

UNIVERSITATEA POLITEHNICA TIMIȘOARA
FACULTATEA DE MECANICĂ

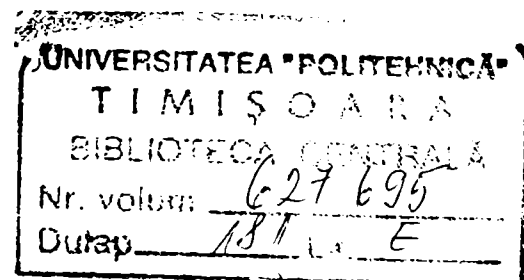
ing. Oana Benghia căs.Căuș

CONTRIBUȚII LA STUDIUL MODELĂRII SISTEMELOR
COMPLEXE DE CORPURI DE TIP ROBOȚI-SPAȚIU DE LUCRU

Teză de doctorat

BIBLIOTECA CENTRALĂ
UNIVERSITATEA "POLITEHNICA"
TIMIȘOARA

Conducător științific:
Prof.dr.ing. Doina Drăgulescu
m.c. al Academiei de Științe Tehnice din România



Timișoara
2000

Cuprins

I.	Introducere	1
II.	Reprezentarea parametrică	8
	II.1. Prezentare generală a curbelor parametrice	8
	II.2. Reprezentarea parametrică a curbelor tridimensionale	10
	○ II.2.1 Curbele cubice Bézier	12
	○ II.2.2 Unirea segmentelor de curbe cubice Bézier	15
	○ II.2.3 Proprietăți ale curbelor Bézier	16
	○ II.2.4 Curbele B-spline	17
	○ II.2.5 Curbele B-spline uniforme	19
	○ II.2.6 Curbele B-spline neuniforme	22
	○ II.2.7 Proprietăți ale curbelor B-spline	24
	II.3. Editarea curbelor parametrice	25
	II.4. Suprafețe cubice biparametrice	31
	○ II.4.1. Porțiuni de suprafață Bézier	31
	○ II.4.2 Combinarea porțiunilor de suprafață Bézier	33
	○ II.4.3 Porțiuni de suprafață B-spline	35
	○ II.4.4. Editarea suprafețelor parametrice	39
	II.5. Reprezentarea parametrică a spațiilor de lucru proprii ale roboților prin utilizarea Matlab	41
	○ II.5.1. Considerații generale	41
	○ II.5.2. Funcții utilizate în programele Matlab scrise pentru generarea reprezentărilor grafice	44
	○ II.5.3. Modelarea suprafețelor descrise de efectorul final	46

III.	Reprezentarea geometrică	67
III.1.	Generalități	67
III.2.	Reprezentarea prin sfere	69
○	III.2.1. Generalități	69
○	III.2.2. Considerații matematice asupra reprezentării prin sfere	72
▪	III.2.2.1. Noțiuni și relații fundamentale	72
▪	III.2.2.2. Reprezentarea exterioară a obiectelor prin sfere	88
▪	III.2.2.3. Reprezentarea interioară a obiectelor prin sfere	100
○	III.2.3. Considerații topologice asupra reprezentării prin sfere	111
▪	III.2.3.1. Reprezentarea exterioară	111
▪	III.2.3.2. Reprezentarea interioară	137
○	III.2.4. Program de sfericizare	154
▪	III.2.4.1. Prezentarea interfeței și utilizarea programului	155
▪	III.2.4.2. Algoritmi de sfericizare	158
IV.	Reprezentarea prin subdivizare spațială	181
IV.1.	Considerații generale	181
IV.2.	Modelarea spațiului cu obstacole	184
○	IV.2.1. Modelarea 2D a spațiului de lucru	186
▪	IV.2.1.1. Metoda grilei omogene	186
▪	IV.2.1.2. Metoda grilei neomogene	192
○	IV.2.2. Modelarea 3D a spațiului de lucru	204
V.	Concluzii și contribuții originale	208
	Bibliografie	213

I. Introducere

Cele mai simple mijloace de acțiune asupra mediului înconjurător sunt mișcarea și manipularea diferitelor obiecte. Vederea oferă informații asupra mediului, în timp ce limbajul și vorbirea servesc la a stabili relații cu ceilalți, dar acțiunea asupra mediului înconjurător se face prin mișcarea subiectului urmărit sau a obiectelor adiacente: prin apucare, transport, împingere, etc. Chiar și o sarcină cu grad de complexitate redus, ce trebuie îndeplinită de un robot, implică o combinație destul de complicată de mișcări. De aceea, este necesară o abordare mai profundă atât a modului de reprezentare a corpurilor cât și a mișcării [4], [156], [157].

Sistemele inteligente sunt concepute să acționeze în același mod ca și subiecții umani: prin mișcare și manipulare. Dacă un robot este o mașină cu abilități în mișcare și/sau de manipulare, atunci una din cele mai importante probleme de rezolvat este de a le planifica mișcările, ceea ce implică modelarea spațiului de lucru, cu obstacolele pe care le conține, și a robotului, ca entitate de formă complexă și variabilă.

Pe parcursul abordării problemei planificării mișcării s-au dezvoltat mai multe tipuri de reprezentări ale corpurilor din spațiul de lucru, pornind de la cazul simplu al roboților mobili reprezentați ca poligoane în spațiul 2D. Ulterior, dezvoltarea metodelor de grafică computerizată a dus și la dezvoltarea complexității reprezentărilor ierarhizate pe nivele. Aceste studii au făcut posibilă obținerea de rezultate importante și în spațiul 3D, fiind chiar materializate prin obținerea unor tehnici speciale de prelucrare a metalelor, de reconstruire a unor părți distruse dintr-un corp a cărui geometrie este cunoscută doar parțial, etc. Astfel, s-au creat modele utilizând diverse forme geometrice elementare (con, cilindru, sferă, etc.) obținându-se pe baza utilizării lor multe concluzii practice [170], [171].

Planificarea mișcărilor fiind o aplicație fundamentală în robotică a principiilor inteligenței artificiale, are drept principal scop înzestrarea unui robot autonom cu capacități primare de generare a propriilor mișcări. O primă condiție ar fi evitarea

coliziunii cu posibile obstacole ce se pot afla pe traiectorie. Deși această capacitate este deținută de cele mai inferioare animale, și în principiu nu necesită multă inteligență, soluția problemei nu este una simplă. Motivația se află în faptul că majoritatea mișcărilor ființelor vii sunt coordonate în mod inconștient, depinzând mai mult de sistemul de percepție fără a interveni și procesul logic deductiv [12], [15], [13], [179].

Planificarea mișcărilor poate fi considerată ca problema realizării algoritmilor pentru a calcula automat o traiectorie continuă pentru o mulțime de obiecte (posibil legate) astfel încât să se mute de la o poziție la alta evitând coliziunile cu alte obiecte fixe sau având mișcare proprie. Pentru un robot cu bază fixă problema se poate formula mai simplu prin alegerea unei traiectorii ferite de coliziuni pentru brațul robotului, între două poziții, în cazul unui spațiu închis. Soluția acestei probleme este un pas înainte spre planificarea acțiunilor robotului la nivel de sarcină; aceasta înseamnă că se pot ignora secvențele intermediare ce trebuie urmate pentru realizarea unei sarcini [191], [192].

Planificarea fără coliziune poate fi calificată ca o problemă de dezvoltare a algoritmilor de calcul automat a unei traiectorii continue și sigure pentru o mulțime dată de obiecte, astfel încât ele să se miște fără a se izbi de obstacole [195], [196]..

Se poate spune că planificarea mișcării este una dintre cele mai complexe probleme din domeniul roboticii. Complexitatea depinde de răspunsul la două întrebări: *cine se mișcă?* și *unde se mișcă?* Cu cât sistemul care se mișcă este mai complex iar spațiul în care se mișcă mai populat cu alte obiecte, problema are complexitate mai mare. Însăși trecerea de la cazul bidimensional la cel tridimensional complică problema chiar dacă rămâne în studiu același robot mobil în același spațiu de lucru. Dificultatea studiului sporește cu atât mai mult cu cât robotului îi crește numărul gradelor de libertate, pentru că aceasta înseamnă o arhitectură mai complicată și o modelare mai dificilă și mai complexă atât a robotului, ca sistem de corpuri, cât și a spațiului propriu de lucru definit prin mișcarea efectorului final [185], [186].

Este evident că problema reprezentării obiectelor are o importanță crucială pentru robotică. În domeniul reprezentării spațiale, au fost folosite pe parcursul timpului mai multe tipuri de modele 3D și 2D, mai ales în aplicațiile grafice computerizate: geometria corpurilor, reprezentarea limitelor, descompunerea în celule sau sub-spații de altă natură decât geometrică, volume acoperitoare, arbori de decizie, etc. [84], [87].

În general, ar fi necesar un compromis între acuratețe și simplitate, un model prea detaliat putând fi prea complex, iar un model prea simplu putând fi mai puțin precis.

În funcție de structurile de date și de informațiile deținute despre corpuri modelele de solide sunt complicate de utilizat și prezintă limitări atât în ceea ce privește complexitatea reprezentării cât și a acurateții geometriei acestora [91], [92].

Modelele de solide se pot obține prin următoarele tehnici mai frecvente:

- Parametric
- Geometrie constructivă
- Reprezentare a frontierelor
- Enumerare spațială
- Baleiere
- Analitic.

La conceperea lor, formele modelelor solide pot fi gândite în termenii unor forme geometrice elementare (prismă, cilindru, sferă, con, etc.) sau în termenii unor forme elementare din punct de vedere al generării tehnologice (de rotație, extrudate, etc.).

Formele geometrice elementare, necesare pentru construirea unor solide complexe se pot obține utilizând accesul la forme primitive standard care sunt fie puse la dispoziție de către un sistem profesional de proiectare, fie pot fi utilizate în conceperea unor programe originale de construcție 3D [1], [2], [5], [7], [88], .

