 6);
15. Pentru funcționarea corectă a sistemului de conducere s-au generat, pe baza unor programe realizate în acest scop, semnalele de comandă necesare (6.1.1.2, pg. 6.7+6.10, 6.1.2.2, pg. 6.13+6.15; 6.1.3.2, pg. 6.16+6.21);
16. A fost verificată funcționarea robocarului echipat cu toate sistemele realizate (7.3, pg. 7.4);
17. Robocarul a fost încadrat în sistemul **CIM** al Laboratorului de RI al Facultății de Mecanică de la Universitatea "Politehnica" din Timișoara.

### 8.2.3. Contribuții originale.

Din analiza obiectivelor cercetării și a rezultatelor obținute de către autor, pot fi evidențiate principalele contribuții originale aduse în prezenta teză. Acestea sunt:

1. *Selecția și prelucrarea sintetică a materialului bibliografic*, urmată de o prelucrare riguroasă a acestuia;
2. *Prezentarea sistematizată a elementelor constitutive ale unui VGA* este rezultatul prelucrării materialului bibliografic și constituie concentrarea unei vaste bibliografii. Abordarea problemelor se face într-un mod personal, iar sinteza reprezintă punctul de vedere al autorului (pg.1.5÷1.23);
3. *Enumerarea avantajelor și dezavantajelor mediului industrial* este necesară pentru a ușura proiectarea robocarului. Stabilirea acestora a avut loc pe baza analizei fenomenelor fizice caracteristice acestui mediu (pg. 1.24);
4. *Descrierea structurii unui robocar utilizat în transportul industrial* este rezultatul cunoașterii elementelor constructive descrise și a avantajelor și dezavantajelor mediului industrial (pg. 1.25);
5. *Prezentarea sistematică și ierarhizată a nivelurilor sistemelor de conducere* pune în evidență sarcinile sistemului de transport automat industrial (pg. 2.7÷2.11);
6. *Prezentarea sistemului de conducere al VGA*, încadrat în hipersistemul **CIM**, este necesară pentru stabilirea locului sistemului de conducere al **VGA** în cadrul procesului de producție (pg. 2.11÷2.13);
7. *Evidențierea funcțiilor îndeplinite de către sistemul de conducere al robocarului* duce la proiectarea optimă a acestui sistem (pg. 2.11÷2.13);
8. *Atribuirea termenului navigație* unui grup de funcții este rezultatul analizei funcțiilor îndeplinite de către sistemul de conducere pentru realizarea deplasării pe traiectorie a robocarului în condiții de maximă siguranță și cu precizia dorită (pg. 2.13);
9. *Definirea navigației* pe baza funcțiilor îndeplinite de către sistemul de conducere și a sensului lingvistic al termenului (pg. 2.13);
10. *Explicitarea termenului strategii de navigare*. În literatura de specialitate se folosește acest termen fără a fi explicitat sensul său (pg. 2.14);
11. *Explicitarea termenului procedee de navigare*. În literatura de specialitate se face distincție între navigare, strategie de navigare și procesul de navigație. Și în acest caz s-a ținut cont de sensul lingvistic (pg. 2.14);
12. *Alegerea microcontrolerului 80C552* s-a făcut din considerentele



prezentate în capitolul 6, în concordanță cu funcțiile pe care trebuie să le îndeplinească sistemul de conducere în cazul concret al robocarului construit de către autor (cap.6, pg. 6.2);

13. *Conceperea programului* pentru realizarea celor patru moduri de funcționare. Pentru a se asigura funcționarea robocarului, necesară experimentării circuitelor electronice și a programelor prezentate, s-a conceput un program pe baza căruia robocarul să se deplaseze (cap.3, pg. 3.2.1÷3.2.5):

- prin comandă manuală,
- pe o traiectorie impusă,
- pe o traiectorie învățată în urma instruirii;

14. *Alegerea modului de funcționare pe întreruperi* a programului a avut loc în scopul de a face posibilă intervenția în orice moment a operatorului uman. Acest mod de lucru este propus de către autor (capitolul 3, 6);

15. *Conceperea și implementarea algoritmului de inițializare* a sistemului de virare. Inițializarea este necesară numai pentru sistemul de virare, în scopul de a poziționa roțile directoare astfel încât robocarul să se deplaseze rectiliniu. Inițializarea sistemului de tracțiune se face prin poziționarea robocarului în poziția de plecare (pg. 3.6÷3.7);

16. *Conceperea și implementarea algoritmului pentru stabilirea limitelor de virare* la stânga și la dreapta. Limitele mecanismului de virare (stânga, respectiv dreapta) nu sunt determinate mecanic sau electric (prin senzori), ci sunt impuse prin soft și astfel pot fi ușor modificate (pg. 6.13÷6.14);

17. *Conceperea și implementarea algoritmului de învățare*. Pe baza acestui algoritm s-a realizat un program care permite instruirea sistemului de conducere al robocarului de către operatorul uman și apoi deplasarea vehiculului în mod automat, pe traseul învățat (pg. 3.20÷3.31);

18. *Eliminarea comenzilor false și redundante* este necesară pentru a nu fi învățate și manevrele greșite s-au acelea care nu sunt necesare. Algoritm este imaginat de către autor (pg. 3.20÷3.28);

19. *Comprimarea numărului comenzilor* este necesară pentru reducerea necesarului de memorie și are loc pe baza unui algoritm conceput de către autor (pg. 3.20÷3.22)

20. *Dezvoltarea principiului referențierii discrete*. Principiul referențierii discrete poate fi utilizat cu succes în mediul industrial, care este un mediu structurat. Prin aplicarea acestui principiu, determinarea situații absolute nu are loc în mod continuu, reducându-se numărul reperelor, iar sistemul de conducere și cel senzorial sunt degrevate de detectarea permanentă a acestora (pg. 4.2÷4.7);

21. *Determinarea situații unui robocar în spațiu prin vizarea a patru*

*puncte țintă.* Situarea robocarului poate fi determinată prin vizarea a patru puncte țintă, aflate pe o față a acestuia. În cazul mișcărilor complexe, se vor amplasa câte patru puncte pe mai multe fețe. Metoda este aplicabilă și pentru determinarea situației unui obiect oarecare în spațiul 3D (pg. 4.20÷4.27);

22. *Determinarea situației unui robocar în plan prin vizarea a două puncte țintă.* Robocarul se deplasează numai în plan (spațiu 2D), pe o traiectorie cunoscută în general. De aceea se poate determina situația acestuia prin vizarea a două puncte țintă aflate pe vehicul. Algoritmul a fost propus și dezvoltat de către autor (pg. 4.31÷4.34);

23. *Determinarea situației unui robocar cu cameră video staționară.* Camera video este fixă, iar pe robocar sunt amplasate cele două puncte țintă (pg. 4.34÷4.36);

24. *Determinarea situației unui robocar cu cameră video amplasată pe vehicul.* Prin amplasarea camerei video pe vehicul se reduce numărul camerelor utilizate la una singură (pg. 4.36÷4.40);

25. *Metoda de calibrare a camerei CCD* a fost elaborată pentru a oferi posibilitatea măsurării unghiurilor cu ajutorul acesteia (pg. 4.40÷4.42);

26. *Conceperea programului pentru măsurarea distanței.* Pe baza acestui program se măsoară distanța în pixeli între două puncte. Cunoscând dimensiunile celulei (CCD) în  $\mu\text{m}$  se pot determina distanțele în metri (pg. 4.44÷4.52);

27. *Măsurarea unghiurilor cu ajutorul camerei video.* Determinarea situației cu ajutorul camerei video se bazează pe măsurarea unor distanțe, pentru aceasta fiind necesară determinarea unor unghiuri. Modul de măsurare a acestora este propus de către autor (pg. 4.42÷4.44);

28. *Stabilirea criteriilor* pentru aprecierea metodelor de determinare a situației. Pentru alegerea unei metode de determinare a situației este necesară compararea, pe baza unor anumite criterii, a metodelor cunoscute. Criteriile cele mai importante pentru realizarea comparației sunt enunțate de către autor (pg. 4.43÷4.56);

29. *Conceperea și alegerea tipului mecanismului de tracțiune și a elementelor sale componente.* Mecanismul de tracțiune a fost conceput având la bază variantele cunoscute. Structura a fost impusă de către performanțele cerute și de limitările de ordin material (pg. 5.5÷5.7);

30. *Conceperea și alegerea tipului mecanismului de virare și a elementelor sale componente.* Mecanismul de virare este proiectat și realizat în întregime de către autor, pentru a se obține performanțe cât mai bune (rază de virare redusă, ceea ce înseamnă mobilitate ridicată) (pg. 5.7÷5.15);

31. *Construirea robocarului.* Vehiculul este construit integral de către

autor;

32. *Stabilirea funcțiilor pe care trebuie să le îndeplinească elementele de execuție* este necesară pentru proiectarea eficientă a circuitelor electronice ale acestora și asigurarea unor performanțe ridicate (pg. 6.3; 6.10);

33. *Conceperea, proiectarea și construirea elementelor de execuție* s-a realizat integral (pg. 6.4÷6.7; 6.11÷6.12; 6.15÷6.16);

34. *Adaptarea circuitelor integrate utilizate* prin modificări ale schemei de aplicație. Modificările au fost necesare pentru ca circuitele integrate utilizate să realizeze funcțiile dorite (comanda PWM pentru EEMCC și alimentarea unipolară pentru EEMPP) (pg. 6.6 ÷6.12);

35. *Modularizarea și tipizarea elementelor de execuție* face posibilă schimbarea modulelor cu altele, care realizează aceleași funcții, dar au structuri diferite (pg. 6.3÷6.13);

36. *Modularizarea și tipizarea sistemului de conducere*. Sistemul de conducere poate fi înlocuit cu altul, cu condiția ca acesta să realizeze cel puțin aceleași funcții ca și cel precedent (pg. 6.22÷6.29);

37. *Conceperea, proiectarea și construirea sistemului de comandă*. Acesta a fost conceput și construit pentru a face posibilă punerea în practică a unui număr cât mai mare de aplicații. În cazul în care au fost definite funcțiile pe care acesta trebuie să le îndeplinească, se poate proiecta un sistem de conducere destinat scopului propus;

38. *Conceperea și realizarea programelor pentru generarea semnalelor de comandă*. Semnalele de comandă sunt generate pe baza unor programe executate de către sistemul de comandă. Se poate însă adopta și varianta în care se realizează un sistem special destinat generării semnalelor de comandă ale elementelor de execuție (pg. 6.7÷6.10; 6.13÷6.14, 6.16÷6.21);

39. *Elaborarea procedurii pentru calibrarea camerei video*. Calibrarea camerei în cadrul experimentelor a pus în evidență justetea considerațiilor teoretice (pg. 4.40÷4.42; 7.5÷7.8);

40. *Efectuarea experimentelor*. Funcționarea robocarului, a sistemului de conducere și a elementelor de execuție, programele pentru determinarea situației, măsurarea unghiurilor și a distanțelor, achiziționarea și prelucrarea imaginilor au fost verificate printr-o serie de experimente propuse și efectuate integral de către autor (cap.7);

41. *Elaborarea programului* pentru măsurarea vitezei prin măsurarea duratei impulsurilor furnizate de către TIRO (7.17÷7.18);

42. *Elaborarea procedurii* pentru verificarea poziționării robocarului când are loc deplasarea rectilinie și pentru verificarea poziționării mecanismului de virare (7.10÷7.14, 7.28÷7.30);

---

43. *Încadrarea robocarului construit în sistemul CIM.* Robocarul a fost realizat cu scopul de a face parte din sistemul CIM al Facultății de Mecanică (7.42);

44. *Planificarea traiectoriei vehiculului.* Traectoria a fost planificată pentru spațiul de lucru real din cadrul Laboratorului de Robotică (sistemul CIM). Urmărind această traiectorie, vehiculul a îndeplinit sarcini simple de transport (7.39÷7.41).

