UNIVERSITATEA "POLITEHNICA" TIMIŞOARA

FACULTATEA DE MECANICĂ

ING. MIRELA TOTH-TAŞCĂU

METODE ŞI TEHNICI DE GENERARE A TRAIECTORIILOR LA ROBOŢI INDUSTRIALI

TEZĂ DE DOCTORAT

BIBLIOTECA CENTRALĂ
UNIVERSITATEA "POLITEHNICA"
TIMIŞOARA

CONDUCĂTOR ŞTIINŢIFIC
PROF.DR.ING. DOINA DRĂGULESCU

1996

BUPT

INTRODUCERE

Cercetarea ştiinţifică întreprinsă în cadrul lucrării de doctorat a fost orientată spre mecanica roboţilor, cinematica şi dinamica acestora constituind un domeniu al mecanicii tehnice încă insuficient studiat. Studiul generării mişcării roboţilor constituie şi subiectul unui program de cercetare aprobat de CNCSU în 1995, cu tema "*Planificarea şi generarea mişcării la roboţi*", care continuă şi în prezent.

Rezolvarea problemelor de cinematică şi dinamică a roboţilor este oglindită de calitatea îndeplinirii sarcinii robotului, de precizia generării şi parcurgerii traiectoriei dorite. Domeniul problemelor abordate cuprinde atât metodele şi tehnicile efective de generare a mişcării, cât şi problemele conexe, precum modelarea roboţilor (modelele geometric, diferenţial, dinamic), precizia cinematică şi dinamică a parcurgerii traiectoriei, cunoaşterea torsorului din cuple, respectiv a torsorilor care acţionează asupra elementelor robotului, vibraţiile din cuple şi deformaţiile elementelor, modelarea spaţiului de lucru, planificarea mişcării robotului în acest spaţiu, etc.

Studiul bibliografic aprofundat întreprins pentru cunoaşterea stadiului actual al cercetărilor în acest domeniu, dar şi a tendinţelor actuale şi de perspectivă, cuprinde lucrări consacrate în domeniu, numeroase reviste de specialitate şi volumele unor manifestări ştiinţifice de prestigiu, naţionale şi internaţionale. Dintre revistele de specialitate şi volumele manifestărilor ştiinţifice care au constituit o importantă sursă bibliografică pot fi amintite: International Journal of Robotics Research, IEEE Transactions on Robotics and Automation, IEEE Transactions on Systems, Man, and Cybernetics, IEEE Transactions on Automatic Control, Robotica, Journal of Robotic Systems, International Journal of Mechanism and Machine Theory, ASME Journal of Applied Mechanics, International Journal of Engineering Science, Proceedings of the International Symposium on Industrial Robots, Proceedings of IEEE International Conference on Robotics and Automation, etc.

Lucrarea prezintă metode consacrate de studiu, propune noi metode sau extensii ale celor existente şi conduce totodată la formarea unor noi direcţii de cercetare în acest domeniu.

În concepţia sa unitară, lucrarea este formată din două părţi şi structurată în opt capitole. ***Partea I*** cuprinde: Capitolul 1. *Problematica generării traiectoriilor*, Capitolul 2. *Metode de generare a traiectoriilor în spaţiul operaţional*, Capitolul 3. *Metode de generare a traiectoriilor în spaţiul cuplelor cinematice*, Capitolul 4. *Interpolarea traiectoriilor folosind funcţii spline*, Capitolul 5. *Generarea traiectoriei unui robot pe baza modelului dinamic*, Capitolul 6. *Studiu experimental privind conducerea unui robot cu comandă punct cu punct, pe o traiectorie continuă, în spaţiul operaţional.* ***Partea a II-a*** cuprinde: Capitolul 7. *Planificarea şi generarea mişcării într-un spaţiu cu obstacole*, Capitolul 8. *Planificarea mişcării roboţilor prin metodele câmpului potenţial.*

BUPT

Capitolul 1. Problematica generării traiectoriilor

Pornind de la definirea traiectoriei unui robot, sunt prezentate diferitele tipuri posibile de traiectorii şi posibilităţile de generare a acestora. De asemenea, este abordată relaţia dintre generarea mişcării şi sistemul de comandă al robotului.

Generarea mişcării unui robot poate fi realizată în spaţiul operaţional sau în spaţiul cuplelor cinematice, fiecare din cele două metode având anumite avantaje, dar şi dezavantaje, care sunt prezentate în acest capitol. Modul în care se realizează generarea mişcării în fiecare din cele două spaţii este prezentat schematizat, deosebit de sugestiv, punând în evidenţă mulţimile parametrilor necesari pentru descrierea mişcării, precum şi algoritmii de calcul.

O altă problemă tratată în acest capitol, deosebit de importantă pentru generarea mişcării, este controlul (optimal). Este subliniat rolul restricţiilor obstacolelor şi ale traiectoriei în determinarea modului de control, dar mai ales rolul modelului dinamic al robotului.

Funcţia generatorului de traiectorie şi relaţia acestuia cu planificatorul de sarcină, planificatorul de traiectorie, optimizatorul, sistemul de acţionare şi sistemul senzorial sunt prezentate într-o manieră sintetică, originală şi deosebit de sugestivă.

Capitolul 2. Generarea traiectoriilor în spaţiul operaţional

În acest capitol sunt tratate problemele generale ale generării traiectoriilor unui robot în spaţiul operaţional, dar şi legătura directă cu coordonatele cuplelor cinematice. Pentru evitatea coliziunii cu obstacolele din spaţiul de lucru, este necesar să se introducă un număr de puncte intermediare între configuraţia iniţială şi cea finală. Astfel, pentru generarea unei traiectorii ce trece printr-un număr dat de puncte în spaţiul operaţional, este propusă interpolarea polinomială, fiind subliniate totodată avantajele şi dezavantajele acestei interpolări.

Pentru generarea mişcărilor interpolate, în spaţiul operaţional, se propune o dublă parametrizare a traiectoriei: un parametru pentru legea orară a poziţiei şi un parametru pentru evoluţia orientării efectorului final.

Ca alternativă pentru generarea mişcării, atunci când nu se dispune de un model geometric invers, se propune inversarea numerică. Metoda propusă se bazează pe principiul metodei Newton-Raphson. Metoda este recomandată în cazul roboţilor cu şase grade de libertate. Pentru acest caz se propune trecerea de la matricea de transformare generală la vectorul coordonatelor carteziene.

Pentru optimizarea mişcării se propune o metodă bazată pe reprezentarea în planul fazelor, precum şi obţinerea unor traiectorii de viteză maximă pe baza modelului dinamic.

O altă metodă propusă în acest capitol constă în generarea mişcării în spaţiul cartezian cu matricea conducere. Metoda propusă de R.Paul pentru deplasarea liniară între două situări succesive este corelată cu structurarea mişcării robotului în spaţiul de lucru, fiind utilizate ecuaţiile de transformare care descriu sarcina robotului.

Pentru mişcarea punct cu punct în spaţiul operaţional sunt studiate două cazuri: mişcarea în timp minim, cu abatere controlată faţă de punctele impuse, cu păstrarea constantă a orientării efectorului final şi mişcarea în timp minim, cu abatere nulă faţă de punctele impuse în coordonate carteziene, cu luarea în considerare a orientării efectorului final. Sunt prezentate două metode de rezolvare a problemei optimizării în cazul menţinerii constante a orientării efectorului final. De asemenea, a fost elaborat un program de calcul care rezolvă problema prin cele două metode amintite.

BUPT

Pornind de la metoda propusă de Taylor pentru interpolarea cvasiliniară în spațiul configurațiilor (spațiul cuplelor cinematice), s-a formulat un algoritm clar şi simplu de aplicat, bazat pe metoda înjumătățirii intervalului drept criteriu de convergență.

Capitolul 3. Metode de generare a traiectoriilor în spațiul cuplelor cinematice

Abordarea metodelor de generare a traiectoriilor în spațiul cuplelor cinematice s-a realizat în mod sistematizat, ținând cont de anumite aspecte legate de robot şi de mişcare. Selectarea metodei de generarea a mişcării în coordonatele cuplelor se face în funcție de:

- tipul de generare a mişcării: între puncte de oprire sau între puncte de trecere;
- sincronizarea cuplelor: mişcări independente sau mişcări coordonate;
- forma traiectoriei: reală sau normalizată;
- timpul de parcurgere: impus sau calculat.

Pentru generarea mişcării între două puncte de oprire s-au studiat interpolarea polinomială (gradul trei şi gradul cinci), profilul triunghiular de viteză, profilul trapezoidal de viteză şi s-au propus, pentru evitatea discontinuităților acceleraților, profilurile, triunghiular şi trapezoidal, netezite. S-au studiat de asemenea, traiectoria critic amortizată, traiectoria cosinusoidală şi traiectoria sinusoidală suprapusă peste rampă.

Pentru generarea mişcării între două puncte de trecere s-au studiat funcțiile de interpolare polinomială, sinusoidală şi sinusoidală ponderată în timp.

Un alt aspect deosebit de important în generarea mişcării îl constituie coordonarea cuplelor cinematice. Indiferent de alegerea metodei pentru coordonarea cuplelor, rolul important îl are calculul timpului minim de parcurgere a traiectoriei. Odata stabilită cupla cinematică ce impune restricții mişcării se poate alege una din cele două metode de coodonare prezentate: metoda timpului minim şi metoda proporțională.

Pentru generarea unei traiectorii care să treacă riguros prin punctele intermediare impuse, cu restricții de viteză şi acceleraţie, în timp minim de parcurgere, se prezintă şi dezvoltă metoda propusă de Khalil, de interpolare polinomială cu restricții în spațiul configurațiilor.

Metodele prezentate în acest capitol au stat la baza elaborării unui program amplu de generare a mişcării în spațiul cuplelor cinematice. Acest program oferă utilizatorului alegerea oricărei variante de mişcare, putând fi implementat cu uşurință oricărui robot.

Capitolul 4. Interpolarea traiectoriilor folosind funcții spline

În ultimii ani s-au impus pe plan teoretic şi mai ales practic metodele de interpolare bazate pe funcțiile spline. Datorită multiplelor avantaje şi a implementării acestor funcții în diferite programe de tip CAD, de calcul numeric şi grafică, precum MATHCAD şi MATLAB sau de proiectare asistată de calculator, precum AUTOCAD, a fost elaborat un capitol separat dedicat acestui tip de interpolări.

Un mod de a folosi funcțiile spline constă în impunerea unor puncte de control. S-au studiat astfel traiectoria 4-3-4, 3-5-3 şi 3-3-3-3-3. Pentru cazul când traiectoria robotului trece printr-un număr dat de puncte de precizie s-au folosit funcțiile spline de tipul 4-3-...-3-4. Pe baza algoritmului metodei prezentate s-a elaborat un program de calcul cu care s-a simulat generarea mişcării unui robot de construcție modulară, cu şase grade de libertate. Programul poate fi folosit separat, pentru generarea mişcării oricărui robot, dar şi incorporat şi în programul general de generare a mişcării în coordonatele cuplelor, prezentat în capitolul precedent.

BUPT

Un prim pas spre folosirea modelului dinamic pentru generarea mişcării îl constituie metoda de interpolare cu funcţii spline cubice, cu restricţii cinematice şi dinamice. Tot în acest capitol a fost tratat şi cazul traiectoriilor cvartice, cu erori pătratice minime ale cuplelor.

Pentru a evita cunoaşterea "a priori" a tuturor punctelor de definire a traiectoriei, se propune o metodă de interpolare a unei traiectorii prin "glisarea" unei cubice care trece prin patru puncte de precizie.

Capitolul 5. Generarea traiectoriei unui robot pe baza modelului dinamic

Controlul optimal al mişcării unui robot presupune cunoaşterea modelului dinamic al acestuia, împreună cu o descriere adecvată a traiectoriei şi a restricţiilor impuse.

Determinarea modelului dinamic s-a realizat pe baza formalismului Lagrange, în varianta cunoscută în mecanica analitică şi folosind transformările omogene. Ca exemplu de aplicare a ambelor metode, s-a elaborat şi rezolvat modelul dinamic direct (**MDD**) şi modelul dinamic invers (**MDI**) pentru un manipulator cu trei grade de libertate. Un alt exemplu de rezolvare a modelului dinamic, în condiţii impuse de mişcare în cuplele cinematice, este prezentat pentru robotul KUKA tip 364/10.0.

O altă problemă deosebit de importantă este determinarea torsorului care acţionează în cuple, respectiv asupra elementelor robotului, când este cunoscut torsorul care acţionează asupra efectorului final. Cunoaşterea torsorului din cuple permite verificarea restricţiilor de torsor, dar, de asemenea, poate folosi pentru optimizarea generării mişcării. Se propune o metodă pentru calculul acestor torsori, pe baza căreia s-a elaborat un program. Exemplificarea este prezentată pentru cazul robotului RTRTTR, dar programul poate fi folosit pentru orice robot şi orice solicitare a efectorului final.

Calitatea rezolvării generării mişcării este oglindită de precizia de parcurgere a traiectoriei. Se defineşte astfel o precizie cinematică, respectiv una dinamică şi se propun două metode de calcul: cea pentru precizia cinematică este bazată pe modelul diferenţial, iar cea pentru precizia dinamică este bazată pe modelul dinamic. Ambele metode sunt exemplificate pentru acelaşi robot TTR.

Se propune de asemenea o metodă originală de calcul al erorilor datorate deformaţiilor elementelor robotului. Metoda propusă adaptează şi aplică principiile generale de calcul dinamic al structurilor la metodele uzuale de modelare şi calcul dinamic al roboţilor.

O altă metodă propusă în acest capitol permite realizarea unei traiectorii dorite, cu o anumită viteză, la un robot cu o structură cunoscută, ţinând cont de vibraţiile care apar în cuplele cinematice.

Capitolul 6. Studiu experimental privind conducerea unui robot cu comandă punct cu punct, pe o traiectorie continuă, în spaţiul operaţional

Majoritatea roboţilor existenţi aparţin generaţiilor mai vechi, aceştia nefiind dotaţi cu generator de traiectorie în sistemul de conducere. Generarea unei traiectorii continue, în spaţiul operaţional, cu un sistem cu comandă punct cu punct, necesită realizarea off-line a unui generator de traiectorie.

Pentru studiul experimental a fost folosit un robot didactic, cu care s-au generat mai multe tipuri de traiectorii interpolate în 3D. Pentru evaluarea preciziei de generare a mişcării a fost folosit modelul geometric direct. Rezultatele studiului experimental sunt prezentate concis sub formă de tabele şi grafice.

BUPT

Capitolul 7. Planificarea şi generarea mişcării într-un spaţiu cu obstacole

În acest capitol sunt tratate problemele legate de modelul mediului, metode de planificare şi generare a mişcării într-un spaţiu cu obstacole.

Ca metodă de modelare a mediului se propune metoda grilei neomogene, metodă pentru care a fost elaborat un program de calcul.

Dintre metodele de planificare prezentate se pot aminti: metoda grafului de vizibilitate, metoda drumului liber, metoda descompunerii celulare exacte.

Pentru fiecare din metodele amintite s-a conceput un algoritm pe baza căruia a fost elaborat un program.

Capitolul 8. Planificarea mişcării roboţilor prin metodele câmpului potenţial

Metodele câmpului potenţial se bucură de cea mai largă audienţă în rândurile celor care studiază problemele planificării mişcării roboţilor. Aceste metode sunt deosebit de utile, şi atractive în acelaşi timp, deşi prezintă un dezavantaj destul de important: blocarea într-un minim de potenţial, altul decât cel corespunzător configuraţiei finale.

Se prezintă principiul acestor metode pentru cazul când robotul execută doar translaţie şi pentru cazul general. Sunt prezentate trei tehnici de abordare a problemei: depth-first palnning, best-first planning şi optimizarea funcţiei construite prin integrarea potenţialului de-a lungul traiectoriei.

S-au conceput şi elaborat două programe bazate pe aceste metode. Unul din programe este exemplificat pentru generarea mişcării unui robot de tip maşină, iar al doilea pentru generarea unei traiectorii plane oarecare.

Concluziile care rezultă din problemele abordate sunt prezentate la sfârşitul fiecărui capitol.

În încheiere, teza conţine câteva consideraţii finale, prezintă într-o manieră sintetică contribuţiile originale în domeniul studiat şi defineşte totodată preocupările viitoare, respectiv noile direcţii de cercetare în acest domeniu.

BUPT

1. PROBLEMATICA GENERĂRII TRAIECTORIILOR

1.1. INTRODUCERE

În accepţiunea actuală, *robotul* poate fi definit ca un sistem cu funcţionare automată, adaptabilă prin reprogramare condiţiilor mediului complex şi variabil în care acţionează, amplificând sau înlocuind una sau mai multe dintre funcţiile umane în acţiunea sa asupra mediului. Manipulatoarele şi roboţii au particularităţi deosebite din punct de vedere mecanic şi al sistemului de comandă. Astfel, un robot dispune de o ierarhie de niveluri de comandă. Funcţie de numărul nivelurilor ierarhice, de complexitatea sistemelor de comandă, roboţii aparţin la *generaţii de roboţi* [A4], [C16], [D6], [D7], [D15], [H3], [I3], [K20], [K21], [M6], [N6], [T6].

A treia generaţie de roboţi ("roboţii inteligenţi"), la care conţinutul tezei face referire, posedă elemente de inteligenţă artificială. Aceştia pot să-şi definescă singuri sarcinile pentru rezolvarea unor probleme particulare, considerând informaţia despre mediul înconjurător (organizată în modelul mediului) şi pot să-şi modifice acţiunile funcţie de informaţiile furnizate de sistemul senzorial. Roboţii inteligenţi pot fi complet autonomi, gradul lor de inteligenţă fiind funcţie de scopul pentru care sunt construiţi. Robotul din generaţia a treia se poate defini ca un sistem capabil să execute sarcini care necesită şi anumite calităţi umane: adaptarea, învăţarea, capacitatea de reprezentare a mediului înconjurător, predicţia şi planificarea, etc.

Robotul generaţiei a treia se constituie ca o ierarhie de structuri corespunzătoare generaţiilor precedente, ierarhie care se regăseşte şi în structura de comandă şi de programare a acestuia. Programarea se realizează într-un limbaj natural sau într-un limbaj la nivel de obiectiv, când sarcina robotului se descrie ca un obiectiv de îndeplinit. Robotul îşi generează şi actualizează permanent un model al mediului înconjurător cu ajutorul sistemului senzorial, al vederii artificiale, al înţelegerii limbajului natural sau prin descrieri într-un limbaj specializat, de nivel înalt. Pe baza sarcinii primite şi a informaţiei din modelul mediului, se generează planul global de acţiune al robotului (etapa de planificare a mişcării). Acest plan este împărţit în acţiuni simple de manipulare sau deplasare în mediul înconjurător (structurarea acţiunilor robotului). Execuţia acţiunilor elementare se poate modifica funcţie de informaţia sistemului senzorial, asigurând adaptarea la mediu.

Pentru roboţii mobili, planificarea mişcării necesită existenţa unui model al terenului, inclus în modelul mediului. Planificarea şi generarea mişcării roboţilor mobili implică multe metode specifice inteligenţei artificiale. Pentru operaţii simple de manipulare, sistemul de comandă generează traiectoriile de mişcare între două *puncte* ale spaţiului de lucru şi controlează mişcarea efectorului robotului pe această traiectorie. Deoarece robotul operează într-un spaţiu real, populat cu obiecte fizice, trebuie să fie capabil să-şi planifice şi genereze propriile mişcări, să-şi activeze cuplele cinematice în

BUPT

acord cu obiectivul sarcinii, în funcție de aranjamentul inițial (static) şi final al obiectelor din spațiul de lucru (dinamic, când informațiile de la sistemul senzorial sunt obținute în timpul evoluției sale).

Sistemul de comandă al robotului trebuie să asigure mişcarea structurii mecanice în două moduri: liber (traiectorii libere) sau în contact cu alte obiecte (traiectorii semilibere). În primul caz, se evită, pe baza modelului mediului, coliziunea cu obiectele din spațiul de lucru şi se controlează poziția şi viteza efectorului pe tot parcursul evoluției sale [C14], [C17], [D7], [D14], [I5], [K19], [L2], [P12], [R9], [S15], [T7]. În al doilea caz, trebuie studiată suplimentar interacțiunea cu mediul şi se controlează inclusiv forța de interacțiune [C15], [E3], [F4], [G2], [H6], [J1], [L21].

Sistemul de decizie-comunicare îi permite robotului să-şi execute sarcinile în mod autonom, printr-o succesiune de acțiuni elementare, adaptate la starea mediului de lucru şi care, parțial sau total, nu au fost explicitate în instrucțiunile date inițial de operatorul uman. Comunicarea eficientă om-robot este una din problemele fundamentale ale automatizării flexibile programabile [F2], [L12], [P2], [P5], [S3], [T17], [V4], [V5]. Astfel, au apărut şi s-au dezvoltat limbaje specializate de programare a roboților, structurate pe patru niveluri ierarhice:

- Nivelul 1: programarea este total dependentă de echipament; referințele date în coordonate robot se transformă în cupluri aplicate servomotoarelor sistemului de acționare;
- Nivelul 2: programarea prin instruire este frecvent utilizată la roboții industriali existenți; punctele din spațiul de lucru sunt memorate, robotul fiind comandat manual de la o consolă specializată. Există puține facilități de control al programului robot (salturi condiționate, subrutine, etc.) şi al semnalelor senzoriale;
- Nivelul 3: are posibilități mari de control al programului robot şi de prelucrare a informației; prezintă primele facilități de executare paralelă a sarcinilor şi descriere a poziției şi orientării corpurilor cu ajutorul coordonatelor omogene;
- Nivelul 4: extinde reprezentarea corpurilor cu ajutorul coordonatelor omogene, transformările de coordonate, folosirea structurilor complexe de date, utilizarea informației senzoriale.

Stabilirea tehnicilor şi metodelor de generare a traiectoriilor este în strânsă legătură cu tipul de comandă şi de programare a roboților.

1.2. TIPURI DE TRAIECTORII ŞI POSIBILITĂȚI DE GENERARE

Dacă robotul trebuie să execute o sarcină ce include o deplasare, el trebuie să urmărească, după o lege de mişcare (orară) dată, o traiectorie definită printr-o secvență de repere, numite impropriu "*puncte*", ce corespund situărilor succesive ale efectorului final. Aceste *puncte* pot fi:

- programate prin învățare;
- furnizate on-line de către sistemul senzorial (un senzor exteroceptiv);
- rezultate (preluate) dintr-o bază de date a unui sistem CAD/CAM.

Problema generării mişcării este de a calcula succesiunea de comenzi care asigură trecerea robotului prin aceste *puncte* (sau prin vecinătatea acestora). Pentru aceasta, se

BUPT

găsesc una sau mai multe funcții matematice de interpolare care să asigure trecerea robotului prin punctele traiectoriei în funcție de timp.

Generarea traiectoriei unui robot este o sarcină funcțională care determină mişcarea acestuia sub controlul senzorial. Parametrii traiectoriei sunt definiți prin program. Există o mare diversitate de metode de calcul al unei traiectorii, atât pentru roboții seriali [B8], [C1], [C13], [C14], [C15], [D14], [D16], [D50], [D51], [K4], [K19], [L15], [M5], [N2], [N3], [P6], [P12], [S20], [V12], cât şi pentru cei paraleli [M9], [M10].

Se numeşte ***traiectorie*** a unui robot, locul geometric al punctului caracteristic al efectorului final al robotului în spațiul operațional. Sarcina robotului impune efectorului final al acestuia o anumită situare (poziție şi orientare), astfel încât, în anumite situații, trebuie luată în considerare şi orientarea efectorului final în timpul evoluției punctului caracteristic pe traiectorie.

Realizarea acțiunilor de către un robot impune numeroase deplasări ale efectorului său. Ansamblul situațiilor corespunzătoare acestor deplasări defineşte ***traiectoriile geometrice*** în ***spațiul operațional***, şi, prin intermediul transformărilor de coordonate (**MGI**), ***traiectoriile cuplelor***, în ***spațiul cuplelor cinematice***. O traiectorie este deci, o entitate pur geometrică definită prin ***sarcina*** şi prin ***restricțiile*** geometrice ale universului său.

Indiferent de modul de conducere a robotului, punctul caracteristic descrie întotdeauna o traiectorie. În unele cazuri, această traiectorie este impusă de procesul tehnologic, iar conducerea robotului se realizează pe traiectorie carteziană continuă. În alte cazuri, sunt impuse numai *punctele țintă* (*via points*), în coordonate carteziene sau în coordonatele cuplelor cinematice conducătoare. Traiectoria rezultată este în general greu previzibilă, conducerea realizându-se simultan pe fiecare cuplă cinematică conducătoare.

Specificarea traiectoriei robotului se poate face în formă parametrică. Acest mod de descriere a traiectoriilor permite: controlul vitezelor şi al accelerațiilor, realizarea unor mişcări line şi a unor mişcări uşor programabile, posibilitatea modificării situării robotului sau efectorului final fără redefinirea programului robot, manipularea obiectelor în mişcare, evitarea coliziunii cu obstacolele din spațiul de lucru. Deoarece traiectoria este memorată ca o funcție parametrizată, capacitatea de memorie utilizată este mai mică decât în cazul memorării traiectoriei într-o fază de învățare.

În majoritatea cazurilor se generează o traiectorie între două *puncte* ale spațiului de lucru (*puncte de oprire*), în care robotul se află în repaus. Pentru evitarea obstacolelor, între aceste două *puncte* principale sunt necesare mai multe *puncte* intermediare prin care trebuie să treacă traiectoria robotului.

În conducerea unui robot interesează nu numai trecerea prin anumite situări impuse sau calculate, ci şi evoluția lină în timp [B15], [B16]. Astfel, se alege pentru coordonata generalizată q^j o anumită funcție $q^j = f^j(t)$, care să impună evoluția lină a acesteia în timp. Legătura între spațiu şi timp este realizată deci prin intermediul acestei funcții: la anumite valori ale timpului, valorile numerice ale funcției $f^j(t)$ trebuie să fie egale cu valorile coordonatei q^j corespunzătoare trecerii prin *punctele* țintă de pe traiectorie. Deci, indiferent de tipul mişcării robotului, trebuie evitate schimbările de sens în mişcarea motoarelor, realizându-se o mişcare fără şocuri sau depăşiri şi întoarceri la *punctele* țintă finale (funcția $f^j(t)$ trebuie să fie monotonă pe intervalul $[t_0, t_f]$). Dispar în acest mod oscilațiile în spațiul de lucru, pierderile de timp sau atingerea obstacolelor.

BUPT

În funcție de sarcina robotului se pot distinge mai multe tipuri de mişcări: dintr-un *punct* în alt *punct* (*traiectorie punct cu punct*), pe segmente de dreaptă, pe curbe definite parametric (*traiectorie continuă*), cu viteze definite pentru manipularea obiectelor în mişcare, pentru evitarea coliziunilor, sub control senzorial. Mişcarea efectorului final impune diferiți algoritmi de interpolare. Cele mai uzuale metode de inperpolare sunt:

- interpolarea liniară pentru segmente de dreaptă;
- interpolarea liniară cu tranziție între segmentele de dreaptă;
- interpolarea traiectoriei prin funcții polinomiale;
- interpolarea între puncte care variază în timp, folosind funcții de timp sau strategii gradient.

În cazul *traiectoriei continue*, executarea unei sarcini specifice se realizează cu toleranțe minime asupra poziției, vitezelor şi caracteristicilor mişcării de-a lungul traiectoriei. Descrierea mişcării rezultă din utilizarea modelelor matematice sau a mecanismelor "master" care introduc informații continue sau eşantionate la înaltă frecvență, de ordinul sutelor de cicluri pe secundă. Pentru aplicarea la programarea unui robot, problema cu care se confruntă această metodă constă în faptul că este necesar să se cunoască toate *punctele* de definire a traiectoriei pentru efectuarea calculelor, ceea ce necesită o memorie mare şi impune o învățare în două faze (achiziția de date şi comprimarea lor). Informațiile trebuie comprimate pentru a fi înregistrate într-o memorie de dimensiuni rezonabile. Rezultatul acestei codificări poate fi utilizat în timp real de către calculator pentru a restitui cât se poate de fidel mişcarea dorită. O modalitate frecvent utilizată este cea a transformatei Fourier pentru modelarea mişcării şi comprimarea datelor, care conduce la traiectorii şi mişcări bine atenuate şi apropiate de cele care corespund fazei de învățare.

Sarcinile de tipul *punct cu punct* impun mai puține restricții. În acest caz, sarcina robotului este specificată printr-o succesiune de situații (configurații), în care, oprit sau deplasându-se cu o anumită viteză, efectorul realizează o anumită operație. Între aceste *puncte*, utilizatorul alege, în urma rezolvării problemelor de planificare a mişcării, un set de *puncte* intermediare, pentru evitarea coliziunii cu obiectele din spațiul de lucru.

_ Pentru descrierea traiectoriei în spațiul operațional (cartezian) sunt necesare patru mulțimi de parametri:

- $\mathbf{X}_k$ (k=0, 1, ..., m) sau $\mathbf{G}_k$ *punctele* din spațiul de lucru prin care trece traiectoria;
- $\mathbf{v}_k$, ω_k vitezele liniare, respectiv unghiulare în *punctele* $\mathbf{X}_k$;
- $\mathbf{a}_k$, ε_k accelerațiile liniare, respectiv unghiulare în *punctele* $\mathbf{X}_k$;
- $\mathbf{t}_k$ timpii când aceste *puncte* sunt atinse de către robot.

Pentru mişcarea efectorului final de-a lungul traiectoriei carteziene calculate trebuie să se determine evoluția în timp a coordonatei generalizate a fiecărei cuple cinematice. Transformarea *spațiu operațional-spațiu robot* (spațiul cuplelor cinematice conducătoare) se realizează pe baza *modelului geometric invers* (**MGI**). În acelaşi timp, are loc şi transformarea *spațiu robot-spațiu operațional*, pe baza *modelului geometric direct* (**MGD**), pentru a cunoaşte în fiecare moment situarea robotului în spațiul operațional. Transformările de coordonate ocupă un rol important în generarea mişcării roboților, indiferent de spațiul în care se realizează această generare. Există în literatura de specialitate numeroase lucrări care abordează această problemă: [A6], [B1], [D5], [G3], [G6], [K1], [L24], [M1], [M12], [S28], [V12], [W4], [W9], inclusiv singularitățile ce pot să apară [R14], [S9], incertitudinea modelului [B22], sau diferitele restricții [H12].

BUPT

Similar, pentru descrierea traiectoriei în spațiul cuplelor, sunt necesare patru mulțimi de parametri:

- $\mathbf{q}_k$ (k=0, 1, ..., m) valorile variabilelor cuplelor corespunzătoare *punctelor* din spațiul de lucru prin care trece traiectoria;
- $\dot{\mathbf{q}}_k$ vitezele cuplelor, corespunzătoare *punctelor* $\mathbf{X}_k$;
- $\ddot{\mathbf{q}}_k$ accelerațiile cuplelor, corespunzătoare *punctelor* $\mathbf{X}_k$;
- t_k timpii când aceste *puncte* sunt atinse de către robot.

La toate tipurile de traiectorii trebuie să se verifice încadrarea în limitele maxime (de viteză şi accelerație) ale sistemului de acționare. Accelerația maximă pentru un grad de libertate este funcție de cuplul motor maxim, de momentul de inerție maxim, de forța de frecare maximă şi de factorul de siguranță ales. Controlul vitezei în mişcarea robotului este utilizat cu precădere când deplasările sunt scurte şi se realizează cu viteze mici.

Funcție de sincronizarea pe traiectorie a cuplelor cinematice conducătoare (motoare) se deosebesc două categorii de mişcări ale robotului: *mişcare coordonată*, când toate motoarele îşi termină mişcarea în acelaşi timp, şi *mişcare necoordonată* (independentă), când fiecare motor ajunge în timp minim în *punctul* final.

1.3. GENERAREA MIŞCĂRII ŞI SISTEMUL DE COMANDĂ

Pentru comanda mişcării *punct cu punct,* sarcina robotului este definită simplu, printr-o succesiune de situări ale efectorului final care, în majoritatea cazurilor, sunt destul de îndepărtate unele de altele. Situarea *k* este exprimată prin matricea coloană $\mathbf{X}_k$ a coordonatelor operaționale (de dimensiune 6 în 3D şi 2 în 2D), care constituie *punctul* $\mathbf{X}_k$ al spațiului operațional. Comanda mişcării *punct cu punct* se numeşte *comandă punct cu punct.*

Într-un *punct* $\mathbf{X}_k$ al traiectoriei, efectorul final poate avea viteza $\mathbf{v}_k = \dot{\mathbf{X}}_k = 0$ sau diferită de zero. În primul caz, *punctul* $\mathbf{X}_k$ se numeşte *punct de oprire*, iar în al doilea caz, *punctul* $\mathbf{X}_k$ se numeşte *punct de trecere.*

Din punctul de vedere al automatistului, comanda punct cu punct înseamnă comanda succesiunii coordonatelor cuplelor cinematice $\mathbf{q}_k$, care asigură situările $\mathbf{X}_k$ ale efectorului final. În simulare, coordonatele cuplelor cinematice $\mathbf{q}_k$ se obțin prin:

$$\mathbf{q}_k = \mathbf{MGI}(\mathbf{X}_k) \tag{1.1}$$

Pentru a asigura deplasarea din *punctul k* în *punctul k+1*, trebuie ca, începând cu configurația $\mathbf{q}_k$, să se comande deplasarea coordonatelor cuplelor cinematice:

$$\mathbf{D}_k = \Delta\mathbf{q}_k = \mathbf{q}_{k+1} - \mathbf{q}_k \tag{1.2}$$

Aceste deplasări sunt variabilele de intrare pentru mecanismul de acționare a axelor manipulatorului.

Cupla cinematică conducătoare (motoare) are un captor de poziție, şi, uneori, şi un captor pentru viteză, ale căror informații, captate de către informatica de comandă, servesc la realizarea reglării poziției. Returul informației pentru poziție nu este activ în timpul mişcării cuplei cinematice, ci doar în momentul atingerii *punctului* de oprire, pentru stabilirea acestui *punct*. Deplasarea $D_k^j = \Delta q_k^j$ a cuplei cinematice *j* se realizează

BUPT

după o anumită lege de variație a vitezei, impusă de generatorul vitezelor prescrise, care acționează asupra sistemului de reglare a vitezei mecanismului de acționare. Corecția buclei de viteză este, în general, un PID (corector proporțional integrat derivat). Acesta conferă mecanismului de acționare o comportare qvasi-ideală, adică viteza de ieşire a mecanismului de acționare este practic egală cu valoarea prescrisă de generatorul vitezelor. În aceste condiții, este posibil să se determine legile de viteză, respectându-se restricțiile proprii fiecărei acționări şi realizându-se coordonarea cuplelor cinematice.

Pentru cazurile de deplasare pură pot exista mai multe tipuri de mişcări:

- mişcările între două *puncte*, traiectoria nefiind constrânsă între *puncte* (mişcări libere);
- mişcările între două *puncte "via"* (*puncte intermediare*), specificate pentru a evita obstacolele, traiectoria fiind liberă între *punctele* intermediare;
- mişcările între două *puncte*, traiectoria fiind constrânsă între aceste *puncte* (traiectorie rectilinie de exemplu);
- mişcările între două *puncte "via"* (*puncte* intermediare), traiectoria fiind constrânsă între *punctele* intermediare;
- urmărirea unui obiect mobil pe un transportor sau mai general, generarea mişcărilor robotului cu restricții provenite de la senzorii exteroceptivi.

În primele două cazuri, generarea mişcării se poate face direct în spațiul cuplelor cinematice: ea se traduce printr-o secvență de poziții ale cuplelor (adică de viteze şi accelerații) constituind restricțiile la care este supusă. În ultimele trei cazuri, traiectoria fiind descrisă în spațiul operațional, este preferabil să se raționeze în acest spațiu. Legea de comandă generată trebuie apoi să fie transformată în coordonatele cuplelor de către transformatorul de coordonate (**MGI**).

1.4. GENERAREA MIŞCĂRII ÎN SPAȚIUL CUPLELOR CINEMATICE

Dacă se lucrează în spațiul cuplelor cinematice conducătoare, generatorul de traiectorie furnizează variațiile în timp ale tuturor coordonatelor cuplelor, împreună cu primele două derivate în raport cu timpul ale acestora, în scopul descrierii traiectoriei dorite. Generarea mişcării în acest spațiu (fig.1.1) prezintă mai multe avantaje:

- mişcarea este minimă pentru fiecare cuplă cinematică conducătoare;
- generarea traiectoriei are loc în timp apropiat de cel real;
- traiectoriile cuplelor sunt uşor de generat;
- sunt necesare mai puține calcule "on-line" pentru că nu se face apel la transformatorul de coordonate;
- mişcarea nu este afectată la trecerea prin configurații singulare;
- restricțiile de viteză şi de torsor sunt cunoscute cu precizie deoarece acestea corespund limitelor fizice ale servomotoarelor;
- traiectoria este generată direct în funcție de variabilele controlate pe parcursul mişcării.

Dezavantajul acestei metode constă în faptul că nu poate fi impusă geometria traiectoriei în spațiul operațional. Între două *puncte* date, efectorul final se deplasează în mod imprevizibil, dar repetabil, existând deci riscul de coliziune când robotul evoluează

BUPT

în spații aglomerate. Acest mod de generare a mişcării este, în consecință, potrivit pentru a realiza deplasări rapide într-un spațiu de lucru degajat.

Generarea mişcării în spațiul coordonatelor cuplelor cinematice conducătoare este prezentată schematizat în fig.1.1.

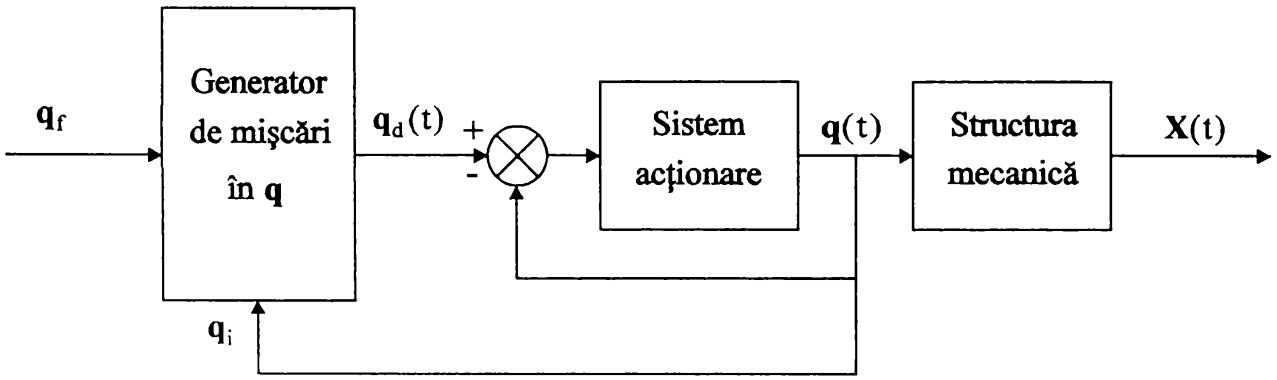


Fig. 1.1. Generarea mişcării în spațiul cuplelor cinematice

În general, algoritmul de bază pentru generarea mulțimii *punctelor* de pe traiectoria cuplelor este de forma:

$t=t_0$;

bucla: Aşteaptă următorul interval de control

$t=t+\Delta t$;

q(t)=poziționarea cuplelor robotului la momentul anterior t;

if $t=t_f$, then părăseşte bucla;

altfel, go to *bucla*;

unde Δt este perioada de eşantionare. Conform algoritmului de mai sus, calculul traiectoriei constă din parametrizarea traiectoriei q(t), care trebuie actualizată pentru fiecare interval de eşantionare ales. Rezultă astfel patru restricții asupra generării traiectoriei:

- *punctele* traiectoriei trebuie să fie uşor calculabile într-o manieră iterativă;
- pozițiile intermediare trebuie să fie determinate şi specificate în mod determinist;
- continuitatea pozițiilor cuplelor şi a primelor două derivate ale acestora trebuie garantate, astfel încât traiectoriile generate ale cuplelor să fie netede;
- fenomenele nedorite, precum oscilațiile, trebuie să fie minime.

1.5. GENERAREA MIŞCĂRII ÎN SPAȚIUL OPERAȚIONAL

Dacă traiectoria este generată în spațiul operațional, generatorul de traiectorie furnizează variațiile în timp ale poziției şi orientării efectorului, respectiv vitezele şi accelerațiile punctului caracteristic al acestuia. Generarea mişcării în spațiul operațional permite controlul geometriei traiectoriei robotului. Dezavantajele acestei metode sunt:

- necesită transformarea în coordonatele cuplelor în fiecare punct al traiectoriei (**MGI**);
- poate eşua când traiectoria calculată trece printr-o poziție singulară;

BUPT

• eşuează de fiecare dată când *punctele* traiectoriei generate nu sunt în volumul accesibil al robotului, sau, de fiecare dată când traiectoria impune o reconfigurare a mecanismului generator de traiectorie;

• limitele în viteze şi torsori ale cuplelor cinematice fiind definite în spaţiul cuplelor, restricţiile corespunzătoare nu pot fi utilizate în spaţiul sarcinii. Se exprimă atunci aceste limite prin valori care reprezintă performanţele medii satisfăcute, oricare ar fi configuraţia robotului. Se impune deci robotului să lucreze sub posibilităţile sale reale.

În cazul controlului traiectoriei carteziene, algoritmul prezentat în paragraful precedent poate fi modificat astfel:

$t=t_0$;

bucla: Aşteaptă următorul interval de control

$t=t+\Delta t$;

$\mathbf{X}(t)$=configuraţia efectorului la momentul anterior t;

$f[\mathbf{X}(t)]$=poziţionarea cuplelor robotului la momentul anterior t, corespunzătoare configuraţiei $\mathbf{X}(t)$;

if $t=t_f$, then părăseşte bucla;

altfel, go to *bucla*;

Spre deosebire de generarea traiectoriei în spaţiul cuplelor cinematice, în acest caz se adaugă calculului traiectoriei parametrizate $\mathbf{X}(t)$ a efectorului, la fiecare pas de eşantionare, realizarea conversiei corespunzătoare $f[\mathbf{X}(t)]$. Funcţia matriceală $\mathbf{X}(t)$ exprimă poziţia şi orientarea efectorului final şi poate fi exprimată printr-o matrice de transformare generală $\mathbf{G}(t)$, de dimensiune 4x4.

În general, generarea mişcării în spaţiul cartezian poate fi realizată în două etape coerente:

• generarea sau selectarea unei mulţimi de *puncte* nodale sau *puncte* de interpolare în coordonate carteziene, în funcţie de anumite reguli;

• specificarea unei clase de funcţii care conectează aceste *puncte* nodale (sau aproximează aceste segmente de traiectorie), conform unor anumite criterii.

Alegerea uneia din cele două metode de generare a mişcării depinde de aplicaţia considerată. Fiecare metodă are propriile sale limite, inerente datorită faptului că restricţiile sunt exprimate fie în spaţiul cuplelor cinematice (viteză, torsor, şocuri), fie în spaţiul operaţional (precizie, obstacole).

Generarea mişcării în spaţiul operaţional este prezentată schematizat în fig.1.2.

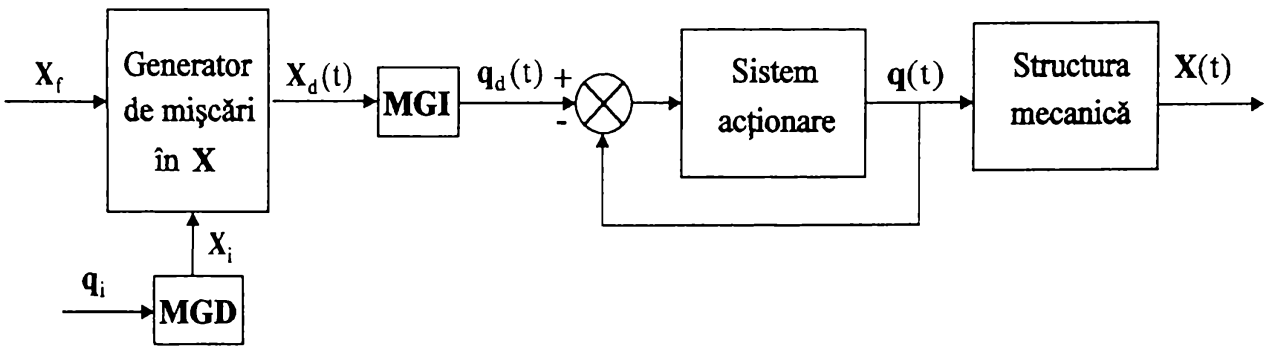


Fig 1.2. Generarea mişcării în spaţiul operaţional

BUPT

1.6. CONTROLUL MIŞCĂRII UNUI ROBOT

Problema controlului mişcării roboţilor poate fi abordată în mod corect doar pe baza modelului dinamic al robotului şi a restricţiilor torsorului, prezentate într-o formă adecvată.

Operaţiunea de control se realizează astfel încât robotul să urmeze o traiectorie prestabilită. Înainte de a pune în mişcare efectorul final al unui robot, este important să se cunoască modelul mediului, deci amplasarea obstacolelor din spaţiul de lucru (*restricţiile furnizate de obstacole*) şi dacă efectorul final trebuie să parcurgă o anumită traiectorie (*restricţiile furnizate de traiectorie*). Aceste două restricţii combinate conduc la patru modele posibile de control [F4], prezentate în Tabelul 1.1.

Tabelul 1.1. Modele de control pentru un robot

RESTRICŢII		FURNIZATE DE OBSTACOLE	
		DA	NU
FURNIZATE DE TRAIECTORIA SPECIFICATĂ	DA	Generarea off-line a traiectoriei fără coliziuni; Parcurgerea on-line a traiectoriei	Generarea off-line a traiectoriei; Urmărirea on-line a traiectoriei
	NU	Control poziţional; Detectarea şi evitarea obstacolelor	Control poziţional

Din acest tabel se observă că problema controlului unui manipulator poate fi divizată în: acţiunea de planificare-generare a mişcării (traiectoriei) şi controlul propriu-zis al mişcării (urmărirea/parcurgerea traiectoriei generate).

În general, schemele de generare a traiectoriilor "interpolează" sau "aproximează" mişcarea dorită printr-o clasă de funcţii polinomiale şi generează o secvenţă de momente de timp în baza setului *punctelor de control,* în scopul controlului manipulatorului din *punctul* de pornire în *punctul* destinaţie. *Punctele* terminale ale traiectoriei pot fi specificate fie în coordonatele cuplelor cinematice conducătoare, fie în coordonate operaţionale. Specificarea în coordonate operaţionale este preferată, întrucât, în acest caz, sunt mai uşor de vizualizat configuraţiile efectorului final al robotului. În situaţia în care se optează pentru lucrul în coordonatele cuplelor, se apelează la soluţionarea modelului geometric invers, pentru realizarea conversiei necesare.

În mod frecvent, există mai multe traiectorii posibile între *punctul* iniţial şi cel final al mişcării. Spre exemplu, se poate dori mişcarea manipulatorului de-a lungul unei traiectorii liniare care leagă *punctele* extreme (iniţial şi final) ale mişcării, sau, se poate opta pentru mişcarea robotului de-a lungul unei traiectorii polinomiale "netede" (continue) care satisface restricţiile de poziţie şi orientare în punctele extreme (traiectorie interpolată în spaţiul cuplelor cinematice).

O tratare sistematică a problemei generării traiectoriei [F4] (fig.1.3) presupune ca generatorul de traiectorie să fie considerat o "cutie neagră" ("black box").

BUPT

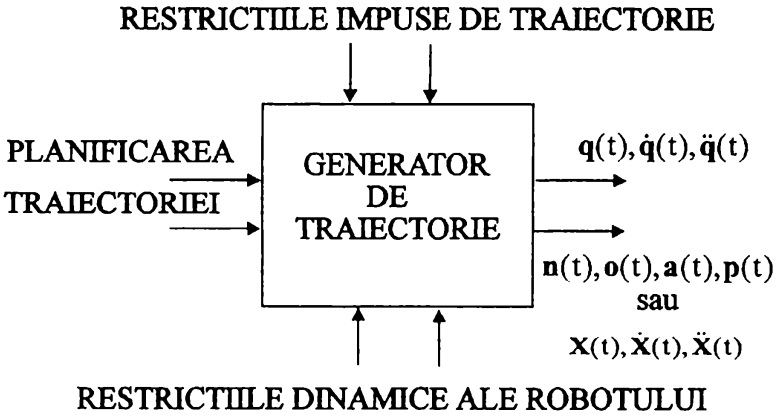


Fig.1.3. Schema bloc a generatorului de traiectorie

Generatorul de traiectorie acceptă variabile de intrare care indică restricţiile impuse de traiectorie şi furnizează la ieşire o secvenţă de configuraţii temporale intermediare ale efectorului manipulatorului (variaţiile în timp ale poziţiei şi orientării **p**(t) şi **n**(t), **o**(t), **a**(t) în spaţiul operaţional şi, respectiv, **q**(t) în spaţiul cuplelor; vitezei **v**(t) în spaţiul operaţional şi, respectiv, $\dot{\mathbf{q}}$(t) în spaţiul cuplelor; acceleraţiei $\dot{\mathbf{v}}$(t) în spaţiul operaţional şi, respectiv, $\ddot{\mathbf{q}}$(t) în spaţiul cuplelor).

Controlul optimal al roboţilor industriali este o problemă dificilă sub trei aspecte:

- Comportamentul dinamic al robotului este pronunţat neliniar şi puternic cuplat. Forţele inerţiale sunt semnificative atunci când mişcarea robotului se face cu viteză mare. Omiţând aceşti termeni se produc erori importante.
- Constrângerile sunt stricte. Când un robot acţionează într-un spaţiu de lucru aglomerat cu obstacole, sau când acesta ajunge într-o poziţie în care are sarcina de a manipula obiecte, trebuie să urmărească exact traiectoria generată pentru a evita orice coliziune posibilă.
- Indicii de performanţă care se pretează la a fi optimizaţi sunt limitaţi.

În problema controlului robotului, *punctele,* iniţial şi final, ale mişcării sunt întotdeauna fixate. Stabilirea timpului necesar mişcării din *punctul* iniţial în cel final presupune ca robotului să i se permită să aibă suficient timp pentru a acţiona. Dacă intervalul de timp este prea scurt, atunci robotul nu-şi poate îndeplini întreaga sarcină; dacă este prea lung, atunci robotul trebuie să aştepte după ce şi-a îndeplinit sarcina. În ambele cazuri, acţiunea robotului nu este eficientă. Dacă intervalul de timp necesar mişcării din *punctul* iniţial în cel final nu se fixează, este "liber", atunci se ajunge la problema timpului minim de parcurgere a traiectoriei. Indicele de performanţă pătratic, $\int \mathbf{e}^T(t)\mathbf{G}(t)\mathbf{e}(t)dt$, unde **G**(t) este matricea forţelor de greutate, poate minimiza deviaţia **e** de la traiectoria planificată. Minimizarea este în funcţie de mărimea medie a intervalului de timp, dar există însă riscul unei coliziuni. Pentru respectarea restricţiilor impuse de traiectorie, se preferă mai întâi minimizarea expresiei $\mathbf{e}^T(t)\mathbf{G}(t)\mathbf{e}(t)$ pentru întregul interval de timp *t*. Această problemă însă este extrem de dificilă. Se poate minimiza controlul efortului mediu $\int |\mathbf{u}(t)|\,dt$ sau controlul energiei medii $\int \mathbf{u}^T(t)\mathbf{G}(t)\mathbf{u}(t)dt$. Prin urmare, din moment ce sistemul de ecuaţii dinamice este puternic neliniar, rezultate semnificative sunt dificil de obţinut.

Din cauza dificultăţii problemei, până înainte de mijlocul anilor 1980, doar câţiva autori [K2], [T2] au încercat să minimizeze timpul sau indicele pătratic de performanţă. În

BUPT

toate publicațiile apărute până în 1983, fie termenii inerțiali au fost omişi, fie condițiile impuse de restricții au fost ignorate, fie ambele.

Este posibil însă şi un alt mod de abordare a problemei optimizării, acela de a genera o traiectorie a robotului de-a lungul căreia optimizarea să nu mai fie o problemă şi apoi, de a manevra robotul astfel încât să urmărească (să parcurgă) acea traiectorie. Astfel, problema controlului optimal s-a divizat în două subprobleme, generarea traiectoriei optime urmată de parcurgerea traiectoriei.

Prima etapă, generarea traiectoriei, constă în decizia asupra profilului vitezei de-a lungul traiectoriei. Această etapă poate fi realizată înaintea execuției mişcării ("off-line"), fapt binevenit întrucât este dorit ca volumul de calcule "on-line" (în timp real) să scadă. A doua etapă, urmărirea (parcurgerea) sau controlul propriu-zis al traiectoriei, constă în calculul "on-line" al forțelor generalizate care trebuie dezvoltate de către servomotoare, în fiecare moment, pentru a obține mişcarea dorită. Această etapă poate folosi ecuațiile diferențiale ale mişcării pentru a calcula forța generalizată ce trebuie dezvoltată de fiecare servomotor. Dacă ecuațiile diferențiale utilizate de către controler constituie un model dinamic perfect, nu este necesară o reacție (feed-back) pentru supravegherea erorii de urmărire a traiectoriei. Totuşi, datorită diferitelor perturbații, este necesară înregistrarea în scopul menținerii acestei abateri în limitele impuse. În timpul execuției mişcării, controlerul calculează forțele generalizate care trebuie dezvoltate, le compară cu cele reale şi le determină pe cele care tind să elimine abaterea, cu o frecvență de actualizare, de obicei, între 10 şi 100 Hz [F4].

Sistemul de control în timp real al robotului este compus din [S15], [S22], [T19]: sistemul de control al structurii mecanice, sistemul de control al axelor, sistemul de control al efectorului final, sistemul de control periferic, sistemul senzorial, sistemul de programare. Optimizarea generării mişcării sub influența sistemului senzorial este reprezentată sugestiv în fig.1.4. Controlul optimal al mişcării unui robot se bazează pe o problemă de optimizare cu restricții combinate, acestea fiind impuse atât în spațiul operațional, cât şi în cel al cuplelor cinematice, după cum se poate vedea în Tabelul 1.2.

Tabel 1.2. Controlul optimal al mişcării unui robot

SPATIUL	OPERATIONAL	CUPLELOR CINEMATICE
RESTRICTII	DE TRAIECTORIE	DE TORSOR / FORTA
	EXPERIMENTAL	PRIN FUNCTII DE APROXIMARE SPLINE
	Robotul este neliniar şi puternic cuplat	Transformarea Jacobiană este punctuală
CONTROL	LA NIVELUL EFECTORULUI FINAL	LA NIVELUL CUPLELOR CINEMATICE CONDUCATOARE

BUPT

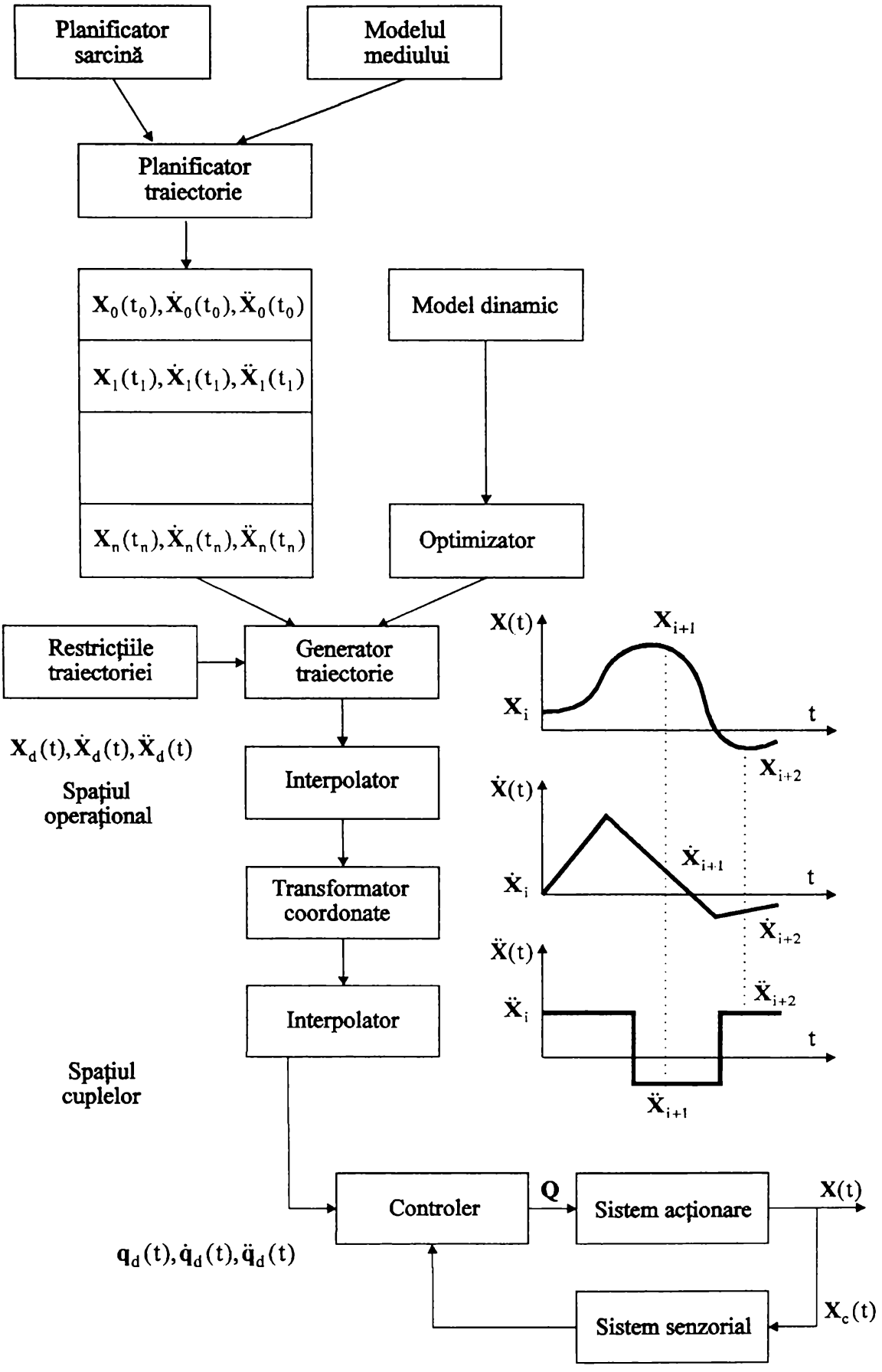


Fig.1.4. Generarea mişcării sub influenţa sistemului senzorial

BUPT

Mişcarea rapidă de-a lungul unei traiectorii geometrice este o problemă centrală în robotica de înaltă performanţă. Configuraţiile care trebuie atinse de către robot sunt, bineînţeles, date de aplicaţia concretă, iar primul pas în atingerea acestora este generarea traiectoriei. Traiectoriile sunt precalculate prin programare interactivă, prin învăţare, cu timp minim de parcurgere sau alte tipuri de optimizare. Traiectoriile nominale obţinute sunt, uzual, la limita torsorului, cel puţin pentru una din cuple.

Metodele aplicate cu succes în generarea traiectoriei sunt acelea care se bazează pe optimizare. Algoritmii de optimizare folosesc modelul dinamic al robotului cu elemente rigide [A5], [B3], [B4], [B6], [D9], [D10], [D18], [D19], [D25], [H9], [I14], [J2], [L6], [L7], [L13], [L16], [L19], [M4], [O1], [R3], [R8], [R15], [V2] sau cu elemente elastice [A1], [B14], [C3], [C5], [H2], [K11], [L19], [N1], [N8], [S24], [S25], [S27], împreună cu o descriere adecvată a traiectoriei şi a restricţiilor impuse pentru evitarea coliziunii [C9], [C10], [K6], [L3] pentru cazul robotului serial fix, sau [B19], [K5], [M2] pentru cazul unui robot mobil. Problemele care necesită minimizarea funcţiei de cost supusă restricţiilor modelului dinamic pot fi exprimate în mod firesc în limbajul teoriei controlului optimal. Metoda uzuală de rezolvare a unor astfel de probleme este folosirea principiului de maxim al lui Potriaghin [B2], [K13], [L4], [L31], [U3].

Problema controlului optimal al mişcării roboţilor este în prezent de mare actualitate în Robotică. Numeroşi cercetători acordă în studiile lor o atenţie deosebită acestei probleme: [A7], [B6], [B10], [B20], [C17], [C19], [F1], [F4], [G4], [G5], [H8], [I2], [K13], [K15], [L8], [L9], [L25], [L26], [L27], [L28], [P6], [P7], [P10], [S21], [S22], [T2], [T19], [V1], [V8], [V13], [W6]. Controlul mişcării robotului poate fi abordat în spaţiul cuplelor, cu profil impus pentru acceleraţie [A3], sau cu interpolare spline cubică [L18], în spaţiul operaţional [L10], [L29], [P3], [P4], [P5], [T1], sau chiar în timpul urmăririi traiectoriei [L22], [W10], [X1]. În funcţie de criteriul de optimizare considerat, controlul poate impune timp minim de parcurgere a traiectoriei [B12], [B13], [C6], [E2], [K2], [L23], [P9], [S16], [S17], [V2], [V3], restricţii de torsor [B11], [B21], [C12], [D1], [D2], [D3], energie consumată minimă [C7], [C11], [D10]. Controlul mişcării poate fi realizat pe baza modelului geometric sau pe baza modelului dinamic [I1], [J3], [J4], [K16], [K18], [L14], [S23], [S26]. Atunci când mişcarea robotului are loc într-un spaţiu cu obstacole, controlul se realizează ţinând cont de acestea [K12], [N7], [P1], [R6], [R7], [S13], [S14], [W7]. O metodă foarte recentă de control foloseşte reţelele neuronale [U1].

BUPT

2. METODE DE GENERARE A TRAIECTORIILOR ÎN SPAȚIUL OPERAȚIONAL

2.1. REPREZENTAREA MIȘCĂRII ÎN COORDONATE CARTEZIENE

Metodele de generare a traiectoriilor robotului în coordonatele cuplelor cinematice folosesc funcții polinomiale (de grad cât mai mic posibil) pentru generarea secvenței *punctelor* de interpolare a traiectoriei fiecărei cuple. Deşi coordonatele cuplelor determină poziția şi orientarea efectorului final al manipulatorului, acestea nu sunt corespunzătuare pentru a specifica sarcina, deoarece majoritatea coordonatelor cuplelor manipulatorului nu sunt ortogonale şi nu se poate separa poziția, de orientare. Pentru un robot mai complex, limbajele de programare sunt dezvoltate pentru controlul unei anumite sarcini. Într-un astfel de sistem, o sarcină este specificată uzual prin secvențe de *puncte nodale* carteziene, prin care trebuie să treacă efectorul final al manipulatorului. Astfel, în descrierea mişcărilor unui manipulator ce are de îndeplinit o anumită sarcină este important formalismul descrierii poziției țintei spre care se mişcă efectorul final al manipulatorului, precum şi cursa în spațiu pe care acesta o parcurge.

Mişcarea manipulatorului poate fi descrisă în funcție de poziția şi orientarea (configurația) efectorului său final, cu ajutorul matricei de transformare generală **G** (a vectorului de poziție **p** şi al versorilor **n, o, a**) [D16], [T3]. Se consideră un sistem de referință fix, ataşat batiului robotului, şi un sistem de referință mobil, ataşat efectorului final (fig.2.1). Matricea de dimensiune 4x4 a transformărilor omogene:

$$\mathbf{G}(t) = \begin{bmatrix} \mathbf{n}(t) & \mathbf{o}(t) & \mathbf{a}(t) & \mathbf{p}(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

descrie configurația efectorului final raportată la sistemul de referință fix.

Aceste coordonate carteziene şi derivatele lor pot fi calculate cu ajutorul coordonatelor cuplelor, pe baza modelelor directe:

$$\mathbf{G}(t) = f(\mathbf{q}(t)) \tag{2.2}$$

$$\dot{\mathbf{G}}(t) = \mathbf{J}\dot{\mathbf{q}}(t) \tag{2.3}$$

$$\ddot{\mathbf{G}}(t) = \mathbf{J}\ddot{\mathbf{q}}(t) + \dot{\mathbf{J}}\dot{\mathbf{q}}(t) \tag{2.4}$$

În relațiile se mai sus, **J** reprezintă matricea Jacobiană a manipulatorului, de dimensiune 6xn, *n* fiind numărul gradelor de libertate ale manipulatorului.

Traiectoria carteziană este generată prin discretizarea funcției **G**(t) într-o secvență de noduri carteziene $\{\mathbf{G}_1, \mathbf{G}_2, ..., \mathbf{G}_n\}$, dar o traiectorie este complet definită prin pozițiile, vitezele şi acceleraţiile din noduri. Pentru traiectoria carteziană, toate aceste date trebuie transformate în corespondentele lor din coordonatele cuplelor, pe baza modelelor inverse. Controlul traiectoriei este apoi realizat la nivelul cuplelor:

BUPT

$$\mathbf{q}_i = f^{-1}(\mathbf{G}_i) \tag{2.5}$$

$$\dot{\mathbf{q}}_i = \mathbf{J}^{-1}\dot{\mathbf{G}}_i \tag{2.6}$$

$$\ddot{\mathbf{q}}_i = \mathbf{J}^{-1}[\ddot{\mathbf{G}}_i - \dot{\mathbf{J}}\dot{\mathbf{q}}] \tag{2.7}$$

Reprezentarea mişcării în spaţiul operaţional se poate realiza şi prin vectorul coordonatelor operaţionale, care, în cazul general, este 6-dimensional, primele trei elemente definind poziţia, iar celelalte trei definind orientarea efectorului final:

$$\mathbf{X} = \left[\mathbf{X}_p \;\; \mathbf{X}_o\right]^T,\; \mathbf{X}_p \subset \mathbf{R}^3,\; \mathbf{X}_o \subset \mathbf{R}^3 \quad \text{sau} \quad \mathbf{X} = \left[x \;\; y \;\; z \;\; \psi \;\; \theta \;\; \phi\right]^T \tag{2.8}$$

Vectorul $\mathbf{X}_p$ poate fi exprimat în coordonate carteziene, cilindrice sau sferice, iar vectorul $\mathbf{X}_o$ poate fi exprimat folosind unghiurile lui Euler sau unghiurile lui Bryant [T9].

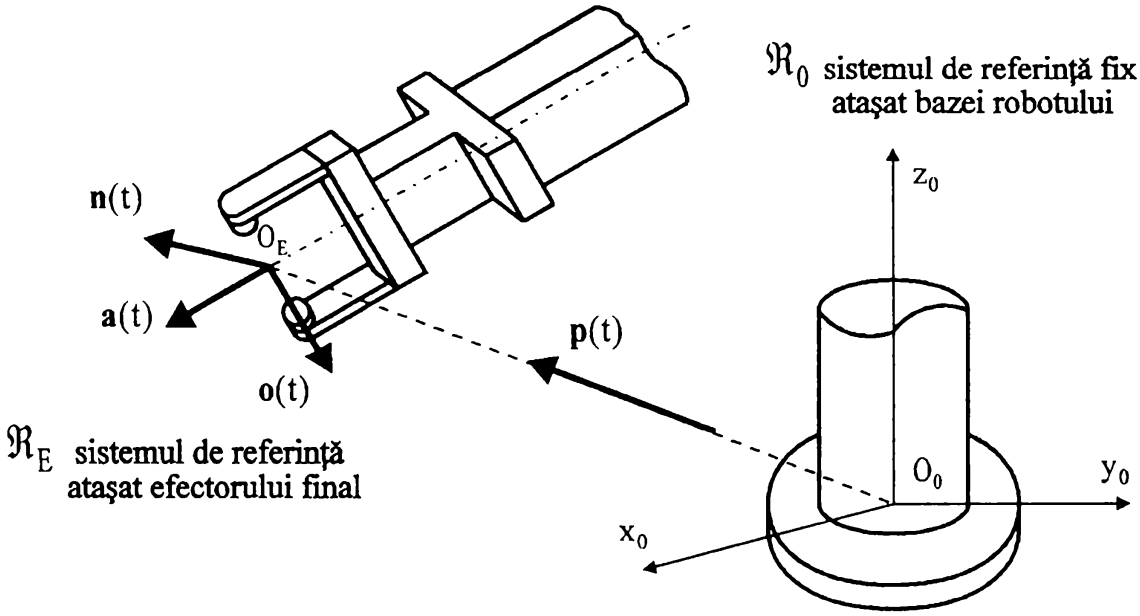


Fig.2.1. Relaţia dintre sistemul de referinţă fix şi cel ataşat efectorului final

Programarea mişcării unui robot se realizează, natural, folosind vectorul operaţional (2.8), iar transformarea de coordonate se realizează pe baza **MGI**, folosind matricea de transformare generală $\mathbf{G}_n^0$. Apare astfel necesitatea trecerii de la matricea $\mathbf{G}_n^0$ la vectorul coordonatelor operaţionale **X**:

$$\begin{aligned} x &= p_x & \psi &= \operatorname{arctg}\left(-\frac{n_z}{o_z}\right) \\ y &= p_y & \theta &= \operatorname{arctg}\frac{\sqrt{n_z^2 + o_z^2}}{a_z} \\ z &= p_z & \phi &= \operatorname{arctg}\frac{a_x}{a_y} \end{aligned} \tag{2.9}$$

dar şi invers, trecerea de la vectorul operaţional **X**, la matricea $\mathbf{G}_n^0$:

BUPT

$$
\begin{aligned}
n_x &= \cos\psi\cos\phi - \sin\psi\cos\theta\sin\phi \\
n_y &= -\cos\psi\sin\phi - \sin\psi\cos\theta\cos\phi \\
n_z &= \sin\psi\sin\theta \\
o_x &= \sin\psi\cos\phi + \cos\psi\cos\theta\sin\phi \\
o_y &= -\sin\psi\sin\phi + \cos\psi\cos\theta\cos\phi \\
o_z &= -\cos\psi\sin\theta \\
a_x &= \sin\theta\sin\phi \\
a_y &= \sin\theta\cos\phi \\
a_z &= \cos\theta \\
p_x &= x \\
p_y &= y \\
p_z &= z
\end{aligned}
\qquad (2.10)
$$

Transformările exprimate prin relațiile (2.9) şi (2.10) folosesc coordonatele carteziene şi unghiurile lui Euler.

În cazul conducerii unui robot în coordonate carteziene se poate impune punctului caracteristic al robotului o traiectorie continuă în spațiul cartezian. Pentru aceasta se determină legea de evoluție în timp a poziției, vitezei şi accelerației punctului caracteristic pe traiectoria impusă. Folosind modelele inverse, se determină pozițiile, vitezele şi accelerațiile cuplelor cinematice conducătoare $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$, care se transmit apoi regulatoarelor corespunzătoare fiecărei cuple cinematice din structura sistemului de acționare.

Între un *punct* inițial şi unul final (țintă), traiectoria este în general formată din segmente de dreaptă. Tranziția între segmentele de dreaptă se face, de obicei, cu controlul accelerației [P4], [T3]. Capetele segmentelor de dreaptă care formează traiectoria sunt definite în raport cu sisteme de coordonate fixe sau mobile, în concordanță cu aplicația particulară. Prima problemă care trebuie rezolvată pentru realizarea unei mişcări carteziene este de a genera *puncte* intermediare între începutul şi sfârşitul segmentului de traiectorie definit de două ecuații de transformare.

Pentru generarea traiectoriei în coordonate carteziene, generatorul de traiectorii execută un algoritm de tipul celui prezentat în capitolul 1.5. Funcția conducere în acest caz se exprimă matriceal. $\mathbf{X}(t)$ este o matrice ale cărei elemente variază în timp şi care exprimă evoluția situării punctului caracteristic de-a lungul traiectoriei. La fiecare etapă de parcurgere a algoritmului se calculează elementele matricelor $\mathbf{X}(t), \dot{\mathbf{X}}(t), \ddot{\mathbf{X}}(t)$ şi se efectuează conversia în coordonatelor cuplelor cinematice conducătoare folosind modelele inverse.

Pentru a putea realiza generarea şi controlul mişcării la nivelul cuplelor cinematice, poate fi aplicată o metodă numerică sau una analitică. Metoda numerică furnizează rezultate numerice, în timp ce, cea analitică poate furniza soluția în forma finală. O soluție analitică corectă este foarte dificil de obținut şi există doar pentru o geometrie specială a robotului. Deci, în mod curent, o metodă numerică este mai generală, dar presupune un volum de calcul considerabil mărit.

Traiectoriile cuplelor sunt generate apoi cuplă cu cuplă. În literatura de specialitate există un mare număr de metode de generare a traiectoriilor, atât în spațiul operațional, cât şi în spațiul cuplelor. Cea mai frecvent utilizată este interpolarea în spațiul cuplelor bazată pe funcțiile polinomiale spline (de gradul trei, patru, cinci, sau combinații ale acestora).

BUPT

Majoritatea abordărilor conduc însă la traiectorii cu siguranță eronate. În scopul micşorării erorilor, o soluție simplă constă în selectarea mai multor *puncte*, soluție care însă cere o cantitate mai mare de calcul şi deci, scade frecvența de eşantionare (creşte pasul de discretizare al mărimii de comandă). O altă metodă constă în a minimiza toate sumele erorilor coordonatelor cuplelor, prin selectarea, în schimb, a unui număr suficient de *puncte* de interpolare. De asemenea, s-au propus diferiți algoritmi de optimizare a generării traiectoriilor cuplelor, pe baza funcțiilor polinomiale spline [A3], [C2], [L18], [L29], etc. Interpolării prin funcții spline îi este dedicat un întreg capitol (capitolul 4).

2.2. TRAIECTORII CARE UNESC *m PUNCTE* ALE SPAȚIULUI DE LUCRU

Într-o aplicație concretă, spațiul de lucru al robotului este populat cu obstacole. Pentru evitarea coliziunii cu acestea, în faza de planificare a traiectoriei, folosind diferite metode de planificare, se introduc *puncte* intermediare între *punctul* inițial şi cel final, garantând ocolirea obiectelor din spațiul de lucru, şi deci, obținerea unei traiectorii libere. Există astfel cinci mulțimi de mărimi caracteristice mişcării:

- $\mathbf{X}_i$ (i = 0, 1, ... m), *punctele* traiectoriei din spațiul operațional;
- $\mathbf{v}_i = (\mathbf{v}, \omega)_i = d\mathbf{X}_i / d\lambda$, (i = 0,1, ..., m), vitezele în *punctele* traiectoriei;
- $\mathbf{a}_i = (\mathbf{a}, \varepsilon)_i = d\mathbf{v}_i / d\lambda$, accelerațiile în *punctele* traiectoriei;
- λ_i (i = 0, 1, ..., m), parametrizarea traiectoriei ($\lambda_0 = 0$, $\lambda_m = 1$);
- t_i (i = 0,1, ..., m), timpii corespunzători mişcării pe traiectorie ($t_i \in [0, t_f]$).

Parametrul traiectoriei $\lambda=\lambda(t)$ poate fi orice funcție de timp. Specificând o parte din mulțimile sau submulțimile mărimilor caracteristice mişcării, se obțin diferite tipuri de traiectorii pentru mişcarea roboților.

Dacă se impune mulțimea vectorilor de poziție, corespunzători unei parametrizări date λ_i (i = 0, 1, ..., m), atunci există câte un polinom de grad *m*, pentru fiecare componentă X_i^j (j = 0,1, ..., n) a vectorului $\mathbf{X}_i$:

$$P^j(\lambda) = C_0^j + C_1^j \cdot \lambda + C_2^j \cdot \lambda^2 + \ldots + C_m^j \cdot \lambda^m; \quad (j = 1, 2, \ldots, n) \tag{2.11}$$

ai cărui $m+1$ coeficienți C_i^j se determină din condițiile:

$$P^j(\lambda_i) = X_i^j; \quad (i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n) \tag{2.12}$$

Determinarea polinoamelor de interpolare $P^j(\lambda)$, pentru fiecare componentă a vectorului $\mathbf{X}$, este incomodă datorită numărului mare de coeficienți C_i^j şi a dificultății calculului extremelor primei şi celei de-a doua derivate. Calculul extremelor derivatelor de ordin superior ale lui $P^j(\lambda)$ este esențial, deoarece vitezele şi accelerațiile coordonatelor cuplelor sunt limitate de sistemul de acționare, iar viteza şi accelerația efectorului final pe traiectorie pot fi impuse de sarcina robotului. De aceea, se preferă interpolarea traiectoriei numai în *puncte* succesive (X_i^j, X_{i+1}^j), cu polinoame de grad cât mai mic, punând condiții de continuitate (inclusiv pentru primele două derivate) în fiecare *punct* X_i^j.

Se consideră o porțiune de traiectorie între *punctele* (X_i^j, X_{i+1}^j) interpolată de un polinom $P_i^j(\lambda)$, $\lambda : [0, 1] \rightarrow \mathbf{R}$, de grad d_i. Cei d_i+1 coeficienți necunoscuți ai polinomului se determină din relațiile:

BUPT

$$P_i^j(0) = X_i^j ; \quad P_i^j(1) = X_{i+1}^j \tag{2.13}$$

şi din condiţiile de continuitate pentru primele două derivate ale lui $P_i^j(\lambda)$.

Numărul coeficienţilor necunoscuţi ai polinoamelor P_i^j (i = 1, ..., m; j = 1, ..., n) este dat de relaţia [D7]: $\sum_{i=1}^{m}(d_i + 1)$, iar numărul total de restricţii care se pot impune polinoamelor $P_i^j(\lambda)$ este de *4m-2*. Rezultă că trebuie îndeplinită condiţia:

$$\sum_{i=1}^{n} d_i = 3m - 2; \quad d_i \geq 1; \quad i = 1, \ldots, m \tag{2.14}$$

Când se specifică vectorii de poziţie $\mathbf{X}_i$ şi vitezele $\mathbf{v}_i$ (i=1, ..., m-1) corespunzătoare unei parametrizări date λ_i (i = 0, ..., m), traiectoria se realizează cu restricţii. Vitezele, iniţială $\mathbf{v}_0$ şi finală $\mathbf{v}_m$, sunt determinate de aplicaţia concretă ce va fi realizată de robot, iar vitezele intermediare $\mathbf{v}_i$ (i = 1, ..., m-1), corespunzătoare *punctelor* de ocolire a obstacolelor, se pot alege după diferite criterii (de exemplu, folosind metoda vitezei medii sau a vitezei medii ponderate).

Cele mai multe traiectorii ale roboţilor specifică mulţimea vectorilor de poziţie $\mathbf{X}_i$ (i = 0, ..., m), vitezele şi acceleraţiile iniţiale şi finale ($\mathbf{v}_0$, $\mathbf{v}_m$, $\mathbf{a}_0$, $\mathbf{a}_m$). În acest fel se asigură ocolirea obstacolelor şi controlul vitezei şi acceleraţiei numai în *punctele* esenţiale ale traiectoriei. Se consideră, ca funcţie de interpolare, mulţimea tuturor polinoamelor $P_i(\lambda)$, generate pentru fiecare pereche de *puncte* sucesive ($\mathbf{X}_i$, $\mathbf{X}_{i+1}$). În stabilirea gradelor polinoamelor $P_i(\lambda)$ se preferă o alegere simetrică ($d_1 = 4$, $d_2 = d_3 = \ldots = d_{m-1} = 3$, $d_m = 4$).

Introducerea unor restricţii suplimentare $\mathbf{v}_i = \mathbf{a}_i = \mathbf{0}$ (i = 1, ..., m-1) simplifică rezolvarea problemei, dar face deplasarea robotului ineficientă. În fiecare *punct* intermediar, pentru ocolirea obstacolelor, robotul se opreşte. Renunţarea la această restricţie suplimentară impune considerarea în calculul coeficienţilor polinoamelor $P_i(\lambda)$ a tuturor vectorilor $\mathbf{X}_i$ (i = 1, ..., m).

Prin specificarea vectorilor de poziţie $\mathbf{X}_i$ şi a vitezelor $\mathbf{v}_i$ (i = 0, ..., m), problema se reduce la *m* traiectorii distincte. Algoritmul de determinare a polinoamelor de interpolare $P_i^j(\lambda)$ se aplică independent pentru fiecare componentă X_i^j a vectorului $\mathbf{X}_i$. Traiectoria rezultată este reuniunea tuturor polinoamelor $P_i^j(\lambda)$ pentru fiecare componentă a vectorului $\mathbf{X}$. După determinarea traiectoriei se trece la calcularea mişcării pe traiectorie (funcţia $\lambda(t) : [0, t_f] \rightarrow \mathbf{R}$). Rezolvarea acestei probleme este simplă numai pentru clasa funcţiilor liniare: $\lambda(t) = \alpha \cdot t$; $t \in [0, t_f]$, unde factorul de proporţionalitate α se calculează în funcţie de limitele de viteză şi acceleraţie ($\mathbf{K}_v$, $\mathbf{K}_a$) impuse de sistemul de acţionare a robotului.

Analizând funcţiile de interpolare folosite de metodele de generare a traiectoriilor, se observă că marea majoritate sunt polinoame. Acest lucru este explicabil prin avantajele oferite de funcţiile polinomiale faţă de cerinţele impuse mişcării roboţilor. O parte din avantajele folosirii polinoamelor, ca funcţii de interpolare a traiectoriilor, constau în [D7]:

- posibilitatea de evaluare "a priori" a vitezei şi acceleraţiei maxime atinse pe parcursul unei mişcări;
- asigurarea acceleraţiei zero la pornirea şi oprirea robotului;

BUPT

•asigurarea unei mişcări line la intersecţia segmentelor şi pe segmentele de dreaptă ale traiectoriilor generate în coordonate carteziene;

•asigurarea continuităţii derivatelor de ordin superior (şocul - $\dddot{a}(t)$), prin mărirea gradului polinomului;

•volumul mic de calcule "*on-line*";

•volumul calculelor creşte liniar cu gradul polinomului;

•polinoamele cubice pentru acceleraţie aproximează funcţiile de interpolare, evitându-se dezavantajele (acceleraţie diferită de zero la pornire şi oprire) şi păstrându-le avantajele (minimizarea distanţei parcurse pe fiecare axă motoare).

Evident, funcţiile polinomiale prezintă şi dezavantaje, de care trebuie să se ţină seama în activitatea de generare a traiectoriilor roboţilor. În continuare sunt prezentate câteva dintre dezavantaje şi metodele pentru evitarea acestora:

•Calculul coeficienţilor este dificil pentru grade mari ale polinoamelor. Evitarea folosirii polinoamelor de grad superior se realizează prin împărţirea traiectoriei totale care trece prin *m+1 puncte*, în *m* traiectorii separate.

•Polinoamele nu sunt limitate intrinsec (ca funcţiile de tip sinusoidal sau cosinusoidal) pe poziţie, viteză şi acceleraţie. Pentru a asigura viteza şi acceleraţia în limitele admise, se folosesc mai multe tehnici, dintre care se menţionează: optimizarea neliniară cu restricţii; împărţirea traiectoriei iniţiale în subtraiectorii, prin introducerea de *puncte* intermediare; înlocuirea unor zone din traiectoria polinomială cu alte tipuri de funcţii de interpolare; modificarea valorii calculate, prin funcţia polinomială, în apropierea limitelor care nu trebuie depăşite.

$$x_m = \ell + (x - \ell)(L - \ell)/(x + L - 2\ell) \qquad (2.15)$$

unde x_m este valoarea modificată; x este valoarea calculată folosind funcţii polinomiale; L este limita care nu trebuie depăşită; ℓ este limita de atenţionare, care arată că mişcarea robotului se apropie de limita absolută L ($\ell < L$). Folosind relaţia (2.15), valoarea modificată x_m rămâne mai mică decât limita L ($x_m \leq L$), pentru orice valoare calculată cu ajutorul funcţiei de interpolare iniţiale.

•Polinoamele de grad superior pot duce la apariţia de depăşiri şi întoarceri faţă de *punctul* ţintă, deoarece un polinom de grad *n* poate avea *n* intersecţii cu o linie dreaptă. Este indicată, şi în acest caz, divizarea traiectoriei în mai multe polinoame de grad inferior şi aplicarea condiţiilor de continuitate pentru poziţie, viteză şi acceleraţie în *punctele* de precizie. Mişcarea este cu atât mai rapidă, cu cât raza de curbură a traiectoriei este mai mare. Din această cauză, sunt preferate, pentru interpolare, funcţiile spline cubice, care asigură un maxim global al razei de curbură.

Rezultă că *punctele* prin care trebuie să treacă traiectoria se împart în două categorii: *puncte* intermediare, pentru ocolirea obstacolelor, şi *puncte* suplimentare, pentru micşorarea gradului polinoamelor, evitarea vitezelor şi acceleraţiilor mari, etc. Localizarea *punctelor* de divizare a traiectoriei, pentru micşorarea gradului polinomului de interpolare, se poate face arbitrar, dar fără introducerea de depăşiri, sau întoarceri, şi minimizând acceleraţiile şi vitezele rezultate. Alegerea optimală a *punctelor* suplimentare este greu de realizat. Acestea se localizează, în general, pe cel mai lung segment de traiectorie, la o treime din lungimea segmentului, sau pe segmentul iniţial, cât mai aproape de capăt.

BUPT

2.3. GENERAREA MIŞCĂRILOR INTERPOLATE

Contrar comenzii pas cu pas, comanda interpolată este destinată să piloteze manipulatorul astfel încât efectorul său final să descrie o traiectorie dată în spaţiul operaţional, după o anumită lege de mişcare. Aplicaţiile industriale tipice acestui tip de comadă sunt sudarea continuă, decuparea cu laser, vopsirea. Schema de comandă corespunzătoare mişcărilor interpolate a fost prezentată în fig.1.1, respectiv fig.1.2.

Figura 2.2 prezintă o mişcare generală a sistemului de referinţă $\Re_E$ ataşat efectorului final, între o situare iniţială $\Re_i$ şi o situare finală $\Re_f$. O situaţie intermediară este reprezentată prin $\Re_\lambda$, unde λ este un parametru care variază între 0 şi 1, astfel ca $\Re_{\lambda=0} = \Re_i$ şi $\Re_{\lambda=1} = \Re_f$. De fapt, se vor putea considera doi parametri λ [L2]: un parametru $\lambda_p = \lambda_p(t)$, definind legea orară a poziţiei lui $\Re_\lambda$, iar celălalt parametru $\lambda_o = \lambda_o(t)$, definind legea orară a orientării lui $\Re_\lambda$.

Mişcarea $\Re_\lambda / \Re_0$ poate fi definită de legile orare $\lambda_p(t)$ şi $\lambda_0(t)$ astfel:

- traiectoria punctului caracteristic E_λ: $\mathbf{OE}_\lambda = \mathbf{OE}_\lambda(\lambda_p)$;
- orientarea efectorului (evoluţia matricei de rotaţie) $\mathbf{R}_\lambda^0 = \mathbf{R}_\lambda^0(\lambda_0)$.

Variaţia parametrului λ determină coordonatele punctului caracteristic al efectorului final E_λ în $\Re_0$, precum şi evoluţia matricei de rotaţie $\mathbf{R}_\lambda^0$, pentru anumite tipuri de mişcări.

2.3.1. INTERPOLAREA LINIARĂ

Interpolarea liniară este caracterizată de faptul că $\mathbf{OE}_\lambda(\lambda_p)$ şi $\mathbf{R}_\lambda^0$ depind de λ_p şi λ_0 ca funcţii de gradul întâi. Pentru simplificare se poate presupune că legile orare sunt identice.

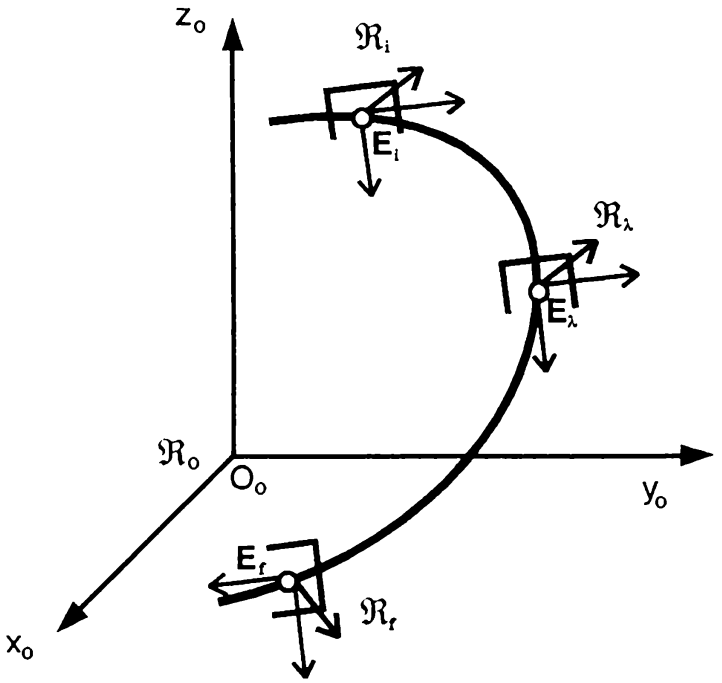


Fig. 2.2. Mişcarea sistemului de referinţă $\Re_\lambda$ ataşat efectorului final

BUPT

Traiectoria punctului caracteristic E_λ

Traiectoria punctului caracteristic E_λ este segmentul de dreaptă E_iE_f. Ecuațiile sale vectoriale şi matriceale proiectate în $\Re_0$ se scriu:

$$\mathbf{OE}_\lambda = \mathbf{OE}_i + \lambda \mathbf{d} \qquad \mathbf{X}_\lambda^0 = \mathbf{X}_i^0 + \lambda \mathbf{d}^0 \tag{2.16}$$

cu $\mathbf{d} = \mathbf{E}_i\mathbf{E}_f$. S-au notat cu $\mathbf{X}_\lambda^0$ coordonatele carteziene ale lui $\mathbf{E}_\lambda$ în $\Re_0$, cu $\mathbf{X}_i^0$ coordonatele carteziene ale lui $\mathbf{E}_i$ şi cu $\mathbf{d}^0$ componentele lui $\mathbf{d}$ în $\Re_0$.

Rotația efectorului final

Dacă interesează doar mişcarea de orientare, se poate presupune că $\Re_i$, $\Re_f$ şi $\Re_\lambda$ au o origine comună. $\Re_i$ şi $\Re_f$ fiind cunoscute, $\Re_\lambda$ va suferi o rotație în jurul unei axe de versor $\mathbf{u}$, cu unghiul $\lambda\theta$, unde θ şi $\mathbf{u}$ sunt parametrii de rotație proprie de la $\Re_i$ la $\Re_f$. Se poate scrie:

$$\mathbf{R}_\lambda^i = \mathbf{R}(\mathbf{u}, \lambda\theta) \tag{2.17}$$

ceea ce permite obținerea lui $\mathbf{R}_\lambda^0$:

$$\mathbf{R}_\lambda^0 = \mathbf{R}_i^0 \mathbf{R}(\mathbf{u}, \lambda\theta) \tag{2.18}$$

În majoritatea cazurilor, θ şi $\mathbf{u}$ sunt direct cunoscute. Se cunosc de asemenea matricele $\mathbf{R}_i^0$ şi $\mathbf{R}_f^0$. Se poate calcula $\mathbf{R}_f^i = \mathbf{R}_0^i \mathbf{R}_f^0$, apoi θ şi $\mathbf{u}$ prin indentitatea $\mathbf{R}(u,\theta) \equiv \mathbf{R}_f^i$ [D16], [L2]:

$$\theta = \pm \text{Arccos} \frac{\text{Trace}\,\mathbf{R}_f^i - 1}{2} \quad \text{sau} \quad \theta = \pm \text{Arccos} \frac{n_x + o_y + a_z - 1}{2} \tag{2.19}$$

$$\tilde{\mathbf{u}} = \frac{1}{2\sin\theta}\left(\mathbf{R}_f^i - \mathbf{R}_i^f\right), \quad \sin\theta \neq 0 \tag{2.20}$$

Relația de mai sus furnizează componentele lui $\mathbf{u}$ în $\Re_i$ (sau $\Re_f$), cu care se poate calcula matricea $\mathbf{R}(\mathbf{u}, \lambda\theta)$ care intervine în relația (2.18):

$$\mathbf{R}(\mathbf{u}, \lambda\theta) = \mathbf{I} + \sin(\lambda\theta)\tilde{\mathbf{u}} + \left[1 - \cos(\lambda\theta)\right]\tilde{\mathbf{u}}^2 \tag{2.21}$$

Referitor la cele prezentate mai sus se pot face unele observații :

- Dacă $\theta = 0$, mişcarea efectorului final este o translație rectilinie.
- Dacă $\mathbf{d} = 0$, mişcarea efectorului final este o rotație pură, în jurul unei axe de versor $\mathbf{u}$, care trece prin *punctul* $\mathbf{E}_i$.

Pentru acest tip de mişcare se poate utiliza, de exemplu, abordarea cinematică.

2.3.2. GENERAREA MIŞCĂRILOR STANDARD PRIN ABORDARE CINEMATICĂ

Mişcările standard sunt mişcările de translație sau rotație ale efectorului final, reprezentat de sistemul de referință ataşat acestuia $\Re_E$, de-a lungul sau în jurul unei axe de referință, în $\Re_0$ sau în $\Re_E$. De exemplu, pornind de la o configurație oarecare dată a manipulatorului, se doreşte realizarea translației lui $\Re_E$ după aza z_E a dispozitivului de prehensiune, sau a rotației dispozitivului de prehensiune în jurul axei sale z_E.

BUPT

Mişcarea dorită a lui $\Re_E/\Re_0$ poate fi definită printr-o legătură fictivă de tip translaţie sau rotaţie, stabilită între efectorul final şi batiul robotului. Figura 2.3 prezintă mişcarea dorită a efectorului final care în cazul (a) este o translaţie rectilinie de la A spre B. Figura 2.3.b prezintă legătura de translaţie fictivă a unei axe de versor **u**. În cazul general, torsorul geometric care caracterizează legătura, într-un punct **P** este [L2]:

$$\hat{\mathbf{G}}(\mathbf{P}) = \begin{bmatrix} \mathbf{u} \\ \mathbf{PA} \times \mathbf{u} \end{bmatrix} \quad \text{pentru rotaţie} \tag{2.22}$$

$$\hat{\mathbf{G}}(\mathbf{P}) = \begin{bmatrix} \mathbf{0} \\ \mathbf{u} \end{bmatrix} \quad \text{pentru translaţie} \tag{2.23}$$

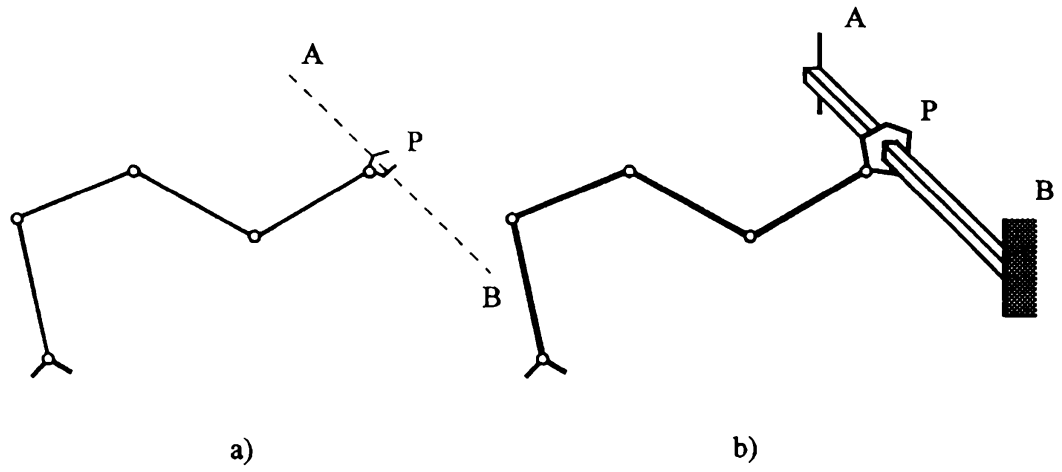


Fig. 2.3. Legătura fictivă între efectorul final şi batiu

Mişcarea dorită este definită de torsorul cinematic:

$$\hat{\mathbf{C}}(\mathbf{P})_{E/0} = \begin{bmatrix} \boldsymbol{\omega}_{E/0} \\ \mathbf{v}_{P \in E/0} \end{bmatrix} = \dot{\lambda}\,\hat{\mathbf{G}}(\mathbf{P}) \tag{2.24}$$

unde $\lambda = \lambda(t)$ este legea orară a mişcării.

Astfel, forma modelului cinematic direct **MCD**:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{\omega} \\ \mathbf{J}_{v} \end{bmatrix} \Rightarrow \boldsymbol{\omega}_k = \mathbf{J}_{\omega,k}\,\dot{\mathbf{q}} \quad \mathbf{v}_k = \mathbf{J}_{v,k}\,\dot{\mathbf{q}} \tag{2.25}$$

se poate scrie pentru acest tip de mişcare impusă, sub forma:

$$\mathbf{J}\,\dot{\mathbf{q}} = \sum_{j=1}^{n} \dot{\mathbf{q}}^j\,\hat{\mathbf{G}}^j = \dot{\lambda}\,\hat{\mathbf{G}} \tag{2.26}$$

În această relaţie s-a neglijat referinţa la punctul de calcul al torsorilor geometrici şi referinţa la baza proiecţiei. Având introdusă o legătură fictivă între $\Re_E$ şi $\Re_0$, o asemenea ecuaţie se numeşte *ecuaţie cinematică de închidere a lanţului*.

BUPT

2.3.3. GENERAREA MIŞCĂRII PRIN INVERSARE NUMERICĂ

Procedura descrisă în paragraful precedent poate fi folosită în cazul manipulatoarelor sau, mai general, al sistemelor mecanice pentru care nu există **MGI** explicit, sau este prea complex. Procedura este totuşi limitată la mişcările numite standard. Pentru mişcări oarecare, dorite pentru efectorul final, dacă nu se dispune de un **MGI**, se poate folosi metoda de inversare numerică a modelului geometric.

Prin inversare numerică se înţelege rezolvarea unui sistem de *n* ecuaţii neliniare, cu *n* necunoscute. Aceasta este de fapt o problemă de analiză numerică, şi există deja programe specializate pentru rezolvarea sa. Se prezintă în continuare principiul metodei Newton—Raphson. Modelul geometric direct **MGD** se poate scrie sub forma:

$$F_i(\mathbf{q}) = f_i\left(q^1, \ldots, q^n\right) - \mathbf{X}_i = 0;\ i = \overline{1,\ n} \tag{2.27}$$

unde $\mathbf{q} \in \mathbf{R}^n$, $\mathbf{X} \in \mathbf{R}^n$, iar $f : \mathbf{R}^n \rightarrow \mathbf{R}^n$ este o funcţie continuă şi diferenţiabilă care transformă coordonatele cuplelor în coordonate operaţionale.

Se dau valorile iniţiale $\mathbf{q}_{(0)}^j$, vecine soluţiilor $\mathbf{q}^j$ căutate. Prin dezvoltarea în serie Taylor, în vecinătatea lui $\mathbf{q}_{(0)}^j$, şi reţinerea termenilor de gradul 1 se obţine:

$$F_i(\mathbf{q}) \cong F_i\left(\mathbf{q}_{(0)}\right) + \sum_{j=1}^{n} \frac{\partial F_i\left(\mathbf{q}_{(0)}\right)}{\partial q^j} \delta q_{(0)}^j \tag{2.28}$$

Considerând că $F_i(\mathbf{q}) = 0$, relaţia (2.28) se poate scrie sub formă matriceală:

$$\mathbf{J}\left(\mathbf{q}_{(0)}\right) \delta \mathbf{q}_{(0)} = -F\left(\mathbf{q}_{(0)}\right) \tag{2.29}$$

unde $\mathbf{J}(\mathbf{q}_{(0)})$ este jacobianul calculat în *punctul* iniţial $\mathbf{q}_{(0)}$ şi definit cu relaţia:

$$J_i^j = \frac{\partial f_i}{\partial q^j} \tag{2.30}$$

Presupunând că $\det \mathbf{J}(\mathbf{q}_{(0)}) \neq 0$, se poate calcula $\delta\mathbf{q}_{(0)}$:

$$\delta \mathbf{q}_{(0)} = -\mathbf{J}^{-1}\left(\mathbf{q}_{(0)}\right) \cdot F\left(\mathbf{q}_{(0)}\right) \tag{2.31}$$

Cunoscând creşterea $\delta\mathbf{q}_{(0)}$, se obţine un nou *punct* $\mathbf{q}_{(1)}$:

$$\mathbf{q}_{(1)} = \mathbf{q}_{(0)} + \delta \mathbf{q}_{(0)} \tag{2.32}$$

Acest *punct* este considerat ca o nouă iniţializare a relaţiei (2.28), unde $\mathbf{q}_{(1)}$ înlocuieşte $\mathbf{q}_{(0)}$. Se obţine $\delta\mathbf{q}_{(1)}$, cu relaţia (2.31), apoi $\mathbf{q}_{(2)}$, cu relaţia (2.32). În final, formula de recurenţă este:

$$\mathbf{q}_{(i+1)} = \mathbf{q}_{(i)} - \mathbf{J}^{-1}\left(\mathbf{q}_{(i)}\right) \cdot F\left(\mathbf{q}_{(i)}\right) \tag{2.33}$$

şi pentru *i* suficient de mare (i = N) astfel încât $|\mathbf{q}_{(N-1)} - \mathbf{q}_{(N)}| \leq e$ dat, se obţine $\mathbf{q}$ cu precizia *e* impusă.

Metoda poate fi folosită în cazul când manipulatorul are o arhitectură particulară complexă, astfel că inversarea explicită este foarte complicată. Metoda are însă mai multe inconveniente. Este necesar ca iniţializarea procesului să se facă într-un punct $\mathbf{q}_{(0)}$ care să

BUPT

nu fie prea depărtat de soluția exactă, sau de una din soluțiile exacte, în cazul soluțiilor multiple. Metoda numerică nu furnizează, prin încercări, ansamblul soluțiilor multiple.

Sistemul de referință ataşat efectorului final $\Re_E$ trebuie să evolueze între situarea inițială $\Re_E^0$ şi cea finală $\Re_E^f$. Pentru fiecare moment t_k se cunosc cele şase coordonate operaționale $\mathbf{X}_i(t_k)$, i = 1, ..., 6, ale poziției şi orientării sistemului de referință ataşat efectorului final ($\Re_E(t_k)$), în raport cu sistemul de referință fix $\Re_0$, trebuind să se calculeze valorile corespunzătoare ale coordonatelor cuplelor cinematice $\mathbf{q}^j(t_k)$. Pentru aceasta se utilizează formula iterativă (2.33), prin care intervin în $F(\mathbf{q}^j)$ matricea inversă $\mathbf{J}^{-1}$ şi **MGD.** Pentru prima iterație a procesului, la momentul t_k, se va lua $\mathbf{q}(t_{k-1})$ corespunzător situării vecine precedente t_{k-1}. Cum $\mathbf{q}(t_k)$ căutat este vecin cu $\mathbf{q}(t_{k-1})$, în general, vor fi suficiente 2 sau 3 iterații pentru a găsi $\mathbf{q}(t_k)$.

Generarea mişcării prin inversare numerică poate fi uşor aplicată în cazul manipulatoarelor cu şase grade de libertate, când, pentru parcurgerea traiectoriei, este luată în considerare şi evoluția orientării efectorului final. Deoarece generarea mişcării prin inversare numerică, pe baza metodei Newton-Raphson poate fi aplicată doar în cazul unui model geometric format din *n* ecuații neliniare, cu *n* necunoscute, se recomandă să se realizeze mai întâi trecerea de la matricea de transformare generală $\mathbf{G}_6^0$ la vectorul coordonatelor operaționale **X**.

2.3.4. RESPECTAREA RESTRICȚIILOR DE MIŞCARE

O altă problemă care trebuie rezolvată constă în faptul că este necesar să se verifice dacă mişcarea prevăzută este efectiv posibilă şi dacă, în particular, îndeplineşte următoarele condiții:

- respectă limitele vitezelor cuplelor cinematice;
- respectă limitele accelerațiilor cuplelor cinematice;
- nu trece printr-o configurație singulară;
- realizează traiectoria într-o configurație impusă a manipulatorului.

Pentru interpolarea liniară expusă se prezintă în continuare procedura de calcul a vitezelor şi accelerațiilor în spațiul cuplelor cinematice.

2.3.4.1. VITEZELE CUPLELOR CINEMATICE CONDUCĂTOARE

Torsorul cinematic în E_λ al mişcării $\Re_\lambda/\Re_0$ are viteza de rotație $\omega_{E/0} = \dot{\lambda}\theta\mathbf{u}$ şi viteza de translație a punctului E_λ, $\mathbf{v}_E = \dot{\lambda}\mathbf{d}$. În $\Re_0$, torsorul cinematic este:

$$\hat{\mathbf{C}} = \dot{\lambda}\begin{bmatrix}\theta\mathbf{u}_0\\ \mathbf{d}_0\end{bmatrix} = \dot{\lambda}\mathbf{D} \tag{2.34}$$

unde **D** este matricea coloană a componentelor deplasării generale a lui $\Re_\lambda$ în $\Re_0$. Dacă **P** este matricea lui **MCD** exprimată în punctul E_λ şi în raport cu sistemul $\Re_0$, atunci rezultă:

$$\mathbf{P}\dot{\mathbf{q}} = \dot{\lambda}\mathbf{D} \tag{2.35}$$

BUPT

Deoarece s-a presupus că nu sunt atinse configurații singulare în timpul mișcării, det $\mathbf{P} \neq 0$, vitezele cuplelor cinematice se exprimă prin:

$$\dot{\mathbf{q}} = \dot{\lambda}\,\mathbf{P}^{-1}\,\mathbf{D} \tag{2.36}$$

Pentru a exploata această relație și pentru a verifica restricțiile limitelor vitezelor cuplelor cinematice este bine să se folosească parametrul λ. Ținând seama de $\dot{\mathbf{q}} = \dot{\lambda}\,\dfrac{d\mathbf{q}}{d\lambda}$, relația (2.26) devine:

$$\frac{d\mathbf{q}}{d\lambda} = \mathbf{P}^{-1} \cdot \mathbf{D} \tag{2.37}$$

Acestă relație, independentă de legea orară, permite să se calculeze, fie literal, în cazul simplu, fie numeric, $dq^j/d\lambda$ în funcție de λ și să se traseze curbele $g^j(\lambda)$ (pentru care se prezintă un exemplu în figura 2.4), definite de:

$$g^j(\lambda) = K_v^j \Big/ \left|\frac{dq^j}{d\lambda}\right| \quad j = \overline{1,\,n} \tag{2.38}$$

unde K_v^j este limita superioară a lui $\dot{q}^j$. Restricția $\left|\dot{q}^j\right| \le K_v^j$ se scrie deci la fel, presupunând că $\dot{\lambda} > 0$:

$$\dot{\lambda}\,(\lambda) \le g^j(\lambda) \tag{2.39}$$

Astfel, în fig.2.4, după ce se reprezintă curba $\dot{\lambda}(\lambda)$, trebuie să se verifice dacă curba $\dot{\lambda}(\lambda)$ este în întregime sub curba $g^j(\lambda)$. Curba $g^j(\lambda)$ definește deci un domeniu admisibil al legii orare în planul fazelor $(\lambda,\ \dot{\lambda})$. Dacă acesta este cazul pentru j = 1, ... , n, se poate atunci optimiza legea orară "*ridicând*" palierul de viteză până când curba $\dot{\lambda}(\lambda)$ devine tangentă la una din curbele $g^j(\lambda)$. În aceste condiții, timpul de transfer va fi cel mai mic posibil, compatibil cu forma legii orare dorite și cu restricția vitezei cuplei cinematice.

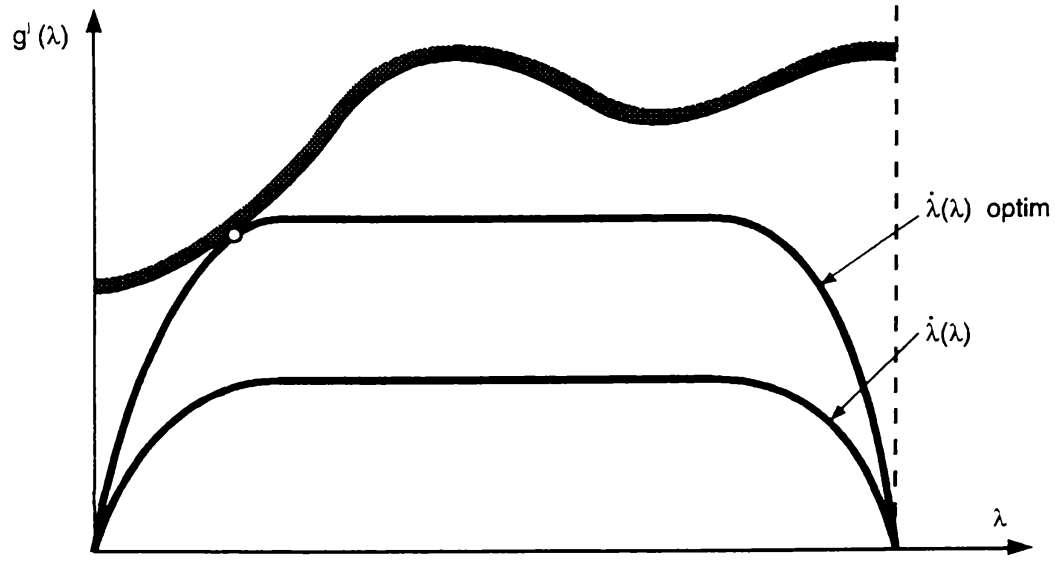


Fig. 2.4. Domeniul admisibil pentru curbele de fază, corespunzător cuplei *j*

BUPT

Exemplu de determinare a curbelor limitelor de viteză pentru un manipulator RT

Fie manipulatorul RT (fig. 2.5.a), pentru care, traiectoria ce trebuie realizată este E_iE_f, cu $E_i = (a, -b)$ şi $E_f = (a, b)$. În acest caz simplu, se pot determina analitic curbele limită ale vitezelor g^j. Pentru acest manipulator se determină modelele directe:

$$\mathbf{MGD} \Rightarrow \begin{cases} x = q^2 \, \mathrm{Cos} q^1 \\ y = q^2 \, \mathrm{Sin} q^1 \end{cases} \qquad \mathbf{MCD} \Rightarrow \mathbf{P} = \begin{bmatrix} -q^2 \, \mathrm{Sin} q^1 & \mathrm{Cos} q^1 \\ q^2 \, \mathrm{Cos} q^1 & \mathrm{Sin} q^1 \end{bmatrix}$$

Un calcul simplu, cu $\mathbf{D} = [0 \; , \; 2b]^T$, conduce la:

$$\mathbf{P}^{-1} \mathbf{D} = \begin{bmatrix} \dfrac{2b \, \mathrm{Cos} q^1}{q^2} & 2b \, \mathrm{Sin} q^1 \end{bmatrix}^T$$

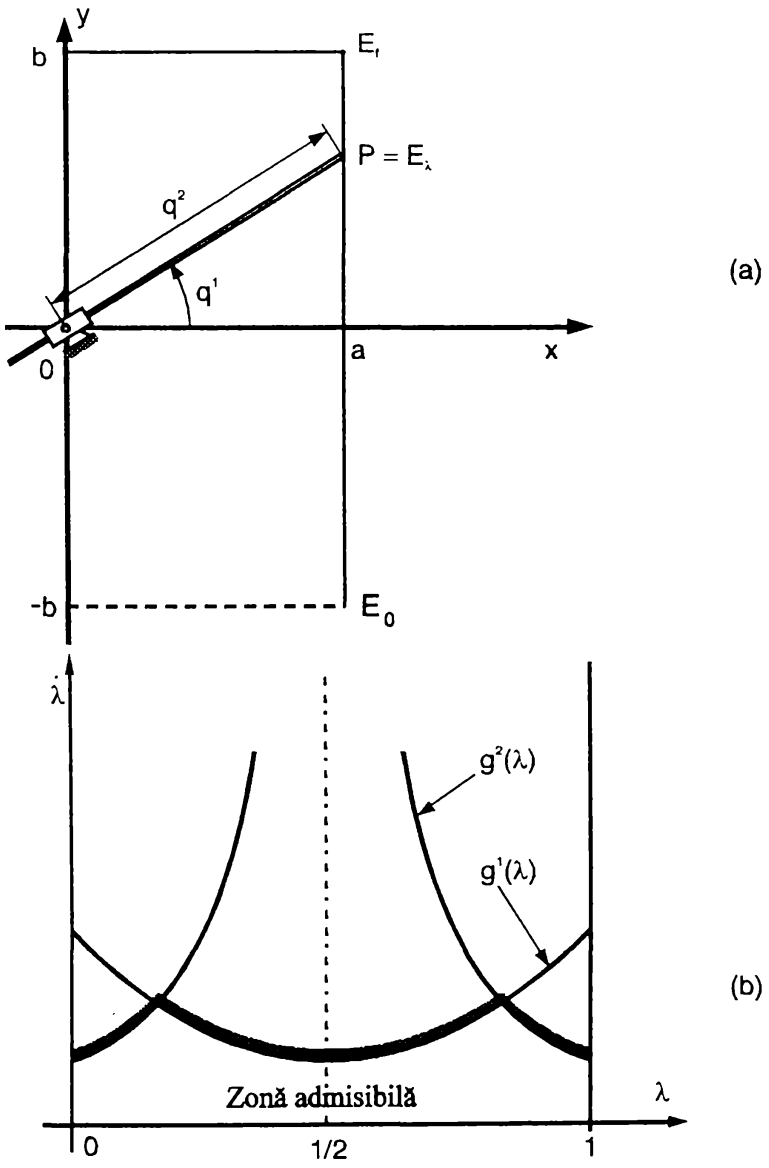


Fig. 2.5. Curbele limitelor de viteză pentru un manipulator plan RT şi o traiectorie rectilinie

BUPT

Traiectoria lui E_λ, definită conform relației (2.16) prin $\mathbf{OE}_\lambda = \mathbf{OE}_i + \lambda\mathbf{d}$, implică: $x = a$ și $y = (2\lambda - 1)b$. Astfel, **MGI** se poate scrie în funcție de λ:

$$\mathbf{MGI}(\lambda) \Rightarrow \begin{cases} q^2 = \sqrt{(2\lambda-1)^2 b^2 + a^2} \\ \mathrm{Sin}q^1 = \dfrac{(2\lambda-1)b}{q^2}, \quad \mathrm{Cos}q^1 = \dfrac{a}{q^2} \end{cases}$$

Se poate obține $\mathbf{P}^{-1}\mathbf{D}$, exprimat în funcție de λ, și deci, din (2.37), $dq/d\lambda$ și în final $g^j(\lambda)$, $j = 1, 2$, din (2.38):

$$g^1(\lambda) = \frac{(2\lambda-1)^2 b^2 + a^2}{2ab} K_v^1, \qquad g^2(\lambda) = \frac{\sqrt{(2\lambda-1)^2 b^2 + a^2}}{2|2\lambda-1|b^2} K_v^2$$

Curbele $g^1(\lambda)$ și $g^2(\lambda)$ sunt prezentate în fig. 2.5.(b). Acestea delimitează zona autorizată pentru legea orară exprimată sub forma unei curbe de fază. Acest exemplu, relativ banal, scoate în evidență complexitatea problemei.

2.3.4.2. ACCELERAȚIILE CUPLELOR CINEMATICE

Pentru determinarea accelerațiilor cuplelor cinematice se încearcă o abordare similară celei pentru viteze. Astfel, **q** fiind funcție de *t* prin intermediul parametrului λ, este ușor de arătat că:

$$\ddot{\mathbf{q}} = \ddot{\lambda}\frac{d\mathbf{q}}{d\lambda} + \dot{\lambda}^2 \frac{d^2\mathbf{q}}{d\lambda^2} \tag{2.40}$$

Se consideră că $\ddot{\lambda}$ și $\dot{\lambda}$ sunt funcții de λ. În această etapă, curba de fază $\dot{\lambda}(\lambda)$ se presupune cunoscută, îndeplinind condițiile cerute de restricțiile vitezelor cuplelor cinematice. Exprimând restricțiile accelerațiilor, rezultă:

$$-\mathbf{K}_a \le \ddot{\mathbf{q}} \le \mathbf{K}_a \;\Rightarrow\; -\mathbf{K}_a - \dot{\lambda}^2(\lambda)\frac{d^2\mathbf{q}}{d\lambda^2} \le \ddot{\lambda}(\lambda)\frac{d\mathbf{q}}{d\lambda} \le \mathbf{K}_a - \dot{\lambda}^2(\lambda)\frac{d^2\mathbf{q}}{d\lambda^2} \tag{2.41}$$

Punerea în aplicare a condiției (2.41) este dificilă. Se poate simplifica mult problema presupunând că accelerațiile cuplelor cinematice sunt mult mai mari la începutul și la sfârșitul mișcării, decât în timpul când efectorul final evoluează cu viteză constantă. Aceasta înseamnă, pe de o parte, limitarea studiului la vecinătățile lui $\lambda = 0$ și $\lambda = 1$, și, pe de altă parte, neglijarea în exprimarea lui $\ddot{\mathbf{q}}$ a termenului $\dot{\lambda}^2(\lambda)\cdot\frac{d^2\mathbf{q}}{d\lambda^2}$. Astfel, condiția (2.41) se simplifică considerabil:

$$\left|\ddot{\lambda}(\lambda)\right| \cdot \left|\frac{d\mathbf{q}}{d\lambda}\right| \le \mathbf{K}_a \tag{2.42}$$

Pentru cupla cinematică *j*, ținând seama de definiția (2.38) a lui $q^j(\lambda)$:

$$\lambda \cong 0 \;\text{ sau }\; \lambda \cong 1 \;\Rightarrow\; \left|\ddot{\lambda}(\lambda)\right| \le \frac{K_a^j}{K_v^j} g^j(\lambda), \qquad j = \overline{1, n} \tag{2.43}$$

BUPT

Pentru a exprima restricția accelerației cuplei cinematice este posibilă exploatarea directă a curbelor limitelor de viteză $q^j(\lambda)$.

2.3.5. ALTE MIŞCĂRI INTERPOLATE

2.3.5.1. MIŞCĂRI CIRCULARE

Majoritatea roboților industriali folosesc fie interpolări liniare ca cele analizate în 2.3.1, fie interpolări circulare. Acestea din urmă permit descrierea unui cerc sau a unei porțiuni de cerc într-un punct Q al lui $\Re_E$. În acest caz, mişcarea de orientare poate îmbrăca două forme:

- este inexistentă: în ansamblu, mişcarea fiind o translație circulară;
- este interpolată liniar între o situare inițială $\Re_i$ şi o situare finală $\Re_f$.

Acest tip de mişcare se tratează ca în paragraful 2.3.1, modificându-se doar ecuația (2.16).

a) <u>Punctul Q este originea lui $\Re_E$</u>

Dacă punctul Q este confundat cu E_λ, scriind $\mathbf{OE}_\lambda = \mathbf{OC} + \mathbf{CE}_\lambda$, rezultă:

$$\mathbf{OE}_\lambda = \mathbf{OC} + \mathbf{R}\left(\mathbf{n},\ \lambda_p\varphi\right)\ \mathbf{CE}_i \tag{2.44}$$

unde **C** este centrul cercului, **n** este normala şi φ este unghiul de rotație necesar pentru a descrie arcul de cerc de la E_i la E_f. Stabilirea acestei relații face apel la formula [L2]:

$$\mathbf{v}^* = \mathbf{R}\left(\mathbf{u},\theta\right)\cdot\mathbf{v} = \mathbf{R}\cdot\mathbf{v} = \mathrm{R}_i^j\,\mathbf{v} \tag{2.45}$$

Expresia (2.8) a matricei de rotație este conservată cu eventual o lege orară de orientare diferită de cea a poziției. Fie:

$$\mathbf{R}_\lambda^0 = \mathbf{R}_i^0\cdot\mathbf{R}\left(\mathbf{u},\ \lambda_0\theta\right) \tag{2.46}$$

Există astfel toate elementele pentru a calcula **MGI**.

b) <u>Punctul Q este un punct oarecare al lui $\Re_E$</u>

Punctul $Q \in \Re_E$ are o mişcare pe un cerc cu centrul C, de normală **n**, după legea orară λ_p. Punctul Q ocupă poziția Q_i cunoscută în $\Re_i$. Poziția punctului Q în $\Re_\lambda$ se notează cu Q_λ. Mişcarea de orientare a lui $\Re_\lambda$ este condusă după o lege de orientare, în a cărei expresie intervin **u**, θ şi λ_0. Poziția lui E_λ în $\Re_0$ depinde deci, în acelaşi timp, de mişcarea circulară a lui Q_λ şi de mişcarea de orientare $\Re_\lambda/\Re_0$. E_λ se obține scriind:

$$\mathbf{OE}_\lambda = \mathbf{OC} + \mathbf{CQ}_\lambda - \mathbf{E}_\lambda\mathbf{Q}_\lambda \tag{2.47}$$

Introducând vectorii cunoscuți $\mathbf{CQ}_i$ şi $\mathbf{E}_i\mathbf{Q}_i$ şi apelând la relația (2.45) se obține:

$$\mathbf{OE}_\lambda = \mathbf{OC} + \mathbf{R}\left(\mathbf{n},\lambda_P\varphi\right)\mathbf{CQ}_i - \mathbf{R}\left(\mathbf{u},\lambda_0\theta\right)\mathbf{E}_i\mathbf{Q}_i \tag{2.48}$$

c) <u>Calculul lui $R(n, \lambda_P\varphi)$</u>

Se face apel la relația (2.48), care, cu $\varphi_\lambda = \lambda_P\varphi$, conduce la relația intrinsecă:

BUPT

$$\mathbf{R}(\mathbf{n},\lambda_P\varphi) = \mathbf{I} + \mathrm{Sin}\varphi_\lambda\ \tilde{\mathbf{n}} + (1-\mathrm{Cos}\varphi_\lambda)\tilde{\mathbf{n}}^2 \qquad (2.49)$$

unde $\tilde{\mathbf{n}}$ este matricea preprodusului vectorial asociată lui **n**. În general, **n** fiind cunoscut prin componentele sale în $\Re_0$ sau în $\Re_i$ se va obține matricea **R**(**n**, $\lambda_P\varphi$) în unul din cele două sisteme de referință.

Există astfel toate elementele pentru a calcula efectiv coordonatele lui E_λ cu relațiile (2.44) sau (2.48), potrivit tipului de traiectorie circulară dorită.

Totuşi, în practică, traiectoria circulară a lui Q_λ nu este dată întotdeauna direct prin centrul C al cercului şi prin normala sa. Cel mai des, traiectoria este definită prin trei puncte necoliniare, care pot fi Q_i, Q_f şi un punct intermediar Q_M. Pornind de la cunoaşterea coordonatelor acestor puncte, trebuie să se determine coordonatele centrului cercului C, unghiul φ şi normala **n** la cerc.

2.3.5.2. MIŞCĂRI OARECARE

Sarcina unui manipulator industrial necesită uneori o mişcare de-a lungul unei curbe predefinite (cerc, elipsă, porțiune de parabolă, etc). Aceste curbe pot fi descrise prin ecuațiile lor parametrice. Cele două categorii de mişcări interpolate considerate până în prezent se caracterizează prin faptul că traiectoria punctului E este o curbă matematică simplă, ale cărei ecuații parametrice sunt cunoscute şi simple. În cazul unei sarcini care necesită o traiectorie complexă, curba ce reprezintă traiectoria este definită printr-un ansamblu discret de puncte P_k. Un segment de traiectorie între punctele P_{k-1} şi P_k, este definit printr-o ecuație parametrică polinomială de forma:

$$0 \le \lambda \le 1, \quad \mathbf{OE}_k = \sum_{i=0}^{N} A_{ki}\,\lambda^i, \quad \lambda = 0 \Rightarrow E_k = P_{k-1}, \quad \lambda = 1 \Rightarrow E_k = P_k \qquad (2.50)$$

Coeficienții vectoriali A_{ki} se determină din condițiile de continuitate pe porțiunile de traiectorie adiacente. Potrivit gradului N al polinomului, pot fi satisfăcute mai multe sau mai puține condiții. Este necesar să se asigure cel puțin continuitatea tangentelor la porțiunile de traiectorie, dar se poate încerca şi obținerea continuității curburilor şi torsiunilor.

2.3.6. PERFORMANȚA OPTIMALĂ ÎN VITEZĂ

Dacă se dispune de o anumită alegere a traiectoriei care trebuie realizată pentru o sarcină dată, trebuie ca aceasta să fie aleasă astfel încât să fie folosită la maxim capacitatea vitezelor manipulatorului.

Vitezele cuplelor cinematice fiind limitate, $\left|\dot{q}^j\right| \le K_v^j$, se propune căutarea, pentru o configurație instantanee dată, a direcției (sau direcțiilor) din spațiul operațional, potrivit căreia (cărora) viteza efectorului final este maximă, ceea ce se exprimă prin $\left|\dot{\mathbf{X}}\right|^2 = \mathrm{maxim}$. Pornind de la $\dot{\mathbf{X}} = \mathbf{J}\dot{\mathbf{q}}$, se presupune că $\det \mathbf{J} \neq 0$. Restricția asupra vitezelor cuplelor cinematice se scrie $\left|\mathbf{J}^{-1}\dot{\mathbf{X}}\right| \le \mathbf{K}_v$. Problema este deci următoarea:

BUPT

Se maximizează $\left|\dot{\mathbf{X}}\right|^2$ impunând restricțiile: $-\mathbf{K}_v \le \mathbf{J}^{-1}\dot{\mathbf{X}} \le \mathbf{K}_v$ (2.51)

Problema de mai sus este o problemă de optimizare a unui criteriu pătratic (în acest caz o "*distanță*"), cu restricții liniare de inegalitate. Această problemă poate fi rezolvată prin metoda programării liniare. Se știe că soluția de dimensiune $n \times n$, pentru **J,** este obținută când există n restricții active. Soluția pentru $\dot{\mathbf{q}}$ este:

$$\dot{\mathbf{q}} = \left[\varepsilon^1 K_v^1 \qquad \varepsilon^n K_v^n\right]^T \tag{2.52}$$

unde $\varepsilon^1, ..., \varepsilon^n$ este o secvență de 1 și -1.

Din punct de vedere geometric, aceasta se exprimă astfel. domeniul ***D*** al spațiului vitezelor $\dot{X}_i$ este un domeniu convex, delimitat de hiperplane; soluția $\mathbf{v}_{max} = \left|\dot{\mathbf{X}}\right|_{max}$ corespunde vârfului cel mai îndepărtat de origine, iar vectorul care leagă originea de acest vârf este $\mathbf{v}_{max}$.

Pornind de la soluția $\dot{\mathbf{q}}$ dată de relația (2.52) se deduce că: $q^j = \varepsilon^j K_v^j t + q_0^j$. Dacă se introduc aceste expresii ale lui q^j în **MGD**, se obțin ecuațiile mișcării efectorului final:

$$\mathbf{X}_i = f_i\left[\mathbf{q}_i(t)\right] \tag{2.53}$$

Eliminând timpul *t*, traiectoria obținută este traiectoria de viteză maximă, care depinde de condițiile inițiale $\mathbf{q}_0$. Dacă variază aceste condiții inițiale, se obține o rețea de de linii de viteză maximă. Utilizarea acestei rețele permite optimizarea sarcinii robotului. De exemplu, pentru o sarcină de transfer între două *puncte*, trebuie să se aleagă cele două *puncte* pe aceeaşi linie de viteză maximă. Această metodă permite deci optimizarea plasării sarcinii în raport cu baza robotului, fiind valabilă atât în cazul comenzii punct cu punct, cât şi în cazul comenzii interpolate.

În mişcarea optimă găsită, restricția $\left|\ddot{q}^j\right| \le K_a^j$ nu mai este luată în considerare, deoarece, $\ddot{\mathbf{q}} = \mathbf{0}$. În acest caz, este convenabil să se verifice restricția asupra forțelor generalizate ale mecanismelor de acționare $\left|\Gamma^j\right| \le K_\Gamma^j$. Modelul dinamic, cu $\ddot{\mathbf{q}} = \mathbf{0}$ se reduce la $\Gamma = \mathbf{D}(\mathbf{q}, \dot{\mathbf{q}})$, unde termenul generic $\Gamma = \mathbf{D}(\mathbf{q}, \dot{\mathbf{q}})$ are o formă pătratică în $\dot{\mathbf{q}}$. Astfel, rezultă:

$$\Gamma^j = D^j(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{q}}^T B^j(\mathbf{q})\dot{\mathbf{q}}, \qquad j = \overline{1, n} \tag{2.54}$$

Scriind relația (2.52) sub forma $\dot{\mathbf{q}} = \varepsilon \mathbf{K}_v$, unde $\varepsilon = \text{diag}(\varepsilon^1, ..., \varepsilon^n)$, condiția care trebuie respectată se scrie:

$$-K_\Gamma^j \le \mathbf{K}_v^T \varepsilon B^j(\mathbf{q}) \varepsilon \mathbf{K}_v \le K_\Gamma^j, \quad j = \overline{1, n} \tag{2.55}$$

<u>Exemplu de determinare a traiectoriilor de viteză maximă la manipulatorul TR</u>

Se tratează un exemplu simplu în 2D, care poate fi rezolvat analitic. Se consideră manipulatorul plan TR, reprezentat în fig. 2.6. Pentru acest manipulator se cunosc:

$$\textbf{MGD} \Rightarrow \begin{matrix} x_1 = q^1 + L\,Cos\,q^2 \\ x_2 = L\,Sin\,q^2 \end{matrix}; \qquad \textbf{MCD}: \dot{\mathbf{X}} = \mathbf{J}\dot{\mathbf{q}} \Rightarrow \mathbf{J} = \begin{bmatrix} 1 & -L\,Sin\,q^2 \\ 0 & L\,Cos\,q^2 \end{bmatrix} \Rightarrow \mathbf{J}^{-1} = \begin{bmatrix} 1 & Tgq^2 \\ 0 & \dfrac{1}{L\,Cos\,q^2} \end{bmatrix}$$

BUPT

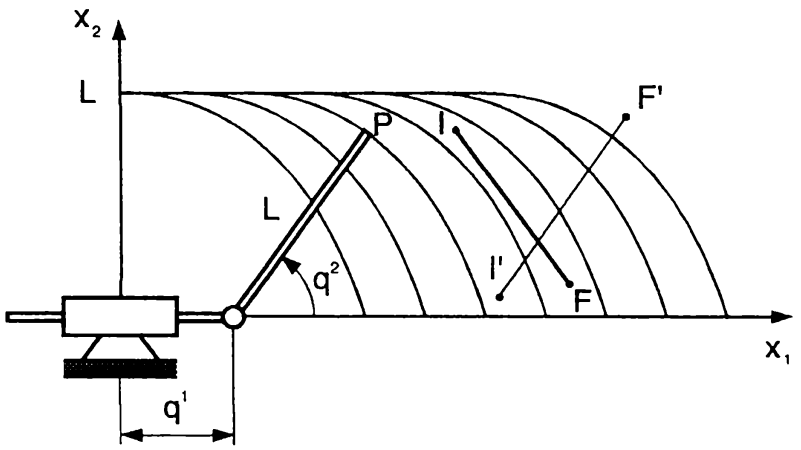


Fig. 2.6. Traiectorii de viteză maximă pentru manipulatorul plan TR

Se consideră $\mathbf{K}_V = [\mathbf{K}_v, \mathbf{K}_\omega]^T$. Sistemul inegalităților (2.51) se scrie:

$$-K_v \le \dot{x}_1 + Tg\,q^2\,\dot{x}_2 \le K_v$$

$$-K_\omega \le \frac{1}{L\,Cos\,q^2}\,\dot{x}_2 \le K_\omega$$

Figura 2.7 reprezintă domeniul D_C al vitezelor admisibile ale cuplelor cinematice (a) şi domeniul D al vitezelor carteziene admisibile (b) ale punctului P. Se poate obsserva corespondența între punctele omoloage ABCD şi A'B'C'D'. Figura reprezintă un caz particular unde $q^2 = \pi/4$, $K_v = V = 1$ m/s, $K_\omega = \omega = 1$ rad/s, L = 1 m. Este evident că v_{max} se obține în punctele B' sau D', omoloagele punctelor B şi D. Pentru relația (2.52) există deci o pereche de soluții:

$$\dot{\mathbf{q}} = [-V \quad \omega]^T \quad \text{sau} \quad \dot{\mathbf{q}} = [V \quad -\omega]^T$$

Linia B'D' reprezintă direcția de viteză maximă şi este tangentă la traiectoria de viteză maximă care trece prin punctul P. Aceasta se obține în maniera următoare: se alege soluția $\dot{\mathbf{q}} = [V \quad -\omega]^T$; se poate pune $q^1 = Vt$ şi $q^2 = -\omega t + \pi/2$ deoarece la t = 0, $\mathbf{X} = [0 \quad L]^T$; înlocuind expresiile lui q^1 şi q^2 în **MGD** şi eliminând timpul, se obține ecuația carteziană a traiectoriei obținute a punctului $\mathbf{X} = [0 \quad L]^T$:

$$x_1 = \frac{V}{\omega} ArcCos\frac{x_2}{L} + \sqrt{L^2 - (x_2)^2}$$

Decalând punctul de plecare, se obține rețeaua reprezentată în fig. 2.7, valabilă pentru:

$$0 \le q^2 \le \frac{\pi}{2}$$

Este evident că pentru a realiza o traiectorie rectilinie IF, sarcina trebuie poziționată în aşa fel încât ea să corespundă mai degrabă lui IF decât lui I'F'. Pentru acest caz simplu, aplicarea teoremei de compunere a vitezelor pentru punctul P permite găsirea soluției mai rapid.