Utilizarea doar a primitivelor standard poate limita aplicabilitatea sistemului. De aceea este necesar să se ofere utilizatorului posibilitatea de a defini, după necesități, entități geometrice primare. Se pot astfel crea curbe pornind de la puncte, suprafețe pornind de la curbe și volume pornind de la suprafețe [10], [11], [19], [22], [23], [25].

Indiferent de modul de obținere, modelele de solide trebuie să satisfacă mai multe cerințe:

- Să fie valide, fiecărui model să-i corespundându-i un obiect real;
- Să fie universale, aceeași tehnică putând fi utilizată pentru orice obiect 3D;
- Să fie unic interpretabile, unei reprezentări corespundându-i un singur obiect;
- Să nu fie ambigue, unui obiect dat corespundându-i un singur model;

- Să fie complet, generând în toate cazurile entități valide ca operanzi pentru toate operațiile;
- Să fie concise, informația conținută în model fiind minimă.

În grafica computerizată s-au dezvoltat diverse metode de reprezentare a obiectelor tridimensionale. Unele din acestea s-au dedus din aplicații, iar structura de date s-a determinat în totalitate prin strategia de modelare. Alte forme sunt determinate prin algoritmi de proiectare [8], [9], [17], [26], [44].

Factorii pe care orice reprezentare trebuie să-i aibă în vedere, în principal, sunt :

- structura de date, forma algoritmilor de procesare și designul programelor fixe de hardware;
- costul procesării unui obiect 3D;
- gradul de aproximare al imaginii finale a unui obiect ;
- ușurința editării formei obiectului.

După importanța și frecvența lor de folosire sunt utilizate patru forme distincte de reprezentare:

1. Reprezentarea poligonală, în care obiectele sunt approximate printr-o rețea de poligoane plane .

2. Reprezentarea prin porțiuni parametrice, în care obiectele sunt reprezentate exact prin rețele poligoane curbilinii, numite în mod uzual petice. Acestea sunt exprimate prin polinoame de două variabile parametrice și, de obicei, sunt cubice.

3. Reprezentarea prin construcția geometrică a solidului, care este folosită în modelarea exactă a unui solid, printr-o colecție de corpuri geometrice primitive ca: sfere, cilindrii, poliedre.

4. Reprezentarea prin tehnici de subdiviziune spațială, ce constă din reprezentarea unui obiect prin asocierea de celule dintr-un spațiu divizat în celule egale sau inegale, tehnica de asociere fiind bazată pe metode selectate din teoria grafurilor sau pe construcția de arbori.

În prezentarea făcută, **1** și **4** sunt metode ce aproximează controlabil forma obiectului pe care-l reprezintă, iar metodele **2** și **3** sunt reprezentări exacte. Din punct de vedere al realizării reprezentării, **1** și **2** realizează corpul prin reprezentări de suprafețe, care în cazul **2** rezultă din reprezentări de curbe, iar **3** și **4** se obțin direct ca reprezentări de volum.

Ocazional, ca modalitate de reprezentare a obiectelor sunt utilizate funcțiile implicite. Acestea sunt puțin folosite în grafica pe calculator, pentru că există un număr limitat de obiecte care pot fi reprezentate în acest fel și pentru că este o formă total nepotrivită din punctul de vedere al proiectării [64], [68], [74], [79], [82], [99].

Prezenta teză tratează modelarea corpurilor de formă complexă prin metodele 2, 3 și 4, utilizând pentru modelare atât programe originale cât și software profesional. Metodele 2 și 3 sunt folosite pentru reconstrucția unor corpuri complexe (roboții și spațiile proprii descrise de efectorul lor final), iar metoda 4 a fost utilizată pentru rezolvarea unor probleme de planificare, reprezentându-se spațiul de lucru în care un robot mobil evoluează. Teza a fost gândită astfel încât să ofere o multitudine de soluții fezabile pentru reprezentarea de corpuri și spații complexe, fiind structurată astfel:

Capitolul I, Introducere, este reprezentat de prezenta introducere prin care se realizează expunerea principială a problemei de rezolvat, importanța sa și enumerarea conținutului capitolelor următoare.

Capitolul II, Reprezentarea parametrică, care se ocupă de reprezentarea parametrică a curbilor și suprafețelor, expunând modul de abordare a reprezentării curbilor Bézier și B-spline, precum și construcția porțiunilor de suprafață pe baza acestor tipuri de curbe.

O curbă parametrică e definită printr-o mulțime discretă de puncte cunoscute ca puncte de control împreună cu un set de funcții de bază. Această metodă de specificare a curbei este complet diferită față de cea matematică normală, care are forma unei funcții implicite.

Funcțiile de bază folosite pentru a trasa și aranja punctele de control într-un segment de curbă pot fi reprezentate de orice mulțime formată dintr-un număr oarecare de funcții de bază cu proprietăți diferite utilizate în grafica computerizată pentru a controla forma unei curbe printr-o interfață interactivă. Dintre acestea curbele Bézier sunt folosite cu succes atât datorită simplității lor cât și din cauză că obiectele pot fi cu ușurință reprezentate dacă sunt utilizate suprafețele Bézier. Curbele Bézier au anumite dezavantaje care pot fi înlăturate folosind curbele B-spline.

În cadrul acestui capitol se prezintă reprezentările prin curbe și suprafețe parametrice utilizând două categorii de programe: unele originale create de autoare și altele din sistemul profesional de software Matlab. Sunt exemplificate reprezentările spațiilor de lucru ale roboților de mai multe tipuri, pornind de la cel mai simplu manipulator de tip RT, la un robot complex de tip RRRRRR.

Capitolul III, Reprezentarea geometrică, tratează acest tip de reprezentare a corpurilor de formă complexă, fiind aleasă din motive justificate teoretic și practic drept primitivă a reprezentării, sfera. Orice model este compus din două mulțimi de sfere: o mulțime realizând reprezentarea exterioară, ce aproximează prin surplus forma sistemului real prin acoperirea sa cu sfere și o altă mulțime realizând reprezentarea interioară, compusă din sfere plasate în interiorul sistemului real ce aproximează forma corpului prin deficit.

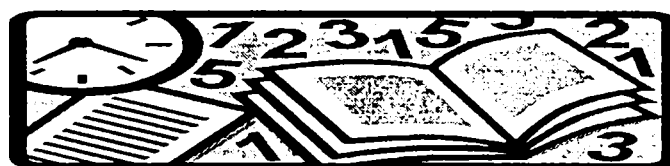
Prima modalitate de reprezentare este în mod uzual utilizată la modelarea corpurilor, cea de-a doua dovedindu-se foarte utilă în procesul de detectare a coliziunilor la problemele de planificare a mișcărilor. Problema inițială este de a realiza modelul spațial al roboților și al obstacolelor și aceasta se rezolvă prin reprezentare exterioară, iar ulterior aceste modele la care se adaugă reprezentarea interioară sunt aplicate în problemele de mișcare, mai precis în detectarea coliziunilor și planificarea mișcărilor în 3D. Programele proprii realizate de autoare sunt utilizate la reprezentarea prin acoperire, știut fiind faptul că în problemele de planificare a mișcărilor toate corpurile din spațiul de lucru se aproximează prin surplus, pentru evitarea coliziunilor. Capitolul III, prezintă astfel algoritmi de sfericizare exterioară pentru toate primitivele geometrice uzuale, cu ajutorul cărora se poate compune orice sistem complex de corpuri.

Capitolul III are o justificare teoretică profundă a alegerii reprezentării prin sfericizare, fiind tratată reprezentarea de acest tip a tuturor celorlate primitive geometrice (cilindru, con, trunchi de con, etc.). Este astfel evident că orice altă reprezentare utilizând primitive geometrice de alt tip, se poate transforma în reprezentare prin sfere dacă se construiește un sistem coerent de teoreme ce asigură acoperirea sau umplerea cu sfere a primitivelor respective. Pe baza acestei considerații, în acest capitol se realizează o expunere exhaustivă a acestui tip de reprezentare, din punct de vedere geometric și topologic. Programele de sfericizare create se bazează pe sistemul de teoreme și considerații teoretice concepute în acest scop.

Capitolul IV, Reprezentarea prin subdivizare spațială, se ocupă de modelarea corpurilor aflate în spațiul de lucru al unui robot mobil în vederea planificării mișcării sale. Metoda de reprezentare cuprinde divizarea întregului spațiu în celule egale sau neegale și aproximarea în acest mod a formei corpurilor. Se prezintă algoritmi de calcul atât pentru reprezentarea 2D cât și pentru reprezentarea 3D, precum și programele create pentru modelare în ambele situații. Ca și la sfericizare, ideea de

bază a reprezentării a fost cea de acoperire, ceea ce implică aproximarea prin surplus formei corpurilor și evitarea coliziunilor.

Capitolul V, Concluzii și contribuții originale, prezintă în mod succint tot ceea ce prezenta teză aduce nou în domeniul reprezentării corpurilor de formă complexă, cu aplicație în robotică. Sunt reluate, pe rând, capitolele tezei și scoase în evidență contribuțiile teoretice și aplicative la cele trei tipuri de reprezentări utilizate.



Autoarea consideră că în aceste rânduri este locul potrivit pentru a mulțumi colectivului Departamentului de Calculatoare din cadrul Facultății de Automatică și Calculatoare, precum și conducerii facultății pentru climatul de muncă ce a făcut posibilă finalizarea tezei.

Mulțumesc membrilor comisei de doctorat în speranța că vor analiza cu răbdare conținutul acestei lucrări și vor găsi elemente științifice care să merite apreciere pozitivă..

Alese mulțumiri li se cuvin părinților mei, Dorina și Bela Rendi, în calitate de autori morali ai lucrării și mai ales aceluia dintre ei care nu mai este și care a contribuit decisiv la orientarea mea în abordarea acestei tematici. Prietenia care l-a legat de conducătorul științific al tezei, dna. prof.dr.ing. Doina Drăgulescu, a fost pentru mine un imbold în plus în abordarea cercetării și mi-a întărit dorința de a o finaliza.

Soțului și fiului meu, care au avut înțelegere pentru timpul pe care l-am consacrat acestei lucrări, le mulțumesc și îi asigur că voi încerca să recuperez momentele pierdute.