*Robocarul și toate elementele prezentate se află în stare de funcționare, iar cercetarea în acest domeniu va fi continuată de către autor și de către colectivul recent format, împreună cu colegii de la secțiile de RI ale Facultății de Electronică și Telecomunicații și ale Facultății de Mecanică.*

---

**ANEXA 1. Programul pentru deplasarea odometrică a robocarului.**

```
# include <reg552.h>
# include "..\..\lib\sorin.h"

unsigned char Data[2] = " ";
bit invatare,dreapta,fan,r,stop,eroare,m,inainte,automat;
unsigned char pasi,k,j,l,acc,decc;
char pas;
int palier,n,i;
int xdata *d;
char xdata *p;
unsigned char xdata *c;

MCC() interrupt 0
{
n++;
if(r)
    {
    if(n==*d)
        {
        n=0;
        pas=0;
        *d++;
        *p++;
        fan=0;
        }
    }
else
    if(n%53 == 0 & automat)
        {
        if(acc != 0)
            {
            acc--;
            l-=4;
            }
        else
            if(palier != 0)
                {
```

```

        palier--;
        l=0;
    }
    else
        if(decc != 0)
            {
                if( --decc < 20 )
                    l+=3;
                else
                    l+=8;
            }
        else
            {
                EX0=0;
                P1 &= 0xFC;
                PWM0=l;
                l=0;
                eroare=1;
            }
        PWM0=l;
    }
}

```

MPP() interrupt 2

```

{
if(invatare)
    pas--;
else
    pas++;
if(r)
    {
        if(pas==*p)
            {
                PWM1=255;
                pas=0;
                EX1=0;
                if(n==*d)
                    {
                        n=0;
                        pas=0;
                    }
            }
    }
}

```

```
        *d++;
        *p++;
        fan=0;
    }
}
else
    if(--pasi == 0)
    {
        PWM1=255;
        EX1=0;
    }
}

int numar()
{
    if(m)
    {
        txstr("\nIntroduceti numarul de cm (0-999):\n",5);
        j=0;
    }
    else
    {
        txstr("\nIntroduceti numarul de grade (0-60°):\n",5);
        j=1;
        Data[0]='0';
    }
    for(k=j;k<3;k++)
        while(1)
            if( rxch(&Data[k]))
                if(Data[k]<58 && Data[k]>47)
                    break;
                else
                    txch(8);
    i=(Data[0]-48)*100+(Data[1]-48)*10+(Data[2]-48);
    txstr("\nfoarte bun\n",5);
    return i;
}

oprit()
```

```
{
P1 = 0x00;
PWM0=0;
EX0=0;
EX1=0;
PWM1=255;
clearscreen();
write("oprit");
delay(800);
}

prezentare()
{
clearscreen();
write("Bine ati venit in cadrul proiectului ROBOWORLD" );
delay( 800 );
for( k=0; k < 37; k++ )
    {
    dleft();
    delay( 140 );
    }
delay(200);
for(k=0;k<10;k++)
    {
    displayoff();
    delay( 250 );
    displayon();
    delay( 250 );
    }
write("Apasati tasta H pentru help.....      ");
for( k=0; k < 37; k++ )
    {
    dright();
    delay( 140 );
    }
}

profil(int nr,unsigned char *acc,int *palier,unsigned char *decc)
{
if(nr<80)
```



```
    {
        *palier=0;
        *acc=nr/2;
        *decc=nr/2;
    }
else
    {
        *acc=40;
        *decc=40;
        *palier=nr-80;
    }
EX0=1;
l=165;
PWM0=l;
}

start_pasi(unsigned char grade,unsigned char *point)
{
    *point=grade*2;
    PWM1=125;
    EX1=1;
}

sterge()
{
    c=0x2300;
    p=0x3300;
    d=0x5300;
    for(i=0;i<2000;i++)
    {
        *p++=0;
        *d++=0;
        *c++=0;
    }
    c=0x2300;
    p=0x3300;
    d=0x5300;
}

scrie(char pas,int n)
```

```
{
*p++=pas;
*d++=n;
}

scanare_tastatura()
{
while(1)
if( RI )
    {
    Data[0] = S0BUF;
    RI = 0;
    if(Data[0]==50)
        {
        if(!inainte)
            oprit();
        inainte=1;
        clearscreen();
        write("2-mers inapoi");
        automat=0;
        PWM0=120;
        EX0=1;
        P1 &= 0xFE;
        P1 |= 0x02;
        }
    if(Data[0]==56)
        {
        if(inainte)
            oprit();
        inainte=0;
        clearscreen();
        write("8-mers inainte");
        automat=0;
        PWM0=120;
        EX0=1;
        P1 &= 0xFD;
        P1 |= 0x01;
        }
    if(Data[0]==54)
        {
```

```
        clearscreen();
        write("6-rotire dreapta");
        P1 |= 0x0C;
        start_pasi(1,&pasi);
    }
if(Data[0]==52)
    {
    clearscreen();
    write("4-rotire stinga");
    P1 &= 0xFB;
    P1 |= 0x08;
    start_pasi(1,&pasi);
    }
if(Data[0]==97)
    {
    clearscreen();
    write("A-mers inapoi");
    automat=1;
    m=1;
    profil(numar(),&acc,&palier,&decc);
    P1 &= 0xFE;
    P1 |= 0x02;
    eroare=0;
    }
if(Data[0]==102)
    {
    clearscreen();
    write("F-mers inainte");
    automat=1;
    m=1;
    profil(numar(),&acc,&palier,&decc);
    P1 &= 0xFD;
    P1 |= 0x01;
    eroare=0;
    }
if(Data[0]==114)
    {
    clearscreen();
    m=0;
    write("R-rotire la dreapta");
```

```
        P1 |= 0x0C;
        start_pasi(numar(),&pasi);
    }
    if(Data[0]==108)
    {
        clearscren();
        m=0;
        write("L-rotire la stinga");
        P1 &= 0xFB;
        P1 |= 0x08;
        start_pasi(numar(),&pasi);
    }
    if(Data[0]==111)
        oprit();
    if(Data[0]==101)
    {
        c=0x2300;
        p=0x3300;
        d=0x5300;
        pas=0;
        n=0;
        clearscren();
        write("Redare");
        delay(800);
        r=1;
        while(*c)
        {
            if(RI)
                break;
            if(*c==111)
            {
                oprit();
                stop=1;
                delay(500);
                fan=0;
            }
            if(*c=='8')
            {
                oprit();
                PWM0=120;
            }
        }
    }
}
```

```
EX0=1;
P1 &= 0xFD;
P1 |= 0x01;
fan=1;
clearscreen();
write("8-mers inainte");
}
if(*c=='2')
{
oprit();
PWM0=120;
EX0=1;
P1 &= 0xFE;
P1 |= 0x02;
fan=1;
clearscreen();
write("2-mers inapoi");
}
if(*c=='4')
{
clearscreen();
P1 &= 0xFB;
P1 |= 0x08;
PWM1=125;
EX1=1;
fan=1;
clearscreen();
write("4-rotire stinga");
}
if(*c=='6')
{
clearscreen();
P1 |= 0x0C;
PWM1=125;
EX1=1;
fan=1;
clearscreen();
write("6-rotire dreapta");
```

```
        }
        while(fan)
        {
            *c++;
        }
        r=0;
        oprit();
        clearscreen();
        write("Gata");
        txstr("\nAm terminat\n",5);
    }
    if(Data[0]==109)
    {
        sterge();
        automat=0;
        n=0;
        pas=0;
        stop=1;
        gata=1;
        clearscreen();
        write("Invatare");
        while(1)
        if( RI )
        {
            Data[0] = S0BUF;
            RI = 0;
            if(Data[0]==50)
            {
                if(stop)
                {
                    if(pas != 0)
                    {
                        if(pas<0)
                        {
                            pas=-pas;
                            *c++='4';
                        }
                        else
                            *c++='6';
                        scrie(pas,0);
                    }
                }
            }
        }
    }
}
```

```
        }
        n=0;
        *c++=Data[0];
    }
else
    if(inainte)
    {
        scrie(pas,n);
        pas=0;
        n=0;
        *c++=Data[0];
        oprit();
    }
stop=0;
inainte=0;
clearscreen();
write("2-mers inapoi");
PWM0=120;
EX0=1;
P1 &= 0xFE;
P1 |= 0x02;
gata=1;
}
if(Data[0]==56)
{
    if(stop)
    {
        if(pas != 0)
        {
            if(pas<0)
            {
                pas=-pas;
                *c++='4';
            }
            else
                *c++='6';
            scrie(pas,0);
        }
        n=0;
        *c++=Data[0];
    }
}
```

```
    }
else
    if(!inainte)
    {
        scrie(pas,n);
        n=0;
        pas=0;
        *c++=Data[0];
        oprit();
    }
stop=0;
inainte=1;
clearscreen();
write("8-mers inainte");
PWM0=120;
EX0=1;
P1 &= 0xFD;
P1 |= 0x01;
gata=1;
}
if(Data[0]==54)
{
    if(!stop)
    {
        scrie(pas,n);
        pas=0;
        n=0;
        *c++=Data[0];
    }
    invatare=0;
    clearscreen();
    write("6-rotire dreapta");
    P1 |= 0x04;
    start_pasi(1,&pasi);
    gata=0;
}
if(Data[0]==52)
{
    if(!stop)
    {
```



```
        scrie(pas,n);
        pas=0;
        n=0;
        *c++=Data[0];
    }
    else
        invatare=1;
        clearscreen();
        write("4-rotire stinga");
        P1 &= 0xFB;
        start_pasi(1,&pasi);
        gata=0;
    }
if(Data[0]==111)
    {
    if(!stop)
        {
        scrie(pas,n);
        pas=0;
        n=0;
        *c++=Data[0];
        }
    oprit();
    stop=1;
    gata=1;
    }
if(Data[0]=='q')
    {
    if(!stop)
        {
        scrie(pas,n);
        pas=0;
        n=0;
        }
    else
        if(pas != 0)
            if(pas<0)
                {
                pas=-pas;
                *c++='4';
                }
```

```
        }
        else
            *c++='6';
        scrie(pas,0);
        pas=0;
        n=0;
        clearscren();
        write("lesire bucla");
        invatare=0;
        break;
    }
}
}
}

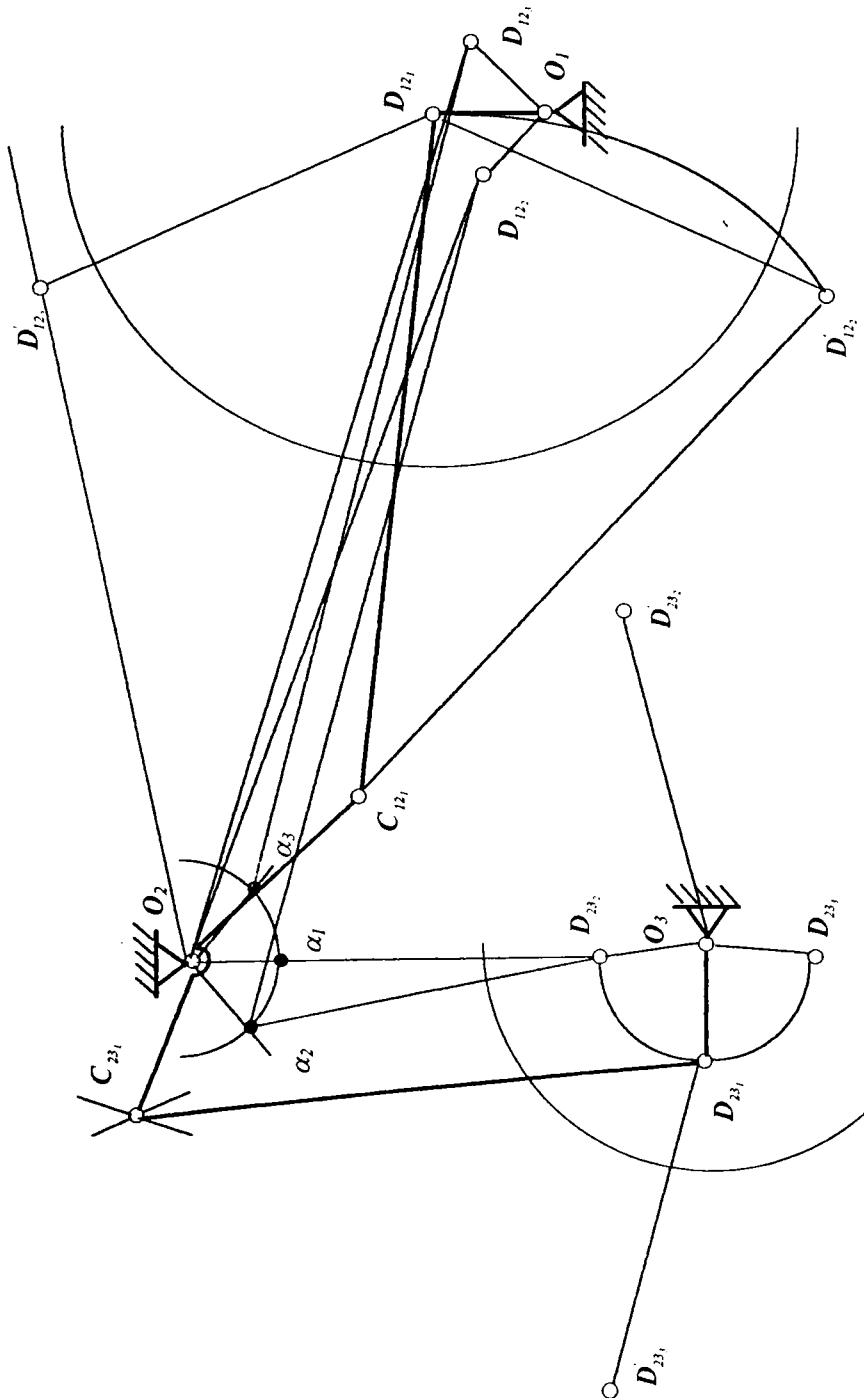
void main(void)
{
    P1=0x00;
    PWMP=140;
    PWM0=255;
    PWM1=255;
    EA = 1;
    r=0;
    invatare=0;
    initdisplay1( 0 );
    initserial();
    TCON |= 0x05;
    prezentare();
    scanare_tastatura();
}
```

## ANEXA 2. Regiștrii microcontrolerului 80C552

SP	STACK POINTER	81
S0BUF	SERIAL 0 DATA BUFFER	99
S0CON*	SERIAL 0 CONTROL	98
S1ADR	SERIAL 1 ADDRESS	DB
S1DAT	SERIAL 1 DATA	DA
S1STA	SERIAL 1 STATUS	D9
S1CON*	SERIAL 1 CONTROL	D8
STE	SET ENABLE	EE
TH1	TIMER HIGH 1	8D
TH0	TIMER HIGH 0	8C
TL1	TIMER LOW 1	8B
TL0	TIMER LOW 0	8A
TMH2	TIMER HIGH 2	ED
TML2	TIMER LOW 2	EC
TMOD	TIMER MODE	89
TCON*	TIMER CONTROL	88
TM2CON	TIMER 2 CONTROL	EA
TM2IR	TIMER 2 INT FLAG REG	C8
T3	TIMER 3	FF