BUPT

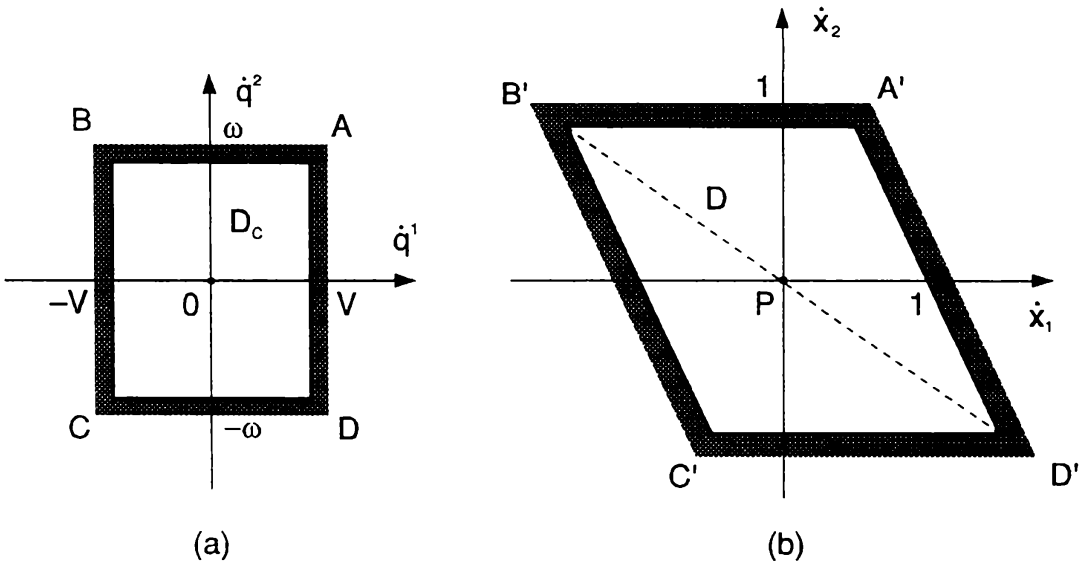


Fig. 2.7. Domeniul admisibil al vitezelor cuplelor cinematice (a) şi al vitezelor carteziene (b) pentru manipulatorul plan TR

2.4. ETAPELE PROCESULUI DE CONDUCERE A UNUI ROBOT

Înainte de a trece la generarea propriu-zisă a traiectoriei, prin parcurgerea algoritmului amintit în paragraful precedent (1.5), trebuie să fie rezolvate două etape preliminare:

a)etapa de determinare a *punctelor* de definire a traiectoriei carteziene;

b)etapa de alegere a funcţiei (sau funcţiilor) conducere şi de calcul al coeficienţilor acesteia (acestora).

a) *Punctele* traiectoriei se pot selecta din expresia analitică a traiectoriei impuse (când aceasta este cunoscută) sau se pot calcula prin procedee de interpolare (când traiectoria este descrisă prin aproximarea ei pe porţiuni cu segmente de curbe simple). *Punctele* selectate sau calculate devin noduri de interpolare pentru etapa de generare.

b) Cu valorile succesive în timp ale funcţiei (funcţiilor) conducere se realizează apoi generarea propriu-zisă a traiectoriei în etapa de generare cu generatorul de traiectorii. Generarea traiectoriei cu funcţii conducere reprezintă cel de-al doilea nivel de interpolare.

Deci conducerea unui robot se desfăşoară în două etape: specificarea mişcării şi generarea mişcării. Specificarea mişcării constă în alegerea funcţiilor conducere şi calculul coeficienţilor din expresiile lor. Se numeşte *funcţie conducere* o funcţie continuă de timp care descrie mişcarea efectorului final al robotului. În principiu, specificarea mişcării poate fi făcută off–line. Calculul off–line este posibil când mişcarea ce urmează a fi efectuată se cunoaşte din timp pentru sarcina respectivă. Pentru sarcini flexibile, în care etapele depind de evenimente produse anterior, pot să apară situaţii când şi specificarea mişcării trebuie efectuată on–line. Funcţiile conducere alese în acest caz trebuie să fie cât mai simple pentru a asigura un timp de calcul cât mai mic (cât mai aproape de timpul real).

BUPT

2.5. GENERAREA MIŞCĂRII ÎN SPAŢIUL CARTEZIAN CU MATRICEA CONDUCERE D(r)

2.5.1. DESCRIEREA MIŞCĂRII PE BAZA ECUAŢIILOR DE TRANSFORMARE

Într-un sistem robotizat programabil mişcarea dorită poate fi specificată printr-o secvenţă de *puncte* nodale carteziene, fiecare dintre ele putând fi descris în termenii transformărilor omogene care leagă sistemul de referinţă ataşat efectorului final al manipulatorului de sistemul coordonatelor spaţiului de lucru (spaţiul operaţional). Coordonatele cuplelor cinematice conducătoare corespunzătoare acestor *puncte* nodale carteziene pot fi calculate pe baza **MGI**, iar pentru racordarea a două segmente adiacente, determinate de *puncte* nodale consecutive, în coordonatele cuplelor, poate fi folosită o funcţie polinomială pătratică. Astfel, efectorul final al manipulatorului este controlat în mişcarea sa de-a lungul unei linii drepte ce conectează două *puncte* nodale. Această tehnică are avantajul de a face posibilă folosirea controlului efectorului final al manipulatorului în urmărirea obiectelor în mişcare. Deşi situările ţintă sunt exprimate prin transformări omogene, este necesar să se specifice însă şi cum trebuie să se mişte efectorul final de la o poziţie la alta (de la o transformare la alta).

În literatura de specialitate sunt prezentate diferite modalităţi de conducere a efectorului final în spaţiul cartezian, astfel încât punctul caracteristic să descrie o anumită traiectorie, iar mişcarea să se efectueze lin [P5], [F4], [T13], [D7]. Cea mai cunoscută variantă este aceea care asigură o deplasare după un segment de dreaptă între două situări succesive (un segment între $\mathbf{POZ}_{i-1}$ şi $\mathbf{POZ}_i$, şi un segment între $\mathbf{POZ}_i$ şi $\mathbf{POZ}_{i+1}$).

În raport cu sistemul de referinţă inerţial ataşat bazei robotului configuraţia acestuia la un moment dat, într-o situare *i*, se exprimă folosind ecuaţiile de transformare:

$$\mathbf{G}_{n_i}^{0}\,\mathbf{EFECTOR}_{\mathrm{EFECTOR}_i}^{n} = \mathbf{COORD}_{\mathrm{BAZA}_i}^{0}\,\mathbf{POZ}_{\mathrm{OBIECT}_i}^{\mathrm{BAZA}} \qquad (2.56)$$

unde:

- $\mathbf{G}_n^0$ este matricea de transformare omogenă care descrie poziţia şi orientarea efectorului final al unui robot cu *n* grade de libertate, în raport cu sistemul coordonatelor de bază.
- $\mathbf{COORD}_{\mathrm{BAZA}}^{0}$ este matricea care caracterizează transformarea sistemului coordonatelor de lucru, în raport cu care se exprimă poziţia şi orientarea obiectului manipulat de robot, faţă de sistemul de referinţă fix. Această matrice de transformare poate fi fixă sau variabilă (**COORD**(s), unde s(t) este o variabilă externă).
- $\mathbf{POZ}_{\mathrm{OBIECT}}^{\mathrm{BAZA}}$ este matricea de transformare care exprimă poziţia şi orientarea obiectului manipulat de robot în sistemul coordonatelor de lucru.
- $\mathbf{EFECTOR}_{\mathrm{EFECTOR}}^{n}$ este matricea de transformare care exprimă poziţia şi orientarea efectorului final în raport cu propriul sistem de referinţă. Această matrice descrie punctul caracteristic a cărui mişcare trebuie controlată.

Dacă $\mathbf{EFECTOR}_{\mathrm{EFECTOR}}^{n}$ se combină cu $\mathbf{G}_n^0$ pentru a forma matricea care descrie poziţia şi orientarea braţului robotului, atunci $\mathbf{EFECTOR}_{\mathrm{EFECTOR}}^{n}$ este matricea unitate de dimensiune 4×4, care se poate neglija. Dacă sistemul coordonatelor de lucru este acelaşi

BUPT

cu sistemul de bază al manipulatorului, atunci $\mathbf{COORD}^0_{BAZA}(t)$ este întotdeauna matricea unitate de dimensiune 4×4.

În ecuația (2.56) se observă că membrul stâng este format din matrice care descriu poziția şi orientarea efectorului final al manipulatorului, pe când membrul drept este format din matrice care descriu poziția şi orientarea obiectului pe care se doreşte ca efectorul manipulatorului să-l prindă. Astfel, ecuația de transformare poate fi rezolvată pentru $\mathbf{G}^0_n$ care descrie configurația manipulatorului pentru prinderea obiectului:

$$\mathbf{G}^0_n = \mathbf{COORD}^0_{BAZA}(t)\cdot \mathbf{POZ}^{BAZA}_{OBIECT}\left(\mathbf{EFECTOR}^n_{EFECTOR}\right)^{-1} \tag{2.57}$$

Dacă matricea $\mathbf{G}^0_n$ este evaluată la o valoare suficient de mare şi convertită în variabilele corespunzătoare ale cuplelor, manipulatorul poate fi acționat pentru a urmări o traiectorie.

Folosind ecuația (2.56), secvența celor N poziții care definesc sarcina, poate fi exprimată astfel:

$$\begin{aligned}
&\mathbf{G}^0_{n1}\left(\mathbf{EFECTOR}^n_{EFECTOR}\right)_1 = \left[\mathbf{COORD}^0_{BAZA}(t)\right]_1\left(\mathbf{POZ}^{BAZA}_{OBIECT}\right)_1\\
&\mathbf{G}^0_{n2}\left(\mathbf{EFECTOR}^n_{EFECTOR}\right)_2 = \left[\mathbf{COORD}^0_{BAZA}(t)\right]_2\left(\mathbf{POZ}^{BAZA}_{OBIECT}\right)_2\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
&\mathbf{G}^0_{nN}\left(\mathbf{EFECTOR}^n_{EFECTOR}\right)_N = \left[\mathbf{COORD}^0_{BAZA}(t)\right]_N\left(\mathbf{POZ}^{BAZA}_{OBIECT}\right)_N
\end{aligned} \tag{2.58}$$

Simplificând notațiile pentru matricele de transformare, respectiv pentru sistemele de referință, ecuațiile de mai sus se pot scrie:

$$\begin{aligned}
&\mathbf{G}_n\,\mathbf{E}_1 = \mathbf{C}_1(t)\mathbf{P}_1\\
&\mathbf{G}_n\,\mathbf{E}_2 = \mathbf{C}_2(t)\mathbf{P}_2\\
&\cdots\cdots\cdots\cdots\cdots\cdots\\
&\mathbf{G}_n\,\mathbf{E}_N = \mathbf{C}_N(t)\mathbf{P}_N
\end{aligned} \tag{2.59}$$

Din pozițiile definite de $\mathbf{C}_i(t)\mathbf{P}_i$ se poate obține distanța între *puncte* consecutive, iar dacă se cunosc şi vitezele liniare şi unghiulare se poate obține timpul necesar T_i mişcării din poziția *i* în poziția *i+1*. Deoarece efectorul final şi sistemele de coordonate în mişcare sunt specificate prin poziții în raport cu sistemul coordonatelor de bază, mişcarea dintr-o poziție în următoarea este cel mai bine dată prin specificarea atât a pozițiilor cât şi a configurației efectorului final în raport cu situarea destinație.

Pentru a face aceasta este necesar să se redefinească poziția prezentă şi efectorul final în raport cu sistemul coordonatelor de lucru. Aceasta poate fi exprimată cu uşurință prin redefinirea transformării $\mathbf{P}_i$ care utilizează notația cu doi indici $\mathbf{P}_i^k$ (poziția $\mathbf{P}_i$ exprimată în raport cu sistemul coordonatelor *k*). Astfel, dacă manipulatorul trebuie controlat din poziția 1 în poziția 2, atunci se exprimă poziția 1 în raport cu propriul sistem de coordonate:

$$\mathbf{G}_n\mathbf{E}_1 = \mathbf{C}_1(t)\cdot\mathbf{P}_1^1 \tag{2.60}$$

şi apoi în raport cu sistemul de coordonate al poziției 2:

$$\mathbf{G}_n\mathbf{E}_2 = \mathbf{C}_2(t)\cdot\mathbf{P}_1^2 \tag{2.61}$$

BUPT

$\mathbf{P}_1^2$ rezultă din aceste ecuații:

$$\mathbf{P}_1^2 = \left(\mathbf{C}_2(t)\right)^{-1} \cdot \mathbf{C}_1(t) \cdot \mathbf{P}_1^1 \cdot \left(\mathbf{E}_1\right)^{-1} \cdot \mathbf{E}_2 \tag{2.62}$$

Scopul ecuației de mai sus constă în a determina $\mathbf{P}_1^2$ când se cunoaşte $\mathbf{P}_1^1$. Astfel, mişcarea între oricare două poziții consecutive *i* şi *i+1* poate fi exprimată ca o mişcare de la:

$$\mathbf{G}_n = \mathbf{C}_{i+1}(t) \cdot \mathbf{P}_i^{i+1} \cdot \left(\mathbf{E}_{i+1}\right)^{-1} \tag{2.63}$$

la:

$$\mathbf{G}_n = \mathbf{C}_{i+1}(t) \cdot \mathbf{P}_{i+1}^{i+1} \cdot \left(\mathbf{E}_{i+1}\right)^{-1} \tag{2.64}$$

unde $\mathbf{P}_i^{i+1}$ şi $\mathbf{P}_{i+1}^{i+1}$ reprezintă transformările amintite mai sus.

În ecuația (2.56), matricea de transformare $\mathbf{POZ}_i$, definită în funcție de matricele $\mathbf{EFECTOR}_i$ şi $\mathbf{COORD}_i$, nu ține seama de destinația fazei respective: $\mathbf{EFECTOR}_{i+1}$ şi $\mathbf{COORD}_{i+1}$. Pentru a realiza legătura cu transformările destinație se defineşte o nouă matrice de transformare $\mathbf{POZ}_i^{i+1}$, care reprezintă transformarea stării curente în funcție de transformările destinație $\mathbf{EFECTOR}_{i+1}$ şi $\mathbf{COORD}_{i+1}$, a cărei expresie este de forma:

$$\mathbf{POZ}_i^{i+1} = \mathbf{COORD}_{i+1}^{-1} \cdot \mathbf{COORD}_i \cdot \mathbf{POZ}_i \cdot \mathbf{EFECTOR}_i^{-1} \cdot \mathbf{EFECTOR}_{i+1} \tag{2.65}$$

Controlul traiectoriei între *punctele* (*i*, *i+1*) impune determinarea unei modificări controlate a transformării $\mathbf{POZ}_i^{i+1}$ în $\mathbf{POZ}_{i+1}$ [P4]. Una din cele mai simple metode de a schimba o poziție în alta este prin intermediul unei translații de-a lungul unei drepte şi unei rotații în jurul unei axe fixe din spațiu. Dacă se poate determina o asemenea axă, atunci se poate produce o mişcare de viteze liniare, respectiv unghiulare controlate. Deci modificarea $\mathbf{POZ}_i^{i+1}$ în $\mathbf{POZ}_{i+1}$ se poate face, în general, printr-o mişcare de translație de-a lungul segmentului de dreaptă care uneşte *punctele* (*i*, *i+1*) şi o mişcare de rotație a efectorului final în jurul unei axe fixe din spațiu, care trece prin punctele respective.

Deoarece determinarea axei echivalente de rotație este dificilă, se consideră că mişcarea între *punctele* (*i*, *i+1*) este compusă dintr-o translație de-a lungul segmentului de dreaptă care uneşte *punctele* (*i*, *i+1*) şi două rotații [P3], [D7]: prima rotație este pentru alinierea efectorului pe direcția dorită, iar a doua rotație este pentru stabilirea (controlul) orientării efectorului final de-a lungul propriei axe. Având în vedere că la majoritatea roboților ultima cuplă este de rotație, a doua rotație (necesară orientării efectorului final) corespunde unei acționări a cuplei finale a robotului. În multe aplicații ale roboților, ca de exemplu ridicarea unei piese aşezate arbitrar pe o bandă rulantă, direcția efectorului este impusă, dar orientarea acestuia variază de la caz la caz. Folosind compunerea celor două rotații, modificarea orientării robotului este determinată de ultima cuplă de rotație.

2.5.2. GENERAREA MIŞCĂRII ÎNTRE DOUĂ SITUĂRI SUCCESIVE

Considerând cazul unei traiectorii formate din segmente de dreaptă şi introducând variabila scalară *r*:

$$r = \frac{t}{T_i} \tag{2.66}$$

unde:

BUPT

• t este timpul curent (real), de la începutul mişcării;

• T_i este timpul total necesar mişcării între *punctele* $(i, i+1)$, transformarea care descrie configuraţia robotului la un moment dat $t \in [0, T_i]$ se poate exprima:

$$\mathbf{G}_n(r) = \mathbf{COORD}_{i+1}(r) \cdot \mathbf{POZ}_i^{i+1} \cdot \mathbf{D}(r) \cdot \mathbf{EFECTOR}_{i+1}^{-1} \tag{2.67}$$

Relaţia (2.67) respectă structura ecuaţiei (2.56), în care:

• $r \in [0, 1]$ este o variabilă externă care controlează mişcarea sistemului de referinţă ataşat;

• $\mathbf{D}(r)$ este transformarea omogenă variabilă care exprimă legea de mişcare între *punctele* $(i, i+1)$. Matricea $\mathbf{D}(r)$ se numeşte *matrice conducere*, deci $\mathbf{D}(r)$ este o funcţie conducere matriceală de timp.

Cele două situări $\mathbf{POZ}_i$ şi $\mathbf{POZ}_{i+1}$ sunt cunoscute din procesul tehnologic, în urma structurării mişcării robotului, structurare exprimată prin ecuaţii de transformare (ecuaţii matriceale), în care fiecare ecuaţie reprezintă una dintre situări.

Transformarea $\mathbf{D}(r)$ satisface relaţiile:

$$\mathbf{D}(0) = \mathbf{I} \tag{2.68}$$

$$\mathbf{POZ}_{i+1}^{i+1} = \mathbf{POZ}_i^{i+1} \cdot \mathbf{D}(1) \tag{2.69}$$

$$\mathbf{D}(1) = \left(\mathbf{POZ}_i^{i+1}\right)^{-1} \cdot \mathbf{POZ}_{i+1}^{i+1} \tag{2.70}$$

Notând $\mathbf{POZ}_i^{i+1}$ cu $\mathbf{P}_i$, $\mathbf{POZ}_{i+1}^{i+1}$ cu $\mathbf{P}_{i+1}$ şi ţinând seama de reprezentarea transformărilor omogene, matricele $\mathbf{P}_i$ şi $\mathbf{P}_{i+1}$ se pot exprima de forma:

$$\mathbf{P}_i = \begin{bmatrix} n_x^{P_i} & o_x^{P_i} & a_x^{P_i} & p_x^{P_i} \\ n_y^{P_i} & o_y^{P_i} & a_y^{P_i} & p_y^{P_i} \\ n_z^{P_i} & o_z^{P_i} & a_z^{P_i} & p_z^{P_i} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{P}_{i+1} = \begin{bmatrix} n_x^{P_{i+1}} & o_x^{P_{i+1}} & a_x^{P_{i+1}} & p_x^{P_{i+1}} \\ n_y^{P_{i+1}} & o_y^{P_{i+1}} & a_y^{P_{i+1}} & p_y^{P_{i+1}} \\ n_z^{P_{i+1}} & o_z^{P_{i+1}} & a_z^{P_{i+1}} & p_z^{P_{i+1}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.71}$$

Prin inversarea simbolică a matricei $\mathbf{P}_i$ şi apoi înmulţirea cu $\mathbf{P}_{i+1}$ se obţine:

$$\mathbf{D}(1) = \begin{bmatrix} \mathbf{n}^{P_i} \cdot \mathbf{n}^{P_{i+1}} & \mathbf{n}^{P_i} \cdot \mathbf{o}^{P_{i+1}} & \mathbf{n}^{P_i} \cdot \mathbf{a}^{P_{i+1}} & \mathbf{n}^{P_i} \cdot \left(\mathbf{p}^{P_{i+1}} - \mathbf{p}^{P_i}\right) \\ \mathbf{o}^{P_i} \cdot \mathbf{n}^{P_{i+1}} & \mathbf{o}^{P_i} \cdot \mathbf{o}^{P_{i+1}} & \mathbf{o}^{P_i} \cdot \mathbf{a}^{P_{i+1}} & \mathbf{o}^{P_i} \cdot \left(\mathbf{p}^{P_{i+1}} - \mathbf{p}^{P_i}\right) \\ \mathbf{a}^{P_i} \cdot \mathbf{n}^{P_{i+1}} & \mathbf{a}^{P_i} \cdot \mathbf{o}^{P_{i+1}} & \mathbf{a}^{P_i} \cdot \mathbf{a}^{P_{i+1}} & \mathbf{a}^{P_i} \cdot \left(\mathbf{p}^{P_{i+1}} - \mathbf{p}^{P_i}\right) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.72}$$

Mişcările de translaţie şi de rotaţie între *punctele* $(i, i+1)$ se execută cu viteză constantă, fiind direct proporţionale cu timpul normat *r*. Dacă *r* variază liniar în raport cu timpul, atunci mişcarea rezultată, reprezentată prin matricea $\mathbf{D}(r)$, va corespunde unei viteze liniare constante şi la două viteze unghiulare constante.

Matricea conducere $\mathbf{D}(r)$ poate reprezenta trei mişcări:

• o translaţie, în coordonate carteziene, de la $\mathbf{P}_i$ la $\mathbf{P}_{i+1}$, reprezentată printr-o matrice de transformare $\mathbf{T}(r)$;

BUPT

• o rotaţie în jurul unei axe oarecare de versor **k**, care asigură alinierea versorului iniţial de apropiere $\mathbf{a}^{Pi}$ cu versorul final $\mathbf{a}^{Pi+1}$, simultan cu desfăşurarea translaţiei carteziene, reprezentată printr-o matrice de transformare $\mathbf{R}_a(r)$;
• o rotaţie în jurul versorului $\mathbf{a}^{Pi+1}$, a versorului de orientare iniţială $\mathbf{o}^{Pi}$ în versorul de orientare finală $\mathbf{o}^{Pi+1}$ a efectorului, simultan cu desfăşurarea translaţiei carteziene, reprezentată printr-o matrice de transformare $\mathbf{R}_0(r)$.
Astfel, matricea conducere **D**(r) are expresia:

$$\mathbf{D}(r) = \mathbf{T}(r) \cdot \mathbf{R}_a(r) \cdot \mathbf{R}_0(r) \tag{2.73}$$

Versorul $\mathbf{k}^{Pi}$ al axei corespunzătoare primei rotaţii se obţine prin rotirea axei y a lui $\mathbf{P}_i$ cu un unghi ψ în jurul axei z:

$$\mathbf{k}^{P_i} = \begin{bmatrix} -S\psi \\ C\psi \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} C\psi & -S\psi & 0 & 0 \\ S\psi & C\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{2.74}$$

Translaţia cartezianã **T**(r) se exprimă:

$$\mathbf{T}(r) = \begin{bmatrix} 1 & 0 & 0 & r \cdot x \\ 0 & 1 & 0 & r \cdot y \\ 0 & 0 & 1 & r \cdot z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.75}$$

unde x, y, z reprezintă componentele translaţiei de la $\mathbf{P}_i$ la $\mathbf{P}_{i+1}$.
Rotaţia $\mathbf{R}_a(r)$ necesară alinierii axelor se exprimă:

$$\mathbf{R}_a(r) = \begin{bmatrix} S^2\psi \cdot V(r\theta) + C(r\theta) & -S\psi \cdot C\psi \cdot V(r\theta) & C\psi \cdot S(r\theta) & 0 \\ -S\psi \cdot C\psi \cdot V(r\theta) & C^2\psi \cdot V(r\theta) + C(r\theta) & S\psi \cdot S(r\theta) & 0 \\ -C\psi \cdot S(r\theta) & -S\psi \cdot S(r\theta) & C(r\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.76}$$

unde s-au folosit notaţiile:

$$\begin{aligned} &C(r\theta) = \cos(r\theta) \qquad S(r\theta) = \sin(r\theta) \\ &C\psi = \cos\psi \qquad S\psi = \sin\psi \\ &V(r\theta) = \mathrm{Vers}(r\theta) = 1 - \cos(r\theta) \end{aligned} \tag{2.77}$$

Rotaţia $\mathbf{R}_a(r)$ reprezintă o rotaţie în jurul versorului $\mathbf{k}^{Pi}$ cu unghiul θ. Rotaţia $\mathbf{R}_0(r)$ reprezintă o rotaţie în jurul versorului de apropiere $\mathbf{a}^{Pi+1}$ al efectorului final cu unghiul ϕ.Rotaţia $\mathbf{R}_0(r)$, în jurul versorului de apropiere $\mathbf{a}^{Pi+1}$, este o rotaţie simplă în jurul axei z:

BUPT

$$\mathbf{R}_0(r)=\begin{bmatrix} C(r\phi) & -S(r\phi) & 0 & 0 \\ S(r\phi) & C(r\phi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.78}$$

unde s-au folosit notaţiile: $S(r\phi) = \sin(r\phi)$ şi $C(r\phi) = \cos(r\phi)$.

Matricea de rotaţie $\mathbf{R}_a(r)$ indică o rotaţie cu un unghi θ în jurul versorului de orientare $\mathbf{o}^{Pi}$ care este rotit cu un unghi ψ în jurul versorului de apropiere. $\mathbf{R}_0(r)$ reprezintă o rotaţie cu unghiul ϕ în jurul versorului de apropiere al efectorului $\mathbf{a}^{Pi}$.

În concluzie, unghiurile ϕ şi θ sunt unghiurile totale de rotaţie necesare ca efectorul final să treacă din poziţia iniţială în poziţia finală, iar mărimile $r\phi$ şi $r\theta$ reprezintă fracţiunile de unghi cu care trebuie rotit efectorul final la fiecare pas ($r \leq 1$). De asemenea, x, y, z reprezintă valorile totale ale translaţiei necesare pentru trecerea din $\mathbf{POZ}_i$ în $\mathbf{POZ}_{i+1}$, iar rx, ry, rz reprezintă fracţiunile de translaţie ce trebuie efectuate la fiecare pas, adică evoluţia în timp a coordonatelor carteziene ocupate de punctul caracteristic în mişcare.

Cu o astfel de structură pentru matricea conducere $\mathbf{D}(r)$ se asigură deplasarea efectorului final în linie dreaptă.

Înlocuind matricele (2.75), (2.76), (2.78) în matricea (2.73), rezultă expresia matricei conducere $\mathbf{D}(r)$ de forma:

$$\mathbf{D}(r)=\begin{bmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.79}$$

unde $\mathbf{n} = \mathbf{o} \times \mathbf{a}$ şi:

$$\mathbf{a}=\begin{bmatrix} -S(r\phi)\left[S^2\psi\, V(r\theta)+C(r\theta)\right]-C(r\phi)S\psi\, C\psi\, V(r\theta) \\ S(r\phi)S\psi\, C\psi\, V(r\theta)+C(r\phi)\left[C^2\psi\, V(r\theta)+C(r\theta)\right] \\ S(r\phi)C\psi\, S(r\theta)-C(r\phi)S\psi\, S(r\theta) \end{bmatrix};\ \mathbf{o}=\begin{bmatrix} C\psi\, S(r\theta) \\ S\psi\, S(r\theta) \\ C(r\theta) \end{bmatrix};\ \mathbf{p}=\begin{bmatrix} rx \\ ry \\ rz \end{bmatrix} \tag{2.80}$$

Problema determinării concrete a matricei conducere $\mathbf{D}(r)$ constă în determinarea valorilor concrete ale unghiurilor ϕ şi θ şi ale translaţiilor x, y, z necesare pentru trecerea din $\mathbf{POZ}_i$ în $\mathbf{POZ}_{i+1}$. Determinarea valorilor ϕ, θ, x, y, z se poate face în faza de specificare a mişcării. În etapa de generare a mişcării se calculează succesiv valori concrete pentru matricea conducere $\mathbf{D}(r)$, dând valori succesiv crescătoare lui *r*. Programul de generare a mişcării între două situări date, în spaţiul cartezian, folosind matricea conducere $\mathbf{D}(r)$, calculează această matrice dând valori succesiv crescătoare lui *r*. Înmulţind valorile lui $\mathbf{D}(r)$ obţinute pentru *k* valori ale lui *r*, cu $\mathbf{POZ}_i$, se obţin *k* situări intermediare între $\mathbf{POZ}_i$ şi $\mathbf{POZ}_{i+1}$. Toate aceste situări intermediare se află pe un segment de dreaptă ce leagă $\mathbf{POZ}_i$ de $\mathbf{POZ}_{i+1}$.

Pentru determinarea valorilor x, y, z, ϕ, θ şi ψ se particularizează r = 1 şi rezultă:

BUPT

$$x = \mathbf{n}^{P_i} \cdot \left(\mathbf{p}^{P_{i+1}} - \mathbf{p}^{P_i}\right) \tag{2.81}$$

$$y = \mathbf{o}^{P_i} \cdot \left(\mathbf{p}^{P_{i+1}} - \mathbf{p}^{P_i}\right) \tag{2.82}$$

$$z = \mathbf{a}^{P_i} \cdot \left(\mathbf{p}^{P_{i+1}} - \mathbf{p}^{P_i}\right) \tag{2.83}$$

$$\operatorname{tg}\psi = \frac{\mathbf{o}^{P_i} \cdot \mathbf{a}^{P_{i+1}}}{\mathbf{n}^{P_i} \cdot \mathbf{a}^{P_{i+1}}} \qquad -\pi \le \psi \le \pi \tag{2.84}$$

$$\operatorname{tg}\theta = \frac{\sqrt{\left(\mathbf{n}^{P_i} \cdot \mathbf{a}^{P_{i+1}}\right)^2 + \left(\mathbf{o}^{P_i} \cdot \mathbf{a}^{P_{i+1}}\right)}}{\mathbf{a}^{P_i} \cdot \mathbf{a}^{P_{i+1}}} \qquad 0 \le \theta \le \pi \tag{2.85}$$

$$\operatorname{tg}\phi = \frac{-S\psi\, C\psi V\theta \cdot \left(\mathbf{n}^{P_i} \cdot \mathbf{n}^{P_{i+1}}\right) + \left[C^2\psi V\theta + C\theta\right] \cdot \left(\mathbf{o}^{P_i} \cdot \mathbf{n}^{P_{i+1}}\right) - S\psi\, S\theta\left(\mathbf{a}^{P_i} \cdot \mathbf{n}^{P_{i+1}}\right)}{-S\psi\, C\psi\, V\theta \cdot \left(\mathbf{n}^{P_i} \cdot \mathbf{o}^{P_{i+1}}\right) + \left[C^2\psi\, V\theta + C\theta\right] \cdot \left(\mathbf{o}^{P_i} \cdot \mathbf{o}^{P_{i+1}}\right) - S\psi\, S\theta\left(\mathbf{a}^{P_i} \cdot \mathbf{o}^{P_{i+1}}\right)}$$

$$-\pi \le \phi \le \pi \tag{2.86}$$

La fiecare perioadă de eşantionare, pentru o valoare a lui *r* ($r \in [0, 1]$), se calculează deci matricea conducere. Se determină apoi configuraţia robotului, sub forma transformării de coordonate, cu ajutorul relaţiei (2.67).

În cazul unei traiectorii formate din mai multe segmente de dreaptă, la care viteza liniară sau unghiulară este constantă pe fiecare segment, dar diferă de la un segment la altul, apare un regim tranzitoriu în punctele de intersecţie a segmentelor. În acest caz este necesar să se rezolve tranziţia între segmentele traiectoriei.

2.5.3. TRANZIŢIA ÎNTRE SEGMENTELE TRAIECTORIEI

În cazul unei traiectorii formate din mai multe segmente de dreaptă, la care vitezele liniare şi unghiulare sunt constante pe fiecare segment, dar diferite de la un segment la altul, apare un regim tranzitoriu în punctele de intersecţie a segmentelor (puncte de discontinuitate pentru viteze).

Pentru a evita discontinuitatea vitezei în punctul de capăt al fiecărui segment de traiectorie, mişcarea trebuie accelerată sau frânată, de la un segment la altul. Aceasta se poate realiza prin iniţierea unei schimbări în viteză cu τ unităţi de timp înainte ca manipulatorul să atingă un punct de capăt şi menţinerea acceleraţiei constante τ unităţi de timp din următorul segment de traiectorie (fig.2.8).

Dacă acceleraţia fiecărei variabile este menţinută la o valoare constantă în intervalul de timp $[-\tau, \tau]$, atunci acceleraţia necesară modificării atât a poziţiei şi a vitezei este [P5], [V12]:

$$\ddot{\mathbf{X}}(t) = \frac{3h}{\tau_1^2}(1-h)\left(\Delta\mathbf{X}_C \frac{\tau_1}{T_1} - \Delta\mathbf{X}_B\right) \tag{2.87}$$

unde $-\tau_1 \le t \le \tau_1$, $h = (t+\tau_1)/2\tau_1$ şi :

BUPT

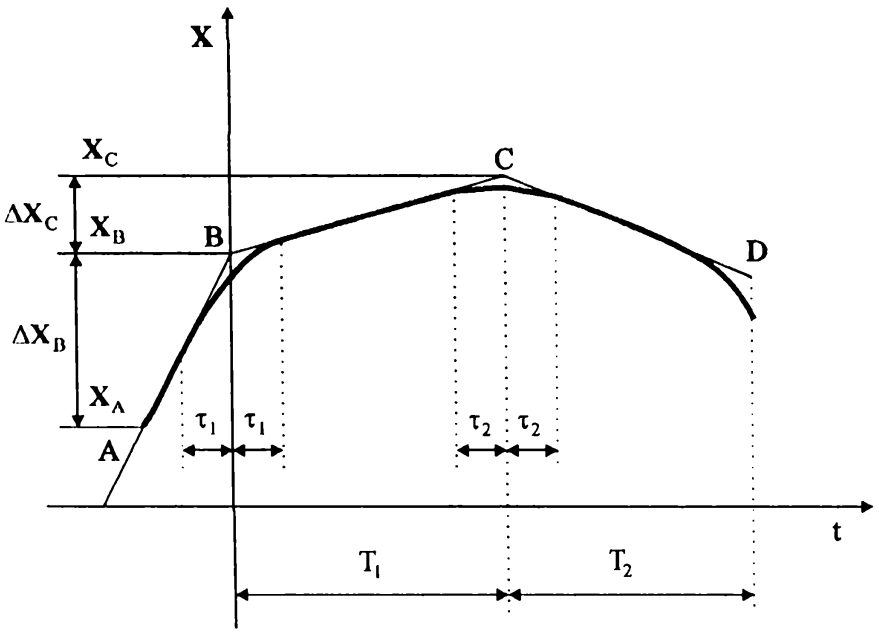


Fig.2.8. Tranziţia între două segmente de traiectorie

$$\ddot{\mathbf{X}}(t)=\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\\ \ddot{\theta}\\ \ddot{\phi}\end{bmatrix} \qquad \Delta\mathbf{X}_C=\begin{bmatrix}x_C-x_B\\ y_C-y_B\\ z_C-z_B\\ \theta_C-\theta_B\\ \phi_C-\phi_B\end{bmatrix} \qquad \Delta\mathbf{X}_B=\begin{bmatrix}x_B-x_A\\ y_B-y_A\\ z_B-z_A\\ \theta_B-\theta_A\\ \phi_B-\phi_A\end{bmatrix} \tag{2.88}$$

unde $\Delta\mathbf{X}_C$ şi $\Delta\mathbf{X}_B$ sunt vectorii ale căror elemente sunt distanţele carteziene şi unghiurile din punctul B spre C, respectiv din A spre B.

Din ecuaţia (2.87), viteza şi poziţia pentru $-\tau \le t \le \tau$ sunt:

$$\dot{\mathbf{X}}(t)=\frac{1}{\tau_1}\left[2h^2(1{,}5-h)\left(\Delta\mathbf{X}_C\frac{\tau_1}{T_1}-\Delta\mathbf{X}_B\right)+\Delta\mathbf{X}_B\right] \tag{2.89}$$

$$\mathbf{X}(t)=\left[h^2(2-h)\left(\Delta\mathbf{X}_C\frac{\tau_1}{T_1}-\Delta\mathbf{X}_B\right)+2\Delta\mathbf{X}_B\right]\cdot h+\mathbf{X}_A \tag{2.90}$$

unde:

$$\mathbf{X}(t)=\left[x\ y\ z\ \theta\ \phi\right]^T;\quad \dot{\mathbf{X}}(t)=\left[\dot{x}\ \dot{y}\ \dot{z}\ \dot{\theta}\ \dot{\phi}\right]^T;\quad h=\frac{t+t_1}{2t_1} \tag{2.91}$$

Pentru $\tau_1 \le t \le T_1 - \tau_2$, mişcarea este descrisă de:

$$\mathbf{X}(t)=h\cdot\Delta\mathbf{X}_C+\mathbf{X}_B \tag{2.92}$$

unde $h = t / T_1$.

Trebuie reţinut faptul că timpul normalizat *h* este diferit pentru diferite intervale de timp.

BUPT

Pentru mişcarea de la A la B şi apoi de la B la C se defineşte un unghi ψ ca o interpolare liniară între mişcările din $-\tau_1 \le t \le \tau$:

$$\psi = (\psi_{BC} - \psi_{AB}) \cdot h + \psi_{AB} \tag{2.93}$$

unde ψ_{AB} şi ψ_{BC} sunt definiţi pentru mişcarea de la A la B şi respectiv de la B la C, ca în relaţia (2.84). Astfel, ψ se modifică de la ψ_{AB} la ψ_{BC}.

Deci, pentru o mişcare din poziţia $\mathbf{P}_i$ în poziţia $\mathbf{P}_{i+1}$ se calculează matricea conducere $\mathbf{D}(r)$ şi apoi matricea de transformare $\mathbf{G}_n(r)$ (relaţia (2.67)). Valorile corespunzătoare ale variabilelor cuplelor se determină pe baza **MGI**. Dacă este necesar, se folosesc funcţii polinomiale pătratice pentru interpolare între punctele obţinute din **MGI**.

Se presupune că traiectoria robotului trebuie să treacă prin punctele A, B şi C.

Se consideră robotul în poziţia A; la momentul $-\tau_1$, se mişcă spre poziţia B unde ajunge în momentul t=0. În momentul T_1 robotul ajunge în poziţia C, descrisă de ecuaţia:

$$\mathbf{G}_{n_C} \cdot \mathbf{EFECTOR} = \mathbf{COORD}(r) \cdot \mathbf{POZ}_C \tag{2.94}$$

unde $\mathbf{G}_{nC}$ exprimă configuraţia $\mathbf{G}_6$ a robotului în momentul când acesta ajunge în $\mathbf{POZ}_C$.

Se evaluează poziţiile A şi B în termenii unor noi sisteme de coordonate mobile **COORD** şi **EFECTOR**, faţă de care se raportează mişcarea. Se exprimă deci matricea $\mathbf{G}_n$ ($\mathbf{G}_{nA}$ şi $\mathbf{G}_{nB}$), care indică situarea (poziţia şi orientarea) efectorului final pentru cele două poziţii A şi B. Se rezolvă în continuare ecuaţiile de transformare corespunzătoare pentru $\mathbf{POZ}_A$ şi $\mathbf{POZ}_B$:

$$\mathbf{POZ}_A = \mathbf{COORD}^{-1}\left(r\big|_{t=-T_2}\right) \cdot \mathbf{G}_{n_A} \cdot \mathbf{EFECTOR} \tag{2.95}$$

$$\mathbf{POZ}_B = \mathbf{COORD}^{-1}\left(r\big|_{t=0}\right) \cdot \mathbf{G}_{nB} \cdot \mathbf{EFECTOR} \tag{2.96}$$

Mişcarea de la punctul B spre punctul C va fi descrisă de o matrice conducere $\mathbf{D}_{BC}(r)$, definită prin:

$$\mathbf{D}_{BC}(1) = \mathbf{POZ}_B^{-1} \cdot \mathbf{POZ}_C \tag{2.97}$$

Se determină astfel punctele intermediare care formează segmentul de traiectorie:

$$\mathbf{G}_n = \mathbf{COORD}(r) \cdot \mathbf{POZ}_B \cdot \mathbf{D}_{BC}(r) \cdot \mathbf{EFECTOR}_C^{-1} \tag{2.98}$$

Totodată se poate descrie o mişcare din punctul B, înapoi spre punctul A, folosind o altă matrice conducere $\mathbf{D}_{BA}(r)$, definită prin:

$$\mathbf{D}_{BA}(1) = \mathbf{POZ}_B^{-1} \cdot \mathbf{POZ}_A \tag{2.99}$$

Din fiecare din cele două matrice conducere $\mathbf{D}_{BC}$ şi $\mathbf{D}_{BA}$ se obţin valorile variabilelor x, y, z, θ, ϕ şi ψ. Pentru a deosebi valorile variabilelor (numite variabile conducere) se vor aplica indicii A, respectiv C. Deoarece ambele mulţimi ale variabilelor conducere sunt exprimate în raport cu acelaşi sistem de coordonate $\mathbf{POZ}_B$, se poate descrie mişcarea din punctul A, prin punctul B spre punctul C, dacă se modifică rx, ry, rz, rθ şi rϕ în expresia matricei conducere $\mathbf{D}(r)$ de la x_A, y_A, z_A, θ_A şi ϕ_A, la x_C, y_C, z_C, θ_C şi ϕ_C. Se presupune pentru început că prima rotaţie de unghi θ se efectuează în jurul

BUPT

aceleiaşi axe pentru ambele matrice conducere, sau, cu alte cuvinte, unghiul ψ este acelaşi pentru cele două matrice conducere.

Se pot defini acum $\Delta \mathbf{X}_B$ şi $\Delta \mathbf{X}_C$ folosind ecuaţiile (2.88). Relaţiile (2.87), (2.89) şi (2.90) se pot exprima sub forma rx, ry, rz, rθ şi rϕ. ψ fiind considerat acelaşi pentru ambele mişcări, nu constituie o variabilă a mişcării. Spre exemplu, coordonata rx a matricei conducere este dată de:

$$rx = \left[h^2(2-h)\left(\Delta \mathbf{X}_C \frac{\tau_1}{T_1} - \Delta \mathbf{X}_B \right) + 2\Delta \mathbf{X}_B \right] \cdot h + \mathbf{X}_A \tag{2.100}$$

$$r\dot{x} = \frac{1}{\tau_1}\left[2h^2(1{,}5-h)\left(\Delta \mathbf{X}_C \frac{\tau_1}{T_1} - \Delta \mathbf{X}_B \right) + \Delta \mathbf{X}_B \right] \tag{2.101}$$

$$r\ddot{x} = \frac{3h}{\tau_1^2}(1-h)\left(\Delta \mathbf{X}_C \frac{\tau_1}{T_1} - \Delta \mathbf{X}_B \right) \tag{2.102}$$

După perioada de tranziţie în momentul $t = \tau_1$ punctele intermediare alese sunt date de:

$$r \cdot x = \Delta \mathbf{X}_C \cdot h + \mathbf{X}_B \tag{2.103}$$

$$r \cdot \dot{x} = \frac{\Delta \mathbf{X}_C}{T_1} \tag{2.104}$$

$$r \cdot \ddot{x} = 0 \tag{2.105}$$

$$h = \frac{t}{T_1} \tag{2.106}$$

Dacă ψ nu este acelaşi pentru ambele transformări, atunci trebuie să se varieze ψ în timpul tranziţiei de la ψ_A la ψ_C (relaţia 2.93). Aceasta atrage după sine o nouă acceleraţie unghiulară. Trebuie menţionat că valoarea lui ψ se determină pentru cazul când θ este pozitiv. Pentru a determina ψ_{AB} se adună sau se scad 180^0 astfel încât să rezulte $|\psi_C - \psi_A| < 180^0$ şi se modifică semnul lui ϕ. ϕ obţinut din ecuaţia (2.86) nu este afectat de aceste modificări. Această modificare efectuată asupra lui ψ_{AB} va minimiza efectele acceleraţiei adiţionale corespunzătoare rotaţiei cu unghiul ψ.

Dacă punctul B se alege ca origine a sistemului de referinţă (fig.2.9), ΔX_C este definit direct de către: x_C, y_C, z_C, θ_C, ϕ_C, iar ΔX_B este definit de către: x_A, y_A, z_A, θ_A, ϕ_A. Expresiile elementelor matricei conducere, în acest caz, se modifică corespunzător.

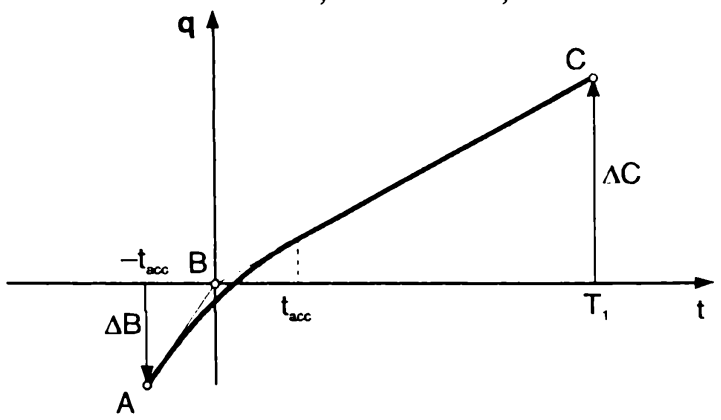


Fig.2.9. Tranziţia între segmentele AB şi BC ale unei traiectorii.
Originea sistemului de referinţă este aleasă în punctul B

BUPT

2.6. MIŞCAREA PUNCT CU PUNCT. FORMULAREA RESTRICŢIILOR DIN SPAŢIUL OPERAŢIONAL

În cazul mişcării *punct* cu *punct*, sarcina robotului este definită prin:

- o situare iniţială $\mathbf{X}_0$ şi o situare finală $\mathbf{X}_m$, cu:

$$\dot{\mathbf{X}}_o = \dot{\mathbf{X}}_m = 0 \tag{2.107}$$

- (m - 1) *puncte* de trecere intermediare $\mathbf{X}_k$ (k = 1, 2, ..., m - 1) stabilite astfel încât să fie evitate obstacolele din spaţiul operaţional.

Problema constă în a parcurge rapid intervalul dintre $\mathbf{X}_0$ şi $\mathbf{X}_m$, trecând prin vecinătatea *punctelor* intermediare şi respectând restricţiile exprimate prin viteze şi acceleraţii maxime liniare K_v şi K_a şi unghiulare K_ω şi K_ε:

$$\begin{cases} 0 \le |\mathbf{v}(t)| \le K_v \\ 0 \le |\mathbf{a}(t)| \le K_a \end{cases} \qquad \begin{cases} 0 \le |\omega(t)| \le K_\omega \\ 0 \le |\varepsilon(t)| \le K_\varepsilon \end{cases} \tag{2.108}$$

Soluţia problemei depinde de tipul de interpolare ales. Interpolările polinomiale sau interpolările polinomiale pe intervale asigură o bună continuitate a traiectoriilor. Curbele spline cubice, de exemplu, minimizează raza de curbură medie, deci o anumită funcţie a acceleraţiilor, dar sinteza mişcării optime nu este sistematică şi poate releva simulări interactive. O altă abordare constă în a urma "cel mai bine" poligonul $\mathbf{X}_0\mathbf{X}_1$ ... $\mathbf{X}_m$, minimizând timpul de parcurgere; traiectoriile în linie dreaptă facilitează de fapt programarea şi evitarea obstacolelor.

În spaţiul operaţional divizat în două subspaţii (poziţie şi orientare), traiectoriile sunt constituite din segmente de dreapă $[X_k\ X_{k-1}]$ parcurse cu viteză constantă v_k şi ω_k, legate prin arce de curbă parcurse cu acceleraţiile maxime $\pm K_a$ sau $\pm K_\varepsilon$. Arcele sunt racordate cu segmentele astfel încât să fie asigurată continuitatea vitezei.

În continuare se prezintă două moduri de abordare a generării mişcării punct cu punct în spaţiul operaţional, după cum este luată în considerare sau nu orientarea efectorului final.

2.6.1. MIŞCAREA PUNCTULUI CARACTERISTIC AL UNUI ROBOT, ÎN TIMP MINIM, CU ABATERE CONTROLATĂ FAŢĂ DE *PUNCTELE* IMPUSE ÎN COORDONATE CARTEZIENE, CU PĂSTRAREA CONSTANTĂ A ORIENTĂRII EFECTORULUI FINAL

Mişcarea efectorului final al robotului este formată din *m* segmente de dreaptă, în coordonate carteziene, conectate prin arce de cerc. Viteza pe segmentele de dreaptă este constantă, iar vitezele iniţiale şi finale sunt nule. Eroarea de poziţie $\mathbf{e}_k$, care apare la tranziţia între două segmente, trebuie controlată pentru evitarea coliziunii cu obstacolele, iar limitele admisibile de viteză şi acceleraţie K_v şi K_a ale sistemului de acţionare nu trebuiesc depăşite. În aceste condiţii, timpul total, necesar pentru parcurgerea traiectoriei, între *punctele* $\mathbf{X}_0$ şi $\mathbf{X}_m$ trebuie să fie minim [D7], [D43], [D44], [L15]:

$$T = \tau_0 + \sum_{k=1}^{m} T_k + \tau_m \tag{2.109}$$

BUPT

2.6.1.1. TRAIECTORIA ŞI FORMA LEGII DE COMANDĂ

Sarcina robotului este definită prin *m+1 puncte* (poligon de definiţie) în $\mathbf{R}^3$: ($\mathbf{P}_0$, $\mathbf{P}_1$, ... $\mathbf{P}_m$). Mişcarea punctului caracteristic al robotului pe traiectorie are două aspecte diferite, corespunzător fazelor de pornire/oprire şi de regim staţionar, pentru restul traiectoriei.

Pornirea şi oprirea: În planul (P_0, P_1, P_2), segmentul [P_0, P_1] este parcurs cu acceleraţie constantă pe durata $2\tau_0$, apoi cu viteză constantă pe o durată ($T_1 - \tau_1 - \tau_0$), unde T_1 şi τ_1 se vor defini pentru cazul general. De asemenea, în planul (P_{m-2}, P_{m-1}, P_m), segmentul [P_{m-1}, P_m] se termină cu o decelerare constantă de durată $2\tau_m$.

Perioada staţionară: În cazul general, în planul (P_{k-1}, P_k, P_{k+1}), segmentul [P_{k-1}, P_k] este parcurs cu viteză constantă, mai puţin în timpul tranziţiilor, în vecinătatea lui P_{k-1} sau P_k, unde traiectoria este un arc de parabolă parcurs cu acceleraţie constantă. Se notează $2\tau_k$ durata tranziţiei numărul *k* şi cu T_k durata de parcurgere a porţiunii [$\underline{P}_{k-1}$, $\underline{P}_k$] unde $\underline{P}_k$ este punctul real atins (care diferă de P_k pentru k = 1, 2, ... , m - 1).

Figura 2.10 ilustrează forma traiectoriei obţinute şi precizează definiţiile diverşilor timpi de parcurgere. Originea timpului *t* se alege în P_0, la începutul primei accelerări.

Ecuaţiile de mişcare se exprimă considerând mişcarea efectorului final, cu păstrarea orientării constante de-a lungul întregii traiectorii şi constau în expresiile poziţiei, vitezei şi acceleraţiei punctului caracteristic pentru fiecare interval de timp T_k.

Pe porţiunile liniare, viteza constantă este viteza medie:

$$\mathbf{v}_k = \begin{cases} \dfrac{\mathbf{P}_k - \mathbf{P}_{k-1}}{T_k} & k = \overline{1,\ m} \\ 0 & k = 0,\ m+1 \end{cases} \tag{2.110}$$

Acceleraţiile nenule, în timpul tranziţiilor între segmentele traiectoriei, sunt:

$$\mathbf{a}_k = \begin{cases} \mathbf{a}_0 = \dfrac{\mathbf{v}_1}{2\tau_0} & k = 0 \\ \mathbf{a}_k = \dfrac{\mathbf{v}_{k+1} - \mathbf{v}_k}{2\tau_k} & k = \overline{1,\ m} \\ \mathbf{a}_m = -\dfrac{\mathbf{v}_m}{2\tau_m} & k = m \end{cases} \tag{2.111}$$

Ecuaţiile de mişcare se deduc din (2.110) şi (2.111) prin integrare.

Ecuaţiile de mişcare pentru faza de pornire: Se consideră $t_1 = t$, $0 \le t_1 \le \tau_0 + T_1 - \tau_1$. Deoarece $\mathbf{v}_0 = 0$, pentru punctul curent rezultă:

$$\begin{cases} \mathbf{P'}_0(t_1) = \mathbf{P}_0 + \dfrac{1}{2}\mathbf{a}_0 t_1^2 & 0 \le t_1 \le 2\tau_0 \\ \mathbf{P''}_0(t_1) = \mathbf{P}_0 + 2\mathbf{a}_0\tau_0^2 + \mathbf{v}_1(t_1 - 2\tau_0) & 2\tau_0 \le t_1 \le \tau_0 + T_1 - \tau_1 \end{cases} \tag{2.112}$$

Ecuaţiile de mişcare pentru faza de sosire. Se consideră: $t_{m+1} = t - \sum_{k=1}^{m} T_k - \tau_0 + \tau_m$.

Deoarece $\mathbf{v}_{m+1} = \mathbf{0}$, rezultă:

BUPT

$$\mathbf{P'}_m(t_{m+1}) = \left(\mathbf{P}_m - 2\mathbf{v}_m\tau_m - 2\mathbf{a}_m\tau_m^2\right) + \mathbf{v}_m t_{m+1} + \frac{1}{2}\mathbf{a}_m t_{m+1}^2 \qquad 0 \le t_{m+1} \le 2\tau_m \quad (2.113)$$

Ecuaţiile de mişcare pentru faza staţionară: Se consideră $t_k = t - \tau_0 - \sum_{j=1}^{k} T_j$ ceeea ce revine la a alege originea timpului t_k în momentul trcerii prin $\mathbf{P}_{k-1}$. Rezultă astfel, fie o fază de tranziţie cu acceleraţia constantă $\mathbf{a}_{k-1}$, fie cu viteză constantă $\mathbf{v}_k$:

$$\begin{cases} \mathbf{P'}_{k-1}(t_k) = \left(\mathbf{P}_{k-1} - \mathbf{v}_{k-1}\tau_{k-1}\right) + \mathbf{v}_{k-1}(t_k + \tau_{k-1}) + \frac{1}{2}\mathbf{a}_{k-1}(t_k + \tau_{k-1})^2 & -\tau_{k-1} \le t_k \le \tau_{k-1} \\ \mathbf{P''}_{k-1}(t_k) = \mathbf{P'}_{k-1}(\tau_{k-1}) + \mathbf{v}_k(t_k + \tau_{k-1}) & \tau_{k-1} \le t_k \le T_k \end{cases} \quad (2.114)$$

2.6.1.2. CRITERIUL DE OPTIMIZARE ŞI RESTRICŢIILE CINEMATICE

Optimizarea generării mişcării în condiţiile precizate mai sus constă în minimizarea timpului total de parcurgere:

$$\begin{cases} T_{opt} = \min T \\ T = \tau_0 + \tau_m + \sum_{k=1}^{m} T_k \end{cases} \quad (2.115)$$

unde variabilele sunt în număr de *m+2*: τ_0, τ_m şi T_k.

Problema determinării unei mişcări în timp se reduce la găsirea a două secvenţe de intervale de timp $\{t_k\}$ şi $\{T_k\}$, pentru toate segmentele (k = 0, 1, ... , m) care minimizează timpul total T şi fac ca mişcările, definite prin relaţiile (2.112), (2.113) şi (2.114) să respecte limitările de viteză, acceleraţie şi eroare de poziţie faţă de punctele de intersecţie a segmentelor. Sistemul general al restricţiilor este:

$$|\mathbf{v}_k| \le K_v \quad (2.116)$$

$$|\mathbf{a}_k| \le K_a \quad (2.117)$$

$$|\mathbf{e}_k| \le kK_k \quad (2.118)$$

$$T_k \ge \tau_k + \tau_{k-1} \quad (2.119)$$

unde *k* este un coeficient pozitiv,

$$K_k = \min.\left[|\mathbf{P}_{k+1} - \mathbf{P}_k|, |\mathbf{P}_k - \mathbf{P}_{k-1}|\right] \quad (2.120)$$

iar $\mathbf{e}_k = \mathbf{P}_k - \underline{\mathbf{P}}_k$ este abaterea (eroarea) la trecerea prin punctul $\underline{P}_k$.

Valorile limită K_v, K_a şi coeficientul *k* se pot determina experimental. Se observă că semnificaţia practică a coeficientului *k* nu este evidentă. Este preferabil să se poată fixa direct toleranţele de precizie, în punctele de trecere, de forma $|\mathbf{e}_k|_{max}$. Ultima restricţie exprimă faptul că tranziţiile parabolice succesive trebuie să fie racordate prin segmente de dreaptă, de lungime nulă la limită.

Pentru a rezolva problema optimizării, se transformă sistemul de restricţii, impunând acceleraţiei o evoluţie de tipul "totul sau nimic", parcurgerea curbelor parabolice efectuându-se cu acceleraţii maxime $|\mathbf{a}_k| = K_a$ şi viteze $\mathbf{v}_k$ [L15]. Restricţiile (2.116)-(2.119) se transformă în:

BUPT

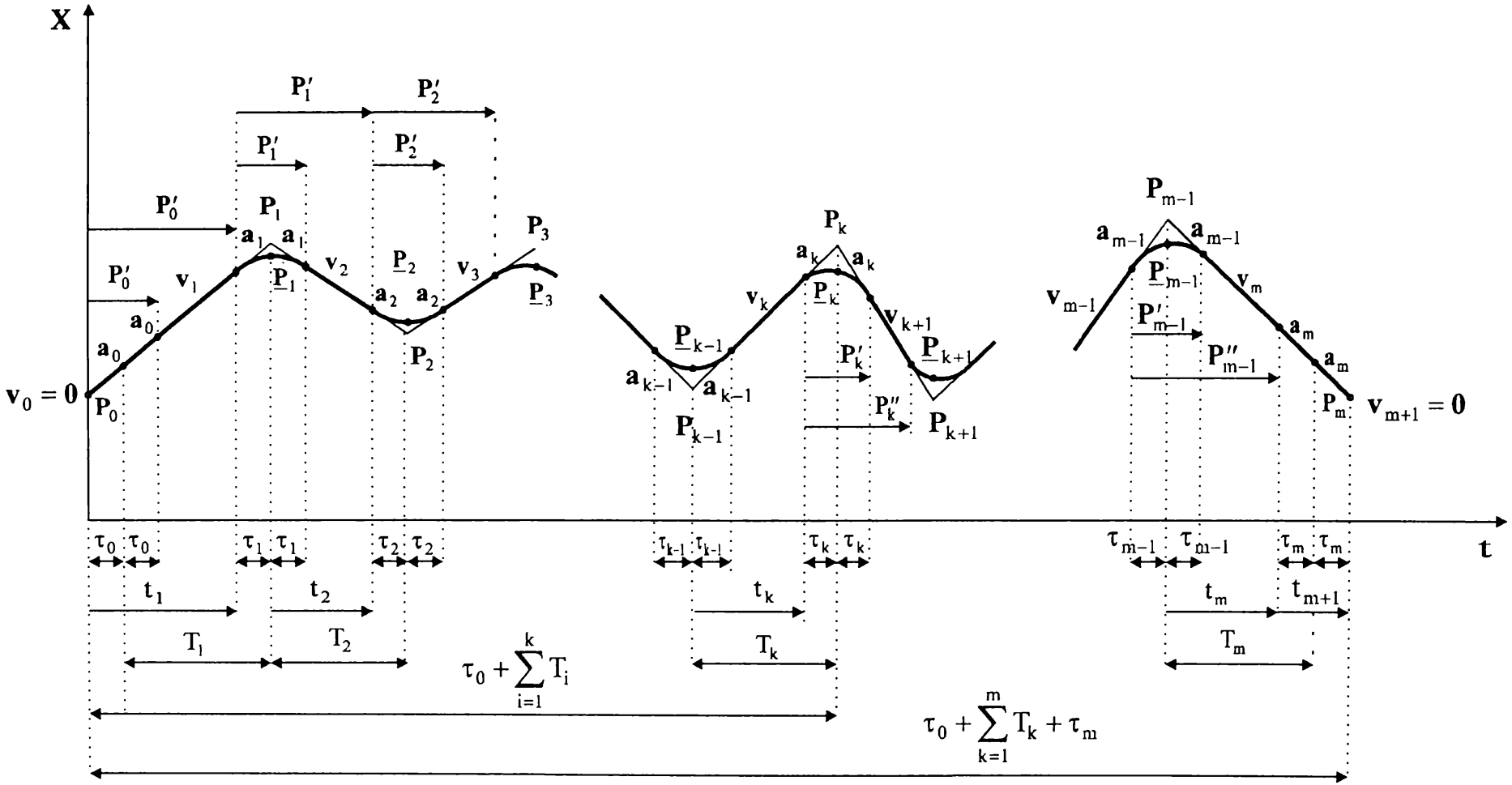


Fig. 2.10. Forma traiectoriei în $\mathbf{R}^3$

BUPT

$$T_k \geq \frac{|\mathbf{P}_k - \mathbf{P}_{k-1}|}{K_v}, \quad k = \overline{1, m} \tag{2.121}$$

$$0 \leq \tau_0 \leq \frac{K_v}{2K_a} \tag{2.122}$$

$$0 \leq \tau_m \leq \frac{K_v}{2K_a} \tag{2.123}$$

$$0 \leq \tau_k \leq \sqrt{\frac{2kK_k}{K_a}} \tag{2.124}$$

$$T_k \geq \tau_k + \tau_{k-1} \tag{2.125}$$

$$\tau_k = \frac{\left| \frac{\mathbf{P}_{k+1} - \mathbf{P}_k}{T_{k+1}} - \frac{\mathbf{P}_k - \mathbf{P}_{k-1}}{T_k} \right|}{2K_a} \tag{2.126}$$

Primele cinci restricții sunt de tip inegalitate, liniare în variabilele T_k şi τ_k. Ultima (2.126) este neliniară, de tip egalitate, provenind din expresia:

$$\mathbf{v}_{k+1} = 2\,\mathbf{a}_k \tau_k + \mathbf{v}_k \tag{2.127}$$

de unde, ținând cont că $|\mathbf{a}_k| = K_a$, rezultă:

$$|\mathbf{v}_{k+1} - \mathbf{v}_k| = 2K_a \tau_k \tag{2.128}$$

Dacă această restricție nu ar exista, sistemul restricțiilor s-ar reduce la rezolvarea unei probleme de programare liniară. Pentru a nu utiliza metode complicate de programare neliniară, Paul sugerează să se neglijeze pentru început restricția (2.126), să se rezolve problema liniară rezultată, să se adapteze apoi timpii τ_k conform (2.126) şi să se continue astfel până la satisfacerea ultimului tip de restricție. O primă metodă de rezolvare a problemei a fost propusă de B. Tonder [L15]. Algoritmul acesteia are următoarele etape:

1. *Inițializare*: se dă un set de valori τ_k^A, care satisfac relațiile (2.122), (2.123) şi (2.124);
2.*Rezolvarea problemei liniare*, definită prin criteriul (2.115) şi restricțiile (2.121) şi (2.125), prin metoda Simplex, timpii T_k fiind variabilele;
3.*Calculul unui set de valori* τ_k^B, cu relația (2.126);
4.*Compararea mulțimilor* τ_k^B *şi* τ_k^A: Dacă τ_k^B este apropiat de τ_k^A se egalează $\tau_k = \tau_k^B$ şi algoritmul este terminat; dacă nu, $\tau_k = (\tau_k^A + \tau_k^B)/2$ şi se reia algoritmul de la pasul 2.

Actualizarea lui τ_k după pasul 4 are în vedere apropierea soluției de restricția de egalitate şi algoritmul converge după câteva iterații.

O metodă pentru rezolvarea sistemului de restricții constă în înlocuirea restricției (2.124) prin următoarele două:

$$\begin{cases} T_k \geq 2\,\text{Max.}[\tau_{k-1}, \tau_k] \\ \tau_k \leq \frac{1}{2}\,\text{Min.}[T_k, T_{k+1}] \end{cases} \tag{2.129}$$

BUPT

care au ca scop să împiedice ca tranzițiile parabolice să nu se racordeze cu segmentele [P_k P_{k+1}] ale poligonului după mijlocul acestor segmente. Considerând $\mathbf{d_k} = \mathbf{P_k} - \mathbf{P_{k-1}}$ pentru k = 1, 2, ... , m, se obține soluția analitică a problemei de optimizare prin relațiile:

$$\begin{cases} T_k = \text{Max}\left(\dfrac{|\mathbf{d_k}|}{K_v}, \sqrt{\dfrac{\mathbf{d_{k-1}} + \mathbf{d_k}}{K_a}}, \sqrt{\dfrac{\mathbf{d_k} + \mathbf{d_{k+1}}}{K_a}}\right). \\ \tau_k = \dfrac{|\mathbf{v_{k+1}} - \mathbf{v_k}|}{2K_a} \end{cases} \tag{2.130}$$

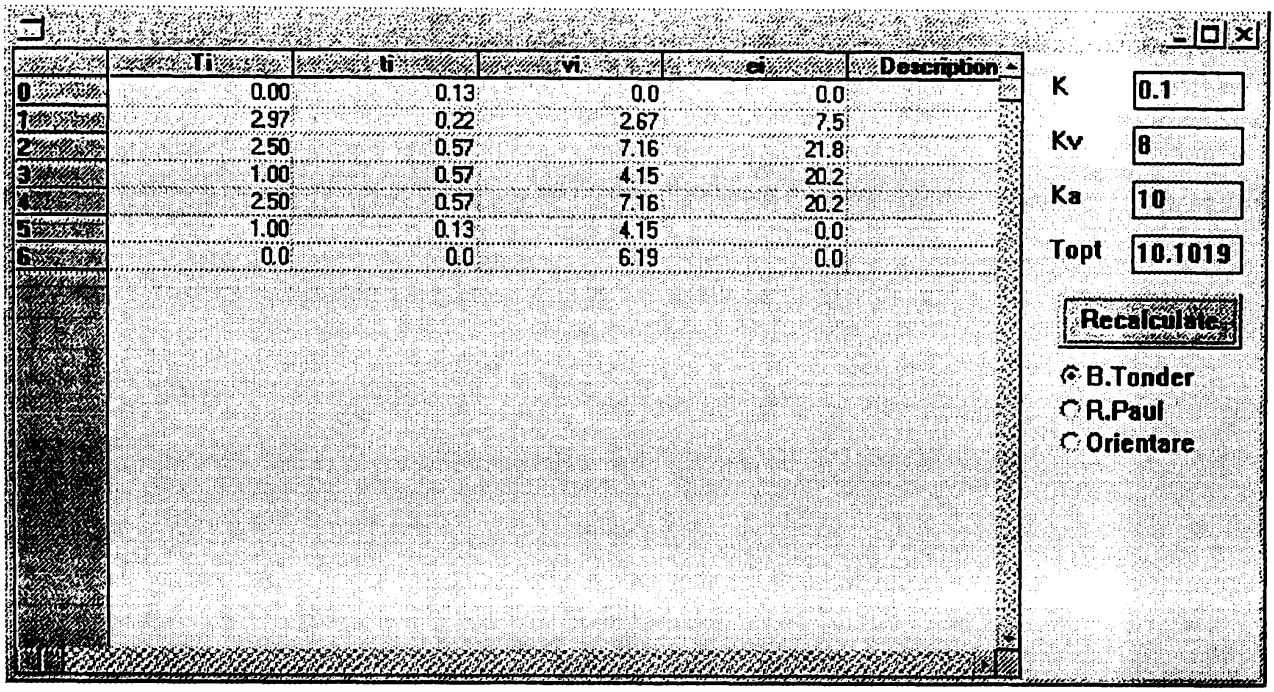
Pe baza metodei prezentate s-a elaborat un program de simulare a mişcării punctului caracteristic al unui robot, în timp minim, cu abatere controlată față de *punctele* de precizie impuse, fără să se ia în considerare orientarea efectorului final. Curbele din fig.2.12 şi fig.2.13 corespund unui exemplu de simulare a mişcării pe baza soluțiilor obținute cu cele două metode, în cazul unei traiectorii formate din şase puncte (fig.2.11).

	Poz.X	Poz.Y	Poz.Z	Description
0	20	-20	10	
1	30	0	2	
2	30	20	2	
3	30	20	10	
4	10	20	10	
5	10	20	2	

Fig.2.11. Interpolare liniară şi tranziție parabolică.
Punctele de definire a traiectoriei

Programul elaborat a fost utilizat pentru un studiu comparativ al celor două metode. Rezultatele obținute din rularea programului demonstrează că atâta timp cât K_v este mic, restricția asupra vitezei este deja saturată, deci importanța zonelor de tranziție cu accelerație constantă este relativ scăzută, şi datorită acestui fapt, a două metodă, care este foarte uşor de implementat, dă rezultate nu prea îndepărtate de cele ale primei metode. Diferența este mult mai sensibilă pentru valori ridicate ale lui K_v; în acest caz, prima metodă utilizează la maximum perioade cu accelerație saturată, pe când a doua, nu permite să se ajungă după mijlocul segmentelor poligonului de definiție [P_k P_{k+1}]. Timpul minim obținut prin metoda propusă de Tonder este mai mic, dar precizia este mai bună prin metoda propusă de Paul.

BUPT



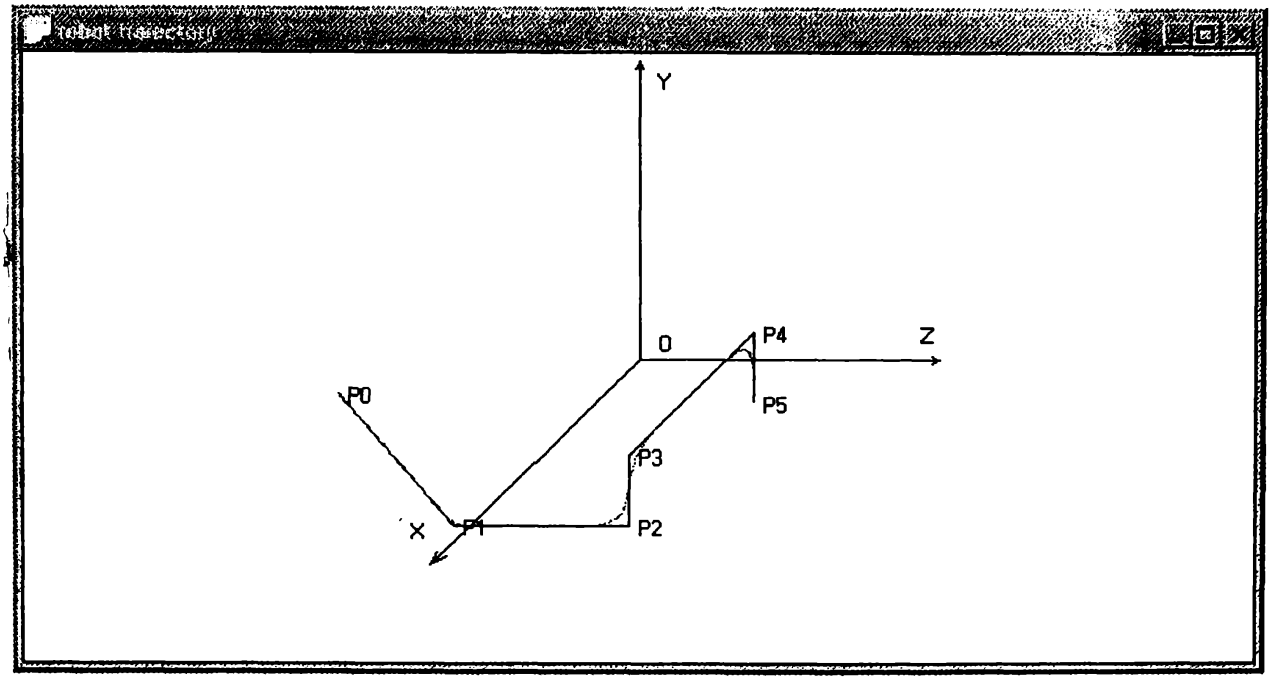
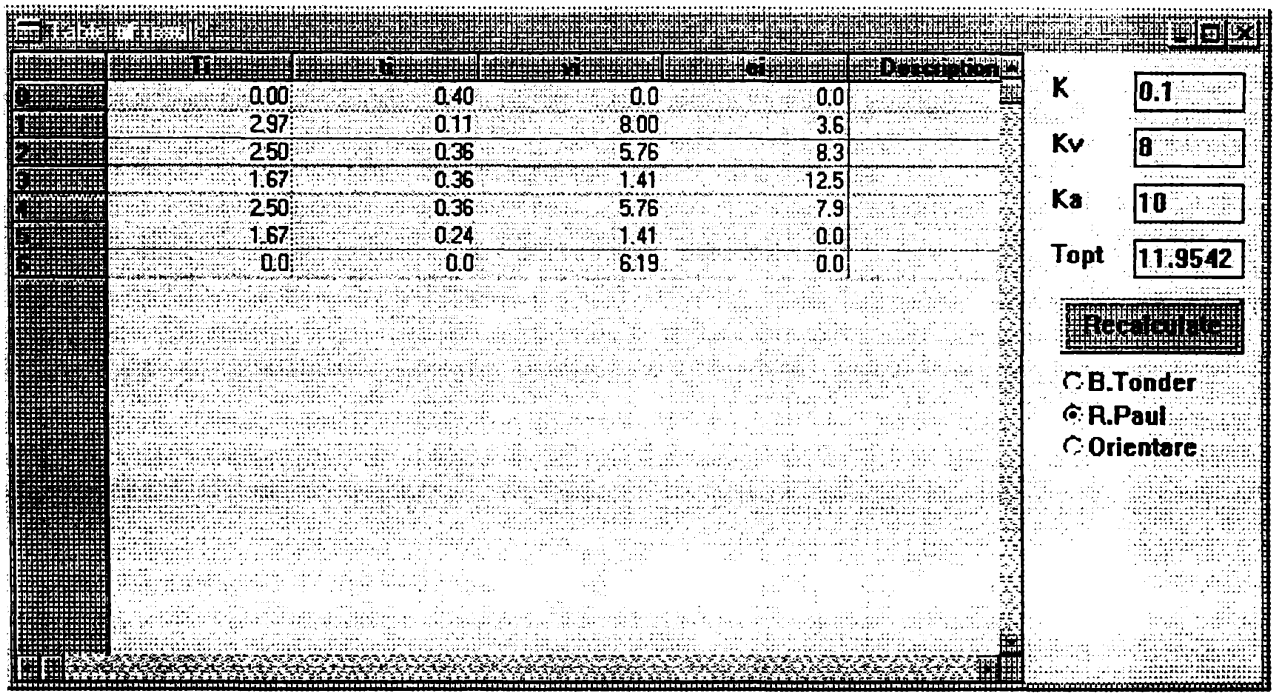


Fig.2.12. Interpolare liniară şi tranziţie parabolică. Metoda B.Tonder

BUPT



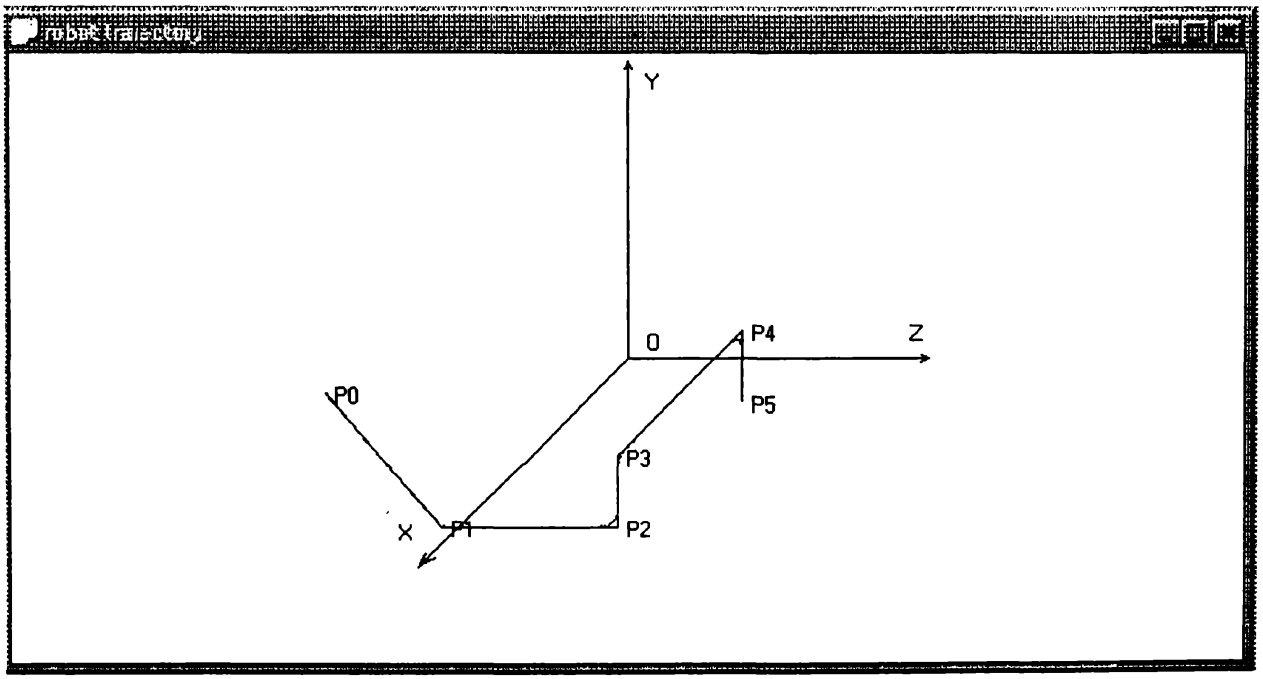


Fig.2.13. Interpolare liniară şi tranziţie parabolică. Metoda R.Paul

BUPT

2.6.2. MIŞCAREA PUNCTULUI CARACTERISTIC AL UNUI ROBOT, ÎN TIMP MINIM, CU ABATERE NULĂ FAŢĂ DE *PUNCTELE* IMPUSE ÎN COORDONATE CARTEZIENE, CU LUAREA ÎN CONSIDERARE A ORIENTĂRII EFECTORULUI FINAL

Metodele prezentate în paragraful precedent nu iau în considerare orientarea efectorului final. Acestea se pot aplica totuşi, atunci când dimensiunile efectorului final sunt suficient de mici pentru a nu introduce, oricare ar fi legea de comandă punct cu punct în spaţiu, riscuri de coliziune şi când vitezele unghiulare ale efectorului final sunt destul de mari pentru a nu influenţa mişcarea optimală a punctului caracteristic.

Faţă de mişcarea punctului caracteristic în timp minim prezentată anterior, se impun două condiţii suplimentare. Efectorul final al robotului îşi modifică şi orientarea de-a lungul traiectoriei, iar trecerea prin toate *punctele* P_k (k = 0, ... , m) se face fără abatere [L25], [L28]. Considerarea în plus a mişcării de rotaţie, implică apariţia unor noi relaţii de mişcare şi limitări pentru viteza unghiulară ω şi acceleraţia unghiulară ε ale efectorului final ale robotului, iar condiţia de atingere a *punctelor* P_k duce la dispariţia restricţiilor de abatere de la traiectoria formată din cele *m* segmente de dreaptă ($|\mathbf{e}_k| = 0$). Pentru o traiectorie formată din *m* segmente de dreaptă, există *m+1 puncte* de definire a traiectoriei. Pentru descrierea structurată a orientării efectorului final, în fiecare *punct* P_k al traiectoriei, se asociază matricea de transformare omogenă $\mathbf{G}_n(k)$. Un segment *k* de traiectorie este divizat în trei intervale:

- tranziţia de la segmentul *k-1* cu $\mathbf{a}_{k-1}$, ε_{k-1} constante ($\mathbf{a}_{k-1}\neq 0$, $\varepsilon_{k-1}\neq 0$), în intervalul τ_{k-1};
- mişcarea cu $\mathbf{v}_k$, ω_k constante de-a lungul segmentului *k*;
- tranziţia la segmentul *k+1* cu $\mathbf{a}_k$, ε_k constante ($\mathbf{a}_k\neq 0$, $\varepsilon_k\neq 0$), în intervalul τ_k.

Timpul necesar trecerii de la configuraţia $\mathbf{G}_n(k-1)$ la configuraţia $\mathbf{G}_n(k)$ este T_k. Timpul total de execuţie este $T = \tau_0 + \sum_{k=1}^{m} T_k + \tau_m$. Când este cazul, trebuie să se specifice, între două *puncte* de oprire, în care valorile unghiulare sunt φ_0 şi φ_m, un număr de situaţii intermediare de tip ($\mathbf{P}_i$, φ_i) astfel încât restricţiile asupra mărimilor vitezei unghiulare ω şi acceleraţie unghiulare ε sunt:

$$|\omega| \le K_\omega; \qquad |\varepsilon| \le K_\varepsilon \tag{2.131}$$

Dacă se alege originea timpului *t* ca în fig.2.14 şi se folosesc relaţiile prezentate în paragrafele anterioare, se exprimă legile de mişcare a efectorului final al robotului (translaţie şi rotaţie), pe segmente de traiectorie şi la tranziţiile între segmente.

Mişcarea de translaţie

$$\mathbf{v}_k = \frac{\mathbf{P}_k - \mathbf{P}_{k-1}}{T_k}; \qquad \tau_0 + \sum_{j=1}^{k-1} T_j + \tau_{k-1} \le t \le \tau_0 + \sum_{j=1}^{k} T_j - \tau_k$$

$$\mathbf{a}_k = \frac{\mathbf{v}_{k+1} - \mathbf{v}_k}{2\tau_k}; \qquad \tau_0 + \sum_{j=1}^{k} T_j - \tau_k \le t \le \tau_0 + \sum_{j=1}^{k} T_j + \tau_k \tag{2.132}$$

BUPT

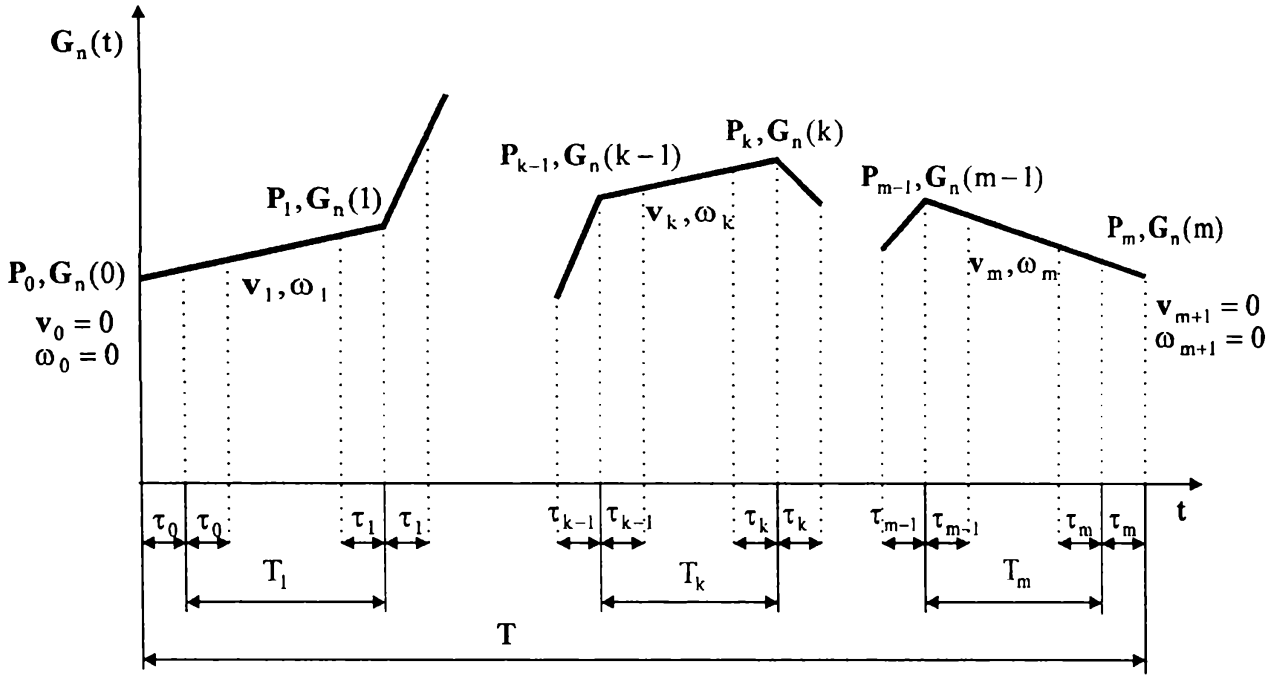


Fig. 2.14. Transformările de coordonate omogene asociate intersecțiilor segmentelor liniare ale traiectoriei robotului

Mişcarea de rotaţie

Dacă orientarea în *punctul* de trecere este dată de matricea cosinuşilor directori:

$$\mathbf{R}_k = \mathbf{R}_o(k) = \left[\mathbf{n}_k \mid \mathbf{o}_k \mid \mathbf{a}_k\right] \tag{2.133}$$

$\Delta\varphi_k = \mathbf{u}_k\left(\varphi_k - \varphi_{k-1}\right)$ este vectorul de rotaţie de trecere între φ_{k-1} şi φ_k cu:

$$\theta_k = \varphi_k - \varphi_{k-1} = \text{Arccos}\left(\mathbf{n}_k \cdot \mathbf{n}_{k-1} - \mathbf{o}_k \cdot \mathbf{o}_{k-1} + \mathbf{a}_k \cdot \mathbf{a}_{k-1}\right) \tag{2.134}$$

şi dacă $\theta_k \neq 0$, atunci:

$$\mathbf{u}_k = \frac{1}{2\sin\theta_k}\begin{bmatrix} \mathbf{a}_k \cdot \mathbf{o}_{k-1} & -\mathbf{o}_k \cdot \mathbf{a}_{k-1} \\ \mathbf{n}_k \cdot \mathbf{a}_{k-1} & -\mathbf{a}_k \cdot \mathbf{n}_{k-1} \\ \mathbf{o}_k \cdot \mathbf{n}_{k-1} & -\mathbf{n}_k \cdot \mathbf{o}_{k-1} \end{bmatrix} \tag{2.135}$$

În relaţiile (2.134) şi (2.135), $\mathbf{u}_k$ şi θ_k descriu rotaţia efectorului final între transformările omogene $\mathbf{G}_n(k\text{-}1)$ şi $\mathbf{G}_n(k)$, iar $\mathbf{n}_k$, $\mathbf{o}_k$ şi $\mathbf{a}_k$ sunt versorii sistemului de referinţă ataşat efectorului final. În spaţiul operaţional, Luh şi Lin [L26] adoptă o evoluţie a acceleraţiei unghiulare de tipul "totul sau nimic", modulul ei trecând de la zero la K_ε într-un mod analog cu acceleraţia $\ddot{\mathbf{P}}$ a punctului caracteristic.

Pe segmentele de dreaptă, viteza unghiulară este constantă:

$$\omega_k = \frac{\theta_k}{T_k}\mathbf{u}_k \tag{2.136}$$

Tranziţia de la valoarea ω_k la ω_{k+1} se efectuează cu acceleraţie constantă:

$$\varepsilon_k = \frac{\omega_{k+1} - \omega_k}{2\tau_k} \tag{2.137}$$

BUPT

Principiul metodei de generare a traiectoriei este analog celui expus anterior (capitolul 2.6.1). Criteriul de optimizare constă, de asemenea, în minimizarea timpului total de parcurgere:

$$T_{opt} = \min_{\tau_0, \tau_m, T_k} \left(\tau_0 + \sum_{k=1}^{m} T_k + \tau_m \right) \tag{2.138}$$

În primul rând, un mod de trecere prin vecinătatea *punctelo*r intermediare constă în a limita raportul dintre T_k şi durata tranzițiilor ce precedă şi urmează fazei de deplasare cu viteză constantă pe al *k*-lea segment de dreaptă ($\alpha > 0$):

$$\begin{cases} 0 \le \tau_k \le \dfrac{T_k}{\alpha} & k = 1, 2, \ldots, m \\ 0 \le \tau_k \le \dfrac{T_{k+1}}{\alpha} & k = 0, 1, \ldots, m-1 \end{cases} \tag{2.139}$$

Restricțiile impuse de cuplele cinematice sunt convertite în $\mathbf{v}_k$, ω_k, respectiv $\mathbf{a}_k$, ε_k. Restricțiile cinematice $|\mathbf{v}| \le K_v$, $|\mathbf{a}| \le K_a$, $|\omega| \le K_\omega$, $|\varepsilon| \le K_\varepsilon$ conduc la expresii de forma:

$$|\mathbf{v}| \le K_v \quad \Rightarrow \quad |\mathbf{P}_k - \mathbf{P}_{k-1}| \le K_v T_k \tag{2.140}$$

$$|\mathbf{a}| \le K_a \quad \Rightarrow \quad |T_k(\mathbf{P}_{k+1} - \mathbf{P}_k) - T_{k+1}(\mathbf{P}_k - \mathbf{P}_{k-1})| \le 2K_a T_k T_{k+1} \tau_k \tag{2.141}$$

$$|\omega| \le K_\omega \quad \Rightarrow \quad |\theta_k| \le K_\omega T_k \tag{2.142}$$

$$|\varepsilon| \le K_\varepsilon \quad \Rightarrow \quad \begin{cases} |\theta_1| \le 2K_\varepsilon T_1 \tau_0 \\ \sqrt{T_k^2 \theta_{k+1}^2 + T_{k+1}^2 \theta_k^2 - 2T_k T_{k+1} u_k^2} \le 2K_\varepsilon T_k T_{k+1} \tau_k \\ |\theta_m| \le 2K_\varepsilon T_m \tau_m \end{cases} \tag{2.143}$$

cu notația următoare: $u_k^2 = (\mathbf{u}_k \cdot \mathbf{u}_{k-1}) \theta_k \theta_{k-1}$.

Rezultă *6m* inecuații liniare şi *2(m+1)* inecuații neliniare, cu *2m+1* necunoscute T_k şi τ_k. Problema generării mişcării optime, definită prin criteriul (2.138) şi prin diversele restricții exprimate prin inegalități, este accentuat neliniară. Aceasta se poate rezolva prin aproximații succesive, liniarizând restricțiile în vecinătatea soluției curente, apoi rezolvând problema liniară astfel obținută şi efectuând în final o revizie asupra restricțiilor. Timpul de calcul al soluției optime este relativ lung, această soluție fiind foarte sensibilă la proprietatea de recalare, şi deci trebuie să se evite eventualele soluții ce minimizează local criteriul, dar nu furnizează minimul absolut. Aceste soluții pot apărea datorită neconvexității problemei.

Funcția obiectiv (2.138), care reprezintă timpul total de mişcare T, este liniară, dar restricțiile sunt mixte: (2.140), (2.142) - liniare, (2.141), (2.143) - neliniare. Din această cauză, pentru determinarea setului de valori $\{T_k, \tau_k\}$ trebuie să se aplice tehnici de programare neliniară [L4], cu funcția obiectiv (2.138) (spre exemplu, metoda programării aproximative).

Volumul mare de calcule necesar pentru generarea unei traiectorii în timp minim nu poate fi efectuat în faza de execuție a mişcării. Astfel de traiectorii se planifică într-o fază "off—line", de compilare a programului robot, memorându-se setul de valori optimale $\{T_k, \tau_k\}$. Mişcarea optimală absolută este furnizată ca mărime de referință sistemului de comandă, mişcarea reală a robotului urmărind mişcarea optimă planificată.

BUPT

Algoritmii prezentați pentru generarea mișcării nu au luat în considerare informația reală a structurii mecanice şi a sistemului de acționare. Din această cauză, traiectoria reală se abate totuşi de la traiectoria dorită.

2.7. INTERPOLAREA CVASILINIARĂ ÎN SPAȚIUL CONFIGURAȚIIOR, CU ABATERE CONTROLATĂ ÎN SPAȚIUL OPERAȚIONAL

Schema de control pentru o traiectorie carteziană necesită un timp de calcul considerabil, fiind dificil de tratat restricțiile comportării în timp real a manipulatorului, în spațiul configurațiilor (spațiul coordonatelor cuplelor cinematice). Pentru tratarea acestei probleme există, în literatura de specialitate, mai multe soluții. O metodă ar fi precalculul şi memorarea soluțiilor cuplelor prin simularea algoritmului în timp real, înainte de execuția mişcării. Astfel, execuția mişcării ar fi neînsemnată dacă mulțimea *punctelor* ar putea fi cu uşurință citită din memorie. O altă metodă constă în precalculul soluțiilor cuplelor pentru fiecare al *n*-lea interval selectat, urmat de realizarea interpolării CCC, folosind funcții polinomiale de grad mic pentru a "potrivi" aceste *puncte* intermediare în vederea generării mulțimii *punctelor* "servo".

Dificultatea acestei metode constă în faptul că numărul *punctelor* intermediare necesare pentru a menține efectorul final suficient de aproape de traiectoria rectilinie carteziană depinde de mişcarea particulară ce trebuie făcută. Orice interval predeterminat, suficient de mic pentru a garanta abateri mici, necesită o cantitate mare de timp pentru precalcul şi memorare. Din această cauză, Taylor [T3] a propus o stategie de mişcare în spațiul variabilelor cuplelor, numită "traiectoria cuplei cu abatere controlată", care selectează suficiente *puncte* intermediare în timpul fazei de pregenerare (specificare) pentru a garanta că abaterea efectorului final de la traiectoria rectilinie carteziană, pentru fiecare segment al mişcării, rămâne în interiorul unui interval specificat. Totodată, metoda propusă de Taylor permite să se diminueze numărul transformărilor de coordonate. Interpolarea se efectuează în spațiul configurațiilor, spațiul în care sunt impuse practic, restricțiile reale asupra vitezelor şi accelerațiilor. Datorită neliniarității transformării geometrice de coordonate $\mathbf{X} = f(\mathbf{q})$, segmentul de traiectorie de la $\mathbf{X}_k = f(\mathbf{q}_k)$ la $\mathbf{X}_{k+1} = f(\mathbf{q}_{k+1})$ nu va fi o linie dreaptă în spațiul operațional. Pentru a se apropia de linia dreaptă (în scopul evitării obstacolelor), evitând totuşi în acelaşi timp efectuarea unui număr prea mare de transformări inverse de coordonate care se repercutează negativ asupra duratei calculelor şi rapidității mişcărilor, Taylor propune un algoritm care permite să se genereze un număr de *puncte* intermediare suplimentare între $\mathbf{q}_k$ şi $\mathbf{q}_{k+1}$. Transformarea inversă de coordonate (neliniară, obținută analitic sau numeric) fiind presupusă cunoscută, se pot calcula $\mathbf{q}_k = f^{-1}(\mathbf{X}_k)$ şi $\mathbf{q}_{k+1} = f^{-1}(\mathbf{X}_{k+1})$. Se admite că valoarea maximă a deviației între $f(\mathbf{q}(t))$ şi dreapta:

$$\mathbf{X} = \left(1 - \frac{t}{T_k}\right) f(\mathbf{q}_k) + \frac{1}{T_k} f(\mathbf{q}_{k+1}) \qquad (2.144)$$

pentru $t \in [0, T_k]$, are loc pentru $\frac{\mathbf{q}_{k+1} + \mathbf{q}_k}{2}$.

Impunându-se abaterile maxime de translație şi de rotație pentru poziția şi orientarea sistemului de referință ataşat efectorului final în *punctele* nodale carteziene, algoritmul generează suficiente *puncte* nodale suplimentare astfel încât să nu fie depăşite

BUPT

aceste erori maxime impuse. Algoritmul propus, bazat pe ideea lui Taylor, utilizează o modificare a metodei înjumătățirii intervalului. Astfel etapele de calcul ale algoritmului sunt:

Pasul 1: Se citesc *punctele* nodale ale traiectoriei carteziene $\mathbf{X}_k$ şi $\mathbf{X}_{k+1}$;

Pasul 2: Se calculează vectorii coordonatelor cuplelor $\mathbf{q}_k$ şi $\mathbf{q}_{k+1}$ corespunzători *punctelor* nodale carteziene $\mathbf{X}_k$ şi $\mathbf{X}_{k+1}$, folosind **MGI**;

Pasul 3: Se calculează *punctul* de mijloc în spațiul configurațiilor $\mathbf{q}_m = \frac{\mathbf{q}_k + \mathbf{q}_{k+1}}{2}$;

Pasul 4: Se calculează *punctul* nodal cartezian corespunzător configurației $\mathbf{q}_m$, folosind **MGD**: $\mathbf{X}_m = f\left(\frac{\mathbf{q}_k + \mathbf{q}_{k+1}}{2}\right)$;

Pasul 5: Se calculează *punctul* de mijloc în spațiul cartezian $\underline{\mathbf{X}}_m = \frac{\mathbf{X}_k + \mathbf{X}_{k+1}}{2}$;

Pasul 6: Se calculează abaterea între $\underline{\mathbf{X}}_m$ şi $\mathbf{X}_m$;

Pasul 7: Se verifică încadrarea în limita admisă a erorii.

Dacă $|\underline{\mathbf{X}}_m - \mathbf{X}_m| \le e$, atunci STOP

Dacă $|\underline{\mathbf{X}}_m - \mathbf{X}_m| > e$, atunci se reia algoritmul pentru cele două subsegmente de traiectorie, considerând: $\mathbf{X}_k$, $\mathbf{X}_m \rightarrow \mathbf{X}_{k+1}$ şi $\mathbf{X}_{k+1} \rightarrow \mathbf{X}_{k+2}$.

Se generează astfel noi *puncte* de trecere intermediare, astfel încât o interpolare, viteză constantă/accelerație constantă, a mişcărilor cuplelor cinematice, conduce în spațiul operațional la o traiectorie suficient de apropiată de fiecare segment $[\mathbf{X}_i\ \mathbf{X}_{i+1}]$ al liniei poligonale inițiale.

Numărul transformărilor de coordonate este astfel considerabil redus în comparație cu alte metode ce lucrează în spațiul operațional. Dacă se notează cu T_D şi T_I duratele de calcul ale transformărilor de coordonate directe şi inverse, durata T_k^* a calculelor între $\mathbf{X}_k$ şi $\mathbf{X}_{k+1}$ va fi astfel [D7]:

$$T_k^* \le n T_D + (n+1) T_I \qquad (2.145)$$

unde n este numărul de diviziuni ale intervalului.

Convergența algoritmului prezentat este destul de rapidă. Eroarea converge rapid, deviația maximă fiind redusă la fiecare nivel de recursivitate. În majoritatea situațiilor, fiecare subdiviziune a intervalului divizează eroarea maximă în $\mathbf{X}$ printr-un factor cuprins între 2 (mişcări largi) şi 4 (mişcări foarte mici), şi algoritmul propus converge foarte rapid.

Problemele de convergență a metodei apar în apropierea configurațiilor degenerate ale manipulatorului, când accidental, un *punct* intermediar se situează într-o singularitate a manipulatorului, sau în vecinătatea acesteia. Această situație se poate întâlni în practică în punctul de intersecție a trei axe concurente, când ultima axă este practic aliniată cu prima axă a articulației. Pentru modificarea algoritmului în apropierea acestor configurații, se ține seama de geometria particulară a robotului.

Calculele necesare utilizării acestei metode "on–line" nu sunt foarte restrictive. În plus, metoda permite modificări "on-line" ale traiectoriilor planificate printr-un nivel superior de comandă, fiindcă nu se referă în principiu decât la segmentul curent $[\mathbf{X}_i\ \mathbf{X}_{i+1}]$.

Totuşi trebuie să se determine duratele T_k, pentru a obține o mişcare rapidă fără inducerea unor erori datorate motoarelor de acționare, inerției şi vibrațiilor mecanismului.

BUPT

Aşadar, ca şi pentru alte metode de comadă cinematică, acceleraţiile folosite sunt de tipul "totul sau nimic". Dacă aceste discontinuităţi nu sunt filtrate, ele conduc la şocuri nedorite şi ar fi convenabil să se recurgă la legi de accelerare şi decelerare prezentând continuitatea dorită (polinomiale de exemplu).

2.8. CONCLUZII

Generarea traiectoriei unui robot se realizează, în mod natural, în spaţiul operaţional. Pentru generarea traiectoriei în spaţiul operaţional, generatorul de traiectorie furnizează evoluţia în timp a poziţiei şi orientării efectorului final, respectiv permite controlul vitezei şi acceleraţiei punctului caracteristic al robotului. Marele avantaj al acestui mod de generare a mişcării constă în faptul că se poate realiza controlul geometriei traiectoriei.

Abordarea concretă a generării traiectoriei necesită rezolvarea unor etape preliminare:

- determinarea volumului accesibil al robotului;
- determinarea modelului mediului;
- impunerea restricţiilor traiectoriei.

Indiferent de modul de generare a traiectoriei în coordonate operaţionale (punct cu punct, traiectorie continuă, etc.), pentru a obţine traiectoria dorită este necesar să se apeleze transformatorul de coordonate (**MGI**).

Avantajele şi dezavantajele generării mişcării în spaţiul operaţional au fost prezentate în capitolul 1.5.

Una dintre metodele cele mai utilizate pentru generarea mişcării în spaţiul operaţional constă în interpolarea liniară. Tranziţia între segmentele de traiectorie se poate face cu acceleraţie constantă, asigurând astfel o deplasare lină. Impunerea restricţiilor cinematice şi a unui criteriu de optimizare (timp de parcurgere minim) conduce la un sistem neliniar. Pentru a nu utiliza metode complicate de programare neliniară sunt prezentate două soluţii. Algoritmii celor două soluţii au fost implementaţi într-un program de calcul care a permis un studiu comparativ al celor două metode. Din calculul timpului minim şi al abaterilor de la traiectoria poligonală impusă s-a constatat că metoda propusă de Tonder necesită un timp de parcurgere mai mic, dar erorile sunt mai mari.

Rezolvarea completă a generării mişcării în spaţiul operaţional necesită şi luarea în considerare a evoluţiei orientării efectorului final. Principiul metodei este similar cu cel al cazului când se neglijează orientarea, tranziţia între segmentele traiectoriei realizându-se cu acceleraţii, liniară şi unghiulară, constante. Considerarea restricţiilor conduce şi în acest caz la un sistem neliniar care apelează, pentru rezolvare, la tehnici de programare neliniară.

Interpolarea liniară cu tranziţie parabolică între segmentele traiectoriei, în ambele cazuri (cu şi fără considerarea orientării efectorului final) se poate aplica simplu, dar volumul mare al calculelor necesare nu poate fi efectuat on-line.

Tot pe principiul interpolării liniare şi tranziţiei parabolice între segmentele de traiectorie se bazează şi metoda de generare a mişcării cu matricea conducere **D**(r). Metoda de generare propusă de R.Paul a fost reluată de mai mulţi autori în literatura de

BUPT

specialitate. Modul în care este prezentată metoda în acest capitol constituie o sinteză a variantelor întâlnite în studiul bibliografic întreprins. De asemenea, abordarea metodei cu matricea conducere **D**(r) a fost corelată cu structurarea acțiunilor robotului. Mişcarea unui robot fiind în directă dependență cu fazele în care este structurată sarcina acestuia, s-a pornit de la exprimarea ecuațiilor de transformare corespunzătoare fiecărei faze. Dezavantajul acestei metode constă în volumul mare de calcule ce se efectuează la fiecare perioadă de eşantionare. Avantajul constă în deplasarea rectilinie, cu viteză constantă, modificarea controlată a orientării efectorului final şi tranziției line între segmente.

Pentru reducerea numărului de transformări de coordonate, pornind de la o metodă propusă de Taylor pentru interpolare cvasiliniară în spațiul configurațiilor, cu abatere controlată în spațiul operațional, s-a formulat un algoritm clar, simplu de aplicat, bazat pe metoda înjumătățirii intervalului drept criteriu de convergență.

BUPT

3. METODE DE GENERARE A TRAIECTORIILOR ÎN SPAȚIUL CUPLELOR CINEMATICE

Generarea mișcării între două *puncte* în spațiul cuplelor cinematice conducătoare poate fi abordată în două moduri:

- traiectoria este liberă între cele două *puncte* și se raționează deci în spațiul cuplelor;
- traiectoria este impusă (mișcare rectilinie de exemplu) și este descrisă în spațiul operațional.

Se consideră un robot cu N grade de libertate. Fie $\mathbf{q}_i$ și $\mathbf{q}_f$ vectorii coordonatelor cuplelor corespunzând configurației inițială și finală. Se notează prin $\mathbf{K}_v$ și $\mathbf{K}_a$ vectorii viteză maximă, respectiv accelerație maximă pentru cuplele cinematice. Parametrii K_v^j sunt, în general, calculați cu exactitate pornind de la caracteristicile servomotoarelor și raportul de transmisie al mecanismelor, în timp ce parametrii K_a^j sunt aproximați prin raportul dintre cuplul motor (forță/moment) maxim și inerția maximă, atinse de cuplele cinematice.

Mișcarea, interpolată între $\mathbf{q}_i$ și $\mathbf{q}_f$ în funcție de timpul t, este descrisă de următoarea ecuație:

$$\mathbf{q}(t) = \mathbf{q}_i + \lambda(t)\,\mathbf{D} \qquad (3.1)$$

unde $0 < t < t_f$, iar $\mathbf{D} = \mathbf{q}_f - \mathbf{q}_i$ este vectorul deplasărilor cuplelor cinematice. Valorile la limită ale funcției de interpolare $\lambda(t)$ sunt date de: $\lambda(0) = 0$ și $\lambda(t_f) = 1$. Ecuația de mișcare (3.1) se mai poate scrie și sub forma:

$$\mathbf{q}(t) = \mathbf{q}_f - [1 - \lambda(t)]\,\mathbf{D} \qquad (3.2)$$

ceea ce convine în cazurile de urmărire a configurației finale, când termenul $\mathbf{q}_f$ variază [T3].

Există mai multe tipuri de funcții de interpolare care permit trecerea prin $\mathbf{q}_i$ la $t = 0$ și prin $\mathbf{q}_f$ la $t = t_f$. În cele două *puncte* exprimate în spațiul cuplelor cinematice prin $\mathbf{q}_i$ și $\mathbf{q}_f$, viteza punctului caracteristic poate fi nulă sau diferită de zero. Rezultă astfel oportunitatea sudierii a două cazuri:

- traiectorii care unesc două *puncte* ale spațiului (cuplelor sau operațional) cu viteze inițiale și finale nule (*puncte* de oprire);
- traiectorii care unesc două *puncte* ale spațiului (cuplelor sau operațional) cu viteze inițiale și finale diferite de zero (*puncte* de trecere).

Interpolarea cea mai folosită, atât în spațiul cuplelor, cât și în spațiul operațional, este interpolarea liniară.

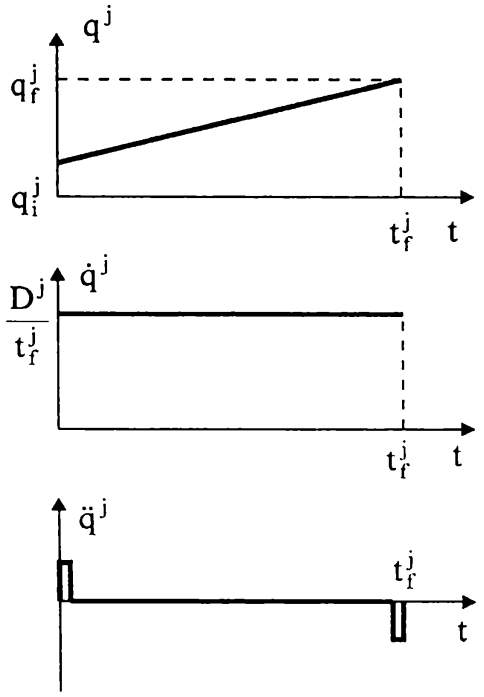
Interpolarea liniară

Interpolarea liniară este interpolarea cea mai simplă. Mișcarea fiecărei cuple cinematice este descrisă de o funcție liniară în timp. În afară de cazul particular al

BUPT

roboţilor cartezieni, traiectoria punctului caracteristic al efectorului final nu este rectilinie. Ecuaţia mişcării se scrie:

$$\mathbf{q}(t) = \mathbf{q}_i + (t/t_f)\,\mathbf{D} \qquad (3.3)$$



Această lege de mişcare, care impune o viteză constantă de-a lungul traiectoriei este continuă în poziţie, dar discontinuă în viteză. Practic, începutul şi sfârşitul mişcării sunt marcate de discontinuităţi.

Fig. 3.1. Interpolare liniară pentru cupla cinematică *j*

3.1. GENERAREA MIŞCĂRII ÎNTRE DOUĂ *PUNCTE* DE OPRIRE

Generarea unei traiectorii în coordonatele cuplelor cinematice, între două *puncte* de oprire din spaţiul de lucru, poate fi realizată cu ajutorul mai multor tipuri de funcţii de interpolare.

3.1.1. INTERPOLAREA POLINOMIALĂ

Polinoamele de grad superior sunt greu de folosit în interpolarea traiectoriilor datorită dificultăţii calculului coeficienţilor funcţiei polinomiale şi al punctelor de extrem. Modelele de interpolare polinomială cel mai frecvent întâlnite sunt interpolarea prin polinoame de gradul trei şi cinci.

Polinomul de gradul trei

Se consideră deci, între *punctele* de oprire, pentru cupla cinematică *j* un profil parabolic de viteză $\dot{\mathbf{q}}^j(t) > 0$. Se notează cu D^j deplasarea cuplei cinematice *j* şi cu t_f^j timpul de parcurs. Dacă se impun viteze nule în *punctul* de plecare şi de sosire, se adaugă două restricţii la cele două restricţii de poziţie de la interpolarea liniară. Se determină legea orară a mişcării, precum şi expresia timpului minim t_f^j, compatibil cu restricţiile de viteză, respectiv acceleraţie (viteza maximă K_v^j şi acceleraţia maximă K_a^j).

Gradul minim al polinomului care satisface cele patru restricţii este trei şi are următoarea formă generală [C3], [D6], [D14]:

BUPT

$$q(t) = a_3t^3 + a_2t^2 + a_1t + a_0 \qquad (3.4)$$

cu condiţiile iniţiale şi finale:

$$q(0) = q_i \qquad q(t_f) = q_f \qquad \dot{q}(0) = 0 \qquad \dot{q}(t_f) = 0 \qquad (3.5)$$

Se determină coeficienţii funcţiei polinomiale (3.4) din condiţiile (3.5) şi se obţine:

$$a_0 = q_i \qquad a_1 = 0 \qquad a_2 = (3/t_f^2)\,\mathbf{D} \qquad a_3 = (-2/t_f^3)\,\mathbf{D}$$

Funcţia polinomială (3.4) se poate scrie sub formele (3.1) şi (3.2) pentru:

$$\lambda(t) = 3\,(t/t_f)^2 - 2\,(t/t_f)^3 \qquad (3.6)$$

Legea orară a mişcării rezultă:

$$q^j(t) = q_i^j + \left[3\left(\frac{t}{t_f^j}\right)^2 - 2\left(\frac{t}{t_f^j}\right)^3\right]\cdot D^j \qquad (3.7)$$

Din legea orară a mişcării se obţin legile de variaţie pentru viteză şi acceleraţie, precum şi valorile maxime ale acestora:

$$\dot{q}^j(t) = 6\left[\frac{t}{t_f^j} - \left(\frac{t}{t_f^j}\right)^2\right]\frac{D^j}{t_f^j} \qquad \ddot{q}^j(t) = 6\left[1 - 2\frac{t}{t_f^j}\right]\frac{D^j}{\left(t_f^j\right)^2} \qquad (3.8)$$

Figura 3.2 prezintă, pentru axa *j*, evoluţia poziţiei, vitezei şi acceleraţiei. Această lege de mişcare asigură continuitatea vitezelor dar nu şi pe cea a acceleraţiilor. În practică, roboţii sunt destul de rigizi pentru ca această discontinuitate să fie filtrată de structura mecanică. Utilizarea unui polinom de gradul trei este deci, în general, satisfăcătoare în majoritatea aplicaţiilor.

Pentru o cuplă cinematică oarecare *j*, viteza este maximă când $t = t_f/2$, având valoarea: $\left|\dot{q}^j_{max}\right| = \frac{3\left|D^j\right|}{2t_f^j} \le K_v^j$, cu $\left|D^j\right| = \left|q_f^j - q_i^j\right|$, iar acceleraţia este maximă la $t = 0$, având valoarea: $\left|\ddot{q}^j_{max}\right| = \frac{6\left|D^j\right|}{\left(t_f^j\right)^2} \le K_a^j$.

Pentru a obţine timpul optim t_f^j, trebuie să se exprime una din cele două restricţii: $\dot{q}^j_{max} = K_v^j$ sau $\ddot{q}^j_{max} = K_a^j$, care sunt echivalente dacă: $\frac{3}{2}\frac{D^j}{K_v^j} = 6\frac{D^j}{K_a^j}$. Rezultă deplasarea $D^j = \frac{8}{3}\frac{\left(K_v^j\right)^2}{K_a^j}$. Cele două cazuri sunt:

$$D^j \ge \frac{8}{3}\frac{\left(K_v^j\right)^2}{K_a^j} \Rightarrow t_f^j = \frac{3}{2}\frac{D^j}{K_v^j} \qquad (3.9)$$

când restricţia de viteză este cea preponderentă, şi:

$$D^j \le \frac{8}{3}\frac{\left(K_v^j\right)^2}{K_a^j} \Rightarrow t_f^j = \sqrt{\frac{6D^j}{K_a^j}} \qquad (3.10)$$

când restricţia de acceleraţie este cea preponderentă.

BUPT

Polinomul de gradul cinci

Dacă se cercetează continuitatea acceleraţiilor, trebuie satisfăcute şase restricţii şi polinomul de interpolare trebuie să fie de gradul cinci [B8], [D7], [D14]. Alegând în plus $\ddot{\mathbf{q}}(0) = \mathbf{0}$ si $\ddot{\mathbf{q}}(t_f) = \mathbf{0}$, se arată că acesta se poate pune sub forma (3.1) sau (3.2), dacă:

$$\lambda(t) = 10\left(\frac{t}{t_f}\right)^3 - 15\left(\frac{t}{t_f}\right)^4 + 6\left(\frac{t}{t_f}\right)^5 \tag{3.11}$$

Legea orară a mişcării rezultă:

$$q^j(t) = q_i^j + \left[10\left(\frac{t}{t_f^j}\right)^3 - 15\left(\frac{t}{t_f^j}\right)^4 + 6\left(\frac{t}{t_f^j}\right)^5\right] D^j \tag{3.12}$$

Legile de variaţie pentru viteză, respectiv pentru acceleraţie, sunt:

$$\dot{q}^j(t) = 30\left[\left(\frac{t}{t_f^j}\right)^2 - 2\left(\frac{t}{t_f^j}\right)^3 + \left(\frac{t}{t_f^j}\right)^4\right]\frac{D^j}{t_f^j} \tag{3.13}$$

$$\ddot{q}^j = 60\left[\frac{t}{t_f^j} - 3\left(\frac{t}{t_f^j}\right)^2 + 2\left(\frac{t}{t_f^j}\right)^3\right]\frac{D^j}{(t_f^j)^2} \tag{3.14}$$

Evoluţiile poziţiei, vitezei şi acceleraţiei pentru cupla cinematică *j*, sunt prezentate în fig.3.3. Pentru polinomul de gradul cinci, viteza maximă se atinge pentru $t = \frac{t_f^j}{2}$ şi are valoarea: $\left|\dot{q}_{max}^j\right| = \frac{15}{8}\frac{\left|D^j\right|}{t_f^j} \le K_v^j$, acceleraţiile extreme se ating în interiorul intervalului $\left[0,\ t_f^j\right]$, pentru $t = \left(3 \pm \sqrt{3}\right)\frac{t_f^j}{6}$ şi sunt: $\left|\ddot{q}_{max}^j\right| = \frac{10\sqrt{3}}{3}\frac{\left|D^j\right|}{(t_f^j)^2} \le K_a^j$.

Unul din avantajele principale ale polinomului de gradul cinci faţă de cel de gradul trei este asigurarea acceleraţiei nule la pornirea şi oprirea mişcării ($t = 0$ şi $t = t_f$). Astfel, se poate asigura o mişcare lină, fără şocuri.

Pentru obţinerea timpului optim t_f^j se exprimă cele două restricţii $\left|\dot{q}_{max}^j\right| = K_v^j$ şi $\left|\ddot{q}_{max}^j\right| = K_a^j$ şi rezultă:

$$t_f^j = \frac{15}{8}\frac{\left|D^j\right|}{K_v^j} \tag{3.15}$$

când restricţia de viteză este cea preponderentă, şi:

$$t_f^j = \sqrt{\frac{10\sqrt{3}}{3}\frac{\left|D^j\right|}{K_a^j}} \tag{3.16}$$

când restricţia de acceleraţie este cea preponderentă.

Impunând aceeaşi deplasare D^j şi aceeaşi acceleraţie maximă $\left|\ddot{q}_{max}^j\right|$, traiectoria polinomială de gradul trei durează mai mult decât traiectoria polinomială de gradul cinci:

$$t_f^{(3)} = \frac{3\sqrt{3}}{5}t_f^{(5)} \tag{3.17}$$

BUPT

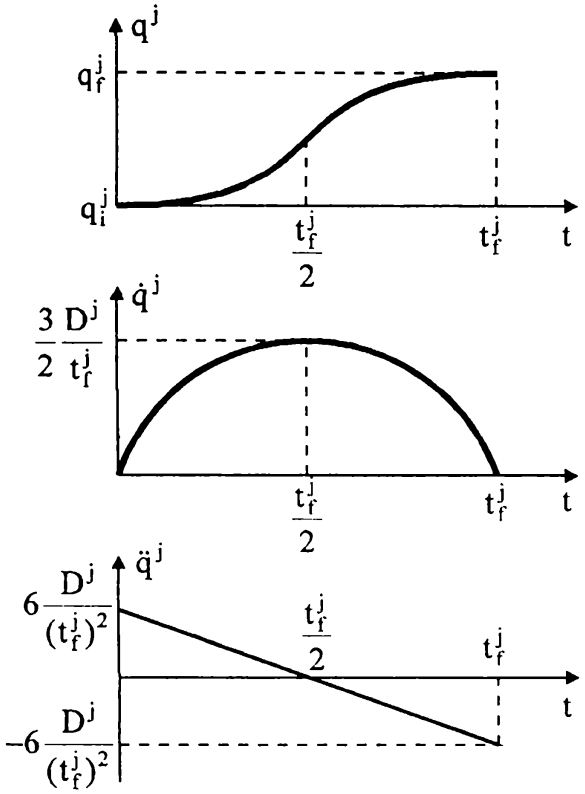


Fig. 3.2. Legea polinomială de gradul trei pentru o cuplă cinematică *j*

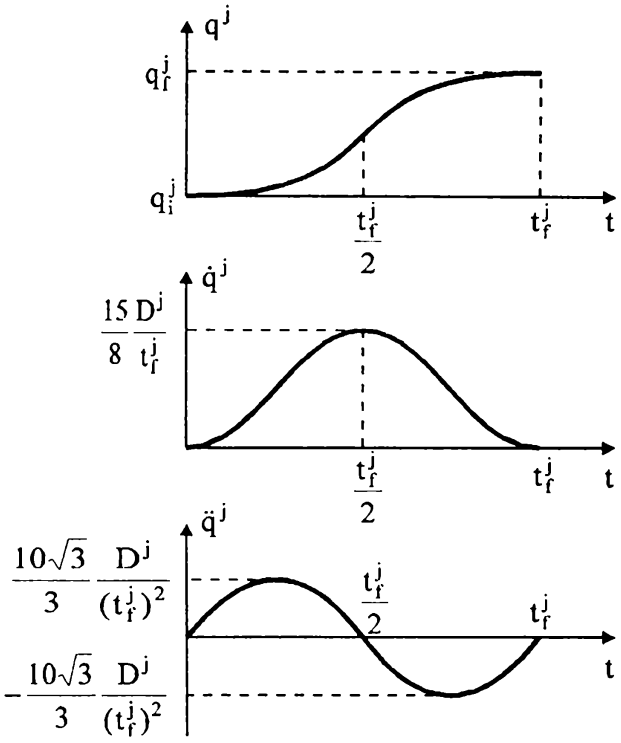


Fig.3.3. Legea polinomială de gradul cinci pentru o cuplă cinematică *j*

3.1.2. LEGEA CU PROFIL TRIUNGHIULAR DE VITEZĂ

Mişcarea este reprezentată în acest caz printr-o fază de accelerare, cu acceleraţie constantă până la $t_f/2$, apoi printr-o fază de decelerare (fig.3.4), de asemenea cu acceleraţie constantă. Vitezele, iniţială şi finală, sunt nule. Mişcarea este deci, continuă în poziţie şi viteză şi discontinuă în acceleraţie.

Poziţia este exprimată prin:

$$\begin{cases} q^j(t) = q_i^j + 2\left(\dfrac{t}{t_f^j}\right)^2 D^j & 0 \le t \le \dfrac{t_f^j}{2} \\ q^j(t) = q_i^j + \left[-1 + 4\left(\dfrac{t}{t_f^j}\right) - 2\left(\dfrac{t}{t_f^j}\right)^2\right] D^j & \dfrac{t_f^j}{2} \le t \le t_f^j \end{cases} \tag{3.18}$$

Legile de variaţie pentru viteză, respectiv pentru acceleraţie, pentru o cuplă cinematică *j* , sunt:

BUPT

$$\dot{q}^j(t) = \begin{cases} 4\dfrac{D^j}{\left(t_f^j\right)^2}t & 0 \le t \le \dfrac{t_f^j}{2} \\ 4\left(1-\dfrac{t}{t_f^j}\right)\dfrac{D^j}{t_f^j} & \dfrac{t_f^j}{2} \le t \le t_f^j \end{cases} \tag{3.19}$$

$$\ddot{q}^j(t) = \begin{cases} 4\dfrac{D^j}{\left(t_f^j\right)^2} & 0 \le t \le \dfrac{t_f^j}{2} \\ -4\dfrac{D^j}{\left(t_f^j\right)^2} & \dfrac{t_f^j}{2} \le t \le t_f^j \end{cases} \tag{3.20}$$

Pentru o cuplă cinematică *j* dată, viteza şi acceleraţia maxime au următoarele expresii:

$$\left|\dot{q}_{max}^j\right| = \frac{2\cdot\left|D^j\right|}{t_f^j}; \qquad \left|\ddot{q}_{max}^j\right| = \frac{4\cdot\left|D^j\right|}{(t_f^j)^2} \tag{3.21}$$

Traiectoria de acest tip, realizând deplasarea cuplei cinematice cu acceleraţie maximă, este optimală din punctul de vedere al timpului de mişcare. Pentru o distanţă D şi o acceleraţie impusă, aceasta este cea mai rapidă traiectorie. Dezavantajul acestei legi de mişcare constă în obţinerea unor supraacceleraţii de valori foarte mari.

3.1.3. LEGEA CU PROFIL TRAPEZOIDAL DE VITEZĂ

Prin adăugarea unui palier de viteză (fig.3.5), se obţine saturarea acceleraţiei şi scăderea timpului de parcurs (fig.3.6).

Se presupune viteza saturată pentru cupla cinematică *j*. Din ecuaţiile (3.21) se poate scrie inegalitatea:

$$\frac{\left|D^j\right|}{K_v^j} > \sqrt{\frac{\left|D^j\right|}{K_a^j}} \tag{3.22}$$

Ridicând la pătrat această relaţie, se deduce condiţia de existenţă a palierului de viteză:

$$\left|D^j\right| > \frac{\left(K_v^j\right)^2}{K_a^j} \tag{3.23}$$

Dintre legile de mişcare care asigură continuitatea vitezei, legea trapezoidală este legea de mişcare optimă în timp. Mişcarea cuplei cinematice *j* este reprezentată prin relaţiile următoare:

BUPT

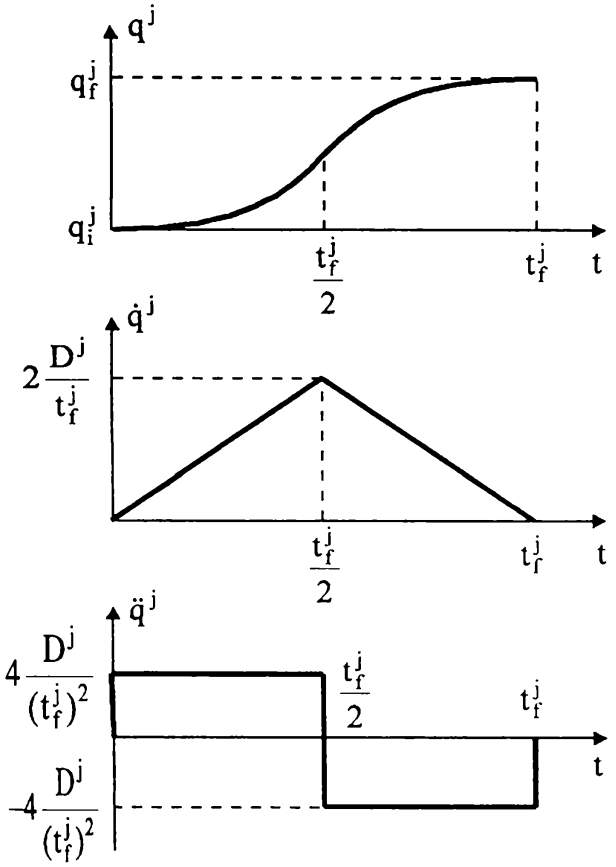


Fig. 3.4. Legea cu profil triunghiular de vitezaă, pentru o cuplă cinematică *j*

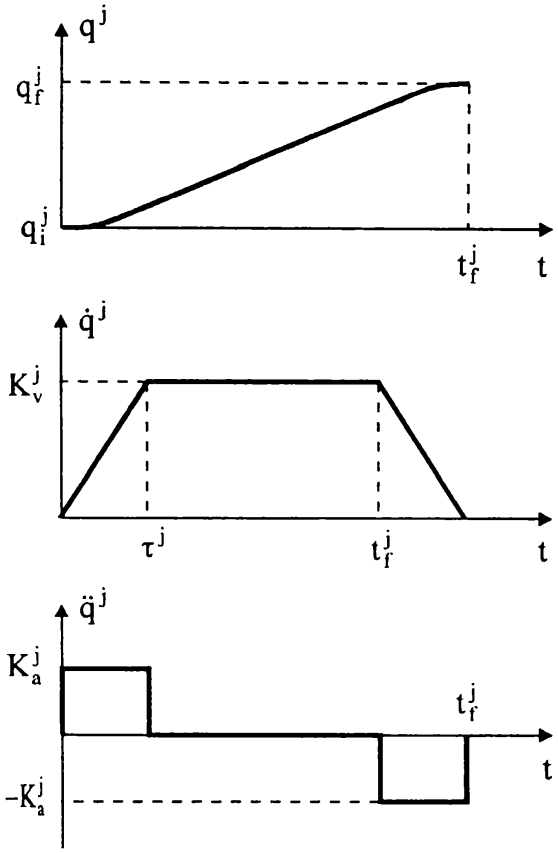


Fig.3.5. Evoluția poziției, vitezei şi accelerației corespunzătoare legii trapezoidale, pentru cupla *j*

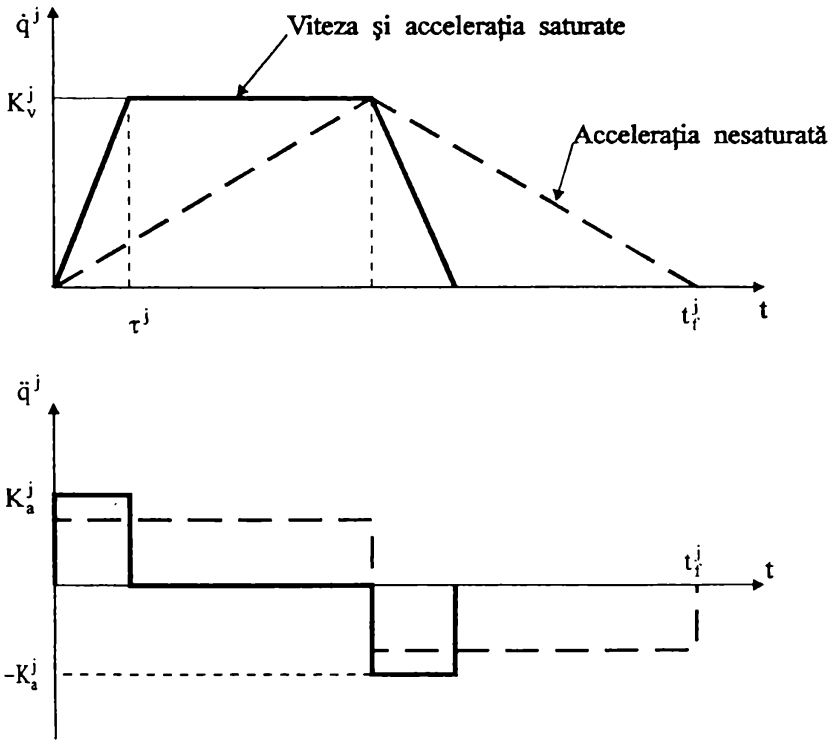


Fig.3.6. Relaţia dintre legea cu profil triunghiular de viteză şi legea cu profil trapezoidal de viteză, pentru o cuplă cinematică *j*

BUPT

$$q^j(t)=\begin{cases} q_i^j+\dfrac{t^2}{2}K_a^j\cdot Sgn(D^j) & 0\le t\le \tau^j \\ q_i^j+(t-\dfrac{\tau^j}{2})\cdot K_v^j\cdot Sgn(D^j) & \tau^j\le t\le t_f^j-\tau^j \\ q_f^j-\dfrac{1}{2}(t_f^j-t)^2\cdot K_a^j\cdot Sgn(D^j) & t_f^j-\tau^j\le t\le t_f^j \end{cases} \tag{3.24}$$

cu $\tau^j=\dfrac{K_v^j}{K_a^j}$.

Legile de variație pentru viteză, respectiv pentru accelerație sunt:

$$\dot q^j(t)=\begin{cases} K_a^j Sgn(D^j)\,t & 0\le t\le \tau^j \\ K_v^j Sgn(D^j) & \tau^j\le t\le t_f^j-\tau^j \\ K_a^j Sgn(D^j)(t_f^j-t) & t_f^j-\tau^j\le t\le t_f^j \end{cases} \tag{3.25}$$

$$\ddot q^j(t)=\begin{cases} K_a^j Sgn(D^j) & 0\le t\le \tau^j \\ 0 & \tau^j\le t\le t_f^j-\tau^j \\ -K_a^j Sgn(D^j) & t_f^j-\tau^j\le t\le t_f^j \end{cases} \tag{3.26}$$

Aria trapezului ce reprezintă evoluția vitezelor este egală cu distanța parcursă în intervalul $\left[0,\ t_f^j\right]$. Se poate scrie:

$$\left|D^j\right|=\left|q_f^j-q_i^j\right|=2\int_0^{\tau^j}K_a^j t\,dt+\int_{\tau^j}^{t_f^j-\tau^j}K_v^j\,dt=K_v^j t_f^j-\frac{\left(K_v^j\right)^2}{K_a^j} \tag{3.27}$$

Din relația de mai sus se deduce timpul minim pentru cupla cinematică *j*:

$$t_f^j=\frac{K_v^j}{K_a^j}+\frac{\left|D^j\right|}{K_v^j}=\tau^j+\frac{\left|D^j\right|}{K_v^j} \tag{3.28}$$

Dacă se doreşte deplasarea într-un timp t_f^j mai mare, trebuie să se reducă palierul de viteză care devine v_{max}^j. Accelerația se va nota cu a_{max}^j, iar timpul obținut se va nota cu $t_f^{'j}$.

3.1.4. PROFILUL DE VITEZĂ NETEZIT

Profilul de viteză trapezoidal sau triunghiular din figurile 3.5 şi 3.4 prezintă discontinuități ale accelerației cuplei cinematice. Aceste discontinuități pot fi responsabile de incidentele ce apar în timpul mişcării. În acest scop, anumite controlere ale roboților sunt prevăzute cu un generator de valori prescrise pentru viteză, care asigură continuitatea accelerației. Punctele de discontinuitate din graficele vitezelor (fig.3.5 şi

BUPT

fig.3.4) sunt înlocuite cu "racorduri" parabolice, stabilite din condiția ca derivata accelerației să fie, în valoare absolută, egală cu o constantă a^j.

Profilul trapezoidal netezit pentru viteză $\dot{q}^j(t)$ şi profilul accelerației $\ddot{q}^j(t)$ sunt prezentate în figura 3.7. Durata τ'^j a racordării este egală cu $\frac{K_a^j}{a^j}$. Este evident că, pentru o aceeaşi deplasare D^j (aceeaşi arie sub curbele vitezelor din figura 3.5), durata parcursului corespunzătoare profilului de viteză netezit t'^j_f este majorată datorită lui τ'^j, în raport cu t_f^j. Astfel:

$$t_f'^j = \frac{D^j}{K_v^j} + \frac{K_v^j}{K_a^j} + \frac{K_a^j}{a^j} \tag{3.29}$$

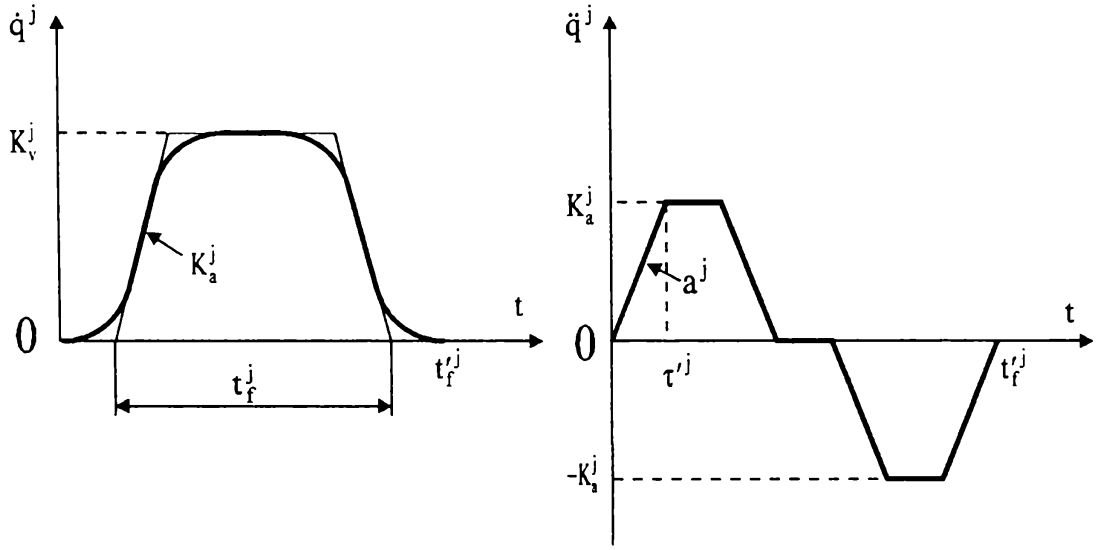


Fig.3.7. Profilul trapezoidal nezit pentru viteză
şi profilul corespunzător pentru accelerație

Trecerea de la profilul triunghiular de viteză din figura 3.4, la un profil triunghiular netezit, reprezentat în figura 3.8, este mai puțin imediată ca în cazul profilului trapezoidal. Timpul corespunzător unei mişcări cu profil triunghiular netezit de viteză este soluția ecuației de gradul doi:

$$\left(t_f'^j\right)^2 - 2\frac{K_a^j}{a^j} t_f'^j - 4\frac{D^j}{K_a^j} = 0 \tag{3.30}$$

Această relație presupune deci existența porțiunilor liniare în curba $\dot{q}^j(t)$. Un interes deosebit prezintă cazul în care această condiție nu este îndeplinită.