II. Reprezentarea parametrică

Reprezentarea parametrică a solidelor și a curbelor este o unealtă de bază în grafica pe calculator, în special în proiectarea asistată. Tehnicile ce au fost dezvoltate inițial doar pentru a modela caroserii de automobil și fuselaje de avion sunt în prezent aplicate în numeroase și variate ramuri ale graficii computerizate. Aceste procedee sunt folosite astăzi în orice proces de modelare a obiectelor, design interactiv și în animație. Una din aplicațiile cele mai importante se regăsește în robotică, domeniul roboților mobili fiind în mod special dependent de o corectă modelare a corpurilor. Toate metodele de planificare a mișcărilor, în spații statice sau dinamice, pretind o modelare corectă pentru a asigura precizia necesară deplasării [3], [24], [31], [32], [47].

Abordarea uzuală a examinării reprezentărilor parametrice începe, de regulă, cu o descriere a curbelor tridimensionale și continuă cu generalizarea acestora la suprafețe. Deoarece proprietățile curbelor se extind și la suprafețe, se impune în primul rând tratarea acestora și extinderea procedurilor stabilite la cazul suprafețelor. În plus, acestea își au propriul lor loc în grafica computerizată.

II.1. Prezentare generală a curbelor parametrice

O **curbă parametrică** e definită printr-o mulțime discretă de puncte cunoscute ca **puncte de control** împreună cu un set de **funcții de bază** sau **funcții de combinare**. Această metodă de specificare a curbei este complet diferită față de cea matematică normală, care are forma unei funcții implicite.

Funcțiile de bază folosite pentru a trasa și aranja punctele de control într-un segment de curbă pot fi reprezentate de orice mulțime formată dintr-un număr oarecare de funcții de bază cu proprietăți diferite utilizate în grafica computerizată

pentru a controla forma unei curbe printr-o interfață interactivă. Dintre acestea curbele Bézier sunt folosite cu succes atât datorită simplității lor cât și din cauză că obiectele pot fi cu ușurință reprezentate dacă sunt utilizate suprafețele Bézier. Curbele Bézier au anumite dezavantaje care pot fi înlăturate folosind curbele B-spline [6], [58], [69].

Se observă din figura II.1 că patru puncte de control, P_0, \dots, P_3 , sunt prelucrate de un algoritm care generează o curbă. În acest caz curba trece prin două din puncte. Există câte o funcție de bază definită pe domeniul în care variază parametrul u pentru fiecare punct de control, iar curba $Q(u)$ este creată folosind punctele de control pentru a trasa funcțiile de bază. În figura II.1 funcțiile de bază sunt funcții Bézier. Astfel:

$$x(u) = \sum_{i=0}^3 P_{xi} \cdot B_i(u) \quad (\text{II.1.1})$$

specifică faptul că se obține componenta pe x a lui $Q(u)$ prin însumarea proiecțiilor pe Ox ale fiecărui punct de control trasat prin valoarea funcției de bază în u . Analog:

$$y(u) = \sum_{i=0}^3 P_{yi} \cdot B_i(u) \quad (\text{II.1.2})$$

și:

$$z(u) = \sum_{i=0}^3 P_{zi} \cdot B_i(u) \quad (\text{II.1.3})$$

sau mai exact:

$$Q(u) = \sum_{i=0}^3 P_i \cdot B_i(u) \quad (\text{II.1.4})$$

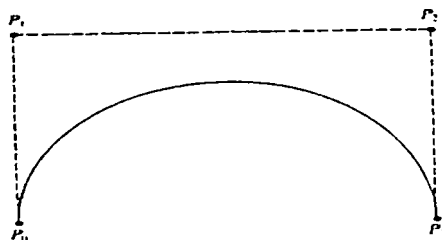


Fig.II.1 O curbă Bézier definită prin patru puncte de control

Specificarea unui segment de curbă (sau a unei porțiuni de suprafață) printr-o mulțime de puncte de control este baza unei metode de design interactiv în proiectarea asistată de calculator, după pozițiile cunoscute ale acestor puncte fiind trasată și vizualizată curba. Dacă forma sa nu este satisfăcătoare, se specifică o nouă mulțime de puncte și operațiile se reiau. Se pot trasa astfel segmente de curbe care se utilizează ulterior la construirea unei curbe formată dintr-un număr de segmente de curbă unite între ele pe baza unor condiții de continuitate.

În grafica pe calculator apare tendința de a folosi curbele cubice. Aceasta se datorează faptului că prezintă o suficientă flexibilitate a formei pentru cele mai multe aplicații, dar și faptului că o curbă de ordin mai mare introduce și costuri mai mari. Ele sunt, de altfel, adevărate curbe spațiale, spre deosebire de cele pătratice care sunt conținute într-un plan. O curbă formată din segmente pătratice poate fi numai un set de segmente bidimensionale pe porțiuni. Această situație nu este satisfăcătoare atunci când asemenea segmente sunt utilizate pentru a modela corpuri tridimensionale [75], [80], [81].

Când o curbă este compusă din segmente de curbă, proprietățile ce exprimă continuitatea depind de modul în care au fost unite segmentele. Simpla unire a două segmente printr-un capăt comun implică continuitatea geometrică G^0 . Dacă vectorii tangenți la fiecare segment de curbă în capătul comun au aceeași direcție și module diferite atunci curba are continuitatea geometrică G^1 . Dacă vectorii tangenți sunt identici, curba prezintă continuitate de gradul întâi sau continuitate C^1 .

În general, continuitatea parametrică C^n este definită astfel:

Dacă direcțiile și mărimile derivatelor $\frac{d^n Q(u)}{dt^n}$ sunt egale pentru cele două segmente la capătul comun, atunci curba prezintă continuitatea C^n .

II.2. Reprezentarea parametrică a curbelor tridimensionale

Un exemplu practic al unei curbe tridimensionale este traiectoria unui punct care se mișcă în spațiu (figura II.2). Poziția sa e definită printr-un vector r , funcție de timp t . Aceasta dă o așa-zisă descriere parametrică a curbei, sub forma a trei ecuații în t :

$$\begin{aligned}x &= x(t) \\ y &= y(t) \\ z &= z(t)\end{aligned}\tag{II.2.1}$$

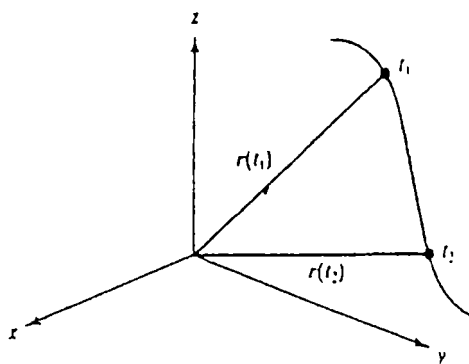


Fig.II.2 Traiectoria 3D a unui punct

Astfel se poate descrie parametric orice curbă în spațiu, folosind un parametru u , curba cubică fiind dată de:

$$\begin{aligned}x(u) &= a_x u^3 + b_x u^2 + c_x u + d_x \\ y(u) &= a_y u^3 + b_y u^2 + c_y u + d_y \\ z(u) &= a_z u^3 + b_z u^2 + c_z u + d_z\end{aligned}\tag{II.2.2}$$

unde: $0 \leq u \leq 1$.

Există multiple motive pentru a folosi în grafica pe calculator exprimarea parametrică în locul funcțiilor implicite și anume:

- Punctele de pe o curbă pot fi precizate direct, în loc să fie necesară rezolvarea unor ecuații neliniare sub formă implicită pentru fiecare punct în parte
- Curbele parametrice (și suprafețele) pot fi transformate cu ușurință prin transformări. Aplicarea unei transformări liniare reprezentării parametrice a curbei realizează chiar o transformare a curbei.
- Majoritatea aplicațiilor de design folosesc curbe complexe (și/sau suprafețe) care nu pot fi exprimate prin funcții implicite simple, pe când reprezentarea parametrică permite descrierea pe porțiuni, iar o curbă sau o suprafață poate fi dată printr-un set de polinoame pe porțiuni.

(Polinoamele de grad mare pot descrie curbe complexe, dar necesită un număr mare de coeficienți, ceea ce poate introduce oscilații nedorite în curbă).

II.2.1 Curbele cubice Bézier

Formularea Bézier a unei curbe cubice implică specificarea unei mulțimi de puncte de control, din care este obținut polinomul cubic. Această formă rezultă pornind de la anumite funcții de bază sau de combinare.

O metodă simplă de obținere, presupune rescrierea ecuațiilor (II.2.2) ca o singură ecuație vectorială:

$$\mathbf{Q}(u) = \mathbf{a}u^3 + \mathbf{b}u^2 + \mathbf{c}u + \mathbf{d} \quad (\text{II.2.3})$$

acesta fiind modul normal de exprimare a unui polinom cubic parametric. Folosind funcțiile de bază Bézier sau funcțiile de combinare Bernstein:

$$(1-u)^3 \quad 3u(1-u)^2 \quad 3u^2(1-u) \quad u^3 \quad (\text{II.2.4})$$

polinomul este exprimat prin aceste funcții și patru puncte de control:

$$\mathbf{Q}(u) = \mathbf{P}_0(1-u)^3 + \mathbf{P}_1 3u(1-u)^2 + \mathbf{P}_2 3u^2(1-u) + \mathbf{P}_3 u^3 \quad (\text{II.2.5})$$

sau sub forma matriceală:

$$\mathbf{Q}(u) = \mathbf{U} \cdot \mathbf{B} \cdot \mathbf{P} = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \quad (\text{II.2.6})$$

Formularea matriceală este utilă când se urmărește implementarea hardware a problemei reprezentării curbelor spațiale prin polinoame de gradul 3.

Figura II.1 prezintă o curbă și poligonul său caracteristic. P_0 , P_1 , P_2 și P_3 sunt patru puncte de control care definesc curba, P_0 și P_3 fiind capete. Ele se numesc **puncte de control** deoarece prin mutarea lor în spațiu este controlată sau influențată forma curbei (figura II.3). Poligonul format prin unirea punctelor de control se numește **poligon caracteristic** sau de control.