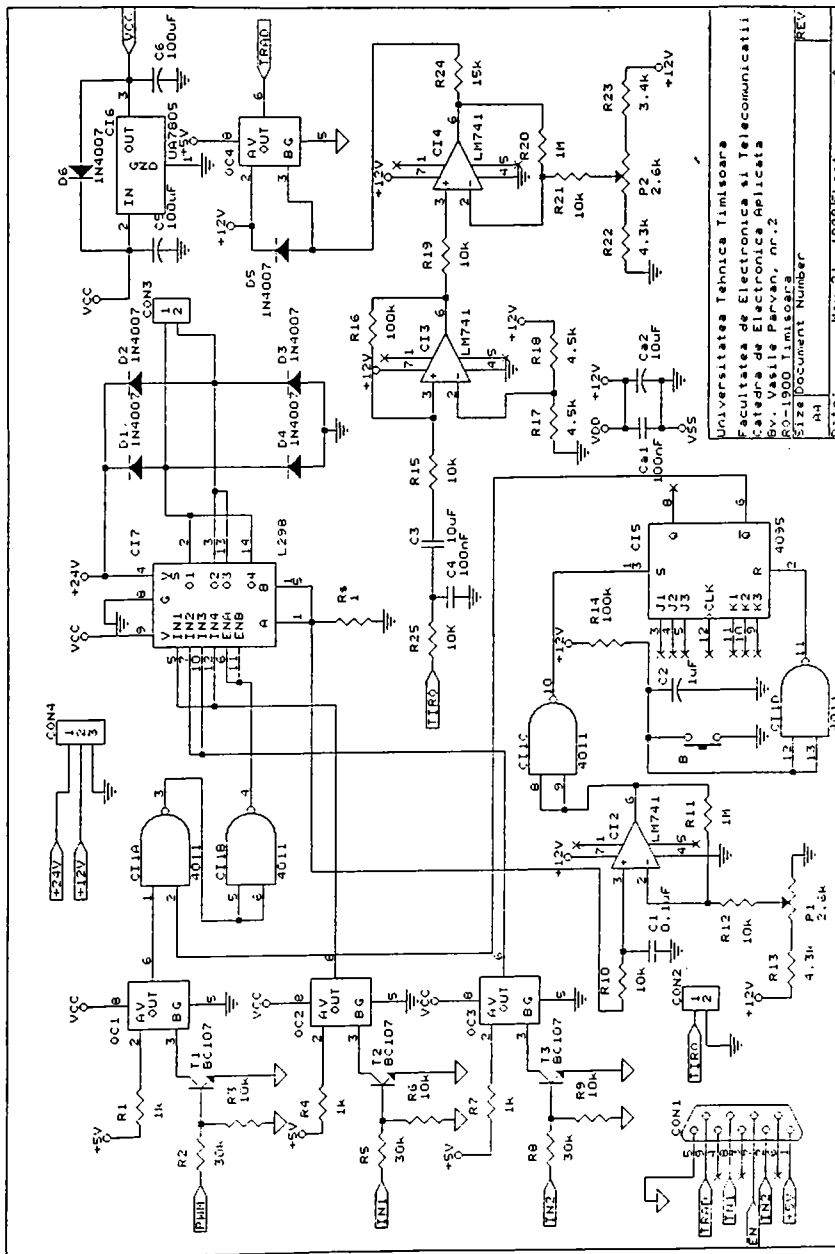


ANEXA 3. Mecanismul de virare





## ANEXA 4. Schema electronică a EEMCC

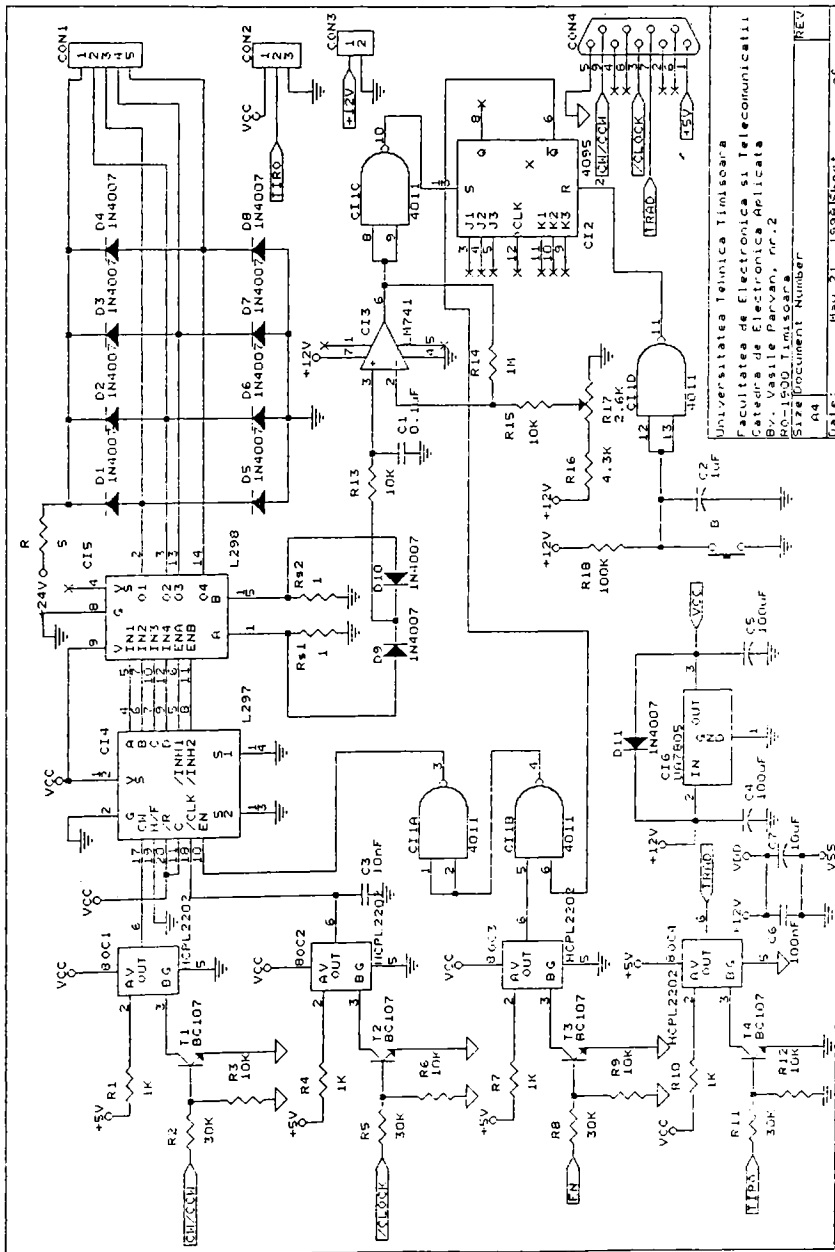


Universitatea Tehnica Timisoara  
 Facultatea de Electronica si Telecomunicatii  
 Catedra de Electronica Aplicata  
 Sv. Vasile Parvan, nr.2  
 RO-1900 Timisoara  
 Size/Document Number  
 Date: May 21, 1998 Sheet 1 of 1



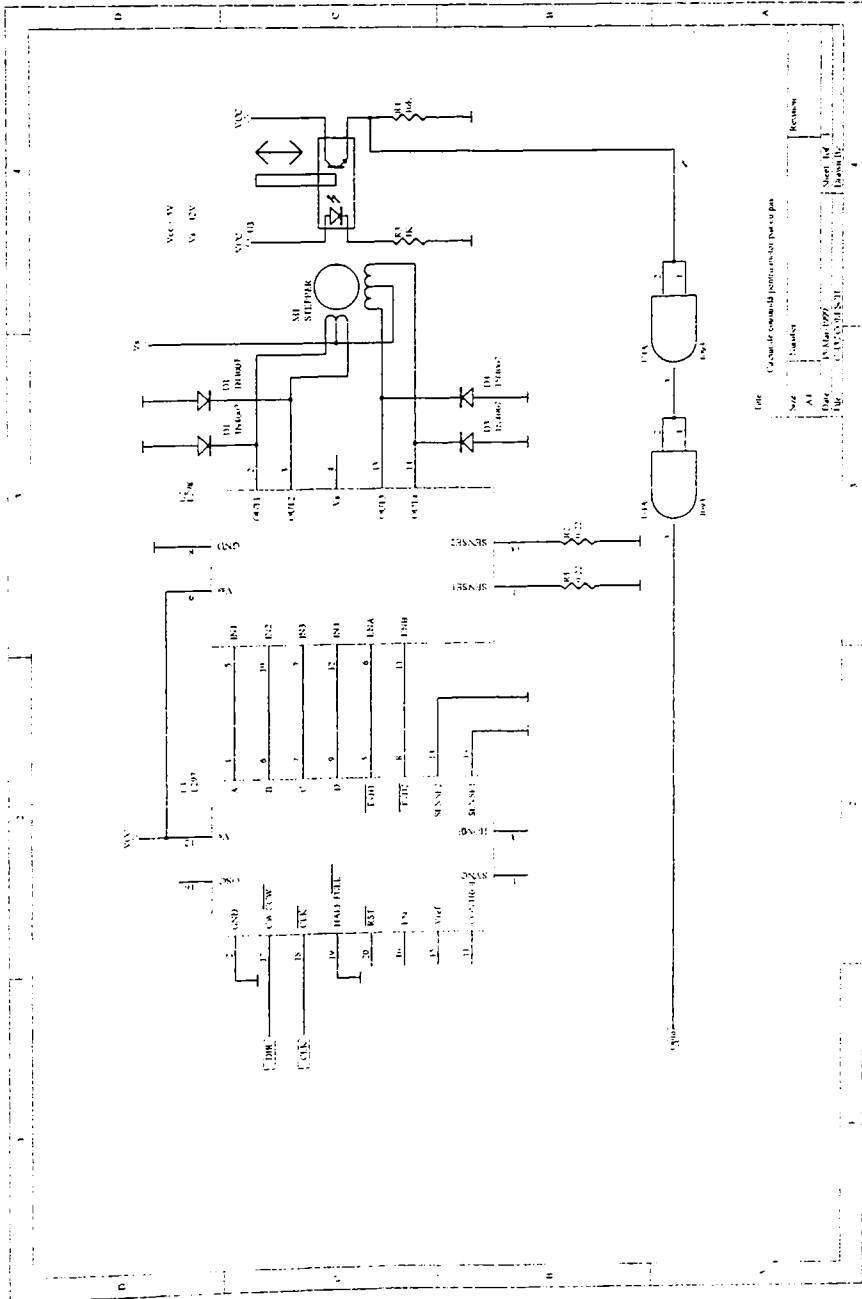


# ANEXA 5. Schema electronică a EEMPP





## ANEXA 6. Schema electronică a elementului de execuție a sistemului de orientare a camerei video.





## ANEXA 7. Biblioteca de funcții

```
#include "...\LIB\SORIN.H"
```

Biblioteca conține module .OBJ:

LIB\_EEP.OBJ - funcții pentru memoria nevolatilă EEPROM;

LIB\_ASM.OBJ - funcții pentru inițializarea modului de lucru TRACE;

LIB\_DAC.OBJ - funcții pentru convertorul numeric/analogic;

LIB\_PORT.OBJ - funcții pentru porturile de intrare/ieșire, intrările analogice;

LIB\_LCD.OBJ - funcții pentru afișor cu cristale lichide;

LIB\_UART.OBJ - funcții pentru întârzieri interfața serială, inițializare timer0;

LIB\_KEYB.OBJ - funcții pentru tastatură;

Aceste module .OBJ sunt reunite într-o singură bibliotecă SORIN.LIB

Funcțiile modulului LIB\_UART:

a.) void delay (unsigned miliseconds);

Funcția generează o întârziere, exprimată în milisecunde, programabilă prin parametrul de intrare miliseconds. Întârzierile sunt aproximative și vor fi ceva mai mari în cazul în care în program sunt activate unele întreruperi.

b.) void initserial (void);

Funcția inițializează interfața serială a microcontrolerului cu parametrii standard folosiți și de programul monitor: modul 3, comunicație multiprocesor, 9600 bauds, 11 biți de date, timer-ul T1 generează rata de transfer, nu generează întrerupere.

c.) unsigned char txch (unsigned char Data);

Funcția transmite pe linia serială octetul Data, dacă transmisia este posibilă (TI=1).

Se returnează valoarea 1 dacă octetul a fost transmis, 0 în cazul în care transmisia nu s-a putut realiza.

d.) unsigned char rxch (unsigned char \*Data);

Funcția recepționează un octet de pe linia serială, dacă recepția este posibilă (RI=1).

Se returnează valoarea 1 și se memorează octetul recepționat în Data dacă octetul a fost recepționat, 0 în cazul în care recepția nu s-a putut realiza.

e.) void txstr (unsigned char \*Text, unsigned miliseconds);

Funcția transmite pe interfața serială șirul de caractere Text (terminat cu '\0'), cu pauză programabilă între caractere, specificată în milisekunde prin parametrul milliseconds. Această pauză este necesară în cazul în care calculatorul este mai lent pentru ca să nu se piardă din octeții transmiși. De asemenea trebuie menționat că dacă în interiorul șirului de caractere se găsește '\n' pe ecranul calculatorului se va face afișarea pe rând nou.

f.) void init0 (void);

Funcția inițializează și pornește timer-ul T0 în modul 1 de lucru (timer pe 16 biți), generând întreruperi la 10 milisekunde. Dacă se folosește această funcție în program, atunci programul trebuie să conțină și rutina de tratare a întreruperii dată de timer-ul T0.

g.) void timerdelay (unsigned milliseconds);

Funcția generează o întârziere exprimată în milisekunde, programabilă prin parametrul de intrare, folosind pentru măsurarea întârzierii timer-ul T0. Întârzierea este exactă dacă milliseconds este un multiplu de 10 milisekunde. Pentru ca funcția să funcționeze corect, este necesar ca timer-ul T0 să fie pornit, iar în cadrul rutinei de întrerupere să se apeleze funcția:

delayupdate ();

h.) void updatet (void);

Această funcție trebuie să fie prima instrucțiune din rutina de întrerupere pentru timer-ul T0.