BUPT

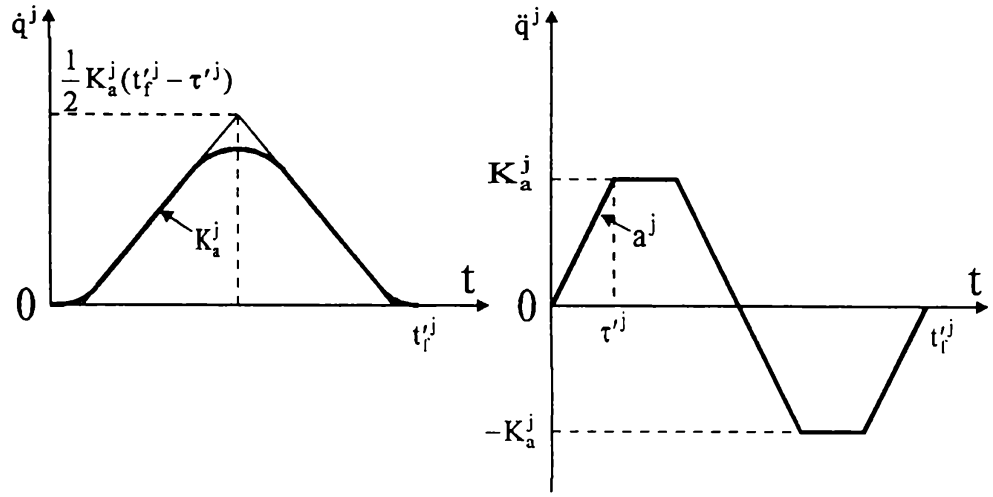


Fig.3.8. Profilul triunghiular netzit pentru viteză
şi profilul corespunzător pentru acceleraţie.

3.1.5. TRAIECTORIA CRITIC AMORTIZATĂ

O traiectorie critic amortizată este o soluţie a ecuaţiei diferenţiale [D7]:

$$\ddot{q}^j(t) + 2\alpha^j\dot{q}^j(t) + (\alpha^j)^2 q^j(t) = (\alpha^j)^2 D^j \qquad (3.31)$$

în care α este o constantă.

În relaţiile următoare, pentru simplificarea redactării, se renunţă la indicarea cuplei *j*.

Pentru condiţiile iniţiale nule, soluţia ecuaţiei diferenţiale este:

$$\begin{aligned} q(t) &= D\left[1 - e^{-\alpha t}(1+\alpha t)\right] \qquad \dot{q}(t) = D\left(\alpha^2 t e^{-\alpha t}\right) \\ \ddot{q}(t) &= D\left[\alpha^2 e^{-\alpha t}(1-\alpha t)\right] \qquad \dddot{q}(t) = D\left[\alpha^2 e^{-\alpha t}(-2+\alpha t)\right] \end{aligned} \qquad t \in [0,\ \infty) \qquad (3.32)$$

Deoarece robotul tinde spre *punctul* final în timp infinit, practic se consideră un interval de timp [0, t_f], astfel încât să fie parcursă distanţa q(1-e), cu eroarea de poziţionare e $\in$ (0, 1).

Pentru a putea compara acest tip de traiectorie cu traiectoriile definite anterior, se introduce factorul adimensional Ωt_f caracteristic mişcării, unde $\Omega = \sqrt{\frac{K_a}{|D|}}$. Se consideră $\alpha = \Omega$ şi se calculează numeric e = f(Ωt_f). Traiectoria critic amortizată este cea mai lentă comparativ cu traiectoria cu profil triunghiular de viteză..

3.1.6. TRAIECTORIA COSINUSOIDALĂ

Forma normalizată a traiectoriei este [D7]:

BUPT

$$q(p)=\frac{1}{2}\left[1-\cos(\pi p)\right] \qquad q'(p)=\frac{\pi}{2}\sin(\pi p)$$
$$q''(p)=\left(\frac{\pi^2}{2}\right)\cos(\pi p) \qquad q'''(p)=-\left(\frac{\pi^3}{2}\right)\sin(\pi p) \qquad p\in[0,\ 1] \tag{3.33}$$

Condițiile inițiale şi finale sunt:

$$q(0)=0 \qquad q(1)=1 \qquad q'(0)=0 \qquad q'(1)=0$$
$$q''(0_+)=\frac{\pi^2}{2} \qquad q''(1_-)=-\frac{\pi^2}{2} \qquad q''(0_-)=q''(1_+)=0 \tag{3.34}$$

Forma reală a traiectoriei este:

$$q(t)=q_i+\frac{D}{2}\left[1-\cos\left(\pi\frac{t}{t_f}\right)\right] \qquad \dot{q}(t)=\frac{D}{2}\frac{\pi}{t_f}\sin\left(\pi\frac{t}{t_f}\right)$$
$$\ddot{q}(t)=\frac{D}{2}\left(\frac{\pi}{t_f}\right)^2\cos\left(\pi\frac{t}{t_f}\right) \qquad \dddot{q}(t)=-\frac{D}{2}\left(\frac{\pi}{t_f}\right)^3\sin\left(\pi\frac{t}{t_f}\right) \qquad t\in[0,\ t_f] \tag{3.35}$$

Această traiectorie are dezavantajul că extremele acceleraţiei $\pm\frac{D}{2}\left(\frac{\pi}{t_f}\right)^2$ sunt atinse la capetele intervalului [0, t_f], dând naştere la şocuri la oprire şi pornire. Parametrul caracteristic mişcării Ωt_f este $\frac{\pi}{\sqrt{2}}$, reprezentând o mişcare cu 11% mai înceată decât cea realizată cu funcţia de interpolare cu profil triunghiular de viteză [D7].

3.1.7. TRAIECTORIA SINUSOIDALĂ SUPRAPUSĂ PESTE RAMPĂ

Forma normalizată a traiectoriei este [D7]:

$$q(p)=p-\left(\frac{1}{2\pi}\right)\sin(2\pi p) \qquad q'(p)=1-\cos(2\pi p) \qquad p\in[0,\ 1]$$
$$q''(p)=2\pi\sin(2\pi p) \qquad q'''(p)=4\pi^2\cos(2\pi p) \tag{3.36}$$

Această traiectorie reprezintă soluţia ecuaţiei diferenţiale:

$$q''(p)=4\pi^2\left[p-q(p)\right] \tag{3.37}$$

corespunzătoare condiţiilor iniţiale nule.

Condiţiile iniţiale şi finale sunt:

$$q(0)=0 \qquad q'(0)=0 \qquad q''(0)=0$$
$$q(1)=1 \qquad q'(1)=0 \qquad q''(1)=0 \tag{3.38}$$

Forma reală a traiectoriei este:

BUPT

$$q(t) = D\left[\frac{t}{t_f} - \left(\frac{1}{2\pi}\right)\sin\left(2\pi\frac{t}{t_f}\right)\right] \qquad \dot{q}(t) = \frac{D}{t_f}\left[1 - \cos\left(2\pi\frac{t}{t_f}\right)\right]$$

$$\ddot{q}(t) = 2\pi\frac{D}{t_f^{\,2}}\sin\left(2\pi\frac{t}{t_f}\right) \qquad \dddot{q}(t) = 4\pi^2\frac{D}{t_f^{\,3}}\cos\left(2\pi\frac{t}{t_f}\right) \qquad t \in [0,\ t_f] \quad (3.39)$$

Extremele accelerației $\pm\frac{D}{2}\left(\frac{\pi}{t_f}\right)^2$ sunt atinse în interiorul intervalului [0, t_f], iar parametrul caracteristic mișcării este $\Omega t_f = \sqrt{2\pi}$. Această traiectorie are o durată cu aproximativ 25% mai mare decât una realizată de funcția de interpolare cu profil triunghiular de viteză [D7].

3.2. GENERAREA MIȘCĂRII ÎNTRE DOUĂ *PUNCTE* DE TRECERE

Pentru ocolirea obstacolelor din spațiul de lucru, între *punctele* principale (inițial $\mathbf{X}_i$ şi final $\mathbf{X}_f$) se introduc *puncte* intermediare. Între două *puncte* intermediare ale traiectoriei, nu este eficient să se interpoleze traiectoria cu una dintre funcțiile de interpolare care asigură viteze inițiale şi finale egale cu zero, deoarece robotul ar trebui să se oprească în toate *punctele* intermediare, mărind mult timpul de execuție a mişcării. Apare astfel necesitatea introducerii unor funcții de interpolare care să permită viteze diferite de zero la capetele unui segment de traiectorie. (polinomiale, sinusoidale, sinusoidale ponderate).

În acest caz, problema constă deci în generarea mişcării, cu viteză diferită de zero, astfel încât traiectoria este constrânsă să treacă prin sau în vecinătatea unui anumit număr de *puncte* intermediare (*puncte de trecere, via points*). Aceste *puncte* nu au alt rol funcțional decât să deformeze traiectoria pentru a evita coliziunile directe dintre robot şi mediul său înconjurător. Impunerea unor viteze diferite de zero în *punctele* intermediare permite reducerea timpului ciclului de mişcare şi netezirea traiectoriei.

Fie $\mathbf{q}_0$ şi $\mathbf{q}_m$ configurațiile cuplelor cinematice corespunzătoare fazelor de pornire şi sosire, şi $\mathbf{q}_1$, $\mathbf{q}_2$, ..., $\mathbf{q}_{m-1}$ configurațiile ce corespund *punctelor* intermediare. Se notează cu t_1, t_2, ..., t_m timpii de parcurs, presupuşi cunoscuți între două *puncte* succesive. Există un polinom de gradul *m* şi o singură funcție de interpolare a acestor *puncte*. Cei *m+1* coeficienți sunt soluția unui sistem de ecuații liniare. Deoarece *m* este ridicat, utilizarea acestui polinom nu este comodă. În particular, nu este uşor să se evalueze extremele pentru poziție, viteză şi accelerație şi nici valorile pentru care se produc (eventualele depăşiri ale restricțiilor riscă să fie mascate). Din aceste motive, se preferă utilizarea, în general, a mai multor polinoame de grad mai mic, mişcarea fiind obținută din porțiuni. Generarea mişcării cu *puncte* intermediare a fost studiată prin mai multe metode:

- interpolare polinomială, sinusoidală şi sinusoidală ponderată, pentru accelerație;
- interpolări liniare racordate prin tranziții cu accelerație constantă;
- interpolări, între fiecare două *puncte* succesive prin polinoame de gradul trei;
- interpolări folosind funcțiile spline.

Pentru simplificarea redactării, se renunță în continuare la indicarea cuplei *j*.

BUPT

3.2.1. FUNCȚIA DE INTERPOLARE POLINOMIALĂ

Pentru interpolarea accelerației se consideră o funcție polinomială de gradul 3, de forma [D7]:

$$\ddot{q}(t) = A_1 \frac{t}{t_f}\left(1-\frac{t}{t_f}\right)\left(\beta-\frac{t}{t_f}\right); \quad t \in [0,\ t_f] \tag{3.40}$$

Accelerația la capetele traiectoriei este nulă, iar constantele A_1 și β se determină din condițiile suplimentare impuse vitezei $\dot{q}(t)$. Integrând relația (3.40), se obține funcția de interpolare pentru viteză și poziție:

$$\dot{q}(t) = A_1\left[\frac{1}{4}\left(\frac{t}{t_f}\right)^4 + \frac{1+\beta}{3}\left(\frac{t}{t_f}\right)^3 + \frac{\beta}{2}\left(\frac{t}{t_f}\right)^2\right]t_f + A_2; \quad t \in [0,\ t_f] \tag{3.41}$$

$$q(t) = A_1\left[\frac{1}{20}\left(\frac{t}{t_f}\right)^5 - \frac{1+\beta}{12}\left(\frac{t}{t_f}\right)^4 + \frac{\beta}{6}\left(\frac{t}{t_f}\right)^3\right]t_f^2 + A_2 t_f\left(\frac{t}{t_f}\right) + A_3; \quad t \in [0,\ t_f] \tag{3.42}$$

Derivând aceeași relație (3.50), se obține funcția de interpolare a derivatei accelerației:

$$\dddot{q}(t) = \frac{A_1}{t_f}\left[3\left(\frac{t}{t_f}\right)^2 - 2\frac{1+\beta}{T}\left(\frac{t}{t_f}\right) + \beta\right]; \quad t \in [0,\ t_f] \tag{3.43}$$

Folosind condițiile inițiale și finale, se determină coeficienții necunoscuți din relațiile (3.40) — (3.42):

$$\begin{aligned} &A_3 = q(0); \quad A_2 = \dot{q}(0) \\ &q(t_f) = A_1\left(-\frac{1}{30}+\frac{\beta}{12}\right)t_f^2 + \dot{q}(0)t_f + q(0) \\ &\dot{q}(t_f) = A_1\left(-\frac{1}{12}+\frac{\beta}{6}\right)t_f + \dot{q}(0) \end{aligned} \tag{3.44}$$

Pentru determinarea domeniilor de variație ale constantelor β și A_1 se pot specifica și restricții suplimentare cum ar fi: accelerația să nu schimbe semnul pe domeniul $[0, t_f]$; viteza să fie monotonă sau să-și atingă extremul pe domeniul $[0, t_f]$. Impunând vitezele la capetele traiectoriei $\dot{q}(0)$, $\dot{q}(t_f)$, și o anumită formă de variație a vitezei pe traiectorie, se determină parametrii β și A_1 care definesc în mod univoc relațiile (3.40)- (3.42).

3.2.2. FUNCȚIA DE INTERPOLARE SINUSOIDALĂ

Pentru interpolarea accelerației se consideră două funcții sinusoidale definite pe intervalele $[0, t_m]$ și $[t_m, t_f]$:

$$\begin{cases} \ddot{q}_1(t) = A_{11}\sin\left(\pi\dfrac{t}{t_m}\right) & 0 \le t \le t_m \\ \ddot{q}_2(t) = A_{21}\sin\left(\pi\dfrac{t-t_m}{t_f-t_m}\right) & t_m \le t \le t_f \end{cases} \tag{3.45}$$

BUPT

care asigură accelerația nulă la $t = 0$, $t = t_m$ şi $t = t_f$.

Integrând relația (3.45), se obțin funcțiile de interpolare pentru viteză şi poziție:

$$\begin{cases} \dot{q}_1(t) = -A_{11}\dfrac{t_m}{\pi}\cos\left(\pi\dfrac{t}{t_m}\right) + A_{12} & 0 \le t \le t_m \\ \dot{q}_2(t) = -A_{21}\dfrac{t_f - t_m}{\pi}\cos\left(\pi\dfrac{t - t_m}{t_f - t_m}\right) + A_{22} & t_m \le t \le t_f \end{cases} \tag{3.46}$$

$$\begin{cases} q_1(t) = -A_{11}\left(\dfrac{t_m}{\pi}\right)^2 \sin\left(\pi\dfrac{t}{t_m}\right) + A_{12}t + A_{13} & 0 \le t \le t_m \\ q_2(t) = -A_{21}\left(\dfrac{t_f - t_m}{\pi}\right)^2 \sin\left(\pi\dfrac{t - t_m}{t_f - t_m}\right) + A_{22}(t - t_m) + A_{23} & t_m \le t \le t_f \end{cases} \tag{3.47}$$

Condițiile la limită şi de continuitate:

$$\begin{aligned} &q_1(0) = q(0) && \dot{q}_1(0) = \dot{q}(0) \\ &q_2(t_f) = q(t_f) && \dot{q}_2(t_f) = \dot{q}(t_f) \\ &q_1(t_m) = q_2(t_m) && \dot{q}_1(t_m) = \dot{q}_2(t_m) \qquad \ddot{q}_1(t_m) = \ddot{q}_2(t_m) \end{aligned} \tag{3.48}$$

sunt folosite pentru determinarea parametrilor necunoscuți din relațiile (3.45) - (3.47), în cele două cazuri $\dot{q}(0) \neq \dot{q}(t_f)$ şi $\dot{q}(0) = \dot{q}(t_f)$. Pentru $\dot{q}(0) \neq \dot{q}(t_f)$, rezultă:

$$\begin{aligned} &A_{11} = \pi t_m \frac{1}{2t_f}\frac{\dot{q}(t_f) - \dot{q}(0)}{2t_m - t_f} && A_{12} = \dot{q}(0) + A_{11}\frac{t_m}{\pi} && A_{13} = q(0) \\ &A_{21} = -\pi\frac{t_f - t_m}{2t_f}\frac{\dot{q}(t_f) - \dot{q}(0)}{2t_m - t_f} && A_{22} = \dot{q}(t_f) - A_{21}\frac{t_f - t_m}{\pi} && A_{23} = A_{12}t_m + A_{13} \end{aligned} \tag{3.49}$$

Timpul t_m este soluția unei ecuații de gradul doi $\left(a\cdot(t_m)^2 + b\cdot t_m + c = 0\right)$, unde:

$$\begin{aligned} a &= -\frac{\dot{q}(t_f) - \dot{q}(0)}{2} \\ b &= 2t_f\dot{q}(t_f) - \frac{t_f}{2}\left[\dot{q}(t_f) - \dot{q}(0)\right] - 2\left[q(t_f) - q(0)\right] \\ c &= t_f\left\{\frac{t_f}{2}\left[\dot{q}(t_f) - \dot{q}(0)\right] - t_f\dot{q}(t_f) + \left[q(t_f) - q(0)\right]\right\} \end{aligned} \tag{3.50}$$

Pentru $\dot{q}(0) = \dot{q}(t_f)$ relațiile de calcul al coeficienților A_{11}, A_{21}, t_m se modifică astfel:

$$A_{11} = -\frac{\pi}{t_f}\left\{\left[\dot{q}(0) + \dot{q}(t_f)\right] - \frac{2}{t_f}\left[q(t_f) - q(0)\right]\right\}; \quad A_{21} = -A_{11}; \quad t_m = t_f/2 \tag{3.51}$$

BUPT

3.2.3. FUNCŢIA DE INTERPOLARE SINUSOIDALĂ PONDERATĂ ÎN TIMP

Pentru interpolarea acceleraţiei se consideră produsul unei funcţii sinusoidale cu una liniară [D7]. Se disting două cazuri posibile. În primul caz, funcţia este nulă la t = 0, datorită componentei sinusoidale, iar în al doilea caz, cele două condiţii sunt inversate.

Anularea acceleraţiei la pornire datorită componentei sinusoidale

$$\begin{aligned}
\ddot{q}(t) &= -A_1\left(\frac{\omega}{t_f}\right)^2\left(1-\frac{t}{t_f}\right)\sin\left(\omega\frac{t}{t_f}\right) && t\in[0,\ t_f]\\
\dot{q}(t) &= \frac{A_2}{t_f}+A_1\left[\omega\frac{t}{t_f}\left(1-\frac{t}{t_f}\right)\cos\left(\omega\frac{t}{t_f}\right)+\frac{1}{t_f}\sin\left(\omega\frac{t}{t_f}\right)\right] && t\in[0,\ t_f] \qquad (3.52)\\
q(t) &= A_3+A_2\left(\frac{t}{t_f}\right)+A_1\left[\left(1-\frac{t}{t_f}\right)\sin\left(\omega\frac{t}{t_f}\right)-\frac{2}{\omega}\cos\left(\omega\frac{t}{t_f}\right)\right] && t\in[0,\ t_f]
\end{aligned}$$

Folosind condiţiile la limită: $q(0)$, $q(t_f)$, $\dot{q}(0)$, $\dot{q}(t_f)$, pentru care se presupun îndeplinite restricţiile $\dot{q}(0)\geq 0$, $\dot{q}(t_f)\geq 0$, $\dot{q}(0)\neq\dot{q}(t_f)$, $q(0) < q(t_f)$, rezultă coeficienţii A_1, A_2 şi A_3:

$$A_1=\frac{t_f\dot{q}(t_f)-q(t_f)+q(0)}{\sin\omega-\frac{2}{\omega}(1-\cos\omega)} \qquad A_2=t_f\dot{q}(0)-A_1\omega \qquad A_3=q(0)+2\frac{A_1}{\omega} \qquad (3.53)$$

Anularea acceleraţiei la pornire datorită componentei liniare:

$$\begin{aligned}
\ddot{q}(t) &= -A_1\left(\frac{\omega}{t_f}\right)^2\frac{t}{t_f}\sin\left[\frac{\omega}{t_f}(t_f-t)\right] && t\in[0,\ t_f]\\
\dot{q}(t) &= \frac{A_2}{t_f}-A_1\left\{\omega\frac{t}{t_f^{\,2}}\cos\left[\frac{\omega}{t_f}(t_f-t)\right]+\frac{1}{t_f}\sin\left[\frac{\omega}{t_f}(t_f-t)\right]\right\} && t\in[0,\ t_f] \qquad (3.54)\\
q(t) &= A_3+A_2\frac{t}{t_f}+A_1\left\{\frac{t}{t_f}\sin\left[\frac{\omega}{t_f}(t_f-t)\right]-\frac{2}{\omega}\cos\left[\frac{\omega}{t_f}(t_f-t)\right]\right\} && t\in[0,\ t_f]
\end{aligned}$$

Punând aceleaşi condiţii ca şi în primul caz, se obţin coeficienţii A_1, A_2, A_3:

$$A_1=\frac{t_f\dot{q}(0)-q(t_f)+q(0)}{\frac{2}{\omega}(1-\cos\omega)-\sin\omega} \qquad A_2=t_f\dot{q}(0)-A_1\sin\omega \qquad A_3=q(0)+2\frac{A_1}{\omega}\cos\omega \qquad (3.55)$$

Dintre funcţiile de interpolare prezentate în acest paragraf, cea mai folosită este funcţia polinomială, datorită volumului redus de calcule şi gamei largi de performanţe asigurate prin variaţia parametrului β.

3.2.4. INTERPOLARE LINIARĂ ŞI TRANZIŢIE PARABOLICĂ

Această metodă poate fi utilizată pentru generarea traiectoriei atât în spaţiul coordonatelor cuplelor, cât şi în spaţiul operaţional. Mişcarea este formată din segmente

BUPT

parcurse cu viteză constantă, corespunzătoare interpolărilor liniare, legate prin tranziţii cu acceleraţie constantă.

Se consideră, pentru o cuplă cinematică oarecare *j*, o mişcare între *punctele* q_{k-1}^j şi q_k^j (fig.3.9), presupunând că timpul *t* este egal cu zero în momentul trecerii prin vecinătatea *punctului* q_{k-1}^j . Pornind de la aceste ipoteze, mişcarea se generează astfel:

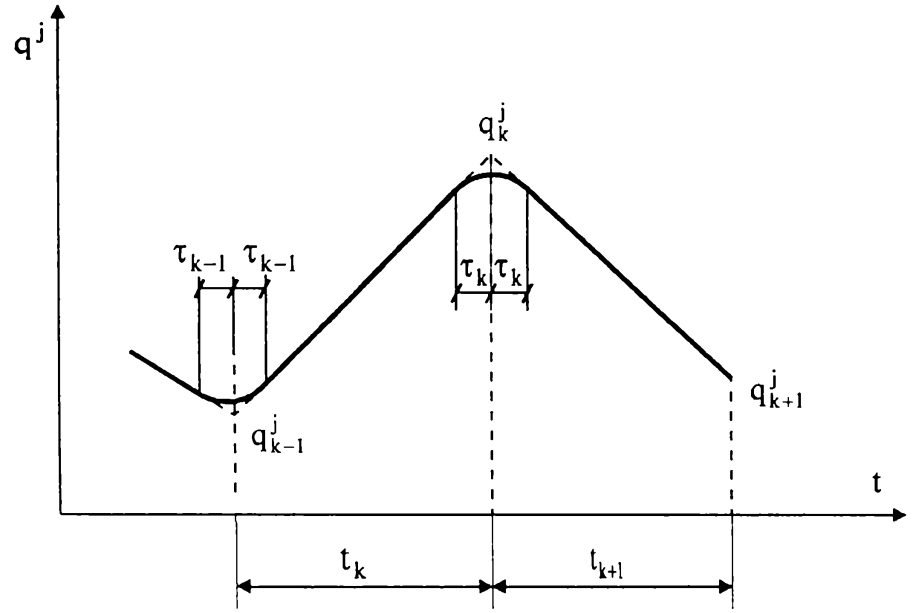


Fig.3.9. Interpolare liniară şi tranziţie parabolică în *punctul* intermediar

• Pentru porţiunea liniară se procedează la fel ca în paragraful 2.4 (mişcarea punct cu punct, în spaţiul operaţional). Mişcarea între momentele τ_{k-1} şi t_k-τ_k este definită prin viteza constantă $\dot{q}_k^j = \frac{D_k^j}{t_k}$ cu $D_k^j = q_k^j - q_{k-1}^j$. Poziţia cuplei cinematice *j* evoluează astfel după legea: $q_k^j = t \cdot \dot{q}_k^j$.

• Tranziţia în poziţia q_k^j a traiectoriei cuplei q^j, presupunând t = 0 în momentul trecerii prin vecinătatea *punctului* q_k^j , începe cu τ_k secunde înaintea acestui moment şi continuă τ_k secunde după acest moment. Ţinând cont de condiţiile de continuitate în poziţie şi în viteză, la extremităţile fazei de tranziţie, precum şi de simetria acestei faze, ecuaţia de mişcare rezultă:

$$q^j(t) = q_k^j - \frac{D_k^j}{4\tau_k t_k}(t-\tau_k)^2 + \frac{D_{k+1}^j}{4\tau_k t_{k+1}}(t+\tau_k)^2 \qquad (3.56)$$

Acceleraţia este constantă şi are expresia:

$$\ddot{q}^j(t) = \frac{1}{2\tau_k}\left(\frac{D_{k+1}^j}{t_{k+1}} - \frac{D_k^j}{t_k}\right) = \frac{1}{2\tau_k}\left(\dot{q}_{k+1}^j - \dot{q}_k^j\right) \qquad (3.57)$$

Deoarece în relaţiile precedente s-a făcut ipoteza existenţei unei porţiuni liniare între două *puncte* intermediare succesive, rezultă că $t_k \geq \tau_{k-1} + \tau_k$.

Analizând relaţia (3.56) se constată că utilizarea unei acceleraţii mari reduce durata tranziţiei şi astfel scade abaterea *punctului* intermediar de la traiectorie, traiectoria

BUPT

trecând prin vecinătatea *punctelor* intermediare. Dacă se doreşte ca traiectoria să treacă riguros prin unul din *punctele* intermediare q_k, este suficient să se adauge după acest *punct*, un *punct* suplimentar, de aceeaşi valoare q_k (fig.3.10). În acelaşi mod se poate impune o viteză nulă în *punctul* iniţial şi în cel final: se adaugă două *puncte* suplimentare având aceleaşi valori ca şi q_0, respectiv q_m (fig.3.11).

În acest caz, mişcarea este continuă în poziţie şi în viteză. Pentru a avea o mişcare continuă şi în acceleraţie, se poate utiliza o funcţie de gradul cinci pentru tranziţie, acceleraţiile iniţiale şi finale fiind astfel egale cu zero.

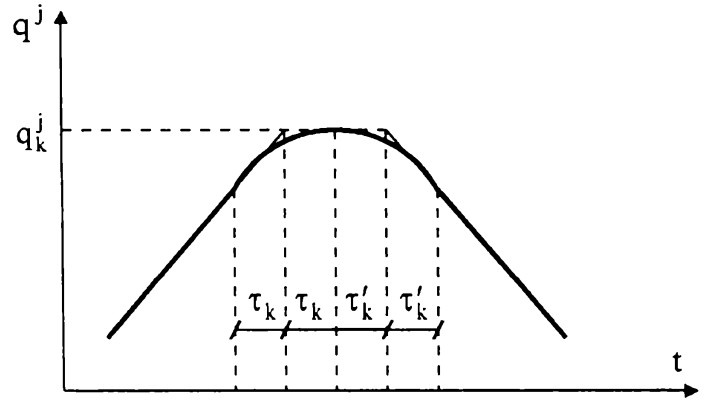


Fig.3.10. Trecerea printr-un *punct* intermediar

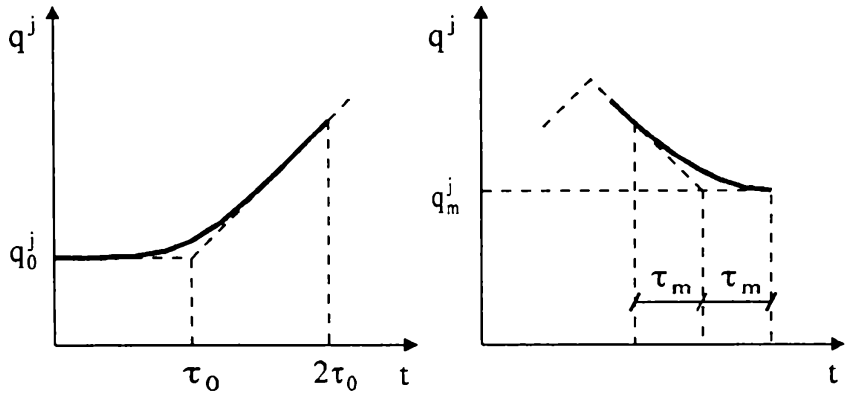


Fig.3.11. Definirea mişcării în *punctele* iniţial şi final

3.2.5. UTILIZAREA POLINOAMELOR DE GRADUL TREI

Această metodă se pretează de asemenea, atât în cazul generării mişcării în spaţiul coordonatelor cuplelor, cât şi în spaţiul operaţional. Între două *puncte* q_{k-1} şi q_k, mişcarea este exprimată printr-un polinom de gradul trei: $q^j(t) = C^j_{0,k} + C^j_{1,k}t + C^j_{2,k}t^2 + C^j_{3,k}t^3$. Spre deosebire de metoda prezentată în paragraful anterior, această metodă generează traiectorii care trec exact prin *punctele* intermediare. Se consideră trei cazuri, după cum vitezele în aceste *puncte* sunt:

- impuse;
- calculate în manieră locală, fără să se cunoască a priori întreaga traiectorie;

BUPT

• calculate în manieră globală pentru a asigura continuitatea în viteză şi acceleraţie de-a lungul traiectoriei. Metoda corespunzătoare este cunoscută sub numele de *metoda funcţiilor spline cubice* şi este tratată detaliat în capitolul următor.

Dacă vitezele în punctele intermediare sunt impuse, pentru a parcurge o porţiune de traiectorie între două *puncte* q_{k-1}, q_k, se determină un polinom de gradul trei, ai cărui coeficienţi polinomiali, pentru fiecare cuplă cinematică *j*, se deduc uşor din condiţiile iniţiale şi finale ce constituie datele problemei.

Pentru calculul vitezelor în manieră locală se poate proceda în două moduri:

a) Se alege pentru viteză panta medie în intervalele $[t_k, t_{k+1}]$ (fig.3.12.a), ceea ce se poate scrie: $\dot{q}_k^j = \frac{q_{k+1}^j - q_{k-1}^j}{t_k + t_{k+1}}$. Atunci când segmentele $[q_{k-1}, q_k]$ şi $[q_k, q_{k+1}]$ au lungimi foarte diferite (fig.3.12.b), panta medie nu convine, deoarece se poate ajunge la viteze prea mari. Un dezavantaj al utilizării acceleraţiilor importante constă în faptul că se pot produce depăşiri şi viteza nu mai este uniformă de-a lungul unui segment. Pentru exemplul considerat, $\dot{q}_k^j$ este vecin cu $\frac{q_k^j - q_{k-1}^j}{t_k}$, chiar dacă este preferată o valoare $\frac{q_{k+1}^j - q_k^j}{t_{k+1}}$.

b) Se alege pentru viteză panta medie ponderată: $\dot{q}_k^j = \alpha \cdot \frac{q_k^j - q_{k-1}^j}{t_k} + (1-\alpha) \cdot \frac{q_{k+1}^j - q_k^j}{t_{k+1}}$, cu $\alpha = \frac{t_{k+1}}{t_k + t_{k+1}}$. Se obţine astfel o traiectorie mai netedă, dar inconvenientele menţionate mai sus nu sunt complet evitate.

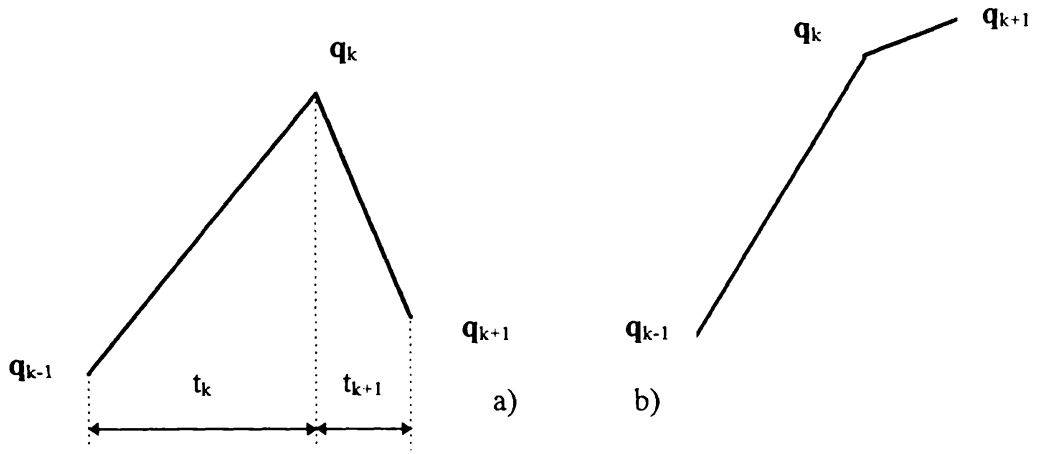


Fig.3.12. Calculul vitezelor în manieră locală

BUPT

3.3. COORDONAREA CUPLELOR CINEMATICE ALE UNUI ROBOT

Dacă pentru a realiza trecerea din situarea $\mathbf{X}_k$ a efectorului final la situarea $\mathbf{X}_{k+1}$, fiecare mecanism de acționare este solicitat prin legea sa optimă pentru viteză, mișcările cuplelor cinematice nu se vor termina în același timp. Pentru a asigura sincronizarea în *punctul* de oprire, trebuie ca timpul t_f^j necesar fiecărei cuple cinematice pentru parcurgerea distanței D^j să fie același. Pentru a sincroniza mișcarea cuplelor cinematice, se studiază două metode de coordonare a acestora [D14], [L15]:

•în prima metodă, timpul global optimal $t_f = \text{Max}(t_f^j)$ se conservă. Mișcarea fiecărei cuple cinematice este asemănătoare legii sale optime dar este etalată în timp pentru a avea aceeași durată ca cea a axei care impune restricțiile.

•a doua metodă asigură duratele de accelerare și frânare egale pentru toate cuplele cinematice. Legile de mișcare sincronizate sunt homotetice. Din acest motiv, acest tip de lege de mișcare este cel mai des implementat controlerelor roboților industriali.

Pentru determinarea axei care impune restricții, procedura este următoarea:

•pentru j = 1, ..., n, se calculează $\dfrac{\left(K_v^j\right)^2}{K_a^j}$ și se compară cu D^j;

•se calculează t_f^j cu una din relațiile de mai jos, în funcție de tipul de interpolare ales (profil trapezoidal sau profil triunghiular de viteză):

$$t_f^j = \frac{D^j}{K_v^j} + \frac{K_v^j}{K_a^j} \qquad \text{sau} \qquad t_f^j = 2\sqrt{\frac{D^j}{K_a^j}} \tag{3.58}$$

•se caută *k* astfel încât:

$$t_f^k = \text{Max}\left(t_f^j, j \in [1,n]\right) \tag{3.59}$$

Axa care impune restricții este axa cuplei cinematice *k*, iar timpul t_f^k este timpul comun pentru efectuarea deplasărilor D^j și se notează cu t_f.

3.3.1. CALCULUL TIMPULUI MINIM ÎN FUNCȚIE DE TIPUL DE INTERPOLARE FOLOSIT

Dacă timpul de parcurgere a traiectoriei t_f nu este specificat, ceea ce se întâmplă de obicei și se caută timpul minim pentru trecerea de la configurația $\mathbf{q}_i$ la configurația $\mathbf{q}_f$, respectând restricțiile de viteză și accelerație, se calculează timpul minim pentru fiecare cuplă cinematică separat și apoi se face coordonarea cuplelor cinematice pentru un timp comun.

Timpul minim pentru o cuplă cinematică *j* trebuie să satisfacă viteza și/sau accelerația. Pornind de la expresiile vitezei și accelerației maxime obținute prin aplicarea diferitelor metode de interpolare se deduce timpul minim t_f^j care este necesar unei cuple cinematice oarecare *j* pentru a realiza o deplasare dată.

O coordonare a cuplelor cinematice constă în sincronizarea mișcărilor astfel încât toate cuplele cinematice să pornească și să termine mișcarea în același timp. Durata deplasării unei cuple cinematice este funcție atât de distanța D^j de parcurs cât și de caracteristicile de viteză și accelerație K_v^j și K_a^j; deci ea nu este identică pentru toate

BUPT

cuplele cinematice. Timpul global minim necesar pentru trecerea de la configurația $\mathbf{q}_i$ la configurația $\mathbf{q}_f$ este timpul dat de cupla cinematică supusă restricțiilor, cuplă pentru care timpul minim este cel mai mare. Din aceasta se deduce că:

$$t_f = \text{Max}\left(t_f^1, \ldots, t_f^n\right) \tag{3.60}$$

Pentru generarea mişcării unui robot între două *puncte* de oprire, în spațiul coordonatelor cuplelor cinematice, cele mai utilizate tipuri de interpolări sunt: polinomul de gradul trei, polinomul de gradul cinci, profilul triunghiular de viteză, profilul trapezoidal de viteză. În Tabelul 3.1 se dau expresiile cu care se poate calcula timpul minim pentru o cuplă cinematică oarecare în funcție de tipul de interpolare ales.

Tabelul 3.1. Timpul minim pentru diferite tipuri de interpolări, pentru o cuplă cinematică oarecare *j*

Nr.	Funcția de interpolare	Timpul minim
1.	Interpolare liniară	$t_f^j = \dfrac{\lvert D^j \rvert}{K_v^j}$
2.	Polinom de gradul trei	$t_f^j = \text{Max}\left[\dfrac{3\cdot\lvert D^j\rvert}{2K_v^j}, \sqrt{\dfrac{6\cdot\lvert D^j\rvert}{K_a^j}}\right]$
3.	Polinom de gradul cinci	$t_f^j = \text{Max}\left[\dfrac{15\cdot\lvert D^j\rvert}{8K_v^j}, \sqrt{\dfrac{10\cdot\lvert D^j\rvert}{\sqrt{3}K_a^j}}\right]$
4.	Profil triunghiular de viteză (Bang - Bang)	$t_f^j = \text{Max}\left[\dfrac{2\cdot\lvert D^j\rvert}{K_v^j}, 2\sqrt{\dfrac{\lvert D^j\rvert}{K_a^j}}\right]$
5.	Profil trapezoidal de viteză	$t_f^j = \dfrac{K_v^j}{K_a^j} + \dfrac{D^j}{K_v^j}$
6.	Profil trapezoidal neted	$t_f'^j = \dfrac{D^j}{K_v^j} + \dfrac{K_v^j}{K_a^j} + \dfrac{K_a^j}{a^j}$
7.	Profil triunghiular neted	$t_f'^j = \dfrac{K_a^j}{a^j} + \sqrt{\left(\dfrac{K_a^j}{a^j}\right)^2 + 4\left(\dfrac{D^j}{K_a^j}\right)^2}$

3.3.2. METODA TIMPULUI MINIM PENTRU COORDONAREA CUPLELOR CINEMATICE

Fie cupla cinematică *k*, care impune mişcării o durată $t_f = t_f^k$. Cupla cinematică *j* este sincronizată în raport cu legea sa proprie prin creşterea duratei palierului de viteză, în

BUPT

timp ce duratele fazelor de accelerație și de decelerație rămân identice cu cele ale legii proprii (fig.3.13).

Pentru cuplele cinematice conduse după o lege trapezoidală de viteză se conservă timpii de accelerare și de frânare la valoarea τ^j. Accelerația este redusă în aceeași proporție ca și palierul de viteză, astfel încât $t_f \geq t_f^j$ se obține prin alungirea palierului de viteză. Noul profil este reprezentat în fig.3.13. Astfel, legea sincronizată a cuplei cinematice *j* se obține realizând o omotetie a unui raport λ^j la legea proprie. Din relația (3.58) se poate scrie:

$$t_f = \tau^j + \frac{|D^j|}{\lambda^j K_v^j} \tag{3.61}$$

din care se deduce valoarea coeficientului de reducere a palierului de viteză și de accelerație ($\lambda^j = \lambda_v^j = \lambda_a^j$):

$$\lambda^j = \frac{|D^j|}{K_v{}^j \cdot (t_f - \tau^j)} = \frac{t_f^j - \tau^j}{t_f - \tau^j} \tag{3.62}$$

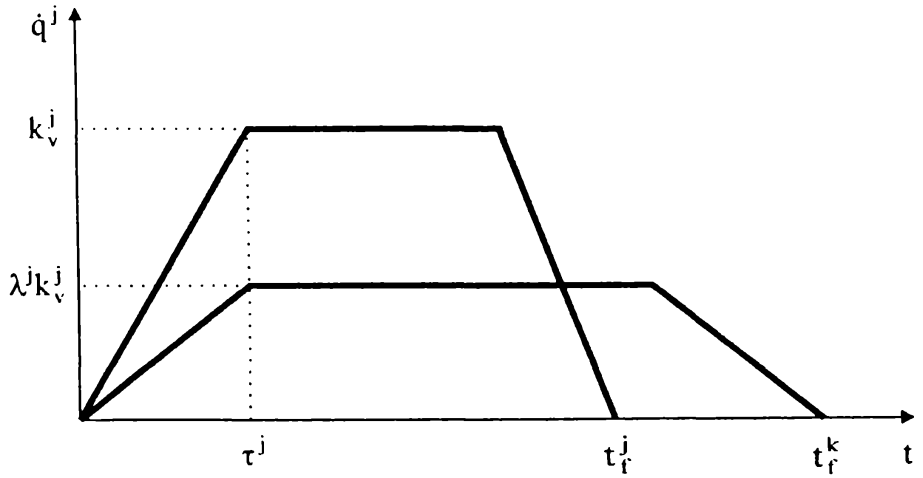


Fig.3.13. Coordonarea cuplelor cinematice prin metoda timpului minim

În cazul profilul triunghiular de viteză, pentru a prelungi durata de la t_f^j la t_f, se ajustează accelerația. Pentru noua lege triunghiulară, din relația:

$$t_f = 2\sqrt{\frac{D^j}{\lambda_a^j K_a^j}} \tag{3.63}$$

se deduce valoarea coeficientului de reducere a accelerației:

$$\lambda_a^j = \frac{a_{max}^j}{K_a^j}; \quad \lambda_a^j = \left(\frac{t_f^j}{t_f}\right)^2 = \left(\lambda_t^j\right)^{-2} \tag{3.64}$$

unde $\lambda_t^j = \frac{t_f}{t_f^j}$ este factorul de amplificare a timpului de parcurs pentru cupla cinematică *j*.

BUPT

Algoritmul metodei timpului minim

Pentru coordonarea cuplelor cinematice prin metoda timpului minim între două *puncte* de oprire se poate aplica următorul algoritm [D22]:

- se introduce (sau se calculează din modelul geometric invers) vectorul coordonatelor cuplelor la momentul inițial t_i, respectiv la momentul final t_f: $\mathbf{q}_i$ și $\mathbf{q}_f$;
- se introduce vectorul vitezelor maxime $\mathbf{K}_v$ și vectorul accelerațiilor maxime ale cuplelor $\mathbf{K}_a$;
- se calculează vectorul deplasărilor $\mathbf{D}= \mathbf{q}_f - \mathbf{q}_i$;
- se stabilește tipul interpolării între cele două puncte de oprire;
- se calculează timpul minim t_f^j pentru fiecare cuplă cinematică, conform Tabelului 3.1 [T9]; se determină timpul global minim cu care se realizează coordonarea cuplelor cinematice: $t_f = \text{Max}\,[t_f^j, j=1,...,n]$;
- se introduc timpii de accelerare/frânare τ^j care se conservă;
- se calculează valoarea coeficientului de reducere a palierului de viteză și de accelerație λ^j;
- se calculează vitezele, respectiv accelerațiile maxime ale cuplelor în mișcarea coordonată: $v_{max}^j = \lambda^j K_v^j$ și $a_{max}^j = \lambda^j K_a^j$.

În mișcarea cu viteză redusă, timpul optim t_f poate conduce la mișcări prea rapide pentru sarcina ce trebuie efectuată. În acest caz, trebuie să se intervină în comandă pentru a crește t_f. Mișcarea, în ansamblu, va fi mai lentă, dar se va conserva sincronizarea. Dacă în cazul profilului trapezoidal se reduce accelerația cu pătratul raportului de reducere a palierului de viteză, atunci se poate arăta că λ_v^j este invers proporțional cu factorul de amplificare a timpului de parcurs λ_t^j: $\lambda_v^j = \frac{1}{\lambda_t^j}$, $\lambda_a^j = \frac{1}{\left(\lambda_t^j\right)^2}$, timpul de accelerare sau de frânare fiind egal cu $\lambda_t^j \tau^j$.

Algoritmul metodei prezentate poate fi aplicat, cu mici modificări, și în cazul coordonării cuplelor cinematice între *punctele* de trecere.

3.3.3. COORDONAREA CUPLELOR ÎNTRE *PUNCTELE* DE TRECERE

Punctele de trecere sunt *punctele* din spațiul operațional, prin care efectorul final al robotului trece fără să se oprească. Aceste *puncte* servesc la evitarea coliziunii cu obstacolele din spațiul de lucru al robotului, reducând totodată durata mișcării (față de cazul când efectorul final al robotului s-ar opri în fiecare *punct*. Viteza dorită într-un *punct* de trecere este exprimată sub forma vitezei în spațiul coordonatelor cuplelor cinematice, printr-un anumit procent din viteza maximă K_v^j a cuplei cinematice *j*. Între două *puncte* de trecere corespunzătoare capacității maxime a cuplei *j*, profilul trapezoidal de viteză este conform celui prezentat în figura 3.14, curba (a).

Se notează v_i^j și v_f^j viteza la începutul și la sfârșitul mișcării între două *puncte* de trecere și cu τ_a^j și τ_f^j timpii de accelerare și de frânare. Se prezintă în continuare, sumar, un procedeu de sincronizare a mișcărilor. Din figura 3.14 se scrie relația care exprimă aria de sub curba (a):

BUPT

$$D^j = t_f^j K_v^j - \frac{1}{2}\tau_a^j\left(K_v^j - v_i^j\right) - \frac{1}{2}\tau_f^j\left(K_v^j - v_f^j\right) \tag{3.65}$$

de unde se calculează t_f^j.

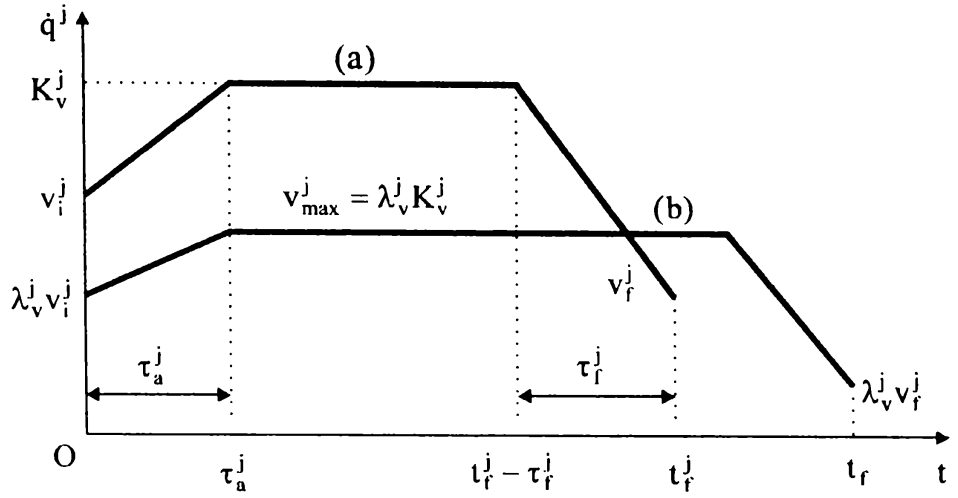


Fig. 3.14. Coordonarea cuplei *j*, în cazul profilului trapezoidal, între două *puncte* de trecere

Timpul optim t_f se calculează: $t_f = \text{Max}\left(t_f^j,\ j \in [1,n]\right)$. În continuare este necesar să se ajusteze profilul de viteză pentru satisfacerea timpului de parcurs t_f. Fie $\lambda_v^j = \frac{v_{max}^j}{K_v^j}$ coeficientul de reducere a palierului de viteză. Acest coeficient de reducere se aplică vitezelor v_i^j şi v_f^j, păstrând aceiaşi timpi de accelerare τ_a^j şi de frânare τ_f^j. Astfel, aria de sub curba (b), care reprezintă noul profil de viteză, se poate scrie:

$$D^j = t_f \lambda_v^j K_v^j - \frac{1}{2}\tau_a^j \lambda_v^j\left(K_v^j - v_i^j\right) - \frac{1}{2}\tau_f^j \lambda_v^j\left(K_v^j - v_f^j\right) \tag{3.66}$$

Adăugând expresia lui D^j de mai sus la precedenta înmulţită cu $(-\lambda_v^j)$, rezultă:

$$D^j - \lambda_v^j D^j = t_f \lambda_v^j K_v^j - t_f^j \lambda_v^j K_v^j \tag{3.67}$$

Expresia coeficientului de reducere a palierului de viteză λ_v^j este:

$$\lambda_v^j = \frac{D^j}{D^j + K_v^j\left(t_f - t_f^j\right)} \tag{3.68}$$

Astfel, se poate să se determine timpul ciclului de lucru al manipulatorului, care constă într-o succesiune de deplasări cu *puncte* de oprire şi/sau *puncte* de trecere. Determinarea timpului optim pentru un ciclu de mişcare se poate realiza reducând timpii secvenţelor de mişcare a efectorului final al robotului. Această reducere poate fi obţinută printr-o alegere judicioasă a *punctelor* de oprire şi a *punctelor* de trecere, în măsura în care există o anumită libertate de alegere a acestor *puncte*. Stabilirea *punctelor* de oprire şi a *punctelor* de trecere se face din analiza sarcinii robotului şi a spaţiului de lucru al acestuia, aplicând una din metodele de planificare a mişcării roboţilor cunoscute [D41], [L5].

BUPT

3.3.4. COORDONAREA CUPLELOR CINEMATICE ALE UNUI ROBOT PRIN METODA PROPORȚIONALĂ

Metoda propusă în paragraful precedent asigură duratele de accelerare şi de frânare egale pentru toate cuplele cinematice, legile de mişcare sincronizate fiind homotetice. În acest capitol se propune un algoritm pentru sincronizarea mişcării cuplelor cinematice prin metoda proporţională şi se prezintă aplicarea acestuia pentru cazul profilului trapezoidal de viteză. Metoda impune satisfacerea restricţiilor de viteză şi de acceleratie pentru fiecare cuplă cinematică şi determină o durată de accelerare optimă, aceeaşi pentru toate cuplele cinematice.

Conform acestei metode, legile de viteză ale diferitelor cuple cinematice sunt homotetice şi comportă faze de accelerare şi de frânare de aceleaşi durate τ (fig.3.15). Din acestea se deduce relaţia [L15]:

$$\dot{q}^j(t) = \lambda^j \cdot \dot{q}^i(t) \qquad j = 1, ..., n \tag{3.69}$$

Din punct de vedere tradiţional, în metodele proporţionale, legile de viteză ale cuplelor cinematice sunt deduse prin homotetie din legea de viteză a cuplei cinematice *k* care impune restricţia. În anumite cazuri nu este posibil să se respecte restricţiile de acceleraţie sau cele de viteză la una sau mai multe cuple cinematice. Trebuie astfel să se introducă un interval de timp Δt şi să se recalculeze raportul de homotetie ţinând cont de noua durată a mişcării $t_f^k + \Delta t$. Metoda poate deci să ceară mai multe iteraţii şi să nu se obţină timpul minim.

Soluţia prezentată asigură un timp t_f minim şi respectarea restricţiilor cinematice în toate cazurile din figură. Soluţia presupune calculul unei durate τ a fazei de accelerare diferită aprioric de τ^j optim al legilor proprii fiecărei cuple cinematice.

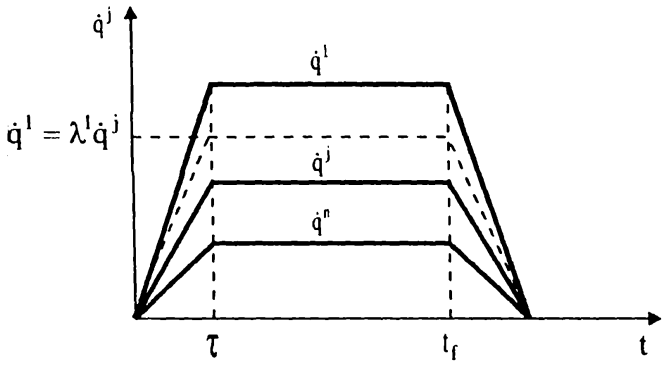


Fig.3.15. Coordonarea cuplelor cinematice prin metoda proporţională

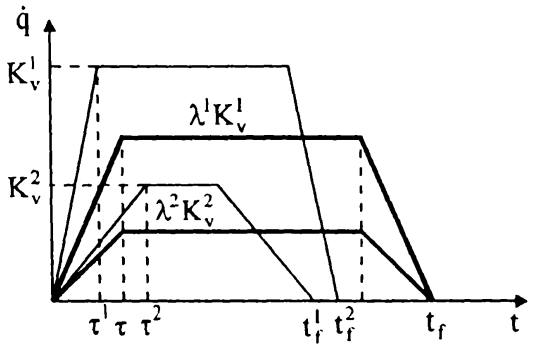


Fig.3.16. Coordonarea cuplelor cinematice.

Metoda proporţională cu evitarea depăşirii restricţiilor

Fie $\lambda^j k_v^j$ viteza cuplei cinematice *j* şi fie $\nu^j k_a^j$ acceleraţia corespunzătoare care produce mişcarea. Se doreşte să se calculeze valoarea lui τ care asigură un timp t_f minim. Se studiază problema pentru două cuple cinematice (fig.3.16). Pentru profilul trapezoidal de viteză, expresia timpului minim necesar parcurgerii unei distanţe D^j, este:

BUPT

$$t_f^j = \frac{k_v^j}{k_a^j} + \frac{D^j}{k_v^j} = \tau^j + \frac{D^j}{k_v^j} \tag{3.70}$$

Din relația (3.70), timpul minim de parcurgere al fiecărei cuple cinematice, calculat în mod separat se scrie:

$$t_f^1 = \tau^1 + \frac{|D^1|}{K_v^1}; \qquad t_f^2 = \tau^2 + \frac{|D^2|}{K_v^2} \tag{3.71}$$

Legea sincronizată a mişcării impune:

$$t_f = \frac{\lambda^1 K_v^1}{\nu^1 K_a^1} + \frac{|D^1|}{\lambda^1 K_v^1} = \frac{\lambda^2 K_v^2}{\nu^2 K_a^2} + \frac{|D^2|}{\lambda^2 K_v^2} \tag{3.72}$$

cu $t_f \geq \text{Max}(t_f^1, t_f^2)$. Pentru a calcula t_f minim trebuie să se determine λ^j şi ν^j. Din relația (3.72) se deduce:

$$\tau = \frac{\lambda^1 K_v^1}{\nu^1 K_a^1} = \frac{\lambda^2 K_v^2}{\nu^2 K_a^2}; \quad \lambda^2 = \lambda^1 \frac{K_v^1 |D^2|}{K_v^2 |D^1|}; \qquad \nu^2 = \nu^1 \frac{K_a^1 |D^2|}{K_a^2 |D^1|} \tag{3.73}$$

Pentru a satisface restricțiile de viteză trebuie îndeplinite condițiile:

$$\begin{cases} 0 \leq \lambda^1 \leq 1 \\ 0 \leq \lambda^2 \leq 1 \end{cases} \tag{3.74}$$

Utilizând expresia lui λ^2 din (3.73), ultima inegalitate devine:

$$0 \leq \lambda^1 \leq \frac{K_v^2 |D^1|}{K_v^1 |D^2|} \tag{3.75}$$

Un raționament analog, în cazul restricțiilor pentru accelerație, conduce la:

$$\begin{cases} 0 \leq \nu^1 \leq 1 \\ 0 \leq \nu^1 \leq \dfrac{K_a^2 |D^1|}{K_a^1 |D^2|} \end{cases} \tag{3.76}$$

Timpul t_f minim este obținut când λ^1 şi ν^1 au valori maxime şi satisfac simultan relațiile de mai jos:

$$\begin{cases} \lambda_{opt}^1 = \text{Min}\left[1, \dfrac{K_v^2 |D^1|}{K_v^1 |D^2|}\right] \\ \nu_{opt}^1 = \text{Min}\left[1, \dfrac{K_a^2 |D^1|}{K_a^1 |D^2|}\right] \end{cases} \tag{3.77}$$

BUPT

Durata corespunzătoare fazei de accelerare are valoarea:

$$\tau_{opt} = \frac{\lambda_{opt}^1 K_v^1}{\nu_{opt}^1 K_a^1} \tag{3.78}$$

Aceste relații sunt uşor generalizabile pentru *n* cuple cinematice,astfel:

$$\begin{cases} \lambda_{opt}^1 = \text{Min}\left[1, \dfrac{K_v^j |D^1|}{K_v^1 |D^j|}\right] \\ \nu_{opt}^1 = \text{Min}\left[1, \dfrac{K_a^j |D^1|}{K_a^1 |D^j|}\right] \end{cases} \qquad j = 2, \ldots, n \tag{3.79}$$

Dacă pentru o cuplă cinematică *j* dată, legea optimă nu atinge viteza maximă k_v^j, se înlocuieşte în funcțiile de mai sus termenul k_v^j prin viteza maximă efectivă atinsă (fig.3.17). Din condiția de existență a palierului de viteză $\left(|D^j| > \dfrac{(K_v^j)^2}{K_a^j}\right)$ rezultă că această situație se prezintă când:

$$\left(|D^j| < \frac{(K_v^j)^2}{K_a^j}\right) \tag{3.80}$$

Se deduce de aici că viteza maximă atinsă are valoarea:

$$K_v'^j = \sqrt{|D^j| \cdot K_a^j} \tag{3.81}$$

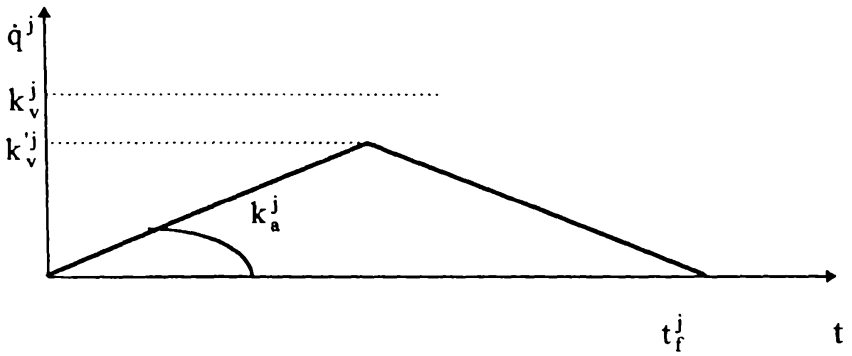


Fig.3.17. Coordonarea cuplelor cinematice.

Metoda proporțională fără depăşirea restricției. Cazul când viteza este saturată

BUPT

3.4. PROGRAM PENTRU GENERAREA MIŞCĂRII ROBOŢILOR INDUSTRIALI ÎN SPAŢIUL CUPLELOR CINEMATICE

În paragrafele precedente ale acestui capitol au fost abordate principalele aspecte ale generării mişcării în spaţiul coordonatelor cuplelor cinematice. Metodele prezentate au constituit baza elaborării unui program amplu pentru generarea mişcării.

Programul care realizează aceste generări de traiectorii a fost conceput ca o aplicaţie Windows, fiind scris într-un limbaj de nivel înalt şi anume în Borland Pascal. Programul este foarte uşor de utilizat şi prezintă un număr mare de posibilităţi de a alege legile de interpolare, punctele nodale şi restricţiile cinematice.

Generarea mişcării se face pe baza unor condiţii iniţiale care pot fi selectate în program prin intermediul meniurilor şi al dialogurilor. Algoritmul programului conţine trei părţi: iniţializarea, calculul propriu-zis şi afişarea (textuală şi grafică) a rezultatelor.

Programul permite utilizatorului selectarea metodei de generare a mişcării, dar mai ales încadrarea în datele concrete ale oricărei probleme. Astfel, programul constituie o sinteză în manieră originală a paragrafelor precedente. Selectarea metodei de generare se realizează ţinând cont de anumite aspecte referitoare la robot şi mişcare, prin activarea primului submeniu. Ecranul care apare la această selecţie este prezentat în figura 3.18.

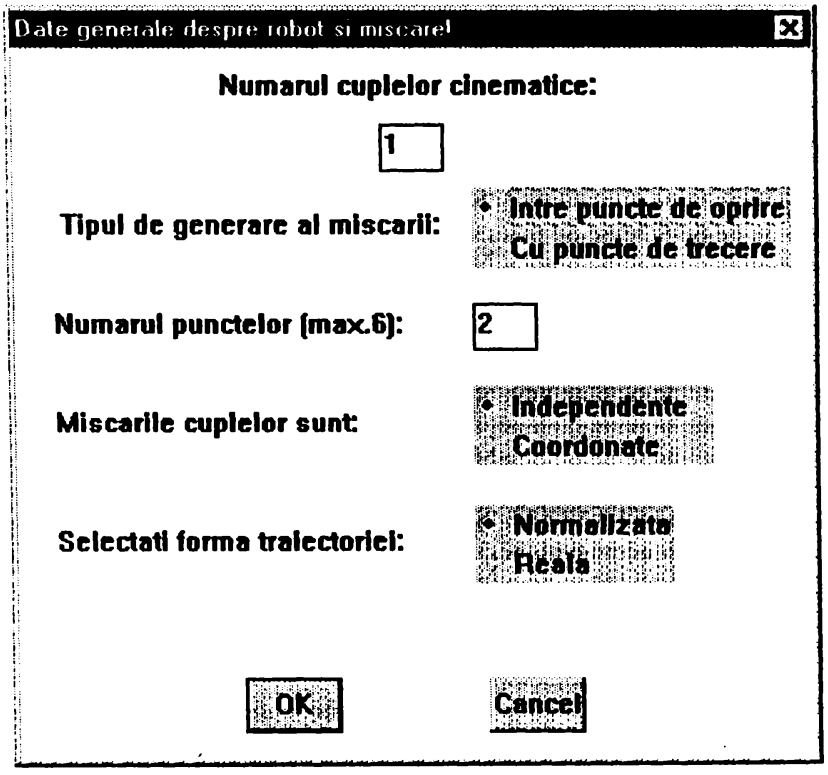


Fig.3.18. Date generale despre robot şi mişcare

Alegerea tipului de interpolare se realizează din submeniurile prezentate în fig.3.19 (mişcare între puncte de oprire) şi fig.3.20 (mişcare între puncte de trecere).

BUPT

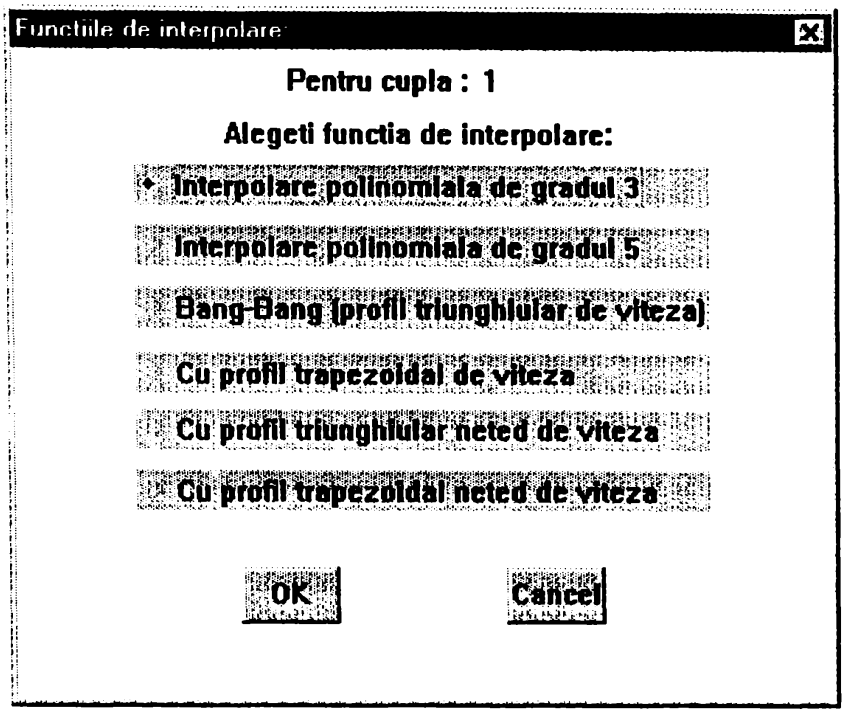


Fig.3.19. Setarea funcţiei de interpolare. Mişcare între *puncte* de oprire

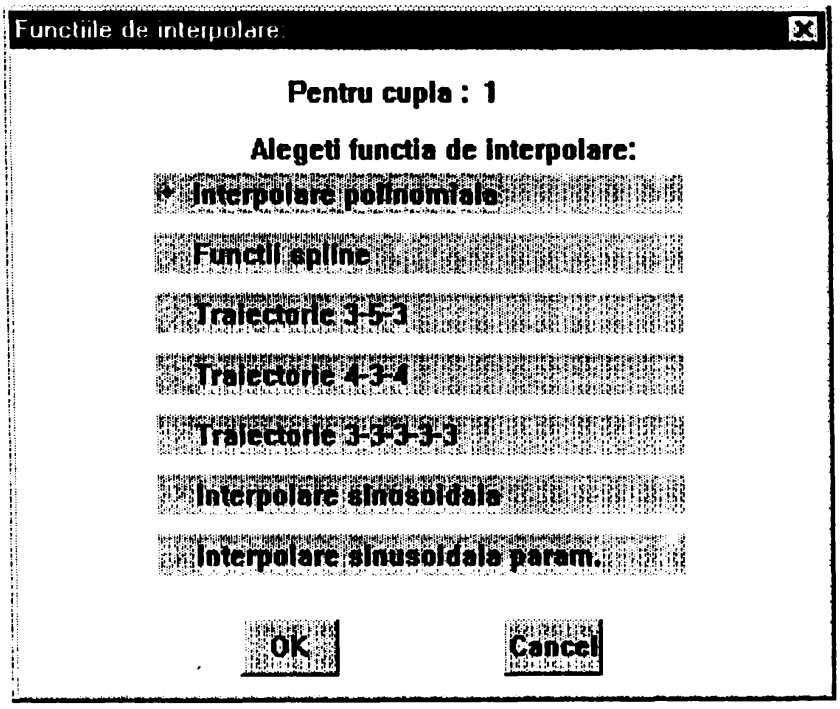


Fig.3.20. Setarea funcţiei de interpolare. Mişcare între *puncte* de trecere

La selectarea *punctelor* de interpolare se activează un dialog pentru specificarea, pentru fiecare cuplă a robotului, a coordonatelor acestora în *punctele* de interpolare. Dialogul dintre program şi utilizator pentru introducerea acestor date este redat în figura 3.21.

BUPT

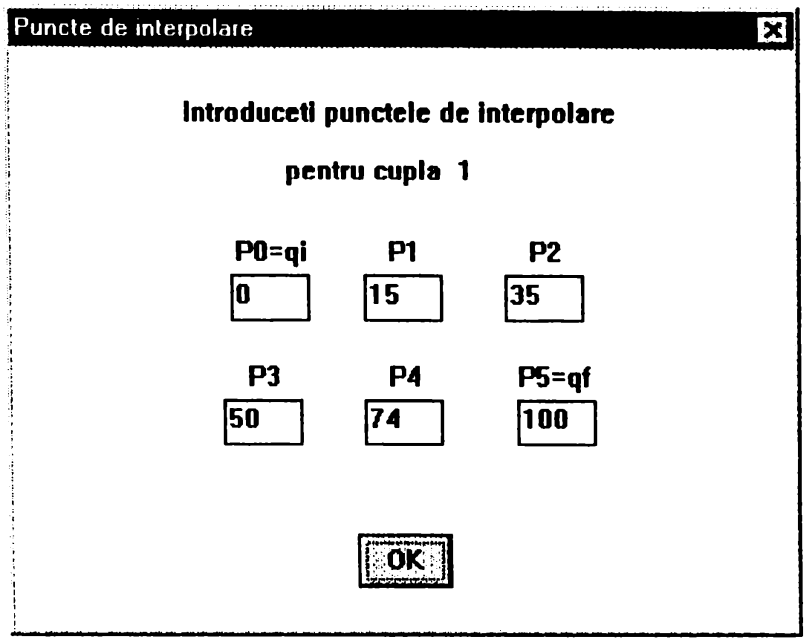


Fig.3.21. Introducerea coordonatelor în *punctele* de interpolare

La selecţia restricţiilor se dau valorile iniţiale şi finale ale coordonatelor cuplelor cinematice precum şi restricţiile de viteză şi acceleraţie pentru fiecare cuplă. Restricţiile sunt introduse printr-un dialog de forma celui din figura 3.22.

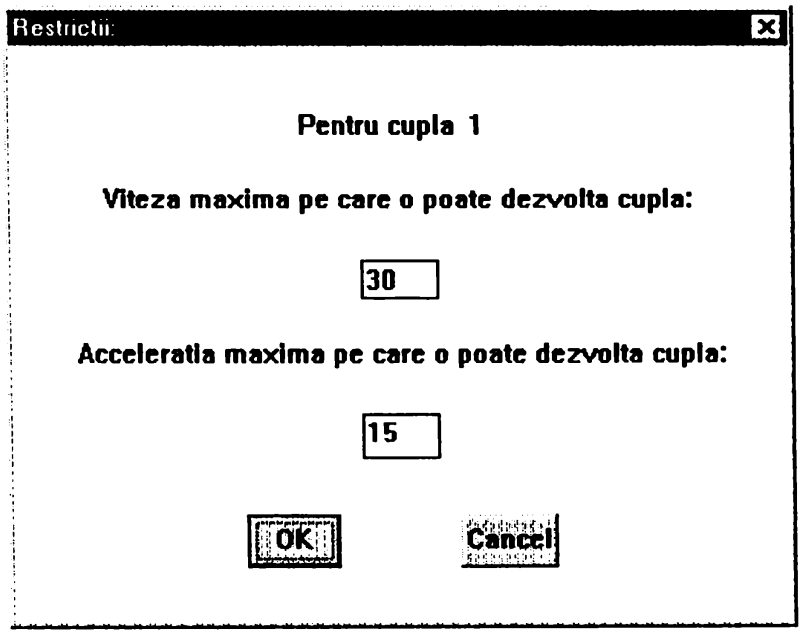


Fig.3.22. Impunerea restricţiilor cinematice

După introducerea datelor iniţiale, prin apelarea meniului **Calcule şi rezultate** se efectuează calculele necesare pentru determinarea timpilor de parcurs, a legilor de variaţie pentru coordonatele cuplelor, vitezelor şi acceleraţiilor acestora, cu respectarea restricţiilor cinematice impuse de sistemul de acţionare a robotului. Prin apelarea meniului **Reprezentări grafice** se realizează reprezentarea grafică a traiectoriilor calculate anterior.

BUPT

3.5. CONCLUZII

În cazul în care, între două *puncte* de precizie ale traiectoriei punctului caracteristic al unui robot mişcarea este liberă, generarea acesteia poate fi abordată direct în spaţiul cuplelor cinematice. Pentru stabilirea funcţiilor de interpolare a coordonatelor cuplelor se ţine cont de viteza pe care o are punctul caracteristic în momentul când trece prin *punctele* de precizie (*puncte* de oprire, când viteza este zero, sau *puncte* de trecere, când viteza este diferită de zero).

Cea mai simplă şi totuşi frecvent utilizată interpolare este cea liniară. Având în vedere discontinuităţile care marchează începutul şi sfârşitul mişcării interpolate liniar, când acestea au o influenţă semnificativă asupra preciziei parcurgerii traiectoriei, sunt folosite funcţiile de interpolare polinomială. Dintre acestea, cel mai frecvent utilizate sunt cele de gradul trei şi cinci. Polinoamele de grad superior prezintă dezavantajul că necesită calculul unui număr mare de coeficienţi polinomiali şi al punctelor de extrem.

Dintre funcţiile polinomiale, o amploare deosebită o are folosirea funcţiilor spline. Datorită avantajelor pe care acestea le prezintă, dar şi impunerii acestor funcţii pe plan teoretic şi practic şi mai ales implementării lor în diferite programe de tip CAD, de calcul numeric şi grafică precum MATHCAD şi MATLAB sau de proiectare asistată de calculator precum AUTOCAD, acestui tip de funcţii îi este dedicat capitolul următor integral. Interpolarea cu funcţii spline poate fi folosită atât în spaţiul cuplelor cinematice, cât şi în spaţiul operaţional.

Fiecare dintre funcţiile de interpolare propuse prezintă avantaje, dar şi dezavantaje. Traiectoria optimală din punct de vedere al timpului de parcurgere este traiectoria cu profil triunghiular de viteză. Dezavantajul acestei metode constă în obţinerea unor supraacceleraţii de valori foarte mari. Pentru a evita discontinuităţile acceleraţiilor cuplelor cinematice au fost propuse profiluri de viteză netezite (triunghiular şi trapezoidal). Dezavantajul acestora constă însă în majorarea duratei de parcurgere a traiectoriei. Pentru fiecare din tipurile de traiectorii propuse au fost prezentate avantajele şi dezavantajele.

Punctul caracteristic la robotului poate să treacă chiar prin *punctele* intermediare introduse pentru o definire cât mai fidelă a traiectoriei sau doar prin vecinătatea acestora. În cazul în care punctul caracteristic nu trece prin *punctele* de precizie, trebuie impuse condiţii de încadrare în toleranţele admisibile ale erorii de poziţionare.

Indiferent de modul de generare (prin *puncte* de oprire sau prin *puncte* de trecere), unul din parametrii importanţi ai mişcării este timpul de parcurgere a traiectoriei. Minimizarea acestuia constituie unul din cele mai frecvent utilizate criterii de optimizare a mişcării, atât în spaţiul cuplelor cinematice, cât şi în spaţiul operaţional. Pentru a putea compara diferitele tipuri de funcţii de interpolare prezentate a fost introdus un factor adimensional $\Omega\ t_f$, pentru fiecare tip de mişcare considerându-se aceeaşi deplasare şi aceeaşi acceleraţie.

Durata deplasării unei cuple cinematice depinde atât de distanţa parcursă, cât şi de caracteristicile de viteză şi acceleraţie impuse de servomotoarele de acţionare a cuplei. Timpul minim de parcurgere a traiectoriei este factorul decisiv în coordonarea cuplelor cinematice. Metoda timpului minim este destul de simplă şi uşor de aplicat, indiferent de tipul de mişcare.

BUPT

Programul realizat pentru generarea mişcării în spaţiul cuplelor oferă utilizatorului diferite variante de alegere a metodei de generare. Imaginile ce prezintă interfaţa pusă la dispoziţie utilizatorului au fost capturate şi introduse în lucrare pentru a sublinia concepţia generală, sistematizată care a stat la baza elaborării acestui program prietenos şi uşor de implemetat.

BUPT

4. INTERPOLAREA TRAIECTORIILOR FOLOSIND FUNCȚII SPLINE

4.1. INTRODUCERE

Problema teoretică clasică a interpolării funcțiilor după date exacte, definite pe o mulțime discretă de puncte, este rezolvată cu ajutorul polinoamelor de interpolare ale lui Lagrange. Această metodă asociază unui set de *n* puncte din plan, de abscise diferite, o curbă polinomială de gradul *n-1*, care trece prin toate cele *n* puncte. Metoda are avantajul continuității tuturor derivatelor curbei interpolatoare. Totuşi, curba interpolatoare Lagrange nu este naturală în sensul că oscilează prea mult între punctele fixe; acest aspect este prohibitiv în cazul utilizării operațiilor de interpolare pentru modelarea unor fenomene naturale [V6], [M11].

În ultimii ani s-au impus pe plan teoretic şi mai ales practic metode noi şi eficiente de interpolare care au primit denumirea de ***funcții spline***. O funcție spline este, de obicei, polinomială pe porțiuni, adică o funcție pentru care există o diviziune a intervalului **I** de interpolare în subintervale, astfel încât în interiorul fiecărui subinterval al diviziunii funcția reprezintă un polinom de un anumit grad *m*. În plus, funcția este de regulă continuă pe **I** împreună cu derivatele sale până la ordinul *m-1* inclusiv, şi are derivata de ordinul *m* cu pătrat integrabil.

Cuplele manipulatorului sunt deseori conduse astfel încât efectorul său final să urmărească o traiectorie operațională prespecificată. Există mai multe metode de generare a traiectoriilor carteziene [B20], [P5], [V12], [W10] care utilizează o mulțime de *puncte* nodale ce descriu traiectoria dorită, asupra cărora operează o transformare de coordonate, din spațiul operațional în spațiul cuplelor robotului, urmând ca apoi să se genereze traiectorii între două *puncte* şi să se concateneze segmentele obținute într-o traiectorie continuă. Astfel se procedează la generarea traiectoriei atât în spațiul operațional, cât şi în spațiul cuplelor cinematice conducătoare. Toate metodele de generare introduc erori, destul de mari uneori, în specificarea traiectoriei operaționale, datorită metodei alese pentru interpolarea traiectoriilor cuplelor. Practic, nici un tip de interpolare nu poate conduce la o traiectorie exactă.

Deşi curbele *spline cubice* nu oferă o precizie prea bună, ele sunt totuşi cele mai folosite în generarea traiectoriilor şi aceasta datorită proprietăților lor extremale naturale, unicității soluției pe care o oferă şi stabilității acesteia. Există şi alte tipuri de curbe spline utilizate în generarea traiectoriilor: curbe spline cuadratice, cvintice, etc.

Un remediu simplu pentru a micşora eroarea traiectoriei este de a selecta mai multe *puncte*. Uneori, acest lucru măreşte numărul polinoamelor şi, prin urmare, îngreunează calculele. Interpolarea la nivelul cuplelor şi aproximarea coordonatelor cuplelor, în sensul erorii pătratice minime, folosind funcții spline cubice au fost propuse în [L17], [L18], [L29]. Dezavantajul acestor metode constă în faptul că manipulatorul trebuie să aştepte până când toate punctele nodale au fost determinate, ceea ce nu convine în generarea traiectoriei în timp real. Această problemă este rezolvată în [L29] prin utilizarea erorii pătratice minime ca şi

BUPT

criteriu de concatenare (a porțiunilor) în interpolarea prin curbe spline *cubice* și *cuadratice* pe porțiuni.

În general, metodele de interpolare nu consideră vitezele și accelerațiile reale în noduri și, de aceea, este necesar un număr mare de noduri. Pentru a utiliza mai eficient datele reale corespunzătoare nodului cartezian, s-a propus [C2] folosirea polinoamelor *cvintice* în locul celor cubice. Polinoamele *cvintice* pe porțiuni sunt determinate pentru toate segmentele prin pozițiile, vitezele si accelerațiile corespunzătoare capetelor fiecărui segment.

Dezavantajele folosirii funcțiilor de interpolare spline sunt legate de faptul că traiectoria operațională obținută nu poate fi prevăzută, iar determinarea coeficienților polinomiali necesită cunoaşterea tuturor punctelor intermediare.

4.2. INTERPRETAREA FIZICĂ A INTERPOLĂRII CU FUNCȚII SPLINE

Funcțiile spline cubice au astfel de proprietăți încât asigură o înaltă eficacitate a acestui tip de interpolare. Se consideră clasa $C^{(2)}_{[t_0,t_n]}$ a funcțiilor care au derivate de ordinul doi cu pătrat integrabil pe segmentul $[t_o, t_n]$. Se pune problema găsirii funcției interpolatoare [V6]:

$$u \in C^{(2)}_{[t_0,t_n]},\ u(t_i) = f_i,\ i = 0,1,\ldots,n \tag{4.1}$$

care minimizează funcționala:

$$\Phi(u) = \int_{t_0}^{t_n} [\ddot{u}(t)]^2 dt \tag{4.2}$$

pe clasa de funcții $C^{(2)}_{[t_0,t_n]}$. Se demonstrează că minimul acestei funcționale este atins de funcția spline $Q_i(t)$ cubică pe porțiuni. În acest scop se consideră mărimea:

$$\Phi(u - Q_i) = \int_{t_0}^{t_n} [\ddot{u}(t) - \ddot{Q}_i(t)]^2 dt \tag{4.3}$$

unde u(t) este o funcție interpolatoare oarecare care îndeplineşte condițiile (4.1), iar $Q_i(t)$ este funcția spline. Integrând prin părți relația (4.3) și folosind proprietățile funcțiilor $Q_i(t)$ și u(t) rezultă:

$$\Phi(u - Q_i) = \int_{t_0}^{t_n} [\ddot{u}(t)]^2 dt - \int_{t_0}^{t_n} [\ddot{Q}_i(t)]^2 dt - 2\int_{t_0}^{t_n} [\ddot{u}(t) - \ddot{Q}_i(t)]\ddot{Q}_i(t)dt =$$

$$= \Phi(u) - \Phi(Q_i) - 2[(\dot{u} - \dot{Q}_i)\ddot{Q}_i\Big|_{t=t_0}^{t=t_n} - \int_{t_0}^{t_n} [\dot{u}(t) - \dot{Q}_i(t)]\dddot{Q}_i(t)dt] =$$

$$= \Phi(u) - \Phi(Q_i) + 2\sum_{i=0}^{n-1}\int_{t_i}^{t_{i+1}} [\dot{u}(t) - \dot{Q}_i(t)]\dddot{Q}_i(t)dt\ . \tag{4.4}$$

S-au folosit condițiile la limită $Q_i(t_0) = Q_i(t_n) = 0$. Deoarece şocul $Q_i(t) = c_i$ = constant pe segmentul $[t_i, t_{i+1}]$, rezultă:

$$\Phi(u - Q_i) = \Phi(u) - \Phi(Q_i) + 2\sum_{i=0}^{n-1} c_i \left\{[u(t_{i+1}) - Q_i(t_{i+1})] - [u(t_i) - Q_i(t_i)]\right\} = \Phi(u) - \Phi(Q_i) \tag{4.5}$$

Deoarece (4.3) implică evident $\Phi(u-Q_i)\geq 0$ pentru orice u din clasa $C^{(2)}_{[t_0,t_n]}$, se obține:

BUPT

$$\Phi(Q_i) \le \Phi(u) \qquad (4.6)$$

pentru orice funcție *u* care îndeplinește condițiile (4.1). Deci minimul funcționalei (4.2) se realizează pe funcția spline cubică pe porțiuni $Q_i(t)$, și nu este greu de arătat că alte puncte de minim nu există.

Se poate demonstra că funcțiile spline cubice construite în alte condiții la limită (cum ar fi impunerea anumitor viteze și/sau accelerații în *punctele de nodale* sau cunoașterea "a priori" a faptului că funcția este periodică) asigură în continuare minimizarea funcționalei (4.2), însă, în acest caz, nu pe întreaga clasă de funcții din (4.1), ci pe o submulțime a acesteia, alcătuită din funcții care satisfac condițiile la limita respectivă.

Pe baza rezultatului (4.6) se poate formula o altă definiție, echivalentă, a funcției spline cubice pe porțiuni: este o funcție din clasa $C^{(2)}_{[t_0,t_n]}$ care ia valori date în nodurile rețelei și minimizează funcționala (4.2). Această proprietate a funcției spline este importantă prin faptul că funcționala $\Phi(u)$ poate fi interpretată ca funcție analoagă energiei potențiale a unei bare elastice obligată să treacă prin punctele (t_i, f_i) ale planului, pe funcțiile spline cubice realizându-se minimul acestei energii, respectiv *ondularea* minimă a barei elastice. Aceasta este proprietatea *naturală* a funcțiilor spline cubice, care face ca acest tip de interpolare să fie cel mai potrivit pentru modelarea curbelor de evoluție a diverselor fenomene reale.

4.3. INTERPOLAREA TRAIECTORIILOR ÎN COORDONATELE CUPLELOR CU *PUNCTE* DE CONTROL

Pentru comanda unui manipulator, configurația efectorului final, atât în poziția inițială, cât și în cea finală, trebuie să fie specificată înainte ca traiectoria mișcării să fie generată. În generarea mișcării prin interpolare în coordonatele cuplelor cinematice, pentru efectorul final al robotului prezintă interes următoarele considerații [P4]:

- Când este selectat un obiect, mișcarea efectorului final trebuie condusă spre acel obiect, dar fără a intra în coliziune cu acesta.
- Dacă se specifică o poziție de pornire (de ridicare) în lungul vectorului normal la suprafața exterioară din poziția inițială și dacă se cere ca punctul caracteristic al robotului să treacă prin această poziție, atunci există o poziție de pornire admisibilă. Dacă suplimentar se specifică timpul necesar atingerii acestei poziții, se poate controla viteza cu care obiectul trebuie ridicat.
- Aceleași cerințe pentru mișcarea efectorului final sunt valabile și în punctul de coborâre pentru poziția finală a mișcării (mișcarea trebuie să aibă loc spre un punct al normalei la suprafață și apoi se coboară încet spre poziția finală), astfel încît se poate obține și controla direcția corectă de apropiere. Rezultă astfel patru poziții pentru mișcarea efectorului final: inițială, ridicare, coborâre și finală.
- Restricțiile impuse celor patru poziții sunt (fig.4.1):

a) Poziția inițială: viteza și accelerația sunt cunoscute (de obicei zero);
b) Poziția de ridicare: mișcare este continuă pentru punctele intermediare;
c) Poziția de coborâre: similar cu cea de ridicare;
d) Poziția finală: viteza și accelerația sunt cunoscute (normal nule).

- Suplimentar față de aceste restricții, valorile extreme ale variabilelor cuplelor trebuie să fie în limitele fizice și geometrice pentru fiecare cuplă.
- Considerații referitoare la timp:

BUPT

a) Pe segmentele inițial şi final ale traiectoriei: timpul este dat de viteza de apropiere a efectorului final spre/dinspre suprafață şi este stabilit constant în funcție de caracteristicile motoarelor cuplelor.

b) În *punctele* intermediare sau segmentul de mijloc: timpul este dat de valorile maxime ale vitezelor şi accelerațiilor cuplelor (se alege timpul maxim al cuplei celei mai lente - valoarea pe baza căreia se face coordonarea cuplelor).

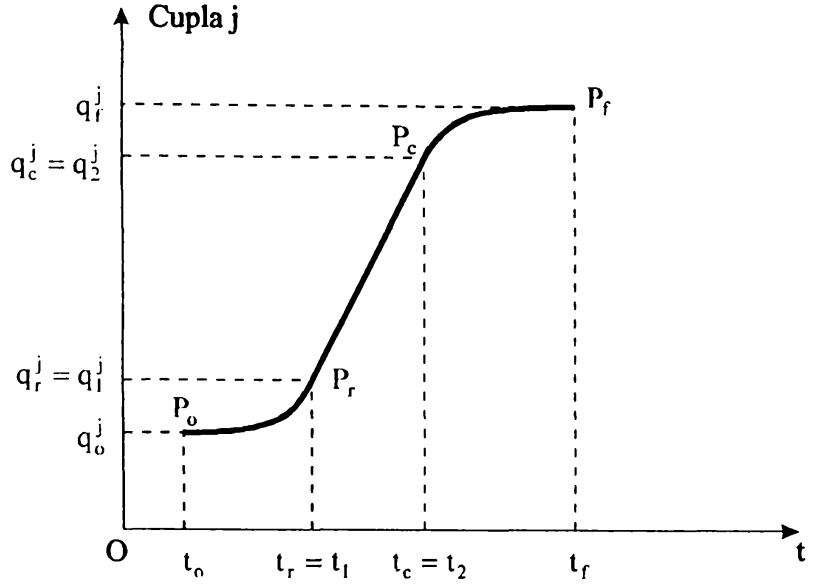


Fig. 4.1. Pozițiile pentru traiectoria cuplei *j*

Restricțiile impuse traiectoriilor cuplelor sunt prezentate în tabelul 4.1.

Tabelul 4.1. Restricțiile pentru generarea traiectoriei interpolate în coordonatele cuplelor cinematice

Poziția inițială:
1.Poziția (impusă) 2.Viteza (cunoscută, normal = zero) 3.Accelerația (cunoscută, normal = zero)
Pozițiile intermediare:
4.Poziția de ridicare (dată) 5.Poziția de ridicare (continuitatea cu segmentul precedent al traiectoriei) 6.Viteza (continuă cu segmentul precedent al traiectoriei) 7.Accelerația (continuă cu segmentul precedent al traiectoriei) 8.Poziția de coborâre (dată) 9.Poziția de coborâre (continuă cu segmentul următor al traiectoriei) 10.Viteza (continuă cu segmentul următor al traiectoriei) 11.Accelerația (continuă cu segmentul următor al traiectoriei)
Poziția finală:
12.Poziția (impusă) 13.Viteza (dată, normal = zero) 14.Accelerația (dată, normal =zero)

BUPT

Pornind de la aceste restricţii, se selectează o clasă de funcţii polinomiale de gradul *n* sau mai mic, astfel încât poziţia cerută a cuplei, viteza şi acceleraţia în aceste puncte nodale (iniţial, ridicare, coborâre, final) sunt satisfăcute, iar poziţia, viteza şi acceleraţia fiecărei cuple sunt continue în întreg intervalul de timp $[t_0, t_f]$. O soluţie o constituie funcţia polinomială de gradul şapte, pentru fiecare cuplă *j* :

$$q^j(t) = a_7t^7 + a_6t^6 + a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t^1 + a_0 \qquad (4.7)$$

unde coeficienţii necunoscuţi a_i se pot determina din cunoaşterea poziţiilor şi din condiţiile de continuitate. Utilizarea unei funcţii polinomiale de grad ridicat pentru interpolarea mişcării date prin punctele nodale nu este însă satisfăcătoare. Determinarea valorilor extreme este dificilă, iar mişcarea rezultată este necunoscută. O alternativă în rezolvarea acestei probleme constă în divizarea traiectoriei în mai multe segmente de traiectorie şi utilizarea unor funcţii polinomiale diferite, de grad mai mic. Există astfel mai multe metode de interpolare a traiectoriei divizate în segmente de traiectorie [F4], fiecare metodă având proprietăţile sale: *traiectorie 4-3-4* (capitolul 4.3.1), *traiectorie 3-5-3*, traiectorie 3-3-3-3-3 (capitolul 4.3.2). Traiectoria fiecărei cuple este divizată în trei sau cinci segmente de traiectorie. Astfel, pentru un manipulator cu N cuple cinematice, pentru traiectoria 4-3-4 se folosesc 3N segmente de traiectorie, se determină 7N coeficienţi polinomiali şi extremele celor 3N segmente.

4.3.1. CALCULUL TRAIECTORIEI DE TIP 4-3-4

Deoarece trebuie să se determine N traiectorii pentru fiecare segment de traiectorie, se introduce timpul normalizat p, $p \in [0,1]$, care permite tratarea ecuaţiilor fiecărui segment de traiectorie, pentru fiecare cuplă cinematică, în acelaşi mod. Pentru toate segmentele de traiectorie, timpul iniţial normalizat p = 0, iar timpul final normalizat p = 1.

Se definesc următoarele variabile:

- timpul normalizat $p = \dfrac{t - t_{i-1}}{t_i - t_{i-1}}$, $\quad t \in [t_{i-1}, t_i]$, $\quad p \in [0,1]$;
- timpul real t, în secunde;
- timpul real t_i la capătul segmentului *i* al traiectoriei;
- timpul real τ_i necesar parcurgerii segmentului *i* al traiectoriei: $\tau_i = t_i - t_{i-1}$

Traiectoria cuplei *j* este exprimată printr-un şir de funcţii polinomiale $h_i(t)$. În continuare, pentru simplificarea redactării, se va renunţa la indicarea cuplei *j*. Ecuaţiile polinomiale, pentru variabila fiecărei cuple şi pentru fiecare segment de traiectorie se exprimă, folosind timpul normalizat:

$$h_1(p) = a_{1,4}p^4 + a_{1,3}p^3 + a_{1,2}p^2 + a_{1,1}p^1 + a_{1,0} \qquad \text{(primul segment)} \qquad (4.8)$$

$$h_1(p) = a_{2,3}p^3 + a_{2,2}p^2 + a_{2,1}p + a_{2,0} \qquad \text{(al doilea segment)} \qquad (4.9)$$

$$h_n(p) = a_{n,4}p^4 + a_{n,3}p^3 + a_{n,2}p^2 + a_{n,1}p^1 + a_{n,0} \qquad \text{(ultimul segment)} \qquad (4.10)$$

Condiţiile la limită, pe care mulţimea funcţiilor polinomiale ce interpolează segmentele traiectoriei trebuie să le îndeplinească, sunt:

BUPT

1. Poziția inițială: $q_0 = q(p_0)$.
2. Valoarea vitezei inițiale: $\dot{q}_0 = \dot{q}(p_0)$, normal $\dot{q}_0 = 0$.
3. Valoarea accelerației inițiale: $\ddot{q}_0 = \ddot{q}(p_0)$, normal $\ddot{q}_0 = 0$.
4. Poziția de ridicare: $q_1 = q(p_1)$
5. Continuitatea poziției la p=p1: $p = p_1 \rightarrow \left[q(p_1^-) = q(p_1^+)\right]$
6. Continuitatea vitezei la p=p1: $p = p_1 \rightarrow \left[\dot{q}(p_1^-) = \dot{q}(p_1^+)\right]$
7. Continuitatea accelerației la p=p1: $p = p_1 \rightarrow \left[\ddot{q}(p_1^-) = \ddot{q}(p_1^+)\right]$
8. Poziția de coborâre: $q_2 = q(p_2)$
9. Continuitatea poziției la p=p2: $p = p_2 \rightarrow \left[q(p_2^-) = q(p_2^+)\right]$
10. Continuitatea vitezei la p=p2: $p = p_2 \rightarrow \left[\dot{q}(p_2^-) = \dot{q}(p_2^+)\right]$
11. Continuitatea accelerației la p=p2: $p = p_2 \rightarrow \left[\ddot{q}(p_2^-) = \ddot{q}(p_2^+)\right]$
12. Poziția finală: $q_f = q(p_f)$
13. Valoarea vitezei finale: $\dot{q}_f = \dot{q}(p_f)$, normal $\dot{q}_f = 0$.
14. Valoarea accelerației finale: $\ddot{q}_f = \ddot{q}(p_f)$, normal $\ddot{q}_f = 0$.

Prima şi a doua derivată ale acestor funcții polinomiale, în raport cu timpul real *t* se pot scrie:

$$\dot{q}_i(p) = \frac{dh_i(p)}{dt} = \frac{dh_i(p)}{dp}\frac{dp}{dt} = \frac{1}{t_i - t_{i-1}}\frac{dh_i(p)}{dt} = \frac{1}{\tau_i}\frac{dh_i(p)}{dp} = \frac{1}{\tau_i}\dot{h}_i(p); \qquad i = 1,2,n \quad (4.11)$$

$$\ddot{q}_i(p) = \frac{d^2h_i(p)}{dt^2} = \frac{1}{(t_i - t_{i-1})^2}\frac{d^2h_i(p)}{dp^2} = \frac{1}{\tau_i^2}\frac{d^2h_i(p)}{dp^2} = \frac{1}{\tau_i^2}\ddot{h}_i(p); \qquad i = 1,2,n \quad (4.12)$$

Funcțiile polinomiale pentru interpolarea *traiectoriei de tip 4-3-4*, obținute pe baza relațiilor (4.8)-(4.10) şi (4.11), (4.12) şi a condițiilor de mai sus sunt prezentate în Tabelul 4.2 unde s = f / g şi:

$$f = 2D_1\left(4 + \frac{2\tau_n}{\tau_2} + \frac{2\tau_n}{\tau_1} + \frac{3\tau_2}{\tau_1}\right) - \frac{D_2\tau_1}{\tau_2}\left(3 + \frac{\tau_n}{\tau_2}\right) + \frac{2D_n\tau_1}{\tau_n} - \dot{q}_0\tau_1\left(6 + \frac{6\tau_2}{\tau_1} + \frac{4\tau_n}{\tau_1} + \frac{3\tau_n}{\tau_2}\right) -$$

$$-\dot{q}_f\tau_1 - \ddot{q}_0\tau_1\tau_n\left(\frac{5}{3} + \frac{\tau_1}{\tau_2} + \frac{2\tau_1}{\tau_n} + \frac{5\tau_2}{\tau_n}\right) + \ddot{q}_f\tau_1\tau_n$$

$$g = \frac{\tau_n}{\tau_2} + \frac{2\tau_n}{\tau_1} + 2 + \frac{3\tau_2}{\tau_1}$$

$$D_1 = q_1 - q_0 = h_1(1) - h_1(0)$$

$$D_2 = q_2 - q_1 = h_2(1) - h_2(0)$$

$$D_n = q_f - q_2 = h_n(0) - h_n(-1)$$

BUPT

Tabel 4.2. Funcţiile polinomiale pentru *traiectoria de tip 4-3-4*

Primul segment de traiectorie:
$h_1(p)=\left(D_1-\dot{q}_0\tau_1-\frac{1}{2}\ddot{q}_0\tau_1^2-s\right)p^4+sp^3+\left(\frac{\ddot{q}_0\tau_1^2}{2}\right)p^2+(\dot{q}_0\tau_1)p+q_0$
$\dot{q}_1=\frac{\dot{h}_1(1)}{\tau_1}=\frac{4D_1}{\tau_1}-3\dot{q}_0-\ddot{q}_0\tau_1-\frac{s}{\tau_1}$
$\ddot{q}_1=\frac{\ddot{h}_1(1)}{\tau_1^2}=\frac{12D_1}{\tau_1^2}-\frac{12\dot{q}_0}{\tau_1}-5\ddot{q}_0-\frac{6s}{\tau_1^2}$
Al doilea segment de traiectorie:
$h_2(p)=\left(D_2-\dot{q}_1\tau_2-\frac{1}{2}\ddot{q}_1\tau_2^2\right)p^3+\left(\frac{\ddot{q}_1\tau_2^2}{2}\right)p^2+(\dot{q}_1\tau_2)p+q_1$
$\dot{q}_2=\frac{\dot{h}_2(1)}{\tau_2}=\frac{3D_2}{\tau_2}-2\dot{q}_1-\frac{\ddot{q}_1\tau_2}{2}$
$\ddot{q}_2=\frac{\ddot{h}_2(1)}{\tau_2^2}=\frac{6D_2}{\tau_2^2}-\frac{6\dot{q}_1}{\tau_2}-2\ddot{q}_1\tau_2$
Ultimul segment de traiectorie:
$h_n(p)=\left(9Dd_n-4\dot{q}_2\tau_n-\frac{1}{2}\ddot{q}_2\tau_n^2-5\dot{q}_f\tau_n+\frac{1}{2}\ddot{q}_f\tau_n^2\right)p^4+\left(-8D_n+5\dot{q}_f\tau_n-\frac{1}{2}\ddot{q}_f\tau_n^2+3\dot{q}_2\tau_n\right)p^3+$
$+\left(\frac{\ddot{q}_2\tau_n^2}{2}\right)p^2+(\dot{q}_2\tau_n)p+q_2$

Funcţiile polinomiale pentru interpolarea *traiectoriei de tip 3-5-3* se determină similar şi sunt prezentate în Tabelul 4.3.

4.3.2. TRAIECTORIA SPLINE CUBICĂ (5 CUBICE)

Funcţiile spline cubice oferă mai multe avantaje: grad scăzut al funcţiei polinomiale care permite continuitatea vitezei şi acceleraţiei, şi care reduce calculul şi posibilitatea instabilităţii numerice. În general, funcţia polinomială pentru fiecare segment al traiectoriei unei cuple este:

$$h_i(p)=a_{i,3}p^3+a_{i,2}p^2+a_{i,1}p+a_{i,0},\ i=1,2,3,4,n,\ p\in[0,1],\ t_{i-1}\le t\le t_i \qquad (4.13)$$

În interpolarea polinomială cu funcţii cubice sunt necesare cinci segmente de traiectorie şi şase puncte de interpolare. Din cele prezentate în paragraful precedent, rezultă că există doar patru poziţii pentru interpolare (P_0, P_r, P_c, P_f). Deci, pentru a avea suficiente condiţii la limită pentru determinarea coeficienţilor polinomiali, trebuie selectate încă două puncte nodale între P_r şi P_c. Nu este absolut necesar să se cunoască cu

BUPT

exactitate aceste două poziții suplimentare, ci doar intervalele de timp corespunzătoare și să fie îndeplinite condițiile de continuitate pentru viteză și accelerație. Astfel, condițiile la limită pe care funcțiile polinomiale ce interpolează segmentele traiectoriei trebuie să le îndeplinească sunt: restricții pentru poziție în P_0, P_r, P_c și P_f; continuitatea vitezei și accelerației în toate punctele de interpolare. În figura (4.2) sunt prezentate condițiile la limită pentru traiectoria interpolată prin cinci funcții polinomiale cubice. Variabilele subliniate reprezintă mărimile cunoscute înaintea calculării celor cinci funcții polinomiale cubice.

Tabel 4.3. Funcțiile polinomiale pentru *traiectoria de tip 3-5-3*

Primul segment de traiectorie:

$$h_1(p) = \left(D_1 - \dot{q}_0\tau_1 - \frac{1}{2}\ddot{q}_0\tau_1^2\right)p^3 + \left(\frac{\ddot{q}_0\tau_1^2}{2}\right)p^2 + (\dot{q}_0\tau_1)p + q_0$$

$$\dot{q}_1 = \frac{\dot{h}_1(1)}{\tau_1} = \frac{3D_1}{\tau_1} - 2\dot{q}_0 - \frac{\ddot{q}_0\tau_1}{2}$$

$$\ddot{q}_1 = \frac{\ddot{h}_1(1)}{\tau_1^2} = \frac{6D_1}{\tau_1^2} - \frac{6\dot{q}_0}{\tau_1} - 2\ddot{q}_0$$

Al doilea segment de traiectorie:

$$h_2(p) = \left(6D_2 - 3\dot{q}_1\tau_2 - 3\dot{q}_2\tau_2 - \frac{\ddot{q}_1\tau_2^2}{2} + \frac{\ddot{q}_2\tau_2^2}{2}\right)p^5 + \left(-15D_2 + 8\dot{q}_1\tau_2 + 7\dot{q}_2\tau_2 + \frac{3\ddot{q}_1\tau_2^2}{2} - \ddot{q}_2\tau_2^2\right)p^4 +$$

$$+\left(10D_2 - 6\dot{q}_1\tau_2 - 4\dot{q}_2\tau_2 - \frac{3\ddot{q}_1\tau_2^2}{2} + \frac{\ddot{q}_2\tau_2^2}{2}\right)p^3 + \left(\frac{\ddot{q}_1\tau_2^2}{2}\right)p^2 + (\dot{q}_1\tau_2)p + q_1$$

$$\dot{q}_2 = \frac{\dot{h}_2(1)}{\tau_2} = \frac{3D_n}{\tau_n} - 2\dot{q}_f - \frac{\ddot{q}_f\tau_n}{2}$$

$$\ddot{q}_2 = \frac{\ddot{h}_2(1)}{\tau_2^2} = -\frac{6D_n}{\tau_n^2} - \frac{6\dot{q}_f}{\tau_n} - 2\ddot{q}_f$$

Ultimul segment de traiectorie:

$$h_n(p) = \left(D_n - \dot{q}_f\tau_n + \frac{1}{2}\ddot{q}_f\tau_n^2\right)p^3 + \left(-3D_n + 3\dot{q}_f\tau_n - \ddot{q}_f\tau_n^2\right)p^2 + \left(3D_n - 2\dot{q}_f\tau_n + \frac{1}{2}\ddot{q}_f\tau_n^2\right)p + q_2$$

Prima și a doua derivată ale funcției polinomiale în raport cu timpul real sunt:

$$\dot{q}_i(p) = \frac{\dot{h}_i(p)}{\tau_i} = \frac{3a_{i,3}p^2 + 2a_{i,2}p + a_{i,1}}{\tau_i}, \qquad i = 1,2,3,4,n \tag{4.14}$$

$$\ddot{q}_i(p) = \frac{\ddot{h}_i(p)}{\tau_j^2} = \frac{6a_{i,3}p + 2a_{i,2}}{\tau_i^2}, \qquad i = 1,2,3,4,n \tag{4.15}$$

BUPT

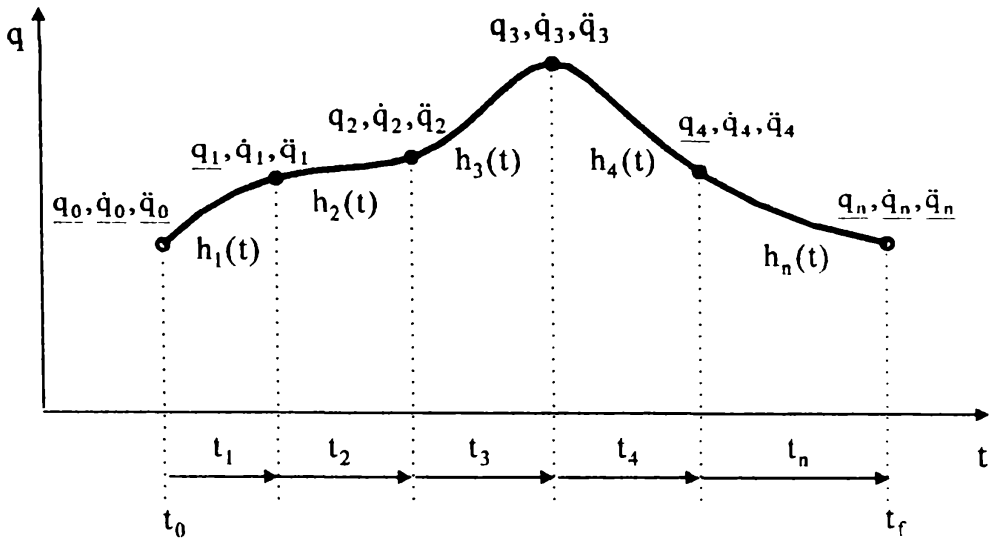


Fig. 4.2. Condiţiile limită pentru traiectoria interpolată prin 5 funcţii polinomiale cubice

Impunând poziţiile, vitezele şi acceleraţiile în poziţia iniţială şi finală, funcţiile polinomiale pentru interpolarea segmentului iniţial şi final ($h_1(p)$ şi $h_n(p)$) sunt determinate complet. O dată ce aceste funcţii polinomiale au fost calculate se pot determina $h_2(p)$, $h_3(p)$ şi $h_4(p)$ folosind restricţiile poziţiilor şi condiţiile de continuitate.

Funcţiile polinomiale pentru cele 5 segmente cubice sunt prezentate în Tabelul 4.4. S-au folosit, următoarele notaţii:

$$a_{2,3} = \tau_2^2 \frac{b_1}{T}, \qquad a_{3,3} = \tau_3^2 \frac{b_2}{T}, \qquad a_{4,3} = \tau_4^2 \frac{b_3}{T}$$

$$k_1 = q_4 - q_1 - \dot{q}_1\tau - \ddot{q}_1 \frac{\tau^2}{2} \qquad k_2 = \frac{\dot{q}_4 - \dot{q}_1 - \ddot{q}_1\tau - (\ddot{q}_4 - \ddot{q}_1)\frac{\tau}{2}}{3} \qquad k_3 = \frac{\ddot{q}_4 - \ddot{q}_1}{6}$$

$$T = \tau(\tau - \tau_2)(\tau - \tau_4) \qquad \tau = \tau_2 + \tau_3 + \tau_4$$

$$c = 3\tau^2 - 3\tau\tau_2 + \tau_2^2 \qquad d = 3\tau_4^2 - 3\tau_3\tau_4 + \tau_3^2$$

$$b_1 = k_1(\tau - \tau_2) + k_2(\tau_4^2 - d) - k_3\left[(\tau - \tau_4)d + \tau_4^2(\tau_4 - \tau_2)\right]$$

$$b_2 = -k_1(\tau + \tau_3) + k_2(c - \tau_4^2) + k_3\left[(\tau - \tau_4)c + \tau_4^2(\tau - \tau_2)\right]$$

$$b_3 = k_1(\tau - \tau_4) + k_2(d - c) + k_3\left[(\tau_4 - \tau_2)c - d(\tau - \tau_2)\right]$$

În concluzie, dacă se impun poziţiile P_0, P_r, P_c şi P_f, precum şi timpii t_i necesari parcurgerii fiecărui segment de traiectorie, se pot determina unic cinci funcţii polinomiale care satisfac toate restricţiile referitoare la poziţii, precum şi condiţiile de continuitate. Metoda prezentată permite interpolarea unei traiectorii prin şase puncte de interpolare, folosind cinci funcţii polinomiale. Pentru generalizarea problemei, se pot determina *n* funcţii polinomiale cubice.

BUPT

Tabelul 4.4. Funcţiile polinomiale pentru *traiectoria de tip 3-3-3-3-3*

Primul segment de traiectorie:		
$h_1(p)=\left(D_1-\dot{q}_0\tau_1+\frac{1}{2}\ddot{q}_0\tau_1^2\right)p^3$ $+\left(\frac{\ddot{q}_0\tau_1^2}{2}\right)p^2+(q_0\tau_1)p+q_0$	$\dot{q}_1=\frac{3D_1}{\tau_1}-2\dot{q}_0-\frac{\ddot{q}_0\tau_1}{2}$	$\ddot{q}_1=\frac{6D_1}{\tau_1^2}-\frac{6\dot{q}_0}{\tau_1}-2\ddot{q}_0$
Al doilea segment de traiectorie:		
$h_2(p)=a_{2,3}p^3+\left(\frac{\ddot{q}_1\tau_2^2}{2}\right)p^2+(\dot{q}_1\tau_2)p+q_1$ $q_2=a_{2,3}+\frac{\ddot{q}_1\tau_2^2}{2}+\dot{q}_1\tau_2+q_1$	$\dot{q}_2=\dot{q}_1+\ddot{q}_1\tau_2+\frac{3a_{2,3}}{\tau_2}$	$\ddot{q}_2=\ddot{q}_1+\frac{6a_{2,3}}{\tau_2^2}$
Al treilea segment de traiectorie:		
$h_3(p)=a_{3,3}p^3+\left(\frac{\ddot{q}_2\tau_3^2}{2}\right)p^2+(\dot{q}_2\tau_3)p+q_2$ $q_3=q_2+\dot{q}_2\tau_3+\frac{\ddot{q}_2\tau_3^2}{2}+a_{3,3}$	$\dot{q}_3=\dot{q}_2+\ddot{q}_2\tau_3+\frac{3a_{3,3}}{\tau_3}$	$\ddot{q}_3=\ddot{q}_2+\frac{6a_{3,3}}{\tau_3^2}$
Al patrulea segment de traiectorie:		
$h_4(p)=a_{4,3}p^3+\left(\frac{\ddot{q}_3\tau_4^2}{2}\right)p^2+(\dot{q}_3\tau_4)p+q_3$	$\dot{q}_4=\frac{3D_n}{\tau_n}-2\dot{q}_f-\frac{\ddot{q}_f\tau_n}{2}$	$\ddot{q}_4=-\frac{6D_n}{\tau_n^2}+\frac{6\dot{q}_f}{\tau_n}-2\ddot{q}_f$
Ultimul segment de traiectorie:		
$h_n(p)=\left(D_n-\dot{q}_f\tau_n+\frac{1}{2}\ddot{q}_f\tau_n^2\right)p^3+$ $+\left(-3D_n+3\dot{q}_f\tau_n-\ddot{q}_f\tau_n^2\right)p^2+$ $+\left(3D_n-2\dot{q}_f\tau_n+\frac{1}{2}\ddot{q}_f\tau_n^2\right)p+q_4$		

4.4. UTILIZAREA FUNCŢIILOR SPLINE PENTRU GENERAREA UNEI TRAIECTORII CE TRECE PRIN *n* PUNCTE DE PRECIZIE

Aproximarea unei traiectorii în coordonate carteziene necesită *m* funcţii de aproximare, câte una pentru fiecare cuplă cinematică. O asemenea funcţie de aproximare trebuie calculată pentru fiecare *punct* de precizie (*way point*) impus pe traiectorie. În plus, funcţia trebuie să fie continuă în poziţie, viteză şi acceleraţie (funcţia de aproximare de ordinul zero, unu şi doi) pentru ca parametrii cinematici să se încadreze în limitele fizice ale robotului (limitele performanţelor motoarelor de acţionare).

De asemenea, nu trebuie depăşită eroarea maximă admisă de la traiectoria geometrică impusă. Aceste condiţii ar putea fi satisfăcute prin derivarea unei singure funcţii polinomiale care să treacă prin toate *punctele* de precizie. S-ar putea însă ca

BUPT

funcţia polinomială să conţină extreme între *punctele* de precizie. Acestea trebuie să verifice de asemenea limitele fizice ale robotului. Ar putea rezulta astfel abateri mari de la traiectoria carteziană dorită. Ţinând cont că fiecare *punct* de precizie necesită trei condiţii iniţiale, gradul funcţiei polinomiale corespunzătoare unei traiectorii ce trece prin *k puncte* de precizie este foarte mare: *n=3k-1*.

Având în vedere acest aspect al problemei se propune aproximarea traiectoriei polinomiale prin segmente spline şi definirea câte unei funcţii polinomiale pentru fiecare segment de traiectorie, cu asigurarea continuităţii în fiecare *punct* de precizie. Pentru a asigura continuitatea în punctele intermediare şi pentru a obţine, în *punctele* de început şi de sfârşit, viteza şi acceleraţia nule, se utilizează funcţii polinomiale de gradul trei pentru segmentele intermediare şi funcţii polinomiale de gradul patru pentru segmentele de început şi de sfârşit [C19]. Astfel, pentru a genera o traiectorie prin *n puncte* de precizie sunt necesare *n-1* funcţii polinomiale pentru fiecare cuplă cinematică conducătoare. Deci, în cazul unui robot cu *m* cuple cinematice conducătoare, pentru a aproxima o traiectorie prin *n puncte* de precizie sunt necesare *(n-1)xm* funcţii polinomiale: *(n-3)xm* funcţii polinomiale de gradul trei pentru segmentele intermediare şi *2xm* funcţii polinomiale de gradul patru pentru segmentele extreme.

Metoda bazată pe utilizarea funcţiilor spline, în contextul de mai sus, a fost aplicată pentru generarea traiectoriei unui robot de tip RTRTTR al cărui punct caracteristic trebuie să ocupe un număr dat de configuraţii impuse în spaţiul operaţional [D23], [D24].

4.4.1. ECUAŢIA SEGMENTULUI SPLINE DINTRE DOUĂ *PUNCTE* INTERMEDIARE DE PRECIZIE

Se consideră două *puncte* intermediare de precizie P_k şi P_{k+1} ($2 \le k \le n-1$) ale unei traiectorii ce trece prin *n puncte* de precizie. Pentru o singură cuplă cinematică conducătoare *i*, ecuaţia segmentului spline determinat de *punctele* P_k şi P_{k+1} este de forma:

$$F_k^i(t) = C_{0,k}^i + C_{1,k}^i \cdot t + C_{2,k}^i \cdot t^2 + C_{3,k}^i \cdot t^3 \qquad (4.16)$$

unde $F_k^i(t)$ reprezintă legea de variaţie în raport cu timpul a coordonatei (generalizate) a cuplei cinematice conducătoare *i*.

Considerând $t_k=0$, în intervalul de timp $t \in [0, t_{k+1}]$, se impun următoarele condiţii limită:

$$\begin{cases} F_k^i(0) = q_k^i \\ F_k^i(t_{k+1}) = q_{k+1}^i \\ \dot{F}_k^i(0) = \dot{q}_k^i \\ \dot{F}_k^i(t_{k+1}) = \dot{q}_{k+1}^i \end{cases} \qquad (4.17)$$

unde q_k^i şi q_{k+1}^i reprezintă valorile variabilei cuplei cinematice conducătoare *i* în punctele P_k şi P_{k+1}, iar $\dot{q}_k^i$ şi $\dot{q}_{k+1}^i$ reprezintă vitezele cuplei cinematice conducătoare *i* în punctele P_k şi P_{k+1}.

Coeficienţii funcţiei polinomiale (4.16) se determină pe baza condiţiilor (4.17) şi se exprimă în formă matriceală astfel:

BUPT

$$\begin{bmatrix} C^i_{0,k} \\ C^i_{1,k} \\ C^i_{2,k} \\ C^i_{3,k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{3}{t^2_{k+1}} & \frac{3}{t^2_{k+1}} & -\frac{2}{t_{k+1}} & -\frac{1}{t_{k+1}} \\ \frac{2}{t^3_{k+1}} & -\frac{2}{t^3_{k+1}} & \frac{3}{t^2_{k+1}} & \frac{1}{t^2_{k+1}} \end{bmatrix} \cdot \begin{bmatrix} q^i_{ki} \\ qk+1 \\ \dot{q}^i_k \\ \dot{q}^i_{k+1} \end{bmatrix} \tag{4.18}$$

Pentru a putea calcula coeficienţii funcţiei polinomiale este necesar să se cunoască valorile t_{k+1}, $q_k{}^i$ şi $q_{k+1}{}^i$, $\dot{q}^i_k$ şi $\dot{q}^i_{k+1}$ asociate condiţiilor impuse prin relaţiile (4.18).

Se selectează valorile lui t_{k+1} pentru care *punctul* de precizie P_{k+1} ocupă cea mai apropiată poziţie raportată la traiectoria impusă. Se alege valoarea lui t_{k+1} ca fiind lungimea coardei dintre *punctele* de precizie P_k şi P_{k+1}, în coordonatele cuplei:

$$t_{k+1} = \sqrt{\sum_{i=1}^{m} \left(q^i_{k+1} - q^i_k\right)^2} \tag{4.19}$$

Vitezele $\dot{q}^i_k$ şi $\dot{q}^i_{k+1}$ se determină folosind condiţia de continuitate a acceleraţiilor în *punctele* de precizie. Considerând trei *puncte* de precizie P_k, P_{k+1} şi P_{k+2} ($0 \le t \le t_{k+1}$ şi $0 \le t \le t_{k+2}$), continuitatea acceleraţiilor la sfârşitul segmentului *k* şi începutul segmentului *k+1* impune egalitatea:

$$\ddot{q}^i_k\left(t = t_{k+1}\right) = \ddot{q}^i_{k+1}\left(t = t_{k+2} = 0\right) \tag{4.20}$$

unde:

$$\begin{aligned} \ddot{q}^i_k\left(t_{k+1}\right) &= 2C^i_{2,k} + 6C^i_{3,k} \cdot t_{k+1} \\ \ddot{q}^i_{k+1}\left(t_{k+2} = 0\right) &= 2C^i_{2,k+1} \end{aligned} \tag{4.21}$$

Matricea vitezelor $\dot{\mathbf{q}}^i$ se determină rezolvând ecuaţia matriceală (4.22):

$$\begin{bmatrix} t_4 & 2(t_3+t_4) & t_3 & 0 & 0 & 0 & 0 \\ 0 & t_5 & 2(t_4+t_5) & t_4 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & t_{n-1} & 2(t_{n-2}+t_{n-1}) & t_{n-2} \end{bmatrix} \cdot \begin{bmatrix} \dot{q}^i_2 \\ \dot{q}^i_3 \\ \vdots \\ \dot{q}^i_{n-1} \end{bmatrix} = \tag{4.22}$$

$$\begin{bmatrix} \frac{3}{t_3 t_4}\left[t^2_3\left(q^i_4 - q^i_3\right) + t^2_4\left(q^i_3 - q^i_2\right)\right] \\ \frac{3}{t_4 t_5}\left[t^2_4\left(q^i_5 - q^i_4\right) + t^2_5\left(q^i_4 - q^i_3\right)\right] \\ \vdots \\ \frac{3}{t_{n-2} t_{n-1}}\left[t^2_{n-2}\left(q^i_{n-1} - q^i_{n-2}\right) + t^2_{n-1}\left(q^i_{n-2} - q^i_{n-3}\right)\right] \end{bmatrix}$$

Ecuaţia (4.22) se poate exprima simbolic:

$$[t]_{(n-4)\times(n-2)} \cdot \left\{\dot{q}^i\right\}_{(n-2)\times 1} = \left\{a^i\right\}_{(n-4)\times 1} \tag{4.23}$$

BUPT

4.4.2. ECUAȚIA SEGMENTULUI SPLINE PENTRU PRIMUL ȘI ULTIMUL SEGMENT

Pentru primul şi ultimul segment se impune condiția inițială de funcționare fără şocuri:

$$\dot{q}_1^i = \ddot{q}_1^i = \dot{q}_n^i = \ddot{q}_n^i = 0 \tag{4.24}$$

ceea ce impune utilizarea unei funcții polinomiale de gradul patru, de forma:

$$F_k^i(t) = C_{0,k}^i + C_{1,k}^i \cdot t + C_{2,k}^i \cdot t^2 + C_{3,k}^i \cdot t^3 + C_{4,k}^i \cdot t^4 \tag{4.25}$$

unde k=1 şi k=n-1.

Pentru segmentul determinat de *punctele* de precizie P_1 şi P_2 ($t \in [0, t_2]$) se impun următoarele condiții limită:

$$\begin{aligned} F_1^i(0) &= q_1^i & F_1^i(t_2) &= q_2^i \\ \dot{F}_1^i(0) &= 0 & \dot{F}_1^i(t_2) &= \dot{q}_2^i \\ \ddot{F}_1^i(0) &= 0 & \ddot{F}_1^i(t_2) &= \ddot{q}_2^i \end{aligned} \tag{4.26}$$

din care se determină valorile coeficienților polinomiali $C_{0,1}^i$, $C_{1,1}^i$, $C_{2,1}^i$, $C_{3,1}^i$, $C_{4,1}^i$.

Pentru segmentul determinat de *punctele* de precizie P_{n-1} şi P_n ($t \in [0, t_n]$) se impun condițiile:

$$\begin{aligned} F_{n-1}^i(0) &= q_{n-1}^i & F_{n-1}^i(t_n) &= q_n^i \\ \dot{F}_{n-1}^i(0) &= \dot{q}_{n-1}^i & \dot{F}_{n-1}^i(t_n) &= 0 \\ \ddot{F}_{n-1}^i(0) &= \ddot{q}_{n-1}^i & \ddot{F}_{n-1}^i(t_n) &= 0 \end{aligned} \tag{4.27}$$

din care rezultă valorile coeficienților polinomiali $C_{0,n-1}^i$, $C_{1,n-1}^i$, $C_{2,n-1}^i$, $C_{3,n-1}^i$, $C_{4,n-1}^i$.

4.4.3. DETERMINAREA VITEZELOR ȘI RESPECTAREA RESTRICȚIILOR DE VITEZĂ ȘI ACCELERAȚIE

Luând în considerare şi segmentele extreme, ecuația (4.22), din care se determină matricea vitezelor $\{\dot{q}^i\}$ devine (4.29), sau în formă simbolică (4.30), unde s-au folosit notațiile (4.28):

$$\begin{aligned} T_{2,2} &= \frac{2}{t_3} + \frac{3}{t_2} \\ T_{2,3} &= \frac{1}{t_3} \\ T_{n-1,n-2} &= \frac{1}{t_{n-1}} \\ T_{n-1,n-1} &= \frac{2}{t_{n-1}} + \frac{3}{t_n} \\ a_2^i &= \frac{3}{t_3^2}\left(q_3^i - q_2^i\right) + \frac{6}{t_2^2}\left(q_2^i - q_1^i\right) \\ a_{n-1}^i &= \frac{3}{t_{n-1}^2}\left(q_{n-1}^i - q_{n-2}^i\right) + \frac{6}{t_n^2}\left(q_n^i - q_{n-1}^i\right) \end{aligned} \tag{4.28}$$

BUPT

$$\begin{bmatrix} T_{2,2} & T_{2,3} & 0 & 0 & \dots & 0 & 0 & 0 \\ t_4 & 2(t_3+t_4) & t_3 & 0 & \dots & 0 & 0 & 0 \\ 0 & t_5 & 2(t_4+t_5) & t_4 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & t_{n-1} & 2(t_{n-2}+t_{n-1}) & t_{n-2} \\ 0 & 0 & 0 & 0 & \dots & 0 & T_{n-1,n-2} & T_{n-1,n-1} \end{bmatrix} \cdot \begin{bmatrix} \dot{q}_2^i \\ \dot{q}_3^i \\ \dot{q}_4^i \\ \vdots \\ \dot{q}_{n-2}^i \\ \dot{q}_{n-1}^i \end{bmatrix} =$$

$$= \begin{bmatrix} a_2^i \\ \frac{3}{t_3 t_4}\left[t_3^2\left(q_4^i - q_3^i\right) + t_4^2\left(q_3^i - q_2^i\right)\right] \\ \frac{3}{t_4 t_5}\left[t_4^2\left(q_5^i - q_4^i\right) + t_5^2\left(q_4^i - q_3^i\right)\right] \\ \vdots \\ \frac{3}{t_{n-2} t_{n-1}}\left[t_{n-2}^2\left(q_{n-1}^i - q_{n-2}^i\right) + t_{n-1}^2\left(q_{n-2}^i - q_{n-3}^i\right)\right] \\ a_{n-1}^i \end{bmatrix} \tag{4.29}$$

respectiv:

$$[T]_{(n-2)\times(n-2)} \cdot \left\{\dot{q}^i\right\}_{(n-2)\times 1} = \left\{A^i\right\}_{(n-2)\times 1} \tag{4.30}$$

Rezolvând ecuația matriceală (4.29) se determină matricea vitezelor $\left\{\dot{q}^i\right\}$, definind astfel toate funcțiile polinomiale utilizate pentru aproximarea traiectoriei prescrise. În ecuația matricelă (4.29), pentru reprezentarea matricelor, s-au folosit paranteze şi acolade pentru a sublinia dimensiunile acestora.

Parametrului timp i se asociază o unitate de timp fizic, înlocuind *t* cu T/S, unde T este timpul fizic scurs de la începutul segmentului spline, iar S este un factor de scară. Valoarea lui S se alege astfel încât, pe de o parte să se evite atingerea valorilor maxime ale vitezei şi acceleraţiei oricărei cuple cinematice, iar pe de altă parte să se realizeze parcurgerea traiectoriei în timp minim, asigurând încadrarea în limitele fizice ale robotului.

Pentru a respecta restricţia de viteză a cuplelor cinematice, S se poate determina cu o relaţie de forma:

$$S_v = \max_i \left[\frac{\max_k \left(\left| \dot{q}_k^i(t) \right|_{0 \le t \le t_{k+1}} \right)}{K_v^i} \right] \tag{4.31}$$

unde $\dot{q}_k^i$ este viteza pe segmentul spline *k* (k=1,...,n-1) prin acţionarea cuplei cinematice conducătoare *i*, (i=1,...m), iar K_v^i este viteza maximă a cuplei cinematice *i*.

Similar, pentru acceleraţie:

BUPT

$$S_a^2 = \max_i \left[\frac{\max_k \left(\left| \ddot{q}_k^i(t) \right|_{0 \le t \le t_{k+1}} \right)}{K_a^i} \right] \tag{4.32}$$

unde $\ddot{q}_k^i$ este accelerația pe segmentul spline k (k=1,...,n-1) prin acționarea cuplei cinematice conducătoare i, (i=1,...m), iar K_a^i este accelerația maximă a acestei cuple cinematice.

Valoarea optimă a factorului S este:

$$S=\max(S_v, S_a) \tag{4.33}$$

4.4.4. EXEMPLU DE APLICARE A METODEI PENTRU GENERAREA MIȘCĂRII ROBOTULUI RTRTTR

Se consideră robotul de tip RTRTTR de construcție modulară, a cărui schemă cinematică structurală este prezentată în fig.4.3. Pentru satisfacerea sarcinii robotului, pentru evitarea coliziunii cu obstacolele, între poziția inițială și cea finală este necesar să se introducă două *puncte* intermediare (*puncte* de precizie).

Poziția și orientarea efectorului final al robotului se exprimă în fiecare *punct* de precizie impus prin câte o matrice de transformare generală $\mathbf{G}_6^0$.

Deoarece metoda utilizată pentru generarea traiectoriei folosește funcții de interpolare pentru variabilele cuplelor cinematice conducătoare este necesară rezolvarea modelului geometric invers.

Folosind convenția Hartenberg-Denavit se atașează elementelor robotului câte un sistem de referință (fig.4.3) și se determină expresia matricei de transformare generală $\mathbf{G}_6^0$ [D50], unde s-a notat: cos q_i=C_i, sin q_i=S_i și $\ell_5'=\ell_5+\ell_6$.

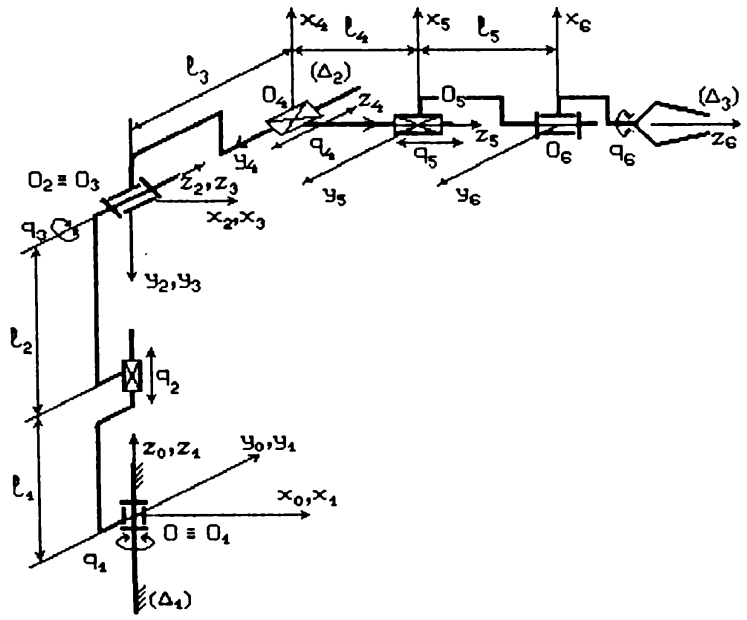


Fig.4.3. Schema cinematică structurală a robotului RTRTTR

BUPT

$$\mathbf{G}_6^0 = \begin{bmatrix} C_1S_3C_6 + S_1S_6 & -C_1S_3S_6 + S_1S_6 & C_1C_3 & C_1C_3(\ell_4 + \ell_5' + q_5) - S_1(\ell_3 + q_4) \\ S_1S_3C_6 - C_1S_6 & -S_1S_3S_6 - C_1S_6 & S_1C_3 & S_1C_3(\ell_4 + \ell_5' + q_5) + C_1(\ell_3 + q_4) \\ C_3C_6 & -C_3S_6 & -S_3 & -S_3(\ell_4 + \ell_5' + q_5) + \ell_1 + \ell_2 + q_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.34)$$

Rezolvând ecuațiile modelului geometric care se obțin din matricea (4.34) se determină expresiile coordonatelor cuplelor cinematice conducătoare, unde **n**, **o** și **a** sunt versorii sistemului atașat efectorului final, iar **p** este vectorul de poziție al originii sistemului atașat efectorului final [D16].

O soluție posibilă a acestor ecuații este [D50]:

$$q_1 = \operatorname{arctg}\frac{a_y}{a_x} \qquad q_4 = \frac{a_x p_y + a_y p_x}{\sqrt{1 - a_z^2}} - \ell_3$$

$$q_2 = p_z - \ell_1 - \ell_2 - a_z \cdot \frac{a_x p_x + a_y p_y}{1 - a_z^2} \qquad q_5 = \frac{a_x p_x + a_y p_y}{1 - a_z^2} - \ell_4 - \ell_5' \quad (4.35)$$

$$q_3 = \operatorname{arctg}\left(-\frac{a_z}{\sqrt{1 - a_z^2}}\right) \qquad q_6 = \operatorname{arctg}\left(-\frac{o_z}{n_z}\right)$$

Poziția și orientarea efectorului final al robotului se exprimă în fiecare *punct* de precizie prin câte o matrice de transformare generală $\mathbf{G}_6^0$. Se stabilește numărul *punctelor* de precizie prin care trebuie să treacă efectorul final al robotului. Se impun poziția și orientarea efectorului final la momentul inițial, la momentul final și în fiecare *punct* de precizie din spațiul de lucru prin matricele $\mathbf{G}_{6_k}^0$. Rezolvând modelul geometric invers se determină valorile coordonatelor cuplelor cinematice conducătoare pentru fiecare *punct* de precizie.