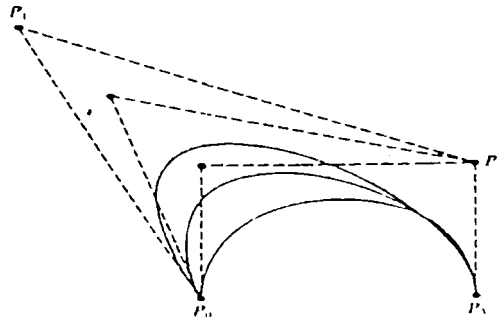


Fig.II.3 Efectul mutării unui punct de control

Dacă este necesar să se reproducă forma unei curbe oarecare, se poate proceda în felul următor:

1. Se aproximează forma curbei formând un poligon de control
2. Se folosește acest poligon pentru a desena curba Bézier asociată.
3. Se ajustează punctele de control.
4. Se repetă operațiile 1. și 2. până când forma obținută este satisfăcătoare.

Revenind la funcția de bază (II.1.4), lucrând cu polinoame de gradul 3, rezultă patru funcții de bază, corespunzătoare și reprezentării din figura II.4, fiecare termen al sumei fiind produsul dintre un punct de control P_i și o funcție de combinare $B_{i,3}$ care este un polinom de gradul trei.

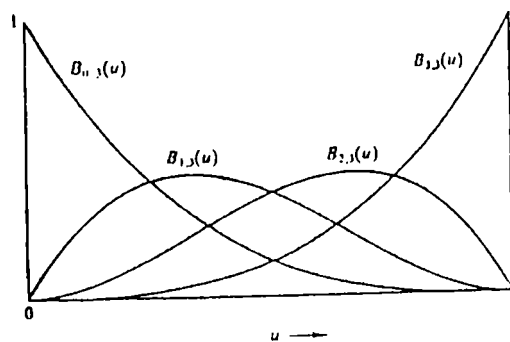


Fig.II.4 Funcțiile de bază Bézier pentru curbe cubice

Pentru o valoare particulară a lui u se însumează valorile obținute de la fiecare din cele patru funcții de combinare. P_0 are influența cea mai mare pentru $u=0$ ($B_{1,3}$, $B_{2,3}$ și $B_{3,3}$ sunt toate nule în acest punct). Pe măsură ce u crește către P_1 , B_0 și B_1 determină în principal forma curbei, $B_{2,3}$ și $B_{3,3}$ neavând vreo influență. Punctele P_1 și P_2 au efect maxim când $u=1/3$ respectiv $2/3$.

De remarcat este faptul că prin mutarea oricărui punct de control este influențată, în mai mică sau mai mare măsură, forma tuturor părților curbei. Acest mod global de control asupra formei curbei este dezavantajul major al curbelor Bézier.

Un alt punct important este faptul că reprezentarea parametrică permite curbe multiplu parametrică. De exemplu, dacă $P_0=P_3$ atunci curba rezultantă va fi o buclă închisă.

Mulțimea de funcții de bază sau de combinare date pentru polinoamele cubice:

$$B_{0,3}(u) = (1-u)^3 \quad B_{1,3}(u) = 3u(1-u)^2 \quad B_{2,3}(u) = 3u^2(1-u) \quad B_{3,3}(u) = u^3 \quad (\text{II.2.7})$$

este un caz particular de funcție de bază de grad n .

Astfel:

$$Q(u) = \sum_{i=0}^3 P_i \cdot B_{i,n}(u) \quad (\text{II.2.8})$$

unde:

$$B_{i,n}(u) = C(n,i) \cdot u^i \cdot (1-u)^{n-1} \quad (\text{II.2.9})$$

coeficientul binomial fiind:

$$C(n,i) = \frac{n!}{i!(n-i)!} \quad (\text{II.2.10})$$

Se remarcă faptul că numărul de puncte de control crește la $n+1$. Mărirea numărului de puncte de control introduce dificultăți prin mărirea gradului funcțiilor de combinare, aceste dificultăți reprezentând un motiv pentru care se preferă curbele B-spline. Avantajele unor curbe de ordin mai mare provin din posibilitatea asigurării unui grad convenabil de continuitate între segmentele de curbă. Dar, pe lângă costurile sporite de calcul la utilizarea curbelor de ordin mai mare, mai există și dezavantajul că, în aceste cazuri, corelația între poligoanele caracteristice formate de punctele de control și curbă devine mai slabă. De aceea, cel mai frecvent utilizate sunt cubicele pentru că sunt și rezonabil de simple, dar și suficient de flexibile pentru un design interactiv [91], [96], [105].

Pe baza celor prezentate este important să se sesizeze aspectele practice ale reprezentării unei curbe prin puncte de control. Cea mai importantă observație, în ceea ce privește interfața cu utilizatorul, este aceea că mutarea unui punct de control dă o modificare intuitivă a formei curbei. Adică, curba mimează forma poligonului de control.

O proprietate importantă din punct de vedere al conceperii algoritmilor ce realizează curbe (și suprafețe) este aceea că o curbă este întotdeauna închisă în înfășurătoarea convexă formată de poligonul de control. Ea provine din faptul că suma funcțiilor de bază are valoarea 1 pentru orice valoare a lui u .

Curbele fiind definite ca și combinații liniare de puncte de control, ele pot fi transformate prin orice transformare afină în spațiu (rotație, translație, etc.), prin aplicarea transformării potrivite mulțimii punctelor de control.

II.2.2 Unirea segmentelor de curbe cubice Bézier

Segmentele de curbă definite prin mulțimi de patru puncte de control pot fi unite pentru a forma curbe mai complexe decât cele care pot fi obținute dintr-o formă cubică polinomială, rezultând prin acest procedeu o curbă polinomială pe porțiuni.

O metodă alternativă de a reprezenta curbe mai complexe este mărirea gradului polinomului, dar aceasta are dezavantaje matematice și de calcul, fiind mai simplu să se împartă curba în segmente cubice.

Conectarea segmentelor de curbe cubice implică aplicarea unor restricții în punctele de joncțiune. Restricția implicită este G^0 , care semnifică faptul că punctul de

sfârșit al primului segment coincide cu punctul de început al celui de-al doilea. Continuitatea G^1 presupune coliniaritatea marginilor polinomului caracteristic, adică faptul că vectorii tangenți la sfârșitul unei porțiuni de curbă și începutul celei următoare, sunt coliniari.

Diferențele între continuitățile G^0 și G^1 pentru segmente de curbă Bézier sunt prezentate în figurile II.5 și II.6.

Dacă punctele de control a două segmente sunt Q_i și R_i atunci continuitatea G^1 este păstrată, spre exemplu pentru curba din figura II.6, dacă:

$$Q_3 - Q_2 = k(R_1 - R_0) \quad (II.2.11)$$

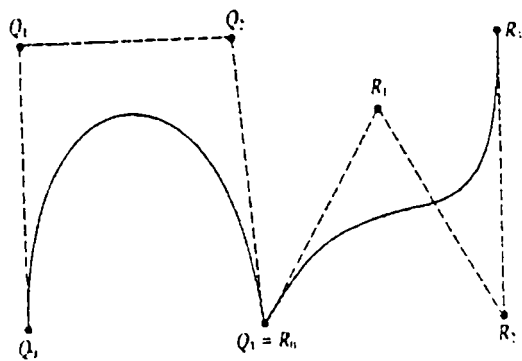


Fig.II.5 Continuitate G^0

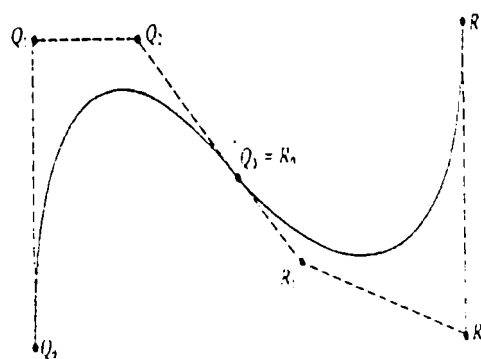


Fig.II.6. Continuitate G^1

Folosind condiții de acest tip, o curbă Bézier compusă este construită ușor adăugând pe rând câte un segment. Totuși, avantajul de a putea construi o formă compusă din segmente este oarecum diminuat de legăturile care sunt aplicate punctelor de control pentru a asigura condițiile de joncțiune. Acest dezavantaj poate fi atenuat numai prin mărirea gradului polinomului sau prin fracționarea segmentului în două sau mai multe segmente mai mici.

II.2.3 Proprietăți ale curbelor Bézier

Pe baza celor menționate se remarcă unele proprietăți ale curbelor Bézier:

- Gradul polinomului este întotdeauna cu unu mai mic decât numărul punctelor de control. În grafica pe calculator cele mai interesante s-au dovedit a fi cele de gradul trei.
- Curba urmărește forma poligonului de control și este conținută în înfășurătoarea convexă formată de punctele de control.
- Punctele de control nu exercită control local. Mutarea oricăruia dintre punctele de control afectează întreaga curbă în mai mică sau mai mare măsură.
- Primul și ultimul punct de control sunt capetele segmentului de curbă.
- Vectorii tangenți la curbă în capete coincid cu prima și ultima latură a poligonului de control.
- Curba nu oscilează față de oricare linie dreaptă mai des decât poligonul de control - aceasta este cunoscută sub numele de proprietatea diminuării variației.
- Curba poate fi transformată aplicând orice transformare afină (adică orice combinație de transformări liniare) punctelor sale de control.

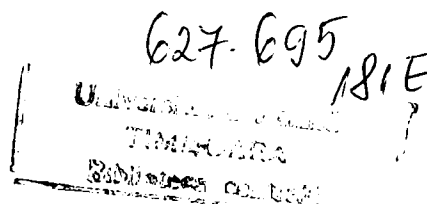
II.2.4 Curbele B-spline

Două neajunsuri majore ale curbelor Bézier (efectul global al modificării poziției punctelor de control și relația dintre gradul curbei și numărul de puncte de control) pot fi înlăturate prin folosirea curbelor B-spline [89], [96], [98], [100].