Ea reîncarcă valorile programate în registrele timer-ului la inițializare.

i.) void delayupdate (void);

Funcția trebuie apelată numai din rutina de întrerupere pentru timer-ul T0, dacă în programul de aplicație se folosește funcția: timerdelay().

Funcțiile modulului LIB\_LCD:

a.) void cursorhome (void);

Cursorul este mutat pe linia 1, în poziția caracterului 1.

b.) void blinkon (void);

Funcția activează afișarea cursorului de tip bloc caracter intermitent.

c.) void blinkoff (void);

Funcția dezactivează afișarea cursorului de tip bloc caracter intermitent.

d.) void cursoron (void);

Funcția activează afișarea cursorului de tip linie continuă (se afișează sub caracter).

e.) void cursoroff (void);

Funcția dezactivează afișarea cursorului de tip linie continuă.

f.) void displayon (void);

Funcția activează afișarea pe ecran a conținutului memoriei interne a afișorului.

g.) void displayoff (void);

Funcția dezactivează afișarea pe ecran a conținutului memoriei interne a afișorului. Mesajele nu se afișează pe ecran dar se încarcă în memoria internă a afișorului.

h.) void movecursor (unsigned char x);

Funcția mută cursorul în poziția x. Cursorul trebuie în prealabil să fie activat.

i.) void cright (void);

Funcția deplasează cursorul cu o poziție la dreapta.

j.) void cleft (void);

Funcția deplasează cursorul cu o poziție la stânga.

k.) void dright (void);

Funcția deplasează afișarea pe ecran cu o poziție caracter la dreapta în cadrul memoriei interne.

l.) void dleft (void);

Funcția deplasează afișarea pe ecran cu o poziție caracter la stânga în cadrul memoriei interne.

m.) void writestr (unsigned char x, unsigned char \*Buffer) reentrant;

Funcția afișează șirul de caractere Buffer, începând din poziția caracter x al liniei.

n.) void write (unsigned char \*Message);

Funcția afișează mesajul Message începând cu poziția caracter 1 a liniei.

o.) void clearscreen (void);

Funcția șterge ecranul afișorului.

p.) void initdisplay (void);

Funcția inițializează afișorul cu cristale lichide cu următorii parametri: interfață de 8 biți, format două linii, matrice caracter 5x7.

Funcțiile modulului LIB\_PORT:

a.) void buserf (unsigned char Frequency);

Funcția pornește buser-ul, cu un anumit ton stabilit prin parametrul Frequency.

De fapt se generează pe ieșirea PWM0 a microcontrolerului, un semnal periodic, dreptunghiular.

b.) void buseron (void);

Funcția pornește buser-ul prin apelul funcției buserf.

c.) void buseroff (void);

Funcția oprește buser-ul.

d.) void buserbeep (unsigned miliseconds);

Funcția generează un semnal sonor cu durata programabilă prin parametrul miliseconds.

Nu alterează sunetul buzzer-ului stabilit cu funcția buserf.

e.) void buserbeeps (unsigned miliseconds, unsigned char beeps);

Funcția generează un număr de beeps sunete cu durata programabilă prin miliseconds

f.) void buserblink (void);

Funcția permite generarea unor semnale intermitente pe buzzer. La fiecare apel se inversează starea buzzer-ului.

Funcțiile ale modulului LIB\_KEYB:

a.) void keyb\_InitKeyboard (bit beeps, bit autorepeat);

Funcția realizează inițializarea tastaturii.

Prin parametrul beeps se poate programa funcționarea în cazul apăsării unei taste invalidate prin masca KeyMask și anume: dacă beeps are valoarea 1, atunci la apăsarea tastei invalidate se generează trei beep-uri scurte. Dacă beeps este 0, atunci la apăsarea tastei invalidate nu se generează nici un sunet.

Prin parametrul autorepeat se poate programa funcționarea autorepeat-ului la apăsarea mai lungă a unei taste. Dacă autorepeat este 1 facilitatea este activată, altfel ea este dezactivată.

b.) unsigned keyb\_ScanKeyboard (unsigned KeysMask);

Funcția scanează tastatura pentru determinarea apăsărilor de taste. Funcția trebuie apelată în buclă.

Se consideră că o tastă a fost apăsată, dacă ea a fost apăsată pentru cel puțin 60 de miliseconde. Apăsările mai scurte se consideră că au fost perturbații și se ignoră.

Prin parametrul KeyMask se pot valida sau invalida anumite taste.

La apăsarea unei taste validate se generează un beep scurt. La apăsarea unei taste invalidate, funcționarea este stabilită la inițializarea tastaturii prin funcția keyb\_InitKeyboard.

Funcția returnează tasta apăsată (dar numai în cazul în care tasta respectivă este activată).

Nu este permisă apăsarea mai multor taste simultan.



---

Pentru ca funcția de scanare să funcționeze corect, este necesar ca timer-ul T0 să fie pornit, iar în cadrul rutinei de întrerupere să se apeleze funcția:

keyb\_Update();

c.) void keyb\_WaitKey (unsigned Key);

Funcția permite oprirea unui program până la apăsarea unei taste.

Dacă parametrul Key este 0, atunci funcția așteaptă apăsarea oricărei taste. Dacă parametrul Key este diferit de 0, atunci funcția așteaptă apăsarea tastei specificate prin Key.

d.) void keyb\_Update (void);

Funcția trebuie apelată numai din rutina de intrerupere pentru timer-ul T0, dacă în programul de aplicație se folosește tastatura. Ea asigură incrementarea contoarelor necesare pentru scanarea tastaturii.



---

## ANEXA 8. Programul pentru capturarea imaginii

```
#include "stdafx.h"
#include <vfw.h>

CAPSTATUS gCapStatus ;
CAPDRIVERCAPS gCapDriverCaps ;
CAPTUREPARMS gCapParms ;
HWND ghWndFrame;
HWND ghWndCap ;
HWND GetMainWindowHwnd(void);
HINSTANCE GetMainInstance(void);

/*
 * Enumerate the potential capture drivers
 * This function is only called once at startup.
 * Returns FALSE if no drivers are available.
 */

BOOL vidcapEnumerateDrivers(void)
{
    #define MAXVIDDRIVERS 10
    char achDeviceVersion[80];
    char achDeviceAndVersion[160];
    UINT ulIndex;
    short gDriverCount = 0;
    for (ulIndex = 0; ulIndex < MAXVIDDRIVERS; ulIndex++)
    {
        if (capGetDriverDescription(ulIndex,
            (LPSTR)achDeviceAndVersion,
            sizeof(achDeviceAndVersion),
            (LPSTR)achDeviceVersion, sizeof(achDeviceVersion)))
        {
            // Concatenate the device name and version strings
            lstrcat (achDeviceAndVersion, ", ");
            lstrcat (achDeviceAndVersion, achDeviceVersion);
            gDriverCount++;
        }
        else

```

```

        break;
    }
    return (gDriverCount);
}

/* Connect the capture window to a capture driver.
 * ulIndex specifies the index of the driver to use.
 * Returns TRUE on success, or FALSE if the driver connection
failed.
 */
BOOL vidcapInitHardware(HWND hwndCap, UINT ulIndex)
{
    char    szName[MAX_PATH];
    char    szVersion[MAX_PATH];
    BOOL    gbHaveHardware;

    // Try connecting to the capture driver
    if (!capDriverConnect(hwndCap, ulIndex))
        {
            MessageBox(NULL, "Error initializing
            video capture device", "Roboview", MB_OK);
            gbHaveHardware = FALSE;
        }
    else
        gbHaveHardware = TRUE;

    // Get the capabilities of the capture driver
    capDriverGetCaps(hwndCap, &gCapDriverCaps,
sizeof(CAPDRIVERCAPS));

    // Get the settings for the capture window
    capGetStatus(hwndCap, &gCapStatus, sizeof(gCapStatus));

    // Unlike all other capture drivers, Scrcap.drv needs to use
    // a Yield callback, and we don't want to abort on mouse
clicks,
    // so determine if the current driver is Scrcap.drv
    capGetDriverDescription (ulIndex,
        szName, sizeof (szName),
        szVersion, sizeof (szVersion));

```

```

        // Get video format and adjust capture window
        // vidcapLayout(ghWndMain);
        // InvalidateRect(ghWndMain, NULL, TRUE);
        // set the preview rate (units are millisecs)
        capPreviewRate(hwndCap, 33);
        capPreviewScale(hwndCap, TRUE);
        // set live/overlay to default

        capPreview(hwndCap, TRUE);
        return gbHaveHardware;
    }

    HWND vidframeCreate(int x,int y,int cx,int cy,HWND FAR *
    phwndCap)
    {
        HWND hwnd, hwndCap;
        WNDCLASS wc;
        static FirstTime = 1;
        if(FirstTime)
        {
            wc.lpszClassName = "VideoCapture";
            wc.hInstance = GetMainInstance();
            wc.lpfWndProc = DefWindowProc;
            wc.hCursor = LoadCursor(NULL,
            IDC_ARROW);
            wc.hIcon = NULL;
            wc.lpszMenuName = NULL;
            wc.hbrBackground = NULL;
            wc.style = CS_HREDRAW |
            CS_VREDRAW ;
            wc.cbClsExtra = 0 ;
            wc.cbWndExtra = 0 ;
            if(!RegisterClass(&wc))
                MessageBox(NULL,"Error registering the
                class","Roboview",MB_OK);
            FirstTime = 0;
        }

        hwnd = CreateWindow(
            "VideoCapture",

```

---

```

        "VideoCapture",
        WS_CHILD | WS_VISIBLE | WS_CLIPCHILDREN |
WS_THICKFRAME,
        x, y, cx, cy,

        GetMainWindowHwnd(),
        (HMENU) 0,
        GetMainInstance(),
        NULL);

/*
 * create an AVICAP window within this window. Leave
vidframeLayout
 * to do the layout
 */

        hwndCap = capCreateCaptureWindow(
            NULL,
            WS_CHILD | WS_VISIBLE,
            0, 0, cx, cy,
            hwnd,          // parent window
            1              // child window id
        );
        if (hwndCap == NULL) {
            return(NULL);
        }
        *phwndCap = hwndCap;
        return(hwnd);
    }

    void VideoCapture(void)
    {
        ghWndFrame =
vidframeCreate(652,32,146,110,&ghWndCap);

        if ((ghWndFrame == NULL) || (ghWndCap == NULL))
            MessageBox(NULL,"Window initialization
failure","Roboview",MB_OK);

        // Get the default setup for video capture from the AVICap

```

---

```
window
    BOOL test = capCaptureGetSetup(ghWndCap, &gCapParms,
    sizeof(CAPTUREPARMS));

    // Create a list of all capture drivers and append them to the
Options menu
    if (!vidcapEnumerateDrivers())
        MessageBox(NULL,"No video capture device
found","Roboview",MB_OK);
    // Try to connect to a capture driver
    else
    if (vidcapInitHardware(ghWndCap,0))
        {
            // Hooray, we now have a capture driver
connected!
            // vidcapSetCaptureFile(gachCaptureFile);
        }
        return;
    }

void MyFileSaveDIB(void)
{
    capFileSaveDIB(ghWndCap,"D:\\Frame.dib");
}

void MyEditCopy(void)
{
    capEditCopy(ghWndCap);
}

void MyDriverDisconnect(void)
{
    capDriverDisconnect(ghWndCap);
}

void MyDestroyWindow(void)
{
    DestroyWindow(ghWndFrame);
}
```





## ANEXA 9. Programul pentru comunicarea serială (selectiv).

```

void CMainFrame::OnToolsRealtime()
{
    CROBOVIEWDoc    *SourceDoc;
    unsigned char *ClipboardBuffer;
    short            j,x,y;
    MSG              msg;
    long             m00,m10;
    POINT            CentruGreutate;
    CDC              *ActiveFrameDC;
    char             sir[40];
    RECT             rect = {10,480,50,500};
    short            TypeOfCapture;
    BYTE             PixelGrayValue;

    unsigned char OutChar;
    DWORD           WrittenBytes;
    HANDLE           hCom;
    BOOL             fSuccess;
    DCB              dcb;
    short           count =0;
    for(;;)
    {
        if(kbhit())
            break;
        _outp(0x378,0x55);
        delay(5);
        _outp(0x378,0x99);
        delay(5);
        _outp(0x378,0xAA);
        delay(5);
        _outp(0x378,0x66);
        delay(5);
    }
    delay(500);
    _outp(0x378,0x00);
    return;
    hCom = CreateFile("LPT1",
        GENERIC_WRITE,

```

---

```

    0, // exclusive access
    NULL, // no security attrs
    OPEN_EXISTING,
    0, // not overlapped I/O
    NULL
);
if (hCom == INVALID_HANDLE_VALUE)
{
    ShowErrorMessage("COM2 device cannot be opened\n");
    return;
}
fSuccess = GetCommState(hCom, &dcb);
if (!fSuccess)
{
    ShowErrorMessage("Cannot get COM2 current
configuration\n");
    return;
}
dcb.DCBlength = sizeof(DCB);
dcb.BaudRate = 12358;
dcb.fBinary = TRUE;
dcb.fParity = FALSE;
dcb.fOutxCtsFlow = FALSE;
dcb.fOutxDsrFlow = FALSE;
dcb.fDtrControl = DTR_CONTROL_DISABLE;
dcb.fDsrSensitivity = FALSE;
dcb.fOutX = FALSE;
dcb.fInX = FALSE;
dcb.fRtsControl = RTS_CONTROL_DISABLE;
dcb.fAbortOnError = TRUE;
dcb.wReserved = 0;
dcb.ByteSize = 8;
dcb.Parity = NOPARITY;
dcb.StopBits = ONESTOPBIT;
fSuccess = SetCommState(hCom, &dcb);
if (!fSuccess)
{
    ShowErrorMessage("Cannot set COM2 new configuration\n");
    return;
}