Programul elaborat pentru generarea traiectoriei, bazat pe metoda prezentată, poate fi folosit pentru orice robot, cu orice număr de grade de libertate și oricâte *puncte* de precizie. Utilizatorul trebuie să introducă numărul cuplelor cinematice conducătoare, numărul *punctelor* de precizie, valorile coordonatelor cuplelor în *punctele* de precizie și vitezele și accelerațiile maxime ale cuplelor robotului. Programul calculează și afișează [D23], [D24]:

- valorile timpilor t_{k+1} necesari descrierii fiecărui segment spline;
- matricea [T];
- matricea [A], ale cărei coloane sunt matricele $[A^i]_{(n-1)x1}$;
- valorile timpilor reali T_{k+1} necesari parcurgerii fiecărui segment spline;
- factorul se scară S;
- matricea vitezelor $[\dot{q}]$, ale cărei coloane sunt matricele $\{\dot{q}^i\}$;
- matricea accelerațiilor $[\ddot{q}]$, ale cărei coloane sunt matricele $\{\ddot{q}^i\}$;
- expresiile funcțiilor de interpolare și graficele acestora.

Se prezintă spre exemplu cazul când efectorul final al robotului studiat trebuie să treacă prin două *puncte* intermediare, cu orientare constantă. La momentul inițial, respectiv final, matricele de transformare $\mathbf{G}_6^0$ sunt:

BUPT

$$\mathbf{G}^0_{6_{initial}} = \begin{bmatrix} 0 & 0 & -1 & -12 \\ 0 & 1 & 0 & 4 \\ 1 & 0 & 0 & 14 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{G}^0_{6_{final}} = \begin{bmatrix} 0 & 0 & -1 & -12 \\ 0 & 1 & 0 & 3 \\ 1 & 0 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.36}$$

Vectorii de poziţie pentru *punctele* intermediare sunt: $\mathbf{p}_1 = -15\,\mathbf{i} - 5\,\mathbf{j} + 11\,\mathbf{k}$ şi $\mathbf{p}_2 = -10\,\mathbf{i} - 9\,\mathbf{j} + 13\,\mathbf{k}$. Aceste *puncte* se ating printr-o ridicare pe verticală şi o extensie a braţului în prima fază, o pivotare în faza a doua şi apoi o coborâre în poziţia finală. Se observă că nu este necesară acţionarea cuplelor (*1*), (*3*) şi (*6*).

Folosind metoda prezentată, funcţiile de interpolare a traiectoriei sunt:

• pentru primul segment de traiectorie:

$F_1^2(t) = 1.000 - 0.012\,t^3 + 0.001\,t^4$
$F_1^4(t) = -8.000 + 0.020\,t^3 - 0.001\,t^4$
$F_1^5(t) = 2.000 + 0.013\,t^3 - 0.001\,t^4$

• pentru al doilea segment de traiectorie:

$F_2^2(t) = -2.000 - 0.001\,t + 0.182\,t^2 - 0.020\,t^3$
$F_2^4(t) = 1.000 + 1.596\,t - 0.064\,t^2 - 0.013\,t^3$
$F_2^5(t) = 5.000 - 0.058\,t - 0.199\,t^2 + 0.014\,t^3$

• pentru ultimul segment de traiectorie:

$F_3^2(t) = 0.000 - 0.322\,t - 0.229\,t^2 + 0.050\,t^3 - 0.003\,t^4$
$F_3^4(t) = 5.000 - 0.971\,t - 0.318\,t^2 + 0.080\,t^3 - 0.005\,t^4$
$F_3^5(t) = 0.000 - 0.784\,t + 0.091\,t^2 - 0.001\,t^3 - 0.000\,t^4$

În fig.4.4 sunt prezentate graficele traiectoriilor obţinute. În orice moment al mişcării pe traiectorie se poate determina poziţia fiecărei cuple cinematice conducătoare.

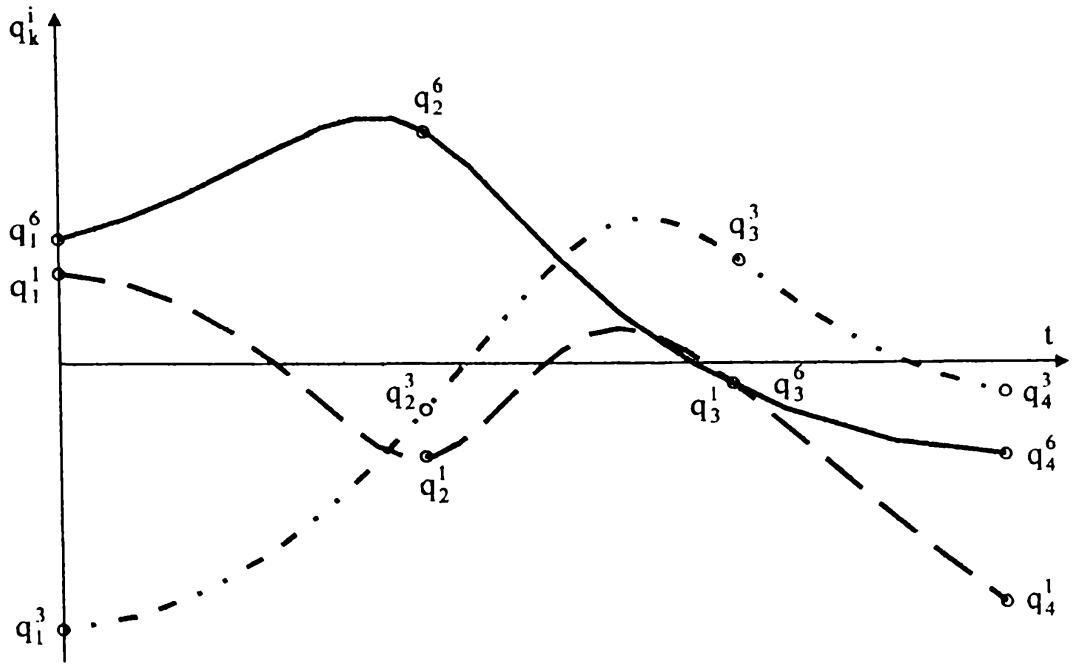


Fig.4.4. Reprezentarea funcţiilor de interpolare

BUPT

4.5. INTERPOLAREA TRAIECTORIILOR SPLINE CUBICE CU RESTRICŢII CINEMATICE ŞI DINAMICE

Tehnica lui Taylor de generare a traiectoriei rectilinii generează vectorii în spaţiul cuplelor cinematice $\{\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t)\}$ de-a lungul traiectoriei carteziene dorite, fără a lua în considerare însă dinamica manipulatorului. Totuşi, servomotorul fiecărei cuple este supus saturării şi nu poate furniza forţe/momente cu valori nelimitate. Astfel, în generarea traiectoriei (rectilinii) trebuie luate în considerare restricţiile pentru forţe şi momente. Aceasta sugerează considerarea controlului manipulatorului în două faze coerente ale execuţiei: generarea traiectoriei optime off-line, urmată de controlul parcurgerii (urmăririi) traiectoriei on-line.

În generarea unei traiectorii rectilinii carteziene, traiectoria este constrânsă în coordonate carteziene, în timp ce forţele/momentele servomotoarelor de acţionare a cuplelor cinematice sunt limitate în coordonatele cuplelor. Generarea mişcării devine deci o problemă de optimizare, cu restricţii mixte (restricţiile traiectoriei şi ale forţelor/ momentelor) în două sisteme de coordonate diferite. Astfel, fie trebuie convertită traiectoria cartezianã în traiectorii ale cuplelor şi controlat robotul la nivelul cuplelor (Lee şi Chung [L8]), fie trebuie convertite limitele forţelor/momentelor din cuplele cinematice în limitele lor corespunzătoare în sistemul cartezian, optimizată traiectoria cartezianã şi controlat robotul la nivelul efectorului final (Lee şi Lee [L9]).

Deşi aceasta implică numeroase transformări neliniare între coordonatele carteziene şi coordonatele cuplelor, este mai uşor de abordat problema generării traiectoriei în spaţiul cuplelor. Lin [L18] a propus o mulţime de funcţii spline la nivelul coordonatelor cuplelor pentru interpolarea segmentelor între *punctele* nodale selectate, de-a lungul unei traiectorii carteziene date. Acest concept presupune conversia traiectoriei carteziene dorite în reprezentarea funcţională a N traiectorii ale cuplelor, câte una pentru fiecare cuplă. Deoarece nu este cunoscută nici-o transformare care să asocieze traiectoriei rectilinii reprezentarea sa echivalentă în spaţiul cuplelor cinematice, metodele cunoscute de aproximare a curbelor trebuie utilizate pentru aproximarea traiectoriei carteziene. Astfel, pentru a aproxima traiectoria cartezianã dorită în spaţiul cuplelor cinematice, se pot selecta suficiente *puncte* nodale de-a lungul traiectoriei, şi fiecare segment de traiectorie specificat de către două *puncte* nodale adiacente poate fi interpolat prin N funcţii polinomiale ale cuplelor, câte o funcţie pentru fiecare traiectorie a unei cuplei. Aceste funcţii trebuie să treacă prin *punctele* nodale selectate. Deoarece traiectoriile polinomiale cubice sunt netede şi au abateri mici ale deplasărilor unghiulare între două *puncte* nodale adiacente, Lin [L18] a adoptat ideea utilizării funcţiilor polinomiale spline cubice pentru interpolarea segmentului între două *puncte* nodale adiacente. Deplasările cuplelor pentru cele n - 2 *puncte* nodale selectate sunt interpolate prin funcţii polinomiale cubice pe porţiuni. Pentru satisfacerea condiţiilor de continuitate pentru deplasările cuplelor, viteza şi acceleraţia pe întreaga traiectorie pentru traiectoria cartezianã, trebuie adăugate două *puncte* nodale cu deplasările cuplelor nespecificate, pentru a fi suficiente grade de libertate în rezolvarea polinoamelor cubice supuse condiţiilor de continuitate. Astfel, numărul total al *punctelor* nodale devine n şi fiecare traiectorie a cuplei constă în n-1 funcţii polinomiale cubice pe porţiuni. Folosind condiţia de continuitate, cele două *puncte* nodale suplimentare sunt exprimate ca o combinaţie de variabile necunoscute şi constante cunoscute. Astfel, trebuie rezolvate doar n-2 ecuaţii. Forma ecuaţiei matriceale

BUPT

rezultate facilitează calculul. După rezolvarea ecuației matriceale, funcțiile spline rezultate sunt exprimate în funcție de intervalele de timp dintre nodurile adiacente. Pentru a minimiza timpul total de parcurs de-a lungul traiectoriei, aceste intervale de timp trebuie ajustate sub influența restricțiilor cuplelor, în fiecare din cele *n - 1* polinoame cubice. Astfel, problema se reduce la optimizarea minimizării timpului total de parcurs prin ajustarea intervalelor de timp.

Fie **G**(t) matricea de transformare omogenă care exprimă poziția și orientarea sistemului de referință atașat efectorului final. Efectorul final trebuie să treacă printr-o secvență de *n puncte* nodale carteziene $[\mathbf{G}(t_1),\mathbf{G}(t_2),\ldots,\mathbf{G}(t_n)]$. Vectorii pozițiilor cuplelor ce corespund acestor *puncte* nodale carteziene: $\left[\left(q_1^1,q_1^2,\ldots,q_1^N\right),\left(q_2^1,q_2^2,\ldots,q_2^N\right),\ldots,\left(q_n^1,q_n^2,\ldots,q_n^N\right)\right]$ pot fi determinați folosind **MGI** (q_i^j variabila cuplei *j* în punctul nodal *i* ce corespunde matricei $\mathbf{G}_i(t)$). Astfel, obiectivul constă în a afla o traiectorie polinomială cubică pentru fiecare cuplă *j* care aproximează pozițiile cuplei $\left[q_1^j(t_1),q_2^j(t_2),\ldots,q_n^j(t_n)\right]$, unde $t_1 < t_2 < \ldots < t_n$ este un șir ordonat de intervale de timp care indică momentele când efectorul final trebuie să treacă prin aceste *puncte* nodale. La momentul inițial $t = t_1$ și la momentul final $t = t_n$, $\left[\left(q_1^j,\dot{q}_1^j,\ddot{q}_1^j\right);\left(q_n^j,\dot{q}_n^j,\ddot{q}_n^j\right)\right]$ specifică deplasarea, viteza și accelerația cuplei cinematice. În plus, sunt de asemenea specificate deplasările cuplelor q_k^j la $t = t_k$, pentru k = 3, 4, ..., n-2, pentru ca traiectoria cuplei să treacă prin aceste *puncte*. Totuși, q_2 și q_{n-1} nu sunt specificate. Acestea sunt cele două *puncte* nodale suplimentare necesare pentru rezolvarea polinoamelor cubice.

4.5.1. DETERMINAREA FUNCȚIEI DE INTERPOLARE

Fie $Q_i^j(t)$ funcția polinomială cubică pe porțiuni pentru cupla *j*, între *punctele* nodale $\mathbf{G}_i$ și $\mathbf{G}_{i+1}$, definită pe intervalul de timp $[t_i,t_{i+1}]$. Astfel, problema constă în a aproxima $Q_i^j(t)$, pentru $i = 1,2,\ldots,n-1$, împreună, astfel încât deplasarea, viteza și accelerația sunt satisfăcute și sunt continue pe întregul interval de timp $[t_1,t_n]$. Deoarece funcția polinomială $Q_i^j(t)$ este cubică, derivata de ordinul doi în raport cu timpul $\ddot{Q}_i^j(t)$ trebuie să fie o funcție liniară cu timpul *t* (fig.4.5):

$$Q_i^j(t) = \frac{t_{i+1}-t}{\tau_i}\ddot{Q}_i^j(t_i) + \frac{t-t_i}{\tau_i}\ddot{Q}_i^j(t_{i+1}),\quad i = 1,\ \ldots,\ n-1,\ \ j = 1,\ \ldots,\ N \tag{4.37}$$

unde $\tau_i = t_{i+1} - t_i$ este timpul consumat pentru parcurgerea segmentului *i*. Integrând de două ori $\ddot{Q}_i^j(t)$ și satisfăcând condițiile la limită ale $Q_i^j(t_i) = q_i^j$; $Q_i^j(t_{i+1}) = q_{i+1}^j$, rezultă următoarele funcții de interpolare pentru $i \in [1, i-1]$ și $j \in [1, N]$:

$$Q_i^j(t) = \frac{\ddot{Q}_i^j(t_i)}{6t_i}(t_{i+1}-t)^3 + \frac{\ddot{Q}_i^j(t_{i+1})}{6t_i}(t-t_i)^3 + \left[\frac{q_{i+1}^j}{t_i} - \frac{t_i\ddot{Q}_i^j(t_{i+1})}{6}\right](t-t_i) + \left[\frac{q_i^j}{t_i} - \frac{t_i\ddot{Q}_i^j(t_i)}{6}\right](t_{i+1}-t) \tag{4.38}$$

BUPT

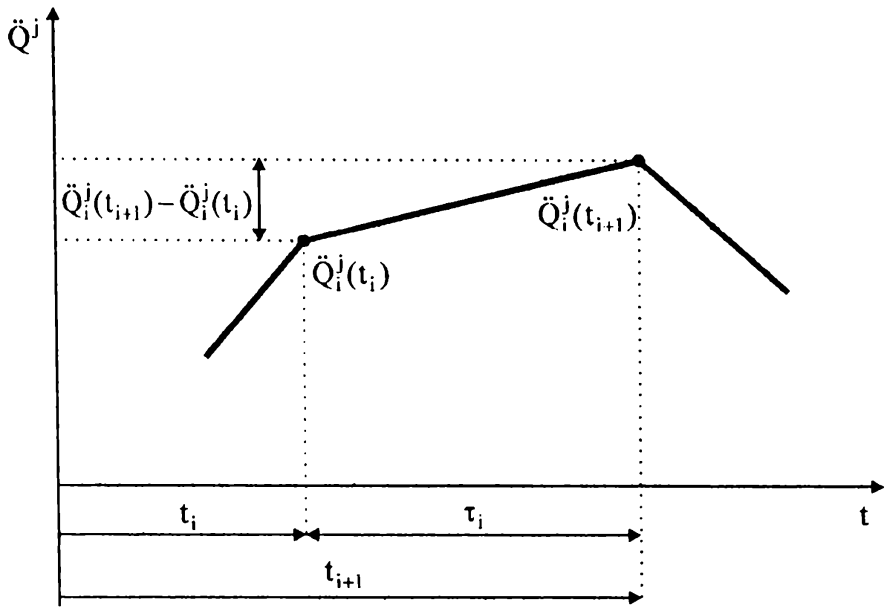


Fig.4.5. Variaţia lui $\ddot{Q}_i^j(t)$ în raport cu timpul

Astfel, pentru i = 1, 2, ..., n-1, $Q_i^j(t)$ se determină dacă se cunosc $\ddot{Q}_i^j(t_i)$ şi $\ddot{Q}_i^j(t_{i+1})$. Aceasta conduce la un sistem de *n-2* ecuaţii liniare cu necunoscutele $Q_i^j(t_i)$, i = 2, ..., n-1 timpii t_i , i = 1, 2, ..., n-1 fiind cunoscuţi. Condiţiile de continuitate pentru viteze conduc la ecuaţiile:

$$\frac{\tau_{i-1}}{\tau_i}\ddot{Q}_{i-1}^j(t_{i-1}) + \frac{2(\tau_i+\tau_{i-1})}{\tau_i}\ddot{Q}_i^j(t_i) + \ddot{Q}_i^j(t_{i+1}) = \frac{6}{\tau_i}\left[\frac{q_{i+1}^j - q_i^j}{\tau_i} - \frac{q_i^j - q_{i-1}^j}{\tau_{i-1}}\right] \tag{4.39}$$

Deplasările nespecificate ale cuplelor, în cele două noduri suplimentare, pot fi exprimate pe baza valorilor limită ale primului şi ultimului nod, împreună cu $\ddot{Q}_1^j(t_2)$ şi $\ddot{Q}_{n-2}^j(t_{n-1})$:

$$q_2^j = q_1^j + \tau_1\dot{q}_1^j + \frac{\tau_1^2}{3}\ddot{q}_1^j + \frac{\tau_1^2}{6}\ddot{Q}_1^j(t_2) \tag{4.40}$$

$$q_{n-1}^j = q_n^j + \tau_{n-1}\dot{q}_n^j + \frac{\tau_{n-1}^2}{3}\ddot{q}_n^j + \frac{\tau_{n-1}^2}{6}\ddot{Q}_{n-2}^j(t_{n-1}) \tag{4.41}$$

Înlocuind relaţiile (4.40) şi (4.41) în (4.39), pentru i=4, 5, ..., n-3 şi j=1, 2, ..., N, se obţine următorul sistem:

$$\left[\frac{2(\tau_1+\tau_2)}{\tau_2} + \frac{\tau_1^2}{\tau_2}\left(\frac{1}{\tau_1}+\frac{1}{\tau_2}\right)\right]\ddot{Q}_2^j(t_2) + \ddot{Q}_2^j(t_3) = \frac{6}{\tau_2}\left(\frac{q_3^j}{\tau_2} + \frac{q_1^j}{\tau_1}\right) - \frac{6}{\tau_2}\left(\frac{1}{\tau_1}+\frac{1}{\tau_2}\right)\left(q_1^j + \tau_1\dot{q}_1^j + \frac{\tau_1^2}{3}\ddot{q}_1^j\right) - \frac{\tau_1}{\tau_2}\ddot{q}_1^j \tag{4.42}$$

$$\left(\frac{\tau_2}{\tau_3} - \frac{\tau_1^2}{\tau_2\tau_3}\right)\ddot{Q}_2^j(t_2) + \frac{2(\tau_2+\tau_3)}{\tau_3}\ddot{Q}_3^j(t_3) + \ddot{Q}_3^j(t_4) = \frac{6}{\tau_2\tau_3}\left(q_1^j + \tau_1\dot{q}_1^j + \frac{\tau_1^2}{3}\ddot{q}_1^j\right) + \frac{6}{\tau_3^2}q_4^j - \frac{6}{\tau_3}\left(\frac{1}{\tau_2}+\frac{1}{\tau_3}\right)q_3^j \tag{4.43}$$

$$\frac{\tau_{i-1}}{\tau_i}\ddot{Q}_{i-1}^j(t_{i-1}) + \frac{2(\tau_i+\tau_{i-1})}{\tau_i}\ddot{Q}_i^j(t_i) + \ddot{Q}_i^j(t_{i+1}) = \frac{6}{\tau_i}\left(\frac{q_{i+1}^j - q_i^j}{\tau_i} - \frac{q_i^j - q_{i-1}^j}{\tau_{i-1}}\right). \tag{4.44}$$

BUPT

În relaţiile (4.42)-(4.46), $\ddot{Q}_i(t_{i+1}) = \ddot{Q}_{i+1}(t_{i+1})$ datorită condiţiilor de continuitate a acceleraţiilor.

$$\frac{\tau_{n-3}}{\tau_{n-2}}\ddot{Q}^j_{n-3}(t_{n-3}) + \frac{2(\tau_{n-2}+\tau_{n-3})}{\tau_{n-2}}\ddot{Q}^j_{n-2}(t_{n-2}) + \left(1-\frac{\tau^2_{n-1}}{\tau^2_{n-2}}\right)\ddot{Q}^j_{n-2}(t_{n-1}) =$$

$$= \frac{6}{\tau^2_{n-2}}\left(q^j_n - \dot{q}^j_n\tau_n + \frac{\tau^2_{n-1}}{3}\ddot{q}^j_n\right) - \frac{6}{\tau_{n-2}}\left(\frac{1}{\tau_{n-2}}+\frac{1}{\tau_{n-3}}\right)q^j_{n-2} + \frac{6}{\tau_{n-2}\tau_{n-3}}q^j_{n-3} \tag{4.45}$$

$$\frac{\tau_{n-2}}{\tau_{n-1}}\ddot{Q}^j_{n-2}(t_{n-2}) + \left[\frac{2(\tau_{n-1}+\tau_{n-2})}{\tau_{n-1}} + \left(\frac{1}{\tau_{n-1}}+\frac{1}{\tau_{n-2}}\right)\tau_{n-1}\right]\ddot{Q}^j_{n-2}(t_{n-1}) =$$

$$= -\frac{6}{\tau_{n-1}}\left(\frac{1}{\tau_{n-1}}+\frac{1}{\tau_{n-2}}\right)\left(q^j_n - \tau_{n-1}\dot{q}^j_n + \frac{\tau^2_{n-1}}{3}\ddot{q}^j_n\right) + \frac{6}{\tau^2_{n-1}}q^j_n + \frac{6}{\tau_{n-1}\tau_{n-2}}q^j_{n-2} - \ddot{q}^j_n \tag{4.46}$$

Relaţiile (4.42)-(4.46) formează un sistem de *n-2* ecuaţii liniare cu *n-2* necunoscute: $\ddot{Q}^j_i(t_i)$, i=2,...,n-1. Sub formă matriceală, sistemul se poate scrie:

$$\mathbf{A}\,\ddot{\mathbf{Q}} = \mathbf{B} \tag{4.47}$$

unde :

$$\ddot{\mathbf{Q}} = \begin{bmatrix} \ddot{Q}^j_2(t_2) \\ \ddot{Q}^j_3(t_3) \\ \vdots \\ \ddot{Q}^j_{n-1}(t_{n-1}) \end{bmatrix} \tag{4.48}$$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & & \mathbf{O} \\ a_{21} & a_{22} & a_{23} & & \\ & a_{32} & a_{33} & a_{34} & \\ & & & \vdots & \\ \mathbf{O} & & & a_{n-2,n-3} & a_{n-2,n-2} \end{bmatrix} \tag{4.49}$$

$$\mathbf{A} = \begin{bmatrix} 3\tau_1 + 2\tau_2 + \frac{\tau_1^2}{\tau_2} & \tau_2 & & & \mathbf{O} \\ \tau_2 - \frac{\tau_1^2}{\tau_2} & 2(\tau_2+\tau_3) & \tau_3 & & \\ & \tau_3 & 2(\tau_3+\tau_4) & \tau_4 & \\ & & & & \\ & & \cdots\ \tau_{n-3} & 2(\tau_{n-3}+\tau_{n-2}) & \tau_{n-2} - \frac{\tau^2_{n-1}}{\tau_{n-2}} \\ \mathbf{O} & & & \tau_{n-2} & 3\tau_{n-1} + 2\tau_{n-2} + \frac{\tau^2_{n-1}}{\tau_{n-2}} \end{bmatrix} \tag{4.49'}$$

BUPT

$$\mathbf{B}=\begin{bmatrix}
\frac{6}{\tau_2}\left(\frac{q_3^j}{\tau_2}+\frac{q_1^j}{\tau_1}\right)-\frac{6}{\tau_2}\left(\frac{1}{\tau_1}+\frac{1}{\tau_2}\right)\left(q_1^j+\tau_1\dot{q}_1^j+\frac{\tau_1^2}{3}\ddot{q}_1^j\right)-\frac{\tau_1}{\tau_2}\ddot{q}_1^j \\
\frac{6}{\tau_2\tau_3}\left(q_1^j+\tau_1\dot{q}_1^j+\frac{\tau_1^2}{3}\ddot{q}_1^j\right)+6\frac{q_4^j}{\tau_3^2}-\frac{6}{\tau_3}\left(\frac{1}{\tau_1}+\frac{1}{\tau_3}\right)q_3^j \\
\frac{6}{\tau_4}\left(\frac{q_5^j-q_4^j}{\tau_4}-\frac{q_4^j-q_3^j}{\tau_3}\right) \\
\frac{6}{\tau_5}\left(\frac{q_6^j-q_5^j}{\tau_5}-\frac{q_5^j-q_4^j}{\tau_4}\right) \\
\vdots \\
\frac{6}{\tau_{n-2}^2}\left(q_n^j+\tau_{n-1}\dot{q}_n^j+\frac{\tau_{n-1}^2}{3}\ddot{q}_n^j\right)-\frac{6}{\tau_{n-2}}\left(\frac{1}{\tau_{n-2}}+\frac{1}{\tau_{n-3}}\right)q_{n-2}^j+\frac{6}{\tau_{n-2}\tau_{n-3}}q_{n-3}^j \\
-\frac{6}{\tau_{n-1}}\left(\frac{1}{\tau_{n-1}}+\frac{1}{\tau_{n-2}}\right)\left(q_n^j-\tau_{n-1}\dot{q}_n^j+\frac{\tau_{n-1}^2}{3}\ddot{q}_n^j\right)+\frac{6}{\tau_{n-1}^2}q_n^j+\frac{6}{\tau_{n-1}\tau_{n-2}}q_{n-2}^j-\ddot{q}_n^j
\end{bmatrix} \tag{4.50}$$

Structura pătratică, compactă a matricei **A** uşurează rezolvarea pentru $\ddot{\mathbf{Q}}$ care este înlocuit în relaţia (4.47) pentru a obţine soluţia $\ddot{Q}_i^j(t)$. Soluţia rezultată $\ddot{Q}_i^j(t)$ este în funcţie de intervalele de timp τ_i şi de valorile date ale deplasărilor, vitezelor şi acceleraţiilor cuplelor.

4.5.2. UNICITATEA SOLUŢIEI

Traiectoria polinomială cubică la nivelul cuplelor are întotdeauna o soluţie unică. Soluţia ecuaţiei matriceale (4.47) este unică dacă şi numai dacă matricea **A** a sistemului este nesingulară. Matricea **A** exprimată prin relaţia (4.49) este întotdeauna nesingulară deoarece intervalele de timp τ_i sunt pozitive.

Se observă că toate liniile *i* ale matricei **A**, cu excepţia liniilor *2* şi *n-3* satisfac inegalitatea:

$$|a_{ii}|>\sum_{i\neq j}|a_{ij}| \tag{4.51}$$

Pentru liniile *2* şi *n-3* se disting două cazuri:

- dacă $\tau_2\geq\tau_1$ şi $\tau_{n-2}\geq\tau_{n-1}$, condiţia este îndeplinită şi matricea **A** devine strict diagonal-dominantă şi nesingulară;
- dacă $\tau_2<\tau_1$ şi $\tau_{n-2}<\tau_{n-1}$, din linia *2* se scade linia *1* înmulţită cu: $\dfrac{\tau_2-\frac{\tau_1^2}{\tau_2}}{3\tau_1+2\tau_2+\frac{\tau_1^2}{\tau_2}}$. Termenul a_{21} este eliminat, iar termenii a_{22} şi a_{23} devin:

$$a_{22}^{'}=\frac{\tau_1^2-\tau_2^2}{3\tau_1+2\tau_2+\frac{\tau_1^2}{\tau_2}}+2(\tau_2+\tau_3)\quad\text{si}\quad a_{23}^{'}=\tau_3 \tag{4.52}$$

BUPT

şi îndeplinesc condiţia (4.51) . În mod analog se demonstrează şi pentru linia *n-3*.

Astfel, matricea **A** a sistemului este echivalentă cu o matrice strict diagonal-dominantă şi sistemul (4.47) are soluţie unică.

4.5.3. OPTIMIZAREA TRAIECTORIILOR POLINOMIALE CUBICE ALE CUPLELOR

Deoarece servomotorul fiecărei cuple nu poate furniza valori nelimitate pentru forţă/moment, timpul total necesar parcurgerii traiectoriei specificate, aproximată prin polinoame cubice, este constrâns de valorile maxime ale vitezei şi acceleraţiei fiecărei cuple, respectiv ale şocului (viteza modificării acceleraţiei). Pentru a maximiza viteza traversării traiectoriei, timpul total de parcurs al manipulatorului trebuie minimizat. Acesta se poate obţine prin ajustarea intervalelor de timp τ_i dintre două *puncte* nodale adiacente supuse restricţiilor de: viteză, acceleraţie, şoc şi forţă/moment. Problema poate fi exprimată astfel:

Minimizează funcţia obiectiv:

$$T = \sum_{i=1}^{n} \tau_i \tag{4.53}$$

supusă următoarelor restricţii:

- Restricţia vitezei:

$$\left|\dot{Q}_i^j(t)\right| \le k_v^j \qquad j = 1,...,N \qquad i = 1,...,n-1 \tag{4.54}$$

- Restricţia acceleraţiei:

$$\left|\ddot{Q}_i^j(t)\right| \le k_a^j \qquad j = 1,...,N \qquad i = 1,...,n-1 \tag{4.55}$$

- Restricţia şocului:

$$\left|\dddot{Q}_i^j(t)\right| \le k_s^j \qquad j = 1,...,N \qquad i = 1,...,n-1 \tag{4.56}$$

-Restricţia torsorului (forţei/momentului):

$$\left|\Gamma^j(t)\right| \le k_\Gamma^j \qquad j = 1,...,N \tag{4.57}$$

unde T este timpul total de parcurs, iar K_v^j, K_a^j, K_s^j si K_Γ^j sunt valorile maxime ale vitezei, acceleraţiei, şocului şi forţei generalizate pentru cupla *j*.

Restricţiile de mai sus pot fi exprimate explicit.

Restricţiile vitezei: Diferenţiind relaţia (4.38) şi înlocuind $\ddot{Q}_i^j(t_i)$ şi $\ddot{Q}_i^j(t_{i+1})$ cu $\ddot{q}_i^j$ şi $\ddot{q}_{i+1}^j$, rezultă următoarele expresii pentru $\dot{Q}_i^j(t)$ şi $\ddot{Q}_i^j(t)$:

$$\dot{Q}_i^j(t) = \frac{\ddot{q}_i^j}{2\tau_i}(t_{i+1} - t)^2 + \frac{\ddot{q}_{i+1}^j}{2\tau_i}(t - t_i)^2 + \left(\frac{q_{i+1}^j}{\tau_i} - \frac{\tau_i \ddot{q}_{i+1}^j}{6}\right) - \left(\frac{q_i^j}{\tau_i} - \frac{\tau_i \ddot{q}_i^j}{6}\right) \tag{4.58}$$

$$\ddot{Q}_i^j(t) = \frac{\ddot{q}_{i+1}^j}{\tau_i}(t - t_i) - \frac{\ddot{q}_i^j}{\tau_i}(t - t_{i+1}) \tag{4.59}$$

BUPT

unde $\ddot{q}_i^j$ este acceleraţia corespunzătoare matricei $\mathbf{G}_i$, $\ddot{q}_i^j = \ddot{Q}_i^j(t_i)$, dacă momentul când $Q_i^j(t)$ trece prin $\mathbf{G}_i$ este t_i.

Valoarea absolută maximă a vitezei există la t_i, t_{i+1} sau $\overline{t_i}$, unde $\overline{t_i} \in [t_i, t_{i+1}]$ şi satisface $\ddot{Q}_i^j(\overline{t_i}) = 0$. Restricţiile vitezei devin:

$$\max\left|\dot{Q}_i^j\right| = \max\left[\left|\dot{Q}_i^j(t_i)\right|, \left|\dot{Q}_i^j(t_{i+1})\right|, \left|\dot{Q}_i^j(\overline{t_i})\right|\right] \le K_v^j,\ i = 1,\ldots,n-1 \text{ si } j = 1,\ldots,N; t \in [t_i, t_{i+1}] \quad (4.60)$$

unde:

$$\left|\dot{Q}_i^j(t_i)\right| = \left|\frac{\ddot{q}_i^j}{2}\tau_i + \frac{q_{i+1}^j - q_i^j}{\tau_i} + \frac{\left(\ddot{q}_i^j - \ddot{q}_{i+1}^j\right)\tau_i}{6}\right| \quad (4.61)$$

$$\left|\dot{Q}_i^j(t_{i+1})\right| = \left|\frac{\ddot{q}_{i+1}^j}{2}\tau_i + \frac{q_{i+1}^j - q_i^j}{\tau_i} + \frac{\left(\ddot{q}_i^j - \ddot{q}_{i+1}^j\right)\tau_i}{6}\right| \quad (4.62)$$

$$\left|\ddot{Q}_i^j(\overline{t_i})\right| = \begin{cases} \left|\dfrac{\ddot{q}_i^j\ddot{q}_{i+1}^j\tau_i}{2\left(\ddot{q}_i^j - \ddot{q}_{i+1}^j\right)} + \dfrac{q_{i+1}^j - q_i^j}{\tau_i} + \dfrac{\left(\ddot{q}_i^j - \ddot{q}_{i+1}^j\right)\tau_i}{6}\right| & \ddot{q}_i^j \ne \ddot{q}_{i+1}^j \text{ si } \overline{t_i} \in [t_i, t_{i+1}] \\ 0 & \ddot{q}_i^j = \ddot{q}_{i+1}^j \text{ sau } \overline{t_i} \notin [t_i, t_{i+1}] \end{cases} \quad (4.63)$$

Restricţiile acceleraţiei: Acceleraţia este o funcţie liniară de timp între două *puncte* nodale adiacente. Astfel, valoarea absolută maximă a acceleraţiei apare fie la momentul t_i, fie la momentul t_{i+1} şi este egală cu valoarea maximă dintre $\left\{\left|\ddot{q}_i^j\right|, \left|\ddot{q}_{i+1}^j\right|\right\}$. Astfel, restricţiile acceleraţiei devin:

$$\max\left\{\left|\ddot{q}_1^j\right|, \left|\ddot{q}_2^j\right|, \ldots, \left|\ddot{q}_n^j\right|\right\} \le K_a^j \qquad j = 1,2,\ldots,N \quad (4.64)$$

Restricţiile şocului: Şocul reprezintă viteza de modificare a acceleraţiei. Astfel, restricţiile sunt exprimate prin:

$$\left|\frac{\ddot{q}_{i+1}^j - \ddot{q}_i^j}{\tau_i}\right| \le K_s^j \qquad j = 1,2,\ldots,N;\quad i = 1,2,\ldots,n-1 \quad (4.65)$$

Restricţiile torsorului: Torsorul $\Gamma(t)$ poate fi calculat din ecuaţiile dinamice ale mişcării:

$$\Gamma^j(t) = \sum_{k=1}^{N} D_k^j(\mathbf{Q}_i(t))\ddot{Q}_i^j(t) + \sum_{k=1}^{N}\sum_{m=1}^{N} D_{km}^j(\mathbf{Q}_i(t))\dot{Q}_{ik}(t)\dot{Q}_{im}(t) + D^j(\mathbf{Q}_i(t)) \le K_l^j \quad (4.66)$$

unde:

$$\mathbf{Q}_i(t) = \left[Q_i^1(t), Q_i^2(t), \ldots, Q_i^n(t)\right]^T, \qquad j = 1,2,\ldots,N \text{ si } i = 1,2,\ldots,n-1 \quad (4.67)$$

Dacă restricţiile forţei generalizate nu sunt satisfăcute, atunci, pentru a asigura satisfacerea restricţiilor torsorului [L18], [H40], trebuie îndeplinit timpul dinamic corespunzător traiectoriei.

BUPT

Cu această formulare, obiectivul constă în a găsi un algoritm de optimizare potrivit care să minimizeze timpul total de parcurs supus restricțiilor vitezei, accelerației, șocului și torsorului. Lin [L18] a folosit cercetarea cu poliedre flexibile Nelder și Mead pentru a obține un algoritm iterativ care să minimizeze timpul total de parcurs, satisfăcînd restricțiile de viteză, accelerație, șoc și torsor.

4.5.4. EXISTENȚA SOLUȚIEI FEZABILE

Problema optimizării, supusă restricțiilor (4.60)-(4.66), are întotdeauna soluții fezabile.

Dacă sse presupun cunoscute intervalele de timp $\{\tau_1, \tau_2, ..., \tau_{n-1}\}$, în baza proprietății care se referă la unicitatea soluției sistemului (4.5.2), $\{\ddot{q}_2, \ddot{q}_3, ..., \ddot{q}_{n-1}\}$ pot fi unic determinate, însă restricțiile în viteze, accelerații și șocuri ale cuplelor este posibil să nu fie satisfăcute. În această situație, intervalele de timp $\{\tau_1, \tau_2, ..., \tau_{n-1}\}$ pot fi extinse pentru a micșora vitezele, accelerațiile și șocurile neconvenabile, conform limitelor impuse. În scopul explicării validității acestui procedeu, se consideră că $\mathbf{Q}_i(t)$ este funcția spline originală a deplasării cuplei, definită în intervalul de timp $[t_i, t_{i+1}] = [t_i, t_i+\tau_i]$. Dacă toate intervalele de timp sunt expandate în sensul $t^* = \lambda t$, astfel încât noile intervale să fie $\tau_i^* = \lambda\tau_i$, atunci noua accelerație $\ddot{q}_i^*$ poate fi determinată ca $\frac{\ddot{q}_i}{\lambda^2}$, utilizând relațiile (4.54) - (4.56). Astfel, se poate arăta că noile funcții spline $\mathbf{Q}_i^*(t^*)$, definite în intervalul de timp $[\lambda t_i, \lambda(t_i+\tau_i)] = [t_i^*, t_i^*+\tau_i^*]$, au aceeași formă ca și $\mathbf{Q}_i(t)$. De asemenea, derivatele de ordinul întâi, doi și trei în raport cu timpul ale lui $\mathbf{Q}_i^*(t)$ vor fi:

$$\frac{1}{\lambda}\dot{\mathbf{Q}}_i\left(\frac{t^*}{\lambda}\right); \quad \frac{1}{\lambda^2}\ddot{\mathbf{Q}}_i\left(\frac{t^*}{\lambda}\right); \quad \frac{1}{\lambda^3}\dddot{\mathbf{Q}}_i\left(\frac{t^*}{\lambda}\right) \tag{4.68}$$

Fie:

$$\lambda_1 = \max_j \left[\max_{t\in[t_i,t_{i+1}],\forall i} |\dot{Q}_i^j(t)| \;/\; K_v^j \right] \tag{4.69}$$

$$\lambda_2 = \max_j \left[\max_{t\in[t_i,t_{i+1}],\forall i} |\ddot{Q}_i^j(t)| \;/\; K_a^j \right] = \max_j \left[\max_i |\ddot{q}_i^j| / K_a^j \right] \tag{4.70}$$

$$\lambda_3 = \max_j \left[\max_{t\in[t_i,t_{i+1}],\forall i} |\dddot{Q}_i^j(t)| \;/\; K_s^j \right] = \max_j \left[\max_i \left| \frac{\ddot{q}_{i+1}^j - \ddot{q}_i^j}{\tau_i} \right| / K_s^j \right] \tag{4.71}$$

$$\lambda = \max(1, \lambda_1, \lambda_2^{1/2}, \lambda_3^{1/3}) . \tag{4.72}$$

Dacă intervalul de timp τ_i este înlocuit cu $\lambda\tau_i$ pentru i=1, 2, ..., n-1, atunci viteza, accelerația și șocul vor fi reduse cu ajutorul factorilor $1/\lambda$, $1/\lambda^2$, respectiv $1/\lambda^3$. Aceste schimbări asigură satisfacerea restricțiilor vitezelor, accelerațiilor și șocurilor. λ se numește *factor de ajustare a fezabilității*. Procedeul de convertire a unui punct nefezabil într-unul fezabil se numește *convertire în soluții fezabile ("the feasible solution converter" - FSC)*. În principiu, procedeul se desfășoară astfel:

- Se stabilește λ în concordanță cu relațiile (4.69) - (4.72).

BUPT

•Se înlocuiesc intervalele de timp $\{\tau_1, \tau_2, ..., \tau_{n-1}\}$ prin $\{\lambda\tau_1, \lambda\tau_2, ..., \lambda\tau_{n-1}\}$.

•Se înlocuiesc valorile lui $\left\{\tilde{q}_2^j, \tilde{q}_3^j, ..., \tilde{q}_{n-1}^j\right\}$ prin $\left\{\frac{\tilde{q}_2^j}{\lambda^2}, \frac{\tilde{q}_3^j}{\lambda^2}, ..., \frac{\tilde{q}_{n-1}^j}{\lambda^2}\right\}$, pentru j=1, 2,..., N.

4.5.5. ALGORITMUL DE OPTIMIZARE

Un vârf X este definit ca şi vector al intervalelor de timp între nodurile selectate, adică este egal cu $[\tau_1 \quad \tau_2 \quad ... \quad \tau_{n-1}]$. Funcţia obiectiv pentru X este reprezentată prin T(X) şi este egală cu $(\tau_1 + \tau_2 + ... + \tau_{n-1})$ [L18]. Pentru început sunt selectate un număr de *n* vârfuri fezabile X_i (i = 1, 2, ..., n) pentru a forma poliedrul iniţial. Înainte de aceasta însă, se consideră că X_M şi X_m au cea mai mare, respectiv, cea mai mică dintre funcţiile obiectiv. Fie X_{n+1} centroidul vârfurilor, exclusiv X_M. Acesta se calculează ca fiind:

$$X_{n+1} = \frac{1}{n-1}\left[\sum_{i=1}^{n} X_i - X_M\right] \tag{4.73}$$

Algoritmul încearcă să selecteze cel mai bun vârf fezabil (corespunzând celei mai mici valori a funcţiei obiectiv) de-a lungul liniei ce conectează X_M şi X_{n+1}, pentru a-l înlocui pe cel mai rău vârf X_M. Dacă algoritmul nu izbuteşte să găsească un vârf mai convenabil, poliedrul cu *n* vârfuri va fi restrâns la o dimensiune scăzută. Operaţiile de căutare a unui vârf mai convenabil şi de reducere a mărimii poliedrului includ reflexia, expandarea, contracţia şi reducerea.

Reflexie: X_M reflectă direct centroidul prin:

$$X_{n+2} = X_{n+1} + \alpha(X_{n+1} - X_M) \tag{4.74}$$

unde $\alpha > 0$ este *coeficientul de reflexie*. Trebuie reţinut faptul că toate elementele lui X_{n+2} sunt intervale de timp. În scopul asigurării pozitivităţii tuturor elementelor, α trebuie să fie determinat convenabil. Astfel, mai întâi se fixează α la valoarea 1. Dacă unul din elementele lui X_{n+2} nu este pozitiv, α va fi modificat la o valoare mai mică. Se notează $X^p = \left[\tau_1^p \quad \tau_2^p \quad ... \quad \tau_{n-1}^p\right]$. Pentru $\alpha = 1$, X_{n+2} se obţine astfel:

$$X_{n+2} = 2X_{n+1} - X_M = \left[2\tau_1^{n+1} - \tau_1^M \quad 2\tau_2^{n+1} - \tau_2^M \quad ... \quad 2\tau_{n-1}^{n+1} - \tau_{n-1}^M\right] \tag{4.75}$$

Toate elementele trebuie să fie pozitive, adică $2\tau_i^{n+1} - \tau_i^M > 0$ pentru toate valorile posibile ale lui *i*. Dacă $2\tau_i^{n+1} - \tau_i^M \le 0$, pentru anumite valori ale lui *i*, atunci α trebuie redus. Din relaţia (4.74) se deduce că:

$$\tau_i^{n+2} = \alpha(\tau_i^{n+1} - \tau_i^M) + \tau_i^{n+1} \tag{4.76}$$

Dacă $2\tau_i^{n+1} - \tau_i^M \le 0$, atunci $\tau_i^{n+1} - \tau_i^M \le 0$. De asemenea, α trebuie să fie mai mic decât $\tau_i^{n+1} / (\tau_i^M - \tau_i^{n+1})$ pentru a-l face pe τ_i^{n+2} pozitiv. În baza celor discutate, α poate fi determinat astfel:

BUPT

$$\alpha = \begin{cases} 1 & \text{daca } 2\tau_i^{n+1} - \tau_i^M > 0, \text{ pentru toate valorile lui } i \\ \delta_1 \min\limits_{\substack{\text{acei } i \text{ pentru care} \\ 2\tau_i^{n+1} - \tau_i^M \le 0}} \left\{ \dfrac{\tau_i^{n+1}}{\tau_i^M - \tau_i^{n+1}} \right\} & \text{daca } 2\tau_i^{n+1} - \tau_i^M \le 0, \text{ pentru anumite valori ale lui } i \end{cases} \quad (4.77)$$

unde $0<\delta_1<1$ este selectat astfel încât să păstreze $\mathbf{X}_{n+2}$ în afara limitei în care ultimul dintre elementele lui $\mathbf{X}_{n+2}$ este nul.

Expandarea: Se expandează vectorul $(\mathbf{X}_{n+2}-\mathbf{X}_{n+1})$ prin:

$$\mathbf{X}_{n+3} = \mathbf{X}_{n+1} + \gamma(\mathbf{X}_{n+2} - \mathbf{X}_{n+1}) \quad (4.78)$$

unde $\gamma>1$ este *coeficientul de expansiune*. Analog, pentru a păstra toate elementele lui $\mathbf{X}_{n+3}$ pozitive, γ poate fi determinat astfel:

$$\gamma = \begin{cases} 2 & \text{daca } 2\tau_i^{n+2} - \tau_i^{n+1} > 0, \text{ pentru toate valorile lui } i \\ \delta_2 \min\limits_{\substack{\text{acei } i \text{ pentru care} \\ 2\tau_i^{n+2} - \tau_i^{n+1} \le 0}} \left\{ \dfrac{\tau_i^{n+1}}{\tau_i^{n+1} - \tau_i^{n+2}} \right\} & \text{daca } 2\tau_i^{n+2} - \tau_i^{n+1} \le 0, \text{ pentru anumite valori ale lui } i \end{cases} \quad (4.79)$$

unde $0<\delta_2<1$.

Contracţia: Se contractă vectorul $(\mathbf{X}_M - \mathbf{X}_{n+1})$ prin:

$$\mathbf{X}_{n+4} = \mathbf{X}_{n+1} + \beta(\mathbf{X}_M - \mathbf{X}_{n+1}) \quad (4.80)$$

unde $0<\beta<1$ este *coeficientul de contracţie*.

Reducerea: Se reduc toţi vectorii $(\mathbf{X}_i - \mathbf{X}_m)$, i = 1, 2, ..., n, cu o jumătate din $\mathbf{X}_m$ prin:

$$\mathbf{X}_i \leftarrow \mathbf{X}_m + 0{,}5\cdot(\mathbf{X}_i - \mathbf{X}_m), \; i = 1,2,\ldots,n. \quad (4.81)$$

Pentru a porni căutarea, sunt selectate mai întâi *n* vârfuri iniţiale. Se notează cu $\{q_1^j, q_3^j, \ldots, q_{n-2}^j, q_n^j\}$ secvenţa deplasărilor cuplei *j*. Întrucât al doilea şi al *(n-1)*-lea nod sunt suplimentare, nu s-au fixat încă q_2^j şi q_{n-1}^j. Temporar, acestora li s-au atribuit valorile $q_2^j = (q_1^j + q_3^j)/2$ şi $q_{n-1}^j = (q_{n-2}^j + q_n^j)/2$. Astfel, limita inferioară a vectorului intervalelor de timp este estimată la:

$$\mathbf{X}' \overset{\Delta}{=} \left[\tau_1' \quad \tau_2' \quad \ldots \quad \tau_{n-1}'\right] = \left[\max_j \left(\frac{\left|q_2^j - q_1^j\right|}{K_v^j}\right) \quad \max_j \left(\frac{\left|q_3^j - q_2^j\right|}{K_v^j}\right) \quad \ldots \quad \max_j \left(\frac{\left|q_n^j - q_{n-1}^j\right|}{K_v^j}\right)\right] \quad (4.82)$$

Primul vârf $\mathbf{X}_1^0$ este selectat apoi ca şi $\mathbf{X}'$, dacă $\mathbf{X}'$ este fezabil, sau ca şi vârf fezabil convertit din $\mathbf{X}'$ prin **FSC**. Pentru a decide restul de *n-1* vârfuri $\{\mathbf{X}_2^0, \mathbf{X}_3^0, \ldots, \mathbf{X}_n^0\}$, sunt calculate mai întâi vârfurile $\{\mathbf{X}'_2, \mathbf{X}'_3, \ldots, \mathbf{X}'_n\}$ în modul următor:

$$\begin{cases} \mathbf{X}_2' = \mathbf{X}_1^0 + [d_1 \quad d_2 \quad d_2 \quad \ldots \quad d_2] \\ \mathbf{X}_3' = \mathbf{X}_1^0 + [d_2 \quad d_1 \quad d_2 \quad \ldots \quad d_2] \\ \qquad\qquad \vdots \\ \mathbf{X}_n' = \mathbf{X}_1^0 + [d_2 \quad d_2 \quad d_2 \quad \ldots \quad d_1] \end{cases} \quad (4.83)$$

BUPT

unde: $d_1 = \frac{D}{\sqrt{2}(n-1)}(\sqrt{n}+n-2)$ si $d_2 = \frac{D}{\sqrt{2}(n-1)}(\sqrt{n}-1)$ [H7], iar D este distanţa selectată. Prin urmare $\mathbf{X}_p^0$ este ales să fie sau $\mathbf{X}'_p$, dacă $\mathbf{X}'_p$ este fezabil, sau vârful fezabil convertit din $\mathbf{X}'_p$ prin **FSC**. Distanţa D afectează mărimea poliedrului iniţial.

Operaţia iterativă de reducere, reduce mărimea poliedrului flexibil iniţial şi apoi controlează paşii căutării şi abordează convergenţa. De-a lungul procesului repetat de reducere, poliedrul va deveni uneori foarte mic, soluţia fiind "prinsă" într-un punct de optim local. Pentru a executa în continuare căutarea, $\mathbf{X}_m^{(k)}$ va fi folosit pentru stabilirea unui alt nou poliedru iniţial, şi, prin urmare, un alt ciclu al procesului iterativ de reducere, care să reducă mărimea noului poliedru iniţial, va fi pornit. Mărimea poliedrului rezultă prin verificarea sumei: $\sum_i |\mathbf{X}_i - \mathbf{X}_m|$. Dacă $\sum_i |\mathbf{X}_i - \mathbf{X}_m|$ este mai mică decât mărimea predefinită e_1, un nou ciclu va fi încercat. Construirea unui nou poliedru mărit conduce la următoarele două avantaje:

- Mărimea poliedrului este crescută astfel încât pasul căutarii poate fi mărit suficient pentru a ajunge la un vârf mai bun.
- Forma poliedrului este schimbată astfel încât o nouă direcţie de căutare poate fi încercată.

Sunt comparate rezultatele obţinute din două cicluri consecutive. Ciclurile se repetă până când, între rezultate, variaţia devine nesemnificativă. Acest lucru înseamnă că diferenţa între două cicluri consecutive este mai scazută decât numărul e_2 prestabilit la o valoare mică. Algoritmul va fi descris în continuare în detaliu. În algoritm, ori de câte ori procesul iterativ ajunge la *Pasul 14*, este un ciclu iterativ complet. Ori de câte ori procesul ajunge la *Pasul 13*, se completează o etapă iterativă a acestui ciclu. Astfel, *kk* este numărul ciclului, care porneşte de la zero, şi *k* este numărul etapei, care porneşte de la zero pentru fiecare ciclu.

Iniţializare

Pasul 1: Se selectează $kk = 0$ şi $k = 0$. Se stabilesc δ_1, δ_2 şi β conform relaţiilor (4.77), (4.79) şi (4.80). De asemenea, se aleg e_1 şi e_2. Se setează $OLD\mathbf{X}_m = [0\ 0\ \ldots\ 0]$.

Calculul primului vârf al poliedrului

Pasul 2: Se calculează $\mathbf{X}'$conform relaţiei (4.82). Dacă $\mathbf{X}'$ este fezabil, atunci se setează $\mathbf{X}_1^{(k)} = \mathbf{X}'$; astfel se setează $\mathbf{X}_1^{(k)}$ = vârful fezabil convertit din $\mathbf{X}'$ prin **FSC**.

Calculul celorlalte n-1 vârfuri

Pasul 3: Pentru i = 2, 3, ..., n, se calculează $\mathbf{X}'_i$ conform relaţiei (4.83). Dacă $\mathbf{X}'_i$ este fezabil, atunci se setează $\mathbf{X}_i^{(k)} = \mathbf{X}'_i$; altfel se setează $\mathbf{X}_i^{(k)}$ = vârful fezabil convertit din $\mathbf{X}'_i$, utilizând **FSC**.

Pasul 4: Din cele *n* vârfuri ale lui $\mathbf{X}_i$:

a) se determină $X_m^{(k)}$ care are cea mai mică valoare a funcţiei obiectiv;

b) se determină $X_M^{(k)}$ care are cea mai mare valoare a funcţiei obiectiv.

Acestea sunt $T(\mathbf{X}_m^{(k)}) = \min_i T(\mathbf{X}_i^{(k)})$ şi $T(\mathbf{X}_M^{(k)}) = \max_i T(\mathbf{X}_i^{(k)})$.

Pasul 5: Se calculează centroidul $X_{n+1}^{(k)}$ cu ajutorul relaţiei (4.73).

Reflexia

Pasul 6: Se reflectă spre a se obţine $\mathbf{X}_{n+2}^{(k)}$ conform relaţiilor (4.74) şi (4.77). Se convertşte $\mathbf{X}_{n+2}^{(k)}$ pentru a deveni fezabil, prin **FSC**, dacă $\mathbf{X}_{n+2}^{(k)}$ nu este iniţial fezabil.

BUPT

Se expandează dacă $T(\mathbf{X}_{n+2}^{(k)}) < T(\mathbf{X}_m^{(k)})$

Pasul 7: Dacă $T(\mathbf{X}_{n+2}^{(k)}) < T(\mathbf{X}_m^{(k)})$, atunci execută a), b) şi c) de mai jos; altfel trece la *Pasul 8.*

a) Dacă $\lambda<1$, setează $\mathbf{X}_M^{(k)} = \mathbf{X}_{n+2}^{(k)}$, apoi setează $k = k+1$ şi trece la *Pasul 13*; altfel continuă.

$\lambda<1$ implică faptul că vârful obţinut în urma reflexiei trebuie să aibă nişte elemente care să fie foarte mici, dar totuşi cu valori pozitive. Nu trebuie încercată nici o expandare, deoarece aceste valori pot deveni negative, ceea ce violează natura fizică pozitivă a intervalelor de timp.

b) Se expandează în vederea obţinerii lui $\mathbf{X}_{n+3}^{(k)}$ cu relaţiile (4.78) şi (4.79). Se converteşte $\mathbf{X}_{n+3}^{(k)}$ pentru a deveni fezabil, prin **FSC**, dacă este iniţial nefezabil.

c) Se setează:

$$\mathbf{X}_M^{(k)} = \begin{cases} \mathbf{X}_{n+2}^{(k)} & \text{daca } T(\mathbf{X}_{n+2}^{(k)}) \le T(\mathbf{X}_{n+3}^{(k)}) \\ \mathbf{X}_{n+3}^{(k)} & \text{daca } T(\mathbf{X}_{n+2}^{(k)}) > T(\mathbf{X}_{n+3}^{(k)}) \end{cases}$$

apoi $k = k+1$ şi se trece la *Pasul 13.*

Pasul 8: Dacă $T(\mathbf{X}_{n+2}^{(k)}) \le T(\mathbf{X}_p^{(k)})$ pentru anumiţi $\mathbf{X}_p^{(k)} \neq \mathbf{X}_M^{(k)}$, se setează $\mathbf{X}_M^{(k)} = \mathbf{X}_{n+2}^{(k)}$, apoi se setează $k = k+1$ şi se trece la *Pasul 13.*

Pasul 9: Dacă $T(\mathbf{X}_{n+2}^{(k)}) < T(\mathbf{X}_M^{(k)})$, atunci se setează $\mathbf{X}_M^{(k)} = \mathbf{X}_{n+2}^{(k)}$.

Contracţia

Pasul 10: Se contractă pentru a se obţine $\mathbf{X}_{n+4}^{(k)}$ în baza relaţiei (4.80). Se converteşte $\mathbf{X}_{n+4}^{(k)}$ spre a deveni fezabil, prin **FSC**, dacă este original nefezabil.

Pasul 11: Dacă $T(\mathbf{X}_{n+4}^{(k)}) < T(\mathbf{X}_M^{(k)})$, se setează $\mathbf{X}_M^{(k)} = \mathbf{X}_{n+4}^{(k)}$ şi apoi se setează $k = k+1$ şi se trece la *Pasul 13.* În caz contrar, se continuă.

Reducerea

Pasul 12: Se reduce poliedrul setând:

$$\mathbf{X}_i^{(k)} \leftarrow \mathbf{X}_m^{(k)} + 0{,}5(\mathbf{X}_i^{(k)} - \mathbf{X}_m^{(k)}), \text{ pentru } i = 1,2,\ldots,n .$$

Se verifică dacă poliedrul este prea mic

Pasul 13: Dacă $\sum_{i=1}^{n} \left| \mathbf{X}_i^{(k)} - \mathbf{X}_m^{(k)} \right| < e_1$, atunci se trece la *Pasul 14*; altfel se trece la *Pasul 4.*

Se verifică dacă soluţiile provenite din două cicluri sunt foarte apropiate

Pasul 14: Dacă $\left| \mathbf{X}_m^{(k)} - OLD\mathbf{X}_m \right| < e_2$, atunci se reţine $\mathbf{X}_m^{(k)}$ ca fiind răspunsul şi STOP; altfel se setează $\mathbf{X}_1^0 = \mathbf{X}_m^{(k)}$, $OLD\mathbf{X}_m = \mathbf{X}_m^{(k)}$. Apoi se setează $kk = kk+1$, $k = 0$ şi se trece la *Pasul 3.*

Algoritmul de optimizare converteşte întotdeauna vârfurile nefezabile într-unele fezabile, printr-o procedură simplă, numită **FSC**, care execută conversia prin multiplicarea vârfului nefezabil cu un scalar λ. Pentru acestă conversie este necesar un timp de execuţie foarte mic.

La începutul fiecărui ciclu sunt selectate iniţial n vârfuri astfel ca oricare din cele $n-1$ vârfuri să fie liniar independente. Un nou vârf care să înlocuiască $\mathbf{X}_M$ este întotdeauna o combinaţie liniară de $\mathbf{X}_M$ şi celelalte $n-1$. Prin urmare, oricare din cele $n-1$ vârfuri din noul set de n vârfuri este încă liniar independent. Aranjamentul evită, de asemenea, posibilitatea căutării într-un subspaţiu.

BUPT

4.6. GENERAREA TRAIECTORIILOR POLINOMIALE CVARTICE CU ERORI PĂTRATICE MINIME ALE CUPLELOR

Pentru a construi *N* funcţii de aproximare a traiectoriilor cuplelor cinematice trebuie să se selecteze un număr suficient de *puncte* pe traiectoria carteziană şi apoi să se transforme aceste *puncte* în deplasări ale celor *N* cuple cinematice. Funcţia de aproximare pentru traiectoria unei cuple trece prin *punctele* corespunzătoare selectate în coordonate carteziene. O soluţie posibilă pentru această problemă este folosirea funcţiilor polinomiale de grad mic, aproximarea pe porţiuni şi concatenarea acestora. Paul [P3] şi Finkel [F2] au investigat calculul traiectoriilor prin interpolarea punctelor nodale. Metodele propuse necesită rezolvarea unui sistem de *3(n-1)* sau *4(n-1)* ecuaţii liniare, unde *n* este numărul punctelor selectate. O metodă mai simplă constă în rezolvarea unui sistem cu *n-2* ecuaţii pentru a aproxima *n-1* funcţii polinomiale pentru fiecare cuplă [L18].

Deşi este evitată problema calcului on-line, în generarea traiectoriei apar totuşi erori datorită aproximării polinomiale a acesteia. Un remediu simplu pentru a evita acest lucru este sporirea numărului de puncte selectate, fapt care are drept urmare mărirea numărului polinoamelor de interpolare care, concatenate, formează traiectoria totală, continuă, şi deci, mărirea volumului de calcule. O situaţie ideală ar fi creşterea numărului de noduri, dar nu şi a numărului de polinoame. În acest caz însă, condiţiile la limită în nodurile de concatenare nu pot fi simultan satisfăcute. O alternativă la această problemă poate fi utilizarea erorii medii.

În acest subcapitol se propune o metodă care minimizează suma erorilor pătratice folosind un număr *m* de *puncte* selectate (noduri) şi *n-1* polinoame cvartice în generarea traiectoriilor polinomiale ale cuplelor, unde m > n [L29]. În interpolările uzuale nu este posibilă interpolarea simultană a tuturor celor *m* noduri în acelaşi timp şi "forţarea" funcţiei de aproximare să satisfacă cerinţele de continuitate şi condiţiile de limită ale poziţiilor, vitezelor şi acceleraţiilor. În acest caz, se renunţă la necesitatea trecerii prin punctele selectate şi se utilizează eroarea pătratică minimă drept criteriu de concatenare a celor *n-1* polinoame cvartice într-o traiectorie totală.

Fie $t_1 < t_2 < ... < t_{n-1} < t_n$ secvenţa ordonată a momentelor de timp. La momentele t_2, t_3, ..., t_{n-1}, vor fi concatenate un număr de *n-1* polinoame cvartice, dar nu va fi necesară traversarea exactă a punctelor selectate corespunzătoare acestor momente. Fiecare funcţie de aproximare pentru o traiectorie a unei cuple va fi generată convenabil din punctul de vedere al erorii pătratice minime, respectându-se deplasările cuplei corespunzătoare celor *m* noduri selectate. Deci, condiţiile care se impun sunt:

- (1) condiţii de continuitate a poziţiilor, vitezelor şi acceleraţiilor;
- (2) condiţii la limită în momentele de timp iniţial şi final.

Mulţimea funcţiilor formate prin concatenarea a *n-1* polinoame cvartice, la momentele de timp t_2, t_3, ..., t_{n-1} şi care satifac cerinţele (1), formează un spaţiu *S*. Fiecare polinom cvartic are cinci coeficienţi. Astfel, pentru *n-1* polinoame cvartice sunt *5(n-1)* coeficienţi. Pentru a fi satisfăcută condiţia (1), sunt necesare *3(n-2)* restricţii. În continuare, dimensiunea spaţiului *S* este calculată ca şi *5(n-1) - 3(n-2) = 2n + 1*.

Problema aproximării pe baza criteriului erorii pătratice minime constă în a selecta o bază pentru *S*, în a reprezenta o funcţie în *S* ca o combinaţie liniară a elementelor bazei, şi apoi, în a impune condiţiile (2) şi a determina funcţia de aproximare obţinută pentru eroarea pătratică minimă. Numărul gradelor de libertate ale funcţiei de aproximare cvartice,

BUPT

determinate în baza criteriului amintit, este de *2n-5*, mai mare deci, decât în cazul funcţiilor de aproximare cubice unde este *n-4*. Aceste grade de libertate ale funcţiei pot fi ajustate în scopul minimizării sumei erorilor pătratice.

Restricţiile de continuitate pot fi construite în interiorul bazei lui S. Pentru a forma baza, se consideră pentru i = 0, ..., n+1:

$$B = \{B_0, B_1, ..., B_n, B_{n+1}, K_2, ..., K_n\} \tag{4.84}$$

unde:

$$B_i(t) = \begin{cases} (t_{i+2} - t)^3 & t_{i+1} < t \le t_{i+2} \\ (t_{i+2} - t)^3 + a_{1i}(t_{i+1} - t)^3 & t_i < t \le t_{i+1} \\ (t_{i+2} - t)^3 + a_{1i}(t_{i+1} - t)^3 + a_{2i}(t_i - t)^3 & t_{i-1} < t \le t_i \\ (t_{i+2} - t)^3 + a_{1i}(t_{i+1} - t)^3 + a_{2i}(t_i - t)^3 + a_{3i}(t_{i-1} - t)^3 & t_{i-2} < t \le t_{i-1} \\ 0 & \text{toti ceilalti t} \end{cases} \tag{4.85}$$

cu a_{1i}, a_{2i}, a_{3i} satisfăcând:

$$(t_{i+2} - t_{i-2})^k + a_{1i}(t_{i+1} - t_{i-2})^k + a_{2i}(t_i - t_{i-2})^k + a_{3i}(t_{i-1} - t_{i-2})^k = 0,\ k = 1,2,3. \tag{4.86}$$

Se menţionează că nodurile suplimentare $t_{-2}<t_{-1}<t_0<t_1$ şi $t_{n+3}>t_{n+2}>t_{n+1}>t_n$ sunt introduse astfel încât toate funcţiile $B_i(t)$ sunt bine definite. $B_i(t)$, $\dot{B}_i(t)$ şi $\ddot{B}_i(t)$ sunt continue pe intervalul $[t_{i-2}, t_{i+2}]$, şi $B_i(t)$ dispare la momentele t_{i+2} şi t_{i-2}. De asemenea, se consideră pentru i = 2, ..., n:

$$K_i(t) = \begin{cases} (t_{i+2} - t)^4 & t_{i+1} < t \le t_{i+2} \\ (t_{i+2} - t)^4 + b_{1i}(t_{i+1} - t)^4 & t_i < t \le t_{i+1} \\ (t_{i+2} - t)^4 + b_{1i}(t_{i+1} - t)^4 + b_{2i}(t_i - t)^4 & t_{i-1} < t \le t_i \\ (t_{i+2} - t)^4 + b_{1i}(t_{i+1} - t)^4 + b_{2i}(t_i - t)^4 + b_{3i}(t_{i-1} - t)^4 & t_{i-2} < t \le t_{i-1} \\ 0 & \text{toti ceilalti t} \end{cases} \tag{4.87}$$

cu b_{1i},b_{2i},b_{3i} satisfăcând:

$$(t_{i+2} - t_{i-2})^k + b_{1i}(t_{i+1} - t_{i-2})^k + b_{2i}(t_i - t_{i-2})^k + b_{3i}(t_{i-1} - t_{i-2})^k = 0,\ k = 2,3,4 \tag{4.88}$$

Dacă cele şase condiţii de limită sunt restrânse, atunci gradele de libertate cresc de la *2n-5* la *2n+1* cât este dimensiunea spaţiului *S*.

Se exprimă $P(t) \in S$ în funcţie de elementele bazei obţinute, astfel:

$$P(t) = \sum_{i=0}^{n+1} C_i B_i(t) + \sum_{i=2}^{n} D_i K_i(t) \tag{4.89}$$

$$P(t) = C_0 B_0 + C_1 B_1 + C_2 B_2 + D_2 K_2 + ... + C_n B_n + D_n K_n + C_{n+1} B_{n+1} \tag{4.89'}$$

unde C_i şi D_i sunt constante care trebuie să fie determinate. Se menţionează că doar B_0, B_1, B_2, K_2 sunt nenule în momentul iniţial t_1, iar B_{n-1}, K_{n-1}, B_n, K_n, B_{n+1} sunt nenule în momentul

BUPT

final t_n.Prin urmare, există doar un grad de libertate dat de C_0, C_1, C_2, D_2 şi două grade date de C_{n-1}, D_{n-1}, C_n, D_n şi C_{n+1}. Relaţiile lor de dependenţă pot fi exprimate:

$$\left[\sum_{i=0}^{2} C_i \frac{d^k B_i}{dt^k} + D_2 \frac{d^k K_2}{dt^k}\right]_{t_1} = \begin{cases} \text{pozitia limita la } t_1 & k = 0 \\ \text{viteza limita la } t_1 & k = 1 \\ \text{acceleratia limita la } t_1 & k = 2 \end{cases} \tag{4.90}$$

$$\left[\sum_{i=n-1}^{n+1} C_i \frac{d^k B_i}{dt^k} + \sum_{j=n-1}^{n} D_j \frac{d^k K_j}{dt^k}\right]_{t_n} = \begin{cases} \text{pozitia limita la } t_n & k = 0 \\ \text{viteza limita la } t_n & k = 1 \\ \text{acceleratia limita la } t_n & k = 2 \end{cases} \tag{4.91}$$

Utilizând relaţiile (4.90) şi (4.91), pot fi exprimate constantele C_0, C_1, C_2 în funcţie de D_2, respectiv C_n, D_n, C_{n+1} în funcţie de C_{n-1} şi D_{n-1}, astfel:

$$\begin{aligned} C_0 &= e_{11} D_2 + e_{12} \\ C_1 &= e_{21} D_2 + e_{22} \\ C_2 &= e_{31} D_2 + e_{32} \\ C_n &= r_{11} C_{n-1} + r_{12} D_{n-1} + r_{13} \\ D_n &= r_{21} C_{n-1} + r_{22} D_{n-1} + r_{23} \\ C_{n-1} &= r_{31} C_{n-1} + r_{32} D_{n-1} + r_{33} \end{aligned} \tag{4.92}$$

unde $e_{\alpha\beta}$ şi $r_{\alpha\gamma}$, pentru $\alpha = 1, 2, 3$, $\beta = 1, 2$, $\gamma = 1, 2, 3$, sunt combinaţii de valori cunoscute ale lui B_i, K_j şi derivatele lor, astfel încât relaţiile (4.92) sunt echivalente relaţiilor (4.90) şi (4.91). În continuare, P(t) poate fi scris:

$$\begin{aligned} P(t) &= g(t) + D_2 \underline{K}_2(t) + C_3 B_3(t) + D_3 K_3(t) + C_4 B_4(t) + D_4 K_4(t) + \\ &\ldots + C_{n-2} B_{n-2}(t) + D_{n-2} K_{n-2}(t) + C_{n-1} \underline{B}_{n-1}(t) + D_{n-1} \underline{K}_{n-1}(t) \end{aligned} \tag{4.93}$$

unde:

$$g(t) = e_{12} B_0(t) + e_{22} B_1(t) + e_{32} B_2(t) + r_{13} B_n(t) + r_{23} K_n(t) + r_{33} B_{n+1}(t) \tag{4.94}$$

$$\underline{K}_2(t) = K_2(t) + e_{11} B_0(t) + e_{21} B_1(t) + e_{31} B_2(t) \tag{4.95}$$

$$\underline{B}_{n-1}(t) = B_{n-1}(t) + r_{11} B_n(t) + r_{22} K_n(t) + r_{31} B_{n+1}(t) \tag{4.96}$$

$$\underline{K}_{n-1}(t) = K_{n-1}(t) + r_{12} B_n(t) + r_{22} K_n(t) + r_{32} B_{n+1}(t) \tag{4.97}$$

Astfel, suma erorilor pătratice este:

$$\begin{aligned} E = \sum_{i=1}^{m} \{ &\underline{P}_i - g(\tau_i) - [D_2 \underline{K}_2(\tau_i) + C_3 B_3(\tau_i) + D_3 K_3(\tau_i) + C_4 B_4(\tau_i) + \ldots \\ &\ldots + D_{n-2} K_{n-2}(\tau_i) + C_{n-1} \underline{B}_{n-1}(\tau_i) + D_{n-1} \underline{K}_{n-1}(\tau_i)] \}^2 \end{aligned} \tag{4.98}$$

unde τ_i este momentul de timp corespunzător lui $\underline{P}_i$ efectiv atins şi unde coeficienţii D_2, C_3, D_3, C_4, ..., C_{n-2}, C_{n-1} şi D_{n-1} se vor determina prin minimizarea sumei E. Fie:

BUPT

$$\tilde{\mathbf{B}} = \begin{bmatrix} \underline{K}_2(\tau_1) & B_3(\tau_1) & K_3(\tau_1) & \cdots & K_{n-2}(\tau_1) & \underline{B}_{n-1}(\tau_1) & \underline{K}_{n-1}(\tau_1) \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ \underline{K}_2(\tau_m) & B_3(\tau_m) & K_3(\tau_m) & \cdots & K_{n-2}(\tau_m) & \underline{B}_{n-1}(\tau_m) & \underline{K}_{n-1}(\tau_m) \end{bmatrix} \quad (4.99)$$

$$\tilde{\mathbf{Y}} = \begin{bmatrix} \underline{P}_1 - g(\tau_1) \\ \underline{P}_2 - g(\tau_2) \\ \vdots \\ \underline{P}_m - g(\tau_m) \end{bmatrix} \quad (4.100)$$

Pentru a minimiza suma erorilor pătratice E, se derivează această sumă în raport cu fiecare dintre coeficienţii care s-au menţionat că vor fi determinaţi prin minimizare. Derivarea se face pe rând, în raport cu fiecare coeficient separat şi apoi fiecare expresie rezultată se egalează cu zero. Astfel, coeficienţii pot fi obţinuţi prin rezolvarea sistemului:

$$\begin{bmatrix} D_2 & C_3 & D_3 & C_4 & \cdots & D_{n-2} & C_{n-1} & D_{n-1} \end{bmatrix}^T = (\tilde{\mathbf{B}}^T\tilde{\mathbf{B}})^{-1}\tilde{\mathbf{B}}^T\tilde{\mathbf{Y}} \quad (4.101)$$

Pentru un robot cu *N* cuple cinematice, algoritmul prezentat pentru o singură cuplă cinematică se aplică de *N* ori, rezultând *N* funcţii de aproximare.

Metoda prezentată pentru funcţii polinomiale cvartice poate fi aplicată şi pentru funcţii polinomiale cubice [L29]. În cazul funcţiilor cubice, comparativ cu cel al funcţiilor cvartice, timpul de calcul este mai mic, dar eroarea este mai mare.

De asemenea, dacă se alege un număr de puncte *m* mai mare, se reduce eroarea, dar se majorează timpul de execuţie.

4.7. INTERPOLAREA UNEI TRAIECTORII PRIN "GLISAREA" UNEI CUBICE CE TRECE PRIN PATRU *PUNCTE*

Folosirea transformatei Fourier pentru modelarea mişcării şi pentru comprimarea datelor conduce la traiectorii netede şi destul de apropiate de cele corespunzătoare din faza de învăţare. De asemenea, interpolările polinomiale, respectiv funcţiile spline permit obţinerea unor traiectorii care trec prin *punctele* nodale, cu asigurarea continuităţii derivatelor până la ordinul *n-1*.

Majoritatea metodelor, printre care şi cele amintite mai sus, prezintă însă două inconveniente când sunt folosite pentru programarea unui robot:

- necesitatea cunoaşterii tuturor punctelor de definire a traiectoriei (abordare globală), ceea ce implică o capacitate de memorie foarte importantă şi impune învăţarea în două faze: achiziţia datelor şi comprimarea lor;
- imposibilitatea modificării unei porţiuni de traiectorie fără a influenţa restul traiectoriei.

Se pune astfel problema generării unei traiectorii, în manieră locală, care să ofere o mai mare flexibilitate curbei obţinute şi care să necesite un timp de calcul redus.

Se consideră că între cele două configuraţii, iniţială $\mathbf{q}_0$ şi finală $\mathbf{q}_m$, există un număr oarecare de puncte intermediare $\mathbf{q}_k$, $k \in [1, m-1]$. Vectorul coordonatelor cuplelor

BUPT

cinematice este n-dimensional. La un moment dat kT (T este perioada de eşantionare), configuraţia unei cuple cinematice se exprimă prin q_k^j, $k\in[0, m]$, $j\in[1, n]$.

Se propune o metodă iterativă ce constă într-o abordare locală a interpolării [C19], [L15] şi permite obţinerea unei curbe cubice între două configuraţii q_{k-1}^j şi q_k^j, luând în considerare doar: q_{k-2}^j, q_{k-1}^j, q_k^j şi q_{k+1}^j. După parcurgerea algoritmului metodei pentru cele patru *puncte* considerate (calculul coeficienţilor polinomiali), se păstrează ultimele trei *puncte* (q_{k-1}^j, q_k^j şi q_{k+1}^j) la care se adaugă q_{k+2}^j şi se reia algoritmul rezultând o nouă cubică între q_k^j şi q_{k+1}^j, care se racordează cu precedenta. Generarea traiectoriei rezultă astfel prin "glisarea" de la stânga la dreapta a unei cubice ce trece prin *punctele* nodale.

Se consideră deci o traiectorie ce trebuie să treacă printr-un număr de *puncte* succesive (fig.4.6). În abscisă se consideră timpul, în perioade de eşantionare T, iar în ordonată se consideră coordonata generalizată a cuplei cinematice. Fie q_{k-2}^j, q_{k-1}^j, q_k^j şi q_{k+1}^j cele patru configuraţii succesive şi $q^j(p)$ expresia parametrică a cubicei, cu parametrul normalizat p:

$$p = \frac{t-(k-1)T}{T} \tag{4.102}$$

$$q^j(p) = \sum_{i=1}^{3} C_i^j p^i \tag{4.103}$$

Coeficienţii polinomiali pot fi determinaţi din următoarele condiţii:

$$p = 0; \quad q^j(0) = q_{k-1}^j = C_0^j; \quad \dot{q}^j(0) = \dot{q}_{k-1}^j = C_1^j \tag{4.104}$$

$$p = 1; \quad q^j(1) = q_k^j = \sum_{i=0}^{3} C_i^j; \quad \dot{q}^j(1) = \dot{q}_k^j = \sum_{i=1}^{3} C_i^j \tag{4.105}$$

În relaţiile (4.104) şi (4.105) s-a considerat p=0 pentru configuraţia k-1 şi p=1 pentru configuraţia k.

Pentru derivatele $\dot{q}_{k-1}^j$ şi $\dot{q}_k^j$ se foloseşte aproximarea Overhauser:

$$\dot{q}_{k-1}^j = \frac{q_k^j - q_{k-2}^j}{2}; \quad \dot{q}_k^j = \frac{q_{k+1}^j - q_{k-1}^j}{2} \tag{4.106}$$

Expresiile coeficienţilor polinomiali sunt:

$$\begin{cases} C_0^j = q_{k-1}^j \\ C_1^j = \frac{1}{2}\left(q_k^j - q_{k-2}^j\right) \\ C_2^j = \frac{1}{2}\left(2q_{k-2}^j - 5q_{k-1}^j + 4q_k^j - q_{k+1}^j\right) \\ C_3^j = \frac{1}{2}\left(-q_{k-2}^j + 3q_{k-1}^j - 3q_k^j + q_{k+1}^j\right) \end{cases} \tag{4.107}$$

Folosind relaţiile (4.107) se poate determina la fiecare interval de timp kT valoarea coordonatei q^j, luând în considerare curba ce trece prin punctele k-2, ..., k+1. Cunoscând

BUPT

valorile lui q^j corespunzătoare perioadelor de eşantionare cuprinse între q^j_{k-2} şi q^j_{k-1}, se calculează o cubică ce trece prin punctele k-1, k, k+1 şi k+2. Cu această cubică se calculează valorile lui q^j dintre k-1 şi k. Glisarea de la stânga la dreapta a cubicei se realizează prin reajustarea coeficienţilor polinomiali C_i.

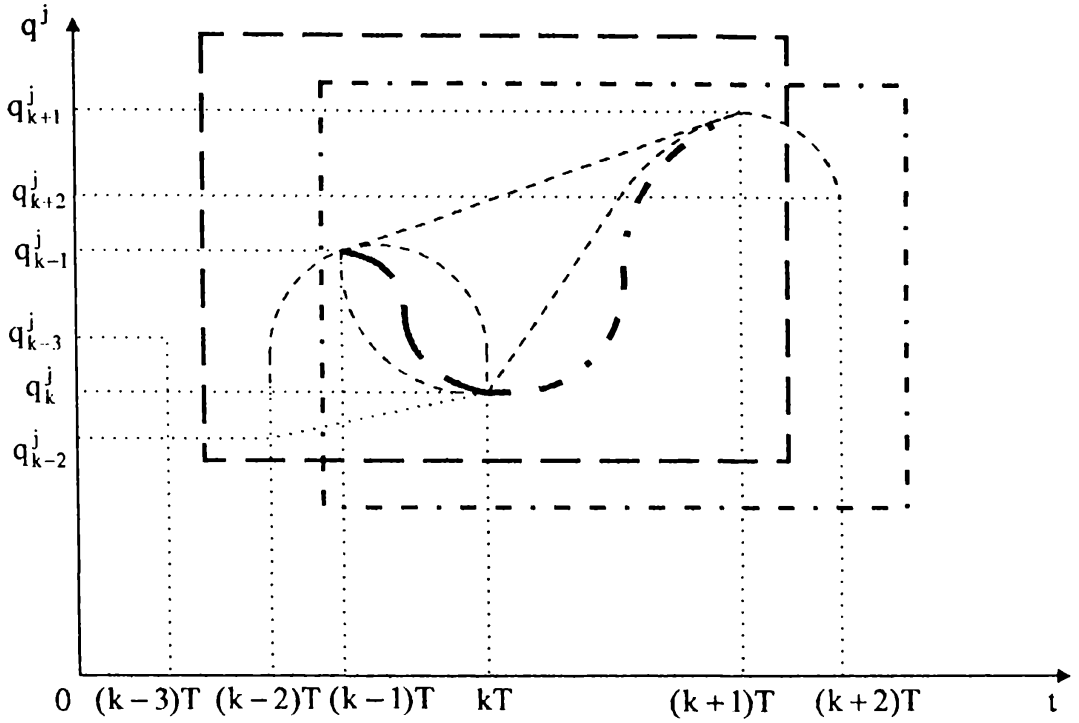


Fig.4.6. Glisarea cubicei definite prin patru *puncte*

Dacă se consideră prima cubică şi se impune p=0 pentru numărul perioadei de eşantionare corespunzătoare lui q_{k-2}, rezultă valorile variabilei q, cuprinse între k-2 şi k-1:

$$p=0:\quad q_{k-2,0}=q_{k-1}+\frac{1}{2}(q_k-q_{k-2})p_{k-2}+C_2p_{k-2}^2+C_3p_{k-2}^3$$

$$p=1:\quad q_{k-2,1}=q_{k-1}+\frac{1}{2}(q_k-q_{k-2})(p_{k-2}+T)+C_2(p_{k-2}+T)^2+C_3(p_{k-2}+T)^3 \qquad (4.108)$$

$$p=m:\quad q_{k-2,m}=q_{k-1}+\frac{1}{2}(q_k-q_{k-2})(p_{k-2}+mT)+C_2(p_{k-2}+mT)^2+C_3(p_{k-2}+mT)^3$$

În acelaşi mod se procedează şi pentru celelalte intervale: se impune p=0 pentru numărul perioadei de eşantionare corespunzătoare lui q_{k-1}, rezultând valorile variabilei q, cuprinse între k-1 şi k, etc:

$$p=0:\quad q_{k-1,0}=q_k+\frac{1}{2}(q_{k+1}-q_{k-1})p_{k-1}+C_2p_{k-1}^2+C_3p_{k-1}^3$$

$$p=1:\quad q_{k-1,1}=q_k+\frac{1}{2}(q_{k+1}-q_{k-1})(p_{k-1}+T)+C_2(p_{k-1}+T)^2+C_3(p_{k-1}+T)^3 \qquad (4.109)$$

BUPT

$$p = m: \quad q_{k-1,m} = q_k + \frac{1}{2}(q_{k+1} - q_{k-1})(p_{k-1} + mT) + C_2(p_{k-1} + mT)^2 + C_3(p_{k-1} + mT)^3$$

Metoda prezentată permite interpolarea între *puncte* relativ îndepărtate şi asigură continuitatea vitezelor. Curba obţinută trece riguros prin punctele selectate corespunzătoare perioadelor de eşantionare, asociind câte un arc cubic fiecărui interval. Dacă este tolerată o anumită eroare faţă de traiectoria dorită, se poate conserva aceeaşi ecuaţie cubică pentru mai multe perioade de eşantionare. De asemenea, se poate aplica principiul metodei şi în condiţiile în care se impune continuitatea acceleraţiilor, caz în care sunt necesare mai multe *puncte* precum şi un polinom de grad superior.

4. 8. CONCLUZII

Funcţiile spline nu mai constituie un subiect nou al analizei matematice. În ultimii ani, teoria şi aplicarea lor în diferite domenii ale analizei numerice au cunoscut o dezvoltare considerabilă. Datorită caracterului lor polinomial, funcţiile spline prezintă avantajul unei bune implementări în orice limbaj de programare. În rezolvarea problemelor de generare a mişcării roboţilor, funcţiile spline sunt tot mai mult folosite în procesele de interpolare, la aproximarea optimală a unor funcţionale liniare.

Cele mai recente limbaje de programare la nivel robot, "off-line", permit utilizatorului să specifice traiectoria printr-o succesiune de *puncte* intermediare. Efectorul final al robotului trebuie să treacă prin aceste *puncte* (sau prin vecinătatea lor), impuse într-o manieră coordonată. Interpolarea traiectoriilor se realizează pe porţiuni. Conectarea segmentelor de traiectorie se realizează cu asigurarea continuităţii poziţiei, vitezei şi acceleraţiei.

Folosirea funcţiilor spline are următoarele avantaje:

- deoarece traiectoria rezultată este continuă până la un anumit grad, mişcarea poate fi echilibrată dinamic;
- deoarece trebuie reţinuţi doar coeficienţii polinomiali, capacitatea de memorie necesară este mică.

Dezavantajele importante sunt:

- traiectoria rezultată este imprevizibilă şi se abate semnificativ de la conectarea liniară a poziţiilor impuse;
- determinarea coeficienţilor polinomiali necesită cunoaşterea tuturor punctelor intermediare, generarea traiectoriei putând începe doar după specificarea completă a acesteia.

Interpolarea traiectoriilor cu *puncte de control* consideră, între cele două puncte, iniţial şi final, încă două *puncte* intermediare. Deoarece utilizarea unei funcţii polinomiale de grad ridicat nu este satisfăcătoare, se propune interpolarea traiectoriei folosind funcţii polinomiale diferite, de grad mai mic. Au fost prezentate astfel trei tipuri de interpolări: *4-3-4, 3-5-3* şi *3-3-3-3-3*. În toate cele trei situaţii prezentate, restricţiile cinematice sunt aceleaşi.

Funcţiile spline cubice, de altfel cele mai utilizate, oferă mai multe avantaje, printre care un grad scăzut al funcţiei polinomiale, ce permite continuitatea vitezelor şi acceleraţiilor şi care reduce calculul şi posibilitatea instabilităţii numerice. Metoda de

BUPT

interpolare *3-3-3-3-3* introduce încă două noduri, pentru a avea suficiente condiţii la limită, necesare determinării coeficienţilor polinomiali. Cele două noduri suplimentare nu trebuie cunoscute, ci doar intervalele de timp corespunzătoare acestora..

Folosind cele trei metode cu *puncte de control*, dacă se impun cele patru noduri şi timpii necesari parcurgerii fiecărui segment de traiectorie, se pot determina unic funcţiile de interpolare care satisfac restricţiile cinematice impuse.

O generalizare a metodelor cu puncte de control este dată de metoda de interpolare prin *n puncte de precizie*: *4-3-...-3-4*. Metoda a fost aplicată pentru generarea mişcării robotului RTRTTR, dar poate fi folosită pentru orice robot. Algoritmul prezentat este simplu şi nu necesită un timp mare de calcul.

Interpolarea traiectoriilor folosind funcţiile spline poate fi utilizată atât în spaţiul cuplelor, cât şi în spaţiul operaţional. Interpolarea în spaţiul cuplelor cinematice prezintă avantajul impunerii restricţiilor dinamice. Metoda prezentată asigură respectarea restricţiilor cinematice şi dinamice şi minimizarea timpului total de parcurs.

O altă metodă prezentată în acest capitol constă în generarea traiectoriilor polinomiale cvartice, cu erori pătratice minime ale cuplelor. În acest caz, traiectoria nu mai trece prin nodurile de interpolare, deci este necesar să se ia în considerare abaterea de la traiectoria prescrisă. Criteriul de optimizare folosit se bazează pe utilizarea erorii pătratice minime.

Metoda prezentată pentru funcţii polinomiale cvartice poate fi aplicată şi pentru funcţii polinomiale cubice. În acest caz, timpul de calcul este mai mic, dar abaterea de la traiectoria prescrisă este mai mare. De asemenea, dacă se alege un număr mai mare de *puncte*, se reduce eroarea, dar se majorează timpul de execuţie.

Pentru a evita cunoaşterea "a priori" a tuturor nodurilor de interpolare, precum şi pentru a oferi o mai mare flexibilitate traiectoriei generate, a fost propusă metoda de interpolare prin "glisarea" unei cubice, prin patru *puncte*.

BUPT

5. GENERAREA TRAIECTORIEI UNUI ROBOT PE BAZA MODELULUI DINAMIC

5.1. MODELUL DINAMIC AL UNUI MANIPULATOR

Modelul dinamic al unui sistem mecanic constă în relaţiile (ecuaţiile diferenţiale) care există între coordonatele cuplelor cinematice, derivatele lor şi forţele şi/sau momentele exterioare şi de legătură care acţionează asupra fiecărui corp din sistem. Pentru majoritatea roboţilor actuali, forţele apar datorită greutăţii elementelor robotului, sarcinii sale şi frecărilor existente în sistemele de transmitere a mişcării. Determinarea ecuaţiilor diferenţiale ale mişcării unui robot, considerat ca un sistem de corpuri, se poate face utilizând diferite formalisme matematice: Newton-Euler, Gibbs, Lagrange.

Ecuaţiile dinamice ale unui manipulator cu *n* grade de libertate reprezintă un sistem de ecuaţii diferenţiale, neliniare, de gradul 2. Ecuaţiile dinamice exprimă legătura între torsorul forţelor exterioare (**F**(t), **M** (t)) şi evoluţia situărilor $\mathfrak{R}(t)$, în raport cu sistemul de referinţă fix $\mathfrak{R}_0$. Cunoscând evoluţia $\mathfrak{R}(t)$ se poate deduce torsorul forţelor exterioare care trebuie aplicate elementelor robotului pentru generarea mişcării dorite (**MDD**). Invers, cunoscând torsorul forţelor exterioare şi starea iniţială ($\mathfrak{R}(t_0)$, $\dot{\mathfrak{R}}(t_0)$) a robotului, se poate calcula, prin integrarea ecuaţiilor diferenţiale, starea ($\mathfrak{R}(t)$, $\dot{\mathfrak{R}}(t)$) la un moment dat $t>t_0$ (**MDI**).

Deci, utilizarea ecuaţiilor diferenţiale ale mişcării este posibilă în două moduri diferite:

- model dinamic direct (**MDD**);
- model dinamic invers (**MDI**).

În robotică, modelul dinamic este folosit în concepţie, simulare şi comandă [T9], [T10].

Modelul dinamic direct permite calculul, în fiecare moment, al forţelor şi momentelor active dezvoltate de servomotoarele robotului pentru a asigura mişcarea dorită. Dacă se impune o mişcare test, utilizarea modelului dinamic direct permite analizarea, într-o manieră mai precisă, a eforturilor cerute servomotoarelor şi sistemului mecanic, iar dacă este cazul, revizuirea proiectului (fig.5.1).

Modelul dinamic invers permite analiza comportamentului robotului echipat cu mecanisme de acţionare (fig.5.2). Astfel, este posibil să se evalueze erorile dinamice în spaţiul cuplelor şi în spaţiul operaţional (spaţiul situărilor) şi sensibilitatea acestor erori la alegerea parametrilor robotului şi a mecanismelor sale de acţionare.

Utilizarea unui model dinamic direct ridică probleme referitoare la derivările şi integrările numerice utilizate. Pentru obţinerea unor performanţe mai bune şi pentru reducerea erorilor dinamice, cele două modele dinamice, direct şi invers, pot fi puse în serie, comparând astfel torsorul de intrare cu cel de ieşire (fig.5.3).

BUPT

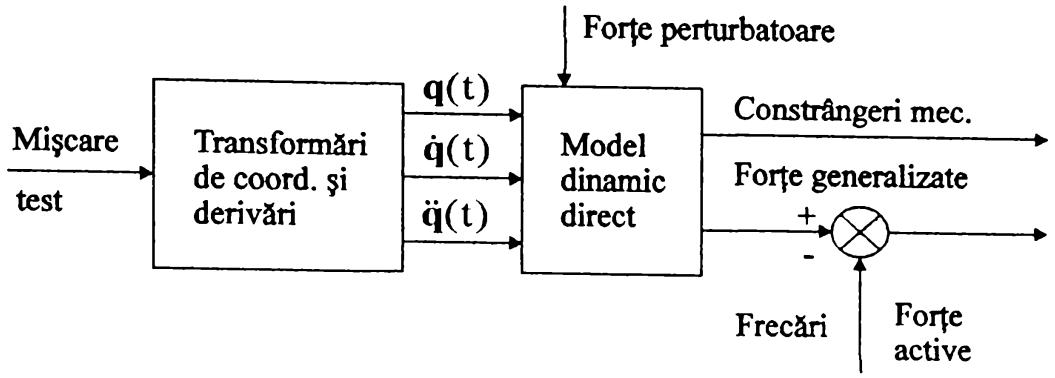


Fig.5.1. Utilizarea modelului dinamic direct

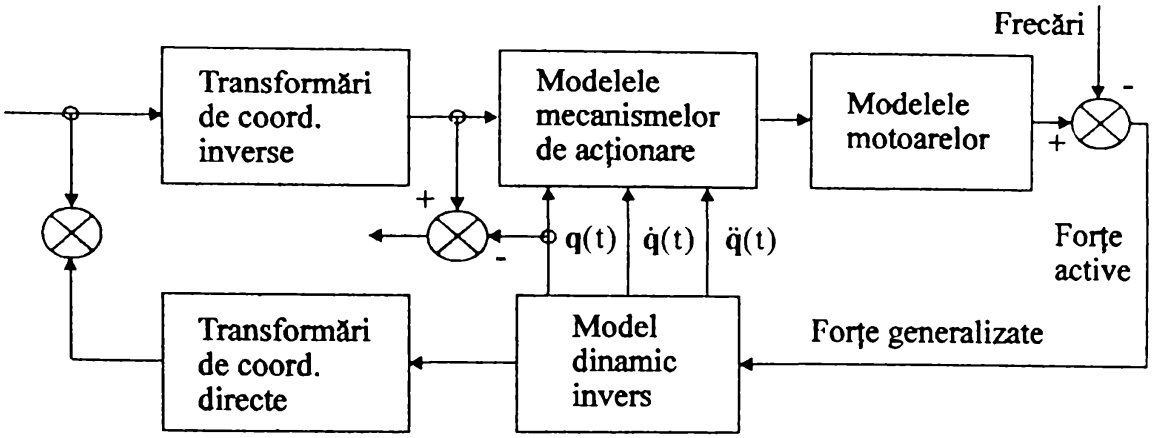


Fig.5.2. Utilizarea modelului dinamic invers

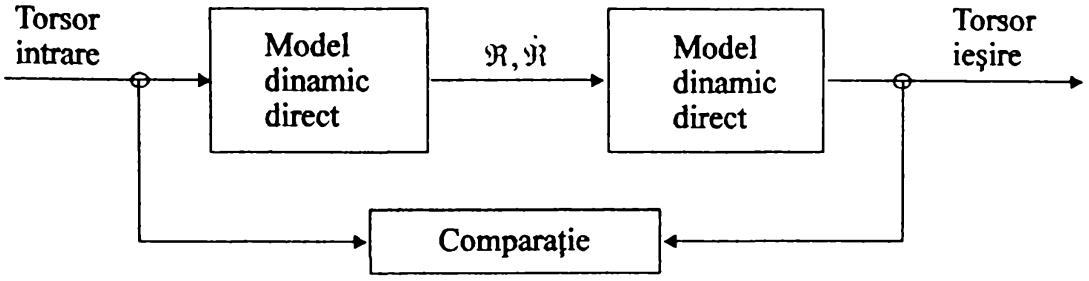


Fig.5.3. Utilizarea în serie a modelului dinamic direct şi invers

Una din cele mai uzuale metode de abordare a dinamicii unui sistem de corpuri, în particular a roboţilor, se bazează pe utilizarea ecuaţiilor lui Lagrange de speţa a II-a. Procesul de modelare se face pe baza unor ipoteze simplificatoare, în vederea obţinerii unor ecuaţii dinamice uşor de prelucrat (rigiditatea corpurilor structurii de manipulare, neglijarea forţelor de frecare uscată, neglijarea jocurilor care afectează precizia, etc.).

Pornind de la ecuaţiile lui Lagrange de speţa a II-a, modelul dinamic se poate obţine în două moduri:

- a) exprimând energia cinetică E_c totală, energia potenţială E_p şi forţele generalizate Q_i şi înlocuind în relaţia [I5]:

BUPT

$$\frac{d}{dt}\left(\frac{\partial E_c}{\partial \dot{q}_i}\right) - \frac{\partial E_c}{\partial q_i} = Q_i \qquad (5.1)$$

- b) folosind transformările omogene [P2].

Modelul dinamic propus de R.Paul se bazează pe formalismul matematic al lui Lagrange. Pentru un modul oarecare *i* al robotului, ecuația diferențială a mișcării este de forma:

$$\sum_{j=1}^{n} D_{ij}\ddot{q}_j + J_{ci}\ddot{q}_i + \sum_{j=1}^{n}\sum_{m=1}^{n} D_{ijm}\dot{q}_j\dot{q}_m + D_i = Q_i \qquad (5.2)$$

unde s-au notat:

$$D_{ij} = \sum_{k=\max i,j}^{n} Tr\left[\frac{\partial \mathbf{G}_k}{\partial q_j}\,\mathbf{J}_k\,(\frac{\partial \mathbf{G}_k}{\partial q_i})^T\right] \qquad (5.3)$$

$$D_{ijm} = \sum_{k=\max i,j,m}^{n} Tr\left[\frac{\partial^2 \mathbf{G}_k}{\partial q_j \partial q_m}\mathbf{J}_k(\frac{\partial \mathbf{G}_k}{\partial q_i})^T\right] \qquad (5.4)$$

$$D_i = \sum_{k=i}^{n}\left[-m_k\,\mathbf{g}^T\,\frac{\partial \mathbf{G}_k}{\partial q_i}\,\mathbf{r}_{ck}^k\right] \qquad (5.5)$$

În relațiile (5.2) - (5.5), termenii reprezintă:

- D_{ii}, coeficient al inerției efective la cupla *i*;
- D_{ij}, coeficient al inerției de cuplare între cuplele cinematice *i* și *j*;
- $\mathbf{J}_k$, pseudomatricea de inerție a modulului *i*;
- J_{ci}, inerția motorului care acționează cupla *i*;
- D_{ijj}, forțele centripete care acționează asupra cuplei *i* datorate vitezelor cuplei *j*;
- D_{ijm}, forțele Coriolis care acționează asupra cuplei *i* datorate vitezelor cuplelor *j* și *m*;
- D_i, sarcina gravitațională la cupla *i*;
- **g**, vectorul accelerație gravitațională;
- Q_i, forțele generalizate.

Pentru comanda manipulatorului sunt importanți în mod deosebit termenii inerțiali și cei gravitaționali, deoarece ei afectează stabilitatea motoarelor și precizia de poziționare.

Pentru explicitarea ecuației (5.2) a modelului dinamic, ținând cont de tipul cuplelor cinematice (translație sau rotație), se procedează la simplificarea termenilor inerțiali.

Termenii inerției efective se calculează, funcție de tipul cuplei cinematice, cu relațiile: (5.6) pentru cuplă de rotație și (5.7) pentru cuplă de translație.

$$D_{ii} = \sum_{k=i}^{n}\left\{\left[n_{k_z}^2 k_{k_{xx}}^2 + o_{k_z}^2 k_{k_{yy}}^2 + a_{k_z}^2 k_{k_{zz}}^2\right] + [\mathbf{p}_k \cdot \mathbf{p}_k] + 2\mathbf{r}_{ck}^k \cdot \left[(\mathbf{p}_k \cdot \mathbf{n}_k)\mathbf{i} + (\mathbf{p}_k \cdot \mathbf{o}_k)\mathbf{j} + (\mathbf{p}_k \cdot \mathbf{a}_k)\mathbf{k}\right]\right\} \qquad (5.6)$$

$$D_{ii} = \sum_{k=i}^{n} m_k \qquad (5.7)$$

unde $\mathbf{n}_k$, $\mathbf{o}_k$, $\mathbf{a}_k$ și $\mathbf{p}_k$ precizează elementele matricei $\mathbf{G}_k^{i-1}$. În general, **n**, **o**, **a** sunt versorii sistemului de referință atașat unui modul al robotului, iar **p** este vectorul de poziție al originii acestui sistem de referință.

BUPT

5.2. DETERMINAREA ECUAȚIILOR DE MIȘCARE PE BAZA MODELULUI DINAMIC PENTRU UN MANIPULATOR CU TREI GRADE DE LIBERTATE

În procesele de sudare se utilizează cu precădere roboți al căror braț urmărește o traiectorie predeterminată ce poate fi descrisă analitic. Important este ca prin impunerea anumitor legi de mișcare a efectorului final să se determine, pe de o parte legile de variație în raport cu timpul ale variabilelor cuplelor și legile de variație ale elementelor mecanice (forțe și momente) active, iar pe de altă parte, posibilitățile concrete de a corecta poziția și orientarea efectorului prin aplicarea corecțiilor în cuplele cinematice, în cazul în care, în timpul procesului sunt sesizate abateri de la traiectoria impusă.

Se consideră, spre exemplu, manipulatorul cu trei grade de libertate de tip TTR, destinat sudării automate, a cărui schemă cinematică structurală este prezentată în figura 5.4 [D18], [D19], [D20], [D21], [D25], [T9].

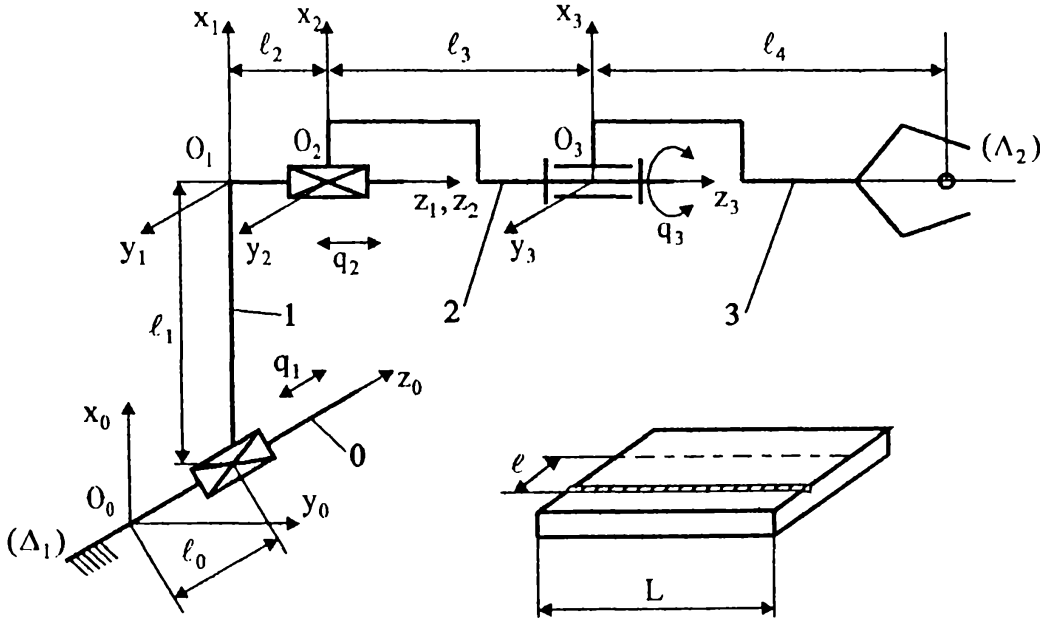


Fig.5.4. Schema cinematică structurală a robotului TTR

Se consideră cunoscute următoarele mărimi:

- parametrii constructivi ℓ_i, (i=1,...,4);
- dimensiunile reperelor ce urmează să fie sudate: lungimea cordonului de sudură, L și distanța dintre două cordoane alăturate, ℓ;
- masele m_i, respectiv forțele de greutate G_i ale modulelor de translație și rotație, inclusiv dispozitivul de sudare ce constituie efectorul final, unde $m_3=m_{\text{modul rotație}}+m_{\text{dispozitiv sudare}}$;
- momentul de inerție mecanic $J_{D2}^{(3)}$, al modulului de rotație și al efectorului final, în raport cu axa (Δ_2);
- distanțele echivalente k_{ixx}^2, k_{iyy}^2 și k_{izz}^2 definite prin: $k_{ixx}^2=J_{ix}/m_i$, $k_{iyy}^2=J_{iy}/m_i$ și $k_{izz}^2=J_{iz}/m_i$;
- pozițiile centrelor de masă ale modulelor, exprimate în sistemele de referință atașate acestora, r_{cki}^k;

BUPT

•torsorul de reducere al sistemului de forţe care acţionează asupra dispozitivului de sudare, calculat în originea sistemului de referinţă ataşat ultimului modul al manipulatorului, τ_{O3}^{G3}.

Se calculează ecuaţiile dinamice prin cele două metode prezentate în paragraful 5.1.

a) Energia cinetică a robotului este:

$$E_c = \frac{1}{2}\left[\left(\sum_{i=1}^{3} m_i\right)\dot{q}_1^2 + \left(\sum_{i=1}^{3} m_i\right)\dot{q}_2^2 + \left(\sum_{i=3}^{3} J_{\Lambda 2}^{(i)}\right)\dot{q}_3^2\right] \tag{5.8}$$

Forţele generalizate Q_i se calculează folosind principiul lucrului mecanic virtual:

$$\begin{aligned} Q_1 &= F_1 - F_{f_1} \\ Q_2 &= F_2 - F_{f_2} \\ Q_3 &= M_3 - M_{f_3} \end{aligned} \tag{5.9}$$

unde F_1, F_2, M_3 reprezintă forţele motoare, respectiv momentul motor din cuplele cinematice, iar F_{f1}, F_{f2}, M_{f3} reprezintă forţele de frecare, respectiv momentul de frecare din cuplele cinematice.

Forţele de frecare se consideră de forma:

$$F_{f_i} = K_i \cdot \dot{q}_i \tag{5.10}$$

pentru i = 1, 2, unde K_i este o constantă ce depinde de aria nominală a suprafeţei de contact, de vâscozitatea dinamică şi de grosimea peliculei de lubrifiant.

Momentul de frecare se consideră de forma:

$$M_{f_3} = K_3 \cdot \dot{q}_3 \tag{5.11}$$

unde K_3 este o constantă ce depinde de diametrul alezajului, de vâscozitatea dinamică şi de jocul diametral.

Făcând calculele impuse de ecuaţia (5.1), rezultă ecuaţiile diferenţiale ale mişcării:

$$\begin{aligned} \left(\sum_{i=1}^{3} m_i\right)\ddot{q}_1 &= F_1 - K_1\dot{q}_1 \\ \left(\sum_{i=2}^{3} m_i\right)\ddot{q}_2 &= F_2 - K_2\dot{q}_2 \\ J_{\Lambda 2}^{(3)}\ddot{q}_3 &= M_3 - K_3\dot{q}_3 \end{aligned} \tag{5.12}$$

b) Poziţia şi orientarea efectorului final al robotului, în sistemul de referinţă fix, se exprimă prin matricea de transformare generală $\mathbf{G}_3^0$ [D31]:

$$\mathbf{G}_3^0 = \begin{bmatrix} \cos q_3 & -\sin q_3 & 0 & \ell_1 \\ 0 & 0 & 1 & \ell_2 + q_2 + \ell_3 \\ -\sin q_3 & -\cos q_3 & 0 & \ell_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.13}$$

Pentru cazul robotului TTR studiat, termenii inerţiei efective sunt:

BUPT

$$D_{11} = \sum_{i=1}^{3} m_i$$

$$D_{22} = \sum_{i=2}^{3} m_i \qquad (5.14)$$

$$D_{33} = m_3 k_{3_{zz}}{}^2$$

Termenii inerţiei de cuplare D_{ij} se determină cu relaţia:

$$D_{ij} = \sum_{k=\max i,j}^{3} \left[m_k (\delta_i^k)^T \mathbf{K}_k (\delta_j^k) + \mathbf{d}_i^k \cdot \mathbf{d}_j^k + \mathbf{r}_{c_k}^k \cdot \left(\mathbf{d}_i^k x \delta_j^k + \mathbf{d}_j^k x \delta_i^k \right) \right] \qquad (5.15)$$

unde $\mathbf{d}_{i,j}{}^k$ şi $\delta_{i,j}{}^k$ reprezintă vectorii translaţiilor diferenţiale, respectiv rotaţiilor diferenţiale, iar $\mathbf{K}_k$ este dat de relaţia:

$$\mathbf{K}_k = \begin{bmatrix} k_{k_{xx}}^2 & -k_{k_{xy}}^2 & -k_{k_{xz}}^2 \\ -k_{k_{xy}}^2 & k_{k_{yy}}^2 & -k_{k_{yz}}^2 \\ -k_{k_{xz}}^2 & -k_{k_{yz}}^2 & k_{k_{zz}}^2 \end{bmatrix} \qquad (5.16)$$

Vectorii de translaţie diferenţială, respectiv de rotaţie diferenţială, pentru cele trei cuple cinematice, rezultă:

$$\begin{array}{llllll} \mathbf{d}_1^1 = -\mathbf{j} & \delta_1^1 = 0 & & & & \\ \mathbf{d}_1^2 = -\mathbf{j} & \delta_1^2 = 0 & \mathbf{d}_2^2 = \mathbf{k} & \delta_2^2 = 0 & \mathbf{d}_3^3 = 0 & \delta_3^3 = \mathbf{k} \\ \mathbf{d}_1^3 = -S_3\mathbf{i} - C_3\mathbf{j} & \delta_1^3 = 0 & \mathbf{d}_2^3 = \mathbf{k} & \delta_2^3 = 0 & & \end{array} \qquad (5.17)$$

Vectorii de poziţie pentru centrele de masă ale modulelor robotului, exprimaţi în sistemele de referinţă ataşate acestora, $\mathbf{r}_{ck}{}^k$ sunt:

$$\begin{cases} \mathbf{r}_{c_1}^1 = -\ell_1 \mathbf{i}_1 + y_1 \mathbf{j}_1 \\ \mathbf{r}_{c_2}^2 = x_2 \mathbf{i}_2 + y_2 \mathbf{j}_2 + z_2 \mathbf{k}_2 \\ \mathbf{r}_{c_3}^3 = x_3 \mathbf{i}_3 + y_3 \mathbf{j}_3 + z_3 \mathbf{k}_3 \end{cases} \qquad (6.18)$$

Înlocuind relaţiile (5.16) - (5.18) în relaţia (5.15), rezultă:

$$D_{12} = 0; \quad D_{23} = 0; \quad D_{13} = m_3(-C_3 x_3 + S_3 y_3) \qquad (6.19)$$

Dacă se consideră că centrul de masă al modulului de rotaţie împreună cu dispozitivul de sudare aparţine axei (Δ_2), rezultă că şi $D_{13}=0$. Termenii inerţiei de cuplare fiind: $D_{12}=D_{23}=D_{13}=0$, rezultă că nu există cuplare între cuplele cinematice, deci ecuaţiile diferenţiale ale mişcării vor fi decuplate.

Termenii gravitaţionali se determină cu relaţia:

$$D_i = \mathbf{g}^{i-1} \sum_{k=i}^{3} m_k \mathbf{r}_{ck}^{i-1} \qquad (5.20)$$

unde:

$$\mathbf{g}^{i-1} = \begin{bmatrix} -\mathbf{g} \cdot \mathbf{o}_{i-1} & -\mathbf{g} \cdot \mathbf{n}_{i-1} & 0 & 0 \end{bmatrix} \qquad (5.21)$$

BUPT

pentru cupla *i* de rotaţie, şi:

$$\mathbf{g}^{i-1} = \begin{bmatrix} 0 & 0 & 0 & -\mathbf{g}\cdot\mathbf{a}_{i-1} \end{bmatrix} \tag{5.22}$$

pentru cupla *i* de translaţie.

Vectorul acceleraţiei gravitaţionale fiind:

$$\mathbf{g} = \begin{bmatrix} -g & 0 & 0 & 0 \end{bmatrix}^T \tag{5.23}$$

rezultă:

$$\begin{aligned} \mathbf{g}^0 &= \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \\ \mathbf{g}^1 &= \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \\ \mathbf{g}^2 &= \begin{bmatrix} 0 & -g & 0 & 0 \end{bmatrix} \end{aligned} \tag{5.24}$$

Vectorii de tipul $\mathbf{r}_{ck}^{i-1}$ din relaţia (5.20) se determină cu o relaţie de forma:

$$\mathbf{r}_{c_k}^{i-1} = \mathbf{G}_k^{i-1}\mathbf{r}_{c_k}^k \tag{5.25}$$

Din relaţia (5.18), prin particularizare, se obţin vectorii:

$$\begin{aligned} \mathbf{r}_{c_1}^1 &= \begin{bmatrix} -\ell_1 & y_1 & 0 & 1 \end{bmatrix}^T \\ \mathbf{r}_{c_2}^2 &= \begin{bmatrix} 0 & 0 & z_2 & 1 \end{bmatrix}^T \\ \mathbf{r}_{c_3}^3 &= \begin{bmatrix} 0 & 0 & z_3 & 1 \end{bmatrix}^T \end{aligned} \tag{5.26}$$

rezultând:

$$\begin{aligned} \mathbf{r}_{c_3}^2 &= \begin{bmatrix} 0 & 0 & z_3 + \ell_3 & 1 \end{bmatrix}^T \\ \mathbf{r}_{c_3}^1 &= \begin{bmatrix} 0 & 0 & z_3 + \ell_2 + \ell_3 + q_2 & 1 \end{bmatrix}^T \end{aligned} \tag{5.27}$$

Cum $\mathbf{g}^0 = \mathbf{0}$ şi $\mathbf{g}^1 = \mathbf{0}$, vectorii de tipul $\mathbf{r}_{ck}^{i-1}$ nu se mai calculează pentru modulele *1* şi *2*, rezultând direct $D_1=D_2=0$. Dacă se efectuează calculele pentru D_3, conform relaţiei (5.20), rezultă $D_3=0$. Deci termenii gravitaţionali sunt: $D_1=D_2=D_3=0$.

Referitor la termenii J_{ci}, inerţia servomotorului de acţionare a cuplei *3* a fost deja introdusă în $J_{\Lambda 2}^{(3)}$ (momentul de inerţie pentru modulul de rotaţie împreună cu efectorul final), iar inerţia motoarelor cuplelor de translaţie s-a neglijat.

Termenii de forma D_{ijm} se consideră, de asemenea, nuli.

Forţele generalizate Q_i sunt deja calculate (5.9).

Dacă se înlocuiesc termenii obţinuţi în ecuaţia (5.2) şi se consideră că $J_{\Lambda 2}^{(3)} = m_3 k_{3zz}^2$ se obţin ecuaţiile diferenţiale ale mişcării (ecuaţiile 5.12):

$$\left(\sum_{i=1}^{3} m_i\right)\ddot{q}_1 = F_1 - k_1\dot{q}_1$$

$$\left(\sum_{i=2}^{3} m_i\right)\ddot{q}_2 = F_2 - k_2\dot{q}_2$$

$$J_{\Lambda 2}^{(3)}\ddot{q}_3 = M_3 - k_3\dot{q}_3$$

Dacă se cunosc, din procesul de sudare, variaţiile în timp ale forţelor şi momentului care acţionează în cuplele cinematice conducătoare, impunând condiţiile limită, prin

BUPT

rezolvarea sistemului ecuaţiilor diferenţiale decuplate (5.12) se determină legile de variaţie în raport cu timpul ale coordonatelor cuplelor (coordonatele generalizate $q_1(t)$, $q_2(t)$, $q_3(t)$), deci **MDI**.

Considerând ecuaţiile diferenţiale de forma:

$$\ddot{q}_i + \frac{K_i}{M_i}\dot{q}_i = \frac{1}{M_i}F_i(t) \tag{5.28}$$

rezultă:

$$q_i(t) = \frac{1}{K_i}\left[\int_0^T F_i(t)\,dt - e^{-\frac{K_i}{M_i}t}\int_0^T F(t)\,e^{\frac{K_i}{M_i}t}\,dt\right] + C_2 e^{-\frac{K_i}{M_i}t} + C_1 \tag{5.29}$$

În anumite situaţii interesează rezolvarea directă a problemei, adică impunerea legilor de mişcare ale coordonatelor cuplelor şi determinarea elementelor mecanice active din cuplele cinematice, deci **MDD**.

Astfel, impunând următoarele legi de variaţie în raport cu timpul pentru coordonatele cuplelor:

$$q_1(t) = \begin{cases} \frac{1}{2}a_1 t^2 & 0 \le t \le \frac{1}{4}T_1 \\ -\frac{\ell}{6} + a_1\sqrt{\frac{1}{3}a_1 t} & \frac{1}{4}T_1 \le t \le \frac{3}{4}T_1 \\ \frac{13}{6}\ell + 4a_1\sqrt{\frac{1}{3}a_1 t} - \frac{1}{2}a_1 t^2 & \frac{3}{4}T_1 \le t \le T_1 \end{cases} \tag{5.30}$$

$q_2(t) = v_S\, t$

$q_3(t) = \omega\, t$

rezultă legile de variaţie ale forţelor şi momentului care acţionează în cuplele cinematice:

$$F_1(t) = \begin{cases} a_1\sum_{i=1}^{4} m_i + K_1 a_1 t & 0 \le t \le \frac{1}{4}T_1 \\ K_1 a_1\sqrt{\frac{\ell}{3a_1}} & \frac{1}{4}T_1 \le t \le \frac{3}{4}T_1 \\ K_1 a_1\left(4\sqrt{\frac{\ell}{3a_1}} - t\right) - a_1\sum_{i=1}^{4} m_i & \frac{3}{4}T_1 \le t \le T_1 \end{cases} \tag{5.31}$$

$F_2(t) = K_2 v_s$

$M_3(t) = k_3\, \omega$

Metoda prezentată pentru determinarea ecuaţiilor de mişcare ale robotului de sudare TTR poate fi aplicată la orice tip de robot. Metoda are un profund caracter de generalitate deoarece fiecare modul al robotului intervine în studiu în mod distinct şi de sine stătător. Dacă procesul de sudare impune legi specifice de variaţie pentru mişcare $q_i(t)$, atunci acestea se vor reflecta în expresiile elementelor motoare $F_i(t)$ şi $M_i(t)$ şi vor permite alegerea corespunzătoare a motoarelor de acţionare din cuple.

BUPT

5.3. STUDIUL DINAMICII UNUI ROBOT KUKA TIP 364/10.0

Robotul KUKA studiat este de tip RRRRRR şi este prezentat principial în figura 5.5 care precizează axele robotului şi posibilităţile sale de mişcare. Modelarea unui robot presupune, aşa cum s-a precizat deja în capitolele anterioare, modelul geometric, diferenţial şi dinamic.

5.3.1. MODELUL GEOMETRIC

Pentru determinarea modelului geometric [D18] s-au ales sistemele de referinţă ataşate elementelor robotului, considerând convenţia Denavit-Hartenberg (fig.5.6). Cu sistemele de referinţă precizate rezultă parametrii de poziţie ai robotului, prezentaţi în Tabelul 5.1.

Tabelul5.1. Parametrii de poziţie ai robotului KUKA 364/10.0

Nr.element	Variabila cuplei	α_i	a_i
1	θ_1	0^o	a_1
2	θ_2	90^o	a_2
3	θ_3	0^o	a_3
4	θ_4	90^o	0
5	θ_5	90^o	0
6	θ_6	-90^o	0

Calculând matricele de poziţie ale fiecărui element faţă de precedentul [D16], [P5], se obţine, prin înmulţirea lor, matricea $\mathbf{G}_6^0$ care exprimă poziţia şi orientarea efectorului final în raport cu sistemul de referinţă fix. Modelul geometric al robotului este exprimat de ecuaţiile (5.32) în care s-au notat: $C_i = \cos\theta_i$ şi $S_i = \sin\theta_i$.

În cadrul problemei directe (**MGD**), impunând legile de variaţie ale coordonatelor cuplelor $\theta_i=\theta_i(t)$, se pot preciza parametrii de poziţie şi orientare (n_x, ..., p_z) ai efectorului final. Rezolvarea modelului geometric invers **MGI** conduce la posibilitatea de exprimare a coordonatelor cuplelor cinematice ca funcţii de poziţia şi orientarea efectorului final (5.33).

Şi într-un caz şi în celălalt, restricţiile impuse de caracterul limitat al spaţiului de lucru sau al procesului tehnologic, determină limitarea domeniului de existenţă pentru coordonatele cuplelor cinematice. Pentru robotul studiat IR 364/10, aceste domenii sunt prezentate în Tabelul 5.2.

BUPT

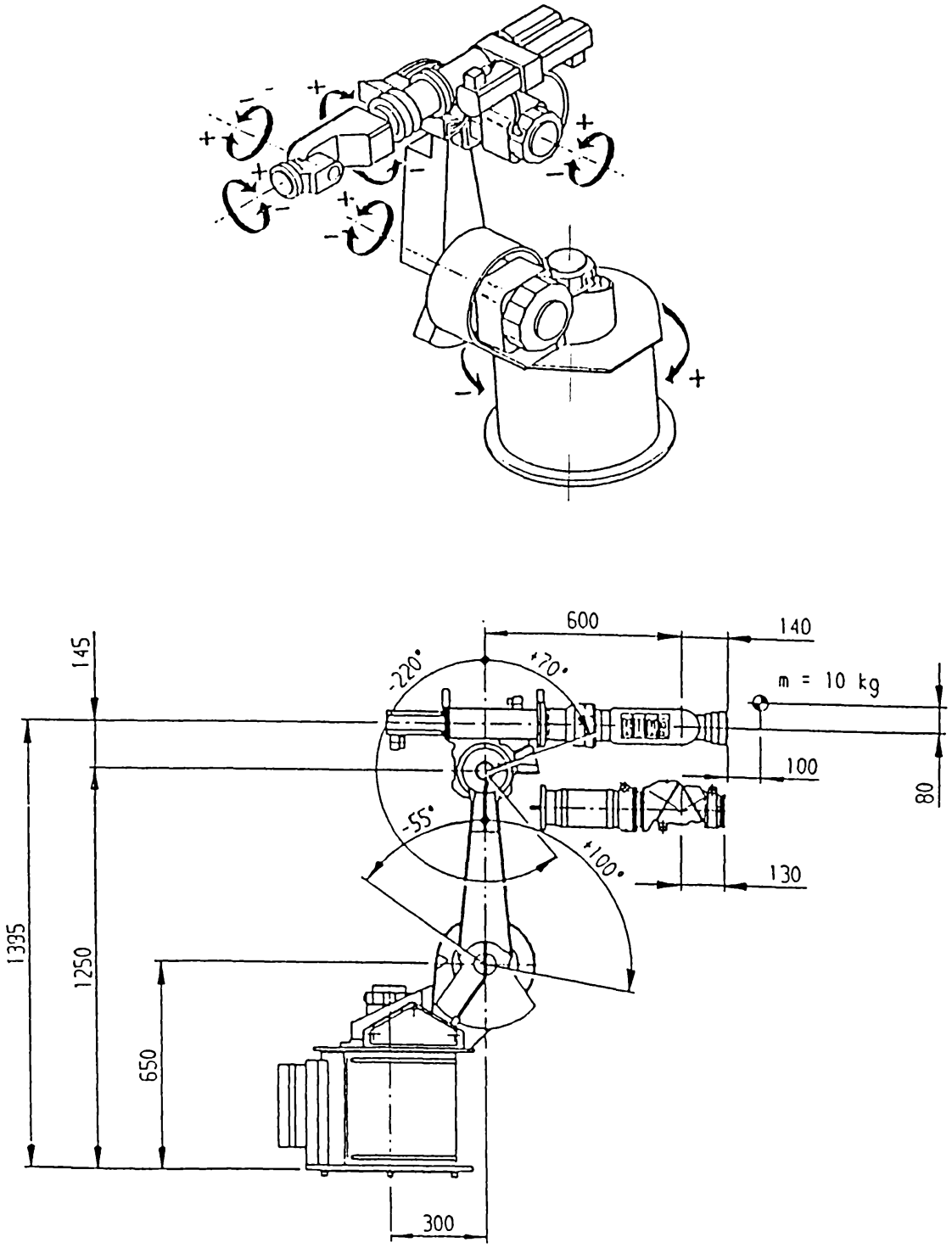


Fig.5.5. Axele şi posibilităţile de mişcare ale robotului KUKA

BUPT

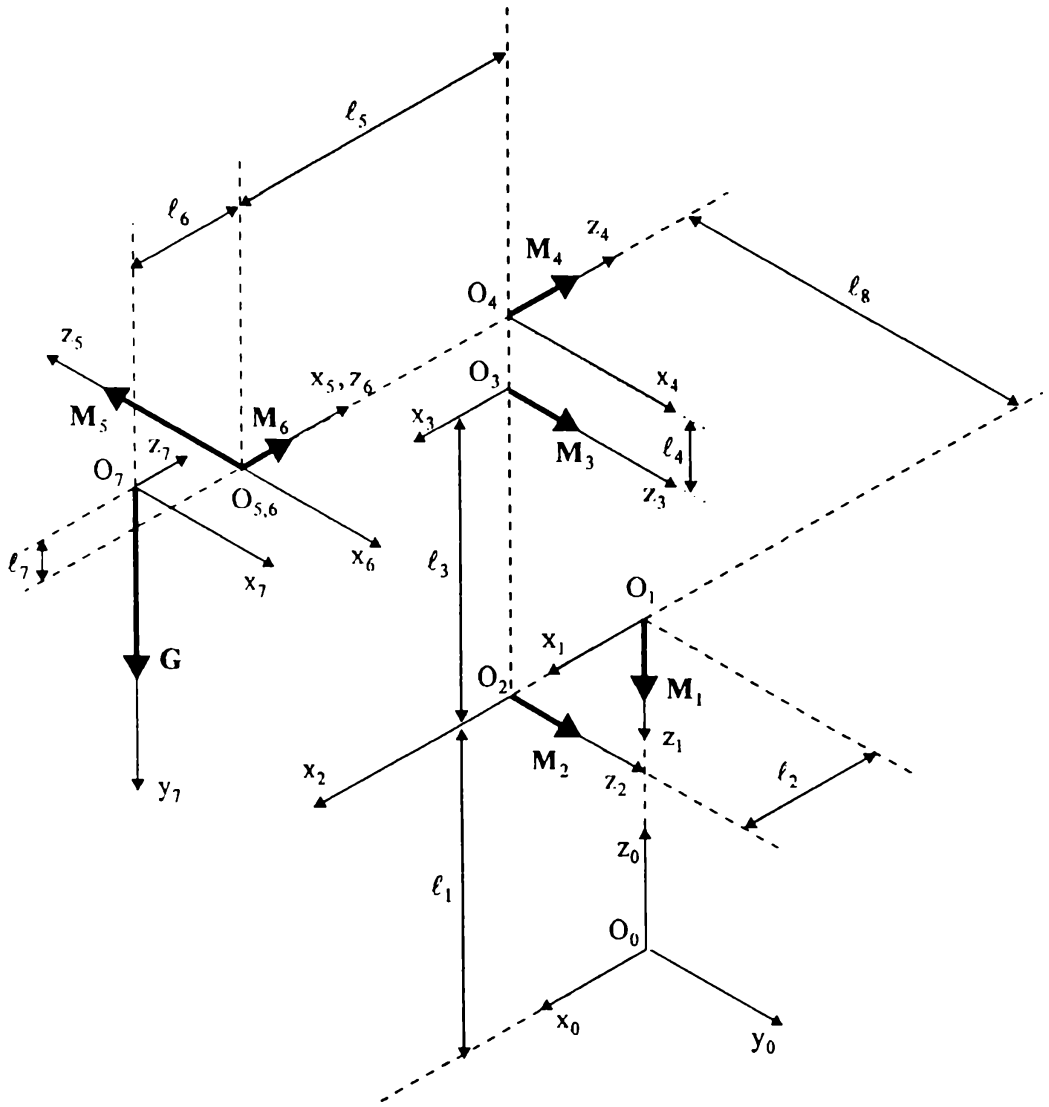


Fig.5.6.Sistemele de referinţă ataşate elementelor robotului

Tabelul 5.2. Domeniile de existenţă ale coordonatelor cuplelor

Axa	Domeniul variabilei
1	±155°
2	[+100°, -55°]
3	[+70°, -220°]
4	±350°
5	±135°
6	±350°

BUPT

MGD:

$$
\begin{aligned}
n_x &= C_1\{C_2[C_3(C_4C_5C_6+S_4S_6)-S_3(S_4C_5C_6-C_4S_6)]+S_2S_5C_6\}-\\
&-S_1\{S_2[C_3(C_4C_5C_6+S_4S_6)-S_3(S_4C_5C_6-C_4S_6)]-C_2S_5C_6\}\\
n_y &= S_1\{C_2[C_3(C_4C_5C_6+S_4S_6)-S_3(S_4C_5C_6-C_4S_6)]+S_2S_5C_6\}+\\
&+C_1\{S_2[C_3(C_4C_5C_6+S_4S_6)-S_3(S_4C_5C_6-C_4S_6)]-C_2S_5C_6\}\\
n_z &= S_3(C_4C_5C_6+S_4S_6)+C_3(S_4C_5C_6-C_4S_6)\\
o_x &= C_1[C_2(-C_3C_4S_5+S_3S_4S_5)+S_2C_5]-S_1[S_2(-C_3C_4S_5+S_3S_4S_5)-C_2C_5]\\
o_y &= S_1[C_2(-C_3C_4S_5+S_3S_4S_5)+S_2C_5]+C_1[S_2(-C_3C_4S_5+S_3S_4S_5)-C_2C_5]\\
o_z &= -S_3C_4S_5-C_3S_4S_5\\
a_x &= C_1\{C_2[C_3(-C_4C_5S_6+S_4C_6)+S_3(S_4C_5S_6+C_4C_6)]-S_2S_5S_6\}-\\
&-S_1\{S_2[C_3(-C_4C_5S_6+S_4C_6)+S_3(S_4C_5S_6+C_4C_6)]+C_2S_5S_6\}\\
a_y &= S_1\{C_2[C_3(-C_4C_5S_6+S_4C_6)+S_3(S_4C_5S_6+C_4C_6)]-S_2S_5S_6\}+\\
&+C_1\{S_2[C_3(-C_4C_5S_6+S_4C_6)+S_3(S_4C_5S_6+C_4C_6)]+C_2S_5S_6\}\\
a_z &= S_3(-C_4C_5S_6+S_4C_6)-C_3(S_4C_5S_6+C_4C_6)\\
p_x &= C_1C_2(\ell_4C_3+\ell_3)-S_1S_2(\ell_4C_3+\ell_3)+\ell_2C_1\\
p_y &= S_1C_2(\ell_4C_3+\ell_3)+C_1S_2(\ell_4C_3+\ell_3)+\ell_2S_1\\
p_z &= \ell_4S_3
\end{aligned}
\qquad (5.32)
$$

MGI:

$$
\begin{aligned}
tg\theta_1 &= \frac{p_y}{p_x}\\
tg\theta_2 &= \frac{-p_x tg\theta_1+p_y}{p_x+p_y tg\theta_1-a_1\sqrt{1+tg^2\theta_1}}\\
tg\theta_3 &= \frac{p_z}{\sqrt{1-p_z^2-a_1^2}-a_2}\\
tg\theta_4 &= \frac{o_z\cos\theta_3-(o_x\cos\theta_1+o_y\sin\theta_1)\cos\theta_2\sin\theta_3}{o_z\sin\theta_3+(o_x\cos\theta_1+o_y\cos\theta_1)\cos\theta_2\cos\theta_3}\\
tg\theta_5 &= \frac{\sqrt{(o_x\cos\theta_1+o_y\sin\theta_1)^2+o_z^2}}{o_y\cos\theta_1-o_x\sin\theta_1}\\
tg\theta_6 &= \frac{a_x\sin\theta_1-a_y\cos\theta_1}{-n_x\sin\theta_1+n_y\cos\theta_1}
\end{aligned}
\qquad (5.33)
$$

Utilizând aceste limite în cadrul problemei directe şi ţinând seama de elementele constructive ale robotului, rezultă dimensiunile şi configuraţia spaţiului activ. Pentru robotul studiat, luând ca punct de referinţă intersecţia axelor 4 şi 5, spaţiul de lucru rezultă de forma din figura 5.7.