Inițial, florarul (spline) era o unealtă a desenatorilor ce consta dintr-o placă metalică (sau de lemn) decupată în mod special și folosită pentru a desena curbe prin anumite puncte fixe. Echivalentul matematic este curba B-spline polinomială.

Ca și o curbă Bézier, o curbă B-spline nu trece prin punctele de control. Ea este polinomială pe bucăți, constând dintr-un număr oarecare de segmente de curbă. Pot exista curbe B-spline de orice grad formate din segmente definite pe anumite intervale, de la un interval la altul schimbându-se coeficienții. Pentru un singur segment se poate compara formularea B-spline cu cea Bézier folosind aceeași notație matriceală.

Formularea B-spline este:



$$Q_i(\mathbf{u}) = \mathbf{U} \cdot \mathbf{B}_s \cdot \mathbf{P}_i = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \cdot \frac{1}{6} \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{bmatrix} \quad (\text{II.2.12})$$

unde $Q_i(\mathbf{u})$ este segmentul i al curbei, iar \mathbf{P}_i este o matrice ce conține un set de patru puncte dintr-o succesiune de puncte de control. Același lucru se poate scrie și sub forma:

$$Q_i(u) = \sum_{k=0}^3 P_{i-3+k} \cdot B_{i-3+k}(u) \quad (\text{II.2.13})$$

unde i este numărul segmentului, iar k este indicele punctului de control local. Valoarea lui u într-un segment de curbă este $0 \leq u \leq 1$. El este un parametru local ce definește un singur segment de curbă B-spline.

Astfel, folosind această notație, se observă că o curbă B-spline este formată dintr-o serie de $m-12$ segmente de curbă notate convențional cu Q_3, Q_4, \dots, Q_m , definite prin $m+1$ puncte de control $P_0, P_1, \dots, P_m, m \geq 3$. Fiecare segment de curbă este definit prin patru puncte de control și fiecare punct de control influențează patru și numai patru segmente de curbă. Aceasta este proprietatea controlului local al curbelor B-spline, constituind și principalul avantaj asupra curbelor Bézier.

Un singur segment de curbă Bézier este supus unui control global deoarece mutând un punct de control se afectează întreaga curbă. Într-o curbă compusă B-spline, mutarea unui punct de control afectează numai câteva segmente de curbă. În mod corect, comparația ar trebui făcută între curbe Bézier multisegment și curbe B-spline. Diferența este că, pentru a menține continuitatea segmentelor Bézier, deplasarea punctelor de control trebuie să satisfacă restricțiile impuse, pe când punctele de control ale unei curbe B-spline pot fi mutate în orice fel.

O curbă B-spline prezintă continuitate pozițională și continuitate a derivatelor de ordinul întâi și doi, deoarece funcțiile de bază sunt ele însele polinoame de clasă C^2 . O combinație liniară de astfel de funcții de bază va fi de asemenea continuă C^2 . Se definește întregul set de segmente ca o curbă B-spline în u :

$$Q(u) = \sum_{i=0}^m P_i \cdot B_i(u) \quad (\text{II.2.14})$$

unde i reprezintă indicii punctelor de control care nu sunt locale, iar u este, în acest caz, un parametru global.

II.2.5 Curbele B-spline uniforme

Relația (II.2.13) arată că fiecare segment dintr-o curbă B-spline este definit prin patru funcții de bază și patru muchii de control. Din acest motiv există trei funcții de bază și trei muchii de control în plus față de numărul de segmente. Valoarea lui u în punctul de unire a două segmente se numește **valoare de nod**, iar faptul că o curbă B-spline este uniformă înseamnă că nodurile sunt plasate la intervale egale ale parametrului u .

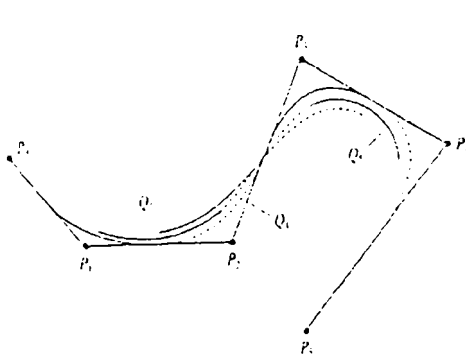


Fig.II.7 Curbă B-spline formată din trei segmente (definită prin șase puncte de control)

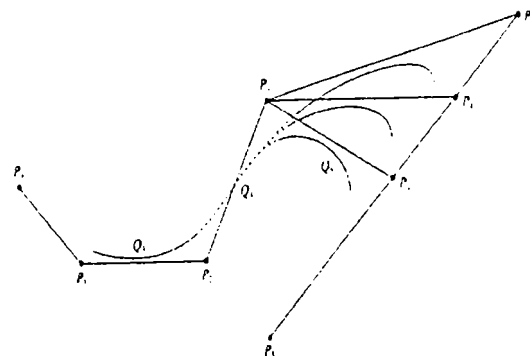


Fig.II.8 Capacitatea de localizare a curbelor B-spline

Figura II.7 prezintă o curbă B-spline definită prin poziția a șase muchii, respectiv puncte de control: P_0, P_1, \dots, P_5 . Curbele trasate sunt de grad 2, 3 și 4 remarcându-se curba de gradul 3, formată din trei segmente cu capătul din stânga al lui Q_3 lângă P_0 și capătul din dreapta al lui Q_5 lângă P_5 . Se observă că, spre deosebire de o curbă Bézier, o curbă B-spline uniformă nu interpolează în general punctele de control de la capete.

Faptul că segmentele de curbă folosesc puncte de control în comun, stă la baza mecanismului prin care se menține continuitatea C^2 între segmente. Figura II.8 prezintă efectul schimbării poziției punctului de control P_j . Deplasarea respectivă trage segmentul Q_5 în direcția corespunzătoare și afectează, într-o mai mică măsură, segmentul Q_4 (care este și el definit prin P_j). Totuși Q_3 nu este afectat, figura demonstrând importanța proprietății de efect local a curbelor B-spline. În general, un singur punct de control influențează patru segmente de curbă.

Fiecare funcție de bază ce definește curba este nenulă pe patru intervale succesive în care variază u (figura II.9). Ea este, de fapt, o cubică formată la rândul ei din patru segmente. Funcția B-spline este nenulă între valorile $u_i, u_{i+1}, \dots, u_{i+4}$ și centrată în u_{i+2} . Fiecare punct de control este trasat de o singură funcție de bază, și, dacă se presupune că valorile de nod sunt egal distanțate, fiecare funcție de bază este o copie sau o translație a celeilalte. Setul de funcții de bază folosit de curba din figura II.7 este prezentat în figura II.10.

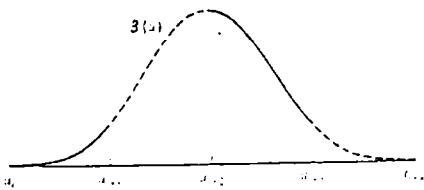


Fig.II.9 Funcția de bază B-spline uniformă

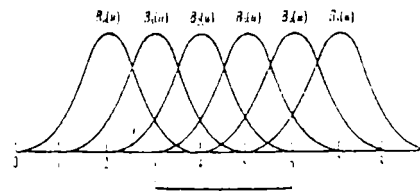


Fig.II.10 Cele 6 funcții B-spline utilizate în fig. 7

Dacă se studiază un singur segment al curbei, acesta definește intervalul $[u_i, u_{i+1}]$ în care variază parametrul u . Funcțiile de bază ce sunt active în intervalul parametric i , de la u_i la u_{i+1} , adică cele ce definesc un singur segment de curbă, sunt prezentate îngroșat în figura II.11.

Aceasta reprezintă o interpretare utilă a comportamentului funcțiilor când u variază. În general, pentru valori ale lui u ce nu sunt valori de nod, există patru funcții de bază active ce au suma egală cu 1. Când este atinsă o valoare de nod $u = u_i$, una din funcții este dezactivată și o alta devine activă. Într-un nod există doar trei funcții a căror sumă este unitară.

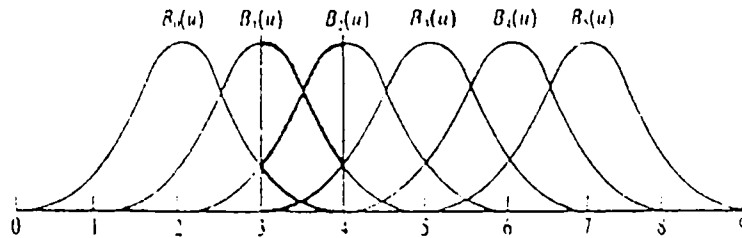


Fig.II.11 Cele patru funcții B-spline nenule pentru primul segment de curbă din fig.II.7

Se poate astfel afirma că o curbă B-spline este compusă din $m-2$ segmente definite prin poziția a $m+1$ funcții de bază peste $m+5$ valori de nod. Astfel, curba din figura II.7 este formată din trei segmente, șase puncte de control și șase funcții de bază definite peste zece valori de nod.

În general, o curbă B-spline nu interpolează nici un punct de control. Ea poate fi obligată să interpoleze puncte de control introducând muchii multiple, aceasta implicând însă o pierdere de continuitate. Influența unui punct de control ar putea fi mărită prin repetarea sa, curba fiind atrasă către acest punct.

Un segment este creat de funcțiile de bază prin trasarea punctelor de control. Dacă un punct de control se repetă, el va fi folosit de mai multe ori pentru evaluarea unui singur segment. O astfel de tehnică poate fi folosită pentru a face curba să interpoleze atât punctele de control intermediare cât și pe cele din capete - unde pierderea continuității ar putea să fie mai puțin semnificativă. De exemplu, comparând figura II.12 cu figura II.7, ultimul punct de control din figura II.7 este repetat de trei ori. Există astfel cinci segmente, iar P_5 este folosit o dată în determinarea lui Q_5 , de două ori în Q_6 și de trei ori în Q_7 . Curba este acum definită pe intervalul $3 \leq u \leq 8$. În punctul $u=8$ curba coincide cu P_5 .