```

```

OutChar = 0x05;
fSuccess = WriteFile(hCom,&OutChar,1,&WrittenBytes,NULL);
OutChar = 0x09;
fSuccess = WriteFile(hCom,&OutChar,1,&WrittenBytes,NULL);
OutChar = 0x0A;
fSuccess = WriteFile(hCom,&OutChar,1,&WrittenBytes,NULL);
OutChar = 0x06;
fSuccess = WriteFile(hCom,&OutChar,1,&WrittenBytes,NULL);
return;
hCom = CreateFile("COM2",
    GENERIC_READ | GENERIC_WRITE,
    0, // exclusive access
    NULL, // no security attrs
    OPEN_EXISTING,
    0, // not overlapped I/O
    NULL
);
if (hCom == INVALID_HANDLE_VALUE)
{
    ShowErrorMessage("COM2 device cannot be opened\n");
    return;
}
// Omit the call to SetupComm to use the default queue sizes.
// Get the current configuration.
fSuccess = GetCommState(hCom, &dcb);
if (!fSuccess)
{
    ShowErrorMessage("Cannot get COM2 current
configuration\n");
    return;
}
// Fill in the DCB: baud=9600, 8 data bits, no parity, 1 stop bit.
dcb.BaudRate = 9600;
dcb.ByteSize = 8;
dcb.Parity = NOPARITY;
dcb.StopBits = TWOSTOPBITS;
fSuccess = SetCommState(hCom, &dcb);
if (!fSuccess)
{
    ShowErrorMessage("Cannot set COM2 new configuration\n");
}

```

```

        return;
    }
    /* for(;;)
        {
            if(PeekMessage(&msg,NULL,0,0,PM_REMOVE))
                {
                    TranslateMessage(&msg);
                    DispatchMessage(&msg);
                    if(msg.message == WM_CHAR)
                        {
                            OutChar = msg.wParam;
                            fSuccess=WriteFile(hCom,&OutChar,1,&WrittenBytes,NULL);
                            if(!fSuccess || WrittenBytes != 1)
                                {
                                    ShowErrorMessage("Cannot write to COM2\n");
                                    return;
                                }
                        }
                }
        }
    OutChar = '8';
    fSuccess = WriteFile(hCom,&OutChar,1,&WrittenBytes,NULL);
    if(!fSuccess || WrittenBytes != 1)
        {
            ShowErrorMessage("Cannot write to COM2\n");
            return;
        }
    OutChar = '6';
    fSuccess = WriteFile(hCom,&OutChar,1,&WrittenBytes,NULL);
    if(!fSuccess || WrittenBytes != 1)
        {
            ShowErrorMessage("Cannot write to COM2\n");
            return;
        }
    ShowErrorMessage("");*/
    SourceDoc = GetActiveSourceDocument();
    /**/ if(SourceDoc->BitmapBuffer == NULL)
    /**/     SourceDoc->BitmapBuffer = (BYTE
*)malloc(DIBSIZE);

```

## ANEXA 10. Programul pentru capturarea, prelucrarea imaginii și urmărirea țintei.

```

//Bucla de captura si prelucrare
for(;;)
    {
        m00 = m10 = 0;
        // CaptureFrame(SourceDoc,FALSE);
        /**/ MyEditCopy();
        /**/ if(!MyOpenClipboard(NULL))
        /**/     {
        /**/         MessageBox("Error opening
clipboard","Roboview",MB_OK);
        /**/         CloseHandle(hCom);
        /**/         return;
        /**/     }
        /**/ if(!(ClipboardBuffer = (unsigned char
*)MyGetClipboardData(CF_DIB)))
        /**/     {
        /**/         ShowErrorMessage("Error reading clipboard
data\n");

                                MyDriverDisconnect();
                                MyDestroyWindow();
                                VideoCapture();
                                continue;

        /**/         //CloseHandle(hCom);
        /**/         //return;
        /**/     }
        /**/ memcpy(SourceDoc->BitmapBuffer,ClipboardBuffer,DIBSIZE);
        /**/ if(SourceDoc->BitmapBuffer != NULL)
        /**/     SourceDoc->BitmapBits =
GetDibBitsAddr(SourceDoc->BitmapBuffer);
        for(y=Ymax-1;y>=0;y--)
            for(x=0;x<Xmax;x++)
                {
                    if(TypeOfCapture = TRUECOLOR)
                        PixelGrayValue =
(BYTE)(0.3*(*(SourceDoc->BitmapBits+3*Xmax*(Ymax-1-y)+3*x)) +

```

```

0.59*(*(SourceDoc->BitmapBits+3*Xmax*(Ymax-1-y)+3*x+1))      +
0.11*(*(SourceDoc->BitmapBits+3*Xmax*(Ymax-1-y)+3*x+2));
    else
        PixelGrayValue = *(SourceDoc-
>BitmapBits+3*Xmax*(Ymax-1-y)+3*x);
        if(PixelGrayValue > 70)
            SourceDoc->BitmapMatrix[x][y][0]
=        SourceDoc->BitmapMatrix[x][y][1]      =        SourceDoc-
>BitmapMatrix[x][y][2] = 0xFF;
        else
            {
                SourceDoc->BitmapMatrix[x][y][0]
=        SourceDoc->BitmapMatrix[x][y][1]      =        SourceDoc-
>BitmapMatrix[x][y][2] = 0;
                m00++;
                m10 += x;
            }
        }
    if(m00 == 0)
        CentruGreutate.x = 160;
    else
        CentruGreutate.x = m10/m00;
    for(x=0;x<Xmax;x++)
        for(y=REALYMAX;y<Ymax;y++)
            SourceDoc->BitmapMatrix[x][y][0]      =
SourceDoc->BitmapMatrix[x][y][1] = SourceDoc->BitmapMatrix[x][y][2]
= 0xFF;
        SourceDoc->BitmapColorType = BINARY;
    /**/
    EmptyClipboard();
    /**/
    CloseClipboard();
    SourceDoc->UpdateAllViews(NULL);
    ActiveFrameDC = AfxGetMainWnd()->GetDC();
    itoa(160-CentruGreutate.x,sir,10);
    ActiveFrameDC->SetTextColor(RGB(255,0,0));
    ActiveFrameDC->SetBkMode(OPAQUE);
    ActiveFrameDC-
>ExtTextOut(10,480,ETO_OPAQUE,&rect,sir,strlen(sir),N
ULL);
    //count++;
    if(abs(160 - CentruGreutate.x) > 10)

```

```

        {
            if((160-CentruGreutate.x) > 0 /*&& count >= 10*/)
            {
                OutChar = '4';
                count = 0;
            }
            if((160-CentruGreutate.x) < 0 /*&& count >= 10*/)
            {
                OutChar = '6';
                count = 0;
            }
            fSuccess = WriteFile(hCom,&OutChar,1,&WrittenBytes,NULL);
            if(!fSuccess || WrittenBytes != 1)
            {
                ShowErrorMessage("Cannot write to
COM2\n");
                CloseHandle(hCom);
                return;
            }
        }
    for(j=0;j<100;j++)
    {
        if(PeekMessage(&msg,NULL,0,0,PM_REMOVE))
        {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
            if(msg.message == WM_CHAR)
            {
                if(msg.wParam != ' ')
                {
                    CloseHandle(hCom);
                    return;
                }
            }
            else
            {
                for(;;)
                {
                    if(PeekMessage(&msg,NULL,0,0,PM_REMOVE))
                    {

```





---

## BIBLIOGRAFIE

- \*\*\* AUTOMATION IN WARE HOUSING, Tokyo, 1987
- \*\*\* FTS in Meinungspiegel der Branche, Forder und Heben, RFG, nr. 11/1988
- \*\*\* Fuzzy Fundamentals, IEEE Spectrum, 1992
- \*\*\* Hewlet Packard, Optoelectronics Designer's Catalog, 1988-1989
- \*\*\* NDC Lazerway, Netzler & Dahlgren Co AB, Särö, Schweden, 1995
- \*\*\* NDC News, Netzler & Dahlgren Co AB (nr. 6), Särö, Schweden, 1995
- \*\*\* Proceedings of the 3<sup>rd</sup> International Conference on AUTOMATED HANDLING, Birmingham, UK, 19-21 march 1986
- \*\*\* SGS-Thompson Microelectronics, Industrial and Computer Peripheral Ics. Databook, 1<sup>st</sup> Edition
- [AAMA] Ahmad A. Masoud, Using Hybrid Vector - Harmonic Potential Fields for Multi-Robot, muti-Target navigation in a Stationary Environment, IEEE, 1996, p. 3564-3569.
- [ACFA] An-Hui Cai, Toshio Fukuda, Fumihito Arai, Hierarchical Control Architecture for Cellular Robotic System - Simulations and Experiments, IEEE, 1995, p.1191-1196.
- [ACKK] A.Curran, K.J.Kyriakopoulos, Sensor - Based Self - Localization for Wheeled Mobile Robots, IEEE, 1993, p. 8-13.
- [AKMK] Akio Kosaka, Min Meng, A.C.Kak, Vision - Guided Mobile robot Navigation using Retroactive Updating of Position Uncertainty, IEEE, 1993, p.1-7.
- [ALG 99] Allegro Data Sheet, 1999, (Documentație obținută de pe Internet).
- [AMLK] Alistair McLean, Stephen Cameron, Snake - Based Path Planning for Redundant Manipulators, IEEE, 1993, p.275-276.
- [AMLM] Alistair McLean, Isabelle , Incremental Roadmaps and Global Path Planning in Evolving Industrial Environments, IEEE, 1996, p. 101-103.
- [AMMB] Ahmad A.Masoud, Mohamed M.Bayoumi, Robot Navigating using the Vector Potential Approach, IEEE, 1993, p. 805-810.
- [AMYY] Akira Mohri, Xiang Dong Yang, Motoji Yamamoto Collision Free Trajectory Planning for Manipulator using Potential Function, IEEE, 1995, p. 3069-3074.
- [AND ME] Andreiciuc D., Meszáros A., Sisteme de transport cu vehicule ghidate automat, Simpozionul Tehnic 2000.
- [AND MS] Andreiciuc D., Mészáros, A., Sistem flexibil automatizat de transport uzinal pentru piese grele de gabarit mare, Simpozionul Tehnic 2000.

- [AND 83a] Andreiciuc D., Bogdanov I., Popescu V., Comanda prin microprocesor a unui chopper de putere utilizat în acționarea cu motoare de c.c., Simpozionul Aplicații ale electronicii industriale, 17-18 nov. 1983, Craiova, pag. 6-13
- [AND 83b] Andreiciuc D., Bogdanov I., Tănase M., Instalație electronică pentru acționarea cu motoare pas cu pas a unui robot industrial, Simpozionul Aplicații ale electronicii industriale, 17-18 nov. 1983, Craiova, pag. 14-20
- [AND 83c] Andreiciuc D., Bogdanov I., Popescu V., Redresor cu tiristoare comandat cu microprocesor, Simpozionul Aplicații ale electronicii industriale, 17-18 nov. 1983, Craiova, pag. 21-26
- [AND 83d] Andreiciuc D., Bogdanov I., Popescu V., Instalație electronică pentru măsurarea și reglarea turăției unui motor de c.c. fără tohogenerator, Simpozionul Aplicații ale electronicii industriale, 17-18 nov. 1983, Craiova, pag. 102-106
- [AND 84a] Andreiciuc D., Bogdanov I., Popescu V., DC motor speed adjusting with microprocesor, Bul. Șt. și tehn. al IPTVT, Tom. 29 (43), Fasc. 1-2, ian.-dec. 1984, pag. 73-74
- [AND 84b] Andreiciuc D., Bogdanov I., Popescu V., Circuite electronice pentru acționarea cu motoare pas cu pas a unui robot industrial, Al IV-lea Simpoz. Robotizarea în industrie, vol. IV, pag. 1-4, Timișoara, 29.XI-2.XII.1984
- [AND 84c] Andreiciuc D., Popescu V., Balaci I., Bogdanov I., Variator de turăție pentru un motor c.c. dintr-o instalație de vopsire a produselor textile, Al III-lea simpoz. Național de Teoria sistemelor, sect. A II-a, vol. I, pag. 341-317, Craiova, 1984
- [AND 85a] Andreiciuc D., Popescu V., Bogdanov I., Dispozitiv electronic pentru măsurarea distanțelor cu ultrasunete, Simpozionul național de Roboți Industriali, Cluj Napoca, 1988.
- [AND 85b] Andreiciuc D., Bogdanov I., Circuite electronice pentru acționarea cu motoare pas cu pas a unui robot industrial, Eficiență și rentabilitate în fabricarea și exploatarea mașinilor și aparatelor electrice, EMTimișoara, 1985, p.232-239.
- [AND 89a] Andreiciuc D., Mészáros, A., Roboții mobili, roboții viitorului, Simpozionul de Roboți industriali, Baia Mare 1989
- [AND 89b] Andreiciuc D., Mészáros, A., Evaluarea comparativă a unor tehnologii pentru sisteme de transport cu vehicule ghidate automat, Simpozionul de Roboți industriali, Baia Mare, 1989.
- [AND 94] Andreiciuc Dan, Metode și mijloace pentru orientarea și poziționarea vehiculelor ghidate automat, Referat 1-Doctorat, Timișoara, 1994.