BUPT

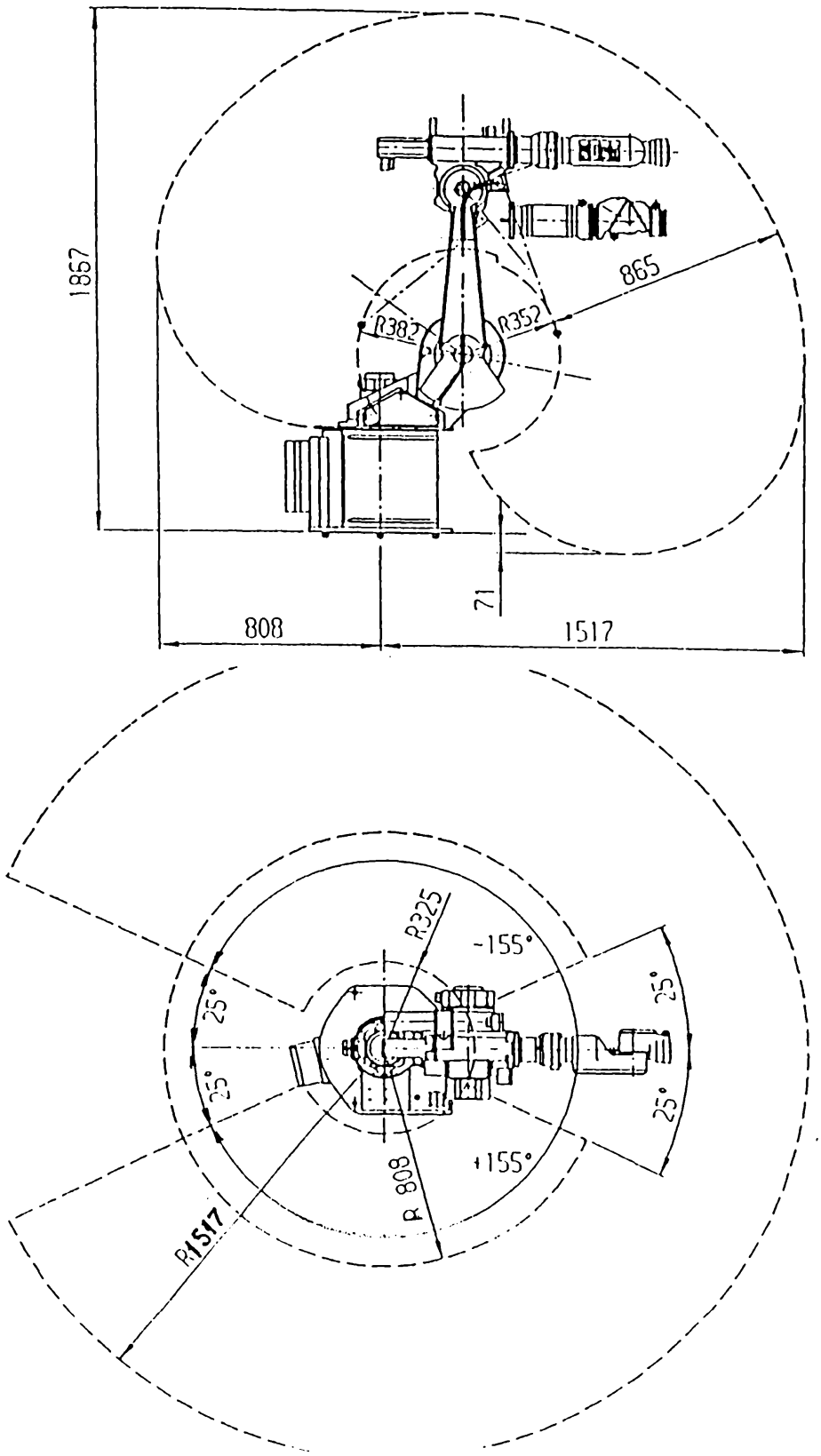


Fig.5.7. Spațiul de lucru al robotului KUKA

BUPT

5.3.2. MODELUL DINAMIC

Procesul de modelare s-a făcut pe baza ipotezelor simplificatoare amintite în paragraful 5.1, folosind ecuaţiile lui Lagrange (ecuaţia 5.1). S-au considerat cunoscute următoarele mărimi:

- parametrii constructivi ℓ_i, (i=1,...,8);
- masele m_i, resppectiv forţele de greutate G_i ale modulelor de rotaţie, inclusiv pentru efectorul final, i=1, ..., 8
- momentele de inerţie mecanice $J_{xi}^{(i)}$, $J_{yi}^{(i)}$, $J_{zi}^{(i)}$ ale modulelor şi ale efectorului final, în raport cu axele x_i, y_i şi z_i;
- centrele de masă ale modulelor sunt considerate în originile sistemelor de referinţă ataşate acestora;
- torsorul de reducere al sistemului de forţe care acţionează asupra ultimului modul al robotului, calculat în originea sistemului de referinţă ataşat acestuia τ_{O6}^{G6}.

Energia cinetică a robotului [D40] este exprimată în relaţia (5.34).

$$
\begin{aligned}
E_c = \frac{1}{2}\Bigg\{ & \dot\theta_1^2\Bigg[J_{z1}^{(1)} + \sum_{i=2}^{7} J_{y_i}^{(i)} + m_2\ell_2^2 + m_3\ell_3^2 + m_4\ell_2^2 + (m_5+m_6)(\ell_2+\ell_5)^2 + m_7(\ell_2+\ell_5+\ell_6)^2 \Bigg] + \\
& + \dot\theta_2^2\Bigg[\sum_{i=2,3,5} J_{z_i}^{(i)} + \sum_{i=4,6,7} J_{x_i}^{(i)} + m_3\ell_3^2 + m_4(\ell_3+\ell_4)^2 + (m_5+m_6)\left((\ell_3+\ell_4)^2+\ell_5^2\right) + m_7\left((\ell_5+\ell_6)^2+(\ell_3+\ell_4+\ell_7)^2\right) \Bigg] + \\
& + \dot\theta_3^2\Bigg[\sum_{i=3,5} J_{z_i}^{(i)} + \sum_{i=4,6,7} J_{x_i}^{(i)} + m_4\ell_4^2 + (m_5+m_6)\left(\ell_4^2+\ell_5^2\right) + m_7\left((\ell_4+\ell_7)^2+(\ell_5+\ell_6)^2\right) \Bigg] + \\
& + \dot\theta_4^2\Bigg[J_{x5}^{(i)} + \sum_{i=4,6,7} J_{z_i}^{(i)} + m_7\ell_7^2 \Bigg] + \dot\theta_5^2\Bigg[J_{z5}^{(i)} + \sum_{i=6}^{7} J_{x_i}^{(i)} + m_7\ell_6^2 \Bigg] + \dot\theta_6^2\Bigg[\sum_{i=6}^{7} J_{z_i}^{(i)} + m_7\ell_7^2 \Bigg] \Bigg\}
\end{aligned}
\tag{5.34}
$$

Forţele generalizate Q_i se determină cu relaţia:

$$Q_i = \frac{\delta L_i}{\delta\theta_i} \tag{5.35}$$

în care dL_i este lucrul mecanic virtual elementar al torsorului forţelor exterioare, corespunzător unei deplasări virtuale elementare $\delta\theta_i$. Pentru întregul robot, lucrul mecanic virtual se calculează cu relaţia:

$$\delta L = \sum_{i=1}^{7} \mathbf{F}_i \delta \mathbf{r}_i + \sum_{i=1}^{7} \mathbf{M}_{O_i} \delta\theta_i \tag{5.36}$$

Calculând lucrul mecanic virtual cu relaţia (5.36) şi înlocuind în relaţia (5.35) rezultă forţele generalizate Q_i:

$$Q_1 = M_1$$

$$Q_2 = M_2 + \ell_3 S_2 \sum_{i=3}^{7} G_i + \ell_4 S_{23} \sum_{i=4}^{7} G_i + \ell_5 C_{23} \sum_{i=5}^{7} G_i + \ell_6 C_{235} G_7$$

$$Q_3 = M_3 + \ell_4 S_{23} \sum_{i=4}^{7} G_i + \ell_5 C_{23} \sum_{i=5}^{7} G_i + \ell_6 C_{235} G_7$$

BUPT

$$
\begin{aligned}
Q_4 &= M_4 \\
Q_5 &= M_5 + \ell_6 C_{235} G_7 \\
Q_6 &= M_6
\end{aligned}
\tag{5.37}
$$

unde s-au notat: $S_i=\sin\theta_i$, $S_{ij}=\sin(\theta_i+\theta_j)$, $C_{ij}=\cos(\theta_i+\theta_j)$, $C_{ijk}=\cos(\theta_i+\theta_j+\theta_k)$.

Înlocuind energia cinetică (5.34) şi forţele generalizate (5.37) în ecuaţia (5.1) rezultă ecuaţiile diferenţiale ale mişcării robotului KUKA studiat:

$$
\ddot{\theta}_1\left[J_{z1}^{(1)}+\sum_{i=2}^{7}J_{y_i}^{(i)}+(m_2+m_4)\ell_2^2+m_3\ell_3^2+(m_5+m_6)(\ell_2^2+\ell_5^2)+m_7(\ell_2+\ell_5+\ell_6)^2\right]=M_2
$$

$$
\ddot{\theta}_2\left[\sum_{i=2,3,5}J_{z_i}^{(i)}+\sum_{i=4,6,7}J_{x_i}^{(i)}+m_3\ell_3^2+(m_4+m_5+m_6)(\ell_3+\ell_4)^2+(m_5+m_6)\ell_5^2+m_7\left((\ell_5+\ell_6)^2+(\ell_3+\ell_4+\ell_7)^2\right)\right]
$$

$$
=M_2+\ell_3 S_2\sum_{i=3}^{7}G_i+\ell_4 S_{23}\sum_{i=4}^{7}G_i+\ell_5 C_{23}\sum_{i=5}^{7}G_i+\ell_6 C_{235}G_7
$$

$$
\ddot{\theta}_3\left[\sum_{i=3,5}J_{z_i}^{(i)}+\sum_{i=4,6,7}J_{x_i}^{(i)}+m_4\ell_4^2+(m_5+m_6)(\ell_4^2+\ell_5^2)+m_7\left((\ell_4+\ell_7)^2+(\ell_5+\ell_6)^2\right)\right]=
$$

$$
=M_3+\ell_4 S_{23}\sum_{i=4}^{7}G_i+\ell_5 C_{23}\sum_{i=5}^{7}G_i+\ell_6 C_{235}G_7
\tag{5.38}
$$

$$
\ddot{\theta}_4\left[J_{x5}^{(5)}+\sum_{i=4,6,7}J_{z_i}^{(i)}+m_7\ell_7^2\right]=M_4
$$

$$
\ddot{\theta}_5\left[J_{z5}^{(5)}+\sum_{i=6}^{7}J_{x_i}^{(i)}+m_7\ell_6^2\right]=M_5+\ell_6 C_{235}G_7
$$

$$
\ddot{\theta}_6\left[\sum_{i=6}^{7}J_{z_i}^{(i)}+m_7\ell_7^2\right]=M_6
$$

Torsorul din cuple

Una din problemele importante din dinamica roboţilor industriali constă în determinarea torsorului din cuple. Acesta se poate calcula atunci când se cunoaşte torsorul în sistemul de referinţă ataşat efectorului final şi Jacobianul robotului:

$$
\mathbf{Q} = [\mathbf{J}_r]^T \tau_{O_6}^{G_6} \tag{5.39}
$$

unde $\mathbf{Q}$ reprezintă torsorul din cuple, $\mathbf{J}_r$ este Jacobianul robotului, iar τ_{O6}^{G6} este torsorul în sistemul G_6.

$$
\mathbf{Q} = [M_1 \quad M_2 \quad M_3 \quad M_4 \quad M_5 \quad M_6]^T \tag{5.40}
$$

$$
\tau_{O_6}^{G_6} = [F_{6x} \quad F_{6y} \quad F_{6z} \quad M_{O_{6x}} \quad M_{O_{6y}} \quad M_{O_{6z}}]^T \tag{5.41}
$$

BUPT

$$\mathbf{J}_r = \begin{bmatrix} -S_5C_6(\ell_4C_3+\ell_3+\ell_2C_2) & -S_5C_6(\ell_4C_3+\ell_3) & \ell_4(S_4C_5C_6-C_4S_6) & 0 & 0 & 0 \\ -C_5(\ell_4S_3+\ell_3+\ell_2C_2) & -C_5(\ell_4C_3+\ell_3) & -\ell_4S_4S_5 & 0 & 0 & 0 \\ S_5S_6(\ell_4S_3+\ell_3+\ell_2C_2) & S_5S_6(\ell_4C_3+\ell_3) & -\ell_5(S_4C_5S_6-C_4C_6) & 0 & 0 & 0 \\ C_5C_6S_{34}-S_6C_{34} & C_5C_6S_{34}-S_6C_{34} & S_5C_6 & S_5C_6 & S_6 & 0 \\ -S_5S_{34} & -S_5S_{34} & C_5 & C_5 & 0 & -1 \\ -C_5S_6S_{34}-C_6C_{34} & -C_5S_6S_{34}-C_6C_{34} & -S_5S_6 & -S_5S_6 & C_6 & 0 \end{bmatrix} \quad (5.42)$$

Torsorul din cuple rezultă:

$$\begin{aligned}
M_1 &= -F_{6x}S_5C_6(\ell_4C_3+\ell_3+\ell_2C_3) + (-F_{6y}C_5+F_{6z}S_5S_6)(\ell_4S_3+\ell_3+\ell_2C_2) + \\
&+(M_{O6x}-M_{O6z})(C_5S_6S_{34}-C_6C_{34}) - M_{O6}+S_5S_{34} \\
M_2 &= -(F_{6x}S_5C_6+F_{6y}C_5-F_{6z}S_5S_6)(\ell_4C_3+\ell_3) + (M_{O6x}-M_{O6z})(C_5S_6S_{34}-C_5C_{34}) - M_{O6y}S_5S_{34} \\
M_3 &= F_{6x}\ell_4(S_4C_5C_6-C_4S_6) - F_{6y}S_4S_5\ell_4 - F_{6z}\ell_4(S_4C_5S_6+C_4C_6) + M_{O6x}S_5C_6 + M_{O6y}C_5 - M_{O6z}S_5S_6 \\
M_4 &= M_{O6x}S_5C_6 + M_{O6y}C_5 - M_{O6z}S_5S_6 \\
M_5 &= M_{O6x}S_6 + M_{O6z}C_6 \\
M_6 &= -M_{O6y}
\end{aligned} \quad (5.43)$$

5.3.3. REZOLVAREA MODELULUI DINAMIC DIRECT ÎN CONDIȚII DATE DE MIȘCARE

Ecuațiile dinamice ale robotului KUKA [D21] se pot scrie:

$$\begin{aligned}
\ddot{\theta}_1K_{11} &= M_1 \\
\ddot{\theta}_2K_{21} &= M_2 + K_{22}S_2 + K_{23}S_{23} + K_{24}C_{23} + K_{25}C_{235} \\
\ddot{\theta}_3K_{31} &= M_3 + K_{23}S_{23} + K_{24}C_{23} + K_{25}C_{235} \\
\ddot{\theta}_4K_{41} &= M_4 \\
\ddot{\theta}_5K_{51} &= M_5 + K_{25}C_{235} \\
\ddot{\theta}_6K_{61} &= M_6
\end{aligned} \quad (5.44)$$

unde:

$$\begin{aligned}
K_{11} &= J_{z1}^{(1)} + \sum_{i=2}^{7} J_{y_i}^{(i)} + (m_2+m_4)\ell_2^2 + m_3\ell_3^2 + (m_5+m_6)(\ell_2^2+\ell_5^2) + m_7(\ell_2+\ell_5+\ell_6)^2 \\
K_{21} &= \sum_{i=2,3,5} J_{z_i}^{(i)} + \sum_{i=4,6,7} J_{x_i}^{(i)} + m_3\ell_3^2 + (m_4+m_5+m_6)(\ell_3+\ell_4)^2 + (m_5+m_6)\ell_5^2 + m_7\left[(\ell_5+\ell_6)^2 + (\ell_3+\ell_4+\ell_7)^2\right] \\
K_{31} &= \sum_{i=3,5} J_{z_i}^{(i)} + \sum_{i=4,6,7} J_{x_i}^{(i)} + m_4\ell_4^2 + (m_5+m_6)(\ell_4^2+\ell_5^2) + m_7\left[(\ell_4+\ell_7)^2 + (\ell_5+\ell_6)^2\right] \\
K_{41} &= J_{x5}^{(5)} + \sum_{i=4,6,7} J_{z_i}^{(i)} + m_7\ell_7^2 \\
K_{51} &= J_{z5}^{(5)} + \sum_{i=6}^{7} J_{x_i}^{(i)} + m_7\ell_6^2 \\
K_{61} &= \sum_{i=6}^{7} J_{z_i}^{(i)} + m_7\ell_7^2
\end{aligned} \quad (5.45)$$

BUPT

$$K_{22} = \ell_3 \sum_{i=3}^{7} G_i; \quad K_{23} = \ell_4 \sum_{i=4}^{7} G_i; \quad K_{24} = \ell_5 \sum_{i=3}^{7} G_i; \quad K_{25} = \ell_6 G_7$$

Prin procesul tehnologic se impune coordonarea cuplelor cinematice şi timpul necesar îndeplinirii fiecărei faze în care este structurată sarcina robotului. Deoarece generarea mişcării se realizează în spaţiul coordonatelor cuplelor se pot impune legile de variaţie ale acestora şi se determină momentele motoare care acţionează în aceste cuple. S-au ales următoarele tipuri de interpolări: cuplele cinematice (2) şi (5) - interpolare liniară, ecuaţia (5.46); cupla (1) - polinom de gradul 5, ecuaţia (5.48); cuplele (3), (4) şi (6) - polinom de gradul 3, ecuaţia (5.47). În expresiile funcţiilor de interpolare (5.46)-(5.48): $D=\theta_f-\theta_i$, t_f -timpul final.

$$\theta(t) = \theta_i + \frac{t}{t_f} D \tag{5.46}$$

$$\theta(t) = \theta_i + \left[3\left(\frac{t}{t_f}\right)^2 - 2\left(\frac{t}{t_f}\right)^3 \right] D \tag{5.47}$$

$$\theta(t) = \theta_i + \left[10\left(\frac{t}{t_f}\right)^3 - 15\left(\frac{t}{t_f}\right)^4 + 6\left(\frac{t}{t_f}\right)^5 \right] D \tag{5.48}$$

Înlocuind legile de variaţie ale coordonatelor cuplelor în ecuaţiile diferenţiale ale mişcării rezultă legile de variaţie ale momentelor active din cuple.

$$M_1(t) = -310K_{11}\left(3t - 0.45t^2 + 0.015t^3\right) \tag{5.49}$$

$$M_2(t) = -K_{22}S_2 - K_{23}S_{23} - K_{24}\left(C_{23} + C_{235}\right) \tag{5.50}$$

$$M_3(t) = -290K_{31}(0{,}3 - 0{,}6t) - K_{23}S_{23} - K_{24}C_{23} - K_{25}C_{235} \tag{5.51}$$

$$M_4(t) = -700K_{41}\left(0.3 - 0.015t^2\right) \tag{5.52}$$

$$M_5(t) = -K_{25}C_{235} \tag{5.53}$$

$$M_6(t) = -700K_{61}\left(0.3 - 0.6t\right) \tag{5.54}$$

5.4. VARIAŢIA ELEMENTELOR TORSORULUI DE REDUCERE ŞI TORSORUL DIN CUPLELE UNUI ROBOT RTRTTR

În acest paragraf se prezintă o metodă de determinare a unui torsor de reducere, exprimat într-un sistem de referinţă ataşat unui element al unui robot, echivalent torsorului care acţionează asupra efectorului final sau echivalent unui alt torsor de reducere exprimat într-un alt sistem de referinţă. De asemenea, se propune o metodă de determinare a forţelor şi momentelor care acţionează în cuplele cinematice conducătoare atunci când se cunoaşte torsorul care acţionează asupra efectorului final al unui robot.

Pentru fiecare din metodele de mai sus s-a realizat un program de calcul al torsorilor de reducere. Programele au fost scrise în limbajul Turbo Pascal şi sunt utilizabile pe calculatoare IBM-PC, care dispun de un monitor VGA sau superior. Programele elaborate sunt rulate spre exemplificare pentru un robot cu şase grade de libertate (RTRTTR), dar sunt utilizabile pentru orice tip de robot, cu orice structură şi oricâte grade de libertate.

BUPT

5.4.1. VARIAȚIA ELEMENTELOR TORSORULUI DE REDUCERE CU SCHIMBAREA SISTEMULUI DE REFERINȚĂ

Poziția și orientarea efectorului final al unui robot, într-un sistem de referință fix, se exprimă printr-o matrice de transformare generală $\mathbf{G}_n$.

Se consideră robotul cu șase grade de libertate de tip RTRTTR, de construcție modulară, a cărui schemă cinematică structurală este prezentată în fig.4.3. Folosind convenția Hartenberg-Denavit se atașează elementelor robotului câte un sistem de referință G_i, se exprimă matricele de transformare $\mathbf{T}_i^{i-1}$ care indică poziția și orientarea sistemului de referință atașat elementului i față de sistemul de referință atașat elementului $i-1$ și se determină, prin produsul acestora, expresia generală a matricei de transformare $\mathbf{G}_6^0$, unde s-au notat: cos q_i=C_i, sin q_i=S_i și ℓ_5'=ℓ_5+ℓ_6, ℓ_6 fiind distanța de la O_6 la punctul caracteristic al robotului.

Torsorul de reducere într-un punct O, al unui sistem de forțe, se exprimă printr-o matrice de forma [D3], [T3]:

$$\tau_O = \begin{bmatrix} F_x & F_y & F_z & M_{O_x} & M_{O_y} & M_{O_z} \end{bmatrix}^T \tag{5.55}$$

Elementele matricei (5.55) sunt proiecțiile elementelor torsorului de reducere pe axele unui sistem de referință fix cu originea în punctul O. Torsorul de reducere $\tau_{O_i}^{G_i}$, exprimat într-un sistem de referință G_i, care se obține din sistemul dat printr-o transformare $\mathbf{G}_i$, echivalent torsorului τ_O, se determină egalând lucrurile mecanice elementare produse de cei doi torsori:

$$\tau_{O_i}^{G_i} = [(\mathbf{J}_i)^{-1}]^T \cdot \tau_O \tag{5.56}$$

unde $\mathbf{J}_i$ este Jacobianul sistemului G_i.

Astfel, relațiile de transformare pentru proiecțiile elementelor torsorului de reducere sunt:

$$\begin{aligned} F_x^{G_i} &= \mathbf{n}_i \cdot \mathbf{F} \\ F_y^{G_i} &= \mathbf{o}_i \cdot \mathbf{F} \\ F_z^{G_i} &= \mathbf{a}_i \cdot \mathbf{F} \end{aligned} \tag{5.57}$$

$$\begin{aligned} M_{O_{i_x}}^{G_i} &= \mathbf{n}_i \cdot (\mathbf{F} \mathrm{x} \mathbf{p}_i + \mathbf{M}_O) \\ M_{O_{i_y}}^{G_i} &= \mathbf{o}_i \cdot (\mathbf{F} \mathrm{x} \mathbf{p}_i + \mathbf{M}_O) \\ M_{O_{i_z}}^{G_i} &= \mathbf{a}_i \cdot (\mathbf{F} \mathrm{x} \mathbf{p}_i + \mathbf{M}_O) \end{aligned} \tag{5.58}$$

La momentul considerat, poziția și orientarea efectorului final în sistemul de referință fix se exprimă (fig.5.9) prin matricea $\mathbf{G}_6^0$:

$$\mathbf{G}_6^0 = \begin{bmatrix} 0 & 0 & -1 & -1.3 \\ 0 & 1 & 0 & -0.6 \\ 1 & 0 & 0 & 1.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.59}$$

Torsorul de reducere care acționează asupra efectorului final, cunoscut din procesul tehnologic, exprimat în raport cu sistemul de referință fix este (fig.5.10):

$$\tau_O = \begin{bmatrix} 300 & 0 & -100 & 50 & 0 & 0 \end{bmatrix}^T \tag{5.60}$$

BUPT

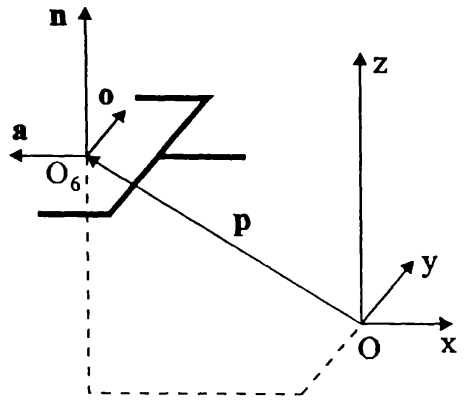


Fig.5.9. Poziția și orientarea sistemului de referință atașat efectorului final

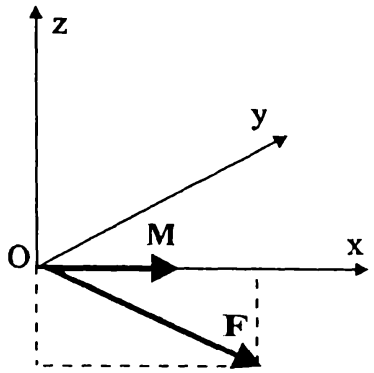


Fig.5.10. Torsorul de reducere τ_O

Folosind relația (5.56) se determină torsorul de reducere echivalent care acționează asupra efectorului final, în raport cu propriul sistem de referință (G_6):

$$\tau_{O_6}^{G_6} = \begin{bmatrix} -100 & 100 & 300 & -900 & 250 & -700 \end{bmatrix}^T \tag{5.61}$$

Pentru determinarea torsorilor echivalenți care acționează asupra elementelor robotului în raport cu propriile sisteme de referință este necesară rezolvarea modelului geometric invers. Rezultă astfel valorile coordonatelor cuplelor cuplelor cinematice la momentul considerat: q_1=0°, q_2=0.5m, q_3=0°, q_4=0.3m, q_5=0.3m și q_6=180°. Valorile parametrilor constructivi pentru robotul studiat sunt: ℓ_1=0.5m, ℓ_2=0.5m, ℓ_3=0.3m, ℓ_4=0.2m și ℓ_5'=0.6m.

Torsorii care acționează asupra elementelor robotului sunt:

$$\begin{aligned}
\tau_{O_1}^{G_1} &= \begin{bmatrix} 300 & 0 & -100 & 50 & 0 & 0 \end{bmatrix}^T \\
\tau_{O_2}^{G_2} &= \begin{bmatrix} 300 & 100 & -100 & 500 & 0 & -450 \end{bmatrix}^T \\
\tau_{O_3}^{G_3} &= \begin{bmatrix} 300 & 100 & -100 & 500 & 0 & -450 \end{bmatrix}^T \\
\tau_{O_4}^{G_4} &= \begin{bmatrix} -100 & 100 & 300 & -900 & 450 & -700 \end{bmatrix}^T \\
\tau_{O_5}^{G_5} &= \begin{bmatrix} -100 & 100 & 300 & -900 & 190 & -7000 \end{bmatrix}^T
\end{aligned} \tag{5.62}$$

În cazul în care este folosit pentru un alt robot decât cel din aplicația prezentată, programul solicită exprimarea matricelor de transformare $\mathbf{T}_i^{i-1}$, deci apelează la modelul geometric direct și invers. Algoritmul programului este următorul:

Pasul 1. Introducerea datelor: parametrii de poziție ai robotului sau matricele de transformări omogene $\mathbf{T}_i^{i-1}$; soluția **MGI**; poziția și orientarea efectorului final la momentul considerat; torsorul care acționează asupra efectorului final, exprimat în raport cu sistemul de referință fix..

Pasul 2. Calculează torsorul în sistemul G_6.

Pasul 3. Calculează matricea $\mathbf{G}_i = \mathbf{T}_1^0 \mathbf{T}_2^1 \ldots \mathbf{T}_i^{i-1}$.

Pasul 4: Identifică versorii $\mathbf{n}_6$, $\mathbf{o}_6$, $\mathbf{a}_6$ și vectorul de poziție $\mathbf{p}_6$ ai sistemului G_i.

BUPT

Pasul 5: Calculează proiecțiile forței rezultante pe axele sistemului de referință G_6 cu relațiile (5.57).

Pasul 6: Calculează proiecțiile momentului rezultant în raport cu originea O_6 a sistemului G_6 ataşat efectorului final, pe axele sistemului G_6 se calculează cu relațiile (5.58).

Pasul 7: Afişează şi reprezintă τ^{Gi}_{Oi}.

Pasul 8: Repetă de la *Pasul 3*.

Pentru exemplul prezentat, torsorii τ^{Gi}_{Oi} sunt prezentați în fig. 5.11 - 5.16

5.4.2. TORSORI ECHIVALENȚI ÎN CUPLELE ROBOTULUI

Una din cele mai importante probleme în studiul roboților este determinarea torsorului echivalent în cuplele robotului când se cunoaşte torsorul care acționează asupra efectorului final τ_{O6}^{G6}.

Considerând cuplele cinematice conducătoare ale robotului simple (rotație sau translație), torsorul cuplelor se notează cu **Q** şi se exprimă:

$$\mathbf{Q} = \begin{bmatrix} Q_1 & \dots & Q_n \end{bmatrix}^T$$

(5.63)sau, pentru robotul studiat:

$$Q = \begin{bmatrix} M_1 & F_2 & M_3 & F_4 & F_5 & M_6 \end{bmatrix}^T \tag{5.64}$$

Determinarea elementelor torsorului din cuple, datorate elementelor torsorului care acționează asupra efectorului final, se face pe baza egalării lucrurilor mecanice elementare produse de cei doi torsori, considerând că aceştia sunt echivalenți din punct de vedere mecanic [D16], [T9]:

$$\mathbf{Q} = (\mathbf{J}_r)^T \cdot \tau_{O6}^{G6} \tag{5.65}$$

unde $\mathbf{J}_r$ este Jacobianul robotului, care se determină cu relația:

$$\mathbf{J}_r = \begin{bmatrix} d_1^{G6} & \dots & d_i^{G6} & \dots & d_6^{G6} \\ \delta_1^{G6} & \dots & \delta_i^{G6} & \dots & \delta_6^{G6} \end{bmatrix} \tag{5.66}$$

unde $\mathbf{d}_i^{G6}$ este vectorul translației diferențiale a elementului *i* în sistemul G_6, iar δ_i^{G6} este vectorul rotației diferențiale a elementului *i* în sistemul G_6.

Vectorii translațiilor şi rotațiilor diferențiale se obțin din matricele de transformare $\mathbf{G}_6^{i-1}$. Pentru cazul studiat, Jacobianul robotului este:

$$\mathbf{J}_r = \begin{bmatrix} 0 & 1 & -1.3 & 0 & 0 & 0 \\ -1.3 & 0 & 0 & -1 & 0 & 0 \\ -0.6 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.67}$$

iar torsorul din cuplele robotului este:

$$\mathbf{Q} = \begin{bmatrix} 0 & -100 & 450 & 0 & -300 & 10 \end{bmatrix}^T \tag{5.68}$$

BUPT

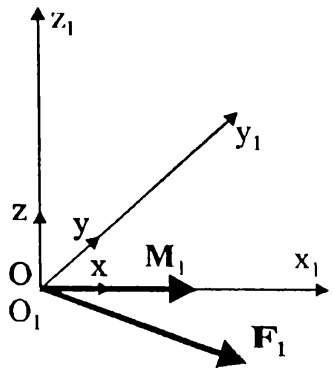


Fig.5.11 Torsorul $\tau_{O_1}^{G_1}$

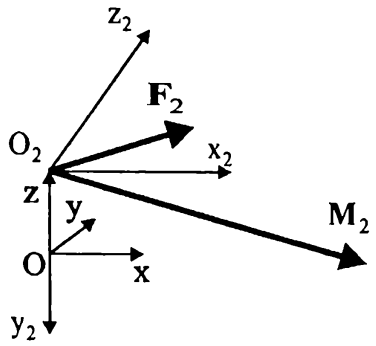


Fig.5.12 Torsorul $\tau_{O_2}^{G_2}$

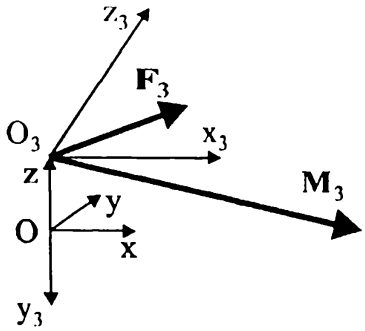


Fig.5.13 Torsorul $\tau_{O_3}^{G_3}$

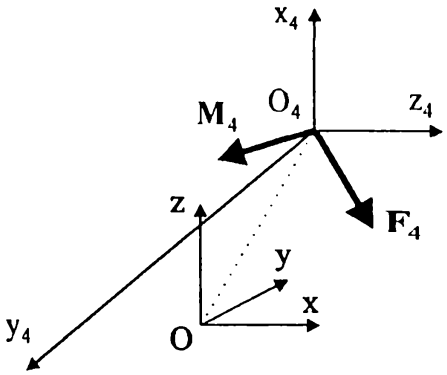


Fig.5.14 Torsorul $\tau_{O_4}^{G_4}$

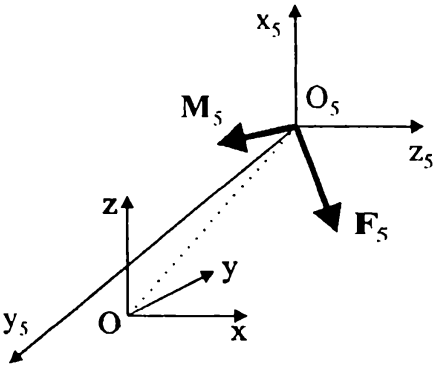


Fig.5.15 Torsorul $\tau_{O_5}^{G_5}$

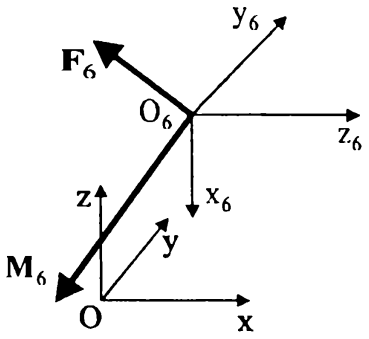


Fig.5.16 Torsorul $\tau_{O_6}^{G_6}$

BUPT

5.5. PRECIZIA PARCURGERII TRAIECTORIEI UNUI ROBOT

Parcurgerea unei anumite traiectorii de către un robot este influenţată de precizia de poziţionare şi de orientare a efectorului final. Valorile instantanee ale erorilor de poziţionare e_p, respectiv de orientare e_o, depind de mărimi ce caracterizează geometria şi construcţia robotului (abaterile geometrice, elasticitatea elementelor, disipările prin frecare), de condiţiile de lucru (temperatură, sarcină manipulată), de valorile vitezelor instantanee corespunzătoare, dar şi de "istoria" evoluţiei situărilor impuse de sarcina robotului.

5.5.1. PRECIZIA CINEMATICĂ A PARCURGERII TRAIECTORIEI

Precizia cinematică a unui robot caracterizează posibilitatea acestuia de a asigura poziţionarea şi orientarea sistemului de referinţă ataşat efectorului final conform poziţiei şi orientării programate [L15]. Pentru determinarea experimentală a preciziei cinematice este necesară efectuarea unui număr N de măsurări (N>10), definind situarea sistemului de referinţă ataşat efectorului final. Precizia de situare se calculează cu relaţiile:

$$|\mathbf{e}_p| = \frac{1}{N}\sum_{i=1}^{N}|\mathbf{e}_{pi}|; \qquad |\mathbf{e}_o| = \frac{1}{N}\sum_{i=1}^{N}|\mathbf{e}_{oi}| \tag{5.69}$$

Indicele de performanţă global se poate defini cu relaţiile:

$$|\mathbf{e}_p| = \max_i|\mathbf{e}_{pi}|; \qquad |\mathbf{e}_o| = \max_i|\mathbf{e}_{oi}| \tag{5.70}$$

Introducerea funcţiilor scalare (5.70) permite compensarea optimă a erorilor geometrice ale transformării inverse de coordonate, prin variaţia coeficienţilor sistemului de comandă, până la atingerea unei situaţii optime:

$$\min\left(|\mathbf{e}_p|,\ |\mathbf{e}_o|\right) \tag{5.71}$$

Precizia de poziţionare şi orientare a efectorului final, depinzând de exactitatea modelului geometric al robotului, poate fi exprimată pe baza unei analize a influenţei abaterilor coordonatelor reale ale punctului caracteristic faţă de valorile teoretice impuse. Astfel, din modelul geometric, se determină poziţia punctului caracteristic (coordonatele x, y, z ale acestuia). Abaterea de la poziţia teoretică se poate determina cu relaţiile:

$$dx = \sum_{i=1}^{n}\frac{\partial x}{\partial q_i}dq_i; \qquad dy = \sum_{i=1}^{n}\frac{\partial y}{\partial q_i}dq_i; \qquad dz = \sum_{i=1}^{n}\frac{\partial z}{\partial q_i}dq_i \tag{5.72}$$

unde dq_i exprimă abaterile de la valorile impuse coordonatelor cuplelor cinematice. Eroarea de poziţionare rezultă:

$$|\mathbf{e}_p| = \sqrt{dx^2 + dy^2 + dz^2} \tag{5.73}$$

Poziţia reală a efectorului final este dată de:

$$x_r = x + dx; \qquad y_r = y + dy; \qquad z_r = z + dz \tag{5.74}$$

Din analiza abaterii faţă de poziţia teoretică, respectiv din studiul valorilor maxime şi minime ale erorilor, se pot determina poziţiile în care aceste erori au valori extreme, precum şi

BUPT

valorile coordonatelor cuplelor cinematice corespunzătoare şi se pot aplica astfel corecţiile necesare. În aceeaşi manieră se poate studia şi precizia de orientare a efectorului final.

Indiferent însă de modul de determinare a erorilor de poziţie şi orientare este necesară elaborarea modelului geometric al robotului studiat.

5.5.1.1. CALCULUL ABATERILOR PE BAZA MODELULUI DIFERENŢIAL

Se propune în continuare o metodă de determinare a erorilor de poziţie şi orientare folosind modelul diferenţial al robotului, respectiv vectorul diferenţial al mişcării exprimat in raport cu sistemul de referinţă fix. Se poate obţine astfel o precizie ridicată a poziţiei şi orientării efectorului final, fiecare cuplă cinematică fiind considerată la o rezoluţie înaltă.

Realizarea, pe baza modelului geometric, a modelului diferenţial al robotului [D16], caracterizat prin relaţia:

$$\mathbf{D}^{G_n} = \mathbf{J}_r \mathbf{D}_c \tag{5.75}$$

permite calculul variaţiilor diferenţiale înregistrate în poziţia şi orientarea efectorului final faţă de propriul sistem de referinţă, de forma:

$$\mathbf{D}^{G_n} = \left[d_x^{G_n} \quad d_y^{G_n} \quad d_z^{G_n} \quad \delta_x^{G_n} \quad \delta_y^{G_n} \quad \delta_z^{G_n} \right]^T \tag{5.76}$$

În relaţia (5.76), $\mathbf{J}_r$ este Jacobianul robotului, iar $\mathbf{D}_c$ este vectorul diferenţial al cuplelor. Presupunând o variaţie a coordonatelor cuplelor de forma:

$$dq_i = a_i\, dt + b_i \tag{5.77}$$

ceea ce corespunde cu situaţia reală când termenii de ordin superior se pot neglija, intervalul de timp considerat fiind elementar, se obţine, din relaţia (5.75), expresia vectorului diferenţial al mişcării în sistemul de referinţă ataşat efectorului final (sistemul G_n):

$$\mathbf{D}^{G_n} = \begin{bmatrix} \sum_{i=1}^{n} a_i d_{i_x}^{G_n} dt + \sum_{i=1}^{n} b_i d_{i_x}^{G_n} \\ \sum_{i=1}^{n} a_i \delta_{i_x}^{G_n} dt + \sum_{i=1}^{n} b_i \delta_{i_x}^{G_n} \end{bmatrix} \tag{5.78}$$

Eroarea de poziţionare, calculată în sistemul de referinţă ataşat efectorului final rezultă:

$$|e_p| = \sqrt{\left[\sum_{i=1}^{n} d_{i_x}^{G_n}\left(a_i dt + b_i\right)\right]^2 + \left[\sum_{i=1}^{n} d_{i_y}^{G_n}\left(a_i dt + b_i\right)\right]^2 + \left[\sum_{i=1}^{n} d_{i_z}^{G_n}\left(a_i dt + b_i\right)\right]^2} \tag{5.79}$$

Indiferent de sistemul de referinţă în care se determină eroarea de poziţionare (orientare), mărimea ei este aceeaşi:

$$\left|e_p^{G_n}\right| = \sqrt{\left(d_x^{G_n}\right)^2 + \left(d_y^{G_n}\right)^2 + \left(d_z^{G_n}\right)^2} = \left|e_p^{G_o}\right| = \sqrt{dx^2 + dy^2 + dz^2} \tag{5.80}$$

Cum poziţia şi orientarea se exprimă de regulă faţă de sistemul de referinţă fix, se determină erorile de poziţie şi de orientare în acest sistem, vectorul diferenţial al mişcării fiind:

BUPT

$$\mathbf{D}^{G0} = \left(\mathbf{J}_{G_n}\right)^{-1} \mathbf{D}^{G_n} \tag{5.81}$$

unde $\mathbf{J}_{Gn}$ este Jacobianul sistemului de referinţă ataşat efectorului final.

Proiecţiile vectorului translaţiei diferenţiale, pe axele sistemului de referinţă fix, sunt:

$$\begin{aligned}
d_x &= n_{n_x} \sum_{i=1}^{n} d_{i_x}^{G_n} (a_i dt + b_i) + n_{n_y} \sum_{i=1}^{n} d_{i_y}^{G_n} (a_i dt + b_i) + n_{n_z} \sum_{i=1}^{n} d_{i_z}^{G_n} (a_i dt + b_i) \\
d_y &= o_{n_x} \sum_{i=1}^{n} d_{i_x}^{G_n} (a_i dt + b_i) + o_{n_y} \sum_{i=1}^{n} d_{i_y}^{G_n} (a_i dt + b_i) + o_{n_z} \sum_{i=1}^{n} d_{i_z}^{G_n} (a_i dt + b_i) \\
d_z &= a_{n_x} \sum_{i=1}^{n} d_{i_x}^{G_n} (a_i dt + b_i) + a_{n_y} \sum_{i=1}^{n} d_{i_y}^{G_n} (a_i dt + b_i) + a_{n_z} \sum_{i=1}^{n} d_{i_z}^{G_n} (a_i dt + b_i)
\end{aligned} \tag{5.82}$$

Eroarea de orientare se poate determina cu o relaţie similară.

5.5.1.2. CAZUL PARTICULAR ROBOTULUI CU TREI GRADE DE LIBERTATE

Se consideră robotul de sudare de tip TTR (fig.5.4) ale cărui ecuaţii dinamice au fost determinate în paragraful 5.2, pentru care poziţia şi orientarea sunt exprimate de matricea $\mathbf{G}_3^0$ [D25] (ecuaţia 5.13).

Jacobianul robotului este [D29]:

$$\mathbf{J}_r = \begin{bmatrix} -\sin q_3 & 0 & 0 \\ -\cos q_3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.83}$$

Considerând vectorul diferenţial al cuplelor de forma:

$$\mathbf{D}_c = \begin{bmatrix} a_1 dt + b_1 \\ a_2 dt + b_2 \\ a_3 dt + b_3 \end{bmatrix} \tag{5.84}$$

se obţine vectorul diferenţial al mişcării capului de sudare în sistemul de referinţă ataşat acestuia:

$$\mathbf{D}^{G_3} = \begin{bmatrix} -\sin q_3 (a_1 dt + b_1) \\ -\cos q_3 (a_1 dt + b_1) \\ (a_2 dt + b_2) \\ 0 \\ 0 \\ (a_3 dt + b_3) \end{bmatrix} \tag{5.85}$$

Eroarea de poziţie, în sistemul de referinţă ataşat efectorului final, este:

$$\left| e_p^{G_3} \right| = \sqrt{(a_1 dt + b_1)^2 + (a_2 dt + b_2)^2} \tag{5.86}$$

Eroarea de orientare, în sistemul de referinţă ataşat efectorului final, este:

BUPT

$$\left|e_o^{G3}\right| = \left|a_3 dt + b_3\right| \tag{5.87}$$

Dacă se calculează vectorul diferențial al mișcării în sistemul de referință fix, cu relația (5.81), se obține:

$$\mathbf{D}^{G0} = \begin{bmatrix} -S_3 C_3 dq_1 - S_3 dq_2 \\ S_3 C_3 dq_1 - C_3 dq_2 \\ -C_3 dq_1 \\ S_3^2 dq_1(\ell_2 + q_2 + \ell_3) - C_3 dq_1[S_3 \ell_1 + C_3(\ell_0 + q_1)] - C_3 dq_2(\ell_2 + q_2 + \ell_3) - S_3 dq_3 \\ S_3 C_3 dq_1(\ell_2 + q_2 + \ell l_3) - C_3 dq_1[C_3 \ell_1 - S_3(\ell_0 + q_1)] + S_3 dq_2(\ell_2 + q_2 + \ell_3) - C_3 dq_3 \\ -S_3 dq_1(\ell_0 + q_1) + \ell_1 dq_2 \end{bmatrix} \tag{5.88}$$

unde S_i=sin q_i şi C_i=cos q_i.

5.5.2. Precizia dinamică a parcurgerii traiectoriei unui robot

Pentru evaluarea performanțelor dinamice ale unui robot trebuie să se țină seama de parametrii săi cinematici în contextul unei distribuții proprii de mase şi al unui sistem specific de forțe exterioare.

Precizia dinamică a unui robot caracterizează posibilitatea acestuia de a parcurge o traiectorie conform unei programări inițiale dorite. Această precizie poate fi funcție de poziția considerată şi de viteză [L15], deci funcție de evoluția în timp a sistemului de referință ataşat efectorului final.

Indicele de performanță global se poate defini cu relațiile:

$$\left|\mathbf{e}_p\right| = \max_{t\in[0,T]} \left|\mathbf{e}_p(t)\right|; \qquad \left|\mathbf{e}_o\right| = \max_{t\in[0,T]} \left|\mathbf{e}_o(t)\right| \tag{5.89}$$

În aceeaşi manieră ca la precizia cinematică, se pot determina valorile extreme şi dispersia erorilor, respectiv corecțiile necesare pentru atingerea situației (optime) dorite.

5.5.2.1. Evaluarea preciziei de poziționare dinamică

Pentru obținerea erorilor dinamice, prin metoda propusă în acest paragraf, este necesar să se determine în prealabil modelul geometric, respectiv modelul dinamic al robotului studiat. Impunând anumite legi de variație în raport cu timpul pentru forțele/momentele care acționează în cuplele cinematice, se determină legile de variație ale coordonatelor cuplelor. Variațiile diferențiale ale acestor coordonate conduc la erorile de poziție căutate.

Poziția şi orientarea efectorului final al robotului la un moment dat se exprimă prin matricea de transformare generală $\mathbf{G}_n^0$, [D16]:

$$\mathbf{G}_n^0 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & o & 1 \end{bmatrix} \tag{5.90}$$

BUPT

unde **n**, **o**, **a** sunt versorii sistemului de referință atașat efectorului final, iar **p** este vectorul de poziție al originii acestui sistem față de sitemul de referință fix.

Deci, evoluția în timp a sistemului de referință atașat efectorului final este determinată de evoluția în timp a matricei $\mathbf{G}_n^0(t)$. Cei patru vectori care definesc situarea efectorului final la un moment dat sunt funcții implicite de timp deoarece depind de coordonatele cuplelor cinematice.

Determinarea variațiilor în raport cu timpul pentru coordonatele cuplelor cinematice rezultă din rezolvarea modelului dinamic invers. Cunoscând sistemul de acționare al robotului și impunând elementelor mecanice anumite legi de variație $Q_i(t)$ se pot determina legile de variație pentru coordonatele cuplelor cinematice.

Legile de variație $q_i(t)$ se pot înlocui în modelul geometric:

$$\begin{aligned}\mathbf{n}(t) &= f_1(\mathbf{q}(t))\\ \mathbf{o}(t) &= f_2(\mathbf{q}(t))\\ \mathbf{a}(t) &= f_3(\mathbf{q}(t))\\ \mathbf{p}(t) &= f_4(\mathbf{q}(t))\end{aligned} \qquad (5.91)$$

și rezultă evoluția în timp a sistemului de referință atașat efectorului final.

5.5.2.2. Cazul particular al robotului cu trei grade de libertate

Se consideră robotul de sudare cu trei grade de libertate de tip TTR din figura 5.4 pentru care se consideră cunoscute aceleași mărimi ca în paragraful 5.2.

Ecuațiile diferențiale ale mișcării efectorului final se obțin utilizând ecuațiile lui Lagrange de speța a 2-a (ecuația 5.1).

Făcând calculele impuse de ecuația (5.1), rezultă ecuațiile diferențiale ale mișcării (ecuațiile 5.12), care se exprimă sub forma:

$$\begin{aligned}\ddot{q}_1\sum_{i=1}^{4} m_i + K_1\dot{q}_1 &= F_1(t)\\ \ddot{q}_2\sum_{i=2}^{4} m_i + K_2\dot{q}_2 &= F_2(t)\\ \ddot{q}_3\sum_{i=3}^{4} J_{\Lambda_2}^{(i)} + K_3\dot{q}_3 &= M_3(t)\end{aligned} \qquad (5.92)$$

Soluția sistemului ecuațiilor diferențiale este de forma (5.29).

Dacă se consideră următoarele expresii ale elementelor active din cuplele cinematice:

$$\begin{aligned}F_1(t) &= KF_1 t\\ F_2(t) &= K' v_s\\ M_3(t) &= K''\omega\end{aligned} \qquad (5.93)$$

unde constantele v_s și ω reprezintă viteza de sudare, respectiv viteza de pendulare, rezultă:

BUPT

$$\begin{cases} q_1(t) = -\dfrac{KF_1M_1^2}{K_1^3}e^{-\frac{K_1}{M_1}t} + \dfrac{KF_1}{2K_1}t - \dfrac{KF_1M_1}{K_1^2}\left(t - \dfrac{M_1}{K_1}\right) \\ q_2(t) = \dfrac{v_sM_2}{K_2} + \dfrac{v_sM_2}{K_2}\left(\dfrac{K'}{K_2} - 1\right)e^{-\frac{K_2}{M_2}t} + \dfrac{v_sK'}{K_2}t - \dfrac{v_sK'M_2}{K_2^2} \\ q_3(t) = \dfrac{\omega M_3}{K_3} + \dfrac{\omega M_3}{K_{32}}\left(\dfrac{K''}{K_3} - 1\right)e^{-\frac{K_3}{M_3}t} + \dfrac{\omega K''}{K_3}t - \dfrac{\omega K''M_3}{K_3^2} \end{cases} \tag{5.94}$$

Pentru $K=K_1$, $K'=K_2$ şi $K''=K_3$, soluţia sistemului ecuaţiilor diferenţiale este:

$$\begin{cases} q_1(t) = -F_1\left(\dfrac{M_1}{K_1}\right)^2 e^{-\frac{K_1}{M_1}t} + \dfrac{1}{2}F_1t^2 - F_1\dfrac{M_1}{K_1}\left(t - \dfrac{M_1}{K_1}\right) \\ q_2(t) = v_st \\ q_3(t) = \omega t \end{cases} \tag{5.95}$$

Punând condiţiile $t=T_2$, respectiv $t=T_3$ în expresiile coordonatelor $q_2(t)$, respectiv $q_3(t)$ rezultă timpul necesar realizării cordonului de sudură T_2 şi perioada mişcării de pendulare T_3:

$$T_2 = \frac{L}{v_s}; \quad T_3 = \frac{4\alpha}{\omega} \tag{5.96}$$

unde α este înclinarea rostului de sudare.

Din modelul geometric al robotului se obţin coordonatele punctului caracteristic:

$$\begin{aligned} x &= \ell_1 \\ y &= q_2 + \ell_2 + \ell_3 + \ell_4 \\ z &= q_1 + \ell_0 \end{aligned} \tag{5.97}$$

Abaterea de la poziţia dorită este dată de:

$$|e_p| = \sqrt{dx^2 + dy^2 + dz^2} \tag{5.98}$$

unde:

$$\begin{aligned} dx &= 0 \\ dy &= dq_2 \\ dz &= dq_1 \end{aligned} \qquad \text{sau} \qquad \begin{aligned} dx &= 0 \\ dy &= v_sdt \\ dz &= \left[F_1\frac{M_1}{K_1}\left(e^{-\frac{K_1}{M_1}t} - 1\right) + F_1t\right]dt \end{aligned} \tag{5.99}$$

Eroarea de poziţie rezultă:

$$|e_p| = \sqrt{v_s^2 + F_1^2\left[\frac{M_1}{K_1}\left(e^{-\frac{K_1}{M_1}} - 1\right) + t\right]^2}\,dt \tag{5.100}$$

BUPT

5.6. INFLUENȚA VIBRAȚIILOR DIN CUPLELE UNUI ROBOT ASUPRA POZIȚIEI EFECTORULUI FINAL

Modelarea roboților, indiferent de natura acesteia (geometrică, diferențială sau dinamică), consideră că, din punct de vedere structural, un robot este constituit din elemente rigide, legate între ele prin cuple cinematice, care la rândul lor acționează "rigid", în sensul că, în intervalul prescris, variabilele cuplelor cinematice respectă strict legea de variația impusă.

În realitate însă, lucrurile nu se petrec în acest mod, datorită unui ansamblu de factori:

- elasticitatea transmisiilor mecanice reale;
- caracterul brusc al cuplării motoarelor de acționare;
- intervenția cu caracter de ciocnire a limitatoarelor de cursă, etc.

Toți factorii enumerați introduc vibrații în sistemul pe care îl reprezintă robotul, ceea ce conduce la abateri dinamice ale efectorului final de la traiectoria impusă [D51], [N8]. Rezultă astfel oportunitatea studiului influenței mișcărilor vibratorii, introduse în cuplele cinematice de intervenția limitatoarelor de cursă, asupra poziționării pe traiectorie a efectorului final. Se iau în considerare doar vibrațiile produse de-a lungul, respectiv în jurul axelor robotului, nu și vibrațiile transversale ale elementelor robotului, care pot fi cuprinse într-o matrice de rigiditate modificată a acestuia [D4], [M3]. Se pot estima astfel, prin utilizarea ecuațiilor lui Lagrange de speța a doua, forțele, respectiv momentele ce trebuie introduse în sistem pentru corectarea abaterilor.

5.6.1. PREZENTAREA METODEI

Fie un robot cu *n* grade de libertate, asigurate de *n* cuple cinematice simple, de translație și de rotație. Studiind mișcarea robotului, sub acțiunea forțelor și momentelor exterioare, rezultă, prin utilizarea ecuațiilor lui Lagrange spre exemplu, legile de variație ale variabilelor cuplelor cinematice $q_i = q_i(t)$. Acesta este cazul ideal, în care legile de mișcare ale cuplelor nu sunt perturbate de alte interacțiuni ale robotului cu exteriorul.

În această situație, poziția și orientarea efectorului final sunt date de elementele matricei $\mathbf{G}_n^0 = \mathbf{T}_1^0 \ldots \mathbf{T}_n^{n-1}$.

Când, în timpul mișcării intervin limitatoarele de cursă ale cuplelor, fenomenul care se produce poate fi asimilat unei ciocniri elastice ce determină modificarea legii de mișcare a cuplei respective prin introducerea unei vibrații de amplitudine redusă, dar de pulsație foarte mare [D4], [N8].

Deoarece fenomenul se reproduce în mod aleator la toate cuplele robotului și se transmite prin elementele acestuia (considerate rigide) la ansamblul celorlalte cuple, este evident că la fiecare cuplă se înregistrează o suprapunere de mișcări vibratorii parazite, a căror rezultantă se presupune de forma $\xi_i = A_i \sin p_i t$. Astfel, variabilele cuplelor cinematice se vor modifica devenind $q_i + A_i \sin p_i t$.

Datorită amplitudinilor A_i reduse, se neglijează influența vibrației asupra legii de mișcare a cuplei, dar nu se poate neglija această influență asupra vitezei cuplei cinematice $\dot{q}_i$. Dacă în fiecare cuplă viteza acesteia devine $\dot{q}_i + \dot{\xi}_i$, atunci componentele vitezei

BUPT

efectorului final, exprimate în funcție de vitezele cuplelor cinematice prin intermediul matricei jacobiene a robotului [D51], devin:

$$\left[v_x^{G_n} \quad v_y^{G_n} \quad v_z^{G_n} \quad \omega_x^{G_n} \quad \omega_y^{G_n} \quad \omega_z^{G_n}\right]^T = \mathbf{J_r} \cdot \left[\dot{q}_1 + \dot{\xi}_1 \quad \dots \quad \dot{q}_i + \dot{\xi}_i \quad \dots \quad \dot{q}_n + \dot{\xi}_n\right]^T \quad (5.101)$$

Ținând seama de expresia matricei $\mathbf{J_r}$, ale cărei coloane sunt vectorii diferențiali ai mișcării, produși de influența fiecărei cuple asupra efectorului final și pentru care există relații de calcul stabilite [P5], [D16], rezultă că proiecțiile vitezei efectorului final, în raport cu propriul sistem de referință, sunt:

$$\begin{aligned}
v_x^{G_n} &= \sum_{i=1}^{n} dx_i^{G_n}\left(\dot{q}_i + \dot{\xi}_i\right) \\
v_y^{G_n} &= \sum_{i=1}^{n} dy_i^{G_n}\left(\dot{q}_i + \dot{\xi}_i\right) \\
v_z^{G_n} &= \sum_{i=1}^{n} dz_i^{G_n}\left(\dot{q}_i + \dot{\xi}_i\right) \\
\omega_x^{G_n} &= \sum_{i=1}^{n} \delta x_i^{G_n}\left(\dot{q}_i + \dot{\xi}_i\right) \\
\omega_y^{G_n} &= \sum_{i=1}^{n} \delta y_i^{G_n}\left(\dot{q}_i + \dot{\xi}_i\right) \\
\omega_x^{G_n} &= \sum_{i=1}^{n} \delta z_i^{G_n}\left(\dot{q}_i + \dot{\xi}_i\right)
\end{aligned} \quad (5.102)$$

Pentru a putea compara aceste viteze cu vitezele teoretic impuse pe traiectoria proiectată, expresiile (5.102) se transcriu în raport cu sistemul de referință fix. Astfel, viteza pe traiectoria reală a efectorului final se exprimă față de reperul fix prin matricea:

$$\begin{bmatrix} v_{x_r}^0 \\ v_{y_r}^0 \\ v_{z_r}^0 \\ 0 \end{bmatrix} = \mathbf{G}_n^0 \cdot \begin{bmatrix} v_x^{G_n} \\ v_y^{G_n} \\ v_z^{G_n} \\ 0 \end{bmatrix} \quad (5.103)$$

iar viteza unghiulară reală a efectorului final, prin matricea:

$$\begin{bmatrix} \omega_{x_r}^0 \\ \omega_{y_r}^0 \\ \omega_{z_r}^0 \\ 0 \end{bmatrix} = \mathbf{G}_n^0 \cdot \begin{bmatrix} \omega_x^{G_n} \\ \omega_y^{G_n} \\ \omega_z^{G_n} \\ 0 \end{bmatrix} \quad (5.104)$$

Vitezele teoretic impuse se cunosc, evident tot față de sistemul de referință fix. Egalarea vitezelor teoretice cu cele reale (relațiile (5.103) și (5.104)), rezultate din considerarea mișcărilor parazite ξ_i, va conduce, în cazul sistemului, la șase relații:

BUPT

$$
\begin{aligned}
v^0_{x_t} &= n_x v^{G_n}_x + o_x v^{G_n}_y + a_x v^{G_n}_z \\
v^0_{y_t} &= n_y v^{G_n}_x + o_y v^{G_n}_y + a_y v^{G_n}_z \\
v^0_{z_t} &= n_z v^{G_n}_x + o_z v^{G_n}_y + a_z v^{G_n}_z \\
\omega^0_{x_t} &= n_x \omega^{G_n}_x + o_x \omega^{G_n}_y + a_x \omega^{G_n}_z \\
\omega^0_{y_t} &= n_y \omega^{G_n}_x + o_y \omega^{G_n}_y + a_y \omega^{G_n}_z \\
\omega^0_{z_t} &= n_z \omega^{G_n}_x + o_z \omega^{G_n}_y + a_z \omega^{G_n}_z
\end{aligned}
\qquad (5.105)
$$

Acestea permit exprimarea vitezelor $\dot{\xi}_i$ în funcție de derivatele variabilelor cuplelor cinematice $\dot{q}_i$ şi de vitezele impuse teoretic efectorului final. Introducerea mărimilor $\dot{\xi}_i$ obținute din relațiile (5.105), în ecuațiile diferențiale rezultate din scrierea ecuațiilor lui Lagrange de speța a II-a, va permite identificarea momentelor şi forțelor suplimentare care apar. Anularea acestora conduce la stabilirea legilor de variație $q_i=q_i(t)$, deci la obținerea traiectoriei dorite.

5.6.2. EXEMPLIFICAREA METODEI PENTRU UN ROBOT DE TIP RT

Se consideră robotul de tip RT din fig.5.16. Prin considerarea sistemelor de referință conform convenției Hartenberg-Denavit, matricea care exprimă poziția şi orientarea efectorului final este:

$$
\mathbf{G}^0_2 = \begin{bmatrix}
\cos q_1 & 0 & -\sin q_1 & -q_2 \sin q_1 \\
\sin q_1 & 0 & \cos q_1 & q_2 \cos q_1 \\
0 & -1 & 0 & \ell_1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\qquad (5.106)
$$

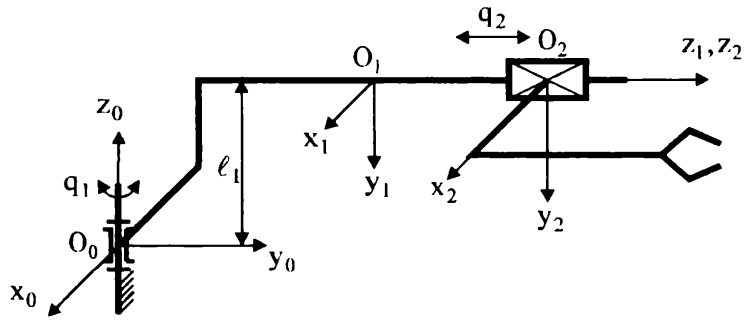


Fig.5.16. Schema cinematică structurală a robotului de tip RT

iar matricea jacobiană este:

BUPT

$$\mathbf{J}_r = \begin{bmatrix} -q_2 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ -1 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.107}$$

Considerând viteze parazit la ambele cuple cinematice, matricea (5.101) a vitezelor efectorului final este:

$$\begin{bmatrix} v_x^{G_2} \\ v_y^{G_2} \\ v_z^{G_2} \\ \omega_x^{G_2} \\ \omega_y^{G_2} \\ \omega_z^{G_2} \end{bmatrix}^T = \mathbf{J}_r \cdot \begin{bmatrix} \dot{q}_1 + \dot{\xi}_1 \\ \dot{q}_2 + \dot{\xi}_2 \end{bmatrix} = \begin{bmatrix} -q_2\left(\dot{q}_1 + \dot{\xi}_1\right) \\ 0 \\ \dot{q}_2 + \dot{\xi}_2 \\ 0 \\ -\left(\dot{q}_1 + \dot{\xi}_1\right) \\ 0 \end{bmatrix} \tag{5.108}$$

Transcrierea relației (5.108), folosind relațiile (5.103) şi (5.104), în raport cu sistemul de referință fix, conduce la proiecțiile reale:

$$\begin{bmatrix} v_{x_r}^0 \\ v_{y_r}^0 \\ v_{z_r}^0 \\ \omega_{x_r}^0 \\ \omega_{y_r}^0 \\ \omega_{z_r}^0 \end{bmatrix} = \begin{bmatrix} -q_2\left(\dot{q}_1 + \dot{\xi}_1\right)\cos q_1 - \left(\dot{q}_2 + \dot{\xi}_2\right)\sin q_1 \\ -q_2\left(\dot{q}_1 + \dot{\xi}_1\right)\sin q_1 + \left(\dot{q}_2 + \dot{\xi}_2\right)\cos q_1 \\ 0 \\ 0 \\ 0 \\ \dot{q}_1 + \dot{\xi}_1 \end{bmatrix} \tag{5.109}$$

Se presupune că efectorul final trebuie să execute o traiectorie plană, ale cărei ecuații parametrice sunt:

$$\begin{aligned} x_0 &= x_0(t) \\ y_0 &= y_0(t) \end{aligned} \tag{5.110}$$

cu viteza unghiulară impusă:

$$\omega_0 = \omega_0 \mathbf{k}_0 \tag{5.111}$$

Viteza efectorului final pe traiectorie va fi:

$$\begin{aligned} v_{0_x} &= \dot{x}_0 \\ v_{0_y} &= \dot{y}_0 \end{aligned} \tag{5.112}$$

Egalând relația (5.109) cu viteza teoretică $\begin{bmatrix} \dot{x}_0 & \dot{y}_0 & 0 & 0 & 0 & \omega_0 \end{bmatrix}^T$, rezultă relațiile (5.105) pentru robotul studiat:

BUPT

$$\begin{aligned}\dot{x}_0 &= -q_2\left(\dot{q}_1+\dot{\xi}_1\right)\cos q_1-\left(\dot{q}_2+\dot{\xi}_2\right)\sin q_1\\ \dot{y}_0 &= -q_2\left(\dot{q}_1+\dot{\xi}_1\right)\sin q_1+\left(\dot{q}_2+\dot{\xi}_2\right)\cos q_1\\ \omega_0 &= \dot{q}_1+\dot{\xi}_1\end{aligned} \tag{5.113}$$

Pentru obținerea ecuațiilor lui Lagrange de speța a II-a pentru robotul studiat se scrie expresia energiei cinetice a sistemului ideal:

$$E_{c_{ideal}} = \frac{1}{2}\dot{q}_1^2\left[J_{z_0}^{(1)}+J_{//z_0}^{(2)}+m_2q_2^2\right]+\frac{1}{2}m_2\dot{q}_2^2 \tag{5.114}$$

şi a sistemului cu perturbații:

$$E_c = \frac{1}{2}\left(\dot{q}_1+\dot{\xi}_1\right)^2\left[J_{z_0}^{(1)}+J_{//z_0}^{(2)}+m_2\left(q_2+\xi_2\right)^2\right]+\frac{1}{2}m_2\left(\dot{q}_2+\dot{\xi}_2\right)^2 \tag{5.115}$$

Calculând derivatele şi neglijând mărimea ξ_2 față de q_2 în expresiile derivatelor respective, rezultă ecuațiile diferențiale corespunzătoare cazului ideal:

$$\begin{cases}M_1 = \ddot{q}_1\left[J_{z_0}^{(1)}+J_{//z_0}^{(2)}+m_2q_2^2\right]+2m_2q_2\dot{q}_1\dot{q}_2\\ F_2 = m_2\ddot{q}_2-m_2\dot{q}_1^2q_2\end{cases} \tag{5.116}$$

şi cazului cu perturbații:

$$\begin{cases}M_1+M_{1_{supl}} = \left(\ddot{q}_1+\ddot{\xi}_1\right)\left[J_{z_0}^{(1)}+J_{//z_0}^{(2)}+m_2q_2^2\right]+2m_2q_2\left(\dot{q}_1+\dot{\xi}_1\right)\left(\dot{q}_2+\dot{\xi}_2\right)\\ F_2+F_{2_{supl}} = m_2\left(\ddot{q}_2+\ddot{\xi}_2\right)-m_2\left(\dot{q}_1+\dot{\xi}_1\right)^2q_2\end{cases} \tag{5.117}$$

Scăzând ecuațiile (5.116) din ecuațiile (5.117), rezultă:

$$\begin{cases}M_{1_{supl}} = \ddot{\xi}_1\left[J_{z_0}^{(1)}+J_{//z_0}^{(2)}+m_2q_2^2\right]+2m_2q_2\left[\dot{\xi}_2\left(\dot{q}_1+\dot{\xi}_1\right)+\dot{q}_2\dot{\xi}_1\right]\\ F_{2_{supl}} = m_2\ddot{\xi}_2-m_2\left(2\dot{q}_1+\dot{\xi}_1\right)q_2\dot{\xi}_1\end{cases} \tag{5.118}$$

Solicitările suplimentare $M_{1_{supl}}$ şi $F_{2_{supl}}$ se pot exprima prin intermediul mărimilor impuse (5.111) şi (5.112). Astfel, se observă din a treia ecuație din (5.113):

$$\dot{\xi}_1 = \omega_0-\dot{q}_1 \tag{5.119}$$

iar relațiile (13), ridicate la pătrat şi adunate, conduc la o ecuație de gradul 2 în $\dot{\xi}_2$, care are soluțiile:

$$\dot{\xi}_2 = \dot{q}_2\pm\sqrt{\dot{x}_0^2+\dot{y}_0^2-q_2^2\omega_0^2} \tag{5.120}$$

În expresia (5.120) se va lua în considerare doar semnul (-) pentru că numai acesta asigură ca în cazul ideal $\dot{\xi}_2 = 0$.

Derivând relația (5.120) în raport cu timpul şi înlocuind-o, împreună cu relația (5.119), în relația (5.118), în expresia forței suplimentare, se obține:

BUPT

$$F_{2_{supl}} = m_2\left[\ddot{q}_2 - \frac{\dot{x}_0\ddot{x}_0 + \dot{y}_0\ddot{y}_0 - q_2\dot{q}_2\omega_0^2 - q_2^2\omega_0\dot{\omega}_0}{\sqrt{\dot{x}_0^2 + \dot{y}_0^2 - q_2^2\omega_0^2}}\right] - m_2q_2\left(\omega_0^2 - \dot{q}_1^2\right) \qquad (5.121)$$

Similar, relaţia (5.119), derivata sa şi relaţia (5.120), introduse în prima ecuaţie (5.118) conduc la:

$$M_{1_{supl}} = \left(\dot{\omega}_0 - \ddot{q}_1\right)\left[J_{z_0}^{(1)} + J_{//z_0}^{(2)} + m_2q_2^2\right] + 2m_2q_2\left[\dot{q}_2\left(\omega_0 - \dot{q}_1\right) + \omega_0\left(\dot{q}_2 - \sqrt{\dot{x}_0^2 + \dot{y}_0^2 - q_2^2\omega_0^2}\right)\right] \qquad (5.122)$$

Anularea relaţiilor (5.121) şi (5.122) conduce la un sistem de două ecuaţii diferenţiale din care se vor determina $q_1=q_1(t)$ şi $q_2=q_2(t)$ în funcţie de caracteristicile (5.110) şi (5.111) impuse pentru mişcarea efectorului final.

Expresiile (5.119) şi (5.120) scrise sub forma:

$$\begin{aligned}\dot{\xi}_1 &= A_1p_1\cos p_1t = \omega_0 - \dot{q}_1\\ \dot{\xi}_2 &= A_2p_2\cos p_2t = \dot{q}_2 - \sqrt{\dot{x}_0^2 + \dot{y}_0^2 - q_2^2\omega_0^2}\end{aligned} \qquad (5.123)$$

permit evaluarea mărimilor caracteristice ale vibraţiilor introduse în cuple şi calculul pe această bază al amortizorilor cu care pot fi echipate limitatoarele de cursă în caz de necesitate.

5.7. STUDIUL ERORILOR DINAMICE DATORATE DEFORMAŢIILOR ELEMENTELOR UNUI ROBOT

Roboţii cu elemente flexibile sunt sisteme dinamice continue cu *n* grade de libertate, a căror mişcare este ilustrată printr-un sistem de ecuaţii diferenţiale neliniare, cu coeficienţi variabili, ale căror soluţii exacte sunt aproape imposibil de determinat. Pentru obţinerea unor soluţii aproximative, cât mai apropiate de realitate, sistemul mecanic trebuie discretizat.

Literatura de specialitate este bogată în abordări ale acestei probleme [A1], [C3], [D36], [M7], [M8], [S7], [T4], fie considerând că un element oarecare al robotului este format din elemente finite legate între ele prin cuple false, ce permit astfel modelarea deformării, fie considerând împărţirea robotului în atâtea elemente finite câte elemente fizice are, fie considerând fiecare element al robotului ca un ansamblu de elemente finite în care nodurile sunt în echilibru şi au deplasări compatibile, etc.

Se propune o metodă de determinare a erorilor dinamice datorate deformaţiilor elementelor unui robot cu *n* grade de libertate, ale cărui elemente sunt deformabile, utilizând metoda deplasărilor. Se consideră că elementele robotului sunt bare, iar discretizarea structurii asimilează fiecărei bare un element finit. Nodurile structurii sunt cuplele cinematice ale robotului [D31], [T16].

5.7.1. RELAŢIA FORŢE-DEPLASĂRI PENTRU O STRUCTURĂ IDEALIZATĂ

Pentru studiul flexibilităţii unui robot serial se consideră lanţul cinematic al acestuia şi se ataşează fiecărui element câte un sistem de referinţă (fig.5.17), conform

BUPT

uneia din convenţiile folosite în robotică, iar pentru determinarea relaţiei forţe-deplasări se folosesc tehnicile şi metodele deja dezvoltate de teoria dinamicii structurilor.

Torsorul unui sistem oarecare de forţe, calculat într-un punct O, poate fi exprimat prin relaţia (5.55).

În timpul îndeplinirii sarcinii, asupra efectorului final al robotului acţionează un torsor τ_{EF}^{0} exprimat în raport cu sistemul de referinţă fix. Torsorul echivalent, care acţionează în cupla cinematică *i*, exprimat în raport cu propriul sistem de referinţă S_i se determină cu relaţia:

$$\tau_i^i = \left[\left(\mathbf{J}_i^0\right)^{-1}\right]^T \cdot \tau_{EF}^0 \qquad (5.124)$$

unde $\mathbf{J}_i^0$ este Jacobianul sistemului S_i.

Se consideră un element oarecare *i* al robotului (fig.5.18). Folosind relaţia (5.124), se pot determina torsorul τ_i^i exprimat în raport cu S_i şi torsorul τ_{i+1}^{i+1} în raport cu S_{i+1}. Cunoscând torsorul τ_{i+1}^{i+1}, rezultă torsorul τ_{i+1}^i:

$$\tau_{i+1}^i = \mathbf{J}_{i+1}^i \tau_{i+1}^{i+1} \qquad (5.125)$$

unde $\mathbf{J}_{i+1}^i$ este Jacobianul sistemului S_{i+1} în raport cu S_i.

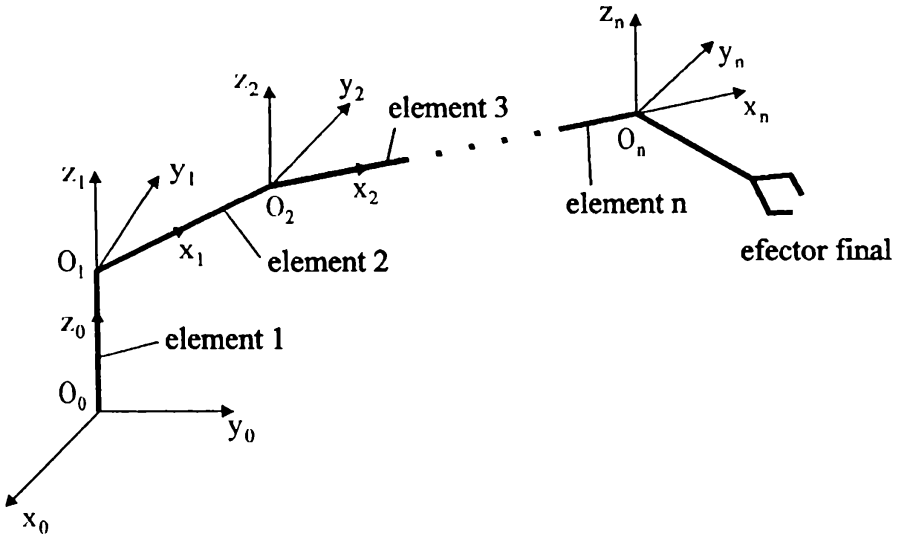


Fig.5.17. Lanţul cinematic şi sistemele de referinţă ataşate elementelor robotului

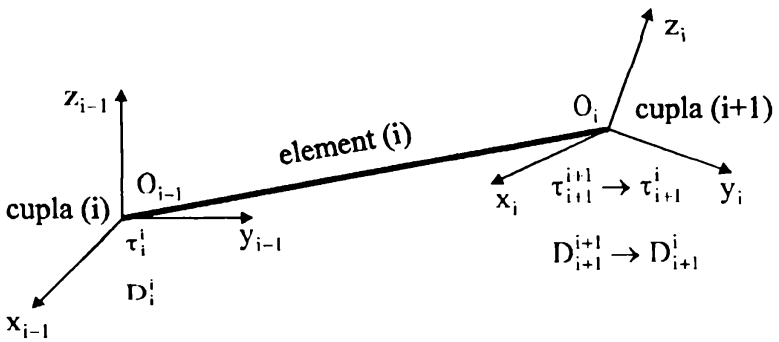


Fig.5.18. Un element oarecare al robotului şi torsorii care acţionează în cuple

BUPT

Astfel, asupra unui element al robotului acționează un torsor $\underline{\tau}_i$ exprimat printr-un vector de dimensiune 12x1:

$$\underline{\tau}_i = \left[\tau_i^i \;\; \tau_{i+1}^i\right]^T \tag{5.126}$$

Torsorul care acționează asupra întregului robot este:

$$\underline{\tau} = \left[\underline{\tau}_1 \; \ldots \; \underline{\tau}_i \; \ldots \; \underline{\tau}_n\right]^T \tag{5.127}$$

În fiecare sistem S_i se poate exprima un vector al deplasărilor liniare $\mathbf{d}_i^i$, respectiv un vector al deplasărilor unghiulare δ_i^i:

$$\mathbf{d}_i^i = \left[d_{x_i} \;\; d_{y_i} \;\; d_{z_i}\right]^T \tag{5.128}$$

$$\delta_i^i = \left[\delta_{x_i} \;\; \delta_{y_i} \;\; \delta_{z_i}\right]^T \tag{5.129}$$

rezultând un vector al deplasărilor $\mathbf{D}_i^i$ de dimensiune 6x1:

$$\mathbf{D}_i^i = \left[\mathbf{d}_i^i \;\; \delta_i^i\right]^T \tag{5.130}$$

În mod similar ca în cazul torsorilor, se pot exprima un vector $\underline{\mathbf{D}}_i$, pentru elementul *i*, respectiv un vector $\underline{\mathbf{D}}$, pentru întregul robot:

$$\underline{\mathbf{D}}_i = \left[\mathbf{D}_i^i \;\; \mathbf{D}_{i+1}^i\right]^T \tag{5.131}$$

$$\underline{\mathbf{D}} = \left[\underline{\mathbf{D}}_1 \; \ldots \; \underline{\mathbf{D}}_i \; \ldots \; \underline{\mathbf{D}}_n\right]^T \tag{5.132}$$

Considerând că variațiile poziției și orientării fiecărui element al robotului sunt datorate flexibilității elementelor, se poate scrie relația:

$$\tau = \underline{\mathbf{K}}\,\underline{\mathbf{D}} \tag{5.133}$$

unde $\underline{\mathbf{K}}$ este matricea de rigiditate a robotului (matrice pătratică, simetrică, de dimensiune 12nx12n). Inversa acestei matrice se numește matricea de flexibilitate a robotului $\underline{\mathbf{f}} = \underline{\mathbf{K}}^{-1}$. Astfel deformațiile elementelor rezultă din:

$$\underline{\mathbf{D}} = \underline{\mathbf{K}}^{-1}\,\tau = \underline{\mathbf{f}}\,\tau \tag{5.134}$$

Dacă vectorul deformațiilor elementelor robotului este cunoscut, se poate determina vectorul deviațiilor efectorului final $\mathbf{D}_{EF}$ de la poziția și orientarea impuse.

5.7.2. Relația forțe-deplasări pentru un element finit de structură

Pentru fiecare element al structurii se poate determina o matrice de rigiditate $\underline{K}_i$, exprimată în raport cu un sistem de referință local, respectiv în raport cu S_i. Dacă se izolează elementele în care a fost discretizată structura, fiecare un asemenea element este considerat ca un element finit, elastic, sub acțiunea unui sistem de forțe generalizate $\underline{F}_i$ aplicate în noduri, cărora le corespunde un sistem de deplasări generalizate $\underline{d}_i$ (fig.5.19):

$$\underline{F} = \left[F_{x_1}\, F_{y_1}\, F_{z_1}\, M_{x_1}\, M_{y_1}\, M_{z_1}\, F'_{x_1}\, F'_{y_1}\, F'_{z_1}\, M'_{x_1}\, M'_{y_1}\, M'_{z_1} \ldots M'_{z_n}\right]^T \tag{5.135}$$

$$\underline{d} = \left[d_{x_1}\, d_{y_1}\, d_{z_1}\, \delta_{x_1}\, \delta_{y_1}\, \delta_{z_1}\, d'_{x_1}\, d'_{y_1}\, d'_{z_1}\, \delta'_{x_1}\, \delta'_{y_1}\, \delta'_{z_1} \ldots \delta'_{z_n}\right]^T \tag{5.136}$$

BUPT

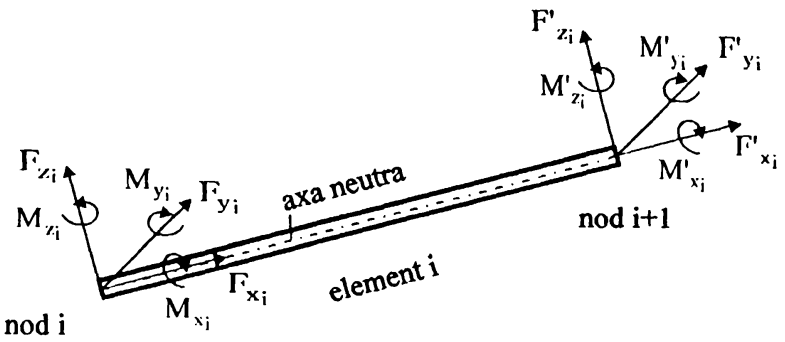


Fig.5.19. Sistemul forțelor generalizate aplicate în noduri

Atât $\underline{F}$, cât şi $\underline{d}$ au dimensiunile 12nx1. Între vectorul forțelor generalizate $\underline{F}$ care acționează în noduri şi vectorul deplasărilor generalizate $\underline{d}$ există relația:

$$\underline{F} = \underline{K}\,\underline{d} \tag{5.137}$$

unde $\underline{K}$ este matricea de rigiditate a sistemului, ale cărei elemente sunt matricele $\underline{K}_i$, exprimate pentru fiecare element în raport cu propriul sistem de referință:

$$\underline{K} = \begin{bmatrix} \underline{K}_1 & & \\ & \underline{K}_2 & \\ & & \underline{K}_n \end{bmatrix} \tag{5.138}$$

Deoarece forțele şi deplasările generalizate din noduri sunt exprimate în raport cu sistemele de referință ataşate elementelor, conform metodelor uzuale de calcul dinamic al structurilor, sistemului de referință ataşat unui element oarecare i se va aplica simbolic, câte o translație spre fiecare din cele două extremități ale elementului, rezultând astfel direcțiile de calcul corespunzătoare forțelor $\underline{F}$ şi deplasărilor $\underline{d}$. Pentru ca orientarea lui S_i să corespundă cu ipotezele din calculul dinamic al structurilor se verifică dacă una din axele lui S_i coincide cu axa longitudinală a elementului (de preferat axa Ox). În caz contrar, se aplică lui S_i o rotație corespunzătoare, exprimată printr-o matrice cu elemente constante.

Folosind metoda propusă, elementele corespunzătoare din matricelor $\underline{\mathbf{K}}$ şi $\underline{K}$ sunt exprimate în aceleaşi sisteme de referință, şi deci, nu mai este necesară transformarea acestor matrice în raport cu sistemul de referință fix.

Totuşi, dacă vectorii din ecuația (5.134) trebuie exprimați în raport cu S_0, între vectorul forțelor generalizate τ şi vectorul forțelor generalizate care acționează în noduri $\underline{F}$, respectiv între vectorul deplasărilor generalizate $\mathbf{D}$ şi vectorul deplasărilor generalizate din noduri $\underline{d}$, există relațiile:

$$\underline{d} = \mathbf{C}\,\mathbf{D} \tag{5.139}$$

$$\tau = \mathbf{C}^{\mathrm{T}}\,\underline{F} \tag{5.140}$$

Matricea $\mathbf{C}$, de dimensiune 12nx12n, din relațiile (5.139) şi (5.140) se numeşte matrice de compatibilitate geometrică şi este exprimată în funcție de coordonatele cuplelor cinematice.

Relația între forțe şi deplasări a fost studiată în mod asemănător şi de Meghdari şi Shahinpoor, dar aceştia au tratat doar cazul când cuplele cinematice ale robotului sunt

BUPT

cuple de translație şi au exemplificat determinarea matricei de rigiditate pentru un lanț cinematic cu două cuple de rotație, cu două elemente.

Matricea de rigiditate a sistemului se determină cu relația:

$$\mathbf{K} = \mathbf{C}^T \underline{\boldsymbol{K}} \mathbf{C} \tag{5.141}$$

Ținând cont de faptul că fiecare element este integrat într-o structură spațială oarecare, astfel încât $\underline{\boldsymbol{K}}_i$ trebuie să reflecte corect comportamentul elementului într-un spațiu cu dimensiuni adecvate, Massonnet [6] propune o metodă de pregătire a matricei de rigiditate în vederea "asamblării". Rezultă astfel o matrice de rigiditate expandată, care se obține din matricea $\underline{\boldsymbol{K}}_i$, prin intermediul matricei de expansiune **E**. Deoarece structura se raportează la un sistem de axe generale, matricele de rigiditate expandate ale elementelor, stabilite în S_i, trebuie să fie transformate prin intermediul unei matrice de rotație **R.** Deci, conform teoriei lui Massonnet, fiecărei matrice $\underline{\boldsymbol{K}}_i$, exprimată în SRA_i, trebuie să i se aplice două transformări pentru a putea fi utilizată pentru determinarea matricei **K** a structurii:

• Expansiunea matricei $\underline{\boldsymbol{K}}_i$, pentru a fi exprimată într-un spațiu cu aceleaşi dimensiuni ca cel care conține structura studiată, prin utilizarea matricei de expansiune **E**. Matricea de rigiditate expandată a elementului este:

$$\underline{\boldsymbol{K}}_i^E = \mathbf{E}^T \underline{\boldsymbol{K}}_i \mathbf{E} \tag{5.142}$$

• Transformarea matricei expandate pentru a o raporta la sistemul de referință fix, prin utilizarea unei matrice de rotație **R**:

$$\mathbf{K}_i = \mathbf{R}^T \underline{\boldsymbol{K}}_i^E \mathbf{R} \tag{5.143}$$

unde matricea **R** este o matrice diagonală, ale cărei elemente $\mathbf{R}_0$ exprimă rotația sistemului S_i, în jurul originii sale O_i, astfel încât să aibă aceeaşi orientare cu sistemul de referință fix. Elementele matricei $\mathbf{R}_0$ sunt cosinuşii directori ai axelor S_i exprimați în. sistemul de referință fix.

Matricea de rigiditate $\underline{\boldsymbol{K}}_i$ a unui element nu trebuie să fie expandată întotdeauna, dar trebuie să se verifice întotdeauna ca, la asamblare, toate matricele $\underline{\boldsymbol{K}}_i$ să fie exprimate într-un spațiu cu aceleaşi dimensiuni.

Având în vedere că în modelarea roboților, pentru alegerea sistemelor de referință ataşate există mai multe convenții, pentru ataşarea sistemului de referință unui element al robotului se propune convenția Hartenberg-Denavit, de altfel cea mai utilizată în cazul roboților seriali. Poziția şi orientarea unui element al robotului, precum şi ale efectorului final, se determină în acest caz, în raport cu elementul precedent, iar în final, în raport cu sistemul de referință fix, folosind matricele de transformări omogene [3].

Atunci când, în timpul îndeplinirii sarcinii, asupra efectorului final al robotului acționează un anumit torsor, se poate determina vectorul **Q** care acționează efectiv asupra fiecărui element al robotului [4].

Poziția şi orientarea sistemului de referință ataşat elementului *i* S_i, în raport cu poziția şi orientarea sistemului de referință ataşat elementului precedent S_{i-1} se exprimă printr-o matrice de transformare $\mathbf{T}_i^{i-1}$, iar în raport cu sistemul de referință fix, prin matricea de transformare $\mathbf{G}_i^0$:

$$\mathbf{G}_i^0 = \mathbf{T}_1^0 \mathbf{T}_2^1 \ldots \mathbf{T}_i^{i-1} \tag{5.144}$$

BUPT

În aceste condiții, matricea de rigiditate a elementului de structură $\underline{K}_i$ poate fi înmulțită cu o matrice de compatibilitate geometrică **C**, de dimensiune 12x12, rezultând matricea de rigiditate $\mathbf{K}_i$ exprimată în sistemul general S_0, astfel:

$$\mathbf{K}_i = \mathbf{C}^T \underline{K}_i \mathbf{C} \tag{5.145}$$

unde matricea **C** are expresia:

$$\mathbf{C} = \begin{bmatrix} \mathbf{G}_i^0 & & \\ & \mathbf{G}_i^0 & \\ & & \mathbf{G}_i^0 \end{bmatrix} \tag{5.146}$$

Matricea de rigiditate a sistemului rezultă ca o matrice diagonală, cu o expresie asemănătoare cu (5.138), ale cărei elemente sunt chiar matricele $\mathbf{K}_i$.

Folosirea matricelor de transformări omogene pentru exprimarea matricei de compatibilitate geometrică permite generalizarea aplicabilității metodei incluzând şi cazul când cuplele cinematice sunt de translație.

5.7.3. MATRICEA DE RIGIDITATE A UNUI ELEMENT DE STRUCTURĂ

După idealizarea stucturii reale într-o structură compusă din elemente discrete, urmează să se determine matricea de rigiditate a fiecărui element. Se consideră deci un element oarecare al robotului ca fiind o bară dreaptă, de lungime ℓ, cu secțiune constantă A şi neîncărcat pe lungimea considerată. Bara are modulul de elasticitate longitudinal E, modulul de elasticitate transversal G, momentul de inerție geometric axial I şi polar I_p. Axa longitudinală a elementului este axa sa neutră.

Se consideră, de asemenea, că cele două capete ale elementului sunt cuplate cu elementele adiacente prin cuple elastice, constituite de motoarele de acționare. Când cuplele cinematice ale robotului nu sunt activate, se consideră că rigiditatea corespunzătoare este aproximativ egală cu zero, în timp ce, în cazul cuplei activate, rigiditatea este finită. Se mai presupune că, un coeficient de rigiditate k_i este funcție de curentul în indusul motorului, care la rândul său depinde de cuplul motor.

La cele două extremități ale sale, elementul robotului este supus următoarelor solicitări (s-a considerat cazul când axa Ox a sistemului de referință coincide cu axa longitudinală a barei):

- forțe axiale F_{xi} şi F'_{xi};
- forțe tăietoare F_{yi}, F_{zi}, F'_{yi}, F'_{zi};
- momente de torsiune M_{xi} şi M'_{xi};
- momente de încovoiere M_{yi}, M_{zi}, M'_{yi}, M'_{zi} .

Fiecare extremitate este deci locul de aplicare a câte şase forțe generalizate, având sensurile pozitive indicate în fig.5.19. Fiecăruia din aceste eforturi îi corespunde câte o deplasare, al cărei sens pozitiv coincide cu cel al efortului corespunzător, astfel încât starea de deformare este complet definită prin: deplasarea axială d_{xi}, rotirea δ_{xi} în jurul axei Ox_i, deplasările d_{yi} şi d_{zi} după axele Oy_i şi Oz_i, rotirile δ_{yi} şi δ_{zi} în jurul axelor Oy_i şi Oz_i. Vectorul deplasărilor corespunzătoare unui element este:

$$\underline{d}_i = \left[d_{x1} \quad d_{y1} \quad d_{z1} \quad \delta_{x1} \quad \delta_{y1} \quad \delta_{z1} \quad d'_{x1} \quad d'_{y1} \quad d'_{z1} \quad \delta'_{x1} \quad \delta'_{y1} \quad \delta'_{z1} \right]^T \tag{5.147}$$

BUPT

Forţele axiale şi momentele de torsiune depind doar de deplasările corespunzătoare, dar momentele încovoietoare şi forţele tăietoare depind şi de deplasările corespunzătoare forţelor din planul xOy. Încovoierea şi forfecarea în cele două plane pot fi considerate independent una de cealaltă dacă şi numai dacă planele xOz şi xOy coincid cu planele determinate de axele principale ale secţiunii transversale.

Coeficienţii de rigiditate obţinuţi din ecuaţiile diferenţiale ale deplasărilor barei uniforme sunt integral în interiorul limitelor ipotezelor teoriei inginereşi generale a barelor supuse unor anumite solicitări. Odată determinaţi, aceşti coeficienţi permit obţinerea relaţiilor forţe-deplasări. Pentru determinarea matricei de rigiditate a unui element se utilizează metoda bazată pe rezolvarea ecuaţiei diferenţiale a deplasării, pentru fiecare din grupele de solicitări amintite, separat. Expresia matricei de rigiditate pentru un element oarecare *i* al robotului este [M9]:

$$\underline{\mathbf{K}}_i = \begin{bmatrix}
\frac{E_iA_i}{\ell_i} & & & & \text{s} & & & & & & & \\
0 & \frac{12E_iI_i\alpha_i}{(1+\psi_i)\ell_i^3} & & & \text{i} & & & & & & & \\
0 & 0 & \frac{12E_iI_i\alpha_i}{(1+\psi_i)\ell_i^3} & & & \text{m} & & & & & & \\
0 & 0 & 0 & \frac{G_iI_{pi}}{\ell_i} & & & \text{e} & & & & & \\
0 & 0 & \frac{6E_iI_i\beta_i}{(1+\psi_i)\ell_i^2} & 0 & \frac{(4+\psi)E_iI_i\gamma_i}{(1+\psi_i)\ell_i} & & & \text{t} & & & & \\
0 & \frac{6E_iI_i\beta_i}{(1+\psi_i)\ell_i^2} & 0 & 0 & 0 & \frac{(4+\psi)E_iI_i\gamma_i}{(1+\psi_i)\ell_i} & & & \text{r} & & & \\
-\frac{E_iA_i}{\ell_i} & 0 & 0 & 0 & 0 & 0 & \frac{E_iA_i}{\ell_i} & & & \text{i} & & \\
0 & -\frac{12E_iI_i\alpha_i}{(1+\psi_i)\ell_i^3} & 0 & 0 & 0 & -\frac{6E_iI_i\beta_i}{(1+\psi_i)\ell_i^2} & 0 & \frac{12E_iI_i\alpha_i}{(1+\psi_i)\ell_i^3} & & & \text{c} & \\
0 & 0 & -\frac{12E_iI_i\alpha_i}{(1+\psi_i)\ell_i^3} & 0 & \frac{6E_iI_i\beta_i}{(1+\psi_i)\ell_i^2} & 0 & 0 & 0 & \frac{12E_iI_i\alpha_i}{(1+\psi_i)\ell_i^3} & & & \\
0 & 0 & 0 & -\frac{G_iI_{pi}}{\ell_i} & 0 & 0 & 0 & 0 & 0 & \frac{G_iI_{pi}}{\ell_i} & & \\
0 & 0 & -\frac{6E_iI_i\upsilon_i}{(1+\psi_i)\ell_i^2} & 0 & \frac{(2-\psi_i)E_iI_i\eta_i}{(1+\psi_i)\ell_i} & 0 & 0 & 0 & \frac{6E_iI_i\upsilon_i}{(1+\psi_i)\ell_i^2} & 0 & \frac{(4+\psi)E_iI_i\xi_i}{(1+\psi_i)\ell_i} & \\
0 & \frac{6E_iI_i\upsilon_i}{(1+\psi_i)\ell_i^2} & 0 & 0 & 0 & \frac{(2-\psi_i)E_iI_i\eta_i}{(1+\psi_i)\ell_i} & 0 & -\frac{6E_iI_i\upsilon_i}{(1+\psi_i)\ell_i^2} & 0 & 0 & 0 & \frac{(4+\psi)E_iI_i\xi_i}{(1+\psi_i)\ell_i}
\end{bmatrix} \tag{5.148}$$

În expresia matricei $\underline{\boldsymbol{K}}_i$ s-au folosit următoarele notaţii:

$$\psi_i = 12\frac{E_iI_i}{G_iA_i\ell_i^2} \qquad k_i^* = k_ik_{i+1} + 4(k_i + k_{i+1}) + 12$$

$$\alpha_i = \frac{k_ik_{i+1} + k_i + k_{i+1}}{k_i^*} \qquad \beta_i = \frac{k_ik_{i+1} + 2k_i}{k_i^*} \tag{5.149}$$

$$\gamma_i = \frac{k_ik_{i+1} + 3k_i}{k_i^*} \qquad \eta_i = \frac{k_ik_{i+1}}{k_i^*}$$

$$\upsilon_i = \frac{k_ik_{i+1} + 2k_{i+1}}{k_i^*} \qquad \xi_i = \frac{k_ik_{i+1} + 3k_{i+1}}{k_i^*}$$

BUPT

5.7.4. ALGORITM PENTRU CALCULUL DEFORMAȚIILOR

Pentru studiul deformațiilor elementelor unui robot serial cu un număr oarecare de grade de libertate s-a conceput un algoritm care permite calculul matricei de rigiditate a structurii mecanice a robotului, şi în final, calculul vectorului deformațiilor.

Datele de intrare necesare algoritmului sunt:

- numărul elementelor structurii n;
- lungimea fiecărui element ℓ_i;
- forma secțiunii fiecărui element: dreptunghiulară, circulară, inelară;
- mărimile caracteristice secțiunii: lungime, lățime sau diametru sau diametru interior şi diametru exterior;
- modulul de elasticitate longitudinal E_i;
- modulul de elasticitate transversal G_i;
- parametrii de poziție corespunzători convenției Hartenberg-Denavit: a_i, α_i, d_i, θ_i şi expresia matricei de transformare $\mathbf{T}_i^{i-1}$;
- matricea de transformare generală $\mathbf{G}_n^0$ la momentul considerat;
- torsorul care acționează asupra efectorului final.

Etapele de calcul ale algoritmului sunt:

- aria secțiunii fiecărui element A_i;
- momentul de inerție geometric axial I_i;
- momentul de inerție geometric polar I_{pi};
- vectorul coordonatelor cuplelor cinematice $\mathbf{q}$ pe baza modelului geometric invers;
- matricele de transformări $\mathbf{T}_i^{i-1}$;
- matricele de transformări $\mathbf{G}_i^0$;
- matricea de compatibilitate geometrică $\mathbf{C}$;
- torsorul în SRA_i;
- matricea de rigiditate $\underline{K}_i$ pentru fiecare element, în SRA_i;
- matricea de rigiditate $\mathbf{K}_i$;
- matricea de rigiditate a structurii mecanice a robotului $\mathbf{K}$;
- matricea de flexibilitate a structurii mecanice a robotului $\mathbf{f}$;
- vectorul deformațiilor $\mathbf{D}$.

5.8. CONCLUZII

În capitolul 1.6 a fost deja subliniat rolul modelului dinamic al unui robot în generarea mişcării, dar, mai ales, în controlul optimal al mişcării. De asemenea, modelul dinamic este folosit în concepție, simulare şi comandă.

Unul din cele mai folosite formalisme pentru determinarea modelului dinamic se bazează pe ecuațiile lui Lagrange de speța a doua. În majoritatea lucrărilor de specialitate, determinarea modelului dinamic a fost realizată pe baza unor ipoteze simplificatoare: rigiditatea elementelor, neglijarea jocurilor şi a vibrațiilor din cuple, neglijarea frecărilor, etc. În ultimii ani însă, cercetările renunță treptat la simplificările amintite.

BUPT

Au fost prezentate două exemple de determinare şi rezolvare a modelului dinamic, pentru un robot de sudare TTR şi pentru robotul KUKA, în condiţiile simplificărilor amintite. Se poate observa că, în primul caz, ecuaţiile diferenţiale ale mişcării sunt simple, decuplate, ceea ce constituie un mare avantaj pentru rezolvarea modelului dinamic invers. În cazul robotului KUKA însă, sistemul ecuaţiilor diferenţiale este neliniar, cuplat, rezolvarea sa realizându-se prin metode numerice.

Pentru impunerea restricţiilor dinamice într-o problemă de generare a mişcării, este necesar să se cunoască torsorul din cuplele robotului. Minimizarea acestuia, sau a energiei consumate, constituie un criteriu de optimizare des utilizat.

Metodele prezentate pentru determinarea torsorilor în cuplele robotului, respectiv pentru variaţia elementelor torsorului de reducere cu schimbarea sistemului de referinţă se bazează pe calculul matriceal şi se pretează la programarea pe calculator. Programul realizat necesită exprimarea matricelor de transformare $\mathbf{G}_i^{i-1}$, stabilirea valorilor coordonatelor cuplelor cinematice conducătoare q_i pentru situaţia considerată G_6^0 şi introducerea elementelor torsorului de reducere care acţionează asupra efectorului final. Caracterul de generalitate al programului şi introducerea interactivă a datelor asigură o interfaţă cu utilizatorul uşor de manevrat.

Determinarea torsorului din cuple indică ce forţe şi ce momente trebuie să acţioneze în cuplele cinematice conducătoare ale unui robot pentru ca acesta să fie în echilibru sub acţiunea torsorului din efectorul final.

Torsorul care acţionează asupra efectorului final poate fi determinat (impus) din procesul tehnologic sau poate rezulta direct din metoda de generare a mişcării, cum este cazul metodei câmpului potenţial. Dacă se consideră punctul caracteristic al efectorului final al robotului ca fiind particula supusă câmpului potenţial, în orice moment al mişcării se cunoaşte torsorul care acţionează asupra efectorului final (cap.8). Aplicând metodele prezentate se poate determina torsorul din cuple, realizând astfel comanda robotului. Dacă sunt selectate mai multe puncte de control supuse potenţialului atractiv, respectiv repulsiv, rezultă torsorii care acţionează în aceste puncte, şi aplicând metodele prezentate, se determină variaţia elementelor torsorilor în raport cu diferitele sisteme de referinţă (deci elementele mecanice care acţionează, în orice moment, asupra fiecărui element al robotului), respectiv forţele generalizate care acţionează în cuplele cinematice.

Metodele prezentate sunt valabile indiferent de numărul gradelor de libertate ale robotului. De asemenea, pe baza metodelor prezentate se poate rezolva şi problema inversă.

Unul din modurile de apreciere a preciziei parcurgerii traiectoriei unui robot, analizată cinematic, necesită controlul poziţiilor, eventual şi al orientărilor. Studiul erorilor de poziţie si de orientare se bazează pe modelele geometric şi diferenţial ale robotului, astfel încât precizia depinde de precizia acestora.

S-au tratat astfel modelele geometric şi diferenţial ale robotului pentru care s-a făcut aplicaţia concretă, în contextul utilizării convenţiei Denavit-Hartenberg. Metoda de calcul propusă este însă valabilă pentru orice altă convenţie adoptată în scrierea celor două modele.

Parcurgerea traiectoriei de către un robot real necesită controlul poziţiilor şi vitezelor instantanee, ceea ce implică cunoaşterea legilor de variaţie ale forţelor şi/sau momentelor care acţionează în cuple.

Abordarea preciziei de poziţionare în mod dinamic este mult mai complexă decât studiul cinematic al aceleiaşi probleme. Ca principiu de calcul, metoda prezentată este generală, dar aplicarea sa la un robot concret, ce evoluează pe o traiectorie precizată, este evident un caz particular.

BUPT

Datorită factorilor perturbatori, mişcarea în cuplele cinematice se abate de la legile prescrise, iar efectorul final al robotului se abate de la poziţia şi orientarea impuse. Luând în considerare vibraţiile care apar în cuplele cinematice, prin metoda prezentată, se pot determina forţele şi/sau momentele necesare corecţiei. Metoda a fost exemplificată pentru două cuple cinematice, complexitatea problemei depinzând de complexitatea modelului dinamic al robotului studiat.

Metoda prezentată este destul de laborioasă, dar ea permite realizarea traiectoriei dorite, cu viteza dorită, la un robot cu structura mecanică cunoscută, respectiv evaluarea mărimilor caracteristice ale vibraţiilor introduse în cuple şi calculul pe această bază al amortizorilor cu care pot fi echipate limitatoarele de cursă în caz de necesitate. Aşa cum a fost prezentată şi exemplificată, metoda de corectare a abaterilor dinamice de la traiectoria impusă se poate aplica oricărui tip de robot cu lanţ cinematic deschis.

În ultimii ani, modelarea dinamicii roboţilor este orientată spre considerarea influenţei flexibilităţii elementelor asupra preciziei poziţionării, respectiv a generării mişcării. Ultimul paragraf al acestui capitol constituie totodată o nouă direcţie de cercetare, şi anume "dinamica roboţilor cu elemente flexibile".

Metoda propusă permite determinarea erorilor dinamice datorate deformaţiilor elementelor. În studiul prezentat, robotul serial este tratat ca un lanţ cinematic deschis, cu elemente flexibile, capabil să rezolve o anumită sarcină într-un spaţiu tridimensional. Folosirea matricei de rigiditate a robotului, considerat ca un sistem de corpuri, discretizat, permite calculul vectorului deformaţiilor elementelor şi, pe baza acestuia, vectorul deviaţiilor efectorului final de la poziţia şi orientarea impuse. Deformaţiile elementelor structurii mecanice a robotului sunt utilizate direct pentru determinarea şi corectarea erorilor cauzate de acestea, în scopul îmbunătăţirii preciziei de parcurgere a traiectoriei.

Metoda prezintă un grad mare de generalitate, rezultat din modul de exprimare a matricei de compatibilitate geometrică. Avantajul metodei constă în faptul că, dacă alegerea sistemelor de referinţă ataşate se realizează în mod corespunzător, nu mai este necesară transformarea matricei de rigiditate. Metoda de calcul a vectorului deformaţiilor prezentată în lucrare adaptează şi aplică principiile generale din calculul dinamic al structurilor la metodele uzuale de modelare şi calcul dinamic al roboţilor.

Metoda propusă poate fi aplicată la orice robot serial, cu orice număr de elemente şi orice solicitare asupra efectorului final. De asemenea, metoda, bazată pe calculul matriceal, se pretează la a fi implementată într-un program de calcul.

Metoda poate fi aplicată atât în cazul unui robot a cărui structură mecanică are la bază un lanţ cinematic deschis, cât şi în cazul roboţilor cu structură arborescentă sau roboţilor paraleli.

BUPT

6. STUDIU EXPERIMENTAL PRIVIND CONDUCEREA UNUI ROBOT CU COMANDĂ PUNCT CU PUNCT PE O TRAIECTORIE CONTINUĂ ÎN SPAȚIUL OPERAȚIONAL

Datorită faptului că există încă o serie de roboți din generații mai vechi, care nu au un generator de traiectorie înglobat în sistemul de conducere, s-a considerat util să se realizeze off-line un generator de traiectorie care să furnizeze datele necesare şi suficiente conducerii unui robot prevăzut numai cu comandă punct cu punct, pe o traiectorie continuă în spațiul operațional.

Pentru realizarea studiului experimental s-a utilizat robotul educațional SCORBOT-ER III, care are un sistem de comandă simplu, punct cu punct. În cadrul determinărilor efectuate s-a încercat verificarea metodelor de generare a mişcării prezentate din punct de vedere teoretic în capitolele 2, 3 şi 4. În acelaşi timp, s-a urmărit să se determine performanțele robotului amintit, deoarece, în documentația tehnică acestea nu sunt specificate.

6.1. DESCRIEREA ROBOTULUI SCORBOT-ER III ŞI DETERMINAREA SPAȚIULUI DE LUCRU

Robotul SCORBOT-ER III este un robot articulat vertical, cu cinci grade de libertate (fig.6.1).

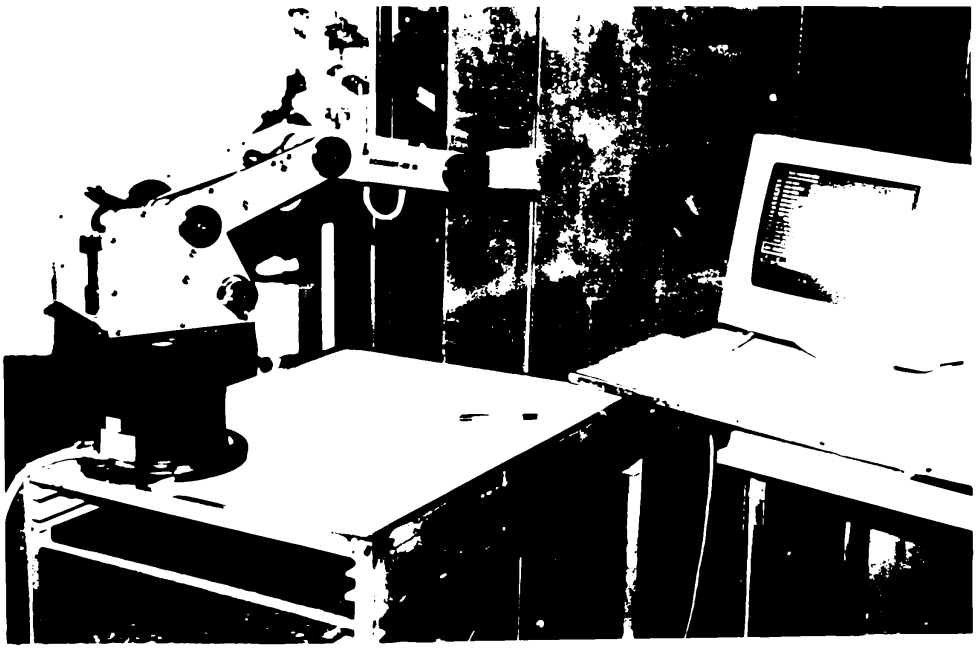


Fig.6.1. Robotul SCORBOT-ER III

BUPT

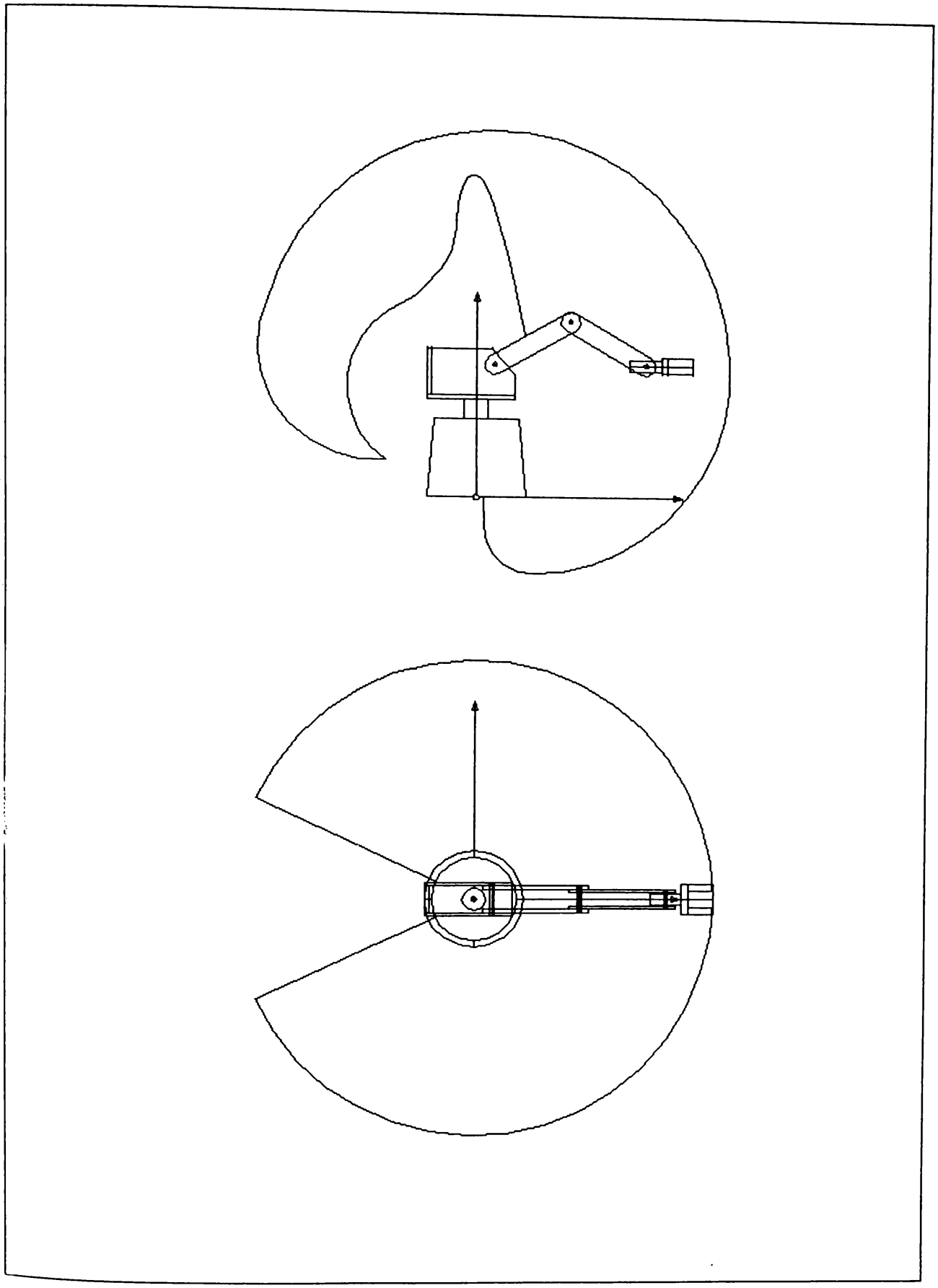


Fig.6.2. Spațiul de lucru al robotului SCORBOT-ER III

BUPT

Sistemul SCORBOT-ER III constă în principal într-o structură mecanică şi un controler. Toate cuplele cinematice ale robotului sunt cuple de rotaţie. Cupla considerată de bază, roteşte braţul robotului în plan orizontal, cele trei cuple de rotaţie SHOULDER, ELBOW şi PITCH rotesc elementele robotului în plan vertical, iar a cincea cuplă ROLL, roteşte doar efectorul final. Spaţiul de lucru corespunzător acestui robot este un spaţiu sferic (fig.6.2).

Indiferent de modul de abordare a problemei de generare (în spaţiul operaţional sau în spaţiul cuplelor), traiectoria punctului caracteristic al robotului trebuie să fie conţinută în spaţiul de lucru al acestuia.

Determinarea spaţiului de lucru s-a realizat prin încercări experimentale şi prin simulare pe calculator. Mărimile care au intervenit în determinarea acestui spaţiu au fost lungimile elemmentelor şi limitele coordonatelor cuplelor cinematice:

- rotaţia de bază: 310°;
- shoulder: +130°/-35°;
- elbow: ± 130°;
- pitch: ± 130°

Raza maximă de lucru a robotului este de 610 mm. Datorită motorului şi traductorului corespunzătoare rotaţiei ROLL, în vecinătatea bazei robotului, pe o rază de 230 mm şi o înălţime de 340 mm, există un spaţiu care nu prezintă siguranţă pentru funcţionarea robotului şi deci, este indicat să fie evitat.

Acţionarea robotului este realizată cu şase servomotoare DC, cu control în buclă închisă. Programarea robotului se poate face pe trei niveluri, cu ajutorul programului *SCORBASE*. În determinările efectuate a fost folosit programul *SCORBASE level 3*, deoarece această variantă este cea mai evoluată, ea incluzându-le pe celelalte două. Modulul TEACH POSITIONS permite controlul mişcărilor robotului şi învăţarea poziţiilor dorite. Instrucţiunile de mişcare şi memorarea poziţiilor sunt realizate prin programele scrise în SCORBASE şi înregistrate în PC RAM.

6.2. MODELUL GEOMETRIC AL ROBOTULUI SCORBOT-ER III

Într-o problemă de generare a mişcării, modelul geometric al robotului este absolut necesar pentru realizarea transformărilor de coordonate.

Schema cinematică a robotului este prezentată în figura 6.3. Cuplele cinematice ale robotului sunt: cupla 1-BASE, cupla 2-SHOULDER, cupla3-ELBOW, cupla 4-PITCH, cupla5-ROLL. Datorită arhitecturii acestui robot, poziţia şi orientarea efectorului final pot fi exprimate folosind coordonatele carteziene pentru poziţie şi unghiurile lui Bryant pentru orientare, cu observaţia că nu există a şasea axă, corespunzătoare unghiului YAW.

Modelul geometric al robotului este determinat de matricea de transformare generală $\mathbf{G}_5^0$ care exprimă poziţia şi orientarea efectorului final în raport cu sistemul de referinţă fix. Expresia matricei de transformare generală a fost prezentată în capitolul 2.1, relaţia 2.1. Pentru calculul acestei matrice s-a aplicat convenţia Denavit-Hartenberg. Pentru robotul studiat, modelul geometric este dat de următoarele ecuaţii:

BUPT

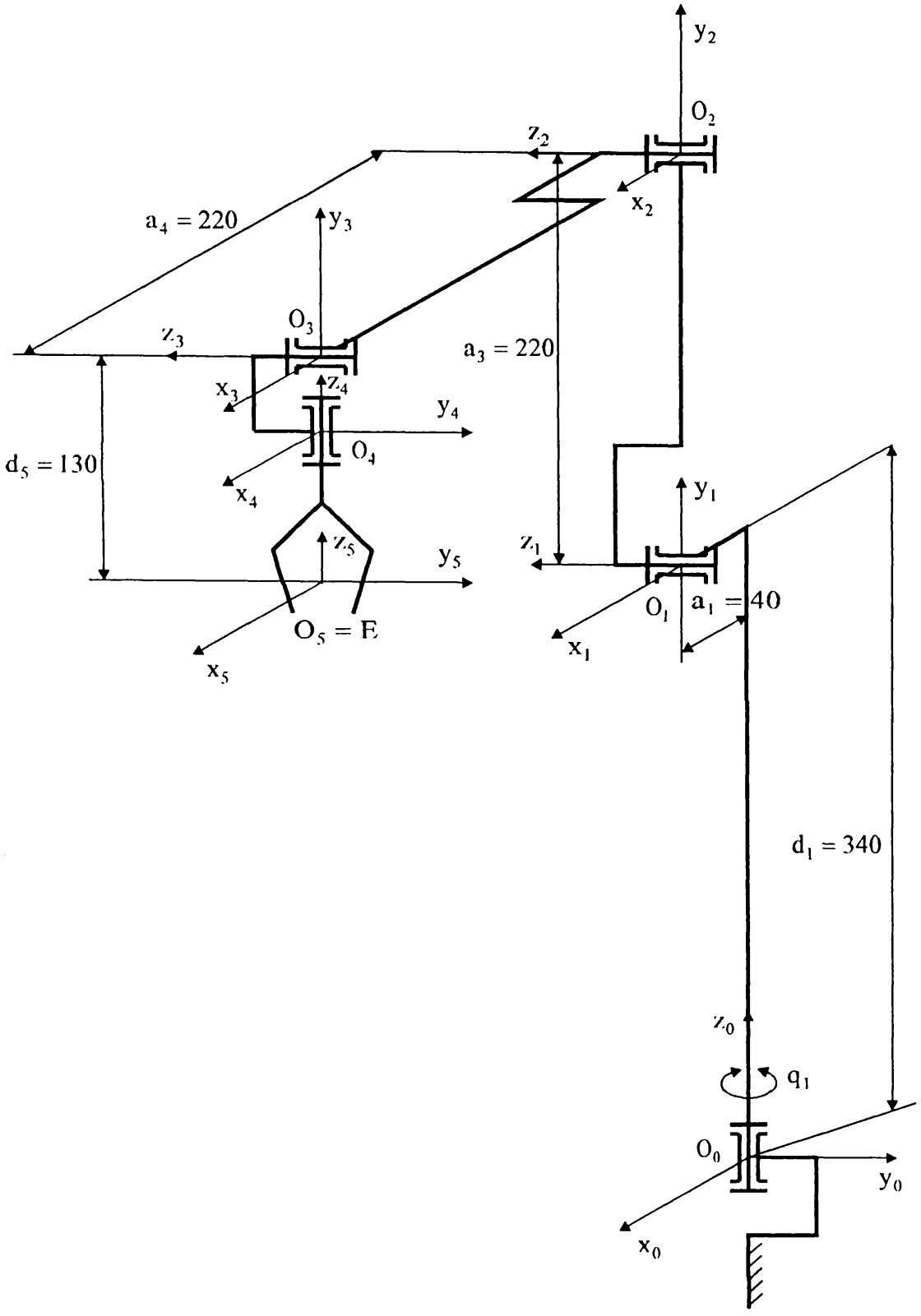


Fig.6.3. Schema cinematică a robotului SCORBOT-ER III

BUPT

$$n_x = \cos\theta_1 \cos\theta_5 \cos(\theta_2 + \theta_3 + \theta_4) - \sin\theta_1 \sin\theta_5$$

$$n_y = \sin\theta_1 \cos\theta_5 \cos(\theta_2 + \theta_3 + \theta_4) + \cos\theta_1 \sin\theta_5$$

$$n_z = \cos\theta_5 \sin(\theta_2 + \theta_3 + \theta_4)$$

$$o_x = -\cos\theta_1 \sin\theta_5 \cos(\theta_2 + \theta_3 + \theta_4) - \sin\theta_1 \cos\theta_5$$

$$o_y = -\sin\theta_1 \sin\theta_5 \cos(\theta_2 + \theta_3 + \theta_4) + \cos\theta_1 \cos\theta_5$$

$$o_z = -\sin\theta_5 \sin(\theta_2 + \theta_3 + \theta_4)$$

$$a_x = -\cos\theta_1 \sin(\theta_2 + \theta_3 + \theta_4)$$

$$a_y = -\sin\theta_1 \sin(\theta_2 + \theta_3 + \theta_4)$$

$$a_z = \cos(\theta_2 + \theta_3 + \theta_4)$$

$$p_x = \cos\theta_1 \left[-a_1 + a_2 \cos\theta_2 + a_3 \cos(\theta_2 + \theta_3) - d_5 \sin(\theta_2 + \theta_3 + \theta_4)\right]$$

$$p_y = \sin\theta_1 \left[-a_1 + a_2 \cos\theta_2 + a_3 \cos(\theta_2 + \theta_3) - d_5 \sin(\theta_2 + \theta_3 + \theta_4)\right]$$

$$p_z = d_1 + a_2 \sin\theta_2 + a_3 \sin(\theta_2 + \theta_3) + d_5 \cos(\theta_2 + \theta_3 + \theta_4)$$

Robotul SCORBOT-ER III realizează transformările de coordonate, dar valorile corespunzătoare axelor sunt exprimate în impulsuri şi nu în grade. Având în vedere acest aspect, s-au determinat relaţiile dintre numărul de impulsuri şi numărul de grade corespunzătoare fiecărei axe:

- AX 1: 9 impulsuri/grad;
- AX 2: 8.3 impulsuri/grad;
- AX 3: 8.3 impulsuri/grad;
- AX 4: 2 impulsuri/grad;
- AX 5: 2 impulsuri/grad.

6.3. GENERAREA MIŞCĂRII ÎN SPAŢIUL OPERAŢIONAL

Pentru studiul mişcării robotului în spaţiul operaţional s-au ales cinci puncte de precizie prezentate în Tabelul 6.1:

Tabel 6.1. Coordonatele punctelor de precizie în spaţiul operaţional

POZ.	X [mm]	Y [mm]	Z [mm]	PITCH [°]	ROLL [°]
1	-50	430	450	-63.56	0
2	150	370	400	-30.00	45.00
3	280	100	200	0.00	75.00
4	350	-80	360	0.00	0.00
5	50	-350	430	60.00	0.00

BUPT

Între aceste puncte de precizie, robotul se poate deplasa liber sau pe o traiectorie impusă în coordonate carteziene. În cadrul determinărilor întreprinse s-au studiat ambele variante, un accent deosebit punându-se pe al doilea caz. Datorită dimensiunilor reduse ale acestui robot, precum şi datorită faptului că masele elementelor sunt mici, toate determinările efectuate au fost abordate doar din punct de vedere cinematic.

Generarea mişcărilor libere între punctele de precizie necesită învățarea (memorarea) acestora într-o fază off line, folosind din programul robotului, din meniul principal, opțiunea TEACH POSITIONS.

Pentru controlul mişcării robotului pe o traiectorie impusă în coordonate carteziene, punctele din Tabelul 6.1 au fost folosite ca noduri de interpolare. Au fost studiate trei cazuri: interpolare liniară, interpolare spline 3 şi interpolare spline 4. În figura 6.4 sunt prezentate cele trei traiectorii carteziene studiate, iar în figura 6.5 este prezentat robotul în HOME POSITION şi în cele cinci poziții impuse. Având în vedere faptul că robotul nu este dotat cu generator de traiectorie, pentru specificarea traiectoriilor impuse în spațiul operațional, cele trei traiectorii carteziene interpolate au fost divizate într-un număr dat de segmente, în funcție de precizia dorită. Determinările au fost efectuate în două cazuri diferite:

- cu menținerea constantă a orientării efectorului final pe traiectorie (au fost menținute valorile unghiurilor PITCH şi ROLL din HOME POSITION);
- cu modificarea orientării efectorului final în timpul parcurgerii traiectoriei, de asemenea, folosind diferite tipuri de interpolări.

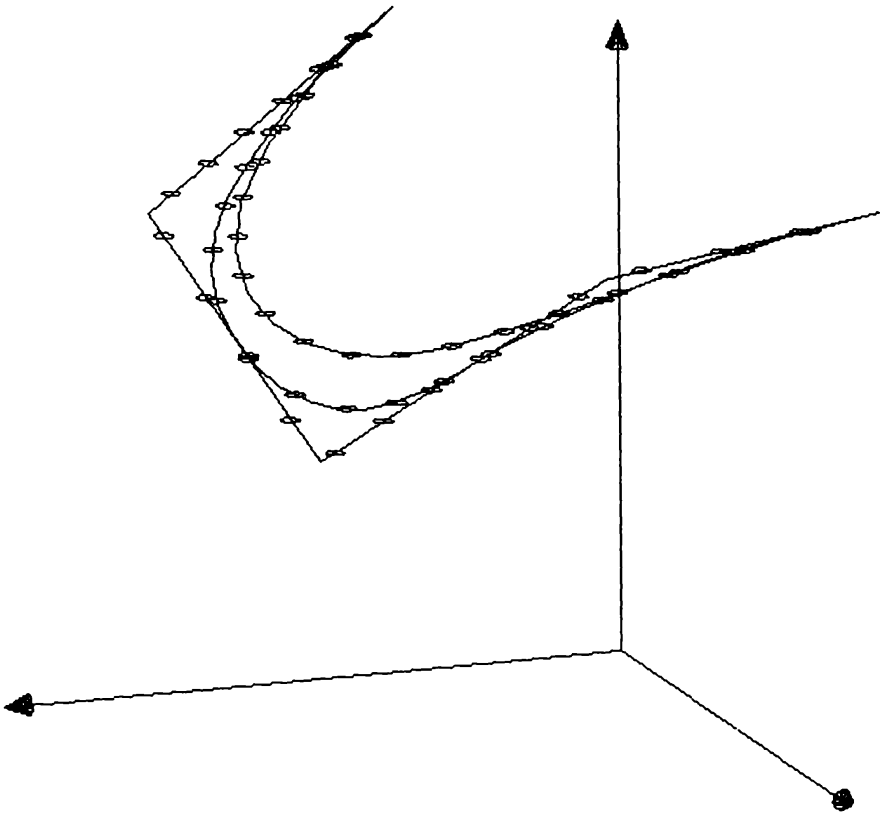


Fig. 6.4. Traiectoriile carteziene interpolate

BUPT

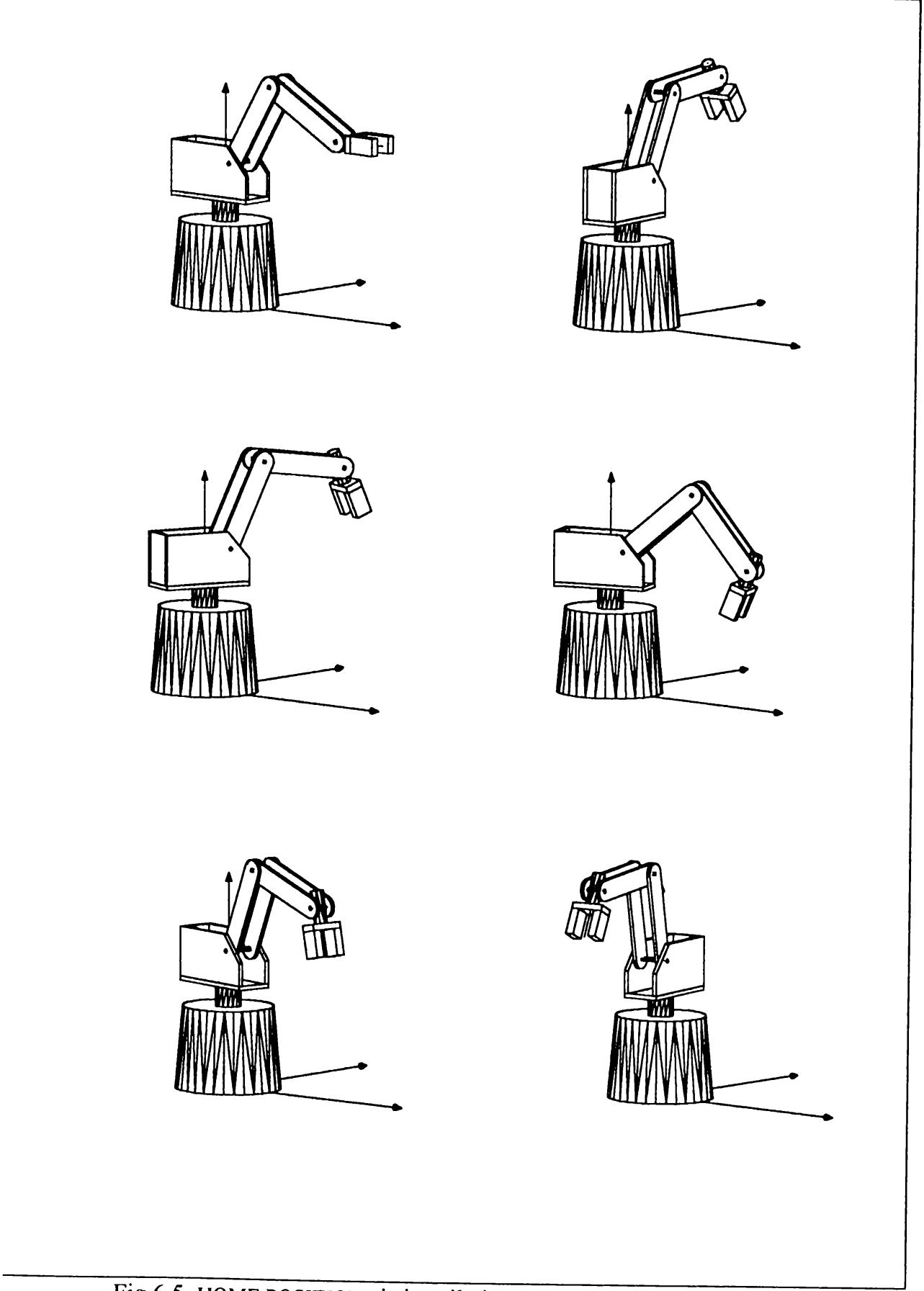


Fig.6.5. HOME POSITION şi situările impuse în spațiul operațional

BUPT

Coordonatele carteziene rezultate în urma interpolărilor şi transformările de coordonate exprimate în impulsuri (furnizate de programul robotului) şi în grade (calculate în urma determinărilor efectuate), în ambele cazuri amintite, sunt prezentate tabelar, astfel:

- Tabel 6.2: Coordonatele carteziene pentru interpolarea liniară;
- Tabel 6.3: Transformările de coordonate pentru interpolarea liniară, cu orientarea constantă a efectorului final;
- Tabel 6.4: Transformările de coordonate pentru interpolarea liniară, cu modificarea orientării efectorului final;
- Tabel 6.5: Coordonatele carteziene pentru interpolarea spline 3;
- Tabel 6.6: Transformările de coordonate pentru interpolarea spline 3, cu orientarea constantă a efectorului final;
- Tabel 6.7: Transformările de coordonate pentru interpolarea spline 3, cu modificarea orientării efectorului final;
- Tabel 6.8: Coordonatele carteziene pentru interpolarea spline 4;
- Tabel 6.9: Transformările de coordonate pentru interpolarea spline 4, cu orientarea constantă a efectorului final;
- Tabel 6.10: Transformările de coordonate pentru interpolarea spline 4, cu modificarea orientării efectorului final.

Din analiza acestor tabele se observă că modificarea orientării efectorului final nu influenţează mişcarea la AX 1 (această concluzie rezultă şi din expresiile proiecţiilor vectorului de poziţie **p** al originii sistemului de referinţă ataşat efectorului final pe axele sistemului de referinţă fix), dar pentru AX 2 şi AX 3 valorile impulsurilor, respectiv ale coordonatelor cuplelor sunt diferite pentru toate tipurile de traiectorii studiate. Concluziile rezultate din determinările efectuate subliniază influenţa orientării efectorului final asupra generării mişcării.

Considerând că un segment de traiectorie rezultat din divizarea acesteia în segmente congruente este echivalent cu o traiectorie parcursă într-o perioadă de eşantionare T, pentru a studia evoluţia coordonatelor cuplelor corespunzătoare mişcării impuse se reprezintă variaţia acestora în funcţie de numărul perioadelor de eşantionare.Variaţiile coordonatelor cuplelor, pentru cele trei traiectorii interpolate, în cele două cazuri studiate, pentru cele trei axe acţionate în ambele cazuri, sunt prezentate în fig.6.6-6.8.

Pentru verificarea preciziei de generare a mişcării se folosesc modelele geometrice al robotului (**MGD** şi **MGI**). Astfel:

- se pot impune valorile coordonatelor cuplelor în punctele de precizie şi se pot determina, din modelul geometric direct coordonatele carteziene ale punctului caracteristic care se compară cu coordonatele carteziene impuse;
- se pot impune coordonatele carteziene ale punctului caracteristic în poziţiile considerate şi pot determina coordonatele corespunzătoare ale cuplelor.

Analiza preciziei de generare a mişcării robotului a fost realizată în cadrul problemei directe. Coordonatele cuplelor din tabele 6.3-6.4, 6.6-6.7 şi 6.9-6.10 au fost introduse în modelul geometric direct şi, pe baza acestuia s-au calculat coordonatele carteziene ale punctului caracteristic, care au fost comparate cu cele din tabelele 6.2, 6.5 şi 6.8. Rezultatele sunt prezentate în fig.6.9-6.11.