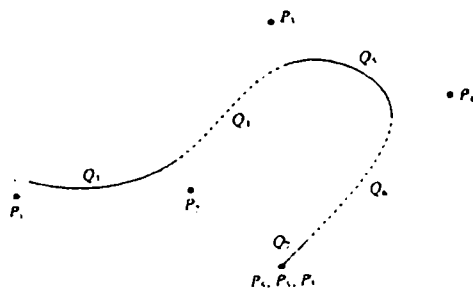


Fig.II.12 Efectul capetelor multiple. $P_{11.5}$ este triplat

Figura II.13,a prezintă efectul introducerii de puncte de control intermediare multiple. În această figură P_3 a fost dublat, el fiind aproape interpolat, apărând un segment în plus. Continuitatea se schimbă de la C^2G^2 la C^2G^1 . Fig.II.13,b îl prezintă pe P_3 triplat. De această dată curba interpolează punctul de control, devenind o linie dreaptă de fiecare parte a punctului, iar continuitatea se reduce la C^2G^0 .

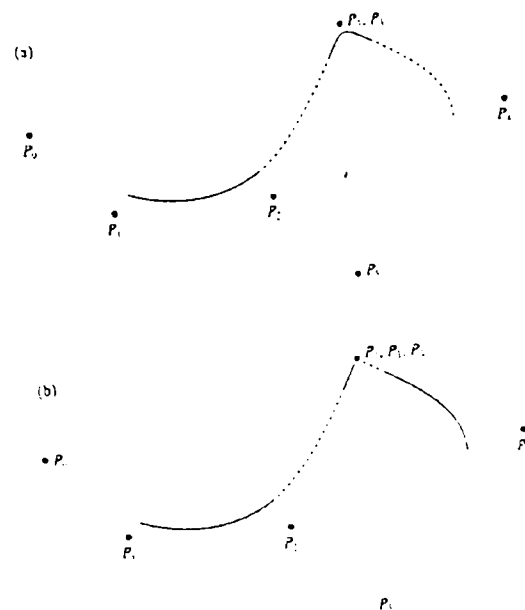


Fig.II.13 Efectul punctelor de control intermediare multiple

a. P_3 este dublat b. P_3 este triplat

II.2.6 Curbele B-spline neuniforme

O curbă B-spline neuniformă este aceea pentru care intervalele parametriche între valorile de nod succesive nu sunt în mod necesar egale. Aceasta implică faptul că funcțiile de combinare nu mai sunt obținute prin translație, ci variază de la interval la interval. Forma cea mai întâlnită a unei curbe B-spline neuniformă este aceea în care unele intervale dintre valori de nod succesive sunt reduse la zero, prin inserarea de noduri multiple. Această facilitate este folosită pentru a interpola puncte de control (capete și intermediare) și prezintă anumite avantaje față de metoda de introducere a punctelor de control multiple [95].

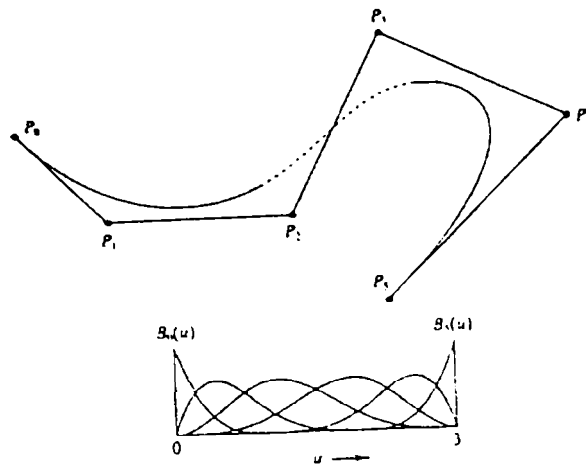


Fig.II.14 O curbă B-spline neuniformă ce interpolează punctele din capete folosind vectorul de nod $[0,0,0,0,1,2,3,3,3,3]$

Fie curba generată în figura II.7. Valorile de nod pentru această curbă sunt $u=3,4,5,6$. În domeniul parametric $3 \leq u \leq 6$ funcțiile de bază au suma de valoare unitară. Intervalul dintre fiecare valoare de nod este 1. Dacă se folosesc valori de nod neuniforme atunci funcțiile de bază nu mai sunt aceleași pentru fiecare interval parametric, ci variază odată cu u . În figura II.14 sunt folosite aceleași puncte de control ca și în figura II.7 și curba B-spline este încă formată din trei segmente. Totuși, această curbă interpolează capetele, deoarece au fost introduse noduri multiple la fiecare capăt al vectorului nod; acesta este $[0,0,0,0,1,2,3,3,3,3]$. Funcțiile de bază sunt și ele prezentate în figură, iar curba are acum nouă segmente, de la Q_0 la Q_8 . Totuși, Q_0 , Q_1 și Q_2 sunt reduse la un singur punct; Q_3 , Q_4 și Q_5 sunt definite pe domeniul $0 \leq u \leq 3$; Q_6 , Q_7 și Q_8 sunt de asemenea reduse la un singur punct $u=3$.

În general, un vector nod este orice secvență crescătoare de valori de nod, de la u_0 la u_{m+1} . Valorile de nod consecutive pot fi egale, iar numărul de valori identice se numește **multiplicitatea** nodului.

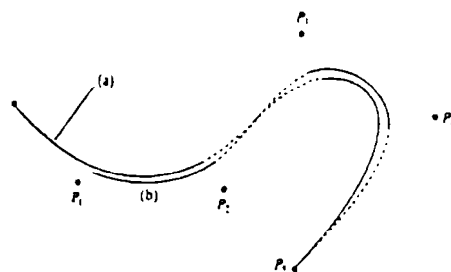


Fig.II.15 Influența interpolării multiple a capetelor

Obligând o curbă să interpoleze capetele folosind muchii de control multiple nu se obține același efect ca și la folosirea nodurilor multiple, iar figura II.15 prezintă ultimul punct de control P_5 interpolat prin utilizarea sa ca punct de control multiplu și, respectiv, folosind un vector nod cu multiplicitatea 4 pentru valoarea de nod finală.

Dacă se folosește vectorul nod $[0,0,0,0,1,1,1,1]$ atunci se obține un segment de curbă ce interpolează pe P_0 și P_3 . În acest caz funcțiile de bază sunt cele Bézier, iar curba rezultantă este o curbă Bézier. Rezultă astfel, că o curbă Bézier este doar un caz particular de curbă B-spline neuniformă.

O metodă recursivă de generare a funcțiilor de bază sau de combinare pentru curbele B-spline neuniforme este cea cunoscută sub numele de *algoritmul Cox De Boor* capabilă să genereze curbe B-spline uniforme sau neuniforme de orice grad folosind o singură relație recursivă. Deoarece funcțiile nu mai provin prin translație una din alta, calculele sunt mai complicate [105], [106], [107], [114].

Pentru o cubică se poate defini formula recursivă în forma sa desfășurată. Extinzând notația la o curbă B-spline pentru a include gradul ca al doilea indice, se definesc funcțiile de bază pentru a evalua punctul de control P_i ca $B_{i,j}(u)$, iar relațiile de recurență pentru o curbă B-spline cubică sunt:

$$B_{i,1}(u) = \begin{cases} 1 & u_i \leq u \leq u_{i+1} \\ 0 & \text{in rest} \end{cases} \quad (II.2.15)$$

$$B_{i,j}(u) = \frac{u - u_i}{u_{i+j-1} - u_i} B_{i,j-1}(u) + \frac{u_{i+j} - u}{u_{i+j} - u_{i+j-1}} B_{i+j-1,j-1}(u) \quad j = 2,3,4$$

Când nodurile sunt repetate, poate apărea în definiția *Cox De Boor* un coeficient nedeterminat $0/0$ și pentru acest caz se impune valoarea 0. Din punct de vedere al calculului, numărătorul este întotdeauna verificat dacă este zero, iar rezultatul este anulat, indiferent de valoarea numitorului. Alegerea unei mulțimi particulare de noduri în sistemele CAD comerciale ce folosesc curbele B-spline este de obicei o parte predefinită a sistemului respectiv.

II.2.7 Proprietăți ale curbelor B-spline

Unele dintre proprietățile curbelor Bézier se aplică și în cazul curbelor B-spline. Astfel se pot menționa:

- Curba urmărește forma poligonului de control și este obligată să respecte înfășurătoarea convexă a punctelor de control
- Curba prezintă proprietatea diminuării variației
- Curba este transformată aplicând orice transformare afină punctelor sale de control.

În plus, ele au o proprietate specifică:

- O curbă B-spline este controlată local, datorită faptului că un punct de control este legat de patru segmente (în cazul unei cubice) iar prin modificarea poziției sale sunt influențate doar aceste segmente.

II.3. Editarea curbelor parametrice

Utilitatea curbelor Bézier și B-spline a fost demonstrată de nenumărate aplicații cu finalitate în CAD, existând mai multe software-uri profesionale integrate în astfel de sisteme. În teză este prezentat un program de editare a curbelor Bézier și B-spline prin adăugare, mutare și ștergere de puncte de control [119], [126], [134], [138], [206], [207].

El este conceput cu pronunțat caracter interactiv afișând meniuri care permit utilizatorului editarea tipului de curbă selectat pe baza principiilor teoretice prezentate anterior. Astfel, pentru editarea unei curbe B-spline se activează meniul **Curbe** selectându-se tipul **B-spline**. Se selectează în continuare **Creare curbă nouă**, operație în cadrul căreia se marchează cu mouse-ul punctele de control. Operația **Desenare curbă**, va trasa curba de tipul ales prin utilizarea punctelor de control așa cum s-a prezentat în capitolul teoretic respectiv. În continuare pot fi activate opțiunile **Modificare curbă**, **Adăugare puncte de control** și **Stergere puncte de control**, cu rezultatele previzibile pe baza considerațiilor teoretice anterioare. Exemplificarea acestor operații posibile este prezentată în figurile II.16, II.17, II.18 pentru categoria de curbe B-spline. În figura II.19 este prezentată o curbă B-spline complexă realizată prin considerarea unui număr de 81 de puncte de control.