- [AND 95] Andreiciuc Dan, Tendințe actuale în realizarea sistemelor de conducere pentru vehicule ghidate automat, Referat 2 - Doctorat, Timișoara, 1995.
- [ANG 92] Angerbauer R., FTS Anwenderprogrammierung in IRL., FTS Fachtagung Dortmund, 1992.
- [ANGM] A.Noble, R.Gupta, J., CAD - Based Inspection using X - Ray Stereo, IEEE, 1996, p.2361-2366.
- [APHA] Anton Pîl, Haruhiko Asada A, Computer Integrated Development System for High Speed robot Design Incorporating Structural Reconfiguration, IEEE, 1993, p.677-678.
- [APPM] Angel P.del Pobil, Miguel Perez, Begona Martinez, A Practical Approach to Collision Detection between General Objects, IEEE, 1996, p.779-784.
- [ARB 95] Magnetsensoren, Firmenschrift der Firma Autonome Roboter GmbH, Hamburg.
- [ASAI] Asai K., On present state of practical research of fuzzy systems theory and fuzzy robot, Japan Robot Soc., 1988
- [ATAR] Abdelhamid Tayebi, Ahmed Rachid, A Time - varying - based Robust Control for the Parking Problem of a Wheeled Mobile robot, IEEE, 1996, p. 3099-3104.
- [AUC 88] Auch W., Optischer Rotationssensoren, tm Technischer Messen 52 (1985) 5, p. 199-207.
- [AWCM] Aaron S.Wallack, John F.Canny, Dinesh Manocha, Object Localization Using Crossbeam Sensing, IEEE, 1993, p. 692-699.
- [BAC] Bachmann O., Neue Entwicklungen bei Fahrerlosen Transportsystemen, Förder und Heben nr. 6/1989
- [BBAR] Billur Barshan, Location and Curvature Estimation of "Spherical"Targets using a Flexible Sonar Configuration, IEEE, 1996, p.1218-1223.
- [BBGC] A. Balluchi, A.Bicchi, G.Casalino, Path Tracking Control for Dubin's Cars, IEEE, 1996, p.3123-3128.
- [BBYX] H.Benjamin Brown,Jr., Yangsheng Xu, A Single - Wheel, Gyroscopically Stabilized Robot, IEEE, 1996, p. 3658-3663.
- [BO] Johann Borenstein, Multi - Layered Control of a Four - Degree - of - Freedom Mobile Robot with Compliant Likage
- [BSLP] J.Borges Sousa, F.Lobo Pereira, On the Design and Implementation of a Control Architecture for a Mobile Robotic System, IEEE, 1996, p.2822-2827.
- [BUI 91] Buitkamp H., Drechmessungen ohne Achse und Teilkreis, Roboter 9, 1991, p.26-28.
- [BYAM 96] Brian Yamauchi, Mobile Robot Localization in Dynamic Environments using Dead Reckoning and Evidence Grids, IEEE, 1996, p.1401-1406.

- [CBST] Craig Becker, J.Salas, Kentaro Tokusei, Reliable navigation using Landmarks, IEEE, 1995, p.401-406.
- [CDW] Jonathan M.Cameron, Douglas C. MacKenzie, Reith R., Reactive Control for Mobile Manipulation
- [CONG] Chong Jin Ong, Distance Computation between Smooth Convex Objects, IEEE, 1996, p. 785-790.
- [CPMM] Claude Pegard, El Mustapfa Mouaddib, A Mobile Robot using a Panoramic View, IEEE, 1996, p.89-94.
- [CTDK] Camillo J.Taylor, David J. Kriegman, Exploration Strategies for Mobile Robots, IEEE, 1993, p.248-253.
- [CWYT] D.Crawford, F.Y.Wong, K.Youcef-Toumi, Modeling and Design of a Sensor for Two Dimensional Linear Motors, IEEE, 1996, p.2367-2372.
- [CZFB] C.S.Zhao, M.Farooq, M.M.Bayoumi, Collision - free Path Planning for a Robot with Two Arms Cooperating in the 3 - D Work Space
- [DAHS] Dimitrios Apostolopoulos, Hagen Schempf, Mobile Robot for Automatic Installation of Floor Tiles, IEEE, 1996, p. 3652-3657.
- [DBES] Daniel L.Boley, Erik S.Steinmetz, Robot Localization from Landmarks using Recursive Total Least Squares, IEEE, 1996, p.1381-1386.
- [DDTL] David Dai, Daryl T.Lawton, Range - Free Qualitative Navigation
- [DIC] Dickmanns E.D., Automonous high speed road vehicle guidance bz computer vision
- [DIN15185] 08.91 Lagersysteme mit leitliniengeführten Flurförderzeugen, Anforderungen an Boden, Regal und sonstige Anforderungen
- [DIN18225] 06.88 Industriebau, Verkehrswegw in Industriebauten
- [DIN51953] 08.75 Prüfung von organischen Bodenbelägen, Prüfung der Ableitfähigkeit für elektrostatische Ladungen in Bodenbeläge in explosionsgefährdeten Räumen
- [DKRA] Don Krantz, Non - Uniform Dead - Reckoning Position Estimate Updates, IEEE, 1996, p.2061-2066.
- [DLAN] Dirk Langer An Integrated MMW Radar System for Outdoor Navigation, IEEE, 1996, p. 417-422.
- [DOIS] T. D'Orazio, M.Ianigro, E.Stella Mobile Robot Navigation by Multi - Sensory Integration, IEEE, 1993, p.373-379.
- [DRA 95] Doina Drăgulescu, Mirela Toth -Tașcău, Planificarea mișcării roboților industriali, Universitatea Tehnică Timișoara, 1995.
- [DRU 90] Drunk G., Sensor und Steuerungssystem für die leitlinienlose Führung automatischer Flurförderzeuge, Springer Verlag, Berlin, 1990.

- [DWBA] Hugh Durrant-Whyte, Edward Bell, Philip Avery, The Design of a Radar - Based Navigation System for Large Outdoor Vehicles, IEEE, 1995, p.764-769.
- [DWS] B. Dacre Wrifght, T. Simeon, Free Space Representation for a Mobile Robot moving on a Rough Terrain, IEEE, 1993, p.37-43.
- [EGAT 96] Erran Gat, Navigation Templates: Enhancements, Extensions and Experiments, IEEE, 1993, p.567-571.
- [EKMH] Eric Krotkov, Martial Hebert, Mapping and Positioning for a Prototype Lunar Rover, IEEE, 1995, p.2913-2919.
- [ELB91] Elbracht, D. (Hrsg.), Die Steuerungsstruktur des FTS bestimmt die Zukunft der Fertigung und der Produktionslogistik, 1. Duisburger FTS-Fachtagung, 27. September 1991, Duisburg, Fertigungstechnisches Labor 1993
- [EPDR] E.G.Papadopoulos, D.A.Rey, A New Measure of Tipover Stability Margin for Mobile Manipulators, IEEE, 1996, p. 3111-3116.
- [EPEM] Erwin A.Prassler, Evangelos E.Milios, Position Estimation Using Equidistance Lines, IEEE, 1995, p.85-92.
- [ERW 95] MSV 100 Positionsscanner, Datenblatt der Firma Erwin Sick GmbH, 1995.
- [EYYA] Eiichi Yoshida, Masakazu Yamamoto, Tamio Arai, A Design Method of Local Communication Area in Multiple Mobile Robot System, IEEE, 1995, p. 2567-2572.
- [FGD 96] Berurungslos wirkendes Sicherheitssystem für automatische Flurförderzeuge, Datenblatt der Fraunhofer Gesellschaft Dortmund, 1996.
- [FHYA] Fumio Hara, Yuji Yasui, Toshiyuki Aritake Kinematic Analysis of Locomotive Cooperation for Two Mobile Robots along a General Wavy Road, IEEE, 1995, p.1197-1204.
- [FLEM] Feng Lu, Evangelos E.Milios, Optimal Global Pose Estimation for Consistent Sensor Data Registration, IEEE, 1995, p.93-100.
- [FLYH] Fang-Chang Lin, Jane Yung-jen Hsu, Cooperation and Deadlock - Handling for an Object - Sorting Task in a Multi - agent Robotic System, IEEE, 1995, p.2580-2585.
- [FOJ 91] Laser Navigation für fahrerlose Transportfahrzeuge, Formermittel Journal 23, 1991, p.18-21.
- [FPYW] Francois G.Pin, Yutaka Watanabe, Using Fuzzy Behaviors for the Outdoor Navigation of a Car with Low - Resolution Sensors, IEEE, 1993, p. 548-553.
- [FRF 95] Fahrerlose Transportsysteme, Firmenschrift der Firma Fritz Fuss GmbH, Albstadt.
- [GBDI] Errann Gat, Albert Behar, Rajiv Desai, Robert Ivlev, Behavior Control for Planetary Exploration: Interim Report

- [GDCZ] Grogory Dudek, Chi Zhang, Vision - Based Robot Localization without Explicit Object Models, IEEE, 1996, p.76-82.
- [GIR 84] Giralt G., Mobile robots - Robotics and Artificial Intelligence, Springer-Verlag, 1984
- [GLGO] Gunnar Lindstedt, Gustaf Olsson, Using Ultrasonics for Sensing in a Robotic Environment, IEEE, 1993, p.671-676.
- [GMNC] P. Di Giamberardino, S. Monaco, D. Normand-Cyrot, Digital Control through Finite Feedback Discretizability, IEEE, 1996, p.3141-3142.
- [GOVU] Giuseppe Oriolo, Marilena Venditteli, Giovanni Ulivi, On - Line Map Building and Navigation for Autonomous Mobile Robots, IEEE, 1995, p.2900-2906.
- [GSMA] E. Gonzales, A. Suarez, C. Moreno, F. Artigue, Complementary Regions: a Surface Filling Algorithm, IEEE, 1996, p.909-914.
- [GTGG] G.H.Tarbox, S.N.Gottschlich, L.A.Gerhardt, Registration of Dissimilar Featureless Models for Inspection, IEEE, 1993, p. 686-691.
- [HALK] Huzefa Akbarally, Lindsay Kleeman, A Sonar Sensor for Accurate 3 D Target Localization and Classification, IEEE, 1995, p. 3003-3008.
- [HALK 96] Huzefa Akbarally, Lindsay Kleeman, 3 D Robot Sensing from Sonar and Vision, IEEE, 1996, p. 686-691.
- [HAM 86a] Hammond, Gary C., AGVS at work; automated guided vehicle systems, Kempston, Bedford; IFS (Publications) (u.a.), vol.I, 1986
- [HAM 86b] Hammond, Gary C., AGVS at work; automated guided vehicle systems, Kempston, Bedford; IFS (Publications) (u.a.), vol.II, 1986
- [HAN 92] Hanebeck Uwe D., A New High Performance Multisonar System For Fast Mobile Robots, Technische Universität München, 1992
- [HCEK] Fabio Cozman, Eric Krotkov, Robot Localization using a Computer Vision Sextant, IEEE, 1995, p.106-111.
- [HOC 90] Hock C., Projekt zur Landmarkennavigation, Selbstverlag, 1990, p.39-56.
- [HEL 91] Heller J, Modulare Steuerungsstruktur, VDI Berichte, Dusseldorf, 1990,p.207-219.
- [HEL 96] Heller J, Fahrerlose Transportsysteme, Dusseldorf, 1996.
- [HGB 89] Kraftbetriebene Flurförderzeuge, Hauptverband der gewerblichen Berufsgenossenschaft, Köln, 1989.
- [HST96] Hiroshi Ishiguro, Saburo Tsuji, T - Net for Navigation a Vision - Guide Robot in a Real World, IEEE, 1995, p.1068-1073.
- [HAY 90] Hayashi S., A Tuning Method of Fuzzy Control System, Tokyo, 1990
- [HIR 85] Hirota K., Fuzzy Control and Intelligent Robot, Japan, 1985

- [HIUT96] Hiroshi ISHIGURO, Kenji UEDA, Saburo TSUJI, Omnidirectional Visual Information for Navigating a Mobile Robot, IEEE, 1993, p. 799-804.
- [HJKK] Hyoung Jo Jeon, Byung Kook Kim, A Study on World Map Building for Mobile Robots with Tri - Aural Ultrasonic Sensor System, IEEE, 1995, p. 2907-2912.
- [HKKC] Joong Hyup Ko, Seung Do Kim, Myung Jin Chung, A Method of Indoor Mobile Robot Navigation using Acoustic Landmarks, IEEE, 1996, p. 1726-1731.
- [HKPL] Sun-Gi Hong, Sung-Woo Kim, Kang-Bark Park, Ju-Jang Lee, Local Motion Planner for Nonholonomic Mobile Robots in the Presence of the Unknown Obstacles, IEEE, 1996, p. 1212- 1217.
- [HMSK] Hideo Mori, Shinji Kotani, Danger Estimation of Vehicles at Intersection, IEEE, 1995, p.781-787.
- [HOL 87] Hollier Robert H., (Hrsg.), Hollier Robert Henry, Automated guided vehicle systems (International trends in manufacturing technology; 11), Kempston, Bedford; IFS (Publ.) (u.a.), 1987
- [HPJC] H.Peremans, J.Van Campenhout, Tri-aural Perception on a Mobile Robot, IEEE, 1993, p. 265-269.
- [HVB 88] van Brussel H., FROG free ranging on grid, Springer Verlag Berlin, 1988, p.223-232.
- [IAF 98] Ianusz Furtac, Calibration of Moving Camera, Congress ISMCR'98, Prague, 1998, p. 359-365.
- [IPCL] Igor E.Paromtchik, Christian Laugier, Motion Generation and Control for Parking an Autonomous Vehicle, IEEE, 1996, p.3117-3122.
- [ISRR] Ishay Kamon, Ehud Rivlin, Elon Rimon, A New Range-Sensor Based Globally Convergent Navigation Algorithm for Mobile Robots, IEEE, 1996, p. 429-435.
- [JAN 90] Jantzer M., Bahnverhalten Und Regelung fahrerloser Transportsysteme ohne Spurbindung, Springer Verlag, Berlin, 1990.
- [JBLF] J,Borenstein, L,Feng, Gyrodometry: A New Method for Combining Data from Gyros and Odometry in Mobile Robots, IEEE, 1996, p. 423-428.
- [JDWZ] Jaydev Desai, Chau-Chau Wang, Milos Zefran, Motion Planning for Multiple Mobile Manipulators, IEEE, 1996, p.2073-2074.
- [JHGS] Joachim Horn, Günther Schmidt, Continuous Localization for Long - Range Indoor Navigation of Mobile Robots, IEEE, 1995, p.387-394.
- [JLLC] Jeong-Hoon Lee, Bum-Hee Lee, Myoung Hwan Choi, A Real Time Traffic Control Scheme for a Multiple AGV System, IEEE, 1996, p.1625-1630.