BUPT

Tabel 6.2. Interpolare liniară. Punctele nodale în coordonate carteziene

POZ.	X [mm]	Y [mm]	Z [mm]	PITCH [°]	ROLL [°]
1	-50	430	450	-63.56	0
2	150	370	400	-30.00	45.00
3	280	100	200	0.00	75.00
4	350	-80	360	0.00	0.00
5	50	-350	430	60.00	0.00
6	7.54	412.74	435.62	-55.17	11.25
7	65.08	395.48	421.23	-46.78	22.5
8	122.62	378.21	406.85	-38.39	33.75
9	161.68	345.74	382.03	-25.71	49.28
10	183.97	299.45	347.74	-21.43	53.57
11	206.26	253.15	313.45	-17.14	57.85
12	228.55	206.86	279.16	-12.86	62.14
13	250.84	160.57	244.86	-8.57	66.42
14	273.13	114.27	210.57	-4.28	70.71
15	291.92	69.34	227.26	0.00	60.00
16	309.17	25.00	266.66	0.00	45.00
17	326.41	-19.33	306.07	0.00	30.00
18	343.65	-63.66	345.48	0.00	15.00
19	321.43	-105.71	366.67	8.57	0.00
20	276.19	-146.43	377.22	17.14	0.00
21	230.95	-187.14	387.78	25.71	0.00
22	185.72	-227.86	398.33	34.28	0.00
23	140.48	-268.57	408.89	42.85	0.00
24	95.24	-309.29	419.44	51.42	0.00

Tabel 6.3. Interpolare liniară. Transformări de coordonate

Poziţia	AX 1		AX 2		AX 3		AX 4	AX 5
	impuls	grad	impuls	grad	impuls	grad		
1	1028	114.22	591	71.20	-83	-10	0	0
2	722	80.22	483	58.19	-252	-30.36	0	0
3	209	23.22	552	66.50	-773	-93.13	0	0
4	-137	-15.22	412	49.63	-370	-44.57	0	0
5	-871	-96.78	374	45.06	-223	-26.86	0	0
6	946	105.11	520	62.65	-157	-18.91	0	0
7	858	95.33	486	58.55	-205	-24.69	0	0
8	766	85.11	478	57.59	-239	-28.79	0	0
9	691	76.78	449	54.09	-307	-36.98	0	0
10	621	69	406	48.91	-401	-48.31	0	0
11	540	60	391	47.10	-491	-58.15	0	0
12	448	49.78	407	49.03	-581	-70	0	0
13	347	38.56	453	54.57	-671	-80.84	0	0
14	241	26.78	526	63.37	-751	-90.48	0	0
15	142	15.78	494	59.51	-709	-85.42	0	0
16	49	5.44	432	52.04	-608	-73.25	0	0
17	-36	-4	401	48.31	-507	-61.08	0	0
18	-111	-12.33	404	48.67	-407	-49.03	0	0
19	-193	-21.44	364	43.85	-368	-44.33	0	0
20	-297	-33	302	36.38	-353	-42.53	0	0
21	-415	-46.11	263	31.68	-331	-39.87	0	0
22	-540	-60	250	30.12	-307	-36.98	0	0
23	-663	-73.67	267	32.16	-284	-34.21	0	0
24	-775	-86.11	309	37.22	-258	-31.08	0	0

BUPT

Tabel 6.4. Interpolare liniară. Transformări de coordonate

Poz.	AX 1		AX 2		AX 3		AX 4		AX 5	
	impuls	grad	impuls	grad	impuls	grad	impuls	grad	impuls	grad
1	1028	114.22	591	71.20	-83	-10.00	0	0	0	0
2	722	80.22	383	46.14	-420	-50.60	70	35	94	47
3	209	23.22	895	107.83	-1154	-139.04	133	66.5	156	78
4	-137	-15.22	458	55.18	-727	-87.59	133	66.5	0	0
5	-871	-96.78	658	79.28	-743	-89.52	258	129	0	0
6	946	105.11	470	56.63	-204	-24.59	18	9	23	11.5
7	858	95.33	408	49.16	-287	-34.59	35	17.5	47	23.5
8	766	85.11	385	46.39	-362	-43.61	53	26.5	70	35
9	691	76.78	368	44.34	-493	-59.40	79	39.5	103	51.5
10	621	69.00	373	44.94	-620	-74.70	88	44	112	56
11	540	60.00	433	52.17	-765	-92.17	97	49.5	121	60.5
12	448	49.78	553	66.63	-924	-111.33	106	53	130	65
13	347	38.56	703	84.70	-1061	-127.83	115	57.5	138	69
14	241	26.78	839	101.08	-1137	-136.99	124	62	147	73.5
15	142	15.78	819	98.67	-1121	-135.06	133	66.5	125	62.5
16	49	5.44	697	83.98	-1034	-124.58	133	66.5	94	47
17	-36	-4.00	576	69.40	-910	-109.64	133	66.5	62	31
18	-111	-12.33	482	58.07	-775	-93.37	133	66.5	31	15.5
19	-193	-21.44	472	56.87	-788	-94.94	150	75	0	0
20	-297	-33.00	482	58.07	-846	-101.93	168	84	0	0
21	-415	-46.11	508	61.20	-886	-106.75	186	93	0	0
22	-540	-60.00	539	64.94	-897	-108.07	204	102	0	0
23	-663	-73.67	512	61.69	-853	-102.77	204	102	0	0
24	-775	-86.11	609	73.37	-820	-98.80	240	120	0	0

Tabel 6.5. Interpolare spline 3. Coordonatele carteziene

POZ.	X [mm]	Y [mm]	Z [mm]	PITCH [°]	ROLL [°]
1	-50	430	450	-63.56	0
2	-1.71	411.90	435.41	-55.17	11.25
3	44.21	390.26	418.70	-46.78	22.5
4	87.06	364.45	399.76	-38.39	33.75
5	126.10	334.24	378.94	-30	45
6	160.42	299.33	357.12	-25.71	49.28
7	190.63	260.60	335.75	-21.43	53.57
8	217.36	218.51	316.20	-17.14	57.85
9	241.32	173.35	300.26	-12.86	62.14
10	263.57	125.64	290.56	-8.57	66.42
11	283.83	76.18	291.14	-4.28	70.71
12	295.73	25.39	302.05	0	75
13	294.32	-25.05	318.88	0	60
14	280.43	-73.09	337.22	0	45
15	258.27	-118.45	354.90	0	30
16	229.21	-160.64	370.67	0	15
17	196.96	-200.90	384.81	0	0
18	162.47	-239.82	397.81	8.57	0
19	126.00	-277.29	409.31	17.14	0
20	88.00	-313.64	419.65	25.71	0
21	50.00	-350.00	430.00	30	0

BUPT

Tabel 6.6. Interpolare spline 3. Transformări de coordonate

Poziţia	AX 1		AX 2		AX 3		AX 4	AX 5
	impuls	grad	impuls	grad	impuls	grad		
1	1028	114.22	591	71.20	-83	-10	0	0
2	960	106.67	517	62.28	-159	-19.15	0	0
3	888	98.67	466	56.14	-218	-26.26	0	0
4	814	90.44	427	51.44	-275	-33.13	0	0
5	737	81.89	397	47.83	-331	-39.81	0	0
6	657	73	374	45.06	-388	-46.74	0	0
7	572	63.56	359	43.25	-443	-53.37	0	0
8	480	53.33	352	42.40	-494	-59.51	0	0
9	379	42.11	352	42.40	-537	-64.69	0	0
10	271	30.11	357	43.01	-563	-67.83	0	0
11	159	17.67	360	43.37	-561	-67.59	0	0
12	52	5.78	349	42.04	-533	-64.21	0	0
13	-51	-5.67	322	38.79	-493	-59.39	0	0
14	-155	-17.22	288	34.69	-450	-54.21	0	0
15	-262	-29.11	258	31.08	-408	-49.15	0	0
16	-372	-41.33	235	28.31	-370	-44.57	0	0
17	-484	-53.78	229	27.59	-337	-40.60	0	0
18	-594	-66	241	29.03	-308	-37.10	0	0
19	-697	-77.44	270	32.53	-283	-34.09	0	0
20	-790	-87.78	314	37.83	-257	-30.96	0	0
21	-871	-96.78	374	45.06	-223	-26.86	0	0

Tabel 6.7. Interpolare spline 3. Transformări de coordonate

Poz	AX 1		AX 2		AX 3		AX 4		AX 5	
	impuls	grad	impuls	grad	impuls	grad	impuls	grad	impuls	grad
1	1028	114.22	591	71.20	-83	-10.00	0	0	0	0
2	960	106.67	468	56.39	-205	-24.70	18	9	23	11.5
3	888	98.67	392	47.23	-298	-35.90	35	17.5	47	23.5
4	814	90.44	342	41.20	-388	-46.75	53	26.5	70	35
5	737	81.99	320	38.55	-486	-58.55	70	35	94	47
6	657	73	322	38.80	-573	-69.04	79	39.5	103	51.5
7	572	63.56	348	41.93	-670	-80.72	88	44	112	56
8	480	53.33	400	48.19	-776	-93.49	97	48.5	121	60.5
9	379	42.11	474	57.11	-880	-106.02	106	53	130	65
10	271	30.11	545	65.66	-958	-115.42	115	57.5	138	69
11	159	17.67	583	70.24	-987	-118.92	124	62	147	73.5
12	52	5.78	582	70.12	-982	-118.31	133	66.5	156	78
13	-51	-5.67	520	62.65	-934	-112.53	133	66.5	125	62.5
14	-155	-17.22	448	53.98	-880	-106.02	133	66.5	94	47
15	-262	-29.11	373	44.94	-819	-98.67	133	66.5	62	31
16	-372	-41.33	305	36.75	-758	-91.33	133	66.5	31	15.5
17	-484	-53.78	258	31.08	-699	-84.22	133	66.5	0	0
18	-594	-66	307	36.99	-731	-88.07	150	75	0	0
19	-697	-72.44	372	44.82	-754	-90.84	168	84	0	0
20	-790	-87.78	441	53.13	-756	-91.08	186	93	0	0
21	-871	-96.78	483	58.19	-717	-86.39	195	97.5	0	0

BUPT

Tabel 6.8. Interpolare spline 4. Coordonatele carteziene

POZ.	X [mm]	Y [mm]	Z [mm]	PITCH [°]	ROLL [°]
1	-50	430	450	-63.56	0
2	1.57	412.27	435.47	-55.17	11.25
3	50.63	390.52	418.10	-46.78	22.5
4	96.25	364.42	397.70	-38.39	33.75
5	137.37	333.86	374.20	-30	45
6	172.92	299	347.77	-25.71	49.28
7	200.99	259.85	318.48	-21.43	53.57
8	223.28	217.67	288.36	-17.14	57.85
9	244.63	172.43	262.36	-12.86	62.14
10	267.15	123.15	247.19	-8.57	66.42
11	289.63	71.88	248.04	-4.28	70.71
12	309.37	24.18	269.50	0	75
13	317.48	-19.94	302.77	0	60
14	301.93	-66.46	330.00	0	45
15	275.34	-111.25	351.34	0	30
16	242.99	-154.00	368.78	0	15
17	207.46	-195.17	383.74	0	0
18	170.04	-235.24	397.11	8.57	0
19	130.94	-274.18	408.96	17.14	0
20	90.70	-312.27	419.70	25.71	0
21	50.00	-350.00	430.00	30	0

Tabel 6.9. Interpolare spline 4. Transformări de coordonate

Poziţia	AX 1		AX 2		AX 3		AX 4	AX 5
	impuls	grad	impuls	grad	impuls	grad		
1	1028	114.22	591	71.20	-83	-10.00	0	0
2	955	106.11	518	62.40	-158	-19.03	0	0
3	879	97.67	468	56.38	-219	-26.38	0	0
4	800	88.89	432	52.04	-275	-33.49	0	0
5	719	79.89	408	49.15	-339	-40.84	0	0
6	638	70.89	393	47.35	-405	-48.79	0	0
7	556	61.78	389	46.86	-478	-57.59	0	0
8	471	52.33	398	47.95	-557	-67.10	0	0
9	374	41.56	421	50.72	-628	-75.66	0	0
10	263	29.22	443	53.37	-669	-80.60	0	0
11	148	16.44	448	53.97	-663	-78.88	0	0
12	47	5.22	427	51.44	-602	-72.53	0	0
13	-38	-4.22	389	46.86	-520	-62.65	0	0
14	-132	-14.67	337	40.60	-462	-55.66	0	0
15	-234	-26.00	289	34.82	-415	-50.00	0	0
16	-344	-38.22	254	30.60	-375	-45.18	0	0
17	-460	-51.11	237	28.55	-340	-40.96	0	0
18	-576	-64.00	243	29.27	-310	-37.35	0	0
19	-686	-76.22	268	32.29	-283	-34.10	0	0
20	-785	-87.22	312	37.59	-259	-30.96	0	0
21	-871	-96.78	374	45.06	-223	-26.87	0	0

BUPT

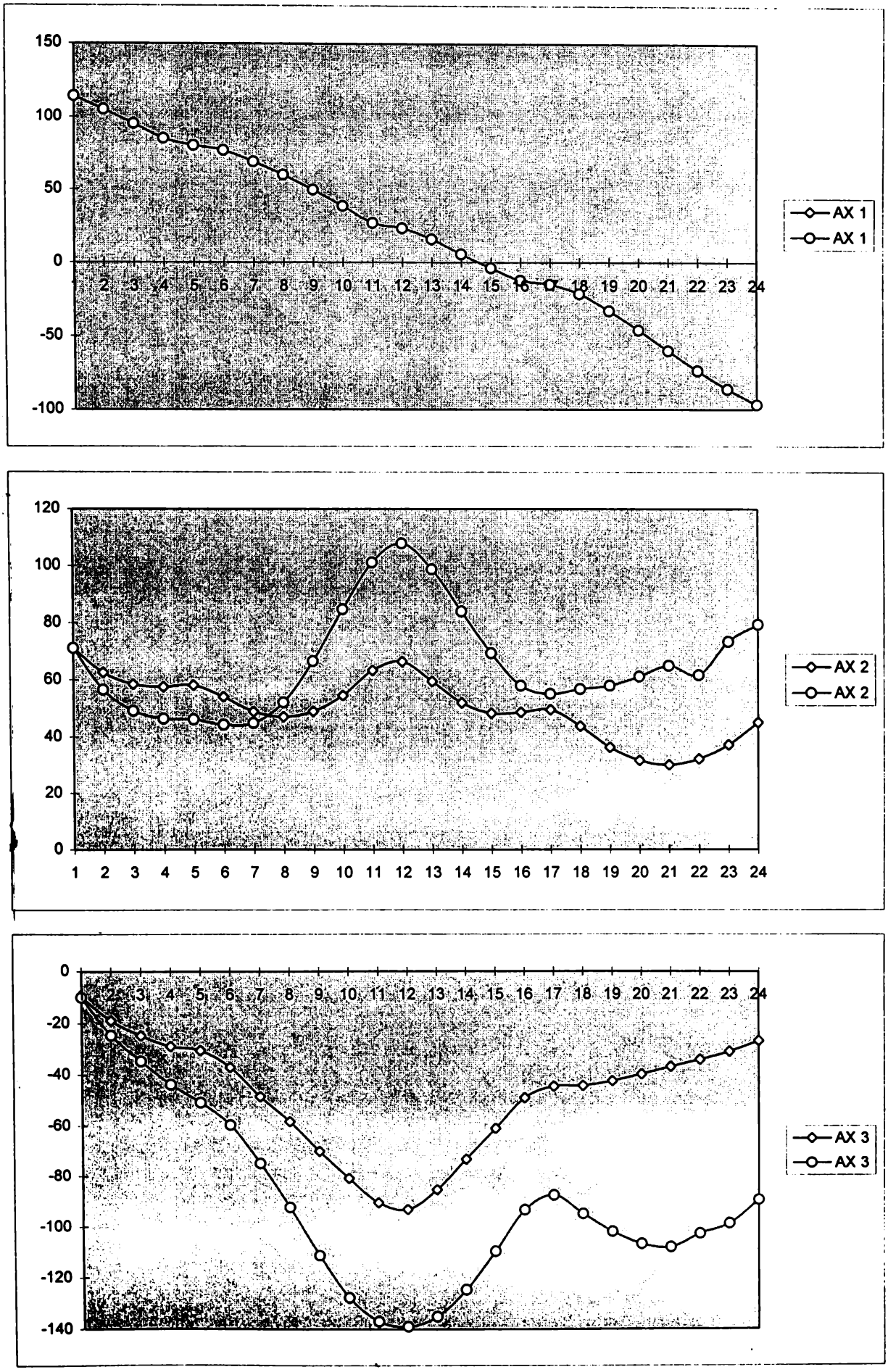


Fig.6.6. Interpolare liniară. Coordonatele cuplelor

BUPT

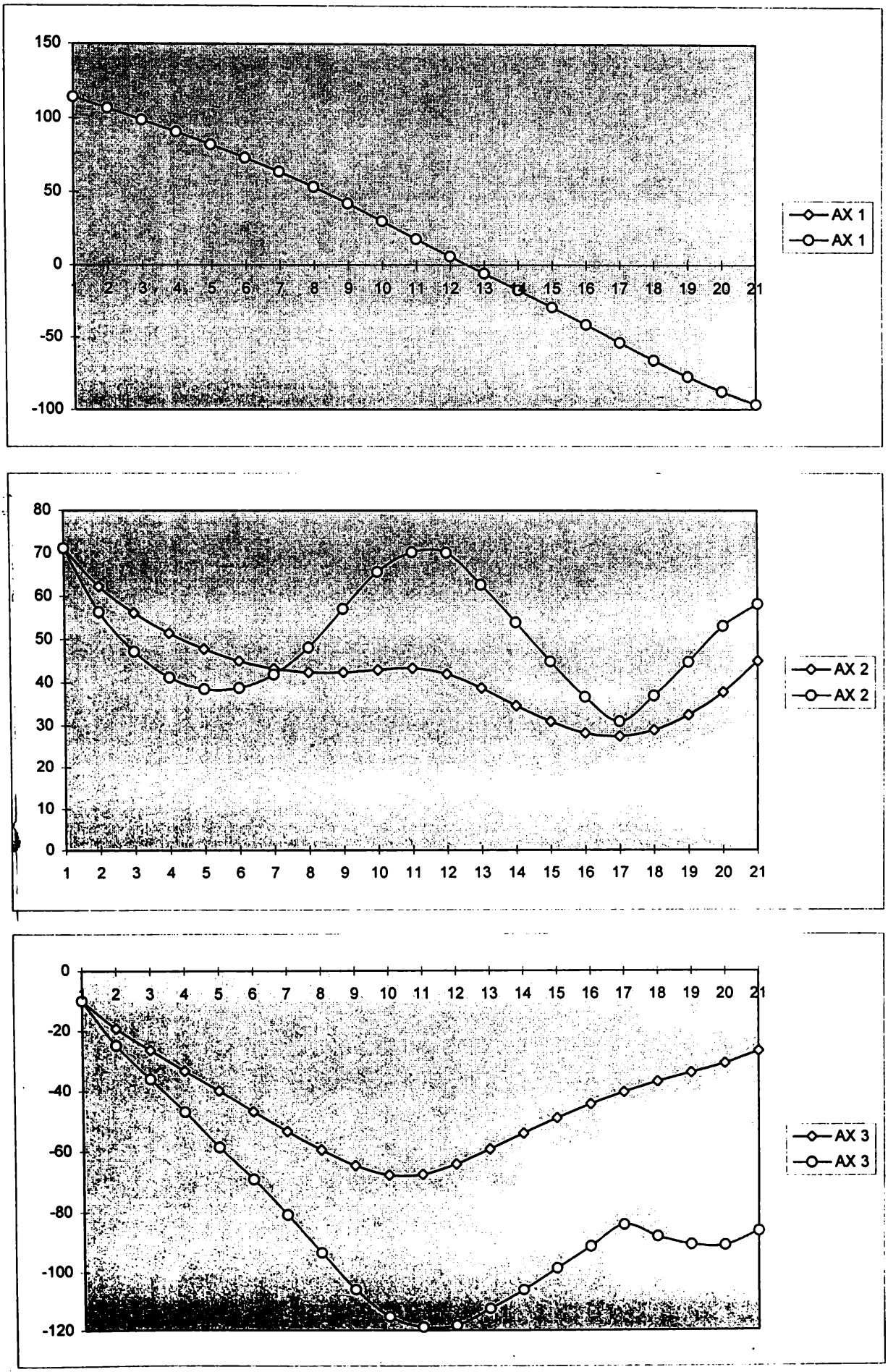
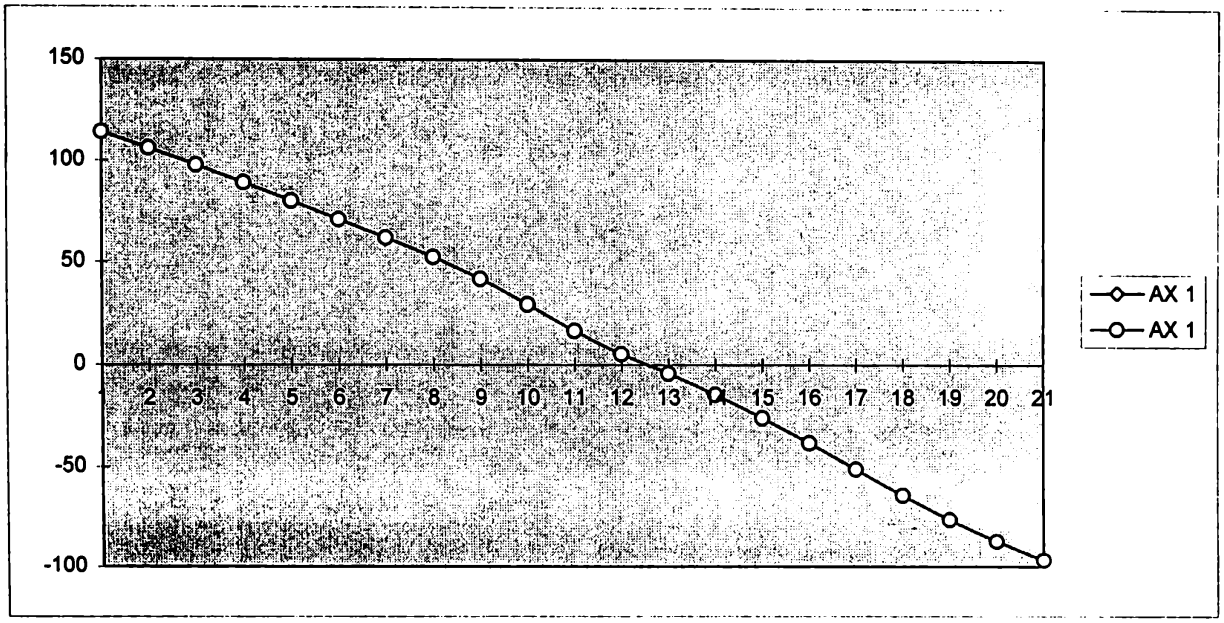
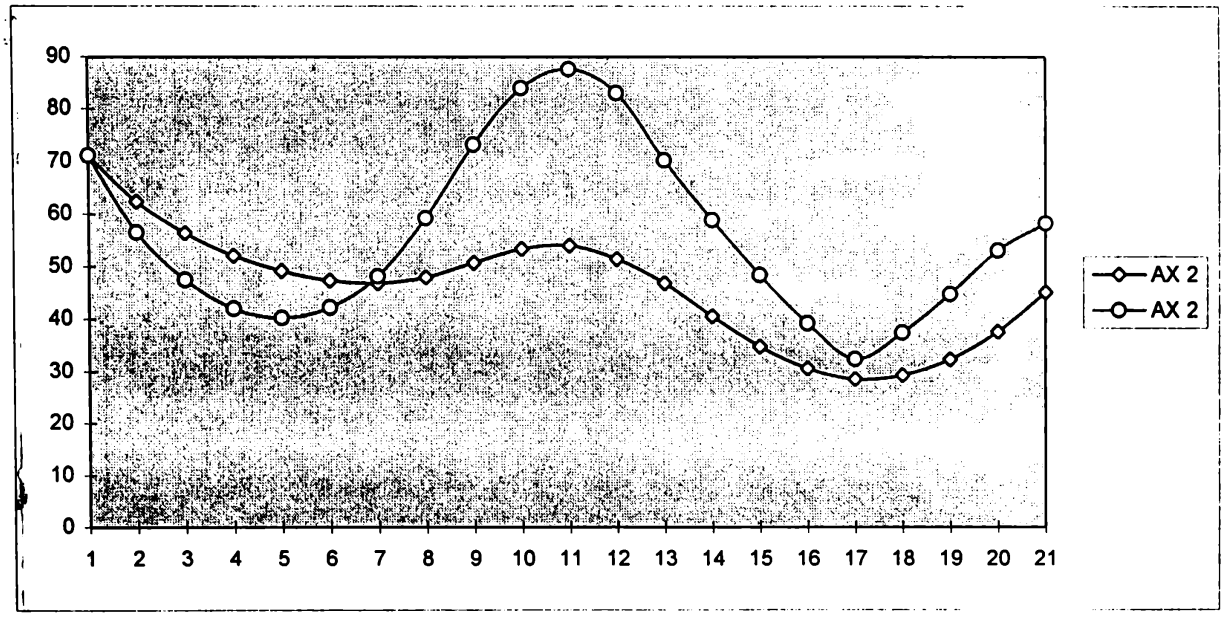


Fig.6.7. Interpolare spline 3. Coordonatele cuplelor

BUPT





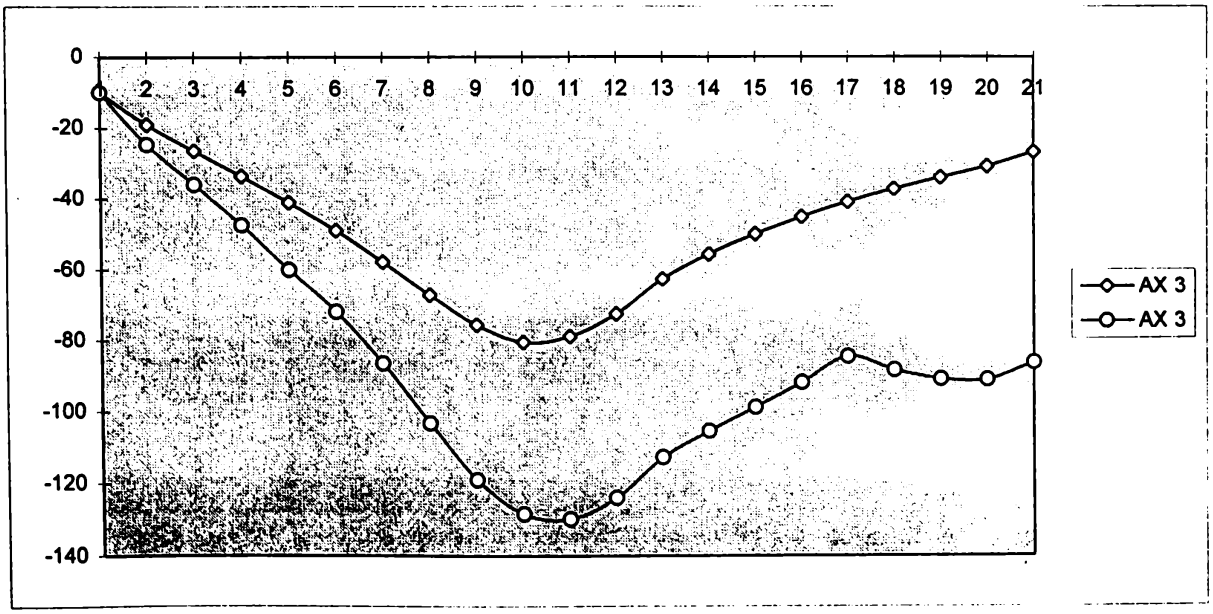


Fig.6.8. Interpolare spline 4. Coordonatele cuplelor

BUPT

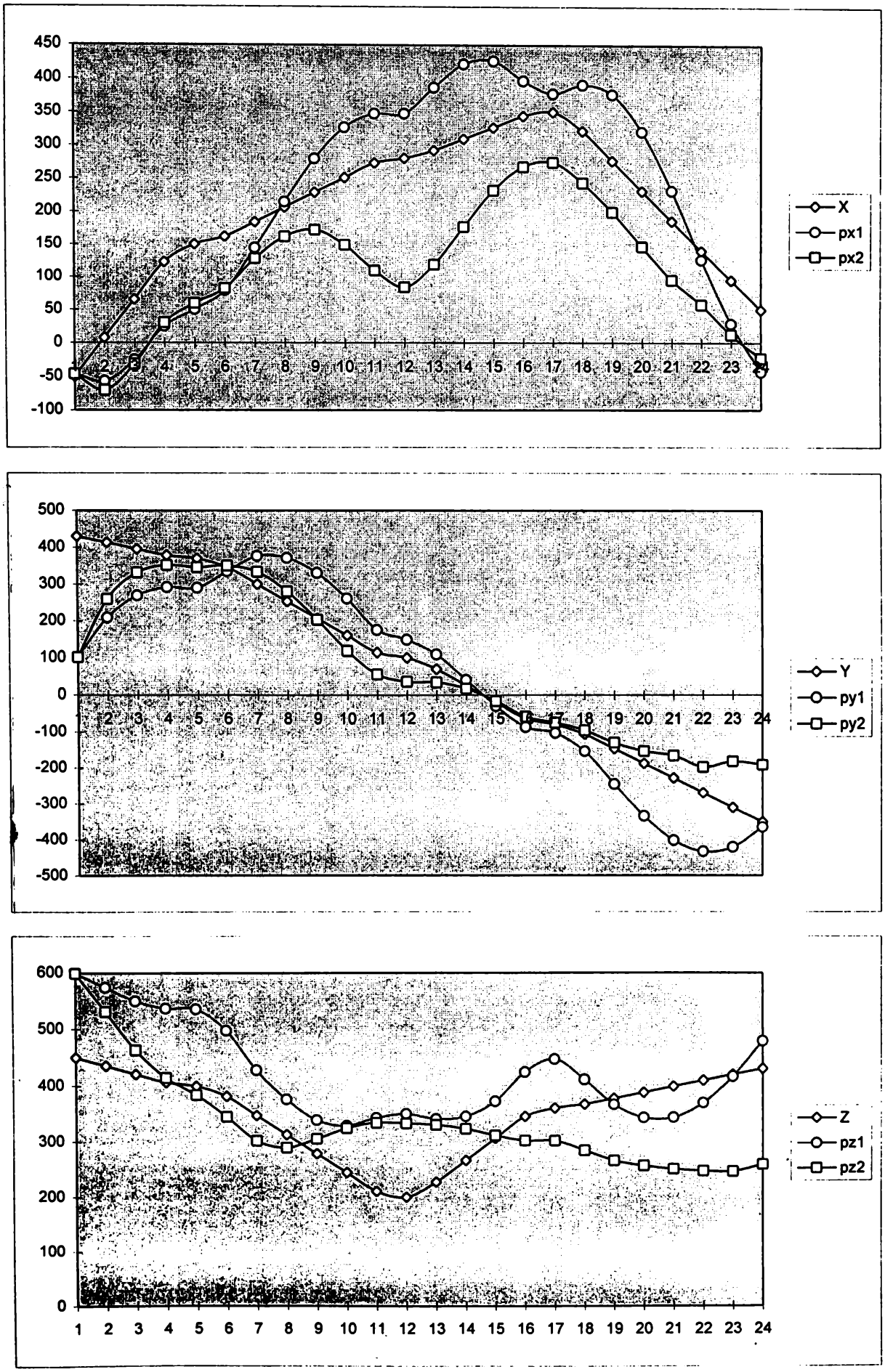


Fig.6.9. Interpolare liniară. Coordonatele carteziene rezultate din MGD

BUPT

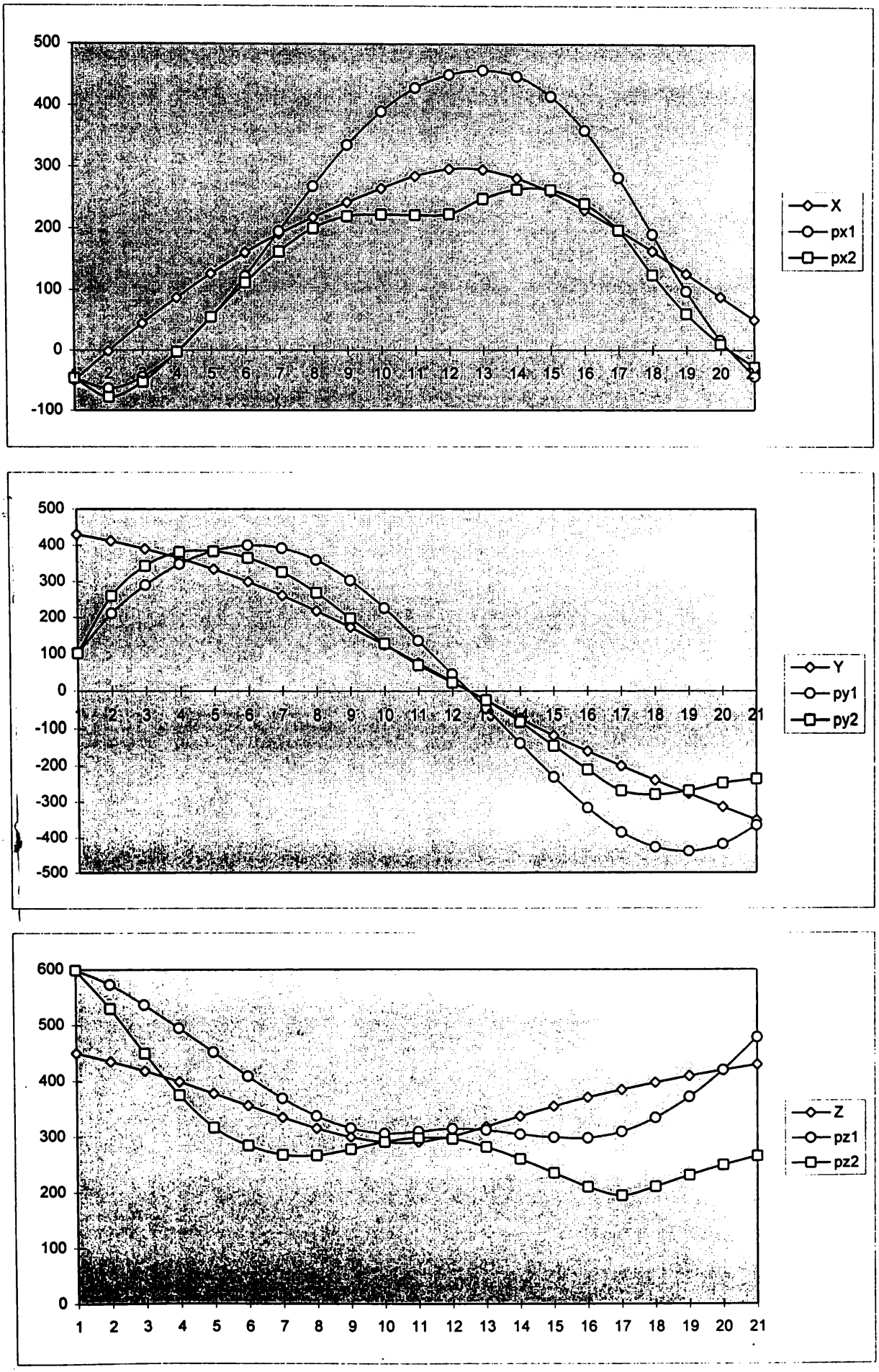
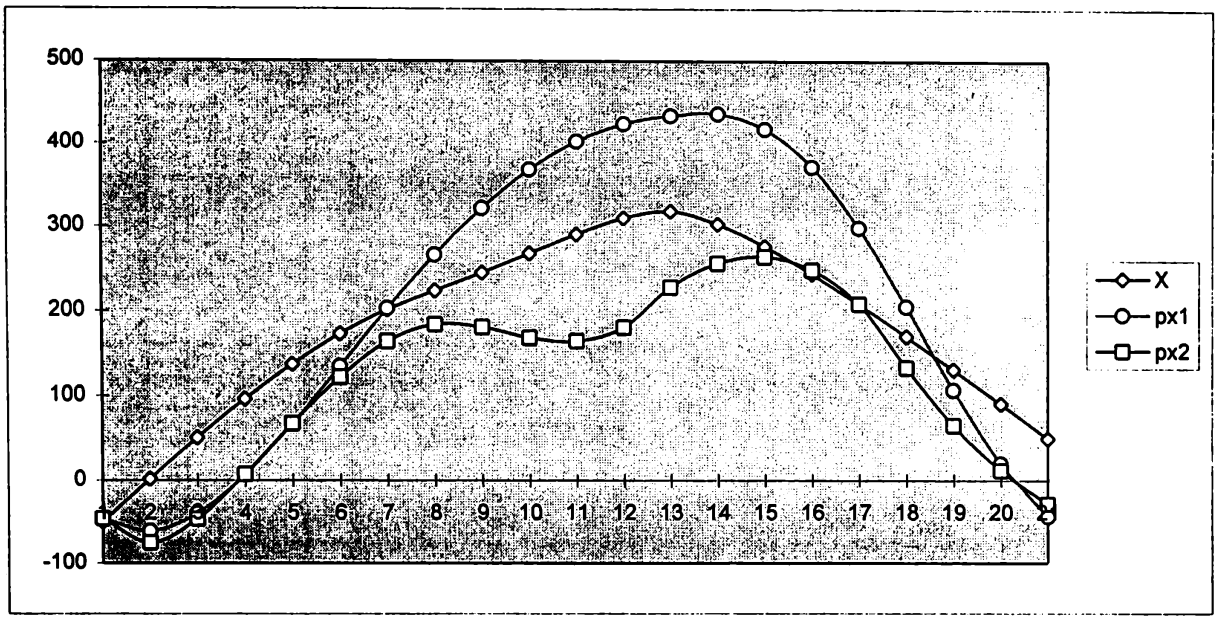
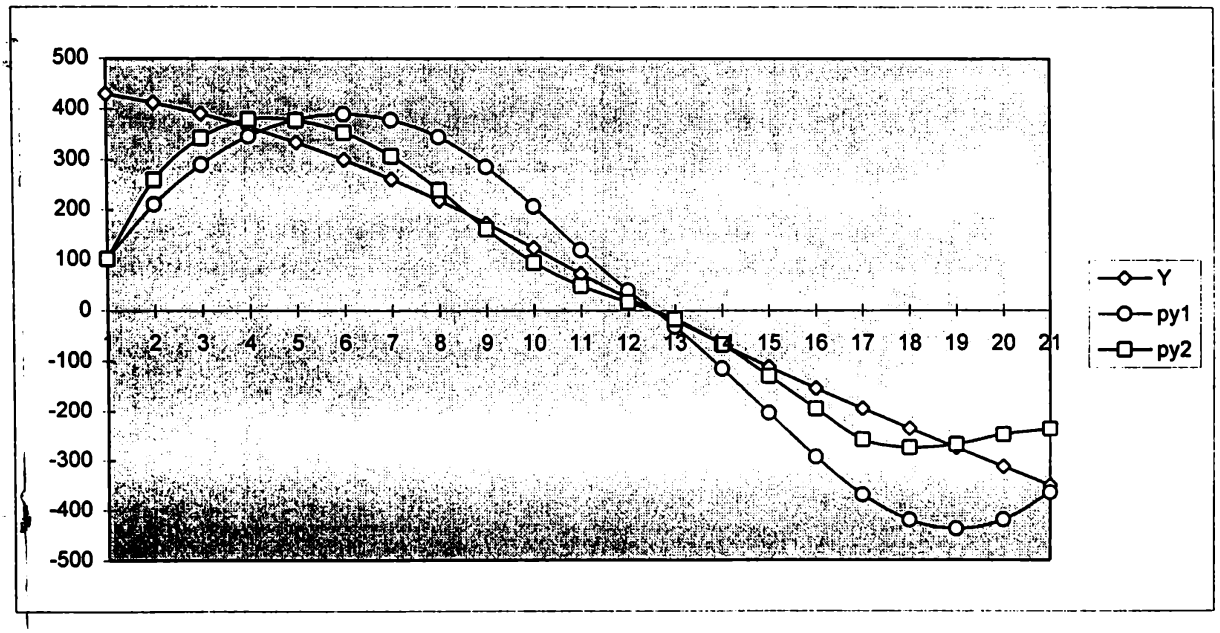


Fig.6.10. Interpolare spline 3. Coordonatele carteziene rezultate din MGD

BUPT





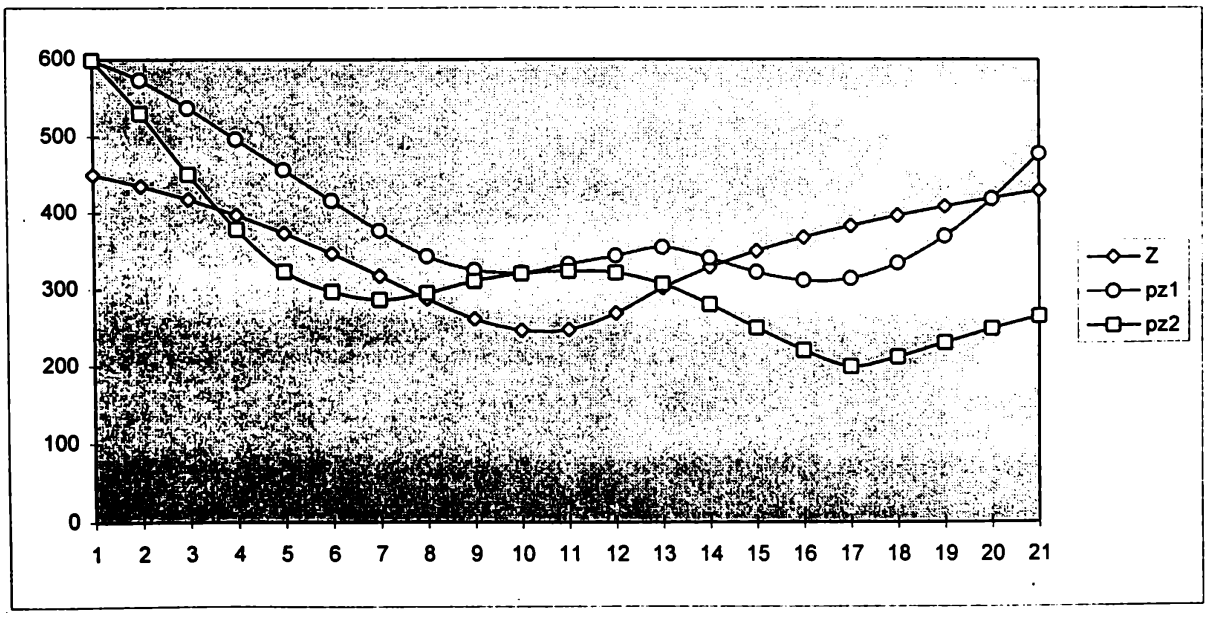


Fig.6.11. Interpolare spline 4. Coordonatele carteziene rezultate din **MGD**

BUPT

Tabel 6.10. Interpolare spline 4. Transformări de coordonate

Poz.	AX 1		AX 2		AX 3		AX 4		AX 5	
	impuls	grad	impuls	grad	impuls	grad	impuls	grad	impuls	grad
1	1028	114.22	591	71.20	-83	-10.00	0	0	0	0
2	955	106.11	469	56.51	-205	-24.70	18	9	23	11.5
3	879	97.67	394	47.47	-298	-35.90	35	17.5	47	23.5
4	800	88.89	349	42.05	-392	-47.23	53	26.5	70	35
5	719	79.89	334	40.24	-496	-59.76	70	35	94	47
6	638	70.89	351	42.29	-595	-71.69	79	39.5	103	51.5
7	556	61.78	400	48.19	-716	-86.27	88	44	112	56
8	471	52.33	492	59.28	-857	-103.25	97	48.5	121	60.5
9	374	41.56	608	73.25	-988	-119.04	106	53	130	65
10	263	29.22	697	83.98	-1066	-128.43	115	57.5	138	69
11	148	16.44	727	87.59	-1077	-129.76	124	62	147	73.5
12	47	5.22	689	83.01	-1028	-123.86	133	66.5	156	78
13	-38	-4.22	583	70.24	-935	-112.65	133	66.5	125	62.5
14	-132	-14.67	489	58.92	-875	-105.42	133	66.5	94	47
15	-234	-2.6	402	48.43	-820	-98.80	133	66.5	62	31
16	-344	-38.22	326	39.28	-762	-91.81	133	66.5	31	15.5
17	-460	-51.11	269	32.41	-703	-84.70	133	66.5	0	0
18	-576	-64	311	37.47	-734	-88.43	150	75	0	0
19	-686	-76.22	371	44.70	-755	-90.96	168	84	0	0
20	-785	-87.22	440	53.01	-756	-91.08	186	93	0	0
21	-871	-96.78	483	58.19	-717	-86.39	195	97.5	0	0

Precizia de parcurgere a traiectoriei este influențată și de vitezele dezvoltate la axele robotului, care pot fi impuse prin program (pot fi setate zece viteze).

O altă categorie de determinări au constat în conducerea punct cu punct pentru generarea unei traiectorii continue, în plan și determinarea abaterilor de la traiectoria prescrisă. Au fost astfel generate curbe uzuale în mișcarea roboților, respectiv cercuri și elipse (fig.8.12).

Fig.6.12. Generarea unei traiectorii circulare în planul xOy

BUPT

În tabelul 6.11 sunt prezentate coordonatele carteziene ale punctelor unui cerc, rezultate din divizarea acestuia în 36 de părți egale. Cercul este determinat de punctele: P_1(391.16, 115.46, 132.24), P_2(473.02, 46.70, 132.24) şi P_3(407.52, 16.05, 132.24) având centrul C(422.41, 69.55, 132.24) şi raza R=55.53 mm. În tabelul 6.12 sunt prezentate transformările de coordonate corespunzătoare datelor din tabelul 6.11.

Tabel 6.11. Coordonatele carteziene pentru CERC 1

POZ.	X [mm]	Y [mm]	Z [mm]	PITCH [°]	ROLL [°]
1	391.16	115.46	132.24	0	0
2	473.02	46.70	132.24	0	0
3	407.52	16.05	132.24	0	0
4	394.64	117.64	132.24	0	0
5	403.41	121.74	132.24	0	0
6	412.76	124.24	132.24	0	0
7	422.41	125.08	132.24	0	0
8	432.05	124.24	132.24	0	0
9	441.40	121.74	132.24	0	0
10	450.17	117.64	132.24	0	0
11	458.10	112.09	132.24	0	0
12	464.95	105.25	132.24	0	0
13	470.50	97.32	132.24	0	0
14	474.59	88.54	132.24	0	0
15	477.10	79.19	132.24	0	0
16	477.94	69.55	132.24	0	0
17	477.10	59.91	132.24	0	0
18	474.59	50.56	132.24	0	0
19	470.50	41.78	132.24	0	0
20	464.95	33.85	132.24	0	0
21	458.10	27.10	132.24	0	0
22	450.17	21.46	132.24	0	0
23	441.40	17.37	132.24	0	0
24	432.05	14.86	132.24	0	0
25	422.41	14.02	132.24	0	0
26	412.76	14.86	132.24	0	0
27	403.41	17.37	132.24	0	0
28	394.64	21.46	132.24	0	0
29	386.71	27.01	132.24	0	0
30	379.86	33.85	132.24	0	0
31	374.31	41.78	132.24	0	0
32	370.22	50.56	132.24	0	0
33	367.72	59.91	132.24	0	0
34	366.87	69.55	132.24	0	0
35	367.72	79.19	132.24	0	0
36	370.22	88.54	132.24	0	0
37	374.31	97.32	132.24	0	0
38	379.86	105.25	132.24	0	0
39	386.71	112.09	132.24	0	0

BUPT

Tabel 6.12. Transformări de coordonate pentru CERC 1

Poziţia	AX 1		AX 2		AX 3		AX 4		AX 5
	impuls	grad	impuls	grad	impuls	grad	impuls	grad	
1	175	19.44	1018	122.65	-923	-111.20	133	66.5	0
2	60	6.67	1060	127.71	-767	-92.41	133	66.5	0
3	24	2.67	1018	122.65	-924	-111.33	133	66.5	0
4	176	19.56	1020	122.89	-915	-110.24	133	66.5	0
5	178	19.78	1023	123.25	-893	-107.59	133	66.5	0
6	178	19.78	1026	123.61	-872	-105.06	133	66.5	0
7	175	19.44	1032	124.34	-850	-102.41	133	66.5	0
8	170	18.89	1038	125.06	-829	-99.88	133	66.5	0
9	164	18.22	1044	125.78	-809	-97.47	133	66.5	0
10	155	17.22	1050	126.51	-791	-95.30	133	66.5	0
11	146	16.22	1057	127.35	-776	-93.49	133	66.5	0
12	135	15.00	1062	127.95	-763	-91.93	133	66.5	0
13	124	13.78	1066	128.43	-753	-90.72	133	66.5	0
14	112	12.44	1069	128.80	-747	-90.00	133	66.5	0
15	100	11.11	1070	128.92	-745	-89.76	133	66.5	0
16	88	9.78	1069	128.80	-747	-90.00	133	66.5	0
17	76	8.44	1067	128.55	-752	-90.60	133	66.5	0
18	64	7.11	1063	128.07	-762	-91.81	133	66.5	0
19	54	6.00	1057	127.35	-774	-93.25	133	66.5	0
20	44	4.89	1051	126.63	-789	-95.06	133	66.5	0
21	35	3.89	1045	125.90	-807	-97.23	133	66.5	0
22	29	3.22	1038	125.06	-826	-99.52	133	66.5	0
23	23	2.56	1032	124.34	-847	-102.05	133	66.5	0
24	20	2.22	1027	123.74	-869	-104.70	133	66.5	0
25	20	2.22	1023	123.25	-890	-107.23	133	66.5	0
26	21	2.33	1020	122.89	-912	-109.88	133	66.5	0
27	26	2.89	1018	122.65	-932	-112.29	133	66.5	0
28	33	3.67	1016	122.41	-951	-114.59	133	66.5	0
29	42	4.67	1015	122.29	-968	-116.63	133	66.5	0
30	54	6.00	1015	122.29	-982	-118.31	133	66.5	0
31	67	7.44	1015	122.29	-992	-119.52	133	66.5	0
32	82	9.11	1015	122.29	-999	-120.36	133	66.5	0
33	98	10.89	1016	122.41	-1001	-120.60	133	66.5	0
34	114	12.67	1016	122.41	-1000	-120.48	133	66.5	0
35	129	14.33	1015	122.29	-994	-119.76	133	66.5	0
36	143	15.89	1015	122.29	-984	-118.55	133	66.5	0
37	155	17.22	1015	122.29	-970	-116.87	133	66.5	0
38	164	18.22	1016	122.41	-954	-114.94	133	66.5	0
39	172	19.11	1017	122.53	-935	-112.65	133	66.5	0

Divizând cercul în 36 de părţi egale, punctele obţinute sunt destul de apropiate. Parcurgerea unui segment de traiectorie se realizează într-un interval de timp scurt, insuficient pentru a asigura o deplasare cu viteză constantă. Mişcarea are loc cu profil triunghiular de viteză, conform celor prezentate în capitolul 3.1.2. Datorită succesiunii fazelor de accelerare/frânare, în fiecare punct de pe cerc se observă profilul corespunzător legii de mişcare (fig.3.4). Cercul astfel obţinut nu este o curbă continuă (fig.6.13).

În ceea ce priveşte precizia de redare a punctelor interpolate, din analiza "punctelor" trasate, se observă că aceasta diferă pe circumferinţa cercului. Dacă se consideră cele patru cadrane, precizia este chiar foarte bună în cadranele 2 şi 4 şi necorespunzătoare în cadranele 1 şi 3. Ceea ce este caracteristic cadranelor 1 şi 3 este faptul că, între două puncte succesive, $\Delta y > \Delta x$.

BUPT

Determinările experimentale au fost realizate pentru acelaşi set de puncte de interpolare, în condiţii diferite: viteze diferite şi plane paralele diferite. Rezultatele obţinute nu diferă semnificativ în situaţiile studiate. Pentru generarea unei traiectorii circulare, corecţia se poate realiza prin rotirea mesei de lucru; se pot trasa arcele corespunzătoare cadranelor 1 şi 3, poziţionate în 2 şi 4, cu o precizie corespunzătoare.

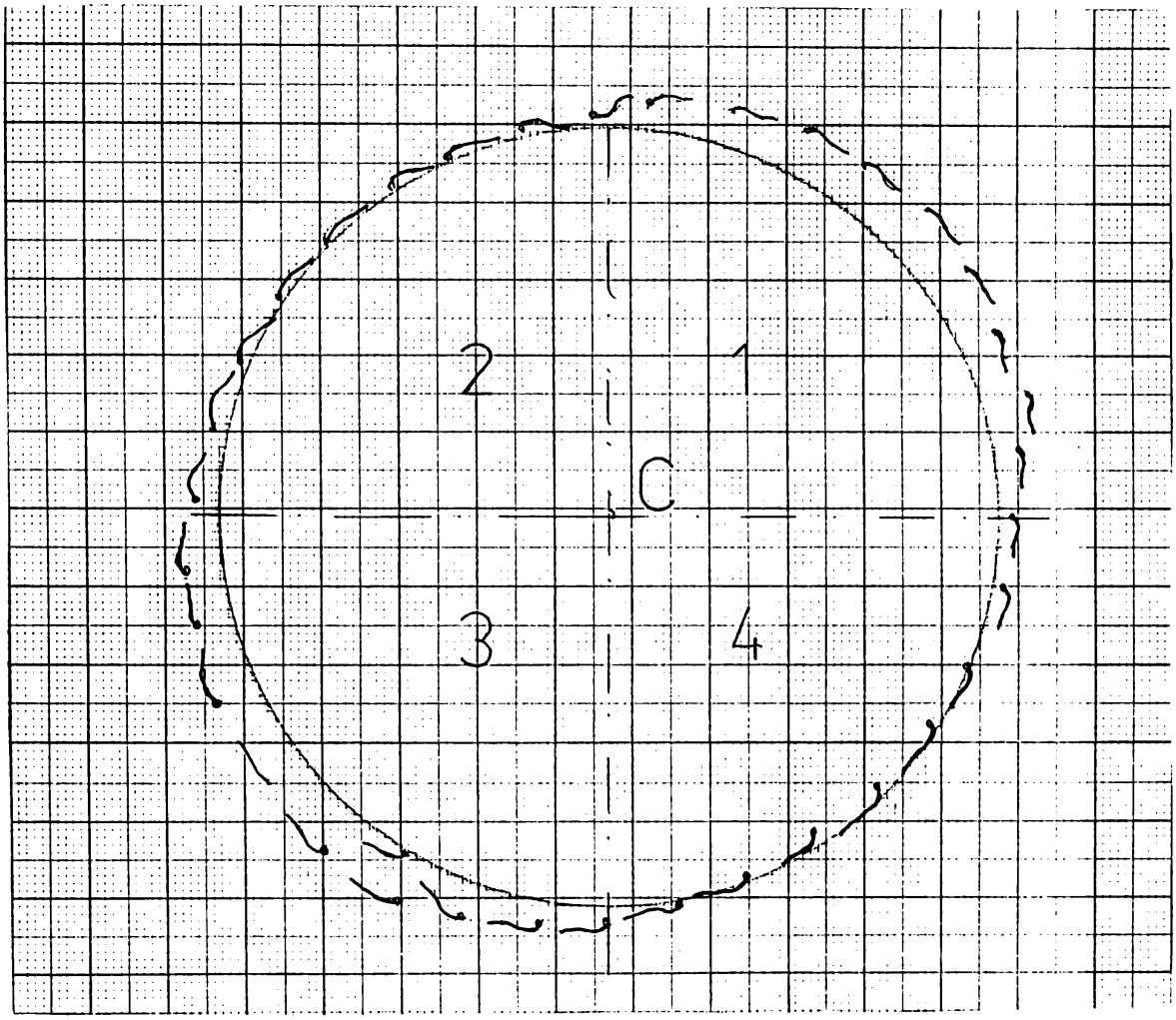


Fig.6.13. Generarea unei traiectorii circulare, în planul xOy

6.4. CONCLUZII

Robotul SCORBOT-ER III este un robot didactic, dintr-o generaţie mai veche, fără un generator propriu de traiectorii.

Prin studiul experimental efectuat s-a urmărit să se genereze o mişcare pe o traiectorie continuă, cu un robot cu comandă punct cu punct.

Realizarea celor două seturi de determinări (cu păstrarea constantă a orientării şi cu modificarea orientării) evidenţiază rolul orientării efectorului final în generarea traiectoriei unui robot şi furnizează informaţii despre modul de solicitare a cuplelor cinematice în cele două cazuri.

Pentru determinarea performanţelor robotului a fost folosit modelul geometric direct, dar poate fi folosit şi modelul invers. Datorită dificultăţilor de rezolvare a acestuia, precizia obţinută în acest caz este şi mai slabă.

BUPT

Concluziile referitoare la precizia de generare a mişcării nu sunt satisfăcătoare. Erorile de poziție rezultate se datorează erorilor de execuție, de comandă, etc.

Ținând cont de faptul că robotul studiat este unul didactic şi nu un robot de precize, performanțele în ceea ce priveşte acuratețea realizării unei traiectorii impuse sunt modeste. Scopul determinărilor întreprinse a constat în verificarea noțiunilor teoretice, în stabilirea unor metode de lucru şi în verificarea posibilității de a implementa diferite metode de generare.

Pentru a putea realiza o traiectorie continuă cu un robot cu comandă punct cu punct, trebuie acceptat un compromis în ceea ce priveşte numărul punctelor care definesc traiectoria: un număr mai mic de puncte permite robotului să se deplaseze cu viteză constantă între acestea, punctele fiind suficient de depărtate, dar deplasarea nu urmăreşte conturul curbei prescrise; un număr mai mare de puncte definesc mai bine traiectoria, dar, între acestea, mişcarea are loc ca o succesiune de accelerări/frânări, rezultând mici deviații de la traiectoria prescrisă, a căror formă caracterizează mişcarea cu profil triunghiular de viteză.

BUPT

7. GENERAREA ŞI PLANIFICAREA MIŞCĂRII ÎNTR-UN SPAŢIU CU OBSTACOLE

7.1. INTRODUCERE

Problema planificării traiectoriilor în prezența obstacolelor poate fi enunțată astfel: *fiind dat un spațiu de lucru populat cu obstacole cunoscute prin frontierele lor, respectiv cu obiecte mobile, trebuie să se determine o traiectorie fără coliziuni cu obstacolele, aducând obiectele mobile din configurația inițială în cea finală.*

Problema poate fi abordată în două moduri (global sau local), de unde rezultă două tipuri de metode de planificare: globale şi locale.

Aplicarea unei metode globale necesită cunoaşterea completă "a priori" a spațiului de lucru, modelarea corespunzătoare a spațiului liber, cercetarea tuturor traiectoriilor posibile şi selectarea unei anumite traiectorii corespunzătoare unui criteriu de cost minim. O astfel de metodă garantează existența sau inexistența unei soluții. De asemenea, metodele globale de planificare se pot adapta uşor la programarea off-line.

Aplicarea unei metode locale necesită cunoaşterea parțială a spațiului de lucru. O astfel de metodă nu garantează atingerea configurației finale, dar prezintă avantajul unei bune adaptări în timp real.

În ambele cazuri, rezolvarea problemei de planificare presupune rezolvarea unor probleme de natură geometrică (geometrie pură) sau de geometrie combinată cu cinematică şi/sau dinamică. În astfel de situații se utilizează cu precădere rezultatele geometriei algoritmice.

Pentru un robot la un post de lucru fix, mobilele pentru care trebuie să se determine traiectoriile fără coliziuni sunt constrânse între ele de anumite legături. Astfel, o problemă de planificare este deci "a priori" complexă. Totuşi, prin transformări adecvate, operate asupra obstacolelor, o problemă de planificare poate fi redusă la o problemă de "navigare" a unui robot punctiform, considerat ca un mobil liber care evoluează printre obstacolele transformate.

Astfel, planificarea traiectoriilor unui robot comportă, în general, două etape:

- modelarea spațiului de lucru, transformarea obstacolelor (determinarea C-obstacolelor) astfel încât robotul poate fi considerat ca un punct material;
- cercetarea unei traiectorii, fără coliziune, pentru acest punct.

În cadrul unui sistem CAD/CAM dotat cu un algoritm de cercetare a spațiului liber se poate aplica planificarea traiectoriilor fără coliziuni, în mod interactiv cu operatorul, prin vizualizarea acestui spațiu.

În general, aplicarea unei metode de planificare trebuie să satisfacă anumite restricții, ca de exemplu: drumul cel mai sigur, drumul cel mai scurt, etc.

Metoda hărții drumurilor se bazează pe ideea captării conexității spațiului liber al robotului într-o rețea de curbe unidimensionale, numită ***harta drumurilor*** *HD*. Odată construită, aceasta este utilizată ca o bancă de traiectorii standard dintre care se alege aceea ce uneşte configurația inițială cu cea finală. Utilizând diferite principii constructive se obțin

BUPT

diferite tipuri de hărți ca: ***graful vizibilității, diagrama Voronoï, rețeaua drumurilor libere, siluetele,*** etc.

Metoda grafului vizibilității se aplică cu precădere la spații ale configurațiilor bidimensionale populate cu C-obstacole poligonale. Harta drumurilor este un graf numit **graful vizibilității**, construit prin conectarea tuturor perechilor de vârfuri din frontiera lui SC_{liber} prin segmente de dreaptă, dacă aceste segmente nu traversează interiorul unui C-obstacol.

O altă metodă, ce constă din construirea unei hărți a drumurilor prin definirea unei aplicații continue (numită retractare) a lui SC_{liber} pe *HD* se numeşte ***metoda retractării***. Când spațiul configurațiilor este bidimensional, metoda conduce la retractarea lui SC_{liber} la diagrama sa Voronoï. Aceasta este definită ca submulțime unidimensională a lui SC_{liber} ce maximizează intervalele spațiale libere dintre robot şi obstacole.

Metoda drumului liber este înrudită cu metoda retractării, punctul de referință al robotului fiind constrâns să rămână pe o rețea de segmente de dreaptă reminiscente dintr-o diagramă Voronoï.

Metoda siluetelor constă din schițarea unei siluete a spațiului liber al robotului când acesta este privit dintr-un punct situat la infinit şi adăugând câteva segmente de curbă obținute prin unirea unor puncte critice ale siluetei cu alte porțiuni de curbă ale acesteia. Silueta şi curbele de legătură formează harta drumurilor *HD*, care va fi cercetată în vederea găsirii unei traiectorii.

Toate metodele enumerate sunt deosebit de eficace în probleme de planificare în spații ale configurațiilor de dimensiuni mici (2 sau 3).

7.2. MODELAREA SPAȚIULUI DE LUCRU PRIN METODA GRILEI NEOMOGENE

Stabilirea unei hărți de navigare în spațiul de lucru al robotului se poate realiza fie independent de orice acțiune a acestuia, prin memorarea configurației spațiului de lucru (de obicei este memorată configurația inițială), fie în timpul executării sarcinii, în interacțiune cu sistemul senzorial.

Întregul proces de planificare a mişcării se bazează pe modelarea spațiului de lucru. Modelul mediului presupune împărțirea spațiului de lucru în celule independente, reprezentând zone accesibile sau zone interzise robotului, ținând cont de geometria şi dimensiunile obstacolelor fizice şi ale robotului.

Metoda grilei constă în împărțirea spațiului de lucru în celule dreptunghiulare în $\mathbf{R}^2$ sau paralelipipedice în $\mathbf{R}^3$, a căror apartenență la o zonă permisă sau interzisă este booleană. Dimensiunile celulelor pot fi egale (grila omogenă) sau neegale (grila neomogenă). Precizia modelului depine de dimensiunile celulelor.

Metoda grilei neomogene presupune toate obstacolele de formă dreptunghiulară. Pentru un spațiu bidimensional, populat cu n obstacole, grila este formată din $(2n+1)^{2n+1}$ dreptunghiuri. Fiecare regiune este reprezentată prin două liste binare de câte 2n+1 biți, dintre care una reprezintă poziția relativă x, iar cealaltă poziția relativă y. Un bit are valoarea "1" dacă celula este liberă şi "0" în caz contrar, pentru prima listă, iar pentru a doua, există un singur "1" pentru poziția verticală a celulei [D41].

BUPT

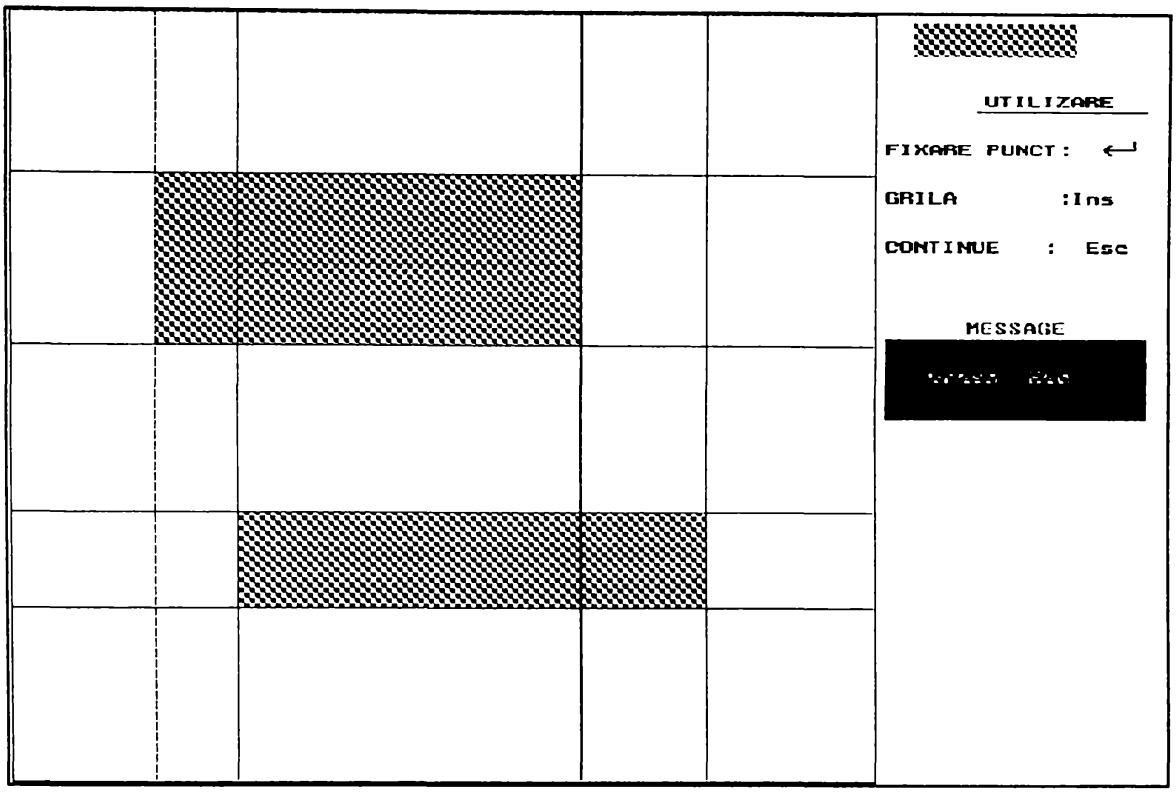


Fig.7.1. Divizarea spaţiului de lucru

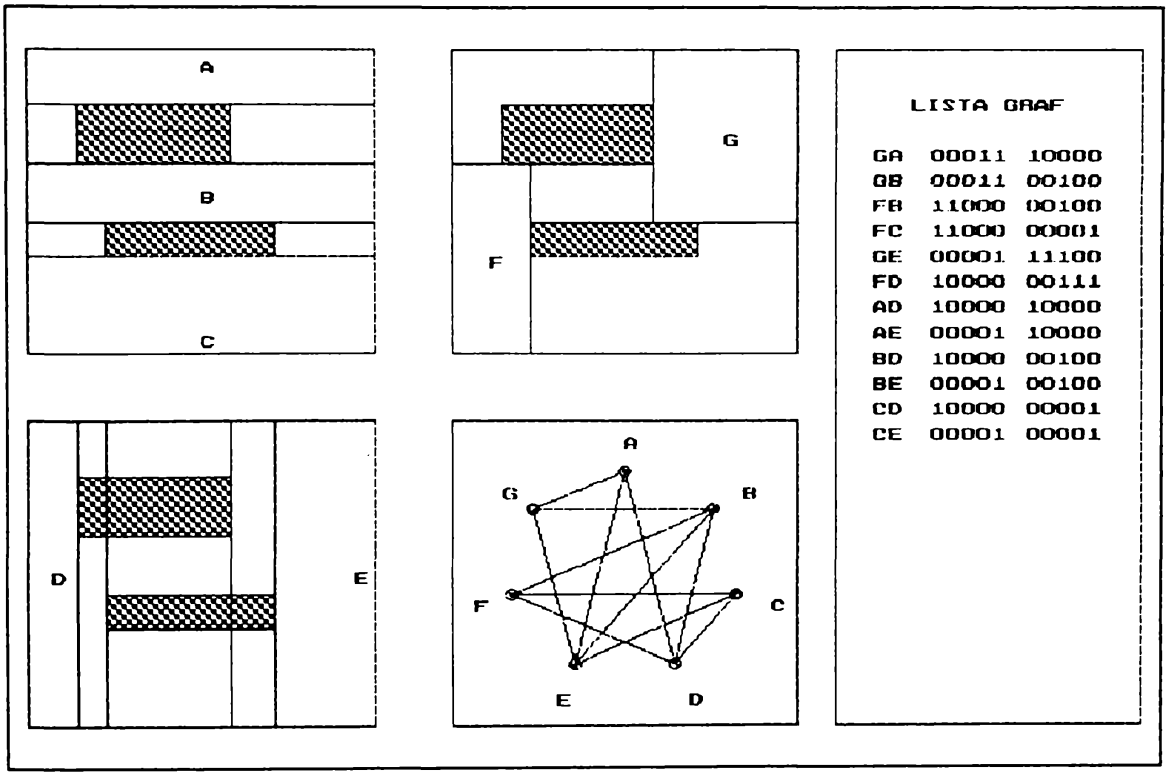


Fig.7.2. Etapele algoritmului de modelare

BUPT

Pentru determinarea zonelor libere, listele binare sunt grupate în raport cu benzile care le-au generat (prin ŞI logic pentru cele din stânga, respectiv prin SAU logic pentru cele din drepta). Metodei i se poate asocia un graf neorientat ale cărui noduri sunt celulele libere.

Dacă algoritmul grilei consideră C-obstacolele în locul obstacolelor, prin cercetarea grafului obținut se poate determina o traiectorie liberă a robotului. În acest context, pentru un spațiu de lucru bidimensional, s-a elaborat un program de modelare, dar şi de planificare a traiectoriei, pe baza metodei grilei neomogene. Programul este simplu de aplicat, iar extensia sa la un spațiu tridimensional este imediată. Algoritmul care a stat la baza acestui program este următorul:

- reprezintă obstacolele poligonale;
- reprezintă robotul şi cele două configurații, inițială şi finală;
- calculează şi reprezintă C-obstacolele;
- aproximează C-obstacolele prin dreptunghiuri acoperitoare;
- construieşte grila şi determină celulele libere;
- construieşte graful neorientat aferent celulelor libere;
- identifică în graf configurațiile inițială şi finală;
- cercetează graful pentru determinarea unui drum liber;
- optimizează drumul liber, dacă sunt mai multe variante sau raportează eşec dacă nu există nici-unul.

În figurile 7.1 şi 7.2 sunt prezentate două imagini capturate în timpul aplicării metodei grilei neomogene.

7.3. DETERMINAREA FRONTIERELOR C-OBSTACOLELOR. POLIGON CONVEX ÎN MIŞCARE DE TRANSLAȚIE

Se consideră robotul A şi obstacolele B_i modelate ca submulțimi compacte ale lui $\mathbf{R}^N$, ale căror poziții sunt descrise de două sisteme de referință distincte: sistemul S_A ataşat robotului şi sistemul S_W fix. Originea O_A a lui S_A este numită punctul de referință al lui A. Spațiul configurațiilor lui A este spațiul SC al tuturor configurațiilor C ale lui A. Un obstacol B_i din spațiul de lucru W defineşte în spațiul configurațiilor SC o regiune:

$$SCB_i = \{\mathbf{C} \in SC / A(\mathbf{C}) \cap B_i \neq \varnothing\} \quad (7.1)$$

denumită C-obstacol. Reuniunea tuturor C-obstacolelor determină regiunea SCB a C-obstacolelor, iar spațiul liber este dat de relația:

$$SC_{liber} = SC \setminus SCB = \{\mathbf{C} \in SC / A(\mathbf{C}) \cap SCB = \varnothing\} \quad (7.2)$$

Reprezentarea frontierei unui C-obstacol în cazul când robotul A este modelat ca poligon convex şi execută doar translație

Fie A şi B două poligoane convexe, A translatând cu orientare fixată θ_0. Se poate observa simplu că, dacă θ_0 este orientarea lui A față de B, când nici-o latură a lui A nu este paralelă cu o latură a lui B, atunci vârfurile lui SCB_{00} sunt toate punctele obținute după cum urmează:

- Dacă $APL_{i,j}^{A}(\theta_0)$ există, atunci punctele precizate de vectorii $(\mathbf{b}_j - \mathbf{a}_i(0,0,\theta_0))$ şi $(\mathbf{b}_j - \mathbf{a}_{i+1}(0,0,\theta_0))$ sunt vârfuri ale lui $SCB_{\theta 0}$.

BUPT

- Dacă $APL_{i,j}^{B}(\theta_0)$ există, atunci punctele precizate de vectorii ($\mathbf{b}_j - \mathbf{a}_i(0,0,\theta_0)$) şi ($\mathbf{b}_{j+1} - \mathbf{a}_i(0,0,\theta_0)$) sunt de asemenea vârfuri ale lui SCB_{q0}.

Această observaţie conduce la un algoritm eficient pentru obţinerea lui $SCB_{\theta 0}$ [L5]. Se trasează toţi versorii $-\mathbf{n}_i^A$ şi $\mathbf{n}_j^B$ în centrul cercului unitar S^1. Deoarece şi *A* şi *B* sunt poligoane convexe, unghiul considerat în sens trigonometric între oricare doi versori succesivi ($-\mathbf{n}_i^A$ şi $-\mathbf{n}_{i+1}^A$), sau ($\mathbf{n}_j^B$ şi $\mathbf{n}_{j+1}^B$), este mai mic decât π.

În aceste condiţii, θ_0 fiind o orientare curentă, oarecare a lui *A* faţă de *B*, rezultă că $APL_{i,j}^{A}(\theta_0)$ există dacă şi numai dacă versorul invers $-\mathbf{n}_i^A$ se află între $\mathbf{n}_{j-1}^B$ şi $\mathbf{n}_j^B$, adică $-\mathbf{n}_i^A$ este o combinaţie liniară pozitivă a celor doi versori ai lui *B* menţionaţi. În acelaşi mod, $APL_{i,j}^{B}(\theta_0)$ există dacă şi numai dacă $\mathbf{n}_j^B$ este plasat între $-\mathbf{n}_{i-1}^A(\theta_0)$ şi $-\mathbf{n}_i^A(\theta_0)$. Parcurgând versorii $-\mathbf{n}_i^A$ şi $\mathbf{n}_j^B$ în sens trigonometric rezultă succesiv vârfurile lui SCB_0.

În figura 7.3 este prezentat un exemplu de aplicare a algoritmului pentru calculul frontierei lui SCB când *A* şi *B* sunt modelate ca poligoane convexe, iar *A* translatează cu orientarea $\theta_0 = 0$, constantă.

Robotul: $a_1(1, -1)$, $a_2(-1, 2)$, $a_3(-1, -1)$;
Obstacolul: $b_1(0, 0)$, $b_2(8, 0)$, $b_3(8, 6)$, $b_4(0, 6)$.

1). $-\mathbf{n}_1^A$ între $\mathbf{n}_4^B$ şi $\mathbf{n}_1^B \Rightarrow APL_{1,1}^{A}$
$\mathbf{b}_1 - \mathbf{a}_1 \rightarrow SCB1(-1, 1)$; $\mathbf{b}_1 - \mathbf{a}_2 \rightarrow SCB2(1, -2)$
2). $\mathbf{n}_1^B$ între $-\mathbf{n}_1^A$ şi $-\mathbf{n}_2^A \Rightarrow APL_{2,1}^{B}$
$\mathbf{b}_1 - \mathbf{a}_2 \rightarrow SCB2(1, -2)$; $\mathbf{b}_2 - \mathbf{a}_2 \rightarrow SCB3(9, -2)$
3). $-\mathbf{n}_2^A$ între $\mathbf{n}_1^B$ şi $\mathbf{n}_2^B \Rightarrow APL_{2,2}^{A}$
$\mathbf{b}_2 - \mathbf{a}_2 \rightarrow SCB3(9, -2)$; $\mathbf{b}_2 - \mathbf{a}_3 \rightarrow SCB4(9, 1)$
4). $\mathbf{n}_2^B$ între $-\mathbf{n}_2^A$ şi $-\mathbf{n}_3^A \Rightarrow APL_{3,2}^{B}$
$\mathbf{b}_2 - \mathbf{a}_3 \rightarrow SCB4(9, 1)$; $\mathbf{b}_3 - \mathbf{a}_3 \rightarrow SCB5(9, 7)$
5). $-\mathbf{n}_3^A$ între $\mathbf{n}_2^B$ şi $\mathbf{n}_3^B \Rightarrow APL_{3,3}^{A}$
$\mathbf{b}_3 - \mathbf{a}_3 \rightarrow SCB5(9, 7)$; $\mathbf{b}_3 - \mathbf{a}_1 \rightarrow SCB6(7, 7)$
6). $\mathbf{n}_3^B$ între $-\mathbf{n}_3^A$ şi $-\mathbf{n}_1^A \Rightarrow APL_{1,3}^{B}$
$\mathbf{b}_3 - \mathbf{a}_1 \rightarrow SCB6(7, 7)$; $\mathbf{b}_4 - \mathbf{a}_1 \rightarrow SCB7(-1, 7)$
7). $\mathbf{n}_4^B$ între $-\mathbf{n}_3^A$ şi $-\mathbf{n}_1^A \Rightarrow APL_{1,4}^{B}$
$\mathbf{b}_4 - \mathbf{a}_1 \rightarrow SCB7(-1, 7)$; $\mathbf{b}_1 - \mathbf{a}_1 \rightarrow SCB1(-1, 1)$

În figura 7.4 este reluat exemplul din figura 7.3 pentru $\theta = 30°$. Pentru transformarea coordonatelor din S_A în A_W se foloseşte relaţia matriceală: $\mathbf{r'} = \mathbf{R}(z, q)\mathbf{r}$.

Robotul: $a_{1A}(1, -1, 30°)$, $a_{2A}(-1, 2, 30°)$, $a_{3A}(-1, -1, 30°)$;
Robotul: $a_{1W}(1.36, -0.36, 0°)$, $a_{2A}(-1.86, 1.22, 0°)$,
$a_{3A}(-0.36, -1.36, 0°)$;
Obstacolul: $b_1(0, 0)$, $b_2(8, 0)$, $b_3(8, 6)$, $b_4(0, 6)$.

1). $-\mathbf{n}_1^A$ între $\mathbf{n}_4^B$ şi $\mathbf{n}_1^B \Rightarrow APL_{1,1}^{A}$
$\mathbf{b}_1 - \mathbf{a}_1 \rightarrow SCB1(-1.36, 0.36)$; $\mathbf{b}_1 - \mathbf{a}_2 \rightarrow SCB2(1.86, -1.22)$
2). $\mathbf{n}_1^B$ între $-\mathbf{n}_1^A$ şi $-\mathbf{n}_2^A \Rightarrow APL_{2,1}^{B}$
$\mathbf{b}_1 - \mathbf{a}_2 \rightarrow SCB2(1.86, -1.22)$; $\mathbf{b}_2 - \mathbf{a}_2 \rightarrow SCB3(9.86, -1.22)$

BUPT

3). $\mathbf{n}_2^B$ între $-\mathbf{n}_1^A$ şi $-\mathbf{n}_2^A \Rightarrow APL_{2,2}{}^B$

$\mathbf{b}_2 - \mathbf{a}_2 \rightarrow$ SCB3(9.86, -1.22); $\mathbf{b}_3 - \mathbf{a}_2 \rightarrow$ SCB4(9.86, 4.78)

4). $-\mathbf{n}_2^A$ între $\mathbf{n}_2^B$ şi $\mathbf{n}_3^B \Rightarrow APL_{2,3}{}^A$

$\mathbf{b}_3 - \mathbf{a}_2 \rightarrow$ SCB4(9.86, 4.78); $\mathbf{b}_3 - \mathbf{a}_3 \rightarrow$ SCB5(8.36, 7.36)

5). $\mathbf{n}_3^B$ între $-\mathbf{n}_2^A$ şi $-\mathbf{n}_3^A \Rightarrow APL_{3,3}{}^B$

$\mathbf{b}_3 - \mathbf{a}_3 \rightarrow$ SCB5(8.36, 7.36); $\mathbf{b}_4 - \mathbf{a}_3 \rightarrow$ SCB6(0.36, 7.36)

6). $-\mathbf{n}_3^A$ între $\mathbf{n}_3^B$ şi $\mathbf{n}_4^B \Rightarrow APL_{3,4}{}^A$

$\mathbf{b}_4 - \mathbf{a}_3 \rightarrow$ SCB6(0.36, 7.36); $\mathbf{b}_4 - \mathbf{a}_1 \rightarrow$ SCB7(-1.36, 7.36)

7). $\mathbf{n}_4^B$ între $-\mathbf{n}_3^A$ şi $-\mathbf{n}_1^A \Rightarrow APL_{1,4}{}^B$

$\mathbf{b}_4 - \mathbf{a}_1 \rightarrow$ SCB7(-1.36, 7.36); $\mathbf{b}_1 - \mathbf{a}_1 \rightarrow$ SCB1(-1.36, 0.36)

Pe baza algoritmului de mai sus a fost realizat un program pentru calculul frontierelor C-obstacolelor, în cazul în care robotul *A* este modelat ca un poligon convex care translatează cu o orientare constantă θ_0 sau ca un cerc de rază *a*, într-un spațiu de lucru bi-dimensional, printre obstacole poligonale. Programul permite modificarea orientării robotului, ceea ce constituie primul pas spre generalizarea problemei. După construirea C-obstacolelor, poate fi aplicată imediat metoda grilei omogene, sau metoda grilei neomogene, sau metoda descompunerii celulare aproximative. De asemenea, programul poate fi implementat în orice alt program care rezolvă o problemă de planificare a traiectoriei în care este necesar să se determine C-obstacolele.

În figura 7.5 sunt prezentate câteva imagini reprezentative din timpul execuției programului:

- a) robotul *A* şi obstacolele B_1 şi B_2;
- b) C-obstacolul SCB_1 pentru $\theta=0^o$;
- c) cercul unitar şi versorii corespunzători robotului *A* şi obstacolului B_1;
- d) C-obstacolul SCB_2 pentru $\theta=0^o$;
- e) cercul unitar şi versorii corespunzători robotului *A* şi obstacolului B_2;
- f) C-obstacolul SCB_1 pentru $\theta=30^o$;
- g) C-obstacolul SCB_2 pentru $\theta=30^o$;
- h) imagine generală ecran;
- i) memorarea suprafețelor C-obstacolelor;
- j) generarea unei traiectorii prin aplicarea metodei grilei omogene.

BUPT

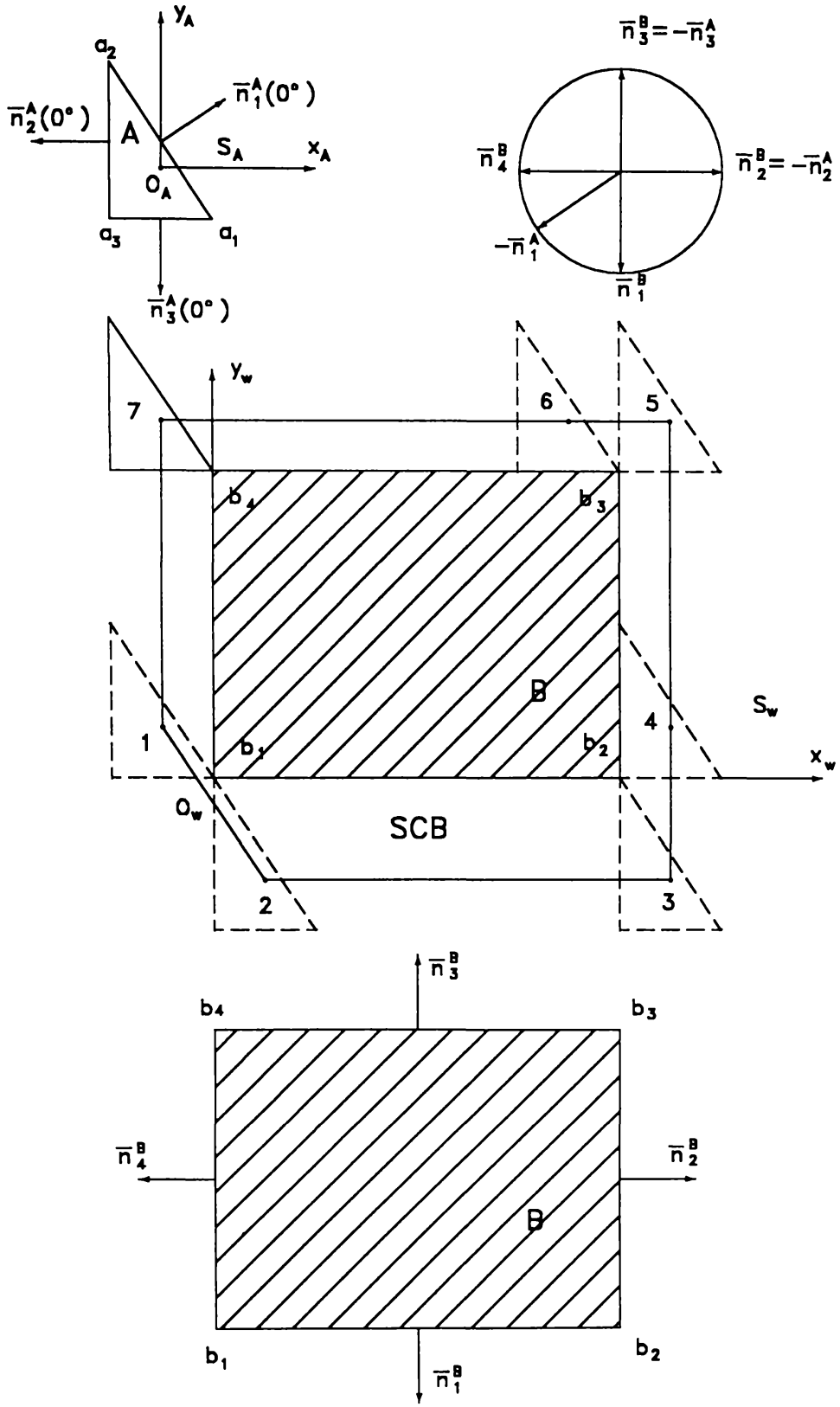


Fig.7.3. Frontiera C-obstacolului SCB pentru $\theta=0^o$

BUPT

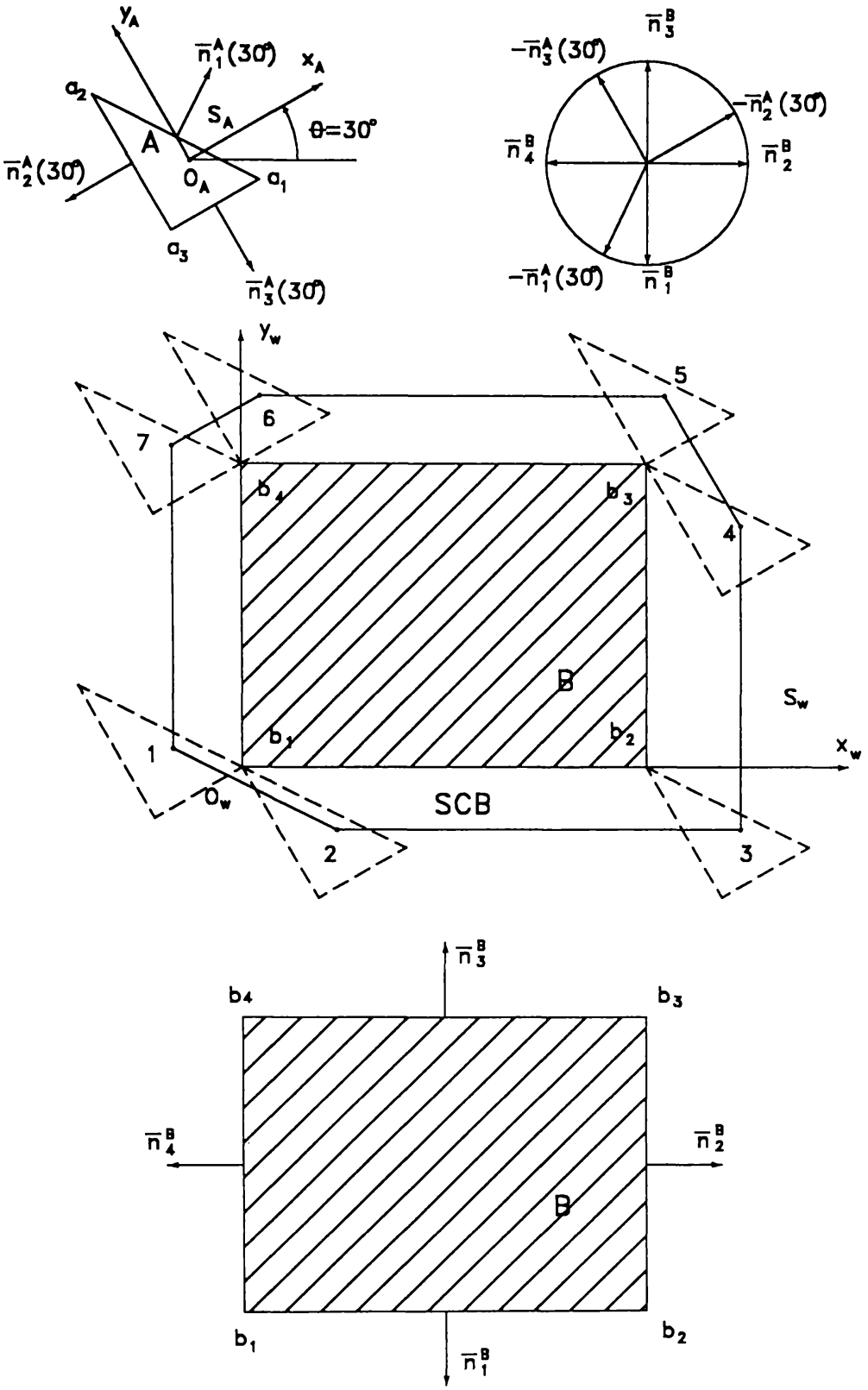
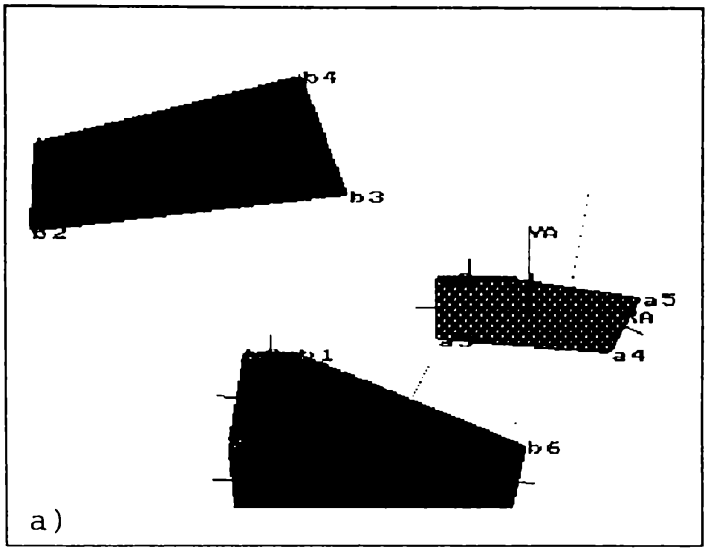
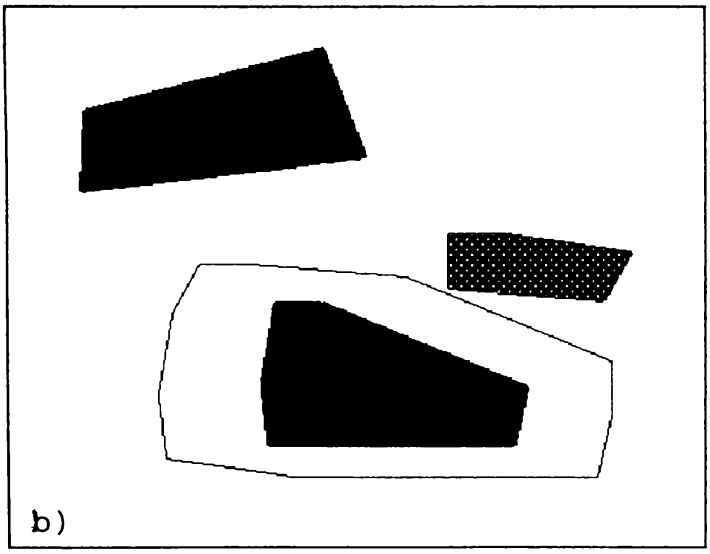
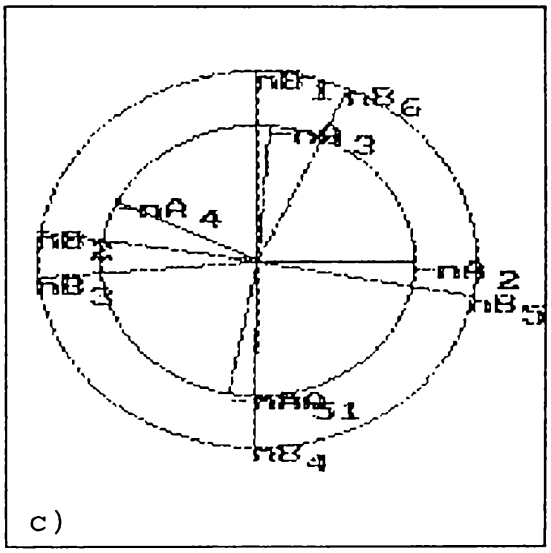


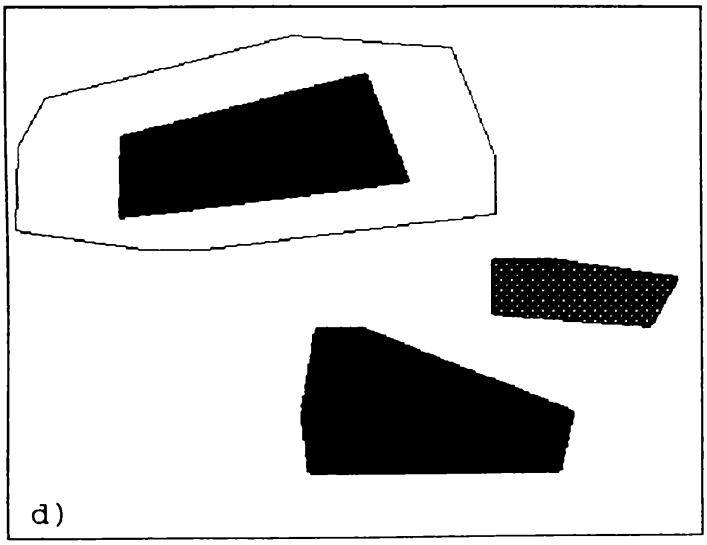
Fig.7.4. Frontiera C-obstacolului SCB pentru θ=30°

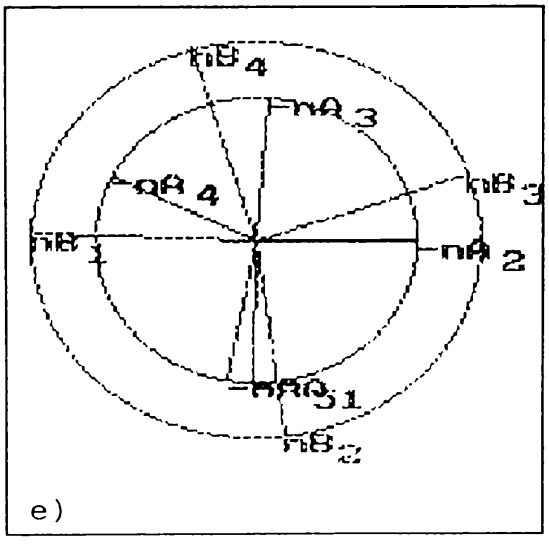
BUPT



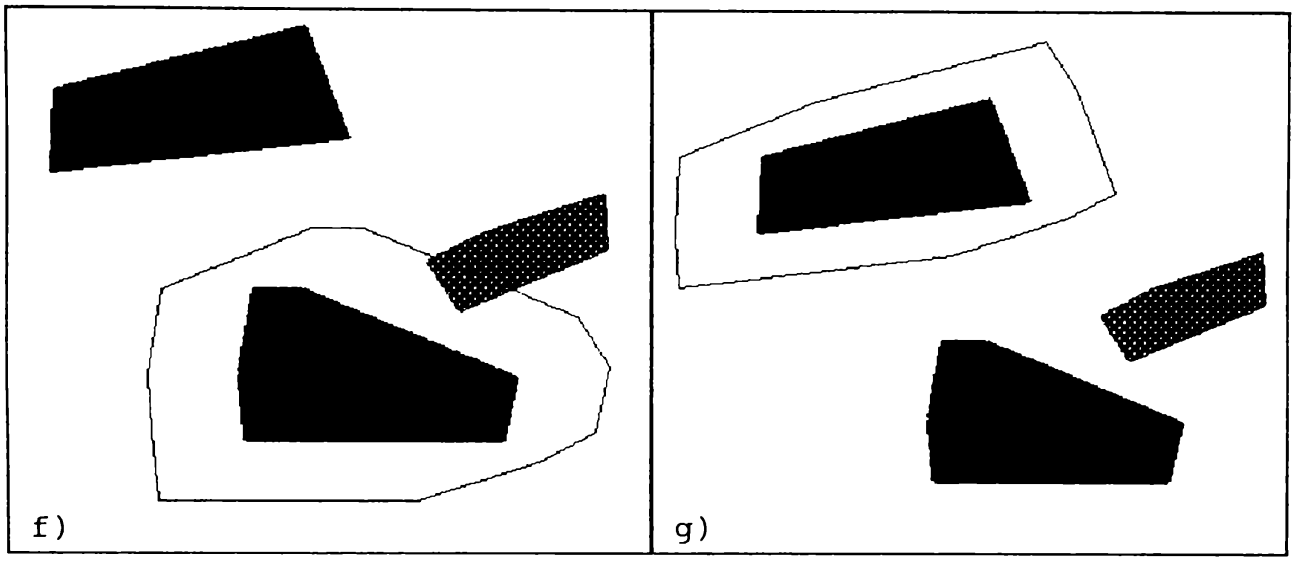


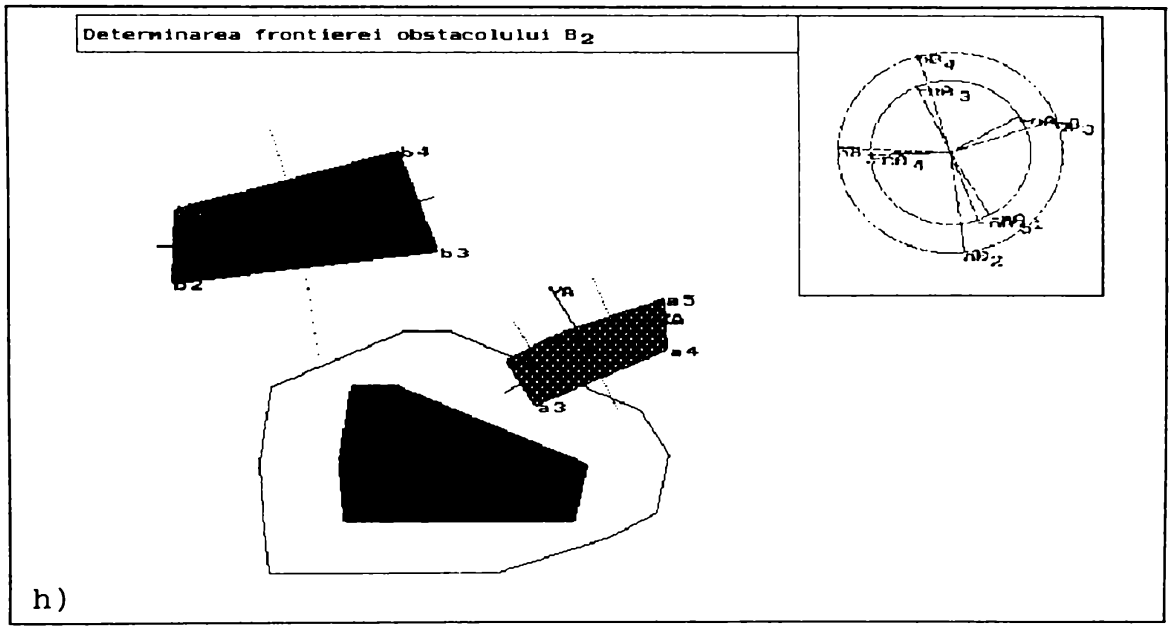






BUPT





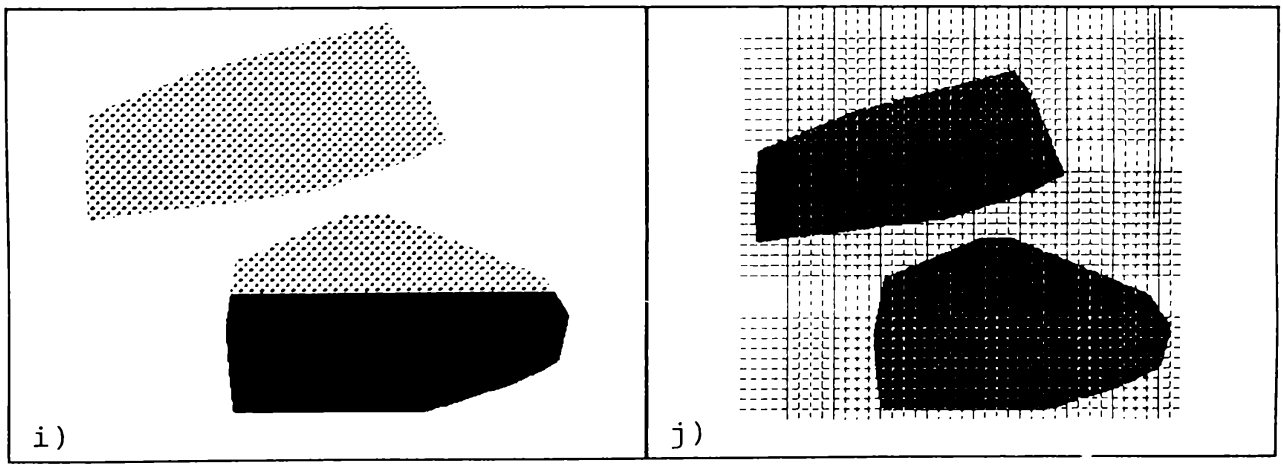


Fig.7.5. Determinarea frontierelor C-obstacolelor

BUPT

7.4. METODA GRAFULUI VIZIBILITĂȚII

Se consideră robotul A modelat ca un poligon în $SC=R^2$. Robotul translatează cu orientare constantă în spațiul de lucru $W=\mathbf{R}^2$. Regiunea SCB a C-obstacolelor este o regiune poligonală a lui $\mathbf{R}^2$.

Principiul metodei constă în determinarea unei traiectorii T semilibere, ca o linie poligonală ce conectează configurația inițială $\mathbf{C}_i$ de configurația finală $\mathbf{C}_f$, trecând prin vârfurile lui SCB. Pentru a determina o traiectorie semiliberă între două configurații se pot considera liniile poligonale ce unesc vârfurile lui SCB. Această mulțime de linii poligonale, numită *graf al vizibilității*, conține și cea mai scurtă traiectorie.

Algoritmul metodei grafului vizibilitătii conține următoarele etape:

- se construiește graful vizibilității G;
- se cercetează G pentru determinarea unei traiectorii între $\mathbf{C}_i$ și $\mathbf{C}_f$;
- dacă se găsește o traiectorie, se comunică aceasta, iar în caz contrar se comunică "eșec".

Graful vizibilității este graful neorientat G, definit astfel:

- nodurile lui G sunt $\mathbf{C}_i$, $\mathbf{C}_f$ și vârfurile lui SCB;
- două noduri ale lui G sunt conectate printr-o legătură în graf dacă și numai dacă, fie segmentul de dreaptă ce le unește este o latură a lui SCB, fie se află în întregime în SC_{liber}.

În fig.7.6 este prezentat graful vizibilității pentru un spațiu SC simplu, în cazul considerat.

Eficiența metodei grafului vizibilității poate fi mărită dacă se elimină anumite legături ale grafului. Astfel, subgraful G', obținut din G prin eliminarea segmentelor netangente și a vârfurilor concave, se numește *graful vizibilității redus*. Ori de câte ori există o traiectorie semiliberă între $\mathbf{C}_i$ și $\mathbf{C}_f$, G' conține cea mai scurtă traiectorie între aceste configurații. În fig.7.7 este prezentat graful vizibilității redus, pentru același spațiu al configurațiilor ca în fig.7.6.

Metoda grafului vizibilității poate fi extinsă în cazul în care C-obstacolele sunt regiuni mărginite de segmente de dreaptă și/sau arce de cerc. Astfel de C-obstacole apar când A este un poligon care translatează cu orientare constantă printre obstacole modelate de asemenea ca regiuni mărginite de segmente de dreaptă și/sau arce de cerc. Modelarea obiectelor din spațiul de lucru W ca "poligoane generalizate" este mai frecvent utilizată deoarece caracterizează cu fidelitate conturul acestora și conduce la o precizie mai bună.

Se consideră $SC=\mathbf{R}^2$ populat cu C-obstacole de tip poligoane generalizate, a căror reuniune este SCB.

Graful vizibilității (redus) generalizat G este definit astfel:

- $\mathbf{C}_i$, $\mathbf{C}_f$ și vârfurile convexe ale lui SCB sunt nodurile lui G;
- Fie X și X' două noduri din cele definite mai sus. Dacă segmentul deschis care unește X și X' se află integral în SC_{liber} și este un segment tangent la SCB, atunci el este o legătură a lui G;
- Fie X un nod din cele definite mai sus și E' o latură circulară a lui SCB. Dacă există un punct X' care aparține lui E', astfel încât segmentul deschis care unește X și X' se află integral în SC_{liber} și dreapta suport a acestui segment este tangentă la SCB în X și X', atunci punctul X' este nod al lui G și segmentul care unește X și X' este o legătură a lui G;
- Fie E și E' două laturi circulare ale lui SCB. Dacă există un punct X în E și un punct X' în E', astfel încât segmentul deschis care unește X și X' se află integral în SC_{liber} și dreapta

BUPT

suport a acestui segment este tangentă la *SCB* în X şi X', atunci cele două puncte X şi X' sunt noduri ale lui *G* şi segmentul care le uneşte este o legătură a lui *G*;

•Fiecare latură dreaptă a lui *SCB*, care conectează două vârfuri convexe, este o legătură a lui *G*;

•Fiecare două noduri X şi X' ale lui *G*, conținute în aceeaşi latură circulară E a lui *SCB*, sunt conectate printr-o legătură în *G*, dacă nu există un alt nod X" al lui *G* care se află în E, între X şi X'.

În fig.7.9 şi 7.10 sunt prezentate graful vizibilității şi graful vizibilității redus generalizat pentru un spațiu al configurațiilor $SC=\mathbf{R}^2$, populat cu C-obstacole de tip poligoane generalizate.

Metoda grafului vizibilității este simplu de aplicat într-un $W=\mathbf{R}^2$, când robotul, modelat ca un poligon (generalizat), translatează cu orientarea fixată printre obstacole modelate, de asemenea, ca poligoane (generalizate). În acest caz, algoritmul metodei permite generarea unei traiectorii minime semilibere, între $\mathbf{C}_i$ şi $\mathbf{C}_f$. În fig.7.8 şi 7.11 sunt prezentate traiectoriile minime pentru cele două cazuri studiate.

Dacă robotul este un poligon care translatează şi se roteşte într-un spațiu $W=\mathbf{R}^2$, metoda grafului vizibilității trebuie combinată cu alte tehnici de planificare a traiectoriilor, pentru a putea trata rotația.

Pentru planificarea mişcării prin metoda grafului vizibilității s-a realizat un program care permite determinarea drumului minim într-un spațiu de lucru bi-dimensional. Programul este scris în limbajul Turbo Pascal, versiunea 6.0.

Coordonatele absolute ale spațiului de lucru sunt limitate în intervalul [-32000, 32000] unități convenționale. Cu această reprezentare, programul prezintă o precizie de 1mm, pentru spații de lucru care nu depăşesc 64m pe una din axe. Afişarea se realizează prin scalarea spațiului W pentru ca reprezentarea să cuprindă întregul ecran.

Etapele de lucru ale programului sunt:

•*Reprezentarea C-obstacolelor.* Fiecare C-obstacol este introdus ca o succesiune a vârfurilor sale (coordonatele x, y ale acestora), parcurse în sens trigonometric.

•*Stabilirea configurațiilor* C_i şi C_f. Cele două configurații sunt introduse prin selectarea coordonatelor acestora.

•*Reprezentarea grafului de vizibilitate (generalizat).*

•*Reprezentarea grafului de vizibilitate redus (generalizat).*

•*Cercetarea grafului redus şi calculul drumului minim.*

Timpul de lucru al programului depinde de numărul şi geometria C-obstacolelor (cazul poligoanelor generalizate).

BUPT

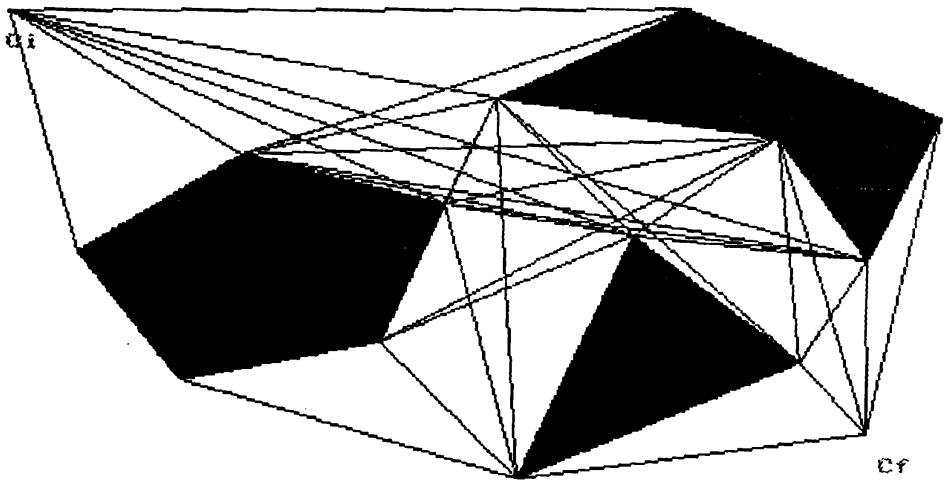


Fig.7.6. Regiuni poligonale. Graful vizibilității

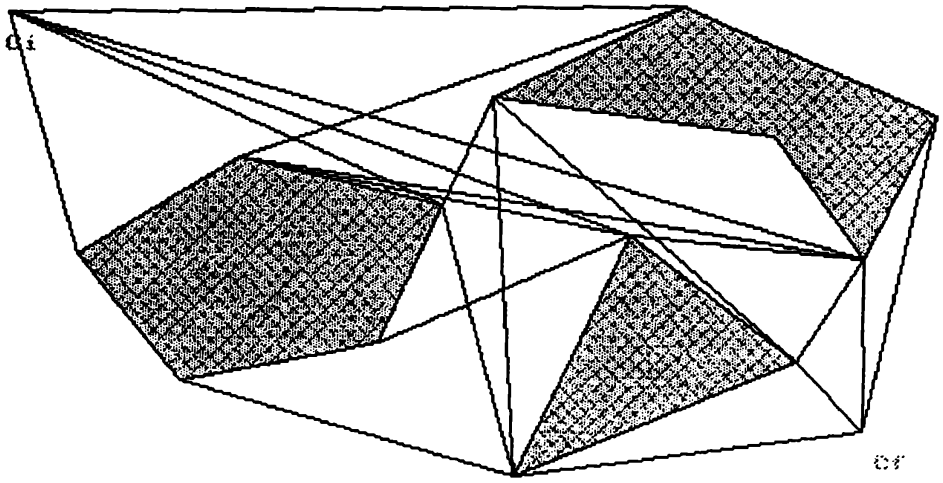


Fig.7.7. Regiuni poligonale. Graful vizibilității redus

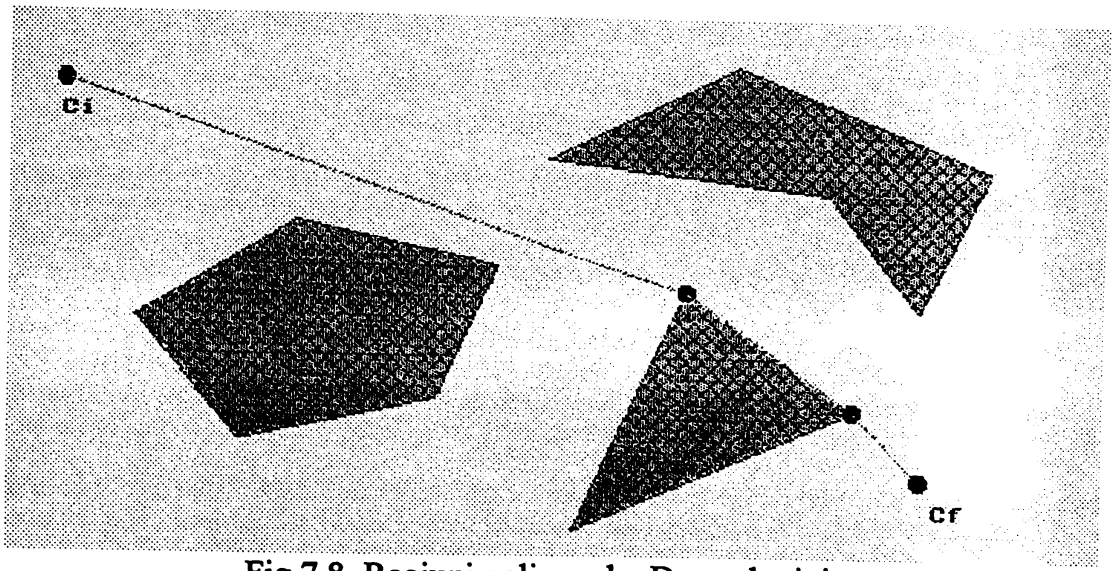


Fig.7.8. Regiuni poligonale. Drumul minim

BUPT

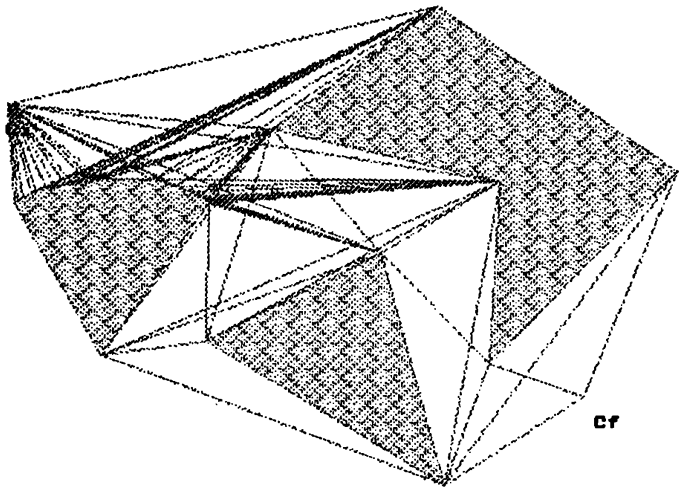


Fig.7.9. Regiuni bi-dimensionale oarecare. Graful vizibilității

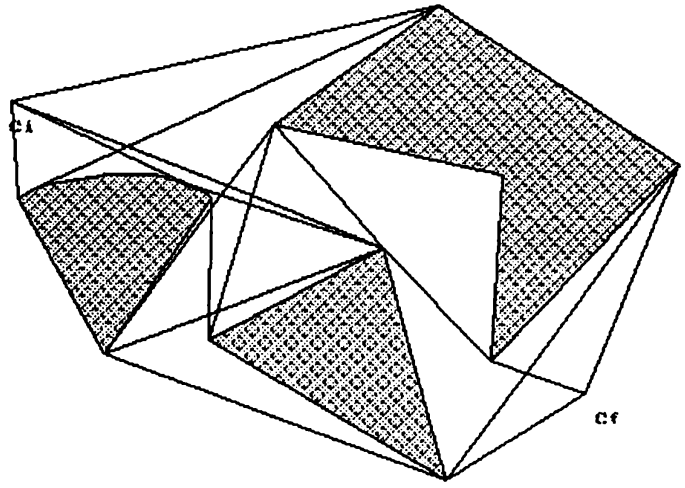


Fig.7.10. Regiuni bi-dimensionale oarecare. Graful vizibilității redus

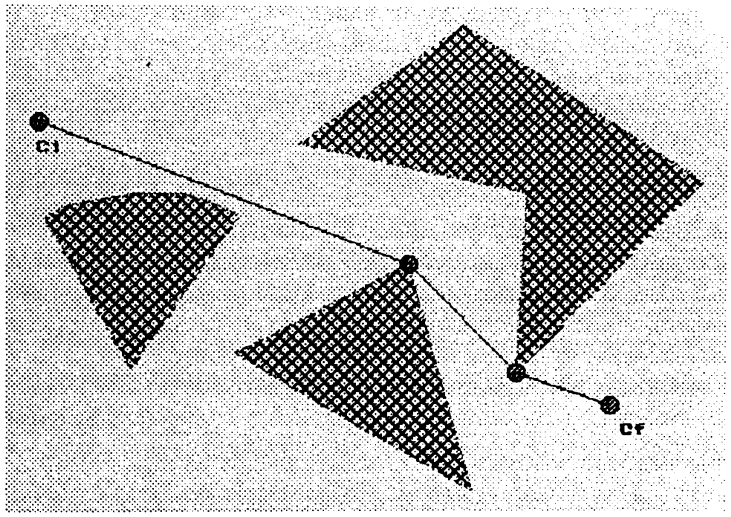


Fig.7.11. Regiuni bi-dimensionale oarecare. Drumul minim

BUPT

7.5. METODA DRUMULUI LIBER

Metoda drumului liber poate fi aplicată într-un spaţiu de lucru poligonal (bi-dimensional), robotul fiind modelat ca un poligon care execută o mişcare plană. Această metodă constă în extragerea unor figuri geometrice, denumite *drumuri libere*, din spaţiul de lucru, conectarea lor într-un graf, denumit *reţeaua drumurilor libere* şi cercetarea acestui graf. Robotul A se poate deplasa de-a lungul unui drum liber, sau de-a lungul unei porţiuni a acestuia, dacă există o mulţime conexă de orientări libere ale lui A când originea sistemului de referinţă ataşat se deplasează de-a lungul axei drumului liber corespunzător [D41], [L5].

Un drum liber poate fi generat de o pereche de laturi ce corespund la două obstacole vecine. Axa drumului liber este bisectoarea unghiului format de dreptele suport ale celor două laturi (fig.7.12). Alegerea bisectoarei ca axă a drumului liber este de fapt o reconsiderare a diagramei Voronoï în spaţiul de lucru respectiv. Prin convenţie, axa unui drum liber este orientată dinspre capătul mai larg, spre capătul mai îngust. Originea axei se plasează la capătul mai larg, astfel încât orice punct din drumul liber să poată fi proiectat pe axă la o abscisă pozitivă.

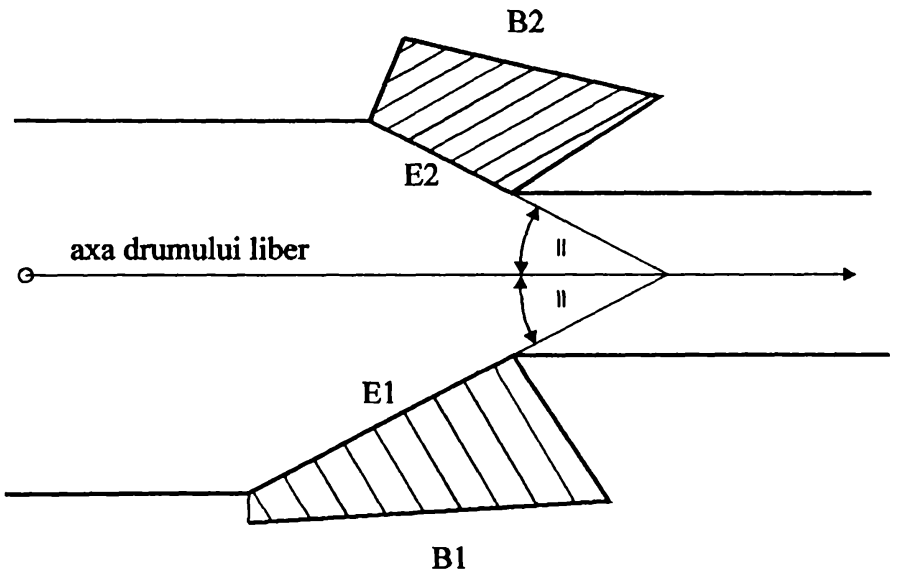


Fig.7.12. Construcţia unui drum liber

Drumul liber generat de o pereche de laturi ar trebui să fie situat în W\B, dar acesta se află doar parţial în spaţiul liber datorită obstacolelor din vecinătatea sa. Apare astfel necesitatea intersectării fiecărui drum liber cu regiunea obstacolelor B, deoarece la planificarea traiectoriei se iau în considerare doar porţiunile de drum liber conţinute în W\B. Intersecţia este proiectată pe axa drumului liber, iar sectoarele care nu aparţin spaţiului liber sunt înlăturate. Rămâne o mulţime finită de porţiuni Φ care constituie reţeaua drumurilor libere (fig.7.13).

BUPT

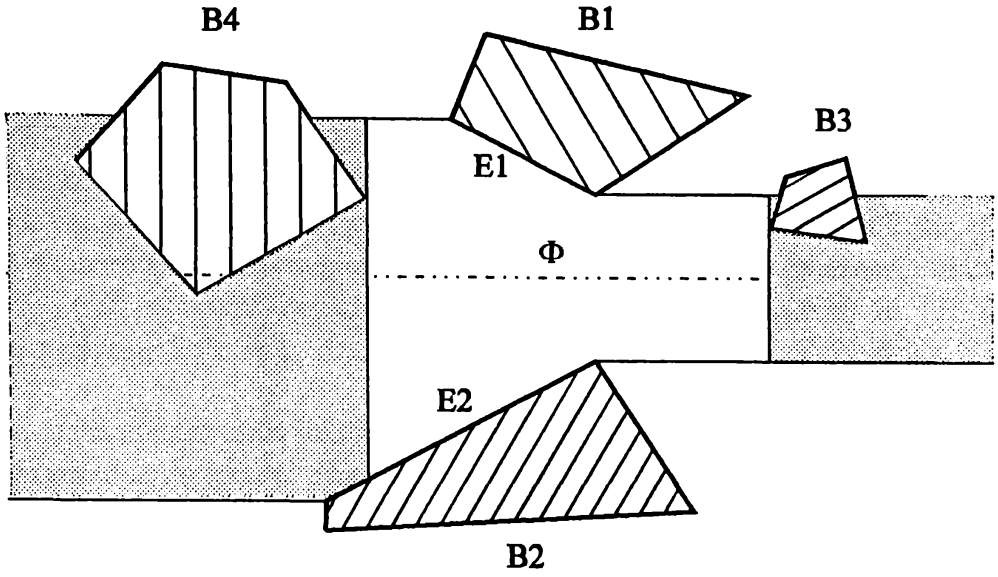


Fig.7.13. Selectarea porţiunilor de drum liber conţinute în W\B

Pentru definirea completă a unui drum liber este necesară descrierea tuturor orientărilor libere ale robotului când originea sistemului de referinţă ataşat se deplasează pe axa sa. Pentru a optimiza intervalul orientărilor libere ale robotului modelat ca poligon, este preferabil să se aleagă O_A astfel încât distanţa maximă de la O_A la punctele de la frontiera lui A să fie cea mai mică posibil. Acest lucru presupune ca O_A să fie centrul celui mai mic cerc care conţine poligonul. O tehnică eficientă de aproximare a intervalului orientărilor libere ale robotului utilizează funcţia rază şi inversa ei [L5].

Respectând metodologia indicată se ajunge la o colecţie de drumuri libere. Se utilizează acum această reprezentare pentru a construi graful neorientat G, care reprezintă reţeaua drumurilor libere. G determină toate drumurile posibile când A se deplasează de-a lungul axelor.

Fie χ mulţimea punctelor P de intersecţie a axelor drumurilor libere; fiecare punct P se află în interiorul ambelor drumuri libere care se intersectează. Algoritmul care creează nodurile lui G conţine următoarele etape:

- selectează toate punctele posibile P ale lui χ;
- determină abscisele s ale punctelor P;
- creează un nod al lui G pentru fiecare interval în $F_\Phi(s)$.

Deoarece fiecare interval este definit iniţial faţă de orientarea unei axe, ambelor extremităţi ale intervalului li se adaugă un unghi potrivit, astfel încât toate intervalele să fie definite faţă de aceeaşi orientare de referinţă, spre exemplu axa Ox a sistemului de referinţă S_W.

Poziţiile iniţială şi finală ale punctului O_A sunt utilizate pentru a crea nodurile, iniţial şi final, ale lui G. Se presupune că aceste poziţii se află pe axele unor drumuri libere Φ_1, respectiv Φ_2, la abscisele $s_{iniţial}$, respectiv s_{final}. Nodurile corespunzătoare ale lui G sunt astfel asociate intervalelor $F_{\Phi 1}(s_{iniţial})$ şi $F_{\Phi 2}(s_{final})$ care conţin orientările, iniţială şi finală, ale robotului. Dacă aceste poziţii nu sunt situate pe axele unor drumuri libere, atunci trebuie concepută o tehnică de deplasare a lui A, din poziţia iniţială pe o axă şi de pe o altă axă în poziţia finală.

Legăturile între nodurile lui G se creează astfel:

BUPT

• dacă N_1 şi N_2 sunt două noduri corespunzătoare la două puncte diferite de pe aceeaşi axă, având abscisele s_1 şi s_2, iar intervalele de orientări libere ale lui A, asociate cu aceste noduri sunt I_1, respectiv I_2, atunci condiţia $I_1 \cap I_2 \neq \varnothing$ arată că cele două noduri sunt conectate printr-o legătură în G; Pentru orice orientare în $I_1 \cap I_2$, A poate translata în spaţiul liber de la abscisa s_1 la abscisa s_2 şi invers;

• dacă N_1 şi N_2 sunt două noduri corespunzătoare aceluiaşi punct P din planul xOy, pentru două drumuri libere diferite, atunci, dacă intervalele de orientări libere satisfac condiţia $I_1 \cap I_2 \neq \varnothing$, nodurile respective vor fi conectate printr-o legătură în G; într-adevăr, pentru orice orientare din $I_1 \cap I_2$, A se poate muta în spaţiul liber, de pe un drum liber pe altul, în P.

Deci, pentru construirea lui G este suficient să se calculeze $F_\Phi(s)$ într-un număr finit de abscise de-a lungul axei fiecărui drum liber ϕ. Reţeaua drumurilor libere astfel construită se va cerceta pentru a găsi o traiectorie între nodul iniţial şi cel final. O traiectorie din reţea determină o mulţime de drumuri libere în SC_{liber}. O traiectorie liberă particulară se poate obţine rotind A doar la intersecţia între două axe. Robotul translatează de-a lungul axelor şi se roteşte, dacă este necesar, numai la intersecţia a două axe.

Un aspect neplăcut în utilizarea acestei metode apare atunci când spaţiul de lucru este atât de dens populat cu obstacole, încât drumurile libere sunt prea scurte pentru a putea conţine dreptunghiul circumscris lui A. Într-un astfel de caz, metoda trebuie să ia în considerare, fie construirea unor drumuri libere adiţionale, fie trasarea unor axe care nu sunt bisectoarele dreptelor suport ale muchiilor obstacolelor.

7.6. PLANIFICAREA TRAIECTORIEI PRIN METODA DESCOMPUNERII CELULARE EXACTE

7.6.1. PRINCIPIUL METODEI DESCOMPUNERII CELULARE

Planificarea mişcării este o problemă majoră în crearea roboţilor autonomi. Există un mare număr de metode de planificare, dar nu toate dintre acestea rezolvă problema în întreaga sa generalitate. De exemplu, anumite metode necesită ca spaţiul de lucru al robotului să fie bi-dimensional, iar obiectele să fie poligonale (metoda hărţii drumurilor, descompunerea celulară, câmpul potenţial).

Metoda descompunerii celulare constă în descompunerea spaţiului liber al robotului într-o colecţie de regiuni simple, numite celule, astfel încât o traiectorie între oricare două configuraţii ($\mathbf{C}_{iniţial}$ şi $\mathbf{C}_{final}$) dintr-o celulă poate fi generată cu uşurinţă. Apoi este construit şi cercetat un graf neorientat care exprimă relaţia de adiacenţă dintre celule. Acest graf se numeşte graf de conexiune. Nodurile sale sunt celulele extrase din spaţiul liber, iar două noduri sunt conectate printr-o legătură în graf dacă şi numai dacă cele două celule corespunzătoare sunt adiacente. Rezultatul cercetării este o secvenţă de celule numită canal, care conectează configuraţia iniţială cu cea finală. În această secvenţă poate fi determinată o traiectorie liberă, continuă.

Metoda descompunerii celulare poate fi descompusă în două metode: descompunere celulară exactă şi aproximativă. Metoda descompunerii celulare exacte descompune spaţiul liber în celule, a căror reuniune este exact spaţiul liber. Frontiera unei celule corespunde unei zone critice, adică apare o modificare în restricţiile care se aplică

BUPT

mişcării robotului. Metoda descompunerii celulare exacte produce celule cu o formă predefinită a căror reuniune este strict inclusă în spaţiul liber. Frontiera unei celule nu caracterizează o discontinuitate de un anumit tip şi nu are o semnificaţie fizică.

Pentru planificarea mişcării unui conducător sau poligon au fost propuse mai multe metode de descompunere celulară exactă. Totuşi, cele mai multe din aceste metode sunt dificil de implementat.

7.6.2. PRINCIPIUL METODEI DESCOMPUNERII CELULARE EXACTE

Robotul A este modelat ca un segment de dreaptă care se poate translata şi roti într-un spaţiu de lucru bi-dimensional $W=R^2$. Spaţiul de lucru este populat cu obstacole a căror reuniune formează o regiune poligonală notată cu B. Spaţiul configuraţiilor robotului SC este $SC=R^2xS^1$. Fiecare configuraţie a robotului este parametrizată prin (x,y,θ), unde x şi y sunt coordonatele extremităţii M în sistemul de coordonate S_W ataşat spaţiului de lucru W, iar $\theta\in[0, 2\pi)$ este unghiul dintre axa x a lui S_W şi vectorul ce conectează M cu N. Lungimea segmentului MN este a.

Ideea de bază a acestui concept este de a descompune mulţimea poziţiilor robotului în regiuni necritice bi-dimensionale, de a transforma aceste regiuni în celule tri-dimensionale şi de a reprezenta relaţia de adiacenţă dintre celule în graful de conexiune. Regiunile necritice sunt astfel încât C-obstacolele menţin o structură constantă în cilindrii de deasupra acestor regiuni. Celulele conţinute în astfel de cilindri fac această structură explicită. Frontierele regiunilor necritice sunt denumite curbe critice.

Curbele critice sunt o mulţime de puncte în care structura regiunii C-obstacolelor de deasupra planului xOy suferă o schimbare calitativă. Aceste curbe pot fi definite ca proiecţii pe planul xOy ale următoarelor curbe: frontierele feţelor regiunii C-obstacolelor; curbele din feţele regiunii C-obstacolelor, unde planul tangent la regiunea C-obstacolelor este perpendicular pe planul xOy. Ori de câte ori este traversată o astfel de curbă se modifică sau mulţimea feţelor regiunii C-obstacolelor care sunt intersectate de o dreaptă perpendiculară pe planul xOy în poziţia curentă, sau numărul punctelor de intersecţie.

Curbele critice pot fi clasificate în şase tipuri diferite (Fig.7.14):

- tip 0: laturile E ale obstacolelor;
- tip 1: segmentul de dreaptă, paralel cu latura E, la distanţa a;
- tip 2: arcul de cerc de rază a, cu centrul în vârful X al unui obstacol şi mărginit de cele două semidrepte care conţin laturile care se intersectează în X;
- tip 3: segmentul de dreaptă trasat de originea sistemului de referinţă ataşat robotului când acesta glisează de-a lungul laturii E, atâta timp cât celălalt capăt este încă conţinut în latura E;
- tip 4: segmentul de dreaptă trasat de originea sistemului de referinţă ataşat robotului când acesta glisează astfel încât atinge simultan vârfurile a două obstacole convexe;
- tip 5: curba trasată de originea sistemului de referinţă ataşat robotului când acesta se mişcă astfel încât atinge simultan vârful unui obstacol convex şi latura unui alt obstacol.