Meniul **Zoom** permite micșorarea de până la patru ori a curbei și respectiv, mărirea curbei două ori.

În același mod, cu aceleași operații se poate edita o curbă Bézier. Exemple sunt prezentate în figurile II.20 și II.21. Singura diferență la editarea curbei în acest caz este selectarea în meniul **Curbe** a opțiunii **Bézier**. Restul instrucțiunilor funcționează în același mod ca și la curbele B-spline.

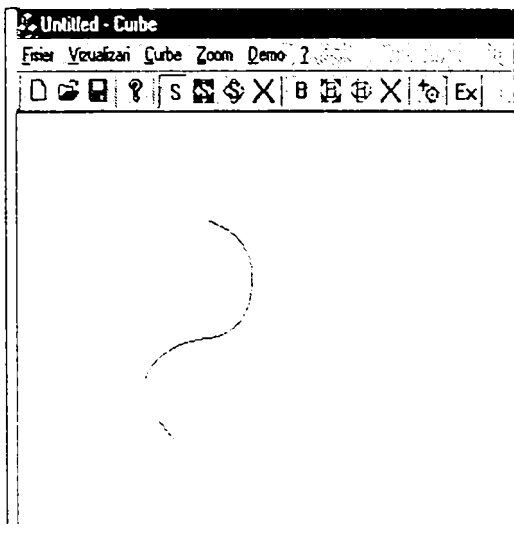


Fig.II.16 Desenare curbă B-spline cu 6 puncte de control

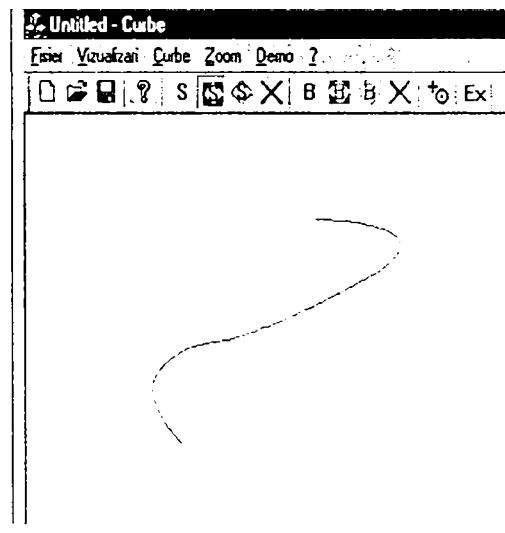


Fig.II.17 Modificarea curbei prin schimbarea poziției unui punct de control

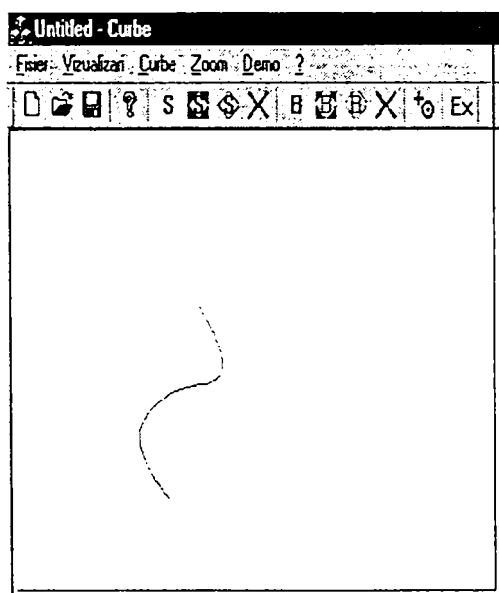


Fig.II.18 Modificarea curbei prin suprimarea unui punct de control

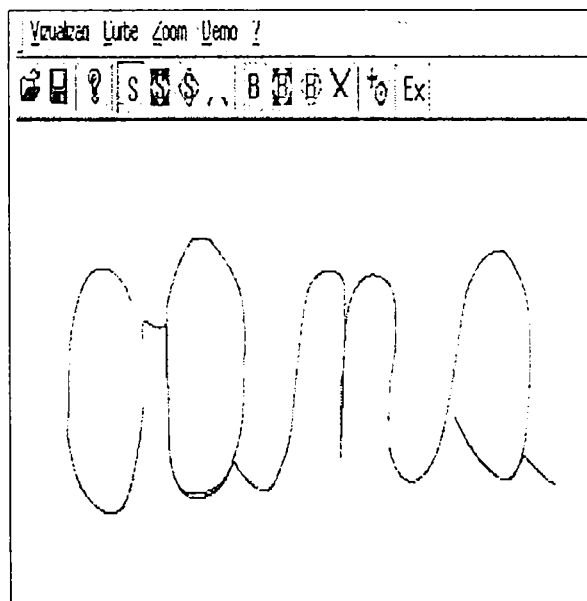


Fig.II.19 Curbă B-spline cu 81 de puncte de control

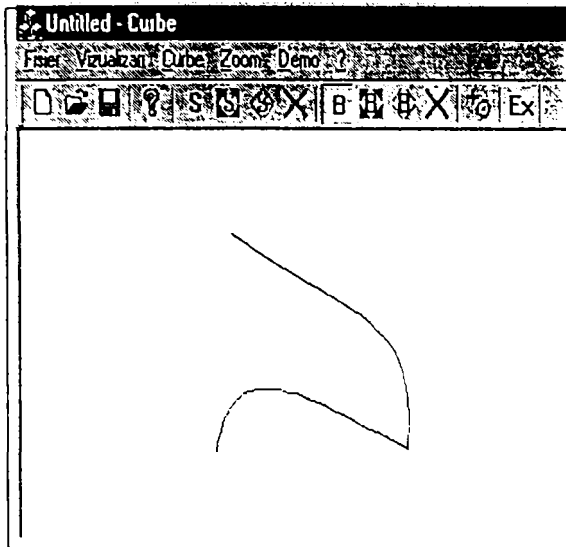


Fig.II.20 Curbă Bézier cu 7
puncte de control

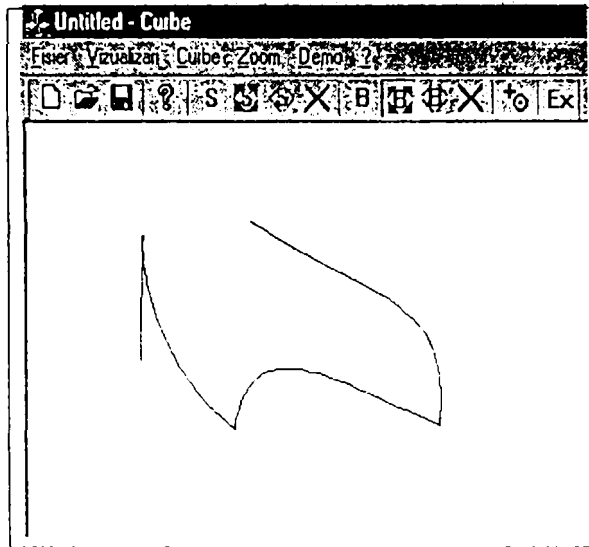


Fig.II.21 Modificarea curbei Bézier
prin adăugarea de 2 noi
puncte de control

Principalele funcții utilizate pentru editarea de curbilor sunt:

```
// CurbeView.cpp : implementation of the CCurbeView class
// Sunt enumerate fișierele ce sunt folosite de către program
#include "stdafx.h"
#include "Curbe.h"
#include "Math.h"
#include "CurbeDoc.h"
#include "CurbeView.h"
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

// CCurbeView drawing

void CCurbeView::OnDraw(CDC* pDC)
{
    // La mutarea ferestrei de lucru, se apelează funcția OnDraw, care păstrează
    // conținutul ferestrei neșters.
}

// CCurbeView message handlers

void CCurbeView::OnCurbeSplineCreare()
{
    // TODO: Add your command handler code here
    // Setează modul de lucru pe Creare curbă B-spline și golește șirul de
    // puncte de control.
}
```

```

}

void CCurbeView::OnUpdateCurbeSplineCreare(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
    // Condițiile în care este selectat butonul de creare a unei curbe B-spline
}

void CCurbeView::OnCurbeBezierCreare()
{
    // Setează modul de lucru pe Creare curbă Bézier și golește șirul de puncte
    de control.
}

void CCurbeView::OnUpdateCurbeBezierCreare(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
    // Condițiile în care este selectat butonul de creare a unei curbe Bézier
}

void CCurbeView::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    // Apăsarea butonului stânga al mouse-ului are efectele următoare:
    - dacă este activat modul de creare a unei curbe, pe ecran este desenat un punct de
    control, care este adăugat la șirul de puncte corespunzător
    - dacă este activat modul de modificare a unei curbe, se verifică dacă am nimerit
    un punct de control al curbei și dacă da acesta se memorează
    - dacă este activat modul de adăugare de puncte la curbă, se desenează noul punct
    de control și se trage o linie de la ultimul punct de control la acesta
    - dacă modul de lucru este de ștergere de puncte de control, se verifică dacă s-a
    nimerit un punct de control, și în caz afirmativ acesta este șters de pe ecran și din
    șirul corespunzător
}

void CCurbeView::DeseneazaPuncteSpline()
{
    // Realizează afișarea punctelor de control și a muchiilor de control
}

void CCurbeView::DeseneazaPuncteBezier()
{
    // Realizează afișarea punctelor de control și a muchiilor de control
}

void CCurbeView::StergEcran()
{
    // Șterge ecranul.
}

void CCurbeView::OnCurbeSplineExecutie()
{