- [JLMC] John J. Leonard, Bradley A. Moran, Ingemar J. Cox, Underwater Sonar Data Fusion using an Efficient Multiple Hypothesis Algorithm, IEEE, 1995, p.2995-3002.
- [JON 87] J.P. Ostrowski, J.W. Burdick, Designing Feedback Algorithms for Controlling the Periodic Motions of Legged Robots, IEEE, 1987, p.260-266.
- [JSOY] Jun Sasaki, Jun Ota, Eiichi Yoshida, Cooperating Grasping of a Large Object by Multiple Mobile Robots, IEEE, 1995, p.1205-1210.
- [JSPL] J. Borges Sousa, F. Lobo Pereira, A General Control Architecture for Multiple Vehicles, IEEE, 1996, p. 692-697.
- [JTFE] John T. Feddema, Kinematically Optimal Robot Placement for Minimum Time Coordinated Motion, IEEE, 1996, p. 3395- 3396.
- [JVAF] J. Vaganay, M.J. Aldon, A. Fournier, Mobile Robot Attitude Estimation by Fusion of Inertial Data, IEEE, 1993, p. 277-282.
- [JVBX] J. Vandorpe, H. Van Brussel, H. Xu, Exact Dynamic Map Building for a Mobile Robot using Geometrical Primitives Produced by a 2 D Range Finder, IEEE, 1996, p. 901-908.
- [JWAN] Jing Wang, Operating Primitives Supporting Traffic Regulation and Control of Mobile Robots under Distributed Robotic Systems, IEEE, 1996, p.1613-1618.
- [JWPT] Jing Wang, Suparek Premvuti, Abdullah Tabbara, A Wireless Medium Access Protocol (CSMA / CD - W) for Mobile Robot Based Distributed Robotic Systems, IEEE, 1995, 2561-2566.
- [JWSP] Jing Wang, Suparek Premvuti, Distributed Traffic Regulation and Control for Multiple Autonomous mobile Robots Operating in Discrete Space, IEEE, 1996, p.1619-1624.
- [JXLZ] Jing Xiao, Lixin Zhang, Computing Rotation Distance between Contacting Polyhedra, IEEE, 1996, p. 791-797.
- [JYSS] J.D. Yoder, S.B. Skaar, Cue verification for an Automatically - Guided Wheelchair System
- [KCW] Kung Chris Wu, A Sensor - Based Direct Search Algorithm for Autonomous Path Planning on 3 - D Terrain Surfaces, IEEE, 1993, p.236-241.
- [KDM] Yutaka Kanayama, David Mac Pherson, Gary Krahn, Two Dimensional transformations and Its Application to Vehicle Motion Control and Analysis
- [KHFH] Daehee nKANG, Hideki HASHIMOTO, Fumio HARASHIMA, Collision Free Navigation using Heuristic Decision Rule, IEEE, 1996, p. 3551-3556.
- [KHNR] Ken Hughes, N. Ranganathan, A Model for Determinating Sensor Confidence, IEEE, 1993, p.136-141.



- [KKLK] K.R.Kim, J.C.Lee, J.H.Kim, Dead - Reckoning for a Two - Wheeled Mobile Robot on Curved Surfaces, IEEE, 1996, p.1732-1737.
- [KKNK] K.J.Kyriakopoulos, N.J.Krikelis, Navigation of Nonholonomic Vehicles in Complex Environments with Potential Fields and Tracking, IEEE, 1996, p. 3389-3394.
- [KOV 94a] Kovacs Francisc, Sisteme de fabricație flexibilă robotizată, vol.I, Lito UPT, Timișoara, 1994.
- [KOV 94b] Kovacs Francisc, Sisteme de fabricație flexibilă robotizată, vol.II, Lito UPT, Timișoara, 1994.
- [KOV 86a] Kovacs Francisc, Manipulatoare, roboți și aplicațiile lor industriale, vol 1, Editura Facla, Timișoara, 1986.
- [KOV 86b] Kovacs Francisc, Manipulatoare, roboți și aplicațiile lor industriale, vol 2, Editura Facla, Timișoara, 1986.
- [KOV 98] Kovacs Francisc, Calculul unui mecanism de virare, Note de curs, Timișoara, 1998.
- [KOB 92] Kobbing h., Berührunglose Geschwindigkeitsermittlung, Selbstverlag FLW., Dortmund, 1992, p.7.73-7.80.
- [KNSY] Keiji Nagatani, Shinichi Yuta, Designing Strategy and Implementation of Mobile Manipulator Control System for Opening Door, IEEE, 1996, p.2828-2834.
- [KSKF] Karansher Singh, Kikuo Fujimura, Map Making by Cooperating Mobile Robots, IEEE, 1993, p.254-259.
- [KSTF] Kosuke Sekiyama, Toshio Fukuda, Modeling and Controlling of Group Behavior Based on Self-Organizing Principle, IEEE, 1996, p.1407-1408.
- [KTJO] Kenji TASHIRO, Jun OTA, Design of the Optimal Arrangement of Artificial Landmarks, IEEE, 1995, p.407-413.
- [KUG 90] Kugler h., Navigation und Führung von frei fahrenden FTS, f+h fordern und heben 40 ,4, 1990, p.235-237.
- [KUIA] Kazunori UMEDA, Kenji IKUSHIMA, Tamio ARAI, Fusion of Range Image and Intensity Image for 3D Shape Recognition, IEEE, 1996, p. 680-685.
- [KVG 90] Kugler h., Laseroptischer Kollisionsschutzsensor, f+h fordern und heben 40, 11, 1990, p.781-784.
- [KMYM] Kazumasa Yamazawa, Yasushi Yagi, Masahiho Yachida, Obstacle Detection with Omnidirectional Image Sensor HyperOmni Vision, IEEE, 1995, p.1062-1067.
- [KYNM] Keiichi Yamada, Tomoaki Nakano, Shin Yamamoto, Wide Dynamic Range Vision Sensor for Autonomous Vehicles, IEEE, 1995, p.770-775.
- [LAT 91] Latombe J.C., Robot Motion Planning, Kluwer Academic Publishers, Norwell - Massachusetts, 1991.

- [LEU 97] Leuze Electronic, Kollisionsschutz mittels Optischen Systemen, 1997.
- [LFCR] L.Ferriere, G.Campion, B.Raucent, Design of Omnimobile Robot Wheels
- [LHTS] Lang Hong, Todd Scaggs, Real - Time Optimal Multiresolutional Sensor / Data Fusion, IEEE, 1993, p.117-122.
- [LIN] Lindsay Kleeman, A Low Sample Rate 3 D Sonar Sensor for Mobile Robots
- [LKAV] Lydia Kavraki, Computation of Configuration - Space Obstacles Using the Fast Fourier Transform, IEEE, 1993, p.255-256.
- [LPWE] Leonard P.Wesley, Autonomous Locative Reasoning:an Evidential Approach, IEEE, 1993, p. 700-707.
- [LSJL] Luis Sobral, Rolf Johansson, Gunnar Lindsetdt, Ultrasonic Detection in Robotic Environments, IEEE, 1996, p. 347-352.
- [LUZ 95] Luy, J., Merklinger, A., Gerland, E., Die FTS-Technik im Umbruch - Sensorik zur leitlinienlosen FTS-Führung, in: Fördermittel Journal 7 (1995) Sonderpublikation Fördertechnik '95, S. 12/16
- [MACA] M.Abdulghafour, T.Chandra, M.A.Abidi, Data Fusion Through Fuzzy Logic Applied to Feature Extraction from Multi - Sensory Images, IEEE, 1993, p.359-366.
- [MAE 92] Maeda M., Forecast Learning Fuzzy Control of an Autonomous Mobile Robot, Fukuoka, Japan, 1992
- [MAN 74] Mandani E.H., Applications of Fuzzy Algorhythm for Control of Simple Dynamic Plant, IEEE, 1974
- [MAT 98] Mateiaş M., Mijloace și metode de determinare a parametrilor calitativi ai roboților industriali. Criterii de apreciere și diagnoză, Universitatea "Politehnica" din Timișoara, Teză de doctorat, 1998.
- [MBJP] Markus Buchberger, Klaus-Werner Jörg, Ewald von Puttkamer, Laserradar and Sonar Based World Modeling and Motion Control for Fast Obstacle Avoidance of the Autonomous Mobile Robot MOBOT - IV, IEEE, 1993, p. 534-540.
- [MCDC] Michael E.Cleary, Jill D. Crisman, Canonical Targets for Mobile Robot Control by Deitic Visual Servoing, IEEE, 1996, p.3093-3098.
- [MCHL] Myoung Hwan Choi, Bum Hee Lee, A Real Time Optimal Load Distribution for Multiple Cooperating Robots, IEEE, 1995, p.1211-1216.
- [MDA] Martin D.Adams, Amplitude Modulated Optical range Data Analysis in Mobile Robotics, IEEE, 1993, p.8-13.
- [MEH 91] Mehlin G., Europaeische Anstrengungen auf dem Gebiet Offener Steuerungsstrukturen furmobile Systeme, Selbstverlag, Duisburg, 1991, p.14-31.

- [MGS] Marc G. Slack, Fixed Computation Real - Time Sonar Fuzion for Local Navigation, IEEE, 1993, p.123-129.
- [MKRL] Michael G.Kay, Ren C.Luo, Global Vision for the Control of Free - Ranging AGV Systems, IEEE, 1993, p.14-19.
- [MOKY] M.Oda, K.Kibe, F.Yamagata, ETS - VII, Space Robot In - Orbit Experiment Satellite, IEEE, 1996, p. 739-744.
- [MPPD] Michel Par ent, Pascal Daviet, Automated urban vehicles: towards a dual mode PRT (Personal Rapid Transit), IEEE, 1996, p.3129-3134.
- [MSTP] N.M.Sgouros, P.Tsanakas, G.Papakonstantinouy, Localized Qualitative Navigation for Indoor Environments, IEEE, 1996, p. 921-922.
- [MHBH] Myoung Hwai Choi, Bum Hee Lee, A Real Time Optimal Load Distribution for Multiple Cooperating Robots, IEEE, 1995, p. 1211-1216.
- [MIL] Miller Richard K., Miller Richard Kendall, Automated guided vehicle systems, Medison, Ga.; SEAI Technical Publications
- [MIL 83] Miller Thomas W., Automated guided vehicle, Kempston, Bedford, IFS (Publications) (u.a.), 1983
- [MOR 80] Moravec H.P., Obstacle avoidance and navigation in the real world by seeing robot rover, Stanford AI-Lab, 1980
- [MTAR] M. Tarokh, Implementation of a Fast Path Planner on an Industrial Manipulator, IEEE, 1996, p. 436-440.
- [MUN 92] Muntermann H., Entwicklung und Einsatz eines frei navigierenden FTS Schleppers, Logistik und Unternehmen 6, 1992, p.46-49.
- [MWSM] Masayoshi Wada, Shunji Mori, Holonomic and Omnidirectional Vehicle with Conventional Tires, IEEE, 1996, p.3671-3676.
- [NHTS] J.Neira, J.Horn, J.D.Tardos, G.Schmidt, Multisensor Mobile Robot Localization, IEEE, 1996, p. 673-679.
- [NIT 97] Niţulescu Mircea, Roboţi mobili cu c ai de ghidare, Tez a de doctorat, Craiova, 1997.
- [NIS 92] Nishimori K., Comparision of Several Fuzzy Reasoning Methods on Driving Control of a Model Car, Iizuka, Japan, 1992
- [NRDF] Nicholas Roy, Gregory Dudek, Paul Freedman, Surface Sensing and Classification for Efficient Mobile Robot Navigation, IEEE, 1996, p. 1224-1228.
- [NUKS] Naohiro Ueno, Makoto Kaneko, Mikhail Svinin, Theoretical and Experimental Investigation on Dynamic Active Antenna, IEEE, 1996, p.3557-3563.
- [NZJC] Nan Zeng, Jill D.Crisman, Categorical Color Projection for Robot Road Following, IEEE, 1995, p.1080-1190.