BUPT

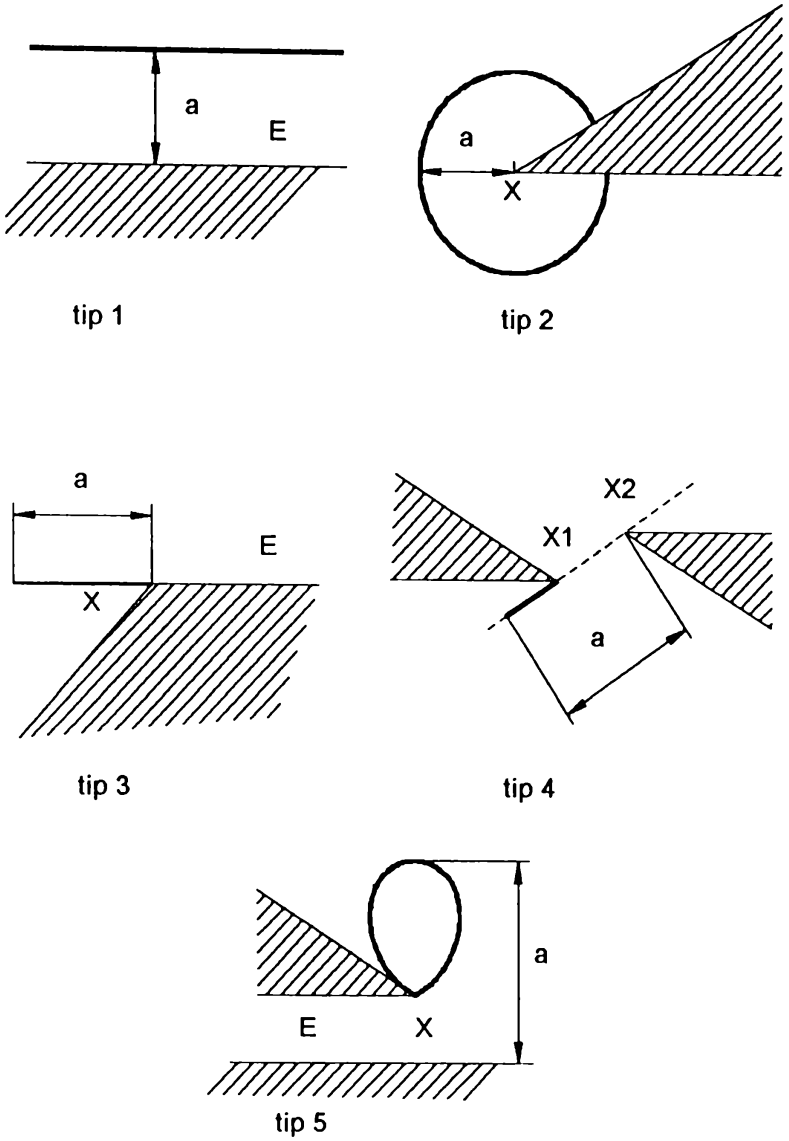


Fig.7.14. Clasificarea curbelor critice

Mulțimea curbelor critice este finită. Fiecare punct de capăt este localizat fie pe altă curbă critică, fie la infinit. Fiecare două curbe critice pot avea doar un număr finit de intersecții, cu excepția cazului când acestea coincid. Astfel, se poate defini o porțiune de curbă critică, ca orice submulțime conexă închisă maximală a curbei, care nu este intersectată de nici-o altă curbă critică, cu excepția extremităților sale. Curbele critice sunt curbe algebrice netede. Unele din curbele critice sunt generate de către configurațiile robotului atunci când acesta se intersectează cu interiorul regiunii obstacolelor. Aceste curbe sunt numite redundante. Ele formează o submulțime a curbelor critice şi pot fi eliminate uşor. Mulțimea finită a curbelor critice determină o colecție finită de regiuni necritice. Fiecare regiune necritică este o submulțime deschisă a planului xOy.

Pentru a determina mulțimea tuturor orientărilor libere ale robotului intr-o poziție necritică, este definită noțiunea de stop. Muchiile şi vârfurile obstacolelor, care pot fi atinse fără intersectarea interiorului regiunii obstacolelor, când robotul este într-o poziție necritică şi se roteşte în jurul originii sistemului ataşat (extremitatea M), sunt numite

BUPT

stopuri. Există două tipuri de stopuri care determină orientările limită ale robotului: stopuri orare şi stopuri trigonometrice.

Fie F(x,y) mulţimea tuturor orientărilor libere ale robotului într-o poziţie necritică (x,y):

$$F(x,y) = \{ \theta / (x,y,\theta) \in SC_{liber} \} \quad (7.3)$$

Se notează cu $s(x,y,\theta_c)$ stopul unic atins de către robot în configuraţia (x,y,θ_c) şi cu $\sigma(x,y)$ mulţimea tuturor perechilor $[s(x,y,\theta_c), s(x,y,\theta'_c)]$, unde θ_c, respectiv θ'_c reprezintă orientarea limită în poziţia necritică (x,y).

Fie R o regiune necritică şi $[s_1,s_2]$ o pereche de stopuri în $\sigma(x,y)=\sigma(R)$. Regiunea tri-dimensională

$$K(R,s_1,s_2)=\{(x,y,\theta) / (x,y)\in \mathbf{R} \quad şi \quad \theta\in[\lambda_1(x,y,s_1), \lambda_2(x,y,s_2)]\} \quad (7.4)$$

este numită celulă. Aceasta este o submulţime conexă deschisă a lui SC_{liber}.

După descompunerea spaţiului liber în celule conexe deschise, trebuie să se definească graful de conexiune asociat acestei descompuneri. Pentru a obţine graful de conexiune trebuie să se studieze relaţia de adiacenţă dintre celule.

Fie două celule $K(R,s_1,s_2)$ şi $K'(R',s'_1,s'_2)$. Acestea sunt adiacente dacă şi numai dacă frontierele lui R şi R' împart o porţiune de curbă critică L şi $\forall(x,y)\in int(L):(\lambda_1,\lambda_2)\cap(\lambda'_1,\lambda'_2)\neq\varnothing$. Dacă două celule sunt adiacente, orice configuraţie care aparţine uneia din ele poate fi conectată cu orice configuraţie care aparţine celeilalte, printr-un drum a cărui proiecţie pe planul xOy traversează L cu o orientare constantă, într-o anumită vecinătate a punctului de trecere.

Graful de conexiune G este graful neorientat ale cărui noduri sunt toate celulele $K(R,s_1,s_2)$, unde R este o regiune necritică şi $[s_1,s_2]\in\sigma(R)$. O legătură în graf conectează două noduri dacă şi numai dacă celulele corespunzătoare sunt adiacente.

Astfel, dându-se două configuraţii: $\mathbf{C}_{iniţial}(x_{iniţial},y_{iniţial},\theta_{iniţial})$ şi $\mathbf{C}_{final}(x_{final},y_{final},\theta_{final})$ din spaţiul liber, astfel încât nici $(x_{iniţial},y_{iniţial})$ şi nici (x_{final},y_{final}) nu sunt situate pe o porţiune de curbă critică, există un drum liber între cele două configuraţii dacă şi numai dacă cele două celule care conţin $\mathbf{C}_{iniţial}$ şi $\mathbf{C}_{final}$ sunt conectate printr-un drum în graful de conexiune G.

7.6.3. ALGORITMUL METODEI DESCOMPUNERII CELULARE EXACTE

S-a realizat un program bazat pe metoda descompunerii celulare exacte, care determină un drum liber pentru un robot modelat ca un segment de dreaptă, care se deplasează printre obstacole poligonale, într-un spaţiu de lucru bi-dimensional. Dacă se cunosc regiunea poligonală a obstacolelor, lungimea *a* a robotului şi cele două configuraţii, iniţială şi finală, din spaţiul liber, algoritmul generează un drum liber de la $\mathbf{C}_{iniţial}$ la $\mathbf{C}_{final}$, ori de câte ori există vreunul şi raportează eşec în caz contrar.

Algoritmul metodei descompunerii celulare exacte pentru planificarea unui drum liber care conectează cele două configuraţii este următorul:

- stabileşte lungimea robotului;
- reprezintă obstacolele care populează spaţiul de lucru;
- selectează configuraţiile $\mathbf{C}_{iniţial}$ şi $\mathbf{C}_{final}$;
- calculează şi reprezintă toate curbele critice din fig.7.14;

BUPT

• calculează toate intersecțiile curbelor critice şi determină porțiunile curbelor critice; elimină porțiunile de curbe critice redundante;
• cercetează şi memorează regiunile necritice. Fiecare regiune este etichetată cu un număr întreg. Pentru exemplul prezentat, $C_{inițial}$ se află în regiunea necritică R2, iar C_{final} în regiunea necritică R46;
• verifică dacă atât $C_{inițial}$, cât şi C_{final} se află pe o porțiune de curbă critică şi modifică $C_{inițial}$, sau C_{final}, dacă una din ele aparține unei porțiuni de curbă critică;
• cercetează şi memorează mulțimea stopurilor orare şi trigonometrice;
• studiază relația de adiacență dintre celule;
• construieşte graful de conexiune asociat colecției regiunilor necritice;
• cercetează G în vederea determinării unei secvențe de celule între $C_{inițial}$ şi C_{final};
• dacă cercetarea se termină cu succes este furnizată secvența de celule; în caz contrar declară eşec.

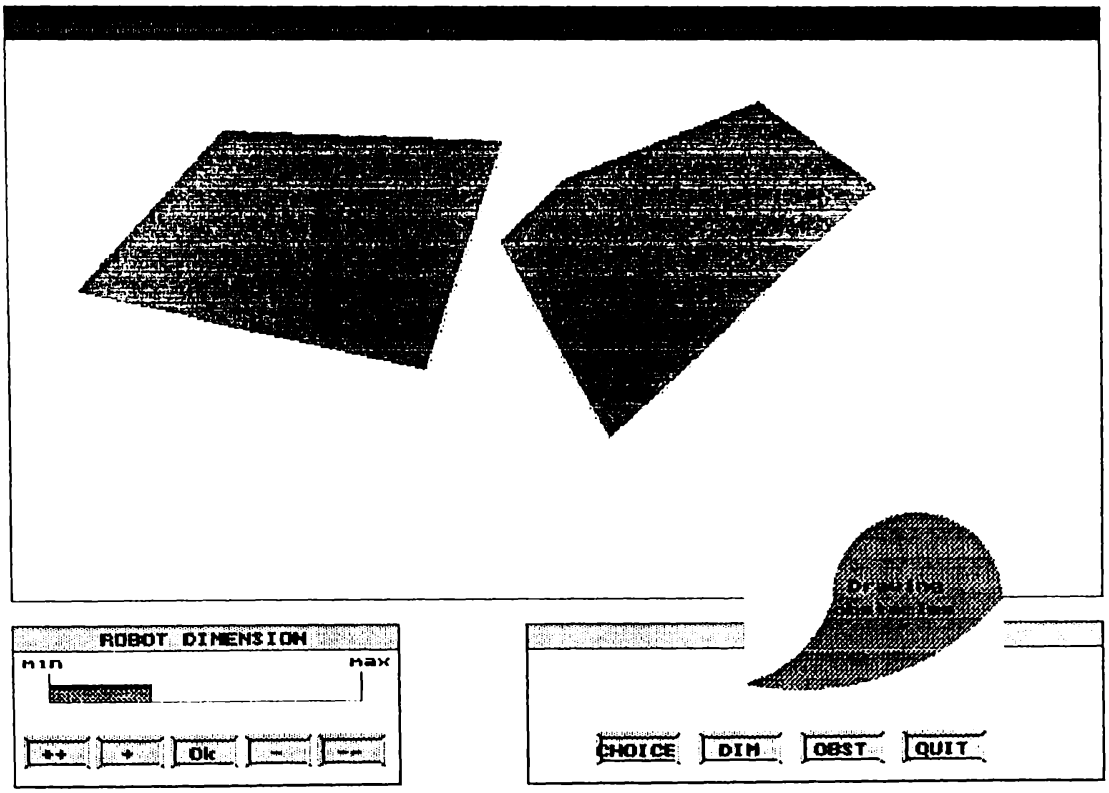


Fig.7.15. Spațiul de lucru populat cu obstacole poligonale

BUPT

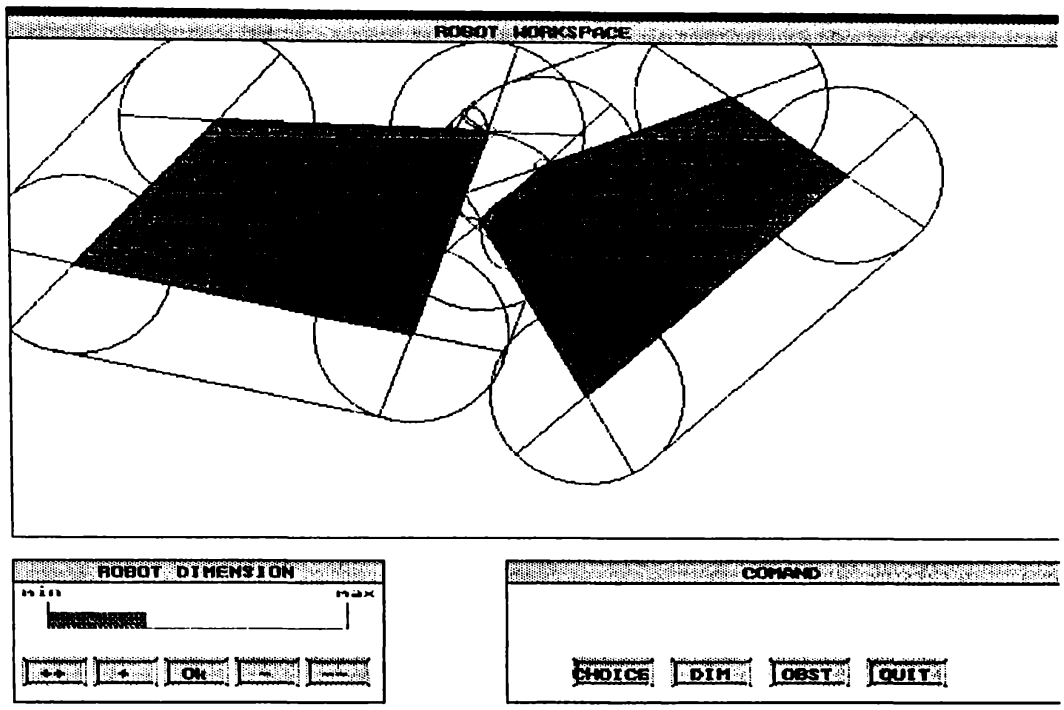


Fig.7.16. Reprezentarea curbelor critice

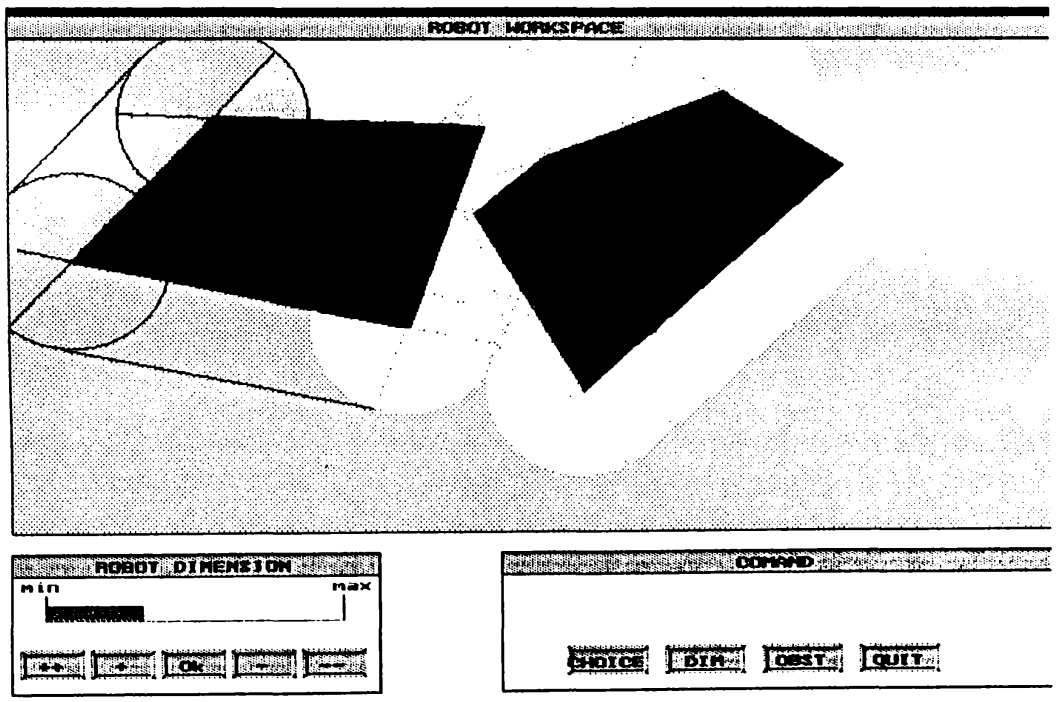


Fig.7.17. Cercetarea şi memorarea regiunilor necritice

BUPT

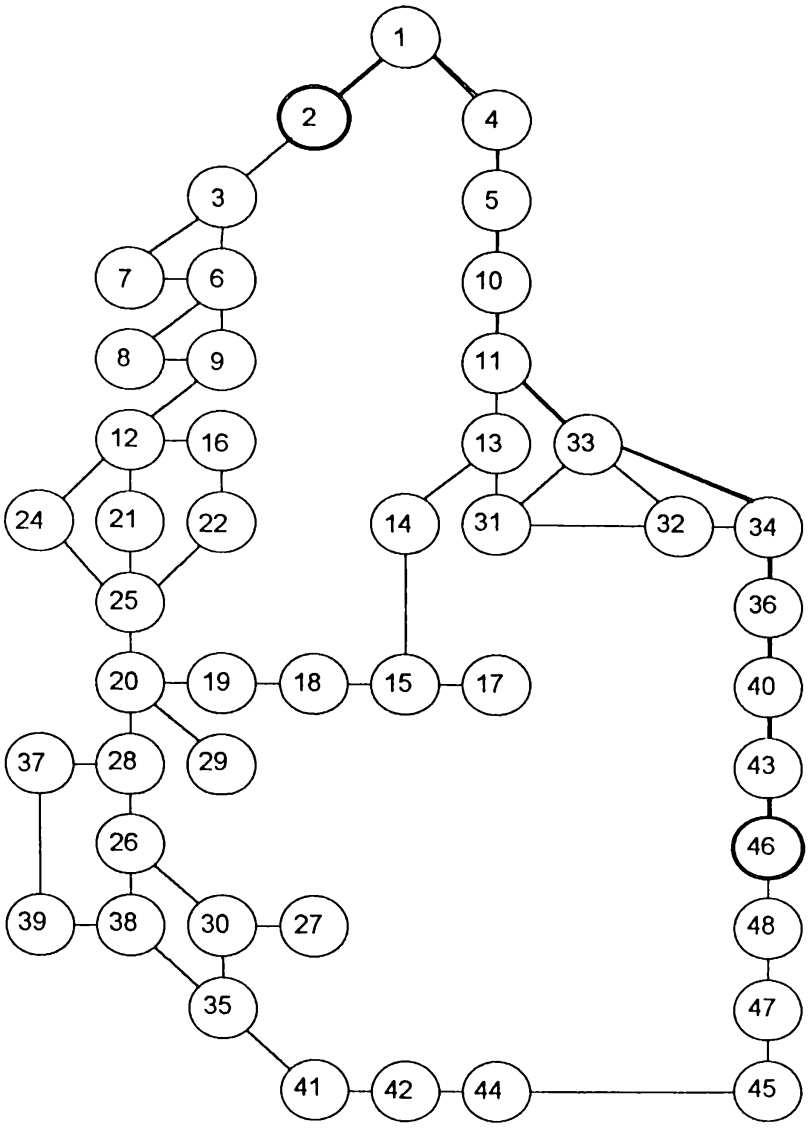


Fig.7.18. Graful de conexiune. Cercetează graful între $\mathbf{C}_{\text{inițial}}$ şi $\mathbf{C}_{\text{final}}$

Programul permite reprezentarea spațiului de lucru al robotului în două cazuri: mărginit şi nemărginit. Configurația spațiului de lucru şi dimensiunea robotului pot fi modificate după caz. Construcția curbelor critice permite descompunerea configurațiilor robotului în regiuni necritice. Geometria acestor celule este destul de simplă pentru a putea calcula cu uşurință un drum între oricare două configurații ale robotului. De asemenea, nu este dificil de testat relația de adiacență dintre oricare două celule şi de determinat un drum care traversează porțiunea frontierei ce aparține celor două celule adiacente. Complexitatea algoritmului depinde semnificativ de implementarea unor anumiți paşi, în particular 6 şi 8. Totuşi metoda este relativ eficientă şi uşor de implementat.

BUPT

7.7. CONCLUZII

Planificarea traiectoriilor în prezența obstacolelor este o problemă de mare actualitate în Robotică, de o importanță majoră în crearea roboților autonomi. Robotul trebuie să fie capabil să-şi planifice propriile mişcări, să decidă automat ce mişcări să execute pentru a realiza o sarcină, în funcție de aranjamentul momentan al obiectelor din spațiul de lucru. Planificarea mişcării unui robot poate fi realizată off-line, sau simultan cu generarea mişcării, on-line.

Există un mare număr de metode de planificare, dar nu toate dintre acestea rezolvă problema în întreaga sa generalitate. Planificarea mişcării unui robot prezintă o varietate neașteptată de aspecte dificile de calcul. În acest capitol, au fost prezentate aspectele semnificative ale unei probleme de planificare, precum şi unele metode de planificare, globale şi locale, dar numai în cadrul problemei de bază a planificării.

Una din metodele de modelare a spațiului de lucru, frecvent utilizată, este metoda grilei. Aceasta este simplu de aplicat, iar modelarea obstacolelor prin suprafețe dreptunghiulare nu influențează semnificativ precizia algoritmului. Aşa cum a fost prezentată, metoda de modelare a spațiului de lucru, completată cu tehnici de cercetare în graf, constituie de fapt o metodă de planificare. De asemenea, modelarea spațiului de lucru prin metoda grilei poate fi considerată ca o descompunere convex poligonală şi deci, continuată cu una din metodele descompunerii celulare exacte. Metoda grilei poate fi aplicată şi în cazul unui spațiu de lucru tridimensional, modelând obstacolele prin volume paralelipipedice.

Indiferent de metoda de planificare aplicată, obstacolele din spațiul de lucru trebuie considerate cu dimensiuni majorate, atât pentru siguranța mişcării robotului, cât şi pentru simplificarea algoritmului de planificare, determinarea frontierelor C-obstacolelor permițând considerarea robotuuli ca un punct material.

Algoritmul de calcul al frontierelor C-obstacolelor, pentru cazul bidimensional, corespunzător translației robotului, poate fi aplicat pentru orice orientare a acestuia. Algoritmul prezentat poate fi incorporat în orice program care rezolvă o problemă de planificare a mişcării. De asemenea, algoritmul poate fi aplicat şi pentru determinarea frontierelor C-obstacolelor în cazul în care orientarea robotului se modifică în timpul deplasării, pentru un $\Delta\theta$ constant.

Metoda grafului vizibilității, mai ales în varianta de graful vizibilității redus, permite determinarea unui drum, într-un spațiu de lucru bidimensional, sub forma unei linii poligonale ce conectează configurația inițială de cea finală. Criteriul de optimizare aplicabil în cazul acestei metode constă în minimizarea drumului parcurs. Metoda grafului vizibilității este simplu de aplicat într-un spațiu de lucru bidimensional, când robotul execută o mişcare de translație, dar nu şi în cazul când robotul execută o mişcare plană. Metoda grafului vizibilității poate fi aplicată şi într-un spațiu tridimensional, dar drumul generat poate să nu fie cel mai scurt.

Programul realizat, bazat pe algoritmul metodei grafului vizibilității permite aplicarea metodei şi în cazul în care obstacolele din spațiul de lucru sunt poligoane oarecare, formate din segmente şi arce.

Metoda drumului liber poate fi aplicată într-un spațiu de lucru bidimensional, fiind recomandată în cazul în care spațiul de lucru nu este prea dens populat cu obstacole. Algoritmul metodei este indicat să fie aplicat off-line, deoarece există riscul de a nu putea determina un drum liber între configurația inițială şi cea finală. Timpul de calcul necesar

BUPT

aplicării algoritmului este relativ mare datorită, în primul rând, determinării intervalului orientărilor libere ale robotului, absolut necesar în timpul deplasării acestuia pe axa drumului liber. În cazul în care există mai multe drumuri libere posibile, criteriul de optimizare constă în minimizarea lungimii drumului.

Metoda drumului liber este recomandată pentru un spațiu de lucru bidimensional, dar poate fi extinsă şi pentru cazul tridimensional, când un drum liber este determinat de două fețe ce corespund la două obstacole vecine.

Metoda descompunerii celulare exacte, bazată pe descompunerea spațiului liber în regiuni necritice, delimitate de o rețea de curbe critice, a fost aplicată pentru cazul când robotul este modelat ca un segment de dreaptă care execută o mişcare plană. Testarea relației de adiacență dintre celule conduce la obținerea grafului de conexiune, prin cercetarea căruia se determină drumul între configurația inițială şi cea finală.

Un alt tip de descompunere celulară exactă constă în descompunerea convex poligonală. Metoda este frecvent utilizată deoarece realizează un optim între drumul cel mai sigur şi drumul minim.

Avantajul metodei de planificare bazată pe descompunerea celulară exactă constă în existența canalului format din celulele adiacente. Conceptul de canal este mai puțin restrictiv decât cel de drum şi poate asigura robotului informații utile pentru tratarea restricțiilor dinamice şi a obstacolelor neaşteptate în timpul execuției.

Metoda descompunerii convex poligonale poate fi extinsă şi în cazul tridimensional.

Programul realizat descompune spațiul liber în regiuni necritice, testează relația de adiacență dintre celule, reprezintă şi cercetează graful de conexiune. Aplicarea programului descompunerii celulare exacte asigură o precizie bună, dar necesită un timp de calcul destul de mare.

Conținutul acestui capitol tratează aspectele şi metodele principale de planificare a mişcării în cadrul problemei de bază şi deschide totodată noi direcții de cercetare, rezultate din extensiile acestei probleme. Astfel, în continuare, vor trebui abordate aspecte ca:

- includerea mai multor obiecte mobile în spațiul de lucru;
- mai mulți roboți, şi deci sincronizarea mişcării acestora;
- considerarea structurii mecanice a roboților (elemente şi cuple cinematice);
- restricții suplimentare impuse mişcării robotului (olonome, neolonome);
- incertitudinea controlului şi a modelului geometric al robotului.

De asemenea, toate cazurile de planificare a mişcării în care factorul timp este indispensabil trebuie abordate în cadrul problemei dinamice a planificării.

BUPT

8. PLANIFICAREA MIŞCĂRII ROBOŢILOR PRIN METODELE CÂMPULUI POTENŢIAL

8.1. INTRODUCERE

Câmpul potenţial artificial a fost gândit ca o unealtă universală pentru ghidarea mişcării, atât a roboţilor manipulatori cu *n* elemente, cât şi a roboţilor mobili [G4], [H4], [H8], [K5], [K6], [K7], [K10], [K14], [K22], [R12].

Ideea de bază a metodelor câmpului potenţial artificial este de a construi un câmp potenţial cu un gradient care acţionează "atractiv" spre configuraţia finală (minim global) şi "repulsiv" dinspre obstacole (maxim local).

S-au studiat diferite abordări care utilizează câmpul potenţial artificial pentru controlul mişcării unui robot.

• *Implementarea potenţialului artificial în dinamica sistemului prin intermediul controlului.* Controlul torsorului a fost aplicat coliniar cu gradientul câmpului potenţial, implementând relaţiile dinamice dintre robot şi mediul său [H8]. Această abordare nu conduce la urmărirea riguroasă a gradientului, dar garantează totuşi siguranţa evitării coliziunilor pentru potenţialele care tind la infinit la frontierele obstacolelor [R12].

• *Considerarea dinamicii robotului şi restricţiilor servomotoarelor în proiectarea câmpului potenţial.* Conceptul de "câmp potenţial generalizat" [K22] consideră prioritar timpul necesar atingerii configuraţiei finale faţă de distanţa respectivă. În această lucrare, pentru proiectarea controlului, un integrator dublu modelează robotul mobil, luând în considerare saturarea servomotoarelor, dar nu şi dinamica robotului.

• *Utilizarea controlului feedback pentru urmărirea gradientului câmpului potenţial.* În [K51] s-a propus a strategie de liniarizare cu feedback, care necesită cunoaşterea exactă a dinamicii robotului. Pentru extinderea strategiei controlului PD la sistemele mecanice neliniare, în [R12] s-a propus proiectarea unei funcţii de navigare şi urmărirea gradientului său care depinde implicit de inerţia sistemului. De asemenea, pentru proiectarea funcţiei de navigare s-a folosit şi o strategie care utilizează energia totală a sistemului ca bază pentru control [R11].

Cercetările şi rezultatele obţinute în folosirea funcţiilor potenţial în aplicaţii ale roboticii au fost sintetizate în [R11]. Lucrările publicate în acest domeniu pot fi împărţite în trei categorii:

• Lucrări în care se studiază minimurile locale nedorite: în [K22] se folosesc potenţiale generalizate; în [C18] se abordează problema cu Laplacian; în [B3] se folosesc tehnici de simulare.

• Lucrări în care se consideră funcţiile potenţial ca "ajutoare" în planificarea traiectoriilor: în [B4] se foloseşte tehnica "valley tracking"; în [W5] se foloseşte o procedură de optimizare a traiectoriilor.

BUPT

• Lucrări în care se extinde folosirea funcțiilor potențial pentru situații care variază în timp: în [N1] se studiază evitarea obstacolelor mobile; în [K5] se folosește o tehnică de evitare a obstacolelor în timp real.

Nici una din lucrările de mai sus nu oferă un algoritm bazat pe funcții potențial exacte care să garanteze atingerea configurației finale în multe implementări practice. În ultimele lucrări publicate se folosesc cu succes funcțiile de navigare. Pentru construirea funcțiilor de navigare se folosesc stategii heuristice, pentru o anumită clasă de corpuri rigide, sau pentru o problemă în ansamblu. Metoda propusă în [G5] garantează urmărirea exactă a liniilor de gradient ale câmpului potențial artificial.

8.2. PRINCIPIUL METODELOR CÂMPULUI POTENȚIAL

Majoritatea metodelor de planificare a mișcării folosesc reprezentarea spațiului liber al robotului sub forma unui graf sau arbore, care este apoi supus unei tehnici de căutare a unui drum.

Metodele câmpului potențial pornesc de la o idee diferită, și anume se reprezintă robotul ca un punct în spațiul configurațiilor, respectiv ca o particulă aflată sub influența unui câmp potențial artificial U, al cărui variații locale vor reflecta "structura" spațiului liber.

Funcția potențial este în mod normal, dar nu obligatoriu, definită ca o sumă a două potențiale în spațiul liber: **atractiv** - care atrage robotul spre configurația finală, și respectiv **repulsiv** - care respinge robotul din vecinătatea obstacolelor. Planificarea mișcării se face apoi în mod iterativ. La fiecare pas al iterației, forța artificială $\vec{F}(\mathbf{C}) = -\vec{\nabla}U(\mathbf{C})$ indusă de către funcția potențial la configurația curentă, este privită drept cea mai probabilă direcție de mișcare și astfel generarea mișcării continuă de-a lungul acestei direcții cu un anumit increment.

Metoda câmpului potențial a fost dezvoltată la început ca o metodă de evitare a coliziunilor, on-line, aplicabilă atunci când robotul nu are un model aprioric al obstacolelor, dar le sesizează în timpul execuției mișcării. S-a pus accent mai ales pe eficiența în timp real a metodei decât pe garantarea atingerii configurației finale. În particular, deoarece o metodă on-line se implementează în general ca o procedură de optimizare cu descendent rapid, aceasta se poate opri din execuție la atingerea unui minim local al funcției potențial, altul decât cel corespunzător configurației finale a robotului. În acest caz, ideea de a utiliza câmpul potențial în rezolvarea planificării mișcării trebuie combinată cu folosirea unor tehnici de căutare în grafuri. Astfel, utilizând un model matematic anterior cunoscut al spațiului de lucru, metoda se transformă într-o metodă sistematică de planificare a mișcării. Chiar dacă se utilizează tehnici de căutare în grafuri, minimurile locale rămân o cauză importantă a ineficienței metodelor de câmp potențial, deci lucrul cu minimuri locale este problema majoră care trebuie rezolvată în determinarea unei traiectorii de mișcare utilizând această metodă.

Problema minimurilor locale poate fi abordată la două niveluri diferite ale execuției metodei câmpului potențial: (1) la definirea funcției potențial, prin încercarea de a preciza o funcție fără nici unul sau cu puține minimuri locale, sau (2), în implementarea algoritmului de căutare a drumului, prin folosirea unor tehnici adecvate pentru a elimina minimurile locale.

Metodele de câmp potențial sunt considerate în literatura de specialitate ca și "metode locale". Acest lucru provine din faptul că majoritatea funcțiilor potențiale sunt definite în așa

BUPT

fel încât valorile lor, în orice configurație, nu depind de distribuția sau de forma C-obstacolelor, decât în limita unei vecinătăți în jurul configurației acestora.

Majoritatea metodelor de planificare bazate pe câmpul potențial au o esență puternic experimentală şi de obicei sunt incomplete. De exemplu, ele pot eşua în determinarea unui drum, chiar dacă acesta există, blocându-se într-un minim local. Dar, în compensarea acestui dezavantaj, unele dintre aceste metode sunt deosebit de rapide într-o arie largă de situații. Puterea metodei constă în faptul că utilizând puține noțiuni inginereşti face posibilă construirea unor algoritmi de căutare, care sunt în acelaşi timp destul de eficienți şi rezonabil de siguri. Acest lucru explică de ce metodele de acest tip sunt tot mai des utilizate în implementarea practică a planificării mişcării.

În paragrafele următoare se va da o definiție clasică a funcției potențial şi a câmpului indus al forțelor. Se va considera mai întâi cazul când robotul poate efectua numai mişcare de translație (8.2.1); apoi, cazul când acesta poate efectua şi mişcare de rotație (8.2.2); în (8.3) vor fi descrise trei tehnici simple de utilizare a metodei câmpului potențial: în adâncime, pe nivel şi variațional; în următorul paragraf (8.4), se va introduce noțiunea de funcție de navigație (asociată cu potențialul liber de minimuri locale) şi se vor defini funcțiile locale care pot genera mai puține sau mai mici minimuri locale decât cele descrise în (8.2.1) şi (8.2.2).

Majoritatea funcțiilor potențial propuse în literatură se bazează pe următoarea idee generală: robotul trebuie să fie atras spre configurația sa finală în timp ce este respins de către obstacole. Câmpul forțelor artificiale $\vec{F}(\mathbf{C})$ în SC este produs de către o funcție potențial diferențiabilă U: $SC_{liber} \rightarrow \mathbf{R}$, unde $\vec{F}(\mathbf{C}) = -\vec{\nabla}U(\mathbf{C})$. Pentru ca robotul să fie atras spre configurația finală în timp ce este respins de către obstacole, U se construieşte ca sumă a două funcții potențial elementare:

$$U(\mathbf{C}) = U_{atr}(\mathbf{C}) + U_{rep}(\mathbf{C}), \qquad (8.1)$$

unde: U_{atr} este potențialul atractiv asociat configurației finale, independent de regiunea C-obstacolelor, iar U_{rep} este potențialul repulsiv asociat regiunii de C-obstacolelor, independent de configurația finală. Cu aceste convenții, forța $\vec{F}$ se poate scrie ca sumă a doi vectori, care sunt denumiți forță atractivă, respectiv repulsivă:

$$\vec{F}_{atr} = -\vec{\nabla}U_{atr} \quad \text{şi} \quad \vec{F}_{rep} = -\vec{\nabla}U_{rep} \qquad (8.2)$$

În acest capitol se va folosi notarea forțelor, respectiv a mărimilor vectorilae cu semnul de vector, pentru a sublinia semnificația acestora în aplicarea metodelor câmpului potențial.

8.2.1. CÂMPUL POTENȚIAL ÎN CAZUL TRANSLAȚIEI

În acest paragraf vor fi definite funcțiile potențial, în cazul în care robotul A translatează liber, având o orientare fixată, în spațiul de lucru $W=\mathbf{R}^N$, unde N=2 sau 3. În paragraful următor, această definiție va fi adaptată la cazul când robotul *A* poate efectua atât mişcare de translație cât şi de rotație. Funcția definită în continuare a fost deja utilizată în câteva sisteme implementate fizic.

BUPT

8.2.1.1. FUNCȚIA POTENȚIAL ATRACTIV

Funcția potențial atractiv U_{atr} poate fi definită simplu, ca o parabolă, de forma:

$$U_{atr}(\mathbf{C}) = \frac{1}{2} \cdot \xi \cdot d_{final}^2(\mathbf{C}) \tag{8.3}$$

unde: ξ este un factor scalar pozitiv, iar $d_{final}(\mathbf{C})$ este distanța euclidiană $\left\| \vec{\mathbf{C}} - \vec{\mathbf{C}}_{final} \right\|$.

Funcția U_{atr} este pozitivă sau nulă şi atinge minimul său în configurația $\mathbf{C}_{final}$, unde $U_{atr}(\mathbf{C}_{final}) = 0$. Funcția d_{final} este diferențiabilă în întreg spațiul SC. La fiecare configurație $\mathbf{C}$, forța atractivă $\vec{F}_{atr}$ derivată din U_{atr} este:

$$\vec{F}_{atr}(\mathbf{C}) = -\vec{\nabla} U_{atr}(\mathbf{C}) = -\xi\, d_{final}(\mathbf{C})\, \vec{\nabla} d_{final}(\mathbf{C}) = -\xi \left(\vec{\mathbf{C}} - \vec{\mathbf{C}}_{final} \right) \tag{8.4}$$

Când este folosită pentru evitarea coliziunilor, forma parabolică a potențialului atractiv are caracteristici stabilizatoare foarte bune, deoarece generează o forță $\vec{F}_{atr}$ care converge în mod liniar către zero, atunci când configurația robotului se apropie de configurația finală. Potentialul atractiv sub formă parabolică generează o forță care creşte odată cu distanța către configurația finală şi tinde la infinit când $d_{final}(\mathbf{C}) \to \infty$. Deoarece forța este proporțională cu parametrii de configurație, sistemul poate avea o comportare de oscilator armonic. Pentru o mai bună stabilitate a sistemului se propune să se adauge un termen de amortizare (forță disipativă proporțională cu viteza):

$$\vec{F}_{atr}(\mathbf{C}) = -\xi \left(\vec{\mathbf{C}} - \vec{\mathbf{C}}_{final} \right) - \psi \dot{\mathbf{C}} \tag{8.5}$$

Aceasta se poate traduce printr-un control în viteză:

$$\vec{F}_{atr}(\mathbf{C}) = -\psi \left(\dot{\mathbf{C}} - \tilde{\mathbf{C}} \right) \tag{8.6}$$

unde: $\tilde{\mathbf{C}} = \dfrac{\xi}{\psi \left(\mathbf{C}_{final} - \mathbf{C} \right)}$ este valoarea nominală a derivatei parametrilor de configurație.

Faptul că forța atractivă devine arbitrar de mare pe măsură ce mobilul se îndepărtează de obstacol (se apropie de $\mathbf{C}_{final}$) nu convine. Pentru a remedia acest aspect se pot trunchia vitezele de comandă. Dacă se folosesc parametri de configurație normalizați, se împarte fiecare parametru cu viteza sa maximă:

$$\vec{F}_{atr}(\mathbf{C}) = -\psi \left(\dot{\mathbf{C}} - \nu \tilde{\mathbf{C}} \right) \tag{8.7}$$

unde $\nu = \min\left(1, \dfrac{1}{\left\| \tilde{\mathbf{C}} \right\|} \right)$.

În practică, potențialul atractiv U_{atr} se poate defini şi sub forma conică:

$$U_{atr}(\mathbf{C}) = \xi\, d_{final}(\mathbf{C}) \tag{8.8}$$

caz în care forța atractivă devine:

$$\vec{F}_{atr}(\mathbf{C}) = -\xi \cdot \vec{\nabla} d_{final}(\mathbf{C}) = -\frac{\xi (\vec{\mathbf{C}} - \vec{\mathbf{C}}_{final})}{\left\| \vec{\mathbf{C}} - \vec{\mathbf{C}}_{final} \right\|} \tag{8.9}$$

Amplitudinea forței $\vec{F}_{atr}(\mathbf{C})$ este constantă în spațiul SC, cu excepția configurației $\mathbf{C}$ unde $U_{atr}(\mathbf{C})$ este singulară. Deoarece amplitudinea forței nu tinde către zero pentru $\mathbf{C} \to \mathbf{C}_{final}$, potențialul sub formă conică nu are caracteristicile stabilizatoare ale formei parabolice. O

BUPT

metodă de a combina avantajele ambelor forme de reprezentare a potențialului atractiv (parabolică şi conică) este aceea de a defini U_{atr} sub formă parabolică în limita unei distanțe față de configurația finală şi sub formă conică după această distanță, având în punctul de joncțiune a celor două regiuni o derivată definită. În continuare se va folosi în mod exclusiv potențialul atractiv sub formă parabolică, ori de câte ori se va construi câmpul potențial ca sumă a celor două componente, atractiv şi repulsiv.

8.2.1.2. FUNCȚIA POTENȚIAL REPULSIV

Ideea principală în cazul definirii potențialului repulsiv este aceea de a crea o barieră de potențial în jurul C-obstacolelor, care să nu poată fi traversată de către configurația robotului. În plus, este de dorit ca potențialul repulsiv să nu afecteze mişcarea robotului atunci când acesta este suficient de depărtat față de obstacole. O metodă de a obține aceste restricții este de a defini potențialul repulsiv după cum urmează:

$$U_{rep}(\mathbf{C}) = \begin{cases} \frac{1}{2}\cdot\eta\cdot\left(\frac{1}{d(\mathbf{C})}-\frac{1}{d_0}\right)^2 & d(\mathbf{C}) \le d_0 \\ 0 & d(\mathbf{C}) > d_0 \end{cases} \qquad (8.10)$$

unde: η este un factor scalar pozitiv, $d(\mathbf{C})$ reprezintă distanța dintre configurația curentă $\mathbf{C}$ şi regiunea C-obstacolelor: $d(\mathbf{C}) = \min_{C' \in SCB} \|\mathbf{C} - \mathbf{C}'\|$, iar d_0 este o constantă pozitivă numită *distanța de influență a C-obstacolelor.*

Funcția U_{rep} este pozitivă sau nulă: tinde la infinit când $\mathbf{C}$ se apropie de regiunea C-obstacolelor şi este nulă atunci când distanța dintre configurația curentă a robotului şi regiunea C-obstacolelor este mai mare decât d_0.

Dacă SCB este o regiune convexă ce are frontiera diferențiabilă pe porțiuni, atunci U_{rep} este diferențiabil în spațiul liber SC_{liber}. Forța repulsivă artificială rezultată din U_{rep}:

$$\vec{F}_{rep}(\mathbf{C}) = -\vec{\nabla}U_{rep}(\mathbf{C}) = \begin{cases} \eta\cdot\left(\frac{1}{d(\mathbf{C})}-\frac{1}{d_0}\right)\cdot\frac{1}{d^2(\mathbf{C})}\vec{\nabla}d(\mathbf{C}) & d(\mathbf{C}) \le d_0 \\ 0 & d(\mathbf{C}) > d_0 \end{cases} \qquad (8.11)$$

Fie $\mathbf{C}_e$ o configurație unică în SCB care este cea mai apropiată de $\mathbf{C}$, adică $\|\mathbf{C} - \mathbf{C}_e\| = d(\mathbf{C})$. Gradientul $-\vec{\nabla} d(\mathbf{C})$ este un vector orientat în afara frontierei SCB şi are ca suport linia ce unește $\mathbf{C}_e$ cu $\mathbf{C}$. Dacă se renunță la presupunerea (nerealistă) că SCB este convex, $d(\mathbf{C})$ rămâne diferențiabilă în orice configurație din spațiul SC_{liber}, cu excepția acelor configurații $\mathbf{C}$ pentru care există anumite $\mathbf{C}_e \in$ SCB care verifică relația $\|\mathbf{C}-\mathbf{C}_e\| = d(C)$. Aceste configurații particulare formează o mulțime de configurații în general (N-1)-dimensională. Câmpul forței $\vec{F}_{rep}$ este definit în ambele părți ale acestei mulțimi, dar cu valori diferite ale vectorilor de orientare. Acest lucru poate duce la producerea unor drumuri care oscilează între cele două părți ale mulțimii.

Un mod de a elimina această dificultate constă în a descompune SCB în componente convexe SCB_k, $k = 1, ..., r$ (cu posibilitatea de suprapunere) şi de a asocia un potențial repulsiv fiecărei componente astfel obținute. În acest fel, potențialul repulsiv este suma potențialelor repulsive create de către fiecare C-obstacol individual, deci:

BUPT

$$U_{rep}(\mathbf{C}) = \sum_{k=1}^{n} U_{SCB_k}(\mathbf{C}), \tag{8.12}$$

unde:

$$U_{SCB_k}(\mathbf{C}) = \begin{cases} \frac{1}{2} \cdot \eta \cdot \left(\frac{1}{d_k(\mathbf{C})} - \frac{1}{d_0} \right)^2 & d_k(\mathbf{C}) \le d_0 \\ 0 & d_k(\mathbf{C}) > d_0 \end{cases} \tag{8.13}$$

unde $d_k(\mathbf{C})$ reprezintă distanța de la $\mathbf{C}$ la SCB_k.

Forța artificială repulsivă care rezultă din acest potențial este:

$$\vec{F}_{rep}(C) = \sum_{k=1}^{r} \vec{F}_{SCB_k}(C) \tag{8.14}$$

$$\vec{F}_{SCB_k}(\mathbf{C}) = -\vec{\nabla} U_{SCB_k}(\mathbf{C}) \tag{8.15}$$

Regiunile convexe SCB_k pot fi construite prin descompunerea lui A şi a obstacolelor din spațiul de lucru în componente convexe şi apoi prin calcularea C-obstacolelor corespunzătoare fiecărei perechi constând dintr-o componentă a lui A şi o componentă a unui obstacol. Descompunerea regiunii C-obstacolelor are particularitatea că mai multe componente mici, care sunt apropiate una față de cealaltă, produc o forță repulsivă mai puternică decât cea produsă de o singură componentă mai mare. O metodă empirică de a trata acest dezavantaj este de a asocia fiecărui potențial individual un coeficient care depinde de dimensiunea C-obstacolului care este descompus.

În definițiile anterioare pentru $U_{rep}(\mathbf{C})$ s-a folosit un factor scalar η unic şi o distanță de influență d_0 de asemenea unică. Dar nu prezintă nici o dificultate folosirea unor parametri η şi d_0 diferiți pentru submulțimi variate de C-obstacole. În particular, dacă configurația finală $\mathbf{C}_{final}$ este apropiată de SCB, distanța de influență d_0 ar trebui aleasă la o valoare mai mică decât distanța de la $\mathbf{C}$ la SCB, astfel încât robotul să nu fie împiedicat să atingă C_{final} din cauza câmpului potențial repulsiv. Dacă potențialul repulsiv este nenul la $\mathbf{C}_{final}$, atunci, în general, minimul global al funcției potențial totale este diferit de C_{final}. Distanța d_0 poate fi aleasă la o valoare mai mare decât cea corespunzătoare altor C-obstacole. În plus, dacă un obstacol specific este cunoscut a fi mai "periculos" decât celelalte, este mai bine să fie utilizate valori mai mari ale parametrilor η şi d_0 pentru C-obstacolul corespunzător.

Potențialul repulsiv $U_{rep}(\mathbf{C})$ şi câmpul de forțe asociat $F_{rep}(\mathbf{C})$ pot fi calculate uşor atunci când regiunea C-obstacolelor este poligonală sau poliedrică, cât şi pentru alte forme particulare (sferice). În majoritatea cazurilor însă se întâmplă să nu existe un algoritm care să continue determinarea căii după blocarea într-un minim local. Când regiunea C-obstacolelor poate fi descompusă în componente convexe SCB_k, fiecare componentă fiind descrisă de un set de inegalități $f_j^{(k)}(\mathbf{C}) \le 0$, $j = 1,2,...$, problema calculului lui $d_k(\mathbf{C})$ devine o problemă de optimizare, care poate fi rezolvată utilizând tehnici iterative. În spațiul de lucru bi-dimensional metoda câmpului potențial prezentată anterior poate fi asociată cu metoda retractării. Formele potențialului repulsiv formează diagrama Voronoï a spațiului liber, iar potențialul atractiv se comportă ca şi o funcție heuristică ce ghidează căutarea drumului în diagramă. Dar, deoarece cele două potențiale sunt însumate într-o funcție totală, metoda câmpului potențial nu restrânge mişcarea robotului doar de-a lungul drumurilor standardizate, aşa cum sunt formate ele de arcele din diagrama Voronoï.

BUPT

8.2.2. CÂMPUL POTENȚIAL ÎN CAZUL GENERAL

O generalizare a evoluției robotului, față de paragraful 8.2.1, în cazul în care spațiul configurațiilor SC este de forma $\mathbf{R}^N$ x SO(N), impune redefinirea funcțiilor potențial.

8.2.2.1. EXTENSII ALE NOȚIUNILOR ANTERIOARE

Noțiunile introduse în paragraful anterior pot fi extinse pentru cazul general când SC este de forma R^N x SO(N). Definiția vectorului gradient negativ $-\vec{\nabla}U$ necesită însă specificarea unui produs scalar în spațiul tangent lui SC [L5]. Fie (U, φ) o aplicație a lui SC, unde:

$$\varphi: \mathbf{C} \in U \rightarrow (x_1(\mathbf{C}),...,x_m(\mathbf{C})) \in R^m \tag{8.16}$$

Fie $\vec{t}$ un vector al spațiului tangent T_C (SC) și $[\vec{t}]_\beta$ un m-vector al componentelor lui $\vec{t}$ în baza indusă de aplicația (U, φ) și fie $\mathbf{C}_\alpha\vec{t}$ configurația definită de φ^{-1} $((x_1(\mathbf{C}), ..., x_m(\mathbf{C}))+\alpha[\vec{t}]_\beta)$. Se determină derivata direcțională a lui U de-a lungul direcției $\vec{t}$ dată de **C** prin $D_{\vec{t}}U(\mathbf{C})$. Aceasta este definită prin următoarea funcție liniară în T_C (SC):

$$D_{\vec{t}}U(\mathbf{C}) = \lim_{\alpha\to 0}\frac{U(\mathbf{C}_{\alpha\vec{t}}) - U(\mathbf{C})}{\alpha} \tag{8.17}$$

Prin definiție, vectorul gradient $\vec{\nabla}U(\mathbf{C})$ este acel vector din T_C (SC) pentru care:

$$D_{\vec{t}}U(\mathbf{C}) = \vec{\nabla}U \cdot \vec{t}\ , \quad \forall\, \vec{t} \in T_C(SC) \tag{8.18}$$

Se observă similitudinile cu definiția vectorului forță: în timp ce derivata direcțională a lui U (respectiv lucrul mecanic al forței) este independentă de alegerea produsului intern, reprezentarea vectorului gradient (respectiv a vectorului forță) depinde de această alegere.

Se alege produsul în T_C (SC) astfel încât β să fie o bază ortonormală. Dacă t_i, i=1,...,m sunt componentele vectorului $\vec{t}$ în baza β, atunci:

$$U(\mathbf{C}) = U(x_1(\mathbf{C}),...,x_m(\mathbf{C})), \text{ și:} \tag{8.19}$$

$$D_{\vec{t}}U(\mathbf{C}) = \sum_{i=1}^{m}\frac{\partial U}{\partial x_i}(x_1(\mathbf{C}),...,x_m(\mathbf{C}))t_i \tag{8.20}$$

$$\Rightarrow \left[\vec{\nabla}U(\mathbf{C})\right]_\beta = \left(\frac{\partial U}{\partial x_1}\cdots\frac{\partial U}{\partial x_m}\right)^T\Big|_{(x_1(\mathbf{C}),...,x_m(\mathbf{C}))} \tag{8.21}$$

relație ce coincide cu definiția convențională a vectorului gradient în spațiul euclidian $\mathbf{R}^m$.

Fie $\vec{\nabla}_1U(C)$ și $\vec{\nabla}_2U(C)$ notații ale vectorilor gradienți ai lui U pentru două produse scalare. Fie β_1 și β_2 două baze ale lui T_C(SC) astfel încât β_1 și β_2 sunt baze ortonormale când este folosit produsul respectiv. Fie **J** o matrice (m x m) a aplicațiilor liniare care reprezintă relația dintre m-vectorul $[\vec{t}]_1$ component al vectorului $\vec{t} \in T_C$ (SC) în β_1 și m-vectorul $[\vec{t}]_2$ component al lui $\vec{t}$ în β_2. Pentru orice $\vec{t} \in T_C(SC)$:

$$\left[\vec{\nabla}_1U(\mathbf{C})\right]_1^T\left[\vec{t}\right]_1 = \left[\vec{\nabla}_2U(\mathbf{C})\right]_2^T\left[\vec{t}\right]_2 = \left[\vec{\nabla}_2U(\mathbf{C})\right]_2^T \cdot \mathbf{J} \cdot \left[\vec{t}\right]_1 \tag{8.22}$$

$$\left[\vec{\nabla}_1U(\mathbf{C})\right]_1 = \mathbf{J}^T \cdot \left[\vec{\nabla}_2U(\mathbf{C})\right]_2 \tag{8.23}$$

BUPT

Fiind dată o metrică d în SC, funcția potențial atractiv și câmpul de forțe asociat pot fi definite ca în paragraful anterior prin:

$$U_{atr}(\mathbf{C}) = 1/2\ \xi\ d_{final}^{2}(\mathbf{C}),\ \text{unde}\ d_{final}(\mathbf{C}) = d(\mathbf{C}, \mathbf{C}_{final}) \tag{8.24}$$

$$\vec{F}_{atr}(\mathbf{C}) = -\vec{\nabla} U_{atr}(\mathbf{C}) = -\xi\ d_{final}(\mathbf{C})\ \vec{\nabla} d_{final}(\mathbf{C}) \tag{8.25}$$

În mod analog, potențialul repulsiv și câmpul de forțe asociat acestuia sunt:

$$U_{rep}(C) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{d(C)} - \frac{1}{d_0}\right)^2 & d(C) \le d_0 \\ 0 & d(C) > d_0 \end{cases} \tag{8.26}$$

$$\vec{F}_{rep}(\mathbf{C}) = -\vec{\nabla} U_{rep}(C) = \begin{cases} \eta\left(\frac{1}{d(\mathbf{C})} - \frac{1}{d_0}\right)\cdot\frac{1}{d_0^2}\cdot\vec{\nabla} d(\mathbf{C}) & d(\mathbf{C}) \le d_0 \\ 0 & d(\mathbf{C}) > d_0 \end{cases} \tag{8.27}$$

unde $d(\mathbf{C}) = \min_{C' \in SCB} d(\mathbf{C}, \mathbf{C}')$.

Aceste extinderi de noțiuni au ca rezultat dificultăți de calcul suplimentare. Astfel, pentru o anumită metrică dată d, în general nu există o metodă simplă și eficientă pentru a calcula atât distanța d(**C**), de la configurația **C** la regiunea C-obstacolelor SCB, cât și gradientul ei. O metodă practică, des utilizată, este aceea de a defini funcțiile potențial atractiv și repulsiv în $W = \mathbf{R}^N$ și de a combina efectele lor în câteva puncte din spațiul robotului *A*. Această tehnică este descrisă în următoarele două paragrafe.

8.2.2.2. POTENȚIALUL ATRACTIV

Fie a_j, N puncte selectate în *A*, unde N este dimensiunea spațiului de lucru (dacă N este 3, atunci cele 3 puncte a_j ar trebui să nu fie aliniate pentru a determina în mod unic configurația lui *A*). Punctele a_j sunt denumite *puncte de control supuse câmpului potențial atractiv*. Pentru fiecare punct a_j, $j \in [1,N]$, se va defini un potențial atractiv distinct prin funcția $V_{atr}^{j} : W \rightarrow \mathbf{R}$, unde:

$$V_{atr}^{j}(x) = 1/2\xi \left\| x - a_j(\mathbf{C}_{final}) \right\|^2,\ \xi \text{ - factor scalar pozitiv} \tag{8.28}$$

Fiecare potențial V_{atr}^{j} induce un câmp de forțe $-\vec{\nabla} V_{atr}^{j}$ asupra spațiului de lucru, dar numai punctul de control a_j este "sensibil" pentru acest câmp. La fiecare configurație **C** a lui *A*, forța artificială $\vec{F}_{atr}^{j}(\mathbf{C}) \in \mathbf{R}^N$ definită de:

$$\vec{F}_{atr}^{j}(\mathbf{C}) = -\vec{\nabla}\left(V_{atr}^{j}(x)\right)\Big|_{x = a_j(C)} = -\xi\left(a_j(\mathbf{C}) - a_j(\mathbf{C}_{final})\right) \tag{8.29}$$

este exercitată asupra robotului A în punctul a_j. Având produsul scalar în $T_C(SC)$, putem aplica această forță într-un vector $\vec{F}_{atr}^{j}(C)$ în $T_C(SC)$. Forța totală atractivă în $T_C(SC)$ este:

$$\vec{F}_{atr}(\mathbf{C}) = \sum_{j=1}^{N} \vec{F}_{atr}^{j}(\mathbf{C}) \tag{8.30}$$

Fie $U_{atr}^{j}(\mathbf{C}) = V_{atr}^{j}(a_j(\mathbf{C}))$. Se poate verifica ușor că:

BUPT

$$\vec{F}_{atr}^{\,j}(\mathbf{C}) = -\vec{\nabla} U_{atr}^{\,j}(\mathbf{C}) \tag{8.31}$$

Rezultă o expresie ce reprezintă potențialul atractiv în **C** şi anume:

$$U_{atr}(\mathbf{C}) = \sum_{j=1}^{N} V_{atr}^{j}(a_j(\mathbf{C})) \tag{8.32}$$

$U_{atr}(\mathbf{C})$ produce forța atractivă:

$$\vec{F}_{atr}(\mathbf{C}) = -\vec{\nabla} U_{atr}(\mathbf{C}) \tag{8.33}$$

Numărul punctelor a_j poate fi mai mic decât N, dar în acest caz specificarea poziției finale a acestor puncte nu determină decât în mod parțial configurația robotului *A* şi deci configurația finală rezultă sub forma unei regiuni a spațiului SC. Dacă mai mult decât un punct de control a_j este supus câmpului potențial atractiv, atunci definiția anterioară a potențialului U_{atr} face ca punctele de control a_j să concureze pentru a atinge configurația finală. Această concurență poate conduce la minimuri locale ale potențialului total, mai ales în pasaje înguste ale spațiului de lucru. O definiție mai bună în acest caz ar putea fi:

$$U_{atr}(\mathbf{C}) = V_{atr}^{1}(a_1(\mathbf{C})) + \varepsilon \cdot \sum_{j>1}^{N} V_{atr}^{j}(a_j(\mathbf{C})), \tag{8.34}$$

unde $\varepsilon < 1$ este o constantă pozitivă.

Această formulă corespunde alegerii unui "leader" printre punctele de control supuse potențialului atractiv (în acest caz s-a ales a_1). De-a lungul traiectoriei generate, acest punct "leader" este atras puternic spre poziția finală, în timp ce constanta mică ε lasă ca orientarea robotului *A* să fie în principal sub influența potențialului repulsiv. Al doilea termen al potențialului atractiv tinde de asemenea să aducă robotul spre orientarea finală. Funcția de mai sus U_{atr} dă rezultate bune atunci când configurația finală este mai degrabă îndepărtată de regiunea de C-obstacole, dar pe de altă parte poate cauza totuşi un minim local adânc care să existe chiar foarte aproape de configurația finală.

8.2.2.3. POTENȚIALUL REPULSIV

În mod similar se poate defini şi calcula potențialul repulsiv. Astfel, pentru toate punctele $x \in W \backslash B$, unde *B* este regiunea obstacolelor în W, se defineşte potențialul repulsiv în spațiul de lucru prin:

$$V_{rep}(x) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{d(x)} - \frac{1}{d_0}\right)^2 & d(x) \le d_0 \\ 0 & d(x) > d_0 \end{cases} \tag{8.35}$$

unde: η este un factor scalar pozitiv; d(x) este distanța euclidiană dintre punctul x şi frontiera lui B; d_0 este distanța de influență a lui *B*.

Fie a_j, j = 1,...,Q, punctele selectate din frontiera robotului *A*. Aceste Q-puncte se numesc *puncte de control supuse câmpului potențial repulsiv*. Punctele de control folosite în definirea potențialului atractiv nu trebuie să se afle printe aceste Q puncte deşi, acest lucru este posibil.

Atunci când robotul *A* este în configurația **C**, V_{rep} exercită o forță asupra lui *A*, care este aplicată în punctul a_j:

BUPT

$$\vec{F}_{rep}^{j}(\mathbf{C}) = \begin{cases} \eta \cdot \left(\dfrac{1}{d(a_j(\mathbf{C}))} - \dfrac{1}{d_0} \right) \cdot \dfrac{1}{d^2(a_j(\mathbf{C}))} \cdot \vec{\nabla} d(x) \Big|_{x=a_j(\mathbf{C})} & d(a_j(\mathbf{C})) \le d_0 \\ 0 & d(a_j(\mathbf{C})) > d_0 \end{cases} \quad (8.36)$$

Dacă B este o regiune convexă ce are frontiera diferențiabilă pe porțiuni, atunci d este diferențiabilă în spațiul SC_{liber}. Dacă B nu este convex, atunci se pote descompune în componente convexe şi se poate asocia câte un potențial repulsiv fiecărei componente.

Fiind dat un produs scalar în $T_C(SC)$, se poate aplica $\vec{F}_{rep}^{j}(\mathbf{C})$ într-un vector $\vec{F}_{rep}(\mathbf{C})$ în $T_C(SC)$. Forța repulsivă rezultată este:

$$\vec{F}_{rep}(\mathbf{C}) = \sum_{j=1}^{Q} \vec{F}_{rep}^{j}(\mathbf{C}) \quad (8.37)$$

Fie $U_{rep}^{j}(\mathbf{C}) = V_{rep}(a(\mathbf{C}))$. Se poate verifica relația următoare: $\vec{F}_{rep}^{j}(\mathbf{C}) = -\vec{\nabla} U_{rep}^{j}(\mathbf{C})$. Astfel,

$$U_{rep}(\mathbf{C}) = \sum_{j=1}^{Q} V_{rep}^{j}(a_j(\mathbf{C})) \quad (8.38)$$

este potențialul repulsiv în $\mathbf{C}$ care produce forța repulsivă.

Pentru a fi siguri că robotul A nu poate veni prea aproape de obstacolul B fără a fi respins, se vor combina un număr mic de puncte de control fixe şi un punct de control variabil care depinde de configurația lui A. Punctul de control variabil este punctul a' de pe frontiera lui A, care este cel mai apropiat de frontiera lui B în configurația curentă $\mathbf{C}$ a lui A. Deci el este soluția ecuației:

$$\min_{b \in B} \|a'(\mathbf{C}) - b\| = \min_{a \in A, b \in B} \|a(\mathbf{C}) - b\| \quad (8.39)$$

Forța repulsivă care este aplicată în a' este:

$$\vec{F}_{rep}^{j}(\mathbf{C}) = \begin{cases} \eta \cdot \left(\dfrac{1}{d(a'(\mathbf{C}))} - \dfrac{1}{d_0} \right) \cdot \dfrac{1}{d^2(a'(\mathbf{C}))} \cdot \vec{\nabla} d(x) \Big|_{x=a'(\mathbf{C})} & d(a'(\mathbf{C})) \le d_0 \\ 0 & d(a'(\mathbf{C})) > d_0 \end{cases} \quad (8.40)$$

Dacă atât A cât şi B sunt poligoane sau poliedre convexe (după cum se lucrează în spațiul cu N = 2 sau 3), atunci calculul minimului $\min_{a \in A, b \in B} \|a(\mathbf{C}) - b\|$, împreună cu o pereche de puncte care îndeplinesc această condiție de distanță minimă, poate fi făcut în timpul $O(n_A + n_B)$, unde n_A şi n_B sunt numărul vârfurilor lui A, repectiv B [D12]. Totuşi, pentru anumite configurații critice ale lui A, atunci când o latură sau o față a sa este paralelă cu o latură sau o față a lui B, distanța minimă este atinsă pentru mai mult decât o singură pereche de puncte. În aceste configurații, punctul variabil de control a' se poate deplasa în mod discontinuu pe frontiera lui A, rezultând o discontinuitate a forței repulsive $\vec{F}'_{rep}{}^{j}(\mathbf{C})$. Acest lucru poate cauza generarea unui drum oscilant. O metodă de a elimina acest dezavantaj este aceea de a distribui forța repulsivă asupra mai multor puncte $A(\mathbf{C})$ care se află printre cele mai apropiate de frontiera lui B. Fie S un astfel de set de puncte $A(\mathbf{C})$ cele mai apropiate de B. Dacă S nu este o mulțime cu un singur element, atunci S este fie un segment de linie, fie un poligon. Forța repulsivă poate fi distribuită asupra elementelor extreme ale mulțimii S, care pot fi capetele segmentului sau vârfurile poligonului.

BUPT

Dacă A şi B nu sunt poligoane convexe, atunci ele pot fi descompuse în două mulţimi de poligoane convexe $\{A_k\}$ şi $\{B_l\}$. Pentru fiecare pereche (A_k, B_l) se poate defini un punct variabil de control a'_{kj}, ca fiind punctul din frontiera lui A_k care se află cel mai aproape de B_l. Toate aceste puncte a'_{kj} vor fi în mod simultan considerate puncte de control supuse câmpului potenţial repulsiv.

Dacă A şi B au forme mai complexe, dar pot fi descompuse în componente convexe $\{A_k\}$ şi $\{B_l\}$, pot fi folosite tehnicile numerice iterative pentru a calcula distanţa dintre A_k şi B_l, precum şi o pereche de puncte aflate la distanţa dată.

Se poate defini câmpul potenţial repulsiv folosind mai multe puncte de control a'_{kj}, câte unul pentru fiecare pereche (A_k, B_l). Dacă A_k sau B_l sunt strict convexe, celălalt fiind convex, şi dacă ambele au frontiere diferenţiabile pe porţiuni, atunci a'_{kl} este unic definit şi deci forţa repulsivă exercitată în punctul de control variază în mod continuu.

8.3. METODE DE PLANIFICARE A MIŞCĂRII UTILIZÂND CÂMPUL POTENŢIAL (POTENTIAL - GUIDED PATH PLANNING)

În continuare vor fi descrise câteva tehnici simple de planificare a mişcării utilizând câmpul potenţial definit anterior. În principiu, aceste tehnici nu se referă la o funcţie potenţial specifică, deci sunt aplicabile atât pentru funcţiile potenţial descrise în cele două paragrafe anterioare, cât şi pentru alte funcţii potenţial.

În concepţia sa originală, metoda câmpului potenţial pentru generarea mişcării constă în considerarea robotului ca o particulă unitară aflată în spaţiul configuraţiilor, care se deplasează sub influenţa forţei $\vec{F} = -\vec{\nabla} U$. La fiecare nouă configuraţie $\mathbf{C}$, forţa $\vec{F}(\mathbf{C})$ determină acceleraţia particulei, viteza şi direcţia de deplasare. Cunoscând ecuaţia dinamică a mişcării robotului A, se pot calcula forţele şi momentele care ar trebui să acţioneze asupra robotului pentru ca acesta să se comporte ca o particulă. Aceste forţe/momente vor fi trimise ca şi comenzi servo-controlerului robotului [H8], [K5], [K15].

Acest mod de a utiliza funcţia potenţial este aplicabil pentru generarea în timp real a traiectoriei de mişcare, fiind potrivit atunci când obstacolele nu sunt cunoscute ca şi repartizare şi influenţă asupra mişcării, ci sunt detectate pe parcursul desfăşurăii acesteia. Dacă este disponibil un model aprioric cunoscut al obstacolelor, atunci metoda prezentată poate fi folosită pentru a simula mişcarea particulei, dar în acest caz există metode de planificare a mişcării mai simple şi mai eficiente, de asemenea folosind câmpul potenţial.

În continuare se prezintă câteva astfel de tehnici. Prima dintre acestea (8.3.1) generează o cale prin căutare "în adâncime" (depth-first planning) [D5], [L5], fără a reveni din căutare. La fel cum este metoda on-line a câmpului potenţial, şi metoda depth-first planning poate fi foarte rapidă dacă i se aplică condiţii favorabile, dar poate de asemenea să se blocheze într-un minim local al funcţiei potenţial. A doua tehnică (8.3.2) operează prin căutare "pe nivel" (best-first planning), eliminând minimurile locale prin "umplerea" lor. A treia metodă (8.3.3) prezentată constă în optimizarea funcţiei construite din integrarea potenţialului de-a lungul unei traiectorii complete între configuraţia iniţială şi cea finală.

BUPT

8.3.1. PLANIFICAREA "ÎN ADÂNCIME" (DEPTH - FIRST PLANNING)

Această tehnică construieşte un drum ca rezultat al unor segmente de drum succesive, începând de la configuraţia iniţială $\mathbf{C}_{init}$, şi terminând în configuraţia finală $\mathbf{C}_{final}$. Fiecare segment este orientat de-a lungul vectorului gradient negativ al funcţiei potenţial, calculat în configuraţia atinsă de segmentul anterior. Amplitudinea segmentelor este aleasă astfel încât ele să rămână în spaţiul liber şi să nu depăşească barierele de potenţial impuse de C-obstacole.

Fie $\mathbf{C}_i$ şi $\mathbf{C}_{i+1}$ originea şi respectiv extremitatea celui de-al *i*-lea segment al traiectoriei. Fie $x_j(\mathbf{C}_i)$, j = 1, ..., m coordonatele lui $\mathbf{C}_i$ într-o aplicaţie (U, φ). Se poate defini produsul scalar în spaţiul tangent $T_C(SC)$ astfel încât baza β indusă de această aplicaţie în $T_C(SC)$ să fie ortonormală:

$$\left[\vec{F}\right]_\beta = -\left[\vec{\nabla}U\right]_\beta = \left(-\frac{\partial U}{\partial x_1} \qquad -\frac{\partial U}{\partial x_m}\cdot\right)^T \tag{8.41}$$

Se notează componentele vectorului unitar $\vec{t}(\mathbf{C}_i) = \dfrac{\vec{F}(\mathbf{C}_i)}{\left\|\vec{F}(\mathbf{C}_i)\right\|}$ în β prin $t_j(\mathbf{C}_i)$.

Coordonatele în configuraţia $\mathbf{C}_{i+1}$ atinsă de robot la cea de-a *i*-a iteraţie în (U, φ) sunt:

$$x_j(\mathbf{C}_{i+1}) = x_j(\mathbf{C}_i) + \delta_i t_j(\mathbf{C}_i) \tag{8.42}$$

unde δ_i este lungimea celui de-al *i*-lea increment (măsurată cu metrica euclidiană din $\mathbf{R}^m$).

Segmentul $[\mathbf{C}_i\mathbf{C}_{i+1}]$ este imaginea inversă în SC a segmentului de dreaptă ce uneşte $\varphi(\mathbf{C}_i)$ cu $\varphi(\mathbf{C}_{i+1})$ în $\mathbf{R}^m$. De exemplu, dacă *A* este un obiect plan, care se deplasează în $W=\mathbf{R}^2$, atunci se poate parametriza orice configuraţie **C** prin:

$$(x_1, x_2, x_3) = (x, y, \theta) \in \mathbf{R}^2 \times [0, 2\pi), \tag{8.43}$$

unde x şi y sunt coordonatele punctului de referinţă al lui *A* în **C** şi θ este unghiul (modulo 2π) dintre axele absciselor sistemelor de referinţă S_W şi S_A ataşate lui W şi respectiv *A*. Astfel:

$$\begin{aligned} x(\mathbf{C}_{i+1}) &= x(\mathbf{C}_i) + \delta_i \frac{\partial U}{\partial x}(x,y,\theta) \\ y(\mathbf{C}_{i+1}) &= y(\mathbf{C}_i) + \delta_i \frac{\partial U}{\partial y}(x,y,\theta) \\ \theta(\mathbf{C}_{i+1}) &= \theta(\mathbf{C}_i) + \delta_i \frac{\partial U}{\partial \theta}(x,y,\theta) \end{aligned} \tag{8.44}$$

Pentru a normaliza deplasarea de-a lungul axei θ relativ la deplasarea de-a lungul axelor x şi y, se poate parametriza **C** prin $(x,y,\varphi) \in \mathbf{R}^2 \times [0, 2\pi r)$, prin transformarea $\varphi = \theta r$, unde $r = \max \| O_A - a \|$ este distanţa maximă dintre punctul de referinţă O_A şi frontiera lui *A*. Rezultă:

$$\varphi(\mathbf{C}_{i+1}) = \varphi(\mathbf{C}_i) + \delta_i \frac{\partial U}{\partial \varphi}(x,y,\varphi) \bmod 2\pi r \tag{8.45}$$

sau în mod echivalent:

BUPT

$$\theta(\mathbf{C}_{i+1}) = \theta(\mathbf{C}_i) + \frac{\delta_i}{r}\frac{\partial U}{\partial \varphi}(x, y, \theta r) \bmod 2\pi \qquad (8.46)$$

În alegerea unei valori pentru δ_i intervin câteva considerații importante. În primul rând, δ_i ar trebui să fie suficient de mic pentru ca atât direcția forței cât şi descompunerea direcției în sistemul de coordonate locale să aibă sens pentru segmentul $[\mathbf{C}_i\mathbf{C}_{i+1}]$. În mod normal, δ_i este luat egal cu un increment dat δ, cu excepția cazului când apar alte condiții suplimentare care necesită un termen δ_i mai mic.

Incrementul δ_i ar trebui de asemenea ales suficient de mic pentru a evita orice coliziune a segmentului $[\mathbf{C}_i\mathbf{C}_{i+1}]$ cu frontiera C-obstacolelor. Se presupune că δ_i este egal cu o constantă δ. Mişcarea din configurația $\mathbf{C}_i$ în $\mathbf{C}_{i+1}$ poate fi exprimată ca şi o mică rotație în jurul originii O_A urmată de o mică translație. Punctul robotului A care se deplasează cel mai mult în timpul rotației este punctul $a \in \partial A$ (aparține frontierei lui A) care este cel mai depărtat de O_A. Fie ℓ_1 lungimea acestei distanțe şi fie ℓ_2 lungimea translației. Suma $\ell=\ell_1+\ell_2$ este o limită maximă a deplasării punctului robotului A în timpul mişcării de la $\mathbf{C}_i$ la $\mathbf{C}_{i+1}$. Dacă ℓ este mai mică decât minimul distanței de la A la C-obstacole, atunci segmentul $[\mathbf{C}_i\mathbf{C}_{i+1}]$ este garantat că rămâne în spațiul liber. Altfel, trebuie aleasă o valoare mai mică pentru δ_i, de exemplu $\delta/2$ şi apoi verificată din nou condiția de coliziune.

În final, valoarea incrementului nu ar trebui să conducă la depăşirea configurației finale. Presupunând că aplicația (U, φ) folosită în $\mathbf{C}_i$ permite de asemenea aplicația în $\mathbf{C}_{final}$ ar trebui să se aleagă δ_i mai mic decât distanța euclidiană (în $\mathbf{R}^m$) dintre aplicațiile $\varphi(\mathbf{C}_i)$ şi $\varphi(\mathbf{C}_{final})$.

În concluzie, tehnica prezentată urmează cea mai apropiată descreştere a câmpului potențial până la atingerea configurației finale. Totuşi, metoda se poate bloca într-un minim al potențialului, altul decât cel final (fig.8. 1).

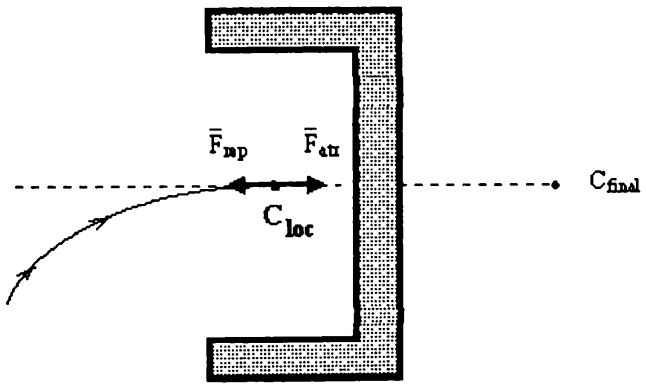


Fig.8.1. Potențialul atractiv atrage traiectoria robotului într-o concavitate a unui C-obstacol. La o anumită configurație $\mathbf{C}_{loc}$, forța repulsivă generată de C-obstacol anulează total forța atractivă generată de $\mathbf{C}_{final}$. Configurația stabilă creată este un minim local al funcției potențial totale

Tratarea minimului local în cadrul metodei de planificare a mişcării *depth-first planning* nu este facilă. Mai întâi trebuie recunoscută situația de atingere a unui minim local. Deoarece mişcarea este discretizată, planificatorul nu se opreşte de obicei exact în configurația cu forță nulă; dimpotrivă, se generează în mod tipic anumite oscilații în jurul acestei configurații. Aceste oscilații pot fi detectate prin verificarea dacă mai multe configurații succesive nu sunt prea apropiate între ele. În al doilea rând, planificatorul de

BUPT

mişcare trebuie să elimine minimul local. O tehnică simplă de rezolvare presupune impunerea unei mişcări de-a lungul unei anumite direcţii cu o distanţă convenabilă şi apoi reluarea algoritmului *depth-first planning* de generare a drumului pornind de la acea nouă configuraţie. O posibilă direcţie pentru a face acest pas suplimentar o poate constitui orice tangentă la curba echipotenţială repulsivă în acel minim local. Acest lucru corespunde unei mişcări repetate în jurul obstacolelor care creează un minim problematic.

O metodă mai sistematică de a trata minimul local constă în reprezentarea drumului generat sub forma unui graf, ale cărui noduri sunt configuraţiile $\mathbf{C}_i$. Această reprezentare conduce la o altă metodă de planificare: *best-first planning*.

8.3.2. PLANIFICAREA "PE NIVEL" (BEST - FIRST PLANNING)

Se presupune că se trasează peste spaţiul configuraţiilor SC o grilă regulată, notată cu SCG [L5]. SCG poate fi definită considerând un sistem de coordonate pe spaţiul SC şi discretizând fiecare din cele *m* axe de coordonate. De exemplu, dacă *A* este un obiect care se deplasează liber în spaţiul $W = \mathbf{R}^2$, grila constă în configuraţiile suprapuse ($k_x\delta_x$, $k_y\delta_y$, $k_0\delta_0$), unde k_x, k_y, $k_0 \in \mathbf{Z}$ şi θ este exprimat în modulo 2π. Incrementul unghiular $\delta\theta$ este o fracţie întreagă a lui 2π. (Ca şi în paragraful anterior, se poate folosi mai degrabă $\varphi = \theta r$ decât θ pentru a treia coordonată, pentru a normaliza scara axei unghiulare relativ la axele de translaţie).

Fiind dată o configuraţie **C** în grila m-dimensională SCG, cei p vecini ai acesteia (unde $1<p<m$) sunt definiţi, ca toate configuraţiile din grila SCG, care au cel mult p coordonate diferite de cele ale configuraţiei **C**, valoarea diferenţei fiind chiar valoarea absolută a incrementului cu care s-a discretizat mişcarea (evident cu aplicaţia modulo 2π pentru incrementul unghiular). Deci sunt (2m) 1-vecini, ($2m^2$) 2-vecini, ..., şi (3^m-1) m-vecini. În acest paragraf se consideră că două configuraţii din SCG sunt vecine dacă şi numai dacă ele sunt p-vecine pentru o valoare p dată $p \in [1,m]$. În practică, se poate considera p = 1 sau 2. În plus, pentru a simplifica prezentarea, se fac următoarele presupuneri:

- $\mathbf{C}_{init}$ şi $\mathbf{C}_{final}$ sunt configuraţii în SCG;
- dacă doi vecini din SCG sunt în spaţiul liber, segmentul de dreaptă ce îi uneşte în $\mathbf{R}^m$ se află de asemenea în spaţiul liber;
- grila SCG este limitată şi formează un rectangloid (acest lucru se obţine prin limitarea mulţimii poziţiilor posibile ale obiectului *A* la o formă rectangulară).

Planificarea prin metoda *best-first planning* constă în construirea iterativă a unui arbore T ale cărui noduri să fie configuraţii în SCG. Rădăcina arborelui este $\mathbf{C}_{init}$. La fiecare iteraţie, algoritmul examinează vecinii ramurii arborelui T care au cea mai mică valoare de potenţial, reţine vecinii care nu sunt deja prezenţi în arbore şi pentru care funcţia potenţial este mai mică decât o valoare M (se presupune că potenţialul U(**C**) este astfel definit încât creşte la infinit când **C** se apropie de regiunea C-obstacolelor şi este infinit când $C \in SCB$). În continuare, algoritmul instalează vecinii reţinuţi din grilă în arbore ca şi succesori direcţi ai ramurii considerate. Algoritmul se termină când a fost atinsă configuraţia finală (succes) sau când toată submulţimea SCG a configuraţiilor accesibile din $\mathbf{C}_{init}$ a fost cercetată (eşec). Fiecare nod al arborelui T are un indicator către părintele său. Dacă s-a ajuns la configuraţia finală, atunci se generează un drum prin urmărirea acestor indicatori de la $\mathbf{C}_{final}$ la $\mathbf{C}_{init}$.

BUPT

Procedura NIV este o exprimare formală a metodei de planificare "pe nivel". Pe lângă arborele T, NIV utilizează o listă LISTA care conține ramurile lui T sortate cu criteriul creşterii valorii funcției potențial. LISTA suportă următoarele trei operații: PRIM(LISTA), care elimină configurația din LISTA care are cel mai mic potențial, INSERT(C, LISTA), care adaugă configurația **C** în LISTA şi VID(LISTA), care returnează adevărat (sau fals) dacă LISTA este vidă (sau nu).

```
procedure NIV;
begin
        install C_init în T;            [inițial T este vid]
        INSERT(C_init, LISTA);          [marchează nodul C_init parcurs]
        [inițial toate configurațiile din SCG sunt marcate neparcurse]
        succes ← false;
        while NOT VID(LISTA) and NOT succes do
        begin
                C ← PRIM(LISTA);
                for every C' in SCG do
                [C' este vecin pentru C]
                        if U(C') < M and C' neparcurs then
                        begin
                                install C' în T cu indicator spre părinte
                                INSERT(C', LISTA);
                                marchez C' parcurs
                                if C' = C_final then succes ← true;
                        end;
        end;
if succes then
        rezultatul se scrie ca o traiectorie de la C_final la C_init
        în arborele T, folosind indicatorii spre rădăcină.
else return "eşec";
end;
```

Această procedură urmează o aproximare discretă a vectorului gradient negat al funcției potențial până când se atinge un minim local. Atunci când s-a ajuns la un minim, algoritmul procedează la "umplerea" acestuia până se atinge un punct unde se iese din regiunea critică din vecinătatea minimului local.

Algoritmul garantează întoarcerea unui drum liber dacă există vreunul în submulțimea grilei SCG sau raportează eşec în caz contrar. Fie O(r) numărul punctelor discretizate de-a lungul fiecărei axe de coordonate. Mărimea lui SCG este $O(r^m)$. LISTA poate fi reprezentată ca un arbore binar, astfel încât fiecare operație de tip PRIM sau INSERT să ocupe un timp logaritmic pentru a rezolva LISTA, timp care este în cel mai rău caz egal cu $O(r^m)$. Complexitatea temporală a procedurii NIV este deci $O(mr^m \log r)$ [L5].

BUPT

În majoritatea timpului, procedura NIV va cerceta doar o mică submulțime a grilei SCG, şi deci este preferabil să nu se reprezinte explicit grila printr-o matrice mare şi să nu se marcheze locațiile ca parcurse sau neparcurse. În schimb se reprezentă doar configurațiile care se află în arborele T. Apoi, în loc să se verifice dacă o configurație C' este parcursă sau nu, se va testa doar dacă aceasta se află în arborele T. Memorând configurațiile din T într-un arbore binar sortat după coordonatele configurațiilor, testul va ocupa acelaşi timp logaritmic $O(r^m)$. Această modificare creşte puțin timpul de execuție al algoritmului, îi scade viteza (fără a modifica complexitatea sa asimptotică), dar reduce spațiul de memorie necesar. De asemenea, testul face posibilă rularea algoritmului şi în cazul unei grile nelimitate (dar atunci există riscul ca şi timpul de execuție să fie infinit).

Procedura NIV este practică doar pentru cazul când m este mic, de exemplu mai mic decât 5. În cazul unui obiect liber într-un spațiu bi-dimensional (*m* este 3) algoritmul poate duce la o rezolvare foarte rapidă şi sigură a grilei de dimensiune 256^3 [B4]. Procedura poate fi făcută mai rapidă (ca şi timp mediu) prin utilizarea unei grile cu rezoluție variabilă. Dar atunci când *m* devine foarte mare, umplerea minimurilor locale nu mai este realizabilă (numărul configurațiilor discretizate aflate în regiunea critică a minimului local creşte exponențial cu dimensiunea *m*). În astfel de cazuri, cu *m* mare, se utilizează în general alte metode de planificare, de exemplu planificarea aleatoare.

8.3.3. PLANIFICAREA VARIAȚIONALĂ

O altă metodă de planificare a mişcării folosind câmpul potențial constă în construirea unei funcționale J a unui drum τ şi optimizarea acestei funcționale pentru toate drumurile posibile [L5]. De exemplu, dacă U este o funcție potențial care este definită în întregul spațiu al configurațiilor, cu valori mari în regiunea C-obstacolelor şi valori mici în spațiul liber, o posibilă definiție pentru J ar fi următoarea:

$$J(\tau) = \int_0^1 U(\tau(s))ds + \int_0^1 \left\| \frac{d\tau}{ds} \right\| ds \qquad (8.47)$$

Scopul primului termen din expresia lui J este de a face ca prin optimizare să se producă un drum liber. Al doilea termen al expresiei lui J este opțional şi are drept scop producerea unor drumuri mai scurte. De fapt, orice alte obiective pot fi codificate în această formă. Optimizarea funcționalei J poate fi făcută folosind metode standard de calcul variațional.

Dezavantajul major al metodei de planificare variațională este, la fel ca şi la metoda *best-first planning*, posibilitatea de a se bloca la atingerea unui minim local al lui J care nu corespunde unui drum liber. În plus optimizarea funcționalei J se efectuează peste un spațiu de dimensiune mare (egală cu numărul configurațiilor dintr-un drum) şi poate fi foarte grea.

Avantajul major al acestei metode este faptul că permite să fie codificat în corpul funcționalei J, sub forma unui termen al acesteia, orice criteriu suplimentar care este impus traiectoriei.

BUPT

8.3.4. GENERAREA MIŞCĂRII UNUI ROBOT TIP MAŞINĂ PRIN METODA CÂMPULUI POTENŢIAL

Pentru a genera mişcarea unui robot tip maşină s-a modelat mai întâi acest robot printr-un obiect rectangular care se mişcă într-un plan. Mişcarea robotului este supusă diferitelor restricţii cinematice, respectiv dinamice. Pentru a simula generarea mişcării acestui robot a fost elaborat un program bazat pe metoda depth-first planning prezentată în paragraful (8.3.1)[D17]. Algoritmul de planificare conceput generează C-obstacolele, calculează forţele atractivă, respectiv repulsive pentru fiecare configuraţie, construieşte segmentele de traiectorie pornind din configuraţia iniţială şi terminând în cea finală, verifică restricţiile neolonome la care este supus robotul şi modifică valorile parametrilor din expresiile potenţialelor atractiv, respectiv repulsiv, pentru optimizarea traiectoriei obţinute [D17], [T11], [D52].

8.3.4.1. MODELAREA ROBOTULUI TIP MAŞINĂ

Robotul tip maşină poate fi modelat ca un obiect rectangular care se mişcă într-un plan. Spaţiul configuraţiilor sale este $SC = \mathbf{R}^2 x S^1$. Din experienţă se cunoaşte că, într-un spaţiu de lucru gol, un mobil poate avea orice poziţie şi orice orientare. Deci, spaţiul configuraţiilor robotului are trei dimensiuni, două corespunzătoare translaţiei şi una corespunzătoare rotaţiei. Se poate reprezenta configuraţia robotului ca: (x,y,θ) [D41], [L5], unde x şi y sunt coordonatele punctului de mijloc dintre cele două roţi din spate $(x,y) \in \mathbf{R}^2$, iar $\theta \in [0,2\pi)$ este unghiul dintre axele x ale sistemului de referinţă (S) ataşat robotului şi sistemului de referinţă (S_0) ataşat spaţiului de lucru (fig.8.2).

Dacă se consideră contactul dintre fiecare din roţi şi teren ca o rostogolire pură între două corpuri rigide, atunci, punctul de mijloc R (originea sistemului de referinţă ataşat robotului) descrie o curbă (c), când robotul se mişcă în plan, care trebuie să fie tangentă la axa principală a mobilului. Astfel, mişcarea mobilului este constrânsă de ecuaţia:

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0 \tag{8.48}$$

care este o restricţie neolonomă. Pentru fiecare configuraţie (x,y,θ), impunem că viteza $(\dot{x},\dot{y},\dot{\theta})$ se află într-un subspaţiu vectorial bi-dimensional al spaţiului tangent. Impunând reprezentarea spaţiului tangent $(\dot{x},\dot{y},\dot{\theta})$ la reprezentarea (x,y,θ) a spaţiului configuraţiilor mobilului, subspaţiul vitezei poate fi reprezentat ca un plan perpendicular pe palnul (x,y) care se proiectează de-a lungul axei principale a robotului. Traiectoria (c) parcursă de mobil în spaţiul (x,y,θ) este local un helicoid.

Fie ϕ unghiul dintre axa principală a robotului şi vectorul viteză a punctului de mijloc dintre cele două roţi din faţă. ϕ se numeşte *unghi conducător* şi trebuie să satisfacă următoarea restricţie:

$$|\phi| \le \phi_{max} < \pi/2 \tag{8.49}$$

Punctul de mijloc dintre cele două roţi din spate trebuie să urmeze o curbă a cărei rază de curbură este limitată superior de ecuaţia:

$$\frac{1}{\rho_{min}} = \frac{1}{L} tg\phi_{max} \tag{8.50}$$

BUPT

Vectorul viteză $(\dot{x},\dot{y},\dot{\theta})$ al maşinii a cărei rază de întoarcere este limitată inferior de $\rho_{min}>0$ verifică:

$$\dot{x}^2+\dot{y}^2-\rho_{min}^2\dot{\theta}^2\geq 0 \qquad (8.51)$$

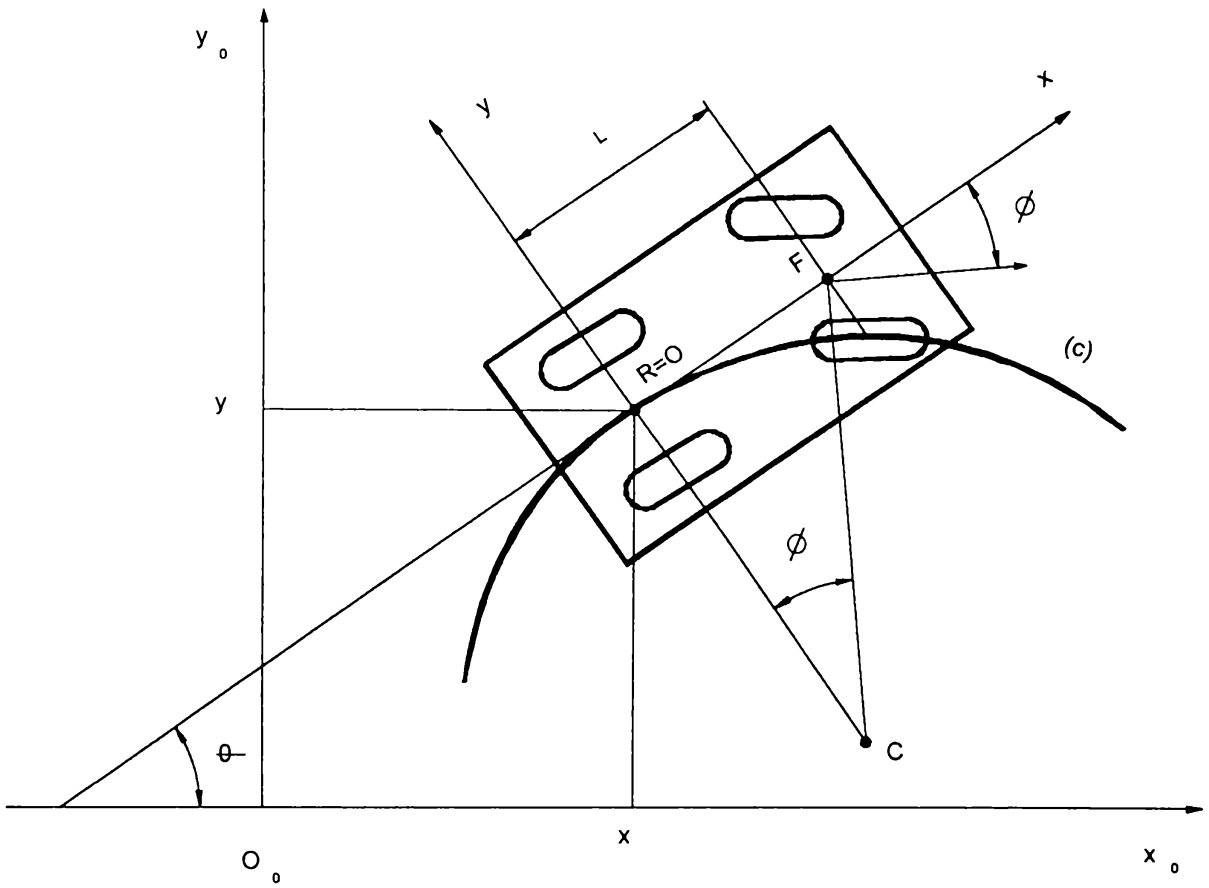


Fig.8.2. Robotul tip maşină modelat ca un obiect rectangular care se mişcă în plan

Problema de planificare este specificată prin configuraţia $(x_{init},y_{init},\theta_{init})$ şi prin configuraţia finală $(x_{final},y_{final},\theta_{final})$. Astfel, dându-se o poziţie iniţială (x_{ini},y_{init}) şi o poziţie finală (x_{final},y_{final}), trebuie să se calculeze o traiectorie semiliberă care conectează configuraţia iniţială de cea finală, astfel încât orientarea vectorului tangent la traiectorie în configuraţia iniţială, respectiv finală, să fie θ_{init}, respectiv θ_{final}. Pentru a putea determina o traiectorie liberă trebuie să se determine în prealabil C-obstacolele. Deci, mişcarea robotului trebuie să respecte două restricţii neolonome:

$$-\dot{x}\sin\theta+\dot{y}\cos\theta=0 \quad \text{şi} \quad \dot{x}^2+\dot{y}^2-\rho_{min}^2\dot{\theta}^2\geq 0 \qquad (8.52)$$

8.3.4.2. IMPLEMENTAREA ALGORITMULUI

Spaţiul de lucru al robotului este populat cu obstacole (obiecte fizice). Robotul trebuie să îndeplinească o anumită sarcină, adică să execute anumite mişcări în spaţiul său de lucru, printre aceste obstacole. Fiecare obstacol B_i din spaţiul de lucru determină în spaţiul configuraţiilor SC o regiune SCB_i numită C-obstacol:

$$SCB_i=\{\mathbf{C}\in SC / A(\mathbf{C})\cap B_i\neq\varnothing\} \qquad (8.53)$$

BUPT

Reuniunea tuturor C-obstacolelor formează regiunea C-obstacolelor SCB. Planificarea mişcării robotului în spaţiul de lucru este echivalentă în acest moment cu planificarea mişcării originii sistemului de referinţă ataşat robotului în SCB.

Pentru planificarea mişcării robotului tip maşină s-a folosit metoda depth-first planning. Traiectoria obţinută este produsul unor segmente succesive, pornind din $\mathbf{C}_{init}$ şi terminând în $\mathbf{C}_{final}$. Pentru a reduce numărul şi mărimea minimurilor locale ale funcţiei potenţial s-a folosit un potenţial atractiv cu simetrie sferică în jurul configuraţiei finale şi potenţiale repulsive eliptice. Contururile echipotenţiale atractive sunt cercuri cu centrul în $\mathbf{C}_{init}$, iar în interiorul zonei de influenţă, echipotenţialele repulsive sunt "dreptunghiuri cu colţurile rotunjite".

Distanţa dintre liniile echipotenţiale repulsive poate fi stabilită în funcţie de precizia dorită. Programul permite modificarea constantelor ξ şi η şi calculează: funcţia potenţial rezultant (în funcţie de δ_j), matricea gradient, versorul segmentului $[\mathbf{C}_i\mathbf{C}_{i+1}]$, coordonatele punctului $\mathbf{C}_{i+1}$ în funcţie de $\mathbf{C}_i$. Calculul se repetă până când $\mathbf{C}_{i+1}$ este configuraţia finală $\mathbf{C}_{final}$.

În figurile 8.3-8.7 sunt prezentaţi paşii algoritmului de planificare:

-figura 8.3 prezintă spaţiul de lucru populat cu obstacole;

-figura 8.4 prezintă potenţialele repulsive generate de C-obstacole;

-figura 8.5 prezintă potenţialul atractiv generat de $\mathbf{C}_{final}$;

-figura 8.6 prezintă potenţialul total ca sumă între potenţialul atractiv şi cele repulsive;

-figura 8.7 prezintă traiectoria obţinută.

În fiecare configuraţie trebuie îndeplinite restricţiile neolonome prezentate în paragraful precedent.

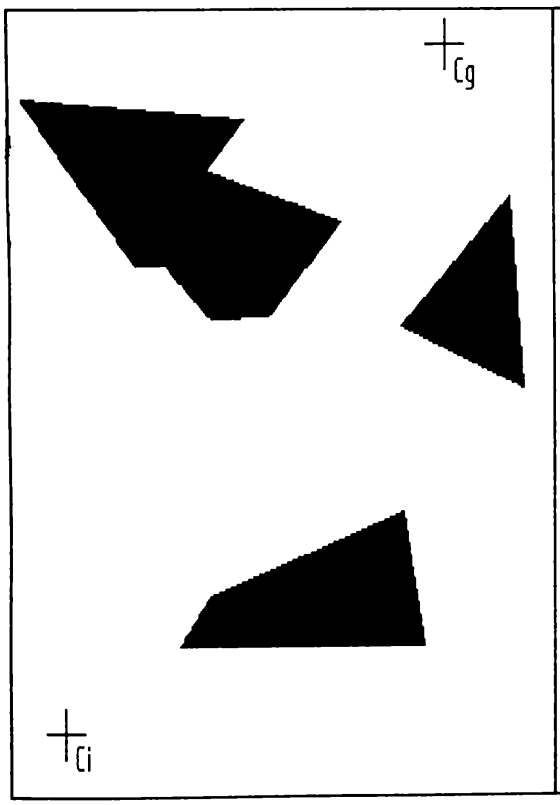


Fig.8.3. Spaţiul de lucru populat cu obstacole

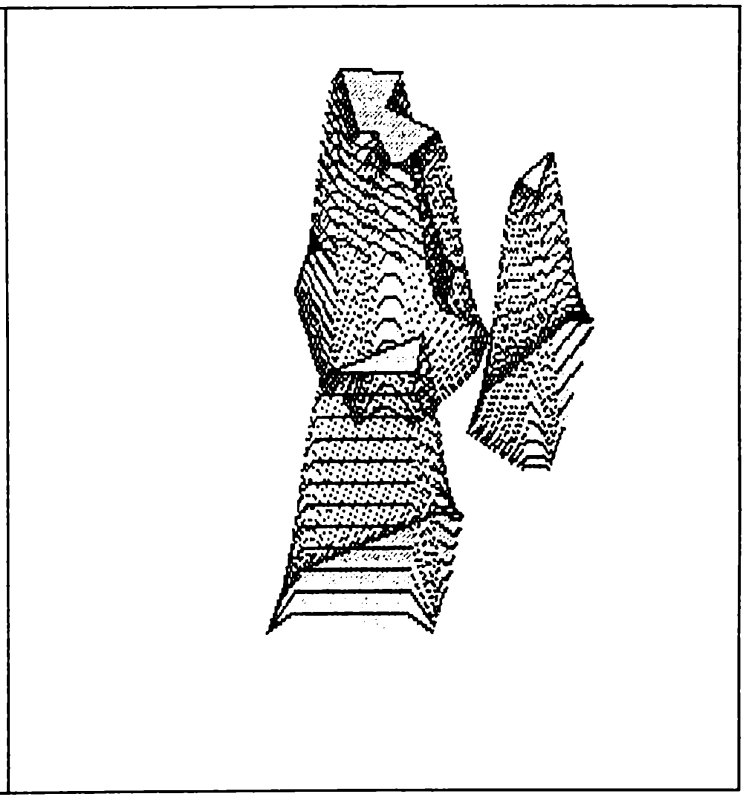


Fig.8.4. Potenţialele repulsive

BUPT

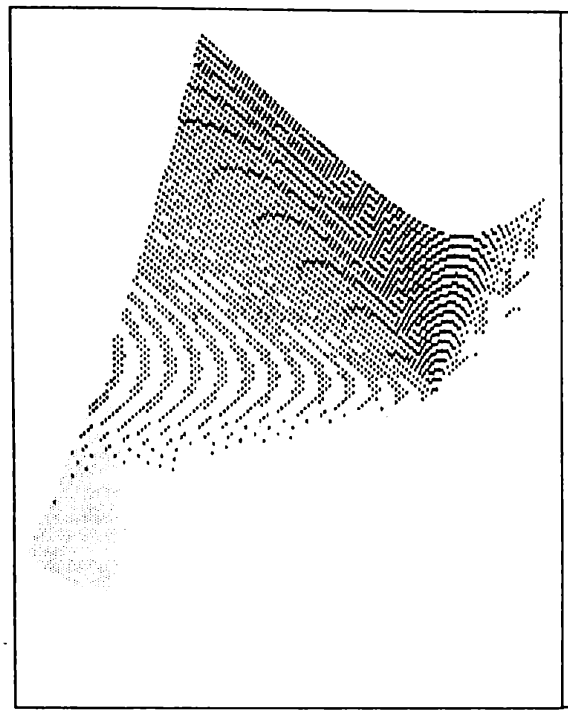


Fig.8.5.Potențialul atractiv generat de $\mathbf{C}_{final}$

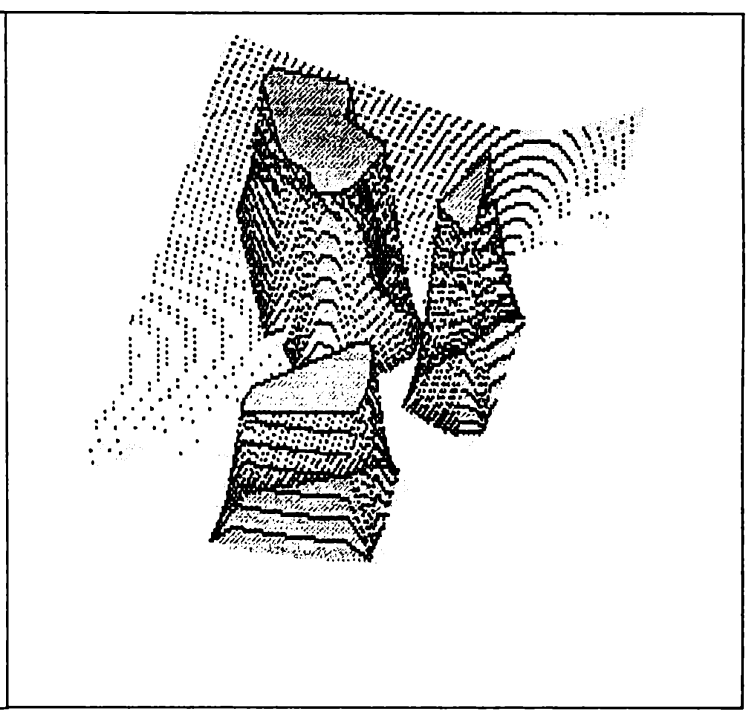


Fig.8.6. Potențialul total

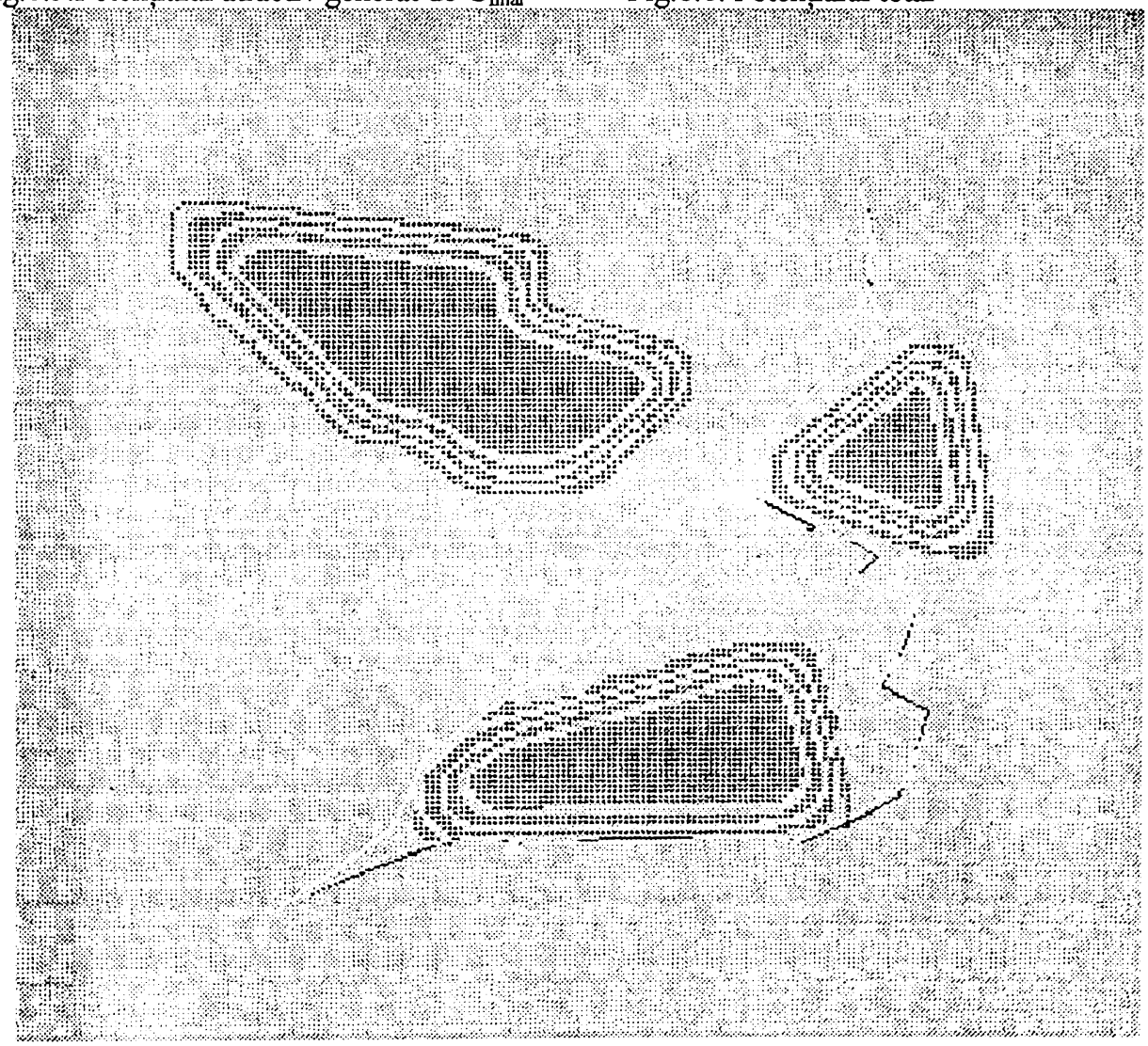


Fig.8.7. Traiectoria obținută

BUPT

8.4. ALTE FUNCȚII POTENȚIAL

În literatura de specialitate au fost prezentate mai multe tipuri de funcții potențial, altele decât cele prezentate în subcapitolele anterioare. Cele mai interesante dintre acestea urmăresc unul din următoarele două scopuri:

- comportamentul dinamic local al robotului de-a lungul drumului generat;
- reducerea numărului de minimuri locale şi/sau a mărimii câmpului de atracție generat de acestea.

Primul scop este important când metoda câmpului potențial este utilizată la generarea on-line a drumului, sau cu ajutorul metodei "în adâncime" (deşi în al doilea caz, drumul poate fi îmbunătățit în paşi recursivi).

Una dintre funcțiile propuse este "câmpul potențial generalizat" [K22] care este o funcție de configurația robotului şi de viteza acestuia. Ea este în aşa fel construită încât robotul este respins de către obstacol doar dacă acesta se apropie de obstacol şi viteza sa este îndreptată înspre obstacol. Dacă robotul se mişcă paralel cu obstacolul, nu este respins de acesta.

Utilizând o idee similară, se poate descrie o metodă de planificare ce constă în minimizarea unui criteriu pătratic supus unor restricții liniare. Această minimizare împinge robotul spre configurația finală, în timp ce fiecare restricție îndepărtează robotul de obstacol, când mobilul se află în apropierea obstacolului şi se deplasează spre acesta. Problema cu care se confruntă metoda constă în prezența minimurilor locale.

În continuare se vor prezenta rezultatele teoretice şi experimentale ale definirii unor funcții potențial "bune", având niciunul sau doar câteva minimuri locale.

Numărul minimurilor locale nu este singurul criteriu de măsură a calității funcției potențial. Un alt factor critic este mărimea domeniului de atracție a acestor minimuri. O tehnică de căutare ar putea scăpa fără dificultate din minimurile locale mici, dar s-ar putea bloca într-unul mai mare.

8.4.1. NOȚIUNEA DE FUNCȚIE DE NAVIGAȚIE

Deorece minimurile locale, altele decât configurația finală, sunt cauza majoră a ineficienței metodelor câmpului potențial, definiția funcției potențial nu este impusă ca o formulare originală a problemei de bază a planificării mişcării. Astfel, o problemă importantă este următoarea: *Este posibilă construirea unei funcții potențial* $U:SC_{liber} \rightarrow \mathbf{R}$, *cu minimul în* $\mathbf{C}_{final}$, *al cărei domeniu de atracție include întreaga submulțime a lui* SC_{liber} *conectat cu* $\mathbf{C}_{final}$? O astfel de funcție, dacă există, se numeşte "funcție de navigație globală". Dacă ea poate fi construită, atunci prin generarea drumului "în adâncime" va fi garantată atingerea $\mathbf{C}_{final}$, ori de câte ori $\mathbf{C}_{initial}$ şi $\mathbf{C}_{final}$ aparțin aceleiaşi submulțimi conexe a lui SC_{liber} şi eşuează în caz contrar.

În general, în prezența C-obstacolelor, când SC_{liber} este strict conținut în SC, funcția de navigație globală nu există. Mai mult, dacă $SC = \mathbf{R}^2$ şi dacă în SC se află q ($q>0$) C-obstacole disjuncte, atunci funcția potențial U trebuie să aibă cel puțin q puncte singulare tip "şa".

În orice caz, nu par a exista obstrucții topologice în existența unei funcții potențial analitice cu minimul în $\mathbf{C}_{final}$ al cărei domeniu de atracție include întreaga submulțime a spațiului SC_{liber} conex cu $\mathbf{C}_{final}$, cu excepția unei mulțimi de puncte de mărime zero care

BUPT

sunt punctele "şa" ale funcţiei potenţial. Aceste puncte sunt configuraţii de echilibru instabil, iar orice deviaţie aleatoare uşoară permite planificatorului să evadeze din ele.

O astfel de funcţie potenţial se numeşte "funcţie de navigaţie globală aproximativă". În continuare ea va fi numită simplu, funcţie de navigaţie.

Dacă SC = $\mathbf{R}^N$ şi dacă toate C-obstacolele sunt regiuni sferice disjuncte, atunci se poate defini o funcţie de navigaţie U utilizând formula dată în subcapitolul (8.3.2), ca şi suma dintre potenţialul atractiv de formă pătratică şi potenţialul repulsiv, generat de fiecare C-obstacol, cu distanţa de influenţă d_0 mai mică decât minimul jumătăţii distanţei minime dintre oricare două C-obstacole şi distanţa dintre $\mathbf{C}_{final}$ şi cel mai apropiat C-obstacol.

Bineînţeles, problema constă în generalizarea acestei construcţii specifice la C-obstacole cu alte forme. Un procedeu ar fi deformarea netedă a C-obstacolelor complexe în forme sferice. Într-adevăr, dacă U este o funcţie de navigaţie peste $\mathbf{R}^N$ şi $\upsilon:\mathbf{R}^N \rightarrow \mathbf{R}^N$ este un difeomorfism, atunci $U = U \circ \upsilon$ este de asemenea o funcţie de navigaţie [L5]. Prin urmare, problema constă în construirea unui difeomorfism υ care deformează mulţimea dată a C-obstacolelor într-o mulţime de regiuni sferice.

8.4.2. POTENŢIALUL GRILĂ

Deşi pare dificilă construirea unei funcţii analitice de navigaţie peste un spaţiu liber al unei geometrii arbitrare, calcularea unei funcţii numerice de navigaţie peste reprezentarea spaţiului configuraţiilor sub formă de grilă pare a simplifica problema .

În continuare vor fi descrişi diferiţi algoritmi propuşi în [B4] pentru calcularea a două astfel de funcţii de navigaţie (8.4.2.1 şi 8.4.2.2). Aceşti algoritmi sunt eficienţi când dimensiunea spaţiului configuraţiilor este mică (m = 2 sau 3) şi timpul de calcul este independent de geometria spaţiului liber, dar viteza lor scade când *m* creşte. Cu toate acestea, când *m* este mare, aceşti algoritmi pot fi încă utilizaţi la generarea unui câmp potenţial cu proprietăţi "bune".

8.4.2.1 FUNCŢII DE NAVIGAŢIE NUMERICE SIMPLE

Se consideră discretizarea lui SC într-o grilă rectangulară SCG. Fiecare configuraţie din SCG va fi notată "0" dacă se află în spaţiul liber şi "1" în caz contrar. Submulţimea configuraţiilor notate "0" este denumită SCG_{liber}. Se presupune că $\mathbf{C}_{initial}$ şi $\mathbf{C}_{final} \in SCG_{liber}$.

O funcţie de navigaţie U este distanţa L^1 (distanţa "Manhattan") la $\mathbf{C}_{final}$ în SCG_{liber}. Ea poate fi uşor calculată utilizând algoritmul "*wavefront expansion*" [B4] (expansiunea frontului de undă). În primul rând, $U = 0 \mid_{C=Cfinal}$. Apoi, U = 1 la fiecare 1-vecin al lui $\mathbf{C}_{final}$ în SCG (se presupune că distanţa dintre doi 1-vecini în SCG este normalizată la 1); U = 2 la fiecare 1-vecin al acestor noi configuraţii (dacă acesta nu a fost calculat anterior), etc. Algoritmul se termină când toată submulţimea grilei SCG_{liber} accesibilă din $\mathbf{C}_{final}$ a fost explorată complet.

Figura 8.8 ilustrează paşii algoritmului când SC = $\mathbf{R}^2$ şi grila are dimensiunea 64^2. $\mathbf{C}_{final}$ se află în colţul stânga sus al grilei. Figurile 8.8.a - 8.8.g arată diferite stări ale

BUPT

expansiunii frontului undei. Liniile de front sunt contururi echipotențiale ale potențialului U calculat de algoritm. Figura 8.8.h arată drumul cel mai abrupt urmat din $\mathbf{C}_{initial}$. Aceasta este cea mai scurtă cale (pentru distanța L^1 din SCG) dintre $\mathbf{C}_{initial}$ şi $\mathbf{C}_{final}$.

Algoritmul calculează U doar în submulțimea conexă a lui SCG_{liber} ce conține $\mathbf{C}_{final}$. Deci, dacă $U(\mathbf{C}_{initial})$ nu a fost calculat de algoritm, se poate şti imediat că nu există un drum liber ce uneşte $\mathbf{C}_{initial}$ de $\mathbf{C}_{final}$ la rezoluția grilei. O expresie simplificată a algoritmului descris mai sus este următoarea:

```
procedure NF1;
begin
        for every C in SCG_liber do U(C) ← M;     [M este cel mai mare număr]
        U(C_final) ← 0;
        insert C_final in L_0;
        [L_i, i = 0,1,... este o listă a configurațiilor inițial goală]
        for i=0,1,... until L_i is empty do
         for every C in L_i do
          for every 1-neighbor C' in SCG_liber do
                if U(C')=M then
                begin
                        U(C') ← i+1;
                        insert C' at the end of L_i+1
                end;
end;
```

Viteza algoritmului este liniară cu numărul configurațiilor din grila SCG şi este independentă de numărul şi forma C-obstacolelor.

8.4.2.2. Funcții de navigație numerice îmbunătățite

Un dezavantaj al funcției de navigație calculată cu procedura NF1 din subcapitolul precedent este că generează drumuri care în general trec foarte aproape de C-obstacole. În schimb, funcția de navigație îmbunătățită calculată în continuare generează drumuri ce trec cât mai departe posibil de C-obstacol, dar lungimile L^1 ale acestor drumuri nu vor mai fi minimale. Funcția de navigație îmbunătățită este calculată în trei paşi. În primul pas se extrage o submulțime S (m-1)-dimensională a grilei SCG_{liber} numită *schelet*. Apoi, în al doilea pas, funcția potențial U este calculată în S, iar în al treilea pas U este calculat în restul grilei SCG_{liber}.

BUPT

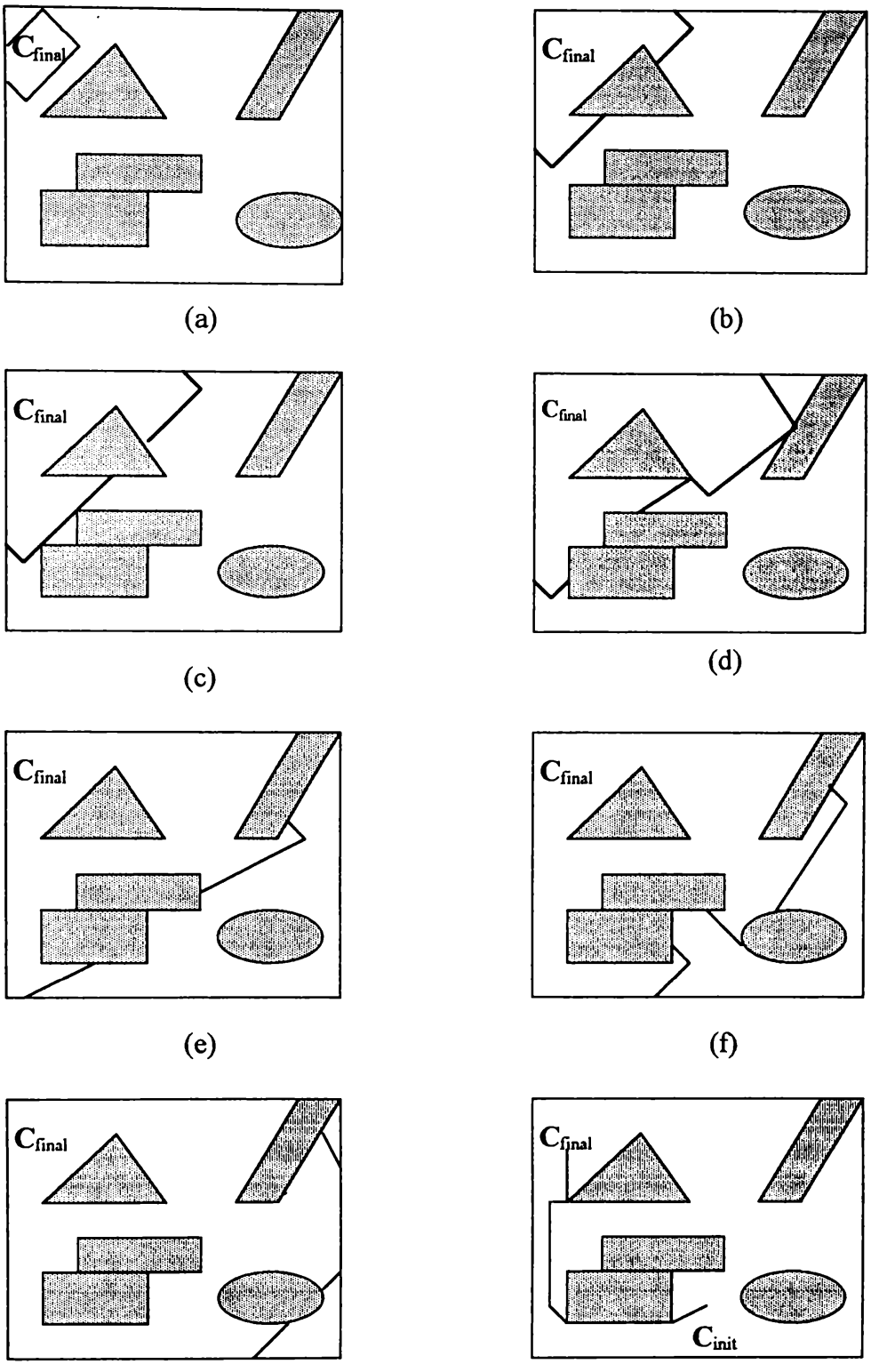


Fig.8.8. Calculul funcției de navigație utilizând procedura NF1 (algoritmul "wavefront expansion") când SC = $\mathbf{R}^2$, într-o grilă 64 x 64

BUPT

1. Calculul scheletului: distanța L^1, $d_1(\mathbf{C})$ de la fiecare configurație $\mathbf{C} \in SCG_{liber}$ până la regiunea C-obstacolelor este calculată cu algoritmul "wavefront expansion" începând de la frontiera lui SCG_{liber}. Această frontieră este construită în liniile 5-10 ale procedurii L1 date mai jos.

Procedura următoare calculează aplicația d_1 peste grila SCG_{liber}:

```
1.  procedure L1;
2.  begin
3.  for every C in SCG_liber do U(C) ←M; [M este cel mai mare număr]
4.  [L_i, i = 0,1,..., este o listă a configurațiilor, inițial goală]
5.  for every C in SCG_liber do
6.      if există un 1-vecin C' al lui C în SCG_liber then
7.      begin
8.          d_1(C) ← 0;
9.          insert C at the end of L_0;
10.     end;
11. for i = 0,1,..., until L_i is empty do
12.     for every C in L_i do
13.         for every 1-vecin C' of C in SCG_liber do
14.             if d_1(C') = M then
15.             begin
16.                 d_1(C') ← i+1;
17.                 insert C' at the end of L_{i+1};
18.             end;
19. end;
```

În paralel cu calculul aplicației d_1 se construieşte submulțimea S (m-1)-dimensională a lui SCG_{liber} formată din configurațiile în care "undele" (listele L_i din procedura de mai sus) intersectează frontiera lui SCG_{liber}. Aceasta se realizează nu numai prin propagarea valorilor lui d_1, dar şi a punctelor din frontiera discretizată a lui SCG_{liber} care sunt la începutul propagării. Pentru fiecare configurație $\mathbf{C}$ din SCG_{liber} pentru care $d_1(\mathbf{C})$ a fost calculat, $O(\mathbf{C})$ denotă următorul punct calculat. În linia 8 a procedurii, $O(\mathbf{C})$ ia valoarea $\mathbf{C}$. În linia 16, $O(\mathbf{C}') = O(\mathbf{C})$. "If"-ul din liniile 14-18 sunt extinse cu partea de "else" care ia în considerare cazul $d_1(\mathbf{C}') \neq M$, caz în care este posibil ca două unde să se intersecteze, şi construieşte S: dacă distanța $L^1 = D_1(O(\mathbf{C}'), O(\mathbf{C}))$ dintre $O(\mathbf{C}')$ şi $O(\mathbf{C})$ este mai mare decât o anumită intrare α, vom include $\mathbf{C}$ în S.

Algoritmul complet care calculează aplicația d_1 şi scheletul S este următoarul (S este inițial gol):

```
procedure SKELETON;
begin
    for every C in SCG_liber do U(C) ← M;    [M este cel mai mare număr]
    [L_i, i = 0,1,... este o listă a configurațiilor inițial goală]
    for every C in SCG\SCG_liber do
    if există un 1-vecin C' al lui C în SCG_liber then
    begin
```


BUPT

```
                d1(C) ← 0; O(C) = C;
                        insert C at the end of L0;
        end;
        for i = 0,1,..., until Li is empty do
                for every C in Li do
                        for every 1-vecin C' of C in SCGliber do
                                if d1(C') = M then
                                begin
                                        d1(C') ← i+1; O(C') ← O(C);
                                        insert C' at the end of Li+1;
                                end;
                                else if D1(O(C'), O(C)) > α then
                                [α este o constantă întreagă cu valori între 2 şi 6]
                                        if C∉S then insert C' in S;
                                        [acest test evită generarea unui dublu schelet
                                        rezultat din faptul că se intersectează două
                                unde]
end;
```

2. Calculul lui U în scheletul S. $\mathbf{C}_{final}$ este în primul rând conectat la S printr-un un drum σ urmat de cel mai abrupt "urcuş" al aplicaţiei d_1 în SCG_{liber}. σ este inclus în S (liniile 6-12 din procedura NF2). Apoi, U este calculat în S cu algoritmul "wavefront expansion" restrâns la S şi începând cu $\mathbf{C}_{final}$. Algoritmul utilizează aplicaţia d_1 calculată anterior pentru controlul expansiunii. În primul rând U= 0 la $\mathbf{C}_{final}$ şi $\mathbf{C}_{final}$ este inserat în lista Q a configuraţiilor sortate după valoarea descrescătoare a lui d_1. Apoi, cât timp Q nu este goală, primul element **C** din listă va fi şters; fiecare m-vecin **C'** al lui **C** din S, al cărui potenţial nu a fost încă calculat va primi valoarea U(**C**)+1 şi va fi inserat în Q. Algoritmul se va termina când Q se va goli, adică când toate configuraţiile din S accesibile din $\mathbf{C}_{final}$ au primit o valoare de potenţial.

O expresie simplificată a algoritmului ce calculează U în S este dată de liniile 14 - 29 ale procedurii NF2. Rezultatul algoritmului este lista L_0 a tuturor configuraţiilor din S accesibile din $\mathbf{C}_{final}$.

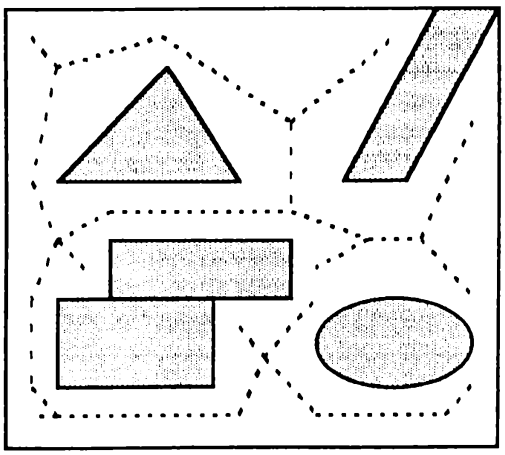
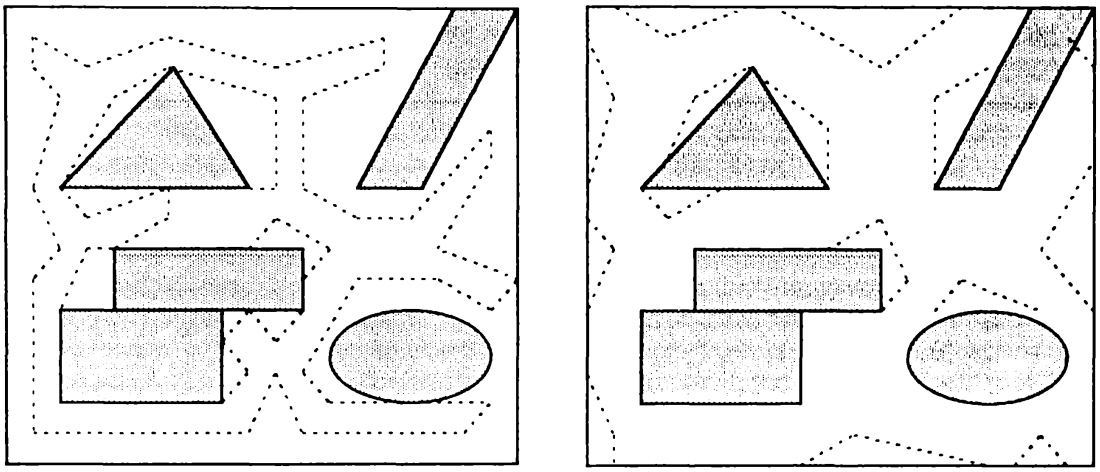


Fig.8.9. Scheletul calculat al aceluiaşi spaţiu bi-dimensional ca în fig.8.8.

În spaţiul bidimensional, scheletul este un fel de diagramă Voronoï pentru distanţa L

BUPT

3. Calculul lui U în restul spațiului liber. Potențialul U în restul submulțimii lui SCG_{liber} accesibil din $\mathbf{C}_{final}$ este calculat cu algoritmul "wavefront expansion" începând cu configurațiile din L_0. Potențialul fiecărui 1-vecin **C'** al fiecărei configurații **C** din L_0 este egal cu U(**C**)+1. Apoi, potențialul fiecărui vecin **C"** al configurației **C'**, al cărei potențial a fost calculat anterior este egal cu U(**C'**)+1, etc. Algoritmul (liniile 24 - 41) din NF2) se termină când s-au luat în considerare toate configurațiile din SCG_{liber} accesibile din $\mathbf{C}_{final}$. Figura 8.10 arată două stadii ale propagării frontului de undă din scheletul din figura 8.9.



(a) (b)

Fig.8.10. Două stări succesive a algoritmului "wavefront expansion" calculate de procedura NF2

Procedura NF2 dată în continuare este o expresie simplificată a algoritmului ce calculează U în submulțimea grilei SCG_{liber} accesibilă din $\mathbf{C}_{final}$.

```
1     procedure NF2;
2     begin
3           for every C in SCG_liber do U(C) ← M;     [M este cel mai mare număr]
4           [următoarea secvență de instrucțiuni conectează C_final la S]
5           [σ este o listă de configurații inițial goală]
6           insert C_final în σ; C←C_final
7           while C∉S do
8           begin
9                 selectează un vecin C' având cea mai mare valoare a lui d_1;
10                insert C' in σ; C←C';
11          end;
12          S←σ ∪ S;
13          [secvența următoare de instrucțiuni calculează U în S]
14          U(C_final) ←0; INSERT(C_final, Q);
15          [Q este o listă de configurații sortate în ordinea descrescătoare a lui d_1;
16          ea suportă operațiile FIRST, INSERT şi EMPTY;
17          inițial este vidă]
18          [L_i, i = 0,1,..., este o listă de configurații, inițial vidă]
```


BUPT

```
19          while not EMPTY(Q) do
20          begin
21                  C←FIRST(Q);
22                  insert C at the end of L0;
23                  for every m-vecin C' al lui C din S do
24                          if U(C') = M then
25                          begin
26                                  U(C') ←U(C)+1;
27                                  INSERT(C',Q);
28                          end;
29          end;
30          [la sfârşitul acestei bucle, L0 va conţine toate configuraţiile din S
31          accesibile din Cfinal]
32          [instrucţiunile următoare calculează U în restul submulţimii lui SCGliber
33          accesibilă din Cfinal]
34          for i = 0,1,..., until Li is empty do
35                  for every C in Li do
36                          for every  vecin C' al lui C din SCGliber do
37                                  if U(C') = M then
38                                  begin
39                                          U(C') ← U(C)+1;
40                                          insert C' at the end of Li+1;
41                                  end;
42  end;
```

Viteza de calcul a algoritmului descris este puţin mai mare decât cea a algoritmului NF1. Fie *n* numărul configuraţiilor din SCG şi *s* (respectiv *s'*) numărul configuraţiilor din S (respectiv S'=S crescut cu σ). Reprezentând S într-un arbore binar, viteza algoritmului SKELETON este O(n+s log s) [L5].

Reprezentând lista Q într-un arbore binar, viteza lui NF2 este O(n+s' log s'). În general, se poate presupune că *s* şi *s'* sunt proporţionale cu $n^{\frac{m-1}{m}}$ deoarece S şi σ sunt submulţimi (m-1)-dimensionale ale SCG. Cu această condiţie, viteza procedurilor SKELETON şi NF2 este $O(n + n^{\frac{m-1}{m}} \log n)$, şi deci este liniară în n.

Variante ale potenţialului descris mai sus sunt uşor de definit. De exemplu, în linia 39 a algoritmului NF2 se poate calcula $U(\mathbf{C}') = U(\mathbf{C}) + 1/d_1(\mathbf{C})$, rezultând un potenţial U ce devine infinit la frontiera SCG_{liber}.

BUPT

8.5. CALCULUL TRAIECTORIEI ÎN CAZUL ÎN CARE OBSTACOLUL *B* ESTE O BARĂ

Se consideră un mobil punctiform A care se deplasează în planul xOy. Mobilul porneşte din originea sistemului de referinţă $A_{initial}(x_{initial}=0, y_{initial}=0)$. Configuraţia finală a acestuia corespunde punctului $A_{final}(x_{final}, y_{final})$. În spaţiul de lucru al mobilului există un obstacol format de o bară $B[B_1(a,0);\ B_2(a,b),\ 0<a<x_{final},\ 0<b<y_{final}]$ (fig.8.11). Se calculează ecuaţiile diferenţiale ale mişcării mobilului A supus acţiunii unui câmp potenţial.

Spre deosebire de paragrafele precedente, când forţa rezultantă s-a obţinut prin aplicarea gradientului negativ funcţiei potenţial, în acest caz se consideră că, prin aplicarea gradientului negativ funcţiei potenţial se obţine viteza mobilului:

$$\vec{v} = -\vec{\nabla}U(\mathbf{C}) \tag{8.54}$$

Funcţia potenţial totală se alege de forma:

$$U(\mathbf{C}) = \frac{1}{2}\xi \cdot (\mathbf{C} - \mathbf{C}_{final}) + \frac{1}{2}\varepsilon \cdot \eta \cdot \left(\frac{1}{d(\mathbf{C})} - \frac{1}{d_0}\right)^2 \tag{8.55}$$

unde $\varepsilon = 0$, dacă $d(\mathbf{C}) > d_0$, respectiv $\varepsilon = 1$, dacă $d(\mathbf{C}) \le d_0$. Generarea mişcării se obţine prin integrarea ecuaţiei diferenţiale $\vec{v} = -\vec{\nabla}U(\mathbf{C})$:

$$\begin{cases} \dot{x} = \xi \cdot (x_{final} - x) + \varepsilon \cdot \eta \cdot \dfrac{d_0 - d}{d_0 d^3} \cdot \dfrac{\partial d}{\partial x} \\ \dot{y} = \xi \cdot (y_{final} - y) + \varepsilon \cdot \eta \cdot \dfrac{d_0 - d}{d_0 d^3} \cdot \dfrac{\partial d}{\partial y} \end{cases} \tag{8.56}$$

Se obţin patru zone (fig.8.11):

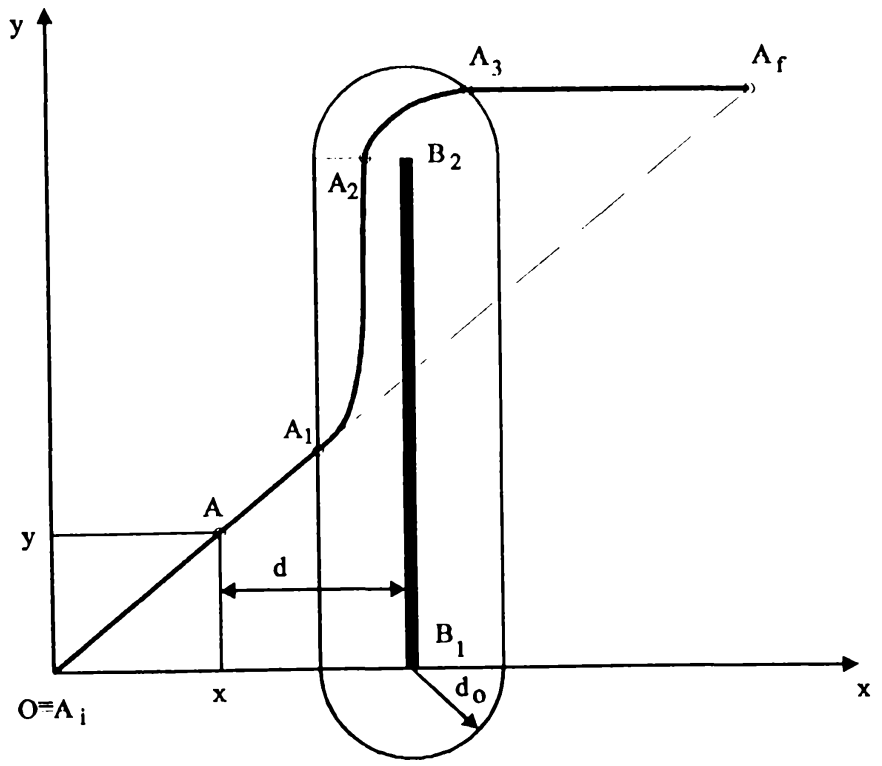


Fig.8.11. Calculul traiectoriei în cazul în care obstacolul B este o bară

BUPT

•*zona 1*: Traiectoria este segmentul $[A_{initial}A_1]$, neperturbat de obstacol. Sistemul (8.56) se scrie:

$$\begin{cases} \dot{x} = \xi \cdot (x_{final} - x) \\ \dot{y} = \xi \cdot (y_{final} - y) \end{cases} \tag{8.57}$$

cu condițiile inițiale: x(0)=0; y(0)=0.

•*zona 2*: Traiectoria este porțiunea A_1A_2, în zona de influență unde d = a - x. Sistemul (8.56) se scrie:

$$\begin{cases} \dot{x} = \xi \cdot (x_{final} - x) - \eta \cdot \dfrac{d_0 - (a - x)}{d_0 (a - x)^3} \\ \dot{y} = \xi \cdot (y_{final} - y) \end{cases} \tag{8.58}$$

•*zona 3*: Traiectoria este porțiunea A_2A_3, unde distanța de la A la obstacol este AB_2. Sistemul (8.56) se scrie:

$$\begin{cases} \dot{x} = \xi \cdot (x_{final} - x) - \eta \cdot \dfrac{d_0 - d}{d_0 d^4} \cdot (a - x) \\ \dot{y} = \xi \cdot (y_{final} - y) - \eta \cdot \dfrac{d_0 - d}{d_0 d^4} \cdot (b - y) \end{cases} \tag{8.59}$$

unde $d = \sqrt{(a - x)^2 + +(b - y)^2}$.

•*zona 4*: Traiectoria este segmentul $[A_3A_{final}]$. Sistemul are aceeaşi formă ca pentru zona 1

8.6. POTENȚIALE ELIPTICE

Un alt mod de abordare privind reducerea numărului şi mărimii minimurilor locale ale funcției potențial U constă în definirea lui U ca o combinație a potențialului atractiv având o simetrie în jurul C_{final} şi un potențial repulsiv eliptic.

Ideea de bază este de a defini potențialul repulsiv în jurul unui obstacol în aşa fel încât conturul echipotențial converge către frontiera C-obstacolului când distanța până la frontiera unui C-obstacol tinde către 0, şi către o suprafață sferică când distanța până la frontiera C-obstacolului creşte. Se dezvoltă această idee într-un spațiu SC bi-dimensional.

Se consideră cazul în care în SCB se află un singur C-obstacol de formă dreptunghiulară şi configurația $\mathbf{C}_{final}$ se află pe o axă de simetrie a lui SCB (fig.8.12).

Se defineşte în spațiul liber SC_{liber} funcția potențial $U = U_{atr} + U_{rep}$. Contururile echipotențiale ale potențialului U_{atr} sunt cercuri concentrice cu centrul în $\mathbf{C}_{final}$. În interiorul distanței de influență d_0 a C-obstacolului SCB, contururile echipotențiale ale U_{rep} sunt "dreptunghiuri cu colțurile rotunjite".

Figura 8.13 ilustrează modul în care compunerea lui U_{atr} cu U_{rep} crează un minim local al lui U de-a lungul axei de simetrie a lui SCB, care conține $\mathbf{C}_{final}$.

Se consideră segmentul PQ. Punctul P aparține frontierei lui SCB, în timp ce $Q \to \infty$. Fie o configurație $\mathbf{C} \in PQ$ la o distanță finită nenulă față de P. Potențialul $U(\mathbf{C})$ este finit. Când $\mathbf{C} \to P$ sau $\mathbf{C} \to Q$, $U(\mathbf{C}) \to \infty$. Fie $\mathbf{C} \in PQ$ la o distanță finită nenulă față de P, şi $\gamma_{rep}(\mathbf{C})$ conturul echipotențial al lui U_{rep} ce conține **C**, şi γ_{atr} (**C**) conturul echipotențial a

BUPT

lui U_{atr} care este tangent la $\gamma_{rep}(\mathbf{C})$ în două puncte distincte B_1 şi B_2. Cât timp robotul se mişcă de-a lungul $\gamma_{rep}(\mathbf{C})$ de la B_1 la B_2, potenţialul U scade până în punctul **C**, după care creşte până când este atins punctul B_2

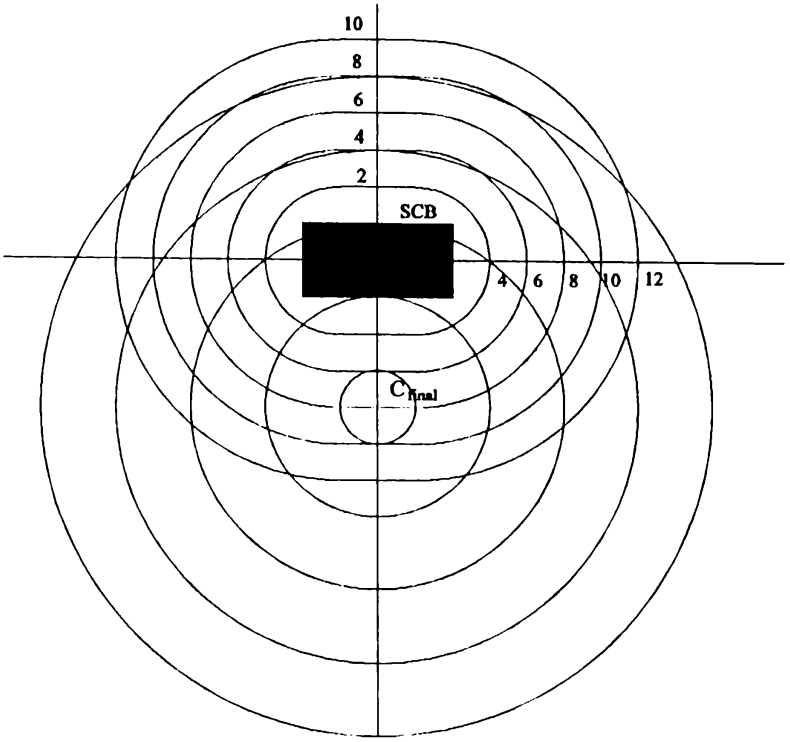


Fig.8.12. Contururile echipotenţiale ale câmpului potenţial atractiv sunt cercuri centrate în $\mathbf{C}_{final}$. Contururile echipotenţiale ale câmpului potenţial repulsiv sunt de forma rectangulară cu colţurile rotunjite.

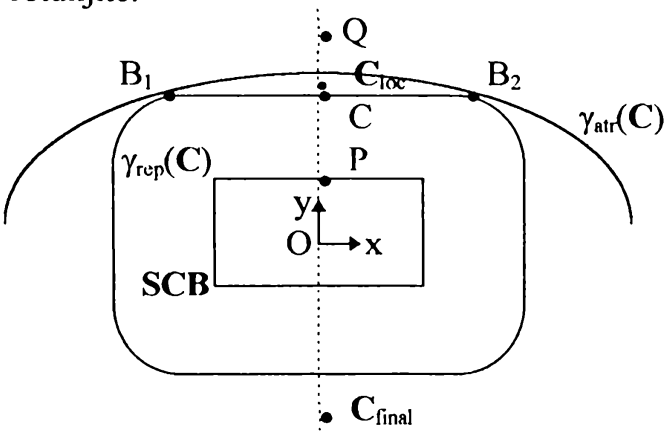


Fig.8.13. Minim local situat pe axa de simetrie a C-obstacolului SCB.

Deci, funcţia potenţial U are un minim local în configuraţia $\mathbf{C}_{loc}$ localizată pe dreapta PQ între P şi Q. Acest minim rezultă din faptul că raza de curbură a liniei echipotenţiale repulsive în $\mathbf{C}_{loc}$ este mai mare decât raza de curbură a liniei echipotenţiale atractive în $\mathbf{C}_{loc}$. Din acest motiv, potenţialul total este neconvex deşi ambele potenţiale, atractiv şi repulsiv, sunt convexe.

Această remarcă sugerează că se poate aproxima regiunea C-obstacolelor cu o colecţie de discuri disjuncte. Dacă C_{final} se află în afara oricăruia din aceste discuri, putem selecta distanţa de influenţă d_0 mai mică decât minimul dintre jumătatea distanţei minime

BUPT

dintre oricare două discuri şi distanţa dintre $\mathbf{C}_{final}$ şi cel mai apropiat disc. Apoi, este sigur că raza de curbură a liniei echipotenţiale repulsive în orice configuraţie **C** este mai mică decât raza de curbură a liniei echipotenţiale atractive în **C**. Potenţialul total este o funcţie de navigaţie propusă anterior (8.4.1). În orice caz, dezavantajul aproximării regiunii C-obstacolelor cu discuri disjuncte este posibil să împiedice accesul robotului la o submulţime mare a spaţiului liber. Acesta este un caz particular când regiunea C-obstacolelor este formată din obiecte conglomerate.

8.6.1. CALCULUL MINIMULUI LOCAL PENTRU C_{loc} DIN EXEMPLUL PREZENTAT

Se consideră un mobil punctiform *A* care se deplasează în planul xOy. Mobilul porneşte din originea sistemului de referinţă. Obstacolul din spaţiul de lucru este un dreptunghi. În banda verticală de deasupra dreptunghiului, potenţialul artificial rezultant se poate scrie [T12]:

$$U(x,y)=\frac{1}{2}\xi\cdot\left(x^2+y^2\right)+\frac{1}{2}\eta\cdot\frac{1}{(y-a)^2} \tag{8.60}$$

Forţa rezultantă se obţine din relaţia $\vec{F}(x,y)=-\vec{\nabla}U(x,y)$:

$$\mathbf{F}(x,y)=\left[-\xi\cdot x \qquad -\xi\cdot y+\frac{\eta}{(y-a)^3}\right]^T \tag{8.61}$$

Forţa rezultantă se anulează în punctul $A_0(x_0, y_0)$ de coordonate: $x_0=0$ şi $y_0=\dfrac{\eta}{\xi\cdot(y-a)^3}$. Acest punct corespunde unui minim local al câmpului potenţial. Dezvoltând în serie Taylor de ordinul 2 în vecinătatea lui $A_0(x_0, y_0)$, rezultă:

$$U(A_0+\delta A)=U(A_0)+\left(\vec{\nabla}U\big|_{A_0}\cdot\delta A\right)+\frac{1}{2}\left(\delta A\cdot H\big|_{A_0}\cdot\delta A\right) \tag{8.62}$$

Punctul A_0 corespunde unui minim local (în sens strict), dacă există o vecinătate a lui A_0 astfel încât pentru $A\neq A_0$, în această vecinătate $U(A_0) < U(A)$. Aceasta se traduce prin următoarele două proprietăţi:

- $\vec{\nabla}U\big|_{A_0}=0$; (8.63)
- matricea hessiană $H\big|_{A_0}$ a derivatelor parţiale de ordinul 2 este definită pozitivă.

Potenţialul U verifică aceste două condiţii în punctul A_0:

$$H\big|_{A_0}=\begin{bmatrix}\dfrac{\partial^2U}{\partial x^2} & \dfrac{\partial^2U}{\partial x\partial y}\\ \dfrac{\partial^2U}{\partial x\partial y} & \dfrac{\partial^2U}{\partial y^2}\end{bmatrix}=\begin{bmatrix}\xi & 0\\ 0 & \xi\left(1+\dfrac{3y_0}{y_0-a}\right)\end{bmatrix} \tag{8.64}$$

Un compromis între potenţialul repulsiv definit în (8.2.1.2), care tinde să creeze un număr mare de minimuri locale, şi potenţialul repulsiv circular prezentat mai sus, care împiedică accesul la o submulţime largă a lui SC_{liber} îl constituie potenţialul repulsiv eliptic definit în continuare.

BUPT

8.6.2. POTENȚIALUL REPULSIV ÎN CAZUL UNEI SUPRAFEȚE DEFINITE A PRIORI

Se consideră din nou cazul simplu în care regiunea de C-obstacole SCB este rectangulară. Fie L şi H lungimile celei mai lungi, respectiv celei mai mici laturi ale lui SCB. Se consideră sistemul de coordonate în SC = $\mathbf{R}^2$ cu originea în centrul lui SCB şi axa Ox de-a lungul celei mai lungi laturi a lui SCB. Potențialul repulsiv eliptic generat de SCB este definit în 2 paşi: în primul pas se specifică contururile echipotențiale, iar în al doilea pas se defineşte potențialul în fiecare contur echipotențial [R10].

O n-elipsă (n≥1) este o curbă definită astfel:

$$\left(\frac{x}{a}\right)^{2n}+\left(\frac{y}{b}\right)^{2n}=1 \tag{8.65}$$

unde *a* este semi-axa mare, iar *b* este semi-axa mică. Dacă :

$$a=\frac{L}{2}\cdot\left(2^{\frac{1}{2n}}\right) \text{ şi } b=\frac{H}{2}\cdot\left(2^{\frac{1}{2n}}\right) \tag{8.66}$$

se obține o n-elipsă ce trece prin vârfurile lui SCB. Mai mult, aria dintre dreptunghi şi n-elipsă este minimă. Când n→∞, n-elipsa converge către frontiera lui SCB.

Pentru a face ca n-elipsa să devină cerc când n =1, se înmulțeşte termenul stâng cu $(b/a)^2$ şi se obține:

$$\left(\frac{x}{a}\right)^{2n}+\left(\frac{b}{a}\right)^{2}\cdot\left(\frac{y}{b}\right)^{2n}=1 \qquad (n\geq 1) \tag{8.67}$$

Această n-elipsă devine cerc pentru n=1. Pentru orice n≥1 ea conține SCB şi converge spre frontiera lui SCB pentru n→ ∞. Pentru un *n* dat, se tratează mărimea:

$$\xi_n(x,y)=\left[\left(\frac{x}{a}\right)^{2n}+\left(\frac{b}{a}\right)^{2}\cdot\left(\frac{y}{b}\right)^{2n}\right]^{\frac{1}{2n}}-1 \tag{8.68}$$

ca o pseudo-distanță dintre configurația (x,y) şi SCB. Ea este zero în curba n-eliptică şi creşte la infinit pentru configurațiile ce se îndepărtează de această curbă. Pentru fiecare valoare constantă K dată, curba $\xi_n(x,y)$ = K este luată ca echipotențial repulsiv, pentru anumiți *n* determinați din:

$$n=\frac{1}{1-e^{-\alpha K}} \tag{8.69}$$

unde α este un parametru ajustabil. Astfel, când K→ ∞, n→1 şi echipotențialul tinde spre cerc. Când K→ 0 şi n → ∞, echipotențialul tinde către frontiera lui SCB. Parametrul α determină cât de repede creşte *n* când K → 0.

Figura 8.14. arată contururile echipotențiale în jurul lui SCB pentru diferite valori ale lui K. Fie R(K) conturul echipotențial repulsiv care corespunde pseudo-distanței K. Forma contururilor echipotențiale fiind specificată, se defineşte valoarea potențialului repulsiv în fiecare dintre ele. Este de dorit ca potențialul repulsiv al lui SCB să aibă un domeniu limitat de influență, invers proporțional cu distanța la C-obstacol. Acest lucru poate fi (aproape) realizat luând:

$$\forall(x,y)\in R(K): U_{rep}(x,y)=\eta\cdot\frac{c^{-\beta K}}{K} \tag{8.70}$$

unde η este un factor scalar. Termenul exponențial face ca potențialul să coboare spre zero mai repede decât K^{-1}, când K creşte.

BUPT

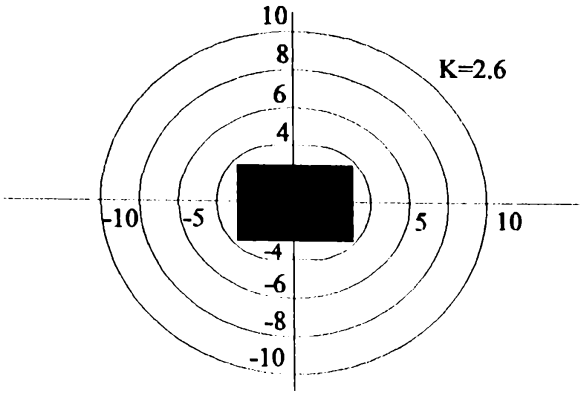


Fig.8.14. Contururi echipotențiale eliptice pentru K luând valori între 0.1 şi 2.6 şi α=1.5. Aceste contururi sunt apropiate de forma rectangulară când K este mic şi aproape circulare când K este mare.

Când funcția potențial repulsiv de mai sus se adună cu potențialul atractiv (cu $\mathbf{C}_{loc}$ ca în fig.8.13), poate apărea un minim local "în susul" lui SCB. Oricum, mărimea domeniului de atracție a acestui minim poate fi modificată setând adecvat η, α şi β. Este posibilă evitarea minimului local făcând minimul potențialului total de-a lungul axei y în punctul $(0, y_0)$ în care raza de curbură a conturului echipotențial repulsiv este mai mică sau egală cu raza de curbură a conturului echipotențialului atractiv (în plan, $(0, y_0)$ este un punct "şa").

Cu toate acestea, potențialele eliptice au mai multe dezavantaje care fac ca punerea lor în practică să fie o problemă dificilă. În primul rând, ele pot fi generalizate doar în cazul formelor simple convexe în spații bi- şi tri-dimensionale, utilizând formulări suprapătratice cu parametri scalari neconstanți. În al doilea rând, ele necesită ca C-obstacolele să fie suficient de depărtate unul față de celălalt, încât nici unul dintre ele să nu fie în interiorul distanței de influență a celuilalt. În al treilea rând, definiția lor nu furnizează o cale directă de calcul al potențialului într-o configurație dată, ele necesitând interpolări între valori precalculate în configurații apropiate. De asemenea, se poate calcula potențialul repulsiv eliptic în spațiul de lucru şi aplica acest potențial la selectarea punctelor de control ale robotului şi astfel se poate construi un potențial în spațiul configurațiilor SC.

8.6.3. Exemplu de aplicare a potențialului repulsiv eliptic în cazul unei suprafețe definite a priori, într-un spațiu bi-dimensional

Suprafața (S) a obiectului (obstacolului) este definită prin ecuația sa carteziană $f(x,y,z) = f(\mathbf{X}) = 0$. Expresia potențialului repulsiv se consideră de forma:

$$U_{rep}(\mathbf{X}) = \begin{cases} \frac{1}{2}\left[\frac{1}{f(\mathbf{X})} - \frac{1}{f(x_0)}\right]^2 & f(\mathbf{X}) \le f(x_0) \\ 0 & f(\mathbf{X}) > f(x_0) \end{cases} \qquad (8.71)$$

unde η este un scalar pozitiv, iar $f(x_0)$ este un scalar care defineşte zona de influență a potențialului repulsiv. Aceasta este delimitată de suprafața (S'), de ecuație:

$$g(x') = f(x') - f(x_0) = 0 \qquad (8.72)$$

Potențialul repusiv devine infinit dacă vectorul de poziție $\mathbf{X}$ este astfel încât $f(\mathbf{X}) = 0$, deci, dacă A aparține suprafeței (S), ceea ce va evita ca mobilul A să vină în contact cu obstacolul.

BUPT

Având în vedere cele prezentate în paragraful precedent, obstacolele simple, dar foarte diferite (paralelipiped, cilindru de lungime finită, con), se pot modela printr-o singură ecuație de suprafață, numită n-elipsoid:

$$f(x) = \left(\frac{x}{a}\right)^{2n} + \left(\frac{y}{b}\right)^{2n} + \left(\frac{z}{c}\right)^{2n} - 1 = 0 \tag{8.73}$$

Într-un spațiu bi-dimensional, se presupune obstacolul *B* modelat printr-un cerc cu centru în O şi de rază R. Se pot defini cele două suprafețe (S) şi (S') (fig.8.15):

$$(S) \rightarrow f(x) = x^2 + y^2 - R^2 = 0 \tag{8.74}$$

Notând $f(x_0) = R'^2 - R^2$, unde R' > R, rezultă:

$$(S') \rightarrow g(x') = x'^2 + y'^2 - R^2 - (R'^2 - R^2) = x'^2 + y'^2 - R'^2 = 0 \tag{8.75}$$

Suprafața (S') este deci un cerc cu centrul în O şi de rază R'.

Pentru acest obstacol şi mobilul care pleacă din $\mathbf{C}_{initial}$, în funcție de domeniul (D) cuprins între (S) şi (S'), rezultă:

• pentru A aflat în exteriorul domeniului (D), mişcarea este rectilinie, definită prin relația:

$$\vec{v} = -\xi\overrightarrow{C_{final}C} = \xi\overrightarrow{CC_{final}}\ ; \tag{8.76}$$

• pentru A aflat în domeniului (D), mişcarea este definită prin ecuțiile următoare:

$$\begin{cases} v_x = \dot{x} = -2\dfrac{\eta}{R'^2 - R^2} \cdot \dfrac{x\left(x^2 + y^2 - R'^2\right)}{\left(x^2 + y^2 - R^2\right)^3} - \xi\left(x - x_{final}\right) \\ v_y = \dot{y} = -2\dfrac{\eta}{R'^2 - R^2} \cdot \dfrac{y\left(x^2 + y^2 - R'^2\right)}{\left(x^2 + y^2 - R^2\right)^3} - \xi\left(y - y_{final}\right) \end{cases} \tag{8.77}$$

S-a obținut de fapt un sistem de ecuații diferențiale de gradul I, de forma $\dot{\mathbf{X}} = f(\mathbf{X})$, care se integrează numeric. Forma traiectoriei care evită obstacolul este reprezentată în (fig. 8.15.a).

Dacă în exemplul prezentat $[\mathbf{C}_{initial}\mathbf{C}_{final}]$ este "axa de simetrie" a câmpului potențial, când mobilul ajunge în interiorul domeniului (D) rămâne blocat, deoarece găseşte în E o poziție de echilibru unde $[U_{atr}(\mathbf{X}) + U_{rep}(\mathbf{X})]$ este staționar (fig.8.15.b).

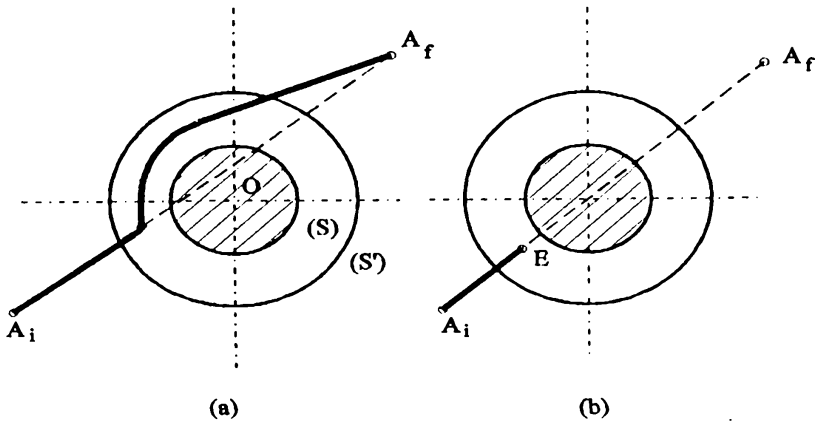


Fig.8.15. Exemplu de aplicare a potențialului repulsiv eliptic
în cazul unei suprafețe definite a priori, într-un spațiu bi-dimensional

BUPT

8.7. PROGRAM PENTRU GENERAREA UNEI TRAIECTORII PLANE PRIN METODA CÂMPULUI POTENȚIAL

8.7.1. PREZENTARE GENERALĂ

Programul elaborat permite simularea mişcării unui robot mobil, de la o configurație inițială, la o configurație finală, evitând coliziunea cu obstacolele din spațiul de lucru, sub influența unui potențial artificial al cărui gradient negativ generează forța rezultantă sub acțiunea căreia are loc mişcarea (8.2). Programul permite simularea mişcării unui robot într-un spațiul de lucru bi-dimensional, prin metoda câmpului potențial, când se cunosc configurațiile inițială, respectiv finală şi regiunea C-obstacolelor. Pentru aceasta utilizatorul trebuie să selecteze configurația inițială şi configurația finală prin punctare cu mouse-ul, să editeze C-obstacolele, să seteze parametrii funcțiilor potențial şi frecvența iterațiilor pentru timer, după care trebuie să trimită comanda de start robotului. Setarea parametrilor funcțiilor potențial se poate face oricând în timpul folosirii programului. În cazul atingerii unui minim local, programul atrage atenția utilizatorului că robotul s-a blocat într-un minim local, iar pe baza unui dialog, utilizatorul hotărăşte dacă se continuă iterația sau nu. Dacă se doreşte continuarea generării drumului, dar robotul nu reuşeşte să scape din minimul local, programul raportează eşec şi iterația se întrerupe. Programul se mai poate opri din iterație şi dacă se dă comanda **Stop Robot**.

C-obstacolele se editează cu ajutorul mouse-ului, prin punctarea vârfurilor acestora. Se verifică restricția de neintersectare a laturilor unui poligon. De aceea, dacă utlizatorul specifică un vârf, dar laturile ce se obțin prin unirea punctului curent cu ultimul vârf şi primul vârf se intersectează cu alte laturi neadiacente, punctul respectiv nu se ia în considerare. Abandonarea editării se face prin apăsarea butonului dreapta al mouse-ului sau dacă numărul de vârfuri depăşeşte un număr maxim admis. De asemenea, C-obstacolele se pot translata, şterge, copia/restaura sau se poate selecta C-obstacolul curent. Lucrul cu C-obstacolele se poate face şi în timpul mişcării robotului, dar trebuie să avem grijă să nu se suprapună C-obstacolul translatat peste robot. Acesta este de fapt un prim pas pentru implementarea algoritmului în cazul în care obstacolele din spațiul de lucru sunt mobile. Se permite de asemenea salvarea/restaurarea sesiunilor de lucru. Pe baza unui dialog se introduce calea şi numele fişierului ce se salvează/restaurează.

Programul necesită existența mediului Windows 3.1 sau Windows'95 şi mouse. A fost implementat şi testat cu succes pe un calculator 80486 cu 8M RAM şi placă grafică cu 512K RAM în mediul Windows'95. Au apărut probleme de reprezentare a regiunilor delimitate de poligoane şi laturile acestora, respectiv a vectorilor forțelor derivate din câmpul potențial potențial pentru plăci video mai puțin performante şi în cazul 4M RAM. De asemenea, în cazul alegerii unei frecvențe de iterație prea mare (perioada dintre două iterații foarte mică) pot apărea erori din cauza funcționării îngreunate a sistemului multitasking.

8.7.2 EXEMPLE DE UTILIZARE

În continuare, sunt prezentate câteva imagini capturate în timpul rulării programului de simulare, pentru anumite situații semnificative:

- un drum generat de robot, în cazul unor obstacole convexe de mici dimensiuni.

BUPT

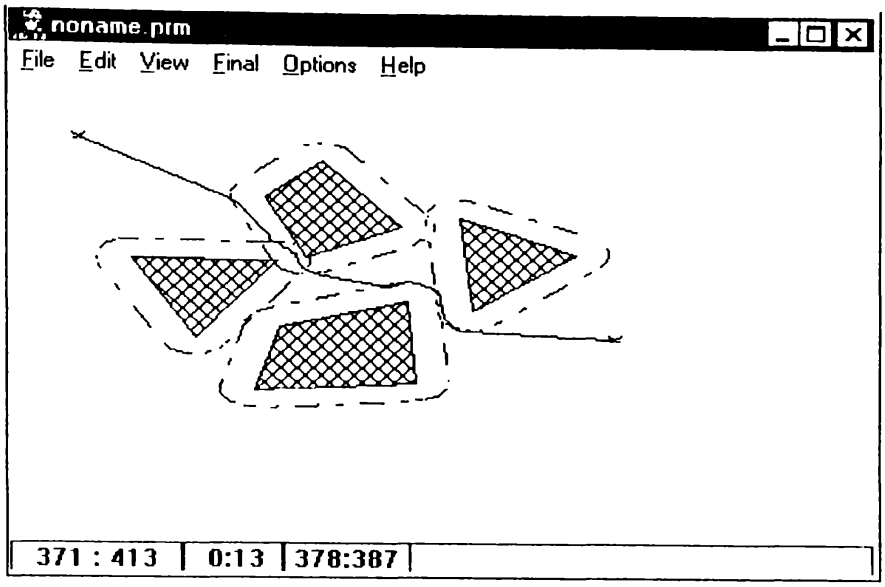


Fig.8.16. Traiectoria obținută

• cazul atingerii unui minim local:

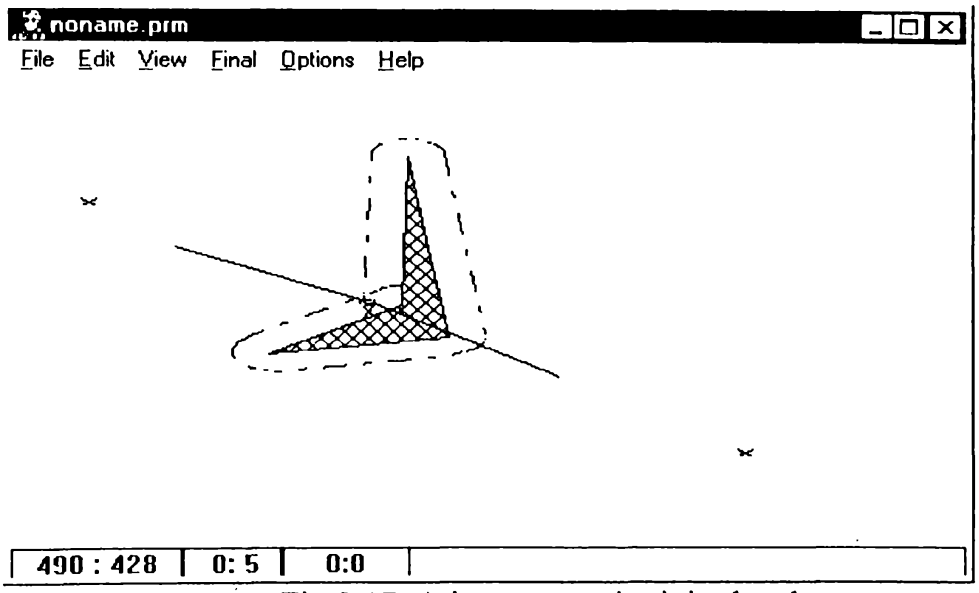


Fig.8.17. Atingerea unui minim local

În acest caz, forța repulsivă anulează forța atractivă, rezultând un minim local.

De fapt, robotul nu se va opri exact în minimul local, ci se vor executa niște oscilații în jurul acestuia. Astfel, se evidențiază necesitatea identificării situației atingerii unui minim local.

• compunerea forțelor repulsivă și atractivă:

BUPT

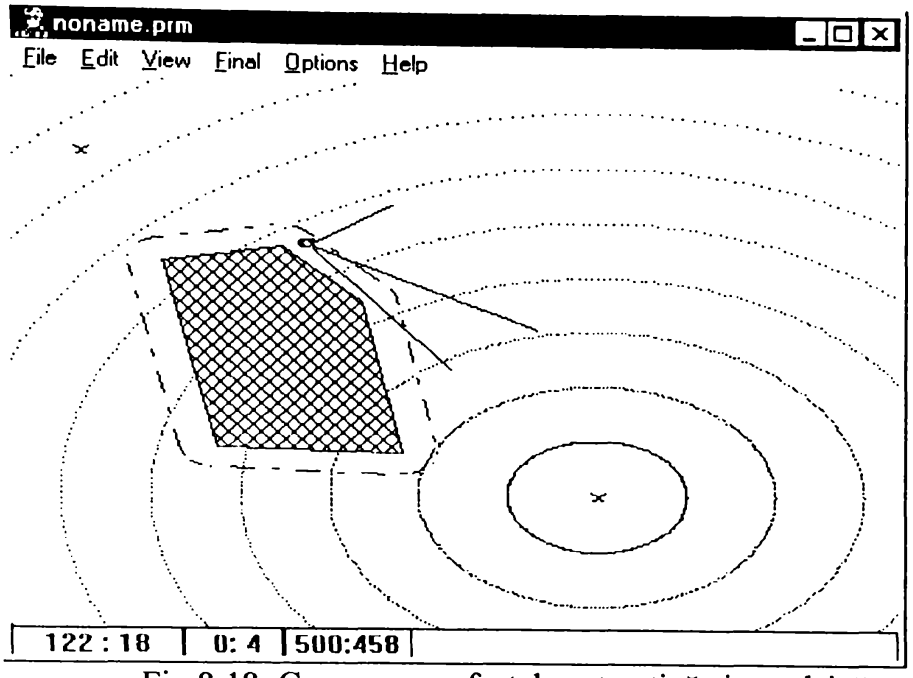


Fig.8.18. Compunerea forțelor atractivă şi repulsivă

•situație de blocare: robotul nu poate să ajungă în configurația finală deorece aceasta segăseşte în câmpul repulsiv al obstacolului.

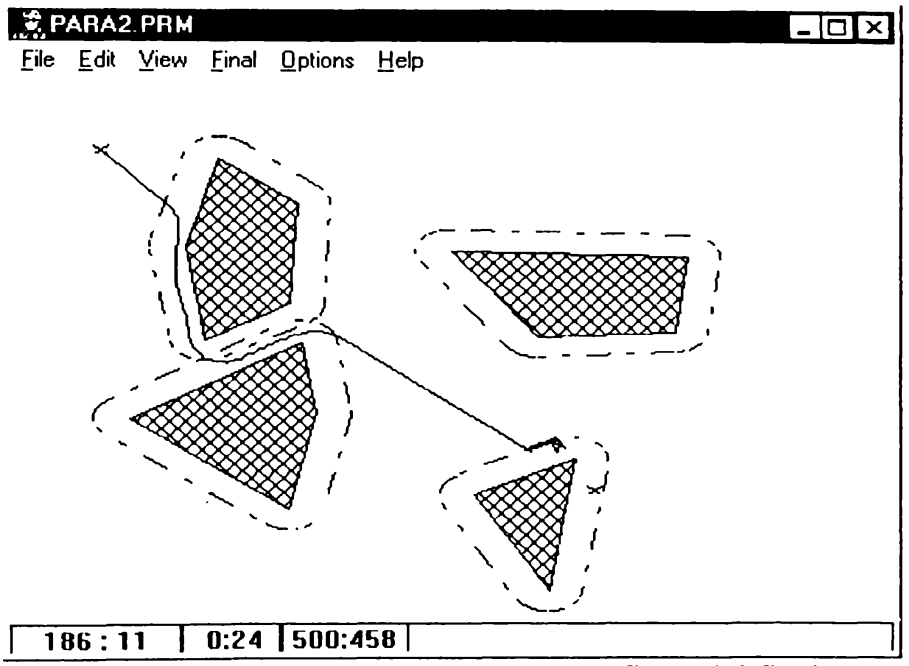


Fig.8.19. Blocaj într-o vecinătate a configurației finale

Programul realizat implementează cu succes majoritatea conceptelor descrise în paragrafele (8.2) şi (8.3) şi permite simularea mişcării unui robot de la configurația inițială la cea finală, sub influența câmpului potențial. Totuşi, nu s-a putut realiza o funcție potențial sigură, care să nu blocheze robotul într-un minim local.

Programul este flexibil şi uşor de utilizat. Se pot construi noi funcții potențial care pot fi adăugate celor existente. În cazul în care se realizează o altă metodă de scoatere a robotului din minimurile locale, se poate selecta care dintre metode să fie activată. Este

BUPT

posibil ca într-un caz să dea rezultate bune o metodă, iar în alt caz să trebuiască să fie folosită altă metodă. Selectarea metodei folosite se poate face on-line printr-un dialog cu robotul.

Programul mai poate fi îmbunătățit cu funcții de listare la imprimantă a sesiunii de lucru, respectiv se mai pot implementa şi facilități de help on-line.

Deoarece robotul nu are un model aprioric al obstacolelor, acestea fiind sesizate doar în cursul mişcării, se poate studia şi cazul când obstacolele nu sunt statice ci se mişcă pe o anumită traiectorie, după o anumită lege de mişcare sau aleator. În acest caz, traiectoria fiind generată iterativ, pe măsură ce robotul "descoperă" obstacolul, trebuie studiată posibilitatea existenței coliziunilor. Metoda, asociată cu un algoritm rapid de calcul al distanței, poate fi implementată on-line.

8.8. CONCLUZII

Metoda câmpului potențial este simplă şi poate da rezultate bune. Această metodă este frecvent utilizată pentru calculul traiectoriei libere a unui mobil deoarece permite să se ia în considerare, într-o manieră simplă, obstacolele neprevăzute şi să coordoneze mişcările mai multor roboți. Controlul coliziunilor poate fi realizat în cadrul acestei metode, printr-un ansamblu de forțe fictive aplicate punctului (punctelor) robotului care realizează distanța minimă până la obstacole. Fiecare din aceste forțe $\mathbf{F}_k$ este de fapt tradusă, prin intermediul Jacobianului robotului, ($\mathbf{J}_k^T \mathbf{F}_k$) la nivelul cuplelor cinematice [D16], [T9].

Această metodă a fost prezentată inițial pentru comanda dinamică a roboților, dar ea poate fi utilizată simplu pentru calculul unei traiectorii, interpretând direct gradientul negativ al potențialului ca o accelerație sau ca o viteză a parametrului de configurație. Totuşi, deoarece metoda se bazează pe calculul minimului unei funcții, care include în acelaşi timp şi un criteriu ce descrie sarcina robotului, precum şi termeni al căror efect este de a respinge mobilele de către obstacole, este dificil de controlat ansamblul mărimilor semnificative ale problemei.

În practică, este delicat să se "ajusteze" coeficienții utilizați pentru ponderea termenului atractiv şi a termenilor repulsivi. Astfel, este dificil de stabilit distanța de influență, începând de la care, îi este interzis robotului să se apropie de obstacol.

În prezent, cele mai folosite tehnici de planificare bazate pe metodele de câmp potențial utilizează funcțiile de navigație. Nici una din metodele de planificare ce folosesc câmpul potențial nu oferă însă un algoritm care să garanteze atingerea configurației finale în orice implementare practică. Marele dezavantaj al metodelor de câmp potențial rămâne riscul de blocaj într-un minim local al câmpului potențial. Deşi există deja mai multe metode de evitare sau de "ieşire" dintr-un astfel de minim, această problemă constituie încă subiectul multor cercetări. Cu toate inconvenientele prezentate, metodele de câmp potențial sunt intens studiate şi aplicate, oferind rezultate bune într-un mare număr de situații.

Acest capitol a fost dedicat integral metodelor de câmp potențial, deoarece, dintre toate metodele de planificare, acestea sunt cele mai atractive, permit luarea în considerare a dinamicii robotului, respectiv a structurii mecanice a acestuia, permit implementarea on-line, putând fi aplicate şi în cazul în care obstacolele sunt mobile.

BUPT

CONSIDERAŢII FINALE. CONTRIBUŢII ORIGINALE. PERSPECTIVE

"Ştiinţa roboticii se naşte sub ochii noştri". Cele mai recente, şi totodată cele mai spectaculoase preocupări ale Roboticii au ca obiect sistemele de programare off-line bazate pe tehnicile realităţii virtuale şi roboţii de servicii (service-robots). Roboţii de servicii sunt roboţi mobili, lucrează în medii diferite, necesitând sisteme de navigaţie şi, mai ales, o siguranţă foarte mare în funcţionare.

Generarea şi controlul mişcării roboţilor industriali sau de servicii sunt imposibil de realizat fără stăpânirea în detaliu a tehnicilor şi metodelor de generare şi planificare a traiectoriilor. Lucrarea de faţă se încadrează deci într-un domeniu de mare actualitate în Robotică.

Un obiectiv major, în prezent, îl constituie comanda roboţilor autonomi. Realizarea acestui deziderat implică rezolvarea problemelor legate de planificarea şi generarea mişcării robotului, de precizia cu care este realizată sarcina acestuia, în directă legătură cu modalităţile de realizare a percepţiei şi controlului.

Fiecare capitol al tezei conţine contribuţii aduse de această lucrare în domeniul cercetării tehnicilor şi metodelor de generare a traiectoriilor roboţilor. Aceste contribuţii sunt prezentate în continuare, într-o manieră sintetică.

Capitolul 1. Problematica generării traiectoriilor

- Sintetizarea rezultatelor cercetărilor pe plan mondial în domeniul abordat de teză, astfel încât lucrarea rezultată constituie un material util celor care au preocupări în acest domeniu;
- Analiza avantajelor şi limitărilor diferitelor metode de generare a traiectoriilor, prezentate în literatura de specialitate, în scopul îmbunătăţirii acestora;
- Abordarea unitară a problematicii generării traiectoriilor;
- Prezentarea sintetică a funcţiilor generatorului de traiectorie şi relaţia acestuia cu planificatorul de sarcină, planificatorul de traiectorie, optimizatorul, sistemul de acţionare şi sistemul senzorial.

Capitolul 2. Metode de generare a traiectoriilor în spaţiul operaţional

- Adaptarea metodei Newton-Raphson pentru rezolvarea modelului geometric invers, în forma $\mathbf{q}=\mathbf{f}^{-1}(\mathbf{X})$.
- Aplicarea matricei conducere pentru generarea mişcării în spaţiul operaţional cartezian.
- Studiu comparativ al metodelor de generare a mişcării punctului caracteristic al unui robot, în timp minim, cu abatere controlată faţă de punctele impuse în coordonate carteziene, cu păstrarea constantă a orientării efectorului final.
- Elaborarea unui program de simulare a mişcării punctului caracteristic, pentru realizarea studiului comparativ al metodelor analizate. Programul permite simularea

BUPT

mişcării între orice situări din spaţiul operaţional tri-dimensional, cu orice viteze şi acceleraţii. Deplasarea între situări succesive se realizează liniar (cu viteză constantă), iar tranziţia între segmentele de traiectorie se realizează cu acceleraţie constantă. Se pot calcula şi optimiza timpii de parcurgere a traiectoriei, respectiv erorile de poziţionare, care trebuie să rămână în limitele stabilite. Considerând poziţiile impuse ca noduri, programul realizează şi alte tipuri de interpolări (Hermite, B-spline, Bezier) şi le compară cu interpolarea liniară.

•Stabilirea unui algoritm de calcul pentru interpolarea cvasi-liniară în spaţiul configuraţiilor, cu abatere controlată în spaţiul operaţional, pe baza unei metode propuse de R.Taylor.

Capitolul 3. Metode de generare a traiectoriilor în spaţiul cuplelor cinematice

•Abordarea sistematică a metodelor de generare a traiectoriilor în spaţiul cuplelor cinematice.

•Studiu comparativ al diferitelor tipuri de interpolări din punct de vedere al timpului de parcurgere a traiectoriei.

• Stabilirea unui algoritm de calcul pentru coordonarea cuplelor cinematice prin metoda timpului minim.

•Elaborarea unui program pentru simularea mişcării roboţilor în spaţiul cuplelor, conceput ca o sinteză originală a diferitelor metode de generare. Programul poate fi aplicat oricărui tip de robot. Selectarea metodei de generare se realizează pe baza unor aspecte legate de robot şi de mişcare. Traiectoria poate fi generată între puncte de oprire (interpolare polinomială de gradul 3 şi 5, profil triunghiular şi trapezoidal de viteză), sau de trecere (interpolare liniară, polinomială, funcţii spline), indiferent de numărul acestora. Programul realizează coordonarea cuplelor pe baza metodei timpului minim şi verifică încadrarea în limitele de viteză şi acceleraţie ale cuplelor.

Capitolul 4. Interpolarea traiectoriilor folosind funcţii spline

•Analiza sistematică a posibilităţilor de utilizare a funcţiilor spline pentru generarea mişcării roboţilor (avantaje şi limitări).

•Stabilirea unui algoritm şi elaborarea unui program original pentru generarea unei traiectorii ce trece prin *n* puncte de precizie, utilizând funcţii spline (4-3-...-3-4). Programul poate fi folosit pentru generarea mişcării oricărui robot, cu orice număr de grade de libertate, prin oricâte puncte de precizie. Programul calculează şi verifică vitezele şi acceleraţiile maxime ale cuplelor, calculează timpii necesari parcurgeii segmentelor de traiectorie, determină şi reprezintă funcţiile de interpolare. Programul realizat pe baza metodei funcţiilor spline 4-3-...-3-4 a fost utilizat la simularea generării mişcării robotului RTRTTR.

Capitolul 5. Generarea traiectoriei unui robot pe baza modelului dinamic

•Abordarea sintetică a folosirii modelului dinamic în Robotică;

•Determinarea şi rezolvarea modelelor dinamice, direct şi invers, pentru un robot de sudare, cu trei grade de libertate, de tip TTR, prin două metode: folosind ecuaţiile lui Lagrange, în maniera clasică din mecanica analitică şi folosind transformările omogene;

•Elaborarea modelelor geometrice şi dinamice, determinarea torsorului din cuple şi rezolvarea modelului dinamic direct în condiţii date de mişcare, pentru robotul KUKA

BUPT

tip 364/10/0, aflat în funcțiune la Universitatea Artois (Franța) din Bethune, în laboratorul de Productică și Robotică.

• Stabilirea unui algoritm pentru calculul torsorului din cuple și pentru determinarea variației torsorului de reducere la modificarea sistemului de referință;

• Elaborarea unui program pentru calculul torsorului din cuple, respectiv al torsorilor care acționează asupra elementelor unui robot. Torsorul din cuplele unui robot, respectiv torsorii care acționează asupra elementelor se pot calcula pornind de la torsorul care acționează asupra efectorului final și folosind Jacobianul robotului și Jacobianul sistemului de referintă atașat elementului. Programul poate fi folosit pentru orice robot, dacă este implementat modelul geometric al acestuia. Aplicarea algoritmului și a programului de calcul al torsorilor s-a realizat pentru cazul concret al robotului RTRTTR.

• Elaborarea unei metode de determinare a preciziei cinematice de parcurgere a traiectoriei unui robot pe baza modelului diferențial al acestuia. Metoda elaborată poate fi aplicată oricărui robot. Exemplificarea este realizată pentru determinarea preciziei de parcurgere a traiectoriei la robotul de sudare de tip TTR.

• Elaborarea unei metode de determinare a preciziei dinamice de parcurgere a traiectoriei unui robot pe baza modelului dinamic al acestuia. Metoda poate fi aplicată la orice tip de robot. În cadrul tezei este exemplificată pentru robotul de sudare de tip TTR

• Elaborarea unei metode pentru determinarea influenței vibrațiilor din cuplele unui robot asupra poziției efectorului final. Exemplificarea metodei s-a realizat pentru determinarea influenței vibrațiilor din cuple asupra poziției efectorului final al unui robot de tip RT.

• Elaborarea unei metode și a unui algoritm pentru determinarea erorilor dinamice datorate deformațiilor elementelor unui robot.

Capitolul 6. Studiu experimental privind conducerea unui robot cu comandă punct cu punct pe o traiectorie continuă în spațiul operațional

• Determinarea spațiului de lucru al robotului SCORBOT ER-III;

• Elaborarea modelului geometric al robotului SCORBOT ER-III;

• Determinarea relațiilor dintre numărul de impulsuri și numărul de grade pentru fiecare ax al robotului SCORBOT ER-III;

• Generarea mișcării punct cu punct, pe o traiectorie continuă, interpolată în spațiul operațional, off-line, pentru trei tipuri de interpolări (liniară, spline 3, spline4), în două cazuri: cu păstrarea constantă, respectiv cu modificarea orientării;

• Generarea mișcării punct cu punct, pe o traiectorie continuă, circulară, în plan și analiza abaterilor de la traiectoria prescrisă;

• Folosirea modelului geometric direct pentru evaluarea preciziei de generare a mișcării robotului SCORBOT ER-III.

Capitolul 7. Planificarea și generarea mișcării într-un spațiu cu obstacole

• Extinderea metodei grilei neomogene și transformarea acesteia într-o metodă de planificare globală, off-line;

• Elaborarea unui algoritm de modelare și planificare și a unui program de modelare a spațiului de lucru prin metoda grilei neomogene. Programul este realizat pentru un spațiu de lucru bi-dimensional, dar extensia la 3D este simplă, principiul fiind același. Spațiul de lucru poate fi populat cu oricâte obiecte, de orice formă. Programul

BUPT

construieşte C-obstacolele, le aproximează cu dreptunghiuri, construieşte grila şi graful neorientat, cercetează graful pentru determinarea unui drum liber şi, eventual îl caută pe cel mai scurt.

• Elaborarea unui program pentru calculul frontierelor C-obstacolelor, în cazul unui robot mobil, în mişcare de translaţie, într-un spaţiu de lucru bi-dimensional, pentru orice orientare a robotului. Programul poate fi aplicat în cazul când robotul este modelat ca un disc, sau ca un poligon în mişcare de translaţie. Programul permite modificarea orientării robotului şi recalculează frontierele C-obstacolului. Numărul maxim al obiectelor din spaţiul de lucru este 10, dar poate fi modificat.

• Stabilirea unui algoritm şi realizarea unui program de planificare a mişcării unui robot mobil, în mişcare de translaţie, într-un spaţiu de lucru bidimensional, prin metoda grafului vizibilităţii. Programul permite reprezentarea C-obstacolelor şi stabilirea configuraţiilor iniţială şi finală, construieşte şi cercetează graful de vizibilitate redus şi determină drumul minim între cele două configuraţii. Algoritmul obişnuit al metodei a fost extins şi pentru cazul obiectelor modelate prin segmente de dreaptă şi arce. Timpul de lucru al programului depinde de numărul şi de geometria C-obstacolelor.

• Stabilirea unui algoritm şi elaborarea unui program pentru planificarea mişcării unui robot mobil, modelat ca un segment de dreaptă, în mişcare plană, într-un spaţiu de lucru bi-dimensional, pe principiul metodei descompunerii celulare exacte. Programul permite modificarea lungimii robotului. Utilizatorul stabileşte configuraţiile iniţială şi finală, precum şi obstacolele din spaţiul de lucru. Programul reprezintă toate curbele critice, cercetează şi memorează regiunile necritice, verifică relaţia de adiacenţă dintre celule, construieşte graful de conexiune şi îl cercetează pentru determinarea canalului între configuraţia iniţială şi cea finală. Spaţiul de lucru poate fi mărginit sau nemărginit, configuraţia sa putând fi modificată după caz. Complexitatea algoritmului depinde de implementarea unor anumiţi paşi.

Capitolul 7. Planificarea mişcării roboţilor prin metodele câmpului potenţial

• Elaborarea unui program de planificare a mişcării unui robot mobil, în mişcare de translaţie, într-un spaţiu de lucru bi-dimensional, pe baza metodei depth-first planning şi aplicarea programului pentru planificarea mişcării unui robot tip maşină. Prin realizarea acestui program s-a realizat în mod original simularea mişcării unui robot tip maşină, în condiţiile respectării restricţiilor cinematice ale acestuia. Spaţiul de lucru poate avea orice configuraţie. Programul calculează potenţialele atractiv şi repulsiv, le însumează şi construieşte segmentele de traiectorie pornind din configuraţia iniţială şi terminând în cea finală. Pentru optimizarea traiectoriei, valorile parametrilor din expresiile potenţialelor atractiv şi repulsiv pot fi modificate.

• Elaborarea unui program pentru generarea unei traiectorii plane, prin metoda câmpului potenţial, pe principiul metodei best-first planning. Programul calculează forţele atractivă şi repulsivă şi determină direcţia de deplasare a robotului. Pentru tratarea minimurilor locale, programul este prevăzut cu o metodă de rezolvare a probemei (umplerea minimului local), dar pot fi implementate şi alte metode, şi anume, introducerea unei configuraţii intermediare pentru "evadarea" din minim. O altă facilitate pe care o oferă programul constă în modificarea poziţiei obstacolelor, pregătind astfel terenul pentru cazul obstacolelor mobile.

BUPT

Sinteza rezultatelor cercetării în domeniul generării şi planificării mişcării roboţilor, realizată pe baza unui studiu bibliografic aprofundat, împreună cu contribuţiile aduse la tehnicile şi metodele de generare şi planificare a traiectoriilor, prezentate ca un întreg în acestă lucrare, reflectă stadiul actual al rezolvării problemelor ridicate de acest capitol al Roboticii, dar conturează în acelaşi timp alte probleme, respectiv noi direcţii de cercetare în acest domeniu.

Din fiecare capitol al tezei rezultă problemele care definesc perspectiva acestui domeniu de cercetare. Elaborarea modelului dinamic al roboţilor, în condiţii cât mai apropiate de realitate (considerarea frecărilor dintre elemente, din cuple, jocurile, flexibilitatea elemcntelor, etc.) permite realizarea controlului optimal al mişcării. Aplicarea teoriei controlului optimal pentru generarea mişcării roboţilor poate conduce la noi tehnici şi metode de generare a traiectoriilor. Extinderea metodelor de planificare, în cadrul problemei dinamice a planificării, va pregăti "terenul" roboţilor mobili autonomi, pentru aplicaţii industriale, dar şi al roboţilor de servicii.

BUPT

BIBLIOGRAFIE

[A1] **Adamiec-Wojcik I., Wojciech S.,** *Application of a rigid finite element method in dynamic analysis of plane manipulator,* Mechanism and Machine Theory, Vol.28, No.3, 1995.

[A2] **Ahlberg H.H.**, *The Theory of Splines and Their Applications*, New-York: Academic, 1967.

[A3] **Aken Ir.L., Brussel H.,** *On Line Robot Trajectory Control in Joint Coordinates by Means of Imposed Acceleration Profiles*, Robotica, Vol.6, p.185-195, 1988.

[A4] **Albus J.S.,** *Brains, Behaviour and Robotics*, BYTE Publication Inc., 1981.

[A5] **Aldon M.J.**, *Elaboration Automatique du Modèle Dinamique des Robots*, RAIRO Automatique/Systems, Analysis and Control, Vol.16, No.3, p.205-220, 1982.

[A6] **Armstrong W.W.**, *A recursive solution to the equation of motion of an N-link manipulator*, Proceedings of the 5th World Congress on Theory of Machines and Mechanisms, Vol.2, p. 1343-1346, Montreal, PQ, Canada, 1979.

[A7] **Asada H., Slotine J.J.E.**, *Robot Analysis and Control*, New-York: Wiley, 1986.

[B1] **Baghat M.B., El-Shakery S.A.**, *A compact kinematic synthesis procedure for robot's manipulator arms*, Mechanisms and Machine Theory, Vol.28, No.3, p.407-415, 1993.

[B2] **Barbu V.**, *Metode matematice în optimizarea sistemelor diferențiale*, Editura Academiei RSR, Bucureşti, 1989.

[B3] **Barraquand J., Latombe J.C.,** *A Monte-Carlo Algorithm for Path-Planning with Many Degrees of Freedom*, Proceedings IEEE International Conference on Robotics and Automation, Cincinnati, OH, p.1712-1717, 1990.

[B4] **Barraquand J., Langlois B., Latombe J.C.,** *Robot Motion Planning with Many Degrees of Freedom and Dynamic Constraints*, Proceedings of the 5-th International Symposium on Robotics Research, Tokyo, p.74-83, 1989.

[B5] **Bartels R.H., Beaty J.C., Barski B.A.,** *Mathèmatique et C.A.O., Bêta-splines*, Edition Hermès, Paris, 1988.

[B6] **Bejczy A.K.**, *Robot arm dynamics and control*, Jet Propulsion Laboratory, Pasadena, CA, Tech. Memo 33-668, 1974.

[B7] **Binford T.O.,** *Exploratory Study of Computer-Integrated Assembly Systems*, Stanford A.I., Memo No. 285.4, 1977.

[B8] **Binford T.O.,** *Discussion of Trajectory Calculation Methods*, Exploratory Study of Computer Integrated Assembly Systems, Stanford Artificial Intelligence Laboratory, Progress Report, Memo AIM-285.4, Stanford, 1977.

[B9] **Blanchon J.L.,** *Génération et identification des lois de mouvements en robotique. Application à l'identification et à la correction de trajectoires du manipulateur PUMA 560*, Thèse de Docteur-Ingenieur, USTL, Montpellier, 1987.

[B10] **Bobrow J.E.**, *Optimal control of robotic manipulators*, Ph. D. dissertation, CA, 1982.

[B11] **Bobrow J.E., Dubowsky S., Gibson J.S**, *On the optimal control robotic manipulators with actuator constraints.*, Proceedings of American Control Conference, San Francisco, CA, p.782-787, 1983.

[B12] **Bobrow J.E., Dubowsky S., Gibson J.S.**, *Time-optimal control of robotic manipulators*, Symposion of the Theory and Practice of Robots and Manipulators, Udine, Italy, 1984.

BUPT

[B13] **Bobrow J.E., Dubowsky, S. and Gibson J.S.**, *Time optimal control of robotic manipulators along specified paths*, International Journal of Robotic Research, Vol. 4, No. 3, 1985.

[B14] **Bodner A.**, *Adding flexibility to the links of a rigid-link dynamic model of an articulated robot*, Robotica, Vol.7, p.165-168, 1989.

[B15] **Bogdanov I.**, *Problema conducerii unui robot industrial*, Simpozionul Național de Roboți Industriali, Timişoara, Ediția a XII-a, Vol.3, p.123-134, 1994.

[B16] **Bogdanov I.**, *Legătura spațiu-timp în conducerea unui robot industrial*, Simpozionul Național de Roboți Industriali, Timişoara, Ediția a XII-a, Vol.3, p.135-144, 1994.

[B17] **Boissonat J.-D., Faverjon B., Merlet J.-P.**, *Technique de la robotique. Perception et planification*, Edition Hermes, 1988.

[B18] **Bolles R., Paul R.**, *An Experimental System for Computer Controlled Mechanical Assembly*, Stanford Artificial Intelligence Laboratory, Memo AIM-220, Stanford University, Calif., 1973.

[B19] **Borenstein J., Koren V.**, *The vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots*, IEEE-Transactions on Robotics and Automation, Vol.7, No.3, p.278-288, 1991.

[B20] **Brady M.**, *Robot Motion: Planning and Control*, The MIT Press, 1982.

[B21] **Bremer H.**, *Combined position and force control for elastic robots*, Technical University of Munich.

[B22] **Bruyninckx H., Demey S., Dutre S., De Schutter J.**, *Kinematic Modell for Model-Based Compliant Motion in the Presence of Uncertainty*, International Journal of Robotics Research, Vol.14, No.5, p.465-482, 1995.

[C1] **Cai M.S.**, *Analisi della traiettoria del movimento di robot*, Internal Report, No.6, 1987.

[C2] **Cai M.S., Rovetta A.**, *A new approach of joint trajectories for cartesian path approximation of mechanical manipulators*, Meccanica, No.24, p.42-46, 1989.

[C3] **Cannon R.H., Schmitz E.**, *Initial experiments on the end-point control of a flexible one-link robot*, International Journal of Robotics Research, Vol.3, No.3, 1984.

[C4] **Carriker W.F., Khosla P.K., Krogh B.H.**, *Path Planning for Mobile Manipulators for Multiple Task Execution*, IEEE Transactions on Robotics and Automation, Vol.7, No.3, p.403-408, 1990.

[C5] **Cetinkunt S., Ittoop B.**, *Computer-Automated Symbolic Modeling of Dynamics of Robotic Manipulators with Flexible Links*, IEEE Transactions on Robotics and Automation, Vol.8, No.1, p.94-105, 1992.

[C6] **Chen Y., Desrochers A.A.**, *A Proof of the Structure of the Minimum-Time Control Law of a Robotic Manipulators Using a Hamiltonian Formulations*, IEEE Transactions on Robotics and Automation, Vol.6, No.3, p.388-393, 1990.

[C7] **Chircor M.**, *Calculul energiei consumate de robotul industrial la manipularea unei sarcini*, Acta Universitatis Cibiniensis, Vol.XIX, Seria Tehnică, Sibiu, p.89-92, 1995.

[C8] **Chircor M.**, *Calculul deplasărilor roboților industriali folosind notațiile Hartenberg-Denavit*, Acta Universitatis Cibiniensis, Vol.XIX, Seria Tehnică, Sibiu, p.93-98, 1995.

[C9] **Chirikjian G.S.I., Burdick, J.W.**, *A geometric approach to hyper-redundant manipulator obstacle avoidance*, Flexible mechanism, dynamic, and robot trajectories, DE-Vol.24, edited by S.Derby, M. McCarthy, A.Pisano, New-York, p.337-342,1990.

[C10] **Choi Y.J., Crane III C.D., Matthew G.K., Dufy J.**, *The geometry of interference with application to obstacle avoidance*, Flexible mechanism, dynamics, and robot trajectories, DE-Vol.24, edited by S. Derby, M. McCarthy, A. Pisano, New-York, p.327-336, 1990.

[C11] **Chou L.-S., Song S.-M.**, *Geometric work of manipulators and path planning based on minimum energy consumption*, Flexible mechanism, dynamics and robot trajectories, DE-Vol.24, edited by S. Derby, M. McCarthy, A. Pisano, New York, p.319-326, 1990.

BUPT

[C12] **Chou H.C., Sadler J.P.,** *Optimal location of robot trajectories for minimization of actuator torque*, Mechanism and Machine Theory, Vol.28, No.1, p.145-158, 1993.
[C13] **Coiffet Ph.,** *Robotique. Principes et Applications*, Éditons Hermès, Paris, 1986, 1992.
[C14] **Coiffet Ph.,** *Les Robots. Modélation et Commande*, Hermes Publishing, Paris, 1987.
[C15] **Coiffet Ph.,** *Les Robots. Interaction avec l'Environement*, Hermes Publishing, Paris, 1981.
[C16] **Cojocaru G., Kovacs F.,** Roboții în acțiune. Sistemele flexibile şi fabricația de serie, Editura Facla, Timişoara, 1985.
[C17] **Couffin P., Herbin J.-Y., Puzenat M.,** *C.A.O. en Mécanique*, Edition Armand Colin, 1987.
[C18] **Connolly, C.I., Burns, J.B., Weis, R.,** *Path Planning Using Laplace's Equations*, Proceedings IEEE International Conference on Robotics and Automation (Cincinnati, OH), p.2102-2106, 1990.
[C19] **Craig J.J.,** *Introduction to Robotics: Mechanics and Control*, Addition Wesley Publishing Company, 1986.

[D1] **Dahl O.,** *Torque limited path following by on-line trajectory time scaling*, Licentiate Thesis TFRT-3204, Department of Automation and Control, Lund Institute Technology, Lund, Sweden, 1989.
[D2] **Dahl O., Nielsen L.,** *Torque-Limited Path Following by On-Line Trajectory Time Scaling*, IEEE Transactions on Robotics and Automation, Vol. 6, No.5, p.554-561, 1990.
[D3] **Dahl O.,** *Path-Constrained Robot Control with Limited Torques- Experimental Evaluation*, IEEE Trans. on Robotics and Automation, Vol.10, No.5, p.658-669, 1994.
[D4] **Darabont Al., Iorga I., Văiteanu D., Simaschevici H.,** *Şocuri şi vibrații. Aplicații în tehnică*, Editura Tehnică, Bucureşti, 1988.
[D5] **Dauchez P., Zapata R., Fournier A.,** *Une Application Particulière du Modèle Cinématique à la Commande Coordonée de deux Robots Manipulateurs*, R.A.I.R.O. Automatique/Systems Analysis and Control, Vol.18, No.1, p.7-24, 1984.
[D6] **Davidoviciu A., Moangă A.,** *Roboți evoluați şi automatizări flexibile*, Seria: Tehnica la zi, Editura Tehnică, Bucureşti, 1985.
[D7] **Davidoviciu A., Drăgănoiu G.H., Moangă A.,** *Modelarea, simularea şi comanda manipulatoarelor şi roboților industriali*, Editura Tehnică, Bucureşti, 1986.
[D8] **Denavit I., Hartenberg R.S.,** *A kinematic notation for lower pair mechanisms based on matrices*, Journal of applied mechanisms, p.215-221, 1955.
[D9] **Dhanaraj C., Sharan A.M.,** *Efficient modelling of rigid link robot dynamic problems with friction*, Mechanism and Machine Theory, Vol.30, No.5, p.749-764, 1995.
[D10] **Diker H.,** *Energy efficient simusoidal path planning of robot manipulators*, Mechanism and Machine Theory, Vol.29, No.6, 1994.
[D11] **Dinu I.,** *Stabilirea preciziei cinematice a roboților industriali*, Acta Universitatis Cibiniensis, Sibiu, Seria Tehnică, Vol.XIX, p.83-88, 1995.
[D12] **Dobkin D.P., Kirkpatrik D.G.,** *A linear Algorithm for Determining the Separation of Convex Polyhedra*, Journal of Algorithmus, No.6, p.381-392, 1985.
[D13] **Dolga V., Gligor A.,** *Determinarea torsorului aplicat efectorului unui robot industrial*, Al XII-lea Simpozion Național deRoboți Industriali, Timişoara, Vol.4, p.187-194, 1994.
[D14] **Dombre E., Khalil W.,** *Modélislation et Commande des Robots*, Edition Hermès, Paris, 1988.
[D15] **Drăgănescu M.,** *A doua revoluție industrială. Microelectronica, automatică, informatică-factori determinanți*, Editura Tehnică, Bucureşti, 1980.
[D16] **Drăgulescu D., Toth-Taşcău M.,** *Elemente de inginerie mecanică. Introducere în cinematica roboților. Introducere în dinamica roboților*, Lito UPT, Timişoara, 1993.

BUPT

[D17] **Drăgulescu D., Toth-Taşcău M.**, *Planificarea traiectoriei unui robot prin metoda câmpului potenţial*, Al XII-lea Simpozion Naţional de Roboţi Industriali, Timişoara, Vol.3, p.39-48, 1994.

[D18] **Drăgulescu D., Toth-Taşcău M.,** *Kinematic Analysis, Joint Torques and Dynamics of a Welding Rrobot*, Analele Universităţii din Oradea, Fascicola Mecanică, p.17-24, 1994.

[D19] **Drăgulescu D., Toth-Taşcău M.,** Dynamic Study of a Welding Robot, Proceedings of the 25-th Conference on Production Engineering, Beograd, p.553-556, 1994.

[D20] **Drăgulescu D., Toth-Taşcău M.,** *Determinarea ecuaţiilor de mişcare pe baza modelului dinamic pentru un manipulator cu trei grade de libertate*, A XVIII-a Conferinţă de Mecanica Solidelor, Constanţa, Buletin, Vol.2, p.106-113, 1994.

[D21] **Drăgulescu D., Toth-Taşcău M.,** *Kinematic and Dynamic Study of a Welding Robot*, Second European Conference on Joining Technology, Florence, p.893-901, 1994.

[D22] **Drăgulescu D., Toth-Taşcău M.,** *Coordonarea cuplelor cinematice ale unui robot prin metoda timpului minimal*, Al XIII-lea Simpozion Naţional de Robotică, Timişoara,1996.

[D23] **Drăgulescu D., Toth-Taşcău M.,** *Generarea traiectoriei unui robot cu şase grade de libertate*, Analele Universităţii din Oradea, Oradea, Fascicola Mecanică, p.9-16, 1994.

[D24] **Drăgulescu D., Toth-Taşcău M.,** *Method to Generate an Imposed Trajectory for a Robot Having 6 Degrees of Freedom*, Proceedings of International Conference on Technical Informatics, Timişoara, Vol.2, p.15-21, 1994.

[D25] **Drăgulescu D., Toth-Taşcău M.,** *Studiul dinamicii unui robot de sudare cu trei grade de libertate*, Sesiunea de Comunicări Ştiinţifice-Concepţie, Tehnologie şi Management în Construcţia de Maşini, Iaşi, Vol.5, Sisteme flexibile şi roboţi industriali, p.69-75, 1992.

[D26] **Drăgulescu D., Toth-Taşcău M.,** *Modelarea roboţilor cu structură arborescentă şi lanţ cinematic închis*, Comunicările celei de-a VII-a Conferinţe Internaţionale de Inginerie Managerială şi Tehnologică, Timişoara, Vol.7, p.71-76, 1995.

[D27] **Drăgulescu D., Toth-Taşcău M.,** *Torque elements variation with respect to the reference system change*, Buletinul Ştiinţific al UTT, Seria Mecanică, Tom 40(54), Fascicola 1, p.31-38, 1995.

[D28] **Drăgulescu D., Toth-Taşcău M.,** *Variaţia elementelor torsorului de reducere şi torsorul din cuplele unui robot RTRTTR*, A XVIII-a Conferinţă de Mecanica Solidelor, Constanţa, Buletin, Vol.2, p.5-10, 1994.

[D29] **Drăgulescu D., Toth-Taşcău M.,** *Torsorul din cuple şi variaţia poziţiei efectorului unui robot de sudare*, Al XI-lea Simpozion Naţional deRoboţi Industriali, Timişoara, 1992.

[D30] **Drăgulescu D., Toth-Taşcău M.,** *Influence of the robot joints vibrations on the position of the end effector*, The Fifth International Conference on Mathematics and Mechanics, Cluj-Napoca, 1996.

[D31] **Drăgulescu D., Toth-Taşcău M.,** *Modelling of robot vibrations by rigid finite element method*, First Romanian-Japonesse Joint Seminar on Applied Electromagnetics on Mechanics, Proceedings, Neptun, 1996.

[D32] **Drăgulescu D., Toth-Taşcău M., Serafin P.,** *Analiza spaţiului de lucru al unui robot RTRTTR ptin metoda modelării statistice*, Al XI-lea Simpozion Naţional de Roboţi Industriali, Timişoara, 1992.

[D33] **Drăgulescu D., Toth-Taşcău M., Neagoe C., Stănciulescu V.,** *Metodă de identificare curentă a planelor autoascunse*, Al XI-lea Simpozion Naţional de Roboţi Industriali, Timişoara, 1992.

[D34] **Drăgulescu D., Toth-Taşcău M.,** *Le torseur dans les articulations d'un robot de soudage soumis à l'action d'une charge aleatoire*, A VII-a Conferinţă de Vibraţii Mecanice, Timişoara, Vol.2, p.7-10, 1993.

[D35] **Drăgulescu D., Toth-Taşcău M.,** *Influenţa vibraţiilor din cuplele unui robot asupra poziţiei efectorului final*, A VIII-a Conferinţă de Vibraţii Mecanice, Timişoara, 1996.

BUPT

[D36] **Drăgulescu D., Toth-Taşcău M.,** *Modelarea vibraţiilor unui manipulator prin metoda elementelor finite*, A VII-a Conferinţă de Vibraţii Mecanice, Timişoara, Vol.2, p.1-6, 1993.

[D37] **Drăgulescu D., Toth-Taşcău M.,** *Contributions to motion planning using potential field method*, The Second International Conference on Technical Informatics, Timişoara, 1996.

[D38] **Drăgulescu D., Toth-Taşcău M.,** *Contributions to cell decomposition method for a two-dimensional workspace*, The Second International Conference on Technical Informatics, Timişoara, 1996.

[D39] **Drăgulescu D., Toth-Taşcău M.,** *Studiul cinematic al unui robot KUKA cu 6 grade de libertate*, Buletinul Conferinţei Naţionale de Dinamica Maşinilor CDM94, Braşov, Vol.1, p.59-64, 1994.

[D40] **Drăgulescu D., Toth-Taşcău M., Buliga C.**, *Studiul dinamicii unui robot tip KUKA*, Comunicările celei de-a VII-a Conferinţe Internaţionale de Inginerie Managerială şi Tehnologică, Timişoara, Vol.7, p.63-70, 1995.

[D41] **Drăgulescu D., Toth-Taşcău M., Moldovan F.,** *Planificarea mişcării roboţilor industriali*, Editura Helicon, Timişoara, 1995.

[D42] **Drăgulescu D., Toth-Taşcău M., Moldovan F.**, *Dinamica robotului KUKA tip 364/10.0*, Acta Universitatis Cibiniensis, Sibiu, Seria Tehnică, Vol.XIX, p.139-144, 1995.

[D43] **Drăgulescu D., Toth-Taşcău M., Moldovan F.,** *Restricţiile cinematice în mişcarea unui robot*, Analele Universităţii din Oradea, Oradea, Fascicola Mecanică, p.109-116, 1995.

[D44] **Drăgulescu D., Toth-Taşcău M., Moldovan F.,** *Studiul mişcării robotului în timp minim în spaţiul operaţional*, Lucrările celei de-a II-a Sesiuni de Comunicări Ştiinţifice a Universităţii "Aurel Vlaicu" Arad, Secţiunea Inginerie Mecanică, Arad, p.109-117, 1994.

[D45] **Drăgulescu D., Toth-Taşcău M., Moldovan F.,** *Metodă şi algoritm de planificare a traiectoriilor plane*, Lucrările celei de-a II-a Sesiuni de Comunicări Ştiinţifice a Universităţii "Aurel Vlaicu" Arad, Secţiunea Inginerie Mecanică, Arad, p.92-96, 1994..

[D46] **Drăgulescu D., Toth-Taşcău M., Moldovan F.,** *Consideraţii asupra metodei de recunoaştere prin arbore de decizie*, Lucrările celei de-a II-a Sesiuni de Comunicări Ştiinţifice a Universităţii "Aurel Vlaicu" Arad, Secţiunea Inginerie Mecanică, Arad, p.102-108, 1994.

[D47] **Drăgulescu D., Toth-Taşcău M., Moldovan F.,** *Precizia dinamică a parcurgerii traiectoriei unui robot*, Acta Universitatis Cibiniensis, Sibiu, Seria tehnică, Vol. XIX, p.133-138, 1995.

[D48] **Drăgulescu D., Moldovan F., Toth-Taşcău M.,** *Soluţionarea problemei cinematice inverse pentru un robot având şase grade de libertate, prin metoda separării matricelor*, Analele Universităţii din Oradea, Oradea, Fascicola Mecanică, p.58-63, 1995.

[D49] **Drăgulescu D., Toth-Taşcău M., Pacso G.,** *Path planning by exact cell decomposition method*, Buletin Ştiinţific al UPT, Tom 41(55), Mecanica, 1996.

[D50] **Drăgulescu D., Toth-Taşcău M.,** *Ecuaţiile cinematice ale unui robot de tip RTRTR*, Al XI-lea Simpozion Naţional de Roboţi Industriali, Timişoara, 1992.

[D51] **Drăgulescu D., Couturier C.,** *Cinematique et dynamique des robots*, Editions IUP-Béthune, 1996.

[D52] **Drăgulescu D., Toth-Taşcău M.,** *Motion planning for a movable robot by potential field method*, Buletin Ştiinţific al UPT, Tom 41(55), Mecanica, 1996.

[D53] **Dumitriu A.,** *Tendinţe şi perspective în robotică,* Simpozionul Naţional de Roboţi Industriali, Ediţia a XII-a, Timişoara, Vol.1, p.159-166, 1994.

[E1] **Elmqvist H., Astrom K.J., Schonthal T.**, *SIMNON User's Guide for MS-DOS Computers*. Department of Automation and Control, Lund Institute Technology, Lund, Sweden, 1986.

[E2] **Epureanu B., Tăbăcaru V.,** *Algoritm pentru minimizarea timpului de mişcare a Roboţilor Manipulatori*, Simpozionul Naţional de Roboţi industriali, Timişoara, Ediţia a XII-a, Vol.2, p.215-222, 1994.

BUPT

[E3] **Erdmann M.,** *On a Represantation of Friction in Configuration Space*, International Journal of Robotics Research, Vol.13 , No.3, p.240-271, 1994.

[E4] **Ernst H.A.,** *MH-1. A Computer-Oriented Mechanical Hand,* Proceedings of Spring Joint Computer Conference, San Francisco, California, p.39-51, 1962.

[F1] **Feng G.,** *A New Adaptive Control Algorithm for Robot Manipulators in Task Space*, IEEE Transactions on Robotics and Automation, Vol.11, No.3, p.457-461, 1995.

[F2] **Finkel R.A.,** *Constructing and debugging manipulator programs*, Ph. D. Thesis, Stanford Artificial Intelligence Lab., Stanford Univ., A.I. Memo 284, 1976.

[F3] **Fleury S., Souères Ph., Laudmond J.-P., Chatila R.,** *Primitives for Smoothing Mobile Robot Trajectories*, IEEE Transactions on Robotics and Automation, Vol.11, No.3, p.441-448, 1995.

[F4] **Fu K.S., Gonzales R.C., Lee C.S.G.,** Control, Sensing, Vision and Inteligence, McGraw-Hill, 1987.

[F5] **Fujimura K.,** *Motion Planning Amid Transient Obstacles*, International Journal of Robotics Research, Vol.13 , No.5, p.395-407, 1994.

[G1] **Ghinea M., Fireteanu V.,** *MATLAB-Calculul numeric şi grafică-Aplicaţii*, Editura Teora, Bucureşti, 1995.

[G2] **Glass K., Colbangh R., Lim D., Seraji H.,** *Real-Time Collision Avoidance for Redundant Manipulators*, IEEE Transactions on Robotics and Automation, Vol.11, No.3, p.448-457, 1995.

[G3] **Grudić G.Z., Lawrence P.D.,** *Iterative Inverse Kinematics with Manipulator Configuration Control*, IEEE Transactions on Robotics and Automation, Vol.9, No.4, 1993.

[G4] **Guldner G., Utkin N.I.,** *Sliding Mode Control for an Obstacle Avoidance Strategy Based on an Electrical Potential Field*, Proceedings of IEEE Conference on Decision and Control, p.424-429, 1993.

[G5] **Guldner G., Utkin N.I.,** *Sliding Mode Control for Gradient Tracking and Robot Navigation using Artificial Potential Fields* , IEEE Transactions on Robotics and Automation, Vol.11, No.2, p.247-254, 1995.

[G6] **Gupta K.C., Singh V.K.,** *A numerical algorithm for solving robot inverse kinematics*, Robotica, Vol.7, p.159-164, 1989.

[H1] **Haar D.T.,** *Elements of Hamiltonian Mechanics*, New-York, p.35-49, 1971.

[H2] **Hać M.,** *Dynamics of flexible mechanisms with mutual dependence between rigid body motion and longitudinal deformation of links*, Mechanism and Machine Theory, Vol.30, No.6, p.837-847, 1995.

[H3] **Handra-Luca V., Mătieş V., Brişan C., Tiuca T.,** *Roboţi: Structură, cinematică şi caracteristici,* Editura Dacia, Cluj- Napoca, 1996.

[H4] **Hashimoto H., Harashima F., Utkin V.I., Krasnova S.A., Kaliko I.M.,** *Sliding Mode Control and Potential Fields in Obstacle Avoidance*, Proceedings of European Control Conference, p.859-862, 1993.

[H5] **Hatvany A.Cs.,** *On conjugate curve schemes for quadratic and cubic curves*, International Conference on Technical Informatics, Timişoara, Proceedings, Vol.2, p.120-129, 1994.

[H6] **Hayward V., Aubry S., Foisy A., Ghallab Y.,** *Efficient Collision Prediction Among Many Moving Objects*, International Journal of Robotics Research, Vol.14 , No.2, p.129-143, 1995.

[H7] **Himmelblau D.M.,** *Applied Nonlinear Programming*, New-York: McGraw-Hill, p.148-157, 1972.

[H8] **Hogan N.,** *Impedance Control: An Aproach to Manipulation*; Parts I-III, ASME Journal of Dynamic Systems Measurement and Control, Vol.107, p.1-24, 1985.

BUPT

[H9] **Hollerbach J.M.,** *A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity*, IEEE Transaction Systems, Man and Cybernetics, Vol. SMC-10, p.730-736, 1980.

[H10] **Hollerbach J.M.,** *Dynamic Scaling of Manipulator Trajectories*, Transactions ASME Journal Dynamic Systems, Measurement and Control, Vol.106, p.102-106, 1984.

[H11] **Hu X., Ahuja N.,** *Motion Estimation under Orthographic Projection*, IEEE Transactions on Robotics and Automation, Vol.7, No.6, p.848-853, 1991.

[H12] **Huang H.P.,** *Mathematical Formulation of Constrained Robot Systems: A Unified Approach*, Mechanism and Machine Theory, Vol.27, No.6, p.687-700, 1992.

[H13] **Hwang Y.K., Ahuja N.,** *A Potential Field Approach to Path Planning*, IEEE Transactions on Robotics and Automation, Vol.8, No.1, p.23-31, 1992.

[I1] **Ider S.K.,** *Open-loop flexibility control in multibody systems dynamics*, Mechanism and Machine Theory, Vol.30, No.6, p.861-869, 1995.

[I2] **Imura J., Sugie T., Yoshikawa T.,** *Adaptive Robust Control of Robot Manipulators-Theory and Experiment*, IEEE Transactions on Robotics and Automation, Vol.10, No.5, p.705-710, 1994.

[I3] **Ispas V.,** *Aplicațiile cinematicii în construcția manipulatoarelor și a roboților industriali*, Editura Academiei Române, Bucureşti, 1990.

[I4] **Ispas V., Pop I., Bocu M.,** *Roboți industriali*, Editura Dacia, Cluj-Napoca, 1985.

[I5] **Ioniță N.,** *Elemente de mecanica automatelor și dinamica automatizării proceselor industriale*, Editura Tehnică, Bucureşti, 1985.

[I6] **Inoue H.**, *Force Feedback in Precise Assembly Tasks*, MIT Artificial Intelligence Laboratory Memo 308, MIT, Cambridge, Mass.,1974.

[J1] **Jacak W.,** *Strategies of searching for collision-free manipulator motions: automata theory approach*, Robotica, Vol.7, p.129-138, 1989.

[J2] **Jain A., Rodriguez G.,** *Diagonalized Lagrangian Robot Dynamics*, IEEE Transactions on Robotics and Automation, Vol.11, No.4, p.571-584, 1995.

[J3] **Jankovski K., Brussel Van H.,** *An Approach to Discrete Inverse Dynamics Control of Flexible-Joint Robots*, IEEE Transactions on Robotics and Automation, Vol.8, No.5, p.651-658, 1992.

[J4] **Jumarie G.,** *Trajectory control of manipulators with time varying inertia links*, Robotica Vol.6, p.197-202, 1088

[J5] **Jun S., Shin K.G.,** *Shortest Path Planning in Discretized Workspaces Using Dominance Relation*, IEEE Transactions on Robotics and Automation, Vol.7, No.3, p.342-350, 1991

[K1] **Kadianakis N.D.,** *The kinematics of continua and the concept of connection on classical space-time*, International Journal of Engineering Science, Vol.34, No.3, p.289-298, 1996.

[K2] **Kahn M.E., Roth B.,** *The near-minimum-time control of open-loop articulated kinematic chains*, ASME Transactions Journal Dynamic Systems, Measurement and Control, Vol.93, p.164-172, 1971.

[K3] **Kazerooni H.,** *Statically balanced direct drive manipulator*, Robotica, Vol.7. p.143-149, 1989.

[K4] **Khalil W., Liégeois A.,** *Generation de Mouvements Optimaux en Robotique*, R.A.I.R.O. Automatique/Systems Analysis and Control, Vol.18, p25-39, 1984.

[K5] **Khatib O.,** *Real-Time Obstacle Avoidance for Manipulators and Mobile Robots*, International Journal of Robotics Research, Vol.5, No.1, p.90-98, 1986.

BUPT

[K6] **Khatib O.,** *Commande Dynamique dans l' Espace Opérationnel des Robots Manipulators en Présence d' Obstacles*, Thesis Dissertation, École Nationale Supéreure de l' Aéronautique et de l' Espace (ENSAE), Toulouse, 1980.

[K7] **Khosla P., Volpe R.,** *Superquadric Artificial Potentials for Obstacle Avoidance and Approach*, Proceedings of IEEE Conference on Robotics and Automation (Philadelphia, PA), p.1778-1784, 1988.

[K8] **Kim B.K., Shin K.G.,** *Suboptimal control of industrial manipulators with weighted minimum-time fuel criterion*, IEEE Transactions on Robotics and Automation, Vol.AC-30, p.1-10, 1985.

[K9] **Kim J.-D., Nam S.-R.,** *An improvement of positioning accuracy by use of piezoelectric voltage in piezoelectric driven micropositioning system simulation*, Mechanism and Machine Theory, Vol.30. No.6, p.1819-827, 1995.

[K10] **Kim J.-O., Khosla P.,** *Real-Time Obstacle Avoidance Using Harmonic Potential Functions*, Proceedings of IEEE Conference on Robotics and Automation, p.790-796, 1991.

[K11] **Kim H.Y., Streit D.A.,** *Configuration dependent stiffnes of the PUMA 560 manipulator: Analytical and experimental results*, Mechanism and Machine Theory, Vol.30. No.8, p.1269-1277, 1995.

[K12] **Kirćanski M., Vukobratović M.,** *Contribution to Control of Redundant Robotic Manipulators in an Environment with Obstacles*, International Journal of Robotics Research, Vol.5, No.4, p.112-119, 1986.

[K13] **Kirk D.E.,** *Optimal Control Theory: An Introduction*, Englewood Cliffs, New York, Prentice-Hall, p.227-238, 1971.

[K14] **Koditschek D.E.,** *Exact Robot Navigation by Means of Potential Functions: Some Topological Considerations*, Proceedings of IEEE Conference on Robotics and Automation, p.1-6, 1987.

[K15] **Koditschek D.E.,** *Robot Planning and Control Functions*, The Robotics Review 1, MIT Press, 1989.

[K16] **Kokkinis T., Sahraian M.,** *Inverse dynamics of a flexible robot arm by optimal control*, Flexible mechanism, dynamics, and robot trajectories, DE-Vo.24, edited by S.Derby, M. McCarthy, A. Pisano, p.497-502, New York, 1990.

[K17] **Kondo K.,** *Motion Planning with Six Degrees of Freedom by Multistrategic Bidirectional Heuristic Free-Space Enumeration*, IEEE Transactions on Robotics and Automation, Vol.7, No.3, p.267-277, 1991.

[K18] **Konjović Z., Vukobratović M., Surla D.,** *Synthesis of the Optimal Trajectories for Robot Manipulators Based on their Complete Dynamic Models*, International Journal of Robotics and Automation, Vol.9, No.1, p.1-28, 1994.

[K19] **Koren Y.,** *La robotique pour ingénieurs*, McGraw-Hill, Paris, 1986.

[K20] **Kovacs Fr., Cojocaru G.,** *Manipulatoare, roboți și aplicațiile lor industriale*, Editura Facla, Timişoara, 1982.

[K21] **Kovacs Fr., Rădulescu C.,** *Roboți industriali*, Lito UTT, Timişoara, 1992.

[K22] **Krogh B.H.,** *A Generalised Potential Field Approach to Obstacle Avoidance Control*, Proceedings of SME Conference on Robotics Research, 1984.

[K23] **Kwatny H.G., Blankenship G.,** *Symbolic Construction of Models for Multibody Dynamics*, IEEE Transactions on Robotics and Automation, Vol.11, No.2, p.271-281, 1995.

[L1] **Lakshmikantham V., Leela S.,** *Differential and Integral Inequalities*, New York: Academic, p.41-43, 1969.

[L2] **Lallemand J.-P., Zeghloul S.,** *Robotique. Aspects Fondamentaux. Modélisation Mécanique. CAO en Robotique. Commande*, Masson, Paris, 1994.

BUPT

[L3] **Lamadrid J.G.,** *Avoidance of Obstacles with Unknown Trajectories: Locally Optimal Path and Periodic Sensor Readings*, International Journal of Robotics Research, Vol.13, No.6, p.496-507, 1994.

[L4] **Larionescu D.,** *Metode numerice*, Editura Tehnică, Bucureşti, 1989.

[L5] **Latombe J.C.,** *Robot Motion Planning*. Kluwer Academic Publishers, Norwell-Massachusetts, USA, 1991.

[L6] **Lee C.S.G.,** *Robot Arm Kinematics and Dynamics*, Advances in Automation and Robotics: Theory and Aplication, JAI Press, Conn., p.21-63, 1985.

[L7] **Lee C.S.G., Chang P.R.,** *Efficient Parallel Algorithm for Robot Inverse Dynamics Computation*, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-16, No.4, p.532-542, 1986.

[L8] **Lee C.S.G., Chung M.J.,** *An Adaptive Control Strategy for Mathematical Manipulators*, IEEE Transactions on Automation and Control, Vol.AC-29, No.9, p.837-840, 1984.

[L9] **Lee C.S.G., Lee B.H.,** *Resolved Motion Adaptive Control for Mechanical Manipulators*, ASME Transactions on Journal of Dynamic Systems, Measurement and Control, Vol.106, No.2, p.134-142, 1984.

[L10] **Lenarcic J., Onlak P., Stanic U.,** *Approximate Calculation of Joint Trajectories for Control along Cartesian Path*, The 15-th International Symposium on Industrial Robots, 1985.

[L11] **Lewis R.A., Bejczy A.K.**, *Planning Considerations for a Moving Robot with Arm*, Proceedings of 3rd International Joint Conference on Artificial Intelligence, Stanford University, California, 1973.

[L12] **Lewis R.A.,** *Autonomous Manipulation on a Robot: Summary of Manipulator Software Functions*, Technical Memo 33-679, Jet Propulsion Laboratory, Pasadena, California, 1974.

[L13] **Li C.J., Sankar T.S.,** *Development of efficient closed-form dynamic equations for robot manipulators using parallel and perpendicular concepts*, Mechanism and Machine Theory, Vol.28, No.2, p.233-248, 1993.

[L14] **Li C.J., Sankar T.S.,** *Fast Inverse Dynamics Computation in Real-Time Robot Control*, Mechanism and Machine Theory, Vol.27, No.6, p.741-750, 1992.

[L15] **Liegeois A.,** *Les Robots. Analyse des Performances et C.A.O.*, Editions Hermès, Paris, 1984.

[L16] **Lieh J.,** *Computer Oriented Closed-Form Algorithmic for Constrained Multibody Dynamics for Robotics Application*, Mechanism and Machine Theory, Vol.29, No.3, p.357-371, 1994.

[L17] **Lin C.S., Chang P.R.,** *Joint Trajectories of Mechanical Manipulators for Cartesian Path Approximation*, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-13, No.16, 1983.

[L18] **Lin C.S., Chang P.R., Luh J.Y.S.,** *Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots*, IEEE Transactions on Automation and Control, Vol.AC-28, No.12, p.1066-1073, 1983.

[L19] **Lin J., Lewis F.L.,** *A Symbolic Formulation of Dynamic Equations for a Manipulator with Rigid and Flexible Links*, International Journal of Robotics Research, Vol.13, No.5, p.454-466, 1994.

[L20] **Lin S.-K.,** *Minimal Linear Combinations of the Inertia Parameters of a Manipulator*, IEEE Transactions on Robotics and Automation, Vol.11, No.3, p.360-373, 1995.

[L21] **Liu J.,** *Spatial Reasoning About Robot Compliant Mouvements and Optimal Path in Qualitatively Modeled Environments*, International Journal of Robotics Research, Vol.15, No.2, p.181-210, 1996.

[L22] **Lorenc S.J., Stanisić M.M.,** *Application of instantaneus invariants to the path tracking control problem of planar two degree-of-freedom systems: a singularity-free mapping of trajectory geometry*, Mechanism and Machine Teory, Vol.30. No.6, p.883-896, 1995.

BUPT

[L23] **Luh J.Y.S., Walker M.W.,** *Minimum-time control along the path for a mechanical arm*, Proceedings CDC, TP-5, p.755-759, 1977.

[L24] **Luh J.Y.S., Walker M.W., Paul R.P.C.,** *On-line computational scheme for mechanical manipulators*, ASME Transactions on Journal of Dynamic Systems, Measurement and Control, Vol.102, p.69-76, 1980.

[L25] **Luh J.Y.S.,Walker M.W., Paul R.P.C.,** *Resolved-acceleration control of mechanical manipulators*, IEEE Transactions on Automation and Control, Vol.AC-25, No.3, p.468-474, 1980.

[L26] **Luh J.Y.S., Lin C.S.,** *Optimum Path Planning for Mechanical Manipulators*, ASME Transactions on Journal of Dynamic Systems, Measurement and Control, Vol.103, p.330-335, 1981.

[L27] **Luh J.Y.S., Lin C.S.,** *Scheduling of Parallel Computation for a Computer Controlled Mechanical Manipulator*, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-12, p.214-234, 1982.

[L28] **Luh J.Y.S.,** *An Anatomy of Industrial Robots and Their Controls*, IEEE Transaction on Automatic and Control, Vol. AC-28, No.2, p.133-153, 1983.

[L29] **Luh J.Y.S., Lin C.S.,** *Approximate Joint Trajectories for Control of Industrial Robots along Cartesian Paths*, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-14, No.3, p.444-450, 1984.

[L30] **Lumelsky V.,** *A Comparative Study on the Path Length Performance of Maze-Searching and Robot Mmotion Planning Algorithms*, IEEE Transactions on Robotics and Automation, Vol.7, No.1, p.57-66, 1991.

[L31] **Lungu N.,** *Matematici cu aplicații tehnice*, Editura Tehnică, Bucureşti, 1990.

[M1] **Manocha D., Canny J.F.,** *Efficient Inverse Kinematics for General 6R Manipulators*, IEEE Transaction on Robotics and Automation, Vol.10, No.5, p.648-657, 1994.

[M2] **Marcé L., Jullière M., Place H.,** *Stratégie de Contournement d'Obstacles pour un Robot Mobile*, R.A.I.R.O. Automatique/Systems Analysis and Control, Vol.15, No.1, p.5-18, 1981.

[M3] **Massonet Ch., Deprez G., Maquoi R., Muller R., Fonder G.,** *Calculul structurilor la calculatoare electronice*, Editura Tehnică, Bucureşti, 1974.

[M4] **Mayeda H., Yoshida K., Osuka K.,** *Base Parameters of Manipulator Dynamic Models*, IEEE Transaction on Robotics and Automation, Vol.6, No.3, p.312-320, 1990.

[M5] **Măties V., Manui A.,** *Asupra unei metode de calcul în analiza cinematică a roboților*, Simpozionul Național de Roboți Industriali, Timişoara, Ediția a XII-a, Vol.2, p.235-240, 1994.

[M6] **McCarthy J.,** *A Computer with Hands, Eyes and Ears*, Fall Joint Computer Conference, AFIPS Proceedings, p.329-338, 1968.

[M7] **Meghdari A., Shahinpoor M.,** *Three-dimensional flexural-joint stiffness analysis of flexible manipulator arms*, Robotica, Vol.6, p.203-212, 1988.

[M8] **Meghdari A., Shahinpoor M.,** *Elastic Deformation Characteristics of a PUMA 560 Robot Manipulator*, International Journal of Robotics and Automation, Vol.2, No.1, 1987

[M9] **Merlet J.P.,** *Trajectory Verification in the Workspace for Parallel Manipulators*, International Journal of Robotics Research, Vol.13, No.4, p.326-333, 1994.

[M10] **Merlet J.P.,** *Les robots parallèles*, Edition Hermes,

[M11] **Micula Gh.,** *Funcții spline și aplicații*, Editura Tehnică, 1978.

[M12] **Mitsi S., Bouzakis K.-D., Mansour G.,** *Optimisation of robot links motion in inverse kinematics solution considering collision avoidance and joint limits*, Mechanism and Machine Theory, Vol.30, No.5, p.653-663, 1995.

[M13] **Moler C., Little J., Bangert S.,** *PRO-MATLAB User's Guide*, Math Works, Sherborn MA, 1987.

BUPT

[N1] **Nagarajan S., Turcic D.A.**, *Dynamic stability constructions in elastic closed loop linkage systems*, Flexible mechanism, dynamics and robot trajectories, DE-Vol.24, edited by S.Derby, M.McCarthy, A.Pisano, New-York, p.1-8, 1990.

[N2] **Neag I.A.**, *Simularea mişcării structurii mecanice a roboţilor industriali*, Simpozion Naţional de Roboţi Industriali, Ediţia aXII-a, Timişoara, Vol.2, p.117-128, 1994.

[N3] **Nearchou A.C., Aspragathos N.A.**, *Application of genetic algorithms to point to point motion of redundant manipulators*, Mechanism and Machine Theory, Vol.31, No.3, p.261-270, 1996.

[N4] **Negrean I., Vuscan I., Itul T.**, *Model de analiză a erorilor statice la roboţi industriali*, AVII-a Conferinţă Internaţională de Ingineria Managerială şi Tehnologică TEHNO'95, Timişoara, Vol.7, p.33-40, 1995.

[N5] **Negrean I., Popescu P., Haiduc N., Vuscan I.**, *Influenţa erorilor geometrice ale parametrilor de tip D-H asupra preciziei cinematice a roboţilor industriali*, Simpozion Naţional de Roboţi Industriali, Ediţia aXII-a, Timişoara, Vol.2, p.161-170, 1994.

[N6] **Nevins J.L., Whitney D.E.**, *Exploratory Research in Industrial Modular Assembly*, NSF Project Reports 1 to 4, C.S. Draper Laboratory, Cambridge, Mass., 1974-1976.

[N7] **Newman W.S., Hogan N.**, *High Speed Control and Obstacle Avoidance Using Dynamic Potential Functions*, Proceedings IEEE International Conference on Robotics and Automation, (Raleigh, NC), p.14-24, 1987.

[N8] **Nikolic I.**, *Determination of elastodynamic errors in joints of industrial robots*, Robotica, Vol.6, 1988.

[N9] **Nitulescu M.**, *Planificarea traiectoriei roboţilor mobili prin decompoziţia celulară aproximativă*, Simpozion Naţional de Roboţi Industriali, Ediţia a XII-a, Timişoara, Vol.2, p.53-60, 1994.

[O1]. **Orin D.E., McGhee R.B., Vukobratović M., Hartoch G.**, *Kinematic and Kinetic Analysis of Open-Chain Linkages Utilizing Newton-Euler Methods*, Math. Biosci., vol.143, p.147-130, 1979.

[O2] **Ozaki H., Shimadzu T., Mohri A.**, *Collision-free path generation for a mobile robot by an artificial transformation of obstacle space*, Robotica, Vol.7, p.139-142, 1989.

[P1] **Papageorgoiu M., Bauschert Th.**, *Stochastic Optimal Control of Moving Vehicles in a Dynamic Environment*, International Journal of Robotics Research, Vol.13, No.4, p.343-354, 1994.

[P2] **Parent M., Laurgeau C.**, *Les Robots. Langages et méthodes de programmation*, Hermes Publishing, Paris, 1983.

[P3] **Paul R.P.C.**, *Modeling Trajectory Calculation and Servoing of a Computer Controlled Arm*, Ph.D. Thesis, Stanford Artificial Intelligence Laboratory, Stanford University, California, AI Memo 177, 1972.

[P4] **Paul R.P.C.**, *Manipulator Cartesian Path Control*, IEEE Transactions on Systems, Man, Cybernetics, Vol.SMC-9,No.11, p.702-711, 1979.

[P5] **Paul R.P.C.**, *Robot Manipulators: Mathematics, Programming, and Control*, Cambridge, The MIT Press, 1981, 1983.

[P6] **Paul R.P.C., Shimano B., Mayer G.E.**, *Kinematic control equations for simple manipulators*, IEEE Transactions on Systems, Man and Cybernetics, Vol.SMC-11, p.449-455, 1981.

[P7] **Paul R.P.C.**, *Advanced Industrial Robot Control Systems*, Purdue University, TREE 78-25, 1978.

[P8] **Payannet D.**, *Modélisation et correction des erreurs statiques des robots manipulateurs*, Thèse de Docteur-Ingénieur, Montpellier, 1985.

BUPT

[P9] **Pfeiffer F., Johanni R.,** *A concept for manipulator trajectory planning,* Proceedings IEEE Conference on Robotics and Automation (San Francisco), 1986.

[P10] **Pieper D.L.,** *The kinematics of manipulators under computer control,* Ph. D. Thesis, Stanford University, Stanford, 1968.

[P11] **Precup T, Handra-Luca V., Precup D., Şipoş C.,** *Robot simulation using object-oriented programming,* International Conference on Technical Informatics, Timişoara, Vol.2, p.42-51, 1994.

[P12] **Pruski A.,** *Robotique Générale,* Editions Ellipses, Paris, 1989.

[R1] **Raibert M.H., Horn B.K.P.,** *Manipulator Control Using the Configuration Space Method.* Industrial Robots, Vol.5, p.69-73, 1978.

[R2] **Raicu A.,** *Consideraţii privind precizia geometrică la un robot RRR,* Simpozion Naţional de Roboţi Industriali, Ediţia aXII-a, Timişoara, Vol.2, p.81-84, 1994.

[R3] **Rastegar J., Fardanesh B.,** *Trajectory pattern specific inverse dynamic formulation of robot manipulators and its applications,* Flexible mechanism, Dynamics, and robot trajectories, DE-Vol.24, edited by S.Derby, M.McCarthy, A.Pisano, New-York, p.103-110, 1990.

[R4] **Rădulescu C.,** *Oscilaţiile punctului caracteristic la comanda punct cu punct a roboţilor industriali,* Analele Universităţii din Oradea, Fascicola Mecanică, Oradea, p.187-192, 1994.

[R5] **Rădulescu C.,** *Comportamentul în regim dinamic al punctului caracteristic la comanda PTP a roboţilor industriali,* Simpozion Naţional de Roboţi Industriali, Ediţia aXII-a, Timişoara, Vol.3, p.83-88, 1994.

[R6] **Reister D.B., Pin Fr.G.,** *Time-Optimal Trajectories for Mobile Robots with two Independently Driven Wheels,* International Journal of Robotics Reserrch, Vol.13, No.1, p.38-54, 1994.

[R7] **Reister D.B., Lenhart S.M.,** *Time-Optimal Paths for High-Speed Maneuvering,* International Journal of Robotics Research, Vol.14, No.2, p.184-194, 1995.

[R8] **Renaud M.,** *Contribution à la modélisation et à la commande dynamique des robots manipulateurs,* Thèse de Docteur d'Etat, Unversité Paul Sabatier, Toulouse, 1980.

[R9] **Richard M.J., Dufour F., Tarasiewicz S.,** *Commande des Robots Manipulateurs par la Programmation Dynamique,* Mechanism and Machine Theory, Vol.28, No.3, p.301-316, 1993.

[R10] **Rimon E., Koditschek D.E.,** *Exact Robot Navigation Using Cost Functions: The Case of Distinct Spherical Boundaries in E^n,* Proceedings of the IEEE International Conference on Robotics and Automation, Philadelphia, p.1791-1796, 1988.

[R11] **Rimon E.,** *Exact Robot Navigation Using Artificial Potential Functions,* Thesis dissertation, Yale University, New-Haven, C.T., 1990.

[R12] **Rimon E., Koditschek D.E.,** *Exact Robot Navigation Using Artificial Potential Functions,* IEEE Transactions on Robotics and Automation, Vol.8, No.5, p.501-518, 1992.

[R13] **Ripianu A., Popescu C., Bălan B.,** *Mecanică tehnică,* Editura Didactică şi Pedagogică, Bucureşti, 1979.

[R14] **Roberts R.G., Maciejewski A.A.,** *Singularities, Stable Surfaces and the Repeatable Behavior of Kinematically Redundant Manipulators,* International Journal of Robotics Research, Vol.13, No.1, p.70-81, 1994.

[R15] **Rodriguez G., Kreutz-Delgado K.,** *Spatial Operator Factorization and Inversion of the Manipulator Mass Matrix,* IEEE Transactions on Robotics and Automation, Vol.8, No.1, p.65-75, 1992.

[S1] **Sanjeev S., Wien J.T.,** *A Global Approach to Path Planning for Redundant Manipulators,* IEEE Transactions on Robotics and Automation, Vol.11, No.1, p.152-159, 1995.

BUPT

[S2] **Santis R.M.,** *Path-Tracking for a Tractor-Trailer-like Robot,* International Journal of Robotics Research, Vol.13, No.6, p.533-543, 1994.

[S3] **Schalkoff R.J.,** *Artificial Inteligence. An Engineering Approach,* Mc.Graw-Hill Publishing Company, New-York, 1990.

[S4] **Schoenberg I.J.,** *Privelişti matematice,* Centrul de Cercetări Matematice, Universitatea Wisconsin-Madison, Traducere la Editura Tehnică, Bucureşti, 1989.

[S5] **Schweikard A.,** *Polynomial Time Collision Detection for Manipulator Path Specified by Joint Motions,* IEEE Transactions on Robotics and Automation, Vol.7, No.6, p.865-870, 1991.

[S6] **Sciavicčo L., Siciliano B.,** *Coordinate transformation: A Solution Algorithm for One Class of Robots,* IEEE Transactions on Systems, Man and Cybernetics, Vol.SMC-16, No.4, p.550-559, 1986.

[S7] **Shahinpoor M., Meghdari A.,** *Combined flexural-joint stiffness matrix and the elastic deformation of a servo-controlled two-link robot manipulator,* Robotica, Vol.4, 1986

[S8] **Shan Y., Koren Y.,** *Obstacle Accommodation Motion Planning,* Transactions on Robotics and Automation, Vol.11, No.1, p.36-49, 1995.

[S9] **Shankar S., Saraf A.,** *Singularities in mechanisms. The local trajectory tracking problem,* Mechanism and Machine Theory, Vol.30, No.8, p.1139-1148, 1995.

[S10] **Sharma R.,** *Locally Efficient Path Planning in an Uncertain Dynamic Environment Using a Probabilistic Model,* Transactions on Robotics and Automation, Vol.8, No.1, p.105-110, 1992.

[S11] **Shieh M.-D., Duffy J.,** *Autonomous rectilinear motion planning for a spatial robot: Path planning for a spatial 4R manipulator with a single spherical obstacle inside the workspace,* Flexible Mechanism, Dynamics, and Robot Trajectories, DE-Vol.24, edited by S.Derby, M.McCarthy, A.Pisano, New-York, p.297-302, 1990.

[S12] **Shiller Z., Gwo Y.-R.,** *Dynamic Motion Planing of Autonomous Vehicles,* IEEE Transactions on Robotics and Automotion, Vol.7, No.2, p. 241-249, 1991.

[S13] **Shiller Z., Dubowsky S.,** *On Computing the Global Time-Optimal Motions of Robotic Manipulators in the Presence of Obstacles,* IEEE Transactions on Robotics and Automation, Vo.7, No.6, p.785-797, 1991.

[S14] **Shiller Z.,** *On Singular Time-Optimal Control Along Specified Paths,* IEEE Transactions on Robotics and Automation, Vol.10, No.4, p.561-566, 1994.

[S15] **Shimon Y. Nof,** *Handbook of Industrial Robotics,* Krieger Publishing Company, Malabar, Florida, 1992.

[S16] **Shin K.G., McKay N.D.,** *Open-loop minimum-time control of mechanical manipulators and its application,* Proceedings of American Control Conference, San Diego, California, p.1231-1236, 1984.

[S17] **Shin K.G., McKay N.D.,** *Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints,* IEEE Transactions on Automatic Control, Vol. AC-30, No.6, p.531-541, 1985.

[S18] **Shin K.G., McKay N.D.,** *Robust Trajectory Planning for Robotic Manipulators under Payload Uncertainties,* IEEE Transactions on Automatic Control, Vol. AC-32, No.12, p.1044-1054, 1987.

[S19] **Silaş Gh., Groşanu I.,** *Mecanică,* Editura Didactică şi Pedagogică, Bucureşti, 1981.

[S20] **Singh J.R., Rastegar J.,** *Optimal Synthesis of Robot Manipulators Based on Global Kinematic parameters,* Mechanism and Machine Theory, Vol.30, No.4, p.569-580, 1995.

[S21] **Slotine J.E., Spong, M.W.,** *Robust robot control with bounded inputs,* Journal of Robotics Systems, Vol.2, No.4, 1985.

[S22] **Solc F.,** *Robot modelling and control,* Internationaler Sommerkurs, Technische Universität Wien, 1993.

BUPT

[S23] **Steinle J.,** *Controller design with observer for the control of elastic joint robots,* Technical University of Munich, 1995.

[S24] **Surdilović D., Vukobratović M.,** *One method for efficient dynamic modeling of flexible manipulators,* Mechanism and Machine Theory, Vol.31, No.3,p.297-315, 1996.

[S25] **Surdilović D., Vukobratović M.,** *Deflection compensation for large flexible manipulators,* Mechanism and Machine Theory, Vol.31, No.3, p.317-329, 1996.

[S26] **Swevers J., Torfs D., Adams M., De Schutter J., Van Brussel H.,** *Flexible Joint Control of a KUKA IR 161/60 Industrial Robot,* Proceedings of the 17-th International Seminar on Modal Analysis, KULeuven, Belgium, p.285-299, 1993.

[S27] **Szkodny T.,** *Dynamics of Industrial Robot Manipulators,* Mechanisms and Machine Theory, Vol.30, No.7, p.1057-1072, 1995.

[S28] **Szkodny T.,** *Forward and inverse kinematics of IRb-6 manipulator,* Mechanism and Machine Theory, Vol.30, No.7, p.1039-1056, 1995.

[S29] **Szwedowicz D., Szwedowicz A.,** *A method of calculation of robot dynamics with consider of disturbance effects of motion,* Proceedings of the 17-th International Seminar on Modal Analysis, KULeuven, Belgium, p.301-314, 1993.

[T1] **Tabarah E., Benhabib B., Fenton R.G., Cohen R.,** *Cycle-time optimization for single-arm and two-arm robots performing continuous path operations,* Flexible mechanism, dynamics, and robot Trajectories, DE-Vol.24, edited by S.Derby, M.McCarthy, A.Pisano, New-York, p.401-406, 1990.

[T2] **Takegaki M., Arimoto S.,** *A new feedback method for dynamic control of manipulator,* ASME Transactions. Journal of Dynamic Systems, Measurement and Control, Vol.103, p.119-125, 1981.

[T3] **Taylor R.H.,** *Planning and execution of straight line manipulator trajectories,* IBM Journal of Research and Development, Vol.23, No.4, p.424-436, 1979.

[T4] **Theodore R.J., Ghosal A.,** *Comparison of the Assumed Modes and Finite Element Models for Flexible Multilink Manipulators,* International Journal of Robotics Research, Vol.14, No.2, p.91-111, 1995.

[T5] **Ting K.-L, Lee N.L., Braudan G.H.,** *Synthesis of Polynomial and Other Curves with the Bezier Technique,* Mechanism and Machine Theory, Vol.29, No.6, p.887-903, 1994.

[T6] **Tomovic R., Boni G.,** *An Adaptive Artificial Hand,* IEEE Transactions on Automatic Control, Vol.AC-7, No.3, p.3-10, 1962.

[T7] **Tondu B.,** *Génération de mouvements point-à-point optimaux en robotique. Application à l'évaluation des performances et à la C.A.O. des robots,* Thèse de Docteur-Ingénieur, USTL, Montpellier, 1984.

[T8] **Tondu B., El-Zorkany H.I.,** *Experimental identification and validation of a model for a robot's trajectory generator,* IFAC Congress "Théorie de Robots", Vienne, p.345-349, 1986.

[T9] **Toth-Taşcău M., Drăgulescu D., Moldovan F.,** *Cinematica şi dinamica roboţilor industriali. Îndrumător de laborator,* Lito UPT, Timişoara, 1995.

[T10]. **Toth-Taşcău M., Drăgulescu D.,** *Generarea traiectoriei unui robot pe baza modelului dinamic,* Buletinul Conferinţei Naţionale de Dinamica Maşinilor CDM 94, Braşov, Vol.2, p.63-70, 1994.

[T11]. **Toth-Taşcău M., Drăgulescu D.,** *Path Planning for a Car-Like Robot by Potential Field Method,* The 29-th International Symposium on Automotive Technology and Automation, Firenze, Proceedings, Vol.1, Mechatronics, p.333-338, 1996.

[T12]. **Toth-Taşcău M., Drăgulescu D.,** *Minimurile locale în planificarea mişcării roboţilor prin metoda câmpului portenţial,* Al XIII-lea Simpozion Naţional de Robotică, Reşiţa, 1996.

BUPT

[T13]. **Toth-Taşcău M., Drăgulescu D.**, *Generarea mişcării în coordonate operaţionale carteziene*, Al XII-lea Simpozion Naţional de Roboţi Industriali, Timişoara, Vol.3, p.31-38, 1994.

[T14]. **Toth-Taşcău M., Drăgulescu D.**, *Coordonarea cuplelor cinematice ale unui robot prin metoda timpului minim*, Al XIII-lea Simpozion Naţional de Robotică, Reşiţa, 1996.

[T15]. **Toth-Taşcău M., Drăgulescu D.**, *Precizia cinematică a parcurgerii traiectoriei unui robot*, Acta Universitatis Cibiniensis, Sibiu, Seria Tehnică, Vo.XIX, p.127-132, 1995.

[T16]. **Toth-Taşcău M.**, *Studiul erorilor dinamice datorate deformaţiilor elementelor unui robot*, A VIII-a Conferinţă de Vibraţii Mecanice, Timişoara, 1996.

[T17] **Tournassoud P.**, *Géometrie et Intelligence Artificielle pour les Robots*, Editions Hermès, Paris, 1982.

[T18] **Tournassoud P.**, *Planification et contrôle en robotique. Application aux robots mobiles et manipulateurs*, Editions Hermès, Pariss, 1992.

[T19] **Troch I.**, *Robot Control*, Internationaler Sommerkurs, Technical University Vienna, 1993.

[T20] **Tu Q., Rastegar J., Singh J.R.**, *Trajectory synthesis and inverse dynamics model formulation and control of tip motion of a high performance flexible positioning system*, Mechanism and Machine Theory, Vol.29, No.7, p.959-958, 1994.

[U1] **Uhl T., Szymkat M.**, *Comparison of the classical and neural-besed approach to control of manipulation robots*, Proceedings of the 17-th International Seminar on Modal Analysis-KU Leuven, Belgium, p.315-330, 1993.

[U2] **Uicker J.J.**, *On the dynamic analysis of spatial linkages using 4x4 matrices*, Ph. D. Dissertation, Northwestern University, Evanston, Il., 1965.

[U3] **Urbaniec K.**, *Optimal Design of Process Equipment*, Ellis Horwood Limited Publishers, Chichester, 1986.

[V1] **Văcărescu I., Văcărescu V., Lovasz E.-C.**, *Optimizarea traiectoriei şi a legilor de mişcare ale punctului caracteristic*, Simpozion Naţional de Roboţi Industriali, Ediţia aXII-a, Timişoara, Vol.3, p.231-242, 1994.

[V2] **Văcărescu V., Lovasz E.-C.**, *Minimizarea timpului de deplasare a punctului caracteristic pe traiectorie*, Analele Univeritătii din Oradea, Fascicola Mecanică, Vol.II, Oradea, p.121-124, 1995.

[V3] **Văcărescu V., Lovasz E.-C.**, *Traiectorii ale punctului caracteristic de acceleraţie minimă*, Analele Univeritătii din Oradea, Fascicola Mecanică, Vol.II, Oradea, p.135-138, 1995.

[V4] **Vertut J., Coiffet Ph.**, *Les Robots. Téléoperation. Vers la téléoperation assistée par l'ordinateur*, Editions Hermès, Paris, 1985.

[V5] **Vibet C.**, *Robots. Principes et contrôle*, Editions Marketing, Paris, 1987.

[V6] **Vlada M., Posea A., Nistor I., Constantinescu C.**, *Grafică pe calculator în limbajele Pascal şi C. Implementare şi aplicaţii*, Editura Tehnică, Bucureşti, 1992.

[V7] **Voinea R., Voiculescu D., Ceauşu V.**, *Mecanică*, Editura Didactică şi Pedagogică, Bucureşti, 1983.

[V8] **Vukobratović M., Stokić D.**, *Control of Manipulation Robots*, Berlin, Heidelberg, New York, Springer-Verlag, 1982.

[V9] **Vukobratović M., Kirćanski M.**, *One Method for Simplified Manipulator Model Construction and its Application in Quazioptimal Trajectory Synthesis*, Mechanism and Machine Theory, Vol.17, No.6,p.369-378, 1982.

[V10] **Vukobratović M., Kirćanski M.**, *A Method for Optimal Synthesis of Manipulation Robot Trajectories*, Journal of Transactions of the ASME. Journal of Dynamic System, Measurement and Control, Vol.104, p.158-193, 1982.

BUPT

[V11] **Vukobratović M., Kirćanski M.**, *A Dynamic Approach to Nominal Trajectory Synthesis for Redundant Manipulators*, IEEE Transactions on Systems, Man, and Cybernetics, Vol.SMC-14, No.4, p.580-586, 1984.

[V12] **Vukobratović M., Kirćanski M.**, *Scientific Fundamentals of Robotics 3: Kinematics and Trajectory Synthesis of Manipulation Robots*, Springer Verlag, 1986.

[V13] **Vukobratović M., Stojić R.**, *Historical Perspective of Hibrid Control in Robotics: Beginnings, Evolution, Criticism, and Trends*, Mechanism and Machine Theory, Vol.30, No.4, p.519-532, 1995.

[W1] **Wang D., Vidyasagar M.**, *Modeling a Class of Multilink Manipulator with the Last Link Flexible*, IEEE Transactions on Robotics and Automation, Vol.8, No.1, 1992.

[W2] **Wang F.-Y.**, *On the External Fundamental Frequencies of One-Link Flexible Manipulators*, International Journal of Robotics Reasearch, Vol.13, No.2, p.162-170, 1994.

[W3] **Wang L.**, *A decomposition theorem of motion*, International Journal of Engineering Science, Vol.34, No.4, p.417-423, 1996.

[W4] **Wang L.C.T., Chen C.C.**, *A Combined Optimization Method for Solving the Inverse Kinematic Problem of Mechanical Manipulators*, IEEE Transactions on Robotics and Automation, Vol.7, No.4, p.489-499, 1991.

[W5] **Warren C.W.**, *Global Path Planning Using Artificial Potential Fields*, Proceedings IEEE International Conference on Robotics and Automation, (Scottsdale, AZ), p.316-321, 1989.

[W6] **Waters R.C.**, *Mechanical arm control*, MIT Artificial Intelligence Laboratory, Memo 549, 1979.

[W7] **Wiens G.J., Jang W.-M.**, *Passive Joint Control of Dynamic Coupling in Mobile Robots*, International Journal of Robotics Research, Vol.13, No.3, p.209-220, 1994.

[W8] **Will P., Grossman D.**, *An Experimental System for Computer Controlled Mechanical Assembly*, IEEE Transactions on Computers, Vol.C-24, No.9, p.879-888, 1975.

[W9] **Williams D.W., Turcic D.A.**, *An Inverse Kinematic Analysis Procedure for Flexible Open-Loop Mechanisms*, Mechanism and Machine Theory, Vol.2, No.6, p.701-714, 1992.

[W10] **Withney D.E.**, *Resolved motion rate control of manipulators and human prostheses*, IEEE Transactions on Man-Machine Systems, Vol.10, 1983.

[W11] **Wittbrodt E., Wojciech S.**, *Modelling of Vibrations of Spatial Changing Configuration Systems by the Rigid Finite Element Method*, Sixth International Conference on the Theory of Machines and Mechanisms, Lieberec, p.319-324, 1992.

[W12] **Wojciech S.**, *Dynamics of Manipulators Considering Flexible Links and Dry Friction in Joints*, Sixth International Conference on thr Theory of Machines and Mechanisms, Lieberec, p.255-266, 1992

[X1] **Xi F.**, *Trajectory tracking of a spatial flexible link manipulator using an inverse dynamics method*, Mechanism and Machine Theory, Vol.30, No.8, p.1113-1126, 1995.

[X2] **Xi F., Fenton R.G.**, *Coupling Effect of a Flexible Link and a Flexible Joint*, International Journal of Robotics Research, Vol.13, No.5, p.443-453, 1994.

[X3] **Xingrui M., Peilin S., Xinggui W., Zhenzhu Z.**, *Dynamics of Constrained Mechanical Systems and its Application in Robotics-Impplementation*, Mechanism and Machine Theory, Vol.29, No.6, p.837-845, 1994.

[X4] **Xu L.-J., Fan S.-W., Chen Y.**, *A numeric-symbolic approach for kinematics and dynamics of robotic manipulators*, Mechanism and Machine Theory, Vol.30, No.5, p.645-652, 1995.

[Y1] **Yang Z., Sadler J.P.**, *Finite element modeling of spatial robot manipulators*, Flexible mechanism, dynamics, and robot trajectories, DE-Vol.24, edited by S. Derby, M. McCarty, A Pisano, New-York, p.489-496, 1990.

BUPT

[Z1] **Zheng Y.-F., Hemami H.,** *Computation of Multibody System Dynamics by a Multiprocessor Scheme,* IEEE Transactions on Systems, Man, and Cybernetics, Vol.SMC-16, No.1, p.102-110, 1986.

[Z2] **Zhu Q.,** *Hidden Markor Model for Dynamic Obstacle Avoidance of Mobile Robot Navigation,* IEEE Transactions on Robotics and Automation, Vol.7, No.3 p.390-403, 1991.

.

[*] *** SCORBOT-ER III, User's Manual.

[*] ***Contract Nr.4004/1995, Tema 15B, CNCSU, *Planificarea şi generarea mişcării la roboţi. Analiza comparativă a diferitelor metode de generare a mişcării roboţilor. Elaborarea programelor de calcul pentru generarea mişcării roboţilor prin metodele studiate.*

[*] ***Contract Nr.5004/1996, Tema 377, CNCSU, *Planificarea şi generarea mişcării la roboţi. Simularea mişcării unui robot pe baza modelului dinamic al acestuia.*

BUPT

CUPRINS

INTRODUCERE....1

1.PROBLEMATICA GENERĂRII TRAIECTORIILOR....6
1.1.Introducere....6
1.2.Tipuri de traiectorii şi posibilităţi de generare....7
1.3.Generarea mişcării şi sistemul de comandă....10
1.4.Generarea mişcării în spaţiul cuplelor cinematice....11
1.5.Generarea mişcării în spaţiul operaţional....12
1.6.Controlul mişcării unui robot....14

2.METODE DE GENERARE A TRAIECTORIILOR ÎN SPAŢIUL OPERAŢIONAL....19
2.1.Reprezentarea mişcării în coordonate carteziene....19
2.2.Traiectorii care unesc m *puncte* ale spaţiului de lucru....22
2.3.Generarea mişcărilor interpolate....25
2.3.1.Interpolarea liniară....25
2.3.2.Generarea mişcărilor standard prin abordare cinematică....26
2.3.3.Generarea mişcării prin inversare numerică....28
2.3.4.Respectarea restricţiilor de mişcare....29
2.3.4.1.Vitezele cuplelor cinematice conducătoare....29
2.3.4.2.Acceleraţiile cuplelor cinematice....32
2.3.5.Alte mişcări interpolate....33
2.3.5.1.Mişcări circulare....33
2.3.5.2.Mişcări oarecare....34
2.3.6.Performanţa optimală în viteză....34
2.4.Etapele procesului de conducere a unui robot....37
2.5.Generarea mişcării în spaţiul cartezian cu matricea conducere D(r)....38
2.5.1.Descrierea mişcării pe baza ecuaţiilor de transformare....38
2.5.2.Generarea mişcării între două situări succesive....40
2.5.3.Tranziţia între segmentele traiectoriei....44
2.6.Mişcarea punct cu punct. Formularea restricţiilor în spaţiul operaţional....48
2.6.1.Mişcarea punctului caracteristic al unui robot, în timp minim, cu abatere controlată faţă de *punctele* impuse în coordonate carteziene, cu păstrarea constantă a orientării efectorului final....48
2.6.1.1.Traiectoria şi forma legii de comandă....49
2.6.1.2.Criteriul de optimizare şi restricţiile cinematice....50
2.6.2. Mişcarea punctului caracteristic al unui robot, în timp minim, cu abatere nulă faţă de *punctele* impuse în coordonate carteziene, cu luarea în considerare a orientării efectorului final....56
2.7.Interpolarea cvasiliniară în spaţiul configuraţiilor, cu abatere controlată în spaţiul operaţional....59
2.8.Concluzii....61

BUPT

3.METODE DE GENERARE A TRAIECTORIILOR ÎN SPAŢIUL CUPLELOR CINEMATICE.....63
3.1.Generarea mişcării între două *puncte* de oprire.....64
3.1.1.Interpolarea polinomială.....64
3.1.2.Legea cu profil triunghiular de viteză.....67
3.1.3.Legea cu profil trapezoidal de viteză.....68
3.1.4.Profilul de viteză netezit.....70
3.1.5.Traiectoria critic amortizată.....72
3.1.6.Traiectoria cosinusoidală.....72
3.1.7.Traiectoria sinusoidală suprapusă peste rampă.....73
3.2.Generarea mişcării între două *puncte* de trecere.....74
3.2.1.Funcţia de interpolare polinomială.....75
3.2.2.Funcţia de interpolare sinosoidală.....75
3.2.3.Funcţia de interppolare sinusoidală ponderată în timp.....77
3.2.4.Interpolare liniară şi tranziţie parabolică.....77
3.2.5.Utilizarea polinoamelor de gradul trei.....79
3.3.Coordonarea cuplelor cinematice ale unui robot.....81
3.3.1.Calculul timpului minim în funcţie de tipul de interpolare folosit.....81
3.3.2.Metoda timpului minim pentru coordonarea cuplelor cinematice.....82
3.3.3.Coordonarea cuplelor între *punctele* de trecere.....84
3.3.4.Coordonarea cuplelor cinematice
ale unui robot prin metoda proporţională.....86
3.4.Program pentru generarea mişcării roboţilor industriali în
spaţiul cuplelor cinematice.....89
3.5.Concluzii.....92

4.INTERPOLAREA TRAIECTORIILOR FOLOSIND FUNCŢII SPLINE.....94
4.1.Introducere.....94
4.2.Interpretarea fizică a interpolării cu funcţii spline.....95
4.3.Interpolarea traiectoriilor în coordonatele cuplelor cu *puncte* de control.....96
4.3.1.Calculul traiectoriei de tip 4-3-4.....98
4.3.2.Traiectorie spline cubică (5 cubice).....100
4.4.Utilizarea funcţiilor spline pentru generarea unei traiectorii
ce trece prin *n puncte* de precizie.....103
4.4.1.Ecuaţia segmentului spline dintre două *puncte* intermediare de precizie....104
4.4.2.Ecuaţia segmentului spline pentru primul şi ultimul segment.....106
4.4.3.Determinarea vitezelor şi respectarea restricţiilor de viteză şi acceleraţie...106
4.4.4.Exemplu de aplicare a metodei pentru
generarea mişcării robotului RTRTTR.....108
4.5.Interpolarea traiectoriilor spline cubice cu restricţii cinematice şi dinamice.....111
4.5.1.Determinarea funcţiei de interpolare.....112
4.5.2.Unicitatea soluţiei.....115
4.5.3.Optimizarea traiectoriilor polinomiale cubice ale cuplelor.....116
4.5.4.Existenţa soluţiei fezabile.....118
4.5.5.Algoritmul de optimizare.....119
4.6.Generarea traiectoriilor polinomiale cvartice cu erori pătratice minime ale cuplelor.123
4.7.Interpolarea unei traiectorii prin "glisarea"unei cubice ce trece prin patru *puncte*....126
4.8.Concluzii.....129

BUPT

5.GENERAREA TRAIECTORIILOR UNUI ROBOT PE BAZA MODELULUI DINAMIC.....131
5.1.Modelul dinamic al unui manipulator.....131
5.2.Determinarea ecuaţiilor de mişcare pe baza modelului dinamic pentru un manipulator cu trei grade de libertate.....134
5.3.Studiul dinamicii unui robot KUKA tip 364/10.0.....139
5.3.1.Modelul geometric.....139
5.3.2.Modelul dinamic.....144
5.3.3.Rezolvarea modelului dinamic direct în condiţii date de mişcare.....146
5.4.Variaţia elementelor torsorului de reducere şi torsorul din cuplele unui robot RTRTTR.....147
5.4.1.Variaţia elementelor torsorului de reducere cu schimbarea sistemului de referinţă.....148
5.4.2.Torsori echivalenţi în cuplele robotului.....150
5.5.Precizia parcurgerii traiectoriei unui robot.....152
5.5.1.Precizia cinematică a parcurgerii traiectoriei.....152
5.5.1.1.Calculul abaterilor pe baza modelului diferenţial.....153
5.5.1.2.Cazul particular al robotului cu trei grade de libertate.....154
5.5.2.Precizia dinamică a parcurgerii traiectoriei unui robot.....155
5.5.2.1.Evaluarea preciziei de poziţionare dinamică.....155
5.5.2.2.Cazul particular al robotului cu trei grade de libertate.....156
5.6.Influenţa vibraţiilor din cuplele unui robot asupra poziţiei efectorului final.....158
5.6.1.Prezentarea metodei.....158
5.6.2.Exemplificarea metodei pentru un robot de tip RT.....160
5.7.Studiul erorilor dinamice datorate deformaţiilor elementelor unui robot.....163
5.7.1.Relaţia forţe-deplasări pentru o structură idealizată.....163
5.7.2.Relaţia forţe-deplasări pentru un element finit de structură.....165
5.7.3.Matricea de rigiditate a unui element de structură.....168
5.7.4.Algoritm pentru calculul deformaţiilor.....170
5.8.Concluzii.....170

6.STUDIU EXPERIMENTAL PRIVIND CONDUCEREA UNUI ROBOT CU COMANDĂ PUNCT CU PUNCT PE O TRAIECTORIE CONTINUĂ ÎN SPAŢIUL OPERAŢIONAL.....173
6.1.Descrierea robotului SCORBOT-ER III şi determinarea spaţiului de lucru.....173
6.2.Modelul geometric al robotului SCORBOT-ER III.....175
6.3.Generarea mişcării în spaţiul operaţional.....177
6.4.Concluzii.....194

7.GENERAREA ŞI PLANIFICAREA MIŞCĂRII ÎNTR-UN SPAŢIU CU OBSTACOLE.....196
7.1.Introducere.....196
7.2.Modelarea spaţiului de lucru prin metoda grilei neomogene.....197
7.3.Determinarea frontierelor C-obstacolelor. Poligon convex în mişcare de translaţie..199
7.4.Metoda grafului vizibilităţii.....206
7.5.Metoda drumului liber.....210
7.6.Planificarea traiectoriei prin metoda descompunerii celulare exacte.....212
7.6.1.Principiul metodei descompunerii celulare.....212
7.6.2.Principiul metodei descompunerii celulare exacte.....213
7.6.3.Algoritmul metodei descompunerii celulare exacte.....215
7.7.Concluzii.....219

BUPT

8.PLANIFICAREA MIŞCĂRII ROBOŢILOR PRIN METODELE CÂMPULUI POTENŢIAL.....221
8.1.Introducere.....221
8.2.Principiul metodelor câmpului potenţial.....222
8.2.1.Câmpul potenţial în cazul translaţiei.....223
8.2.1.1.Funcţia potenţial atractiv.....224
8.2.1.2.Funcţia potenţial repulsiv.....225
8.2.2.Câmpul potenţial în cazul general.....227
8.2.2.1.Extensii ale noţiunilor anterioare.....227
8.2.2.2.Potenţialul atractiv.....228
8.2.2.3.Potenţialul repulsiv.....229
8.3.Metode de planificare a mişcării utilizând câmpul potenţial
(potential-guided path planning).....231
8.3.1.Planificarea "în adâncime"(depth-first planning).....232
8.3.2.Planificarea "pe nivel" (best-first planning).....234
8.3.3.Planificarea variaţională.....236
8.3.4.Generarea mişcării unui robot tip maşină prin metoda câmpului potenţial 237
8.3.4.1.Modelarea robotului tip maşină.....237
8.3.4.2.Implementarea algoritmului.....238
8.4.Alte funcţii potenţial.....241
8.4.1.Noţiunea de funcţie de navigaţie.....241
8.4.2.Potenţialul grilă.....242
8.4.2.1.Funcţii de navigaţie numerice simple.....242
8.4.2.2. Funcţii de navigaţie numerice îmbunătăţite.....243
8.5.Calculul traiectoriei în cazul în care obstacolul *B* este o bară.....249
8.6.Potenţiale eliptice.....250
8.6.1.Calculul minimului local pentru C_{loc} din exemplul prezentat.....252
8.6.2.Potenţialul repulsiv în cazul unei suprafeţe definite "a priori".....253
8.6.3.Exemplu de aplicare a potenţialului repulsiv eliptic
în cazul suprafeţe definite "a priori", într-un spaţiu bidimensional.....254
8.7.Program pentru generarea unei traiectorii plane prin metoda câmpului potenţial.....256
8.7.1.Prezentare generală.....256
8.7.2.Exemple de utilizare.....256
8.8.Concluzii.....259

CONSIDERAŢII FINALE. CONTRIBUŢII ORIGINALE. PERSPECTIVE.....260

BIBLIOGRAFIE.....265

BUPT