- [OBHS] Otman A.Basir, Helen C.Shen, Goal - Driven Task Assignment and Sensor Control in Multi - Sensor Systems: A Multicriteria Approach, IEEE, 1993, p.559-566.
- [OHAP] Olivier Habert, Alan Pruski, Dynamic Modeling of an Indoor Environment
- [OPLP] O.Pinchart, A.Liegeois, F.Pouget, Generalized Polar Polynomials for Vehicle Path Generation with Dynamic Constraints, IEEE, 1996, p. 915-920.
- [prEN1525] 11.94 Sicherheit von Maschinen - Fahrerlose Flurförderzeuge und ihre systeme
- [prEN1526] 11.94 Sicherheit von Maschinen - Automatische Funktionen für Flurförderzeuge
- [PBG] Ph.Bonnifait, G.Garcia, A Multisensor Localization Algorithm for Mobile Robots and its Real - Time Experimental Validation, IEEE, 1996, p.1395-1400.
- [PBLB] Parag H.Batavia, M.Anthony Lewis, George A.Bekey, A Reduced Complexity Vision System for Autonomous Helicopter Navigation, IEEE, 1995, p.776-780.
- [PCIM 90] Power Conversion and Intelligent Motion, Oct. 1998, p.306-311.
- [PGLM] Pierrick Grandjean, Larry Matthies, Perception Control for Obstacle Detection by a Cross - country Rover, IEEE, 1993, p.20-27.
- [PHBC] P.Hebert, S.Betge-Brezetz, R.Chatila, Decoupling Odometry and Exteroceptive Perception in Building a Global World Map of a Mobile Robot: The Use of Local Maps, IEEE, 1996, p. 757-764.
- [PNDW] Daniel Pagac, Eduardo M. Nebot, Hugh Durrant-Whyte, An Evidential Approach to Probabilistic Map - Building, IEEE, 1996, p. 745-750.
- [PPDB] Polly K.Pook, Dana H.Ballard, Recognizing Teleoperated Manipulations, IEEE, 1993, p.578-585.
- [PPRO] Nicholas Pears, Penelope Probert, An Optical Range Sensor for Mobile Robot Guidance, IEEE, 1993, p.659-664.
- [PRI 89] Pritschow G., Bewegungssteuerung und Programmierung mobiler Systeme, Selbstverlag, Karlsruhe, 1989, p.102--125.
- [PRI 91] Pritschow G., Advanced Sensor Guided Motion Control for Mobile Robots, Proceedings of the International Workshop on Information Processing in Autonomous Mobile Robots, Springer Verlag, Berlin, 1991, p.201-215.
- [PRI 92] Pritschow G., CIM,eine ganzheitliche steuerungstechnische Aufgabe, VDI Berichte, Nr. 881, 1991, p.1-22.
- [PRU 89] Alen Pruski, Robotique Generale, Editions Ellipses, Paris, 1989.

- [PSMO] Petr Svestka, Mark Overmars, Coordinated Motion Planning for Multiple Car - Like Robots using Probabilistic Roadmaps, IEEE, 1996, p.1631-1635.
- [RARI] Rachid Alami, Frederic Robert, Felix Ingrand, Multi - Robot Cooperating through Incremental Plan - Merging, IEEE, 1995, p.2573-2579.
- [REF 97] Richtlinien für fahrerlose Flurförderzeuge, Carl Heiman Verlag, Köln, 1972.
- [RFLK] Reda E.Fayek, Ramiro Liscano, Gerald M.Karam, A System Architecture for a Mobile robot Based on Activities and a Blackboard Control Unit, IEEE, 1993, p.267-274.
- [RGFG] Ruvinda Gunawardana, Fathi Ghorbel, The Class of Robot Manipulators with Bounded Jacobian of the Gravity Vector, IEEE, 1996, p. 3677- 3678.
- [RGMH] Roderic A.Grupen, Manfred Huber, 2D Contact Detection and Localization Using Proprioceptive Information, IEEE, 1993, p.130-135.
- [RHEK] Regis Hoffman, Eric Krotkov, Terrain Mapping for Outdoor Robot: Robot Perception for Working, IEEE, 1993, p. 529-533.
- [RMHN] Robin R.Murphy, Ken Hugh, Eva Noll, An Explicit Path Planner to Facilitate Reactive Control and Terrain Preferences, IEEE, 1996, p.2067-2072.
- [RRM] Robin R.Murphy, Robust Sensor Fusion for Teleoperations, IEEE, 1993, p.572-577.
- [RSAP] Raymond J.Spiteri, Uri M.Ascher, Dinesh K.Pai, Numerical Solution of Differential Systems with Algebraic Inequalities Arising in Robot Programming, IEEE, 1996, p.2373-2375.
- [RSIM] Reid Simmons, The Curvature - Velocity Method for Local Obstacle Avoidance, IEEE, 1996, p.3375-3382.
- [RTPP] R.M.Taylor, P.J.Probert, Range Finding and Feature Extraction by Segmentation of Images for Mobile Robot Navigation, IEEE, 1996, p. 95-100.
- [SBCD] Stephane Betge-Brezetz, Raja Chatila, Michel Devy, Object - Based Modelling and Localization in Natural Environments, IEEE, 1995, p.2920-2927.
- [SCH 94] Schmidt G., Integrated Mobile Robot Motion Planning and Execution in Changing Indoor Environments, Technische Universität München, 1994
- [SCH 96] Schraft, R.D. (Hrsg.), Westkämper, E., 2. Fraunhofer IPA-Technologie-Forum F 17, Innovative Technologien für Dienstleistungen, Entwicklung und Einsatz von Servicerobotern, 6. und 7. März 1996, Stuttgart. Stuttgart, FpF 1996.

- [SEW 94] Severin, Klein, Konzept für rechnergesteuerte Flurförderzeuge auf virtuellen Leitlinien, Deutsche Hebe- und Fordertechnik, dhf, 12/85, p.15-19.
- [SJDW] Simon Julier, Hugh Durrant-Whyte, Process Models for the High - Speed Navigation of Road Vehicles, IEEE, 1995, p.101-105.
- [SLJD] Sukhan Lee, Jignesh Desai, Implementation and Evaluation of Hex EYE: A Distributed Optical Proximity Sensor System, IEEE, 1996, p.2353-2360.
- [SMVa] Steven M.LaValle. Rajeev Sharma Framework for Motion Planning in Stochastic Environments: Modeling and Analysis, IEEE, 1995, p. 3057-3062.
- [SMVb] Steven M.LaValle. Rajeev Sharma, A Framework for Motion Planning in Stochastic Environments: Applications and Computational Issues, IEEE, 1995, p. 3063-3068.
- [SRG] Steven Ratering, Maria Gini, Robot Navigation in a Known Environment with Unknown Moving Obstacles, IEEE, 1993, p.25-29
- [SSAD] Subir Kumar Saha, Jorge Angeles, John Darcovich, The Kinematic Design of a 3 - dof Isotropic Mobile Robot, IEEE, 1993, p. 283-288.
- [SSJL] S. Sekhavat, J.P. Laumond, Topological Property of Trajectories Computed from Sinusoidal Inputs for Nonholonomic Chained Form Systems, IEEE, 1996, p. 3383-3388.
- [SSZS] S.Sundar, Z.Shiller, Time - Optimal Obstacle Avoidance, IEEE, 1995, p. 3075-3080.
- [STFC] S.K.Tso, Y.H.Fung, Y.P. Cheung, Fuzzy - logic Control for Differential - wheel - drive AGVs using Linear Opto - sensor Arrays, IEEE, 1996, p. 2816-2821.
- [STO 89] Storjohann K., Realisierung einer optischen Hinderniserkennung, Proceedings of the 3<sup>rd</sup> International Conference on Automatic Identification and Sensors, Sindelfingen, 1989, p.261-271.
- [SVZC] Ian Sillitoe, Antonio Visioli, Francesco Zanichelli, Stefano Caselli, Experiments in the Piece-wise Linear Approximation of Ultrasonic Echoes for Object Recognition in Manipulation Tasks
- [TAOS] Takeshi Aoki, Toshiaki Oka, Tatsuya Suzuki, Acquisition of Optimal Action Selection to Avoid Moving Obstacles in Autonomous Mobile Robot, IEEE, 1996, p.2055-2060.
- [TAK] Takashi T., Automated guide vehicle systems, Proceedings of the 5-th International Conference 6-8 October 1987, Tokyo, Japan
- [TBRA] Tucker Balch, Ronald Arkin, Avoiding the Past: A Simple but Effective Strategy for Reactive Navigation, IEEE, 1993, p. 678-685.

- [TCAH] Thomas R.Collins, Ronald C.Arkin, Andrew M.Henshaw, Integration of Reactive Navigation with a Flexible Parallel Hardware Architecture, IEEE, 1993, p. 271- 276.
- [TFIH] Toshio Fukuda, Hidenori Ishihara, Naohira Hiraoka, Deadlock Resolution in Distributed Autonomous Robotic System with Hand - to - Hand Motion, IEEE, 1996, p.2049-2054.
- [TFYY] Toshio Fukuda, Ysunari Yokohama, Navigation System Based on Ceilling Landmark Recognition for Autonomous Mobile Robot, IEEE, 1996, p.1720-1725.
- [THBD] Thomas C.Henderson, Beat Bruderlin, Mohamed Dekhil, Sonar Sensing Strategies
- [TNAO] Toshihiro NISHIZAWA, Akihisa OHYA, An Implementation of On - board Position Estimation for a Mobile Robot, IEEE, 1995, p.395-400.
- [TOT] J.Y.Tigli, M.Ocello, M.C. Thomas Toward, a new Intelligent Reactive Controller for autonomous Mobile Robots, IEEE, 1993, p.249-254.
- [TSIM] Tomohiro Shibata, Yoshio Matsumoto, Hyper Scooter:a Mobile Robot Sharing Visual Information with a Human, IEEE, 1995, p.1074-1079.
- [TSWT] Karen T.Sutherland, William B.Thompson, Inexact Navigation
- [UHGS] Uwe D.Hanebeck, Günther Schmidt, Set Theoretic Localization of Fast Mobile Robots using an Angle Measurement Technique, IEEE, 1996, p.1387-1394.
- [VDI2199] 06.86 Empfehlungen für bauliche Planungen beim Einsatz von Flurförderzeuge
- [VDI2510] Fahrerlose Transportfahrzeuge (FTF)
- [VDI2513] 08.95 FTS-Checkliste - Eine Planungshilfe für Betreiber und Hersteller von Fahrerlosen Transportsystemen (FTS)
- [VDI3586] E 03.94 Flurförderzeuge, Begriffe, Kurzzeichen
- [VDI3641] 05.88 Mobile Datenübertragungssysteme im innerbetrieblichen Transport
- [VDI3645] 03.88 Empfehlungen für Boden, Regale und Leitlinienführungen beim Einsatz von Flurförderzeugen im Lager
- [VDI4451] Blatt 1 08.95 Kompatibilität von Fahrerlosen Transportsystemen (FTS) - Handsteuergerät
- [VDI4451] Blatt 3 03.95 Kompatibilität von Fahrerlosen Transportsystemen (FTS) - Fahr- und Lenkantrieb
- [VDI4451] Blatt 4 01.95 Kompatibilität von Fahrerlosen Transportsystemen (FTS) - Offene Steuerungsstruktur für Fahrerlose Transportfahrzeuge (FTF)



- [VDI4451] Blatt 5 12.94 Kompatibilität von Fahrerlosen Transportsystemen (FTS) - Schittstelle zwischen Auftraggeber und FTS-Steuerung
- [VGAS] Sjiur J. Vestli, Nadine Tschichold- Gürman, Martin Adams, Sandra Sulzberger, Integration of Path Planning, sensing and Control in Mobile Robotics, IEEE, 1993, p.243-247.
- [VGSR] V.Gabbani, S.Rocchi, Multielement Ultrasonic System for Robotic Navigation, IEEE, 1995, p. 3009-3014.
- [VOG 92] Vogler H., Steuerungsverfahren zur leitdrahtlosen Führung Fahrerlosen Transportfahrzeuge, Selbstverlag FLW,, Dortmund, 1992, p.7.119-7.126.
- [WBN] Thomas S. Wikman, Michael S. Branicky, Wyatt S. Newman, Reflexive Collision Avoidance: A Generalized Approach.
- [WIK 88] Wiklund M., AGV Navigation by Angle measurements. Springerler Verlag berlin, 1988, p.199-212.
- [WHUO] Wei Huo, New Formulas for Complete Determining Base Parameters of Robots, IEEE, 1995, p. 3021-3023.
- [XLZD] Xueyin Lin, Zhigang Zhu, Wen Deng, A Stereo Matching Algorithm Based on Shape Similarity for Indoor Environment Model Building, IEEE, 1996, p. 765- 770.
- [XUCH] Hong XU, Xiaochen CHI, Calibration and Parameter Identification for Laser Scanning Sensor, IEEE, 1993, p.665-670.
- [XYNS] Xiiaping Yun, Nilanjan Sarkar, Dynamic Feedback Control of Vehicles with Two Steerable Wheels, IEEE, 1996, p.3105-3110.
- [YHC] Yen- Hui Chuang, Potential - Based Modeling of Three Dimensional Workspace for Obstacle Avoidance
- [YAS 85] Yasunobu S., Automtic Train Operation System by Predictive Fuzzy Control, North Holland, 1985
- [YOS] Yosimura T., An integrated phisical distribution system for steel coils, AUTOMATION IN WARE HOUSING 8-th international conference, Tokyo, Japan, 1987
- [YSHM] Yuichi Sato, Mitsunori Hirata, Tsugito Maruyama, Efficient Collision Detection using Fast Distance - Calculation Algorithms for Convex and Non - Convex Objects, IEEE, 1996, p. 771-778.
- [YX] Yoshio Yamamoto and Xiaoping Yun, Control of mobile Manipulators Following a Moving surface.
- [ZEE 91] Zeeh M., Ein Steuerungskonzept auf der Basis speicherprogrammierbarer Steuerungen, Selbstverlag duisburg, 1991, p. 107-135.
- [ZLDZ] Ze-Nian Li, Danpo Zhang, Real - time Line - based Motion Stereo, IEEE,, 1993, p. 367-372.



---

[ZZWH] Zhongfei Zhang, Richard Weiss, Allen R.Hanson, Automatic calibration and Visual Servoing for a Robot Navigation System, IEEE, 1993, p. 14-19.