```

```

    // TODO: Add your command handler code here
    // Șterge ecranul, desenează punctele de control și curba B-spline
}

void CCurbeView::OnUpdateCurbeSplineExecutie(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
    // Pune condițiile de activare a butonului de desenare a curbei B-spline
}

void CCurbeView::OnCurbeBezierDesenare()
{
    // TODO: Add your command handler code here
    // Șterge ecranul, desenează punctele de control și curba B-spline
}

void CCurbeView::OnUpdateCurbeBezierDesenare(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
    // Pune condițiile de activare a butonului de desenare a curbei B-spline
}

void CCurbeView::OnCurbeSplineModificare()
{
    // TODO: Add your command handler code here
    // Schimbă modul de lucru în modificare a curbelor B-spline și redesenează
    curba
}

void CCurbeView::OnUpdateCurbeSplineModificare(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
    // Pune condițiile de activare și apăsare a butonului de desenare a curbei B-
    spline
}

void CCurbeView::OnCurbeBezierModificare()
{
    // TODO: Add your command handler code here
    // Schimbă modul de lucru în modificare a curbelor B-spline și redesenează
    curba }

void CCurbeView::OnUpdateCurbeBezierModificare(CCmdUI* pCmdUI)
{
    // TODO: Add your command update UI handler code here
    // Pune condițiile de activare și apăsare a butonului de desenare a curbei
    Bézier
}

void CCurbeView::OnMouseMove(UINT nFlags, CPoint point)

```

```

{
    // TODO: Add your message handler code here and/or call default
    // În cazul în care la apăsarea butonului stânga al mouse-ului a fost selectat
    un punct al curbei, la mișcarea cursorului pe ecran, dacă este activat modul de
    modificare a curbei, acest punct este mutat și el
}

void CCurbeView::OnLButtonUp(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    // La eliberarea butonului stânga al mouse-ului este desenată curba
    corespunzătoare
}

int CCurbeView::NimeritPunctSpline(CPoint point)
{
    // Se verifică dacă a fost nimerit un punct al curbei B-spline sau nu
}

int CCurbeView::NimeritPunctBezier(CPoint point)
{
    // Se verifică dacă a fost nimerit un punct al curbei Bézier sau nu
}

void CCurbeView::MutaPunctSpline(CPoint point)
{
    // Deplasează punctul de control selectat al curbei B-spline
}

void CCurbeView::MutaPunctBezier(CPoint point)
{
    // Deplasează punctul de control selectat al curbei Bézier
}

BOOL CCurbeView::Create(LPCTSTR lpszClassName, LPCTSTR
lpszWindowName, DWORD dwStyle, const RECT& rect, CWnd* pParentWnd,
UINT nID, CCreateContext* pContext)
{
    // TODO: Add your specialized code here and/or call the base
    class
    // Inițializează toate variabilele folosite în program
}

void CCurbeView::FisierNou()
{
    // Golește șirurile de puncte de control
}

void CCurbeView::OnCurbeSplineStergerepuncte()
{
    // TODO: Add your command handler code here
    // Trece în modul de ștergere a punctelor de control ale curbei B-spline
}

```

```

}

void CCurbeView::OnCurbeBezierStergerePuncte()
{
    // TODO: Add your command handler code here
    // Trece în modul de ștergere a punctelor de control ale curbei Bézier
}

void CCurbeView::OnCurbeAdaugarePuncte()
{
    // TODO: Add your command handler code here
    // Selectează modul de adăugare a punctelor de control la curbă
}

```

II.4. Suprafețe cubice biparametrice

II.4.1. Porțiuni de suprafață Bézier

Tratarea segmentelor de curbă cubică parametrică prezentată anterior poate fi generalizată la porțiunile de suprafață cubică biparametrică. Un punct de pe porțiunea de suprafață este dat printr-o funcție de doi parametri, iar pentru fiecare parametru este folosită o mulțime de funcții de bază. O suprafață cubică Bézier e definită astfel:

$$Q(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 P_{ij} \cdot B_i(u) \cdot B_j(v) \quad (\text{II.4.1})$$

Matematic, suprafețele tridimensionale sunt generate prin produsul cartezian a două curbe. O suprafață Bézier și punctele sale de control sunt prezentate în figura II.22, unde suprafețele sunt reprezentate prin linii izoparametrice.

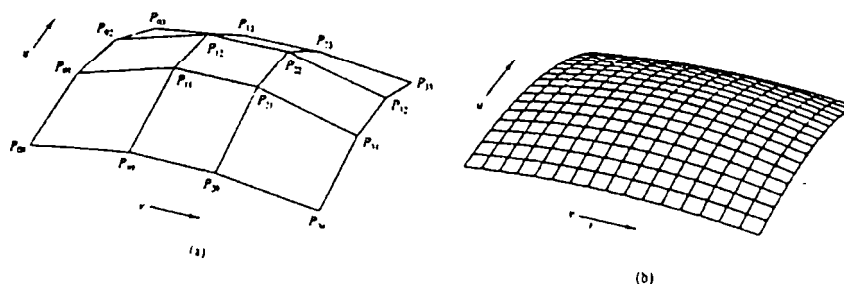


Fig.II.22 Realizarea unei suprafețe Bézier

a) Un poliedru de control b) Porțiunea de suprafață Bézier obținută

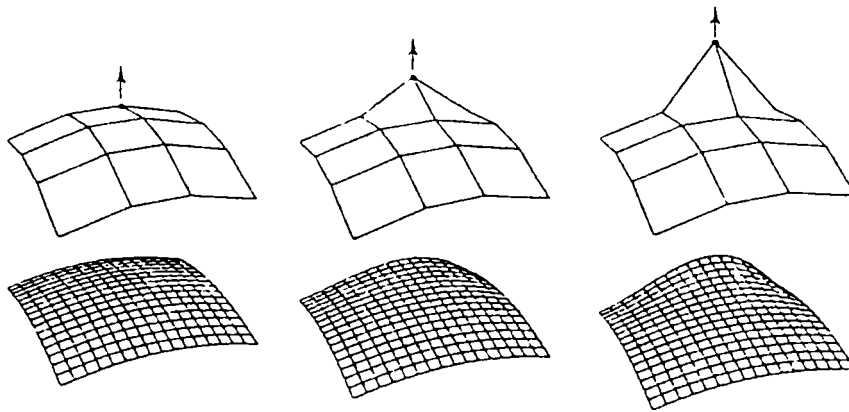


Fig.II.23 Efectul deplasării unui punct de control asupra suprafeței

Punctele de control au în cazul reprezentării unei suprafețe același rol ca în cazul reprezentării unei curbe, astfel că proprietățile curbelor Bézier se extind și în domeniul suprafețelor. Astfel, se observă din figura II.23 că o porțiune de suprafață poate fi deformată dacă se modifică poziția unui singur punct de control, printr-un mecanism identic cu cel de deformare a curbei prin schimbarea poziției punctelor de control.

Formularea matriceală pentru reprezentarea unei suprafețe este:

$$\mathbf{P}(u, v) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \cdot \mathbf{B} \cdot \mathbf{P} \cdot \mathbf{B}^T \cdot \begin{bmatrix} v^3 \\ v^2 \\ v \\ 1 \end{bmatrix} \quad (\text{II.4.2})$$

unde:

$$\mathbf{B} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} \quad (\text{II.4.3})$$

Este importantă examinarea relației dintre punctele de control și vectorii tangenți în vârfurile porțiunii de suprafață. De exemplu, în vârful $u = v = 0$, relația între punctele de control și vectorii asociați cu muchia P_{00} este:

$$\begin{aligned} Q_u(0,0) &= 3(P_{10} - P_{00}) \\ Q_v(0,0) &= 3(P_{01} - P_{00}) \\ Q_{uv}(0,0) &= 9(P_{00} - P_{01} - P_{10} + P_{11}) \end{aligned} \tag{II.4.4}$$

În figura II.24 se pot vedea acești vectori la colțul unei porțiuni de suprafață. $Q_u(0,0)$ are direcția vectorului tangent la muchia curbilinie $Q(0,0)$ pe direcția parametrului u și modulul modificat prin înmulțire cu o constantă. Analog, $Q_v(0,0)$ este un multiplu al vectorului tangent pe direcția parametrului v .

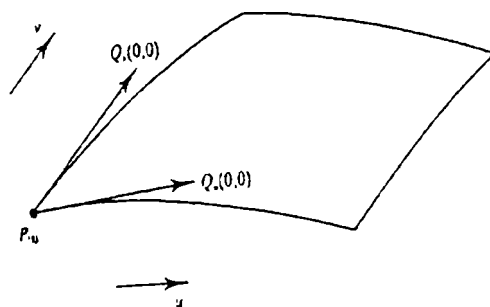


Fig.II.24 Vectorul tangent în P_{00}

Derivatele mixte în fiecare punct de control, definesc **vectorii de răsucire**, specificând viteza de variație a vectorilor tangenți în raport cu u și respectiv. v . Vectorul de răsucire este un vector normal la planul ce conține vectorii tangenți.

Avantajele folosirii descrierii unei suprafețe prin porțiuni parametrice nu constă în faptul că o coordonată precisă este disponibilă pentru fiecare punct al suprafeței, ci în faptul că face mult mai ușoară modelarea obiectelor [145].

II.4.2 Combinarea porțiunilor de suprafață Bézier

În majoritatea aplicațiilor este necesar ca din mai multe porțiuni de suprafață să se modeleze suprafața exterioară complexă a unui corp oarecare. Pentru aceasta

trebuie ca suprafața realizată să fie continuă, ceea ce în mod firesc impune anumite restricții.

Realizarea continuității de gradul întâi între două porțiuni de suprafață se obține printr-o generalizare a restricțiilor impuse la unirea segmentelor de curbă, abordarea cea mai simplă fiind cea geometrică.

Fie două porțiuni de suprafață Q și R , cu continuitate pozițională (de gradul zero) având o muchie comună. Condiția unei astfel de continuități este:

$$Q(1, v) = R(0, v) \quad 0 \leq v \leq 1 \quad (\text{II.4.5})$$

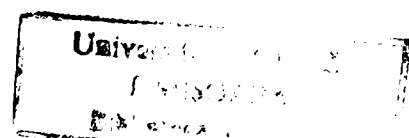
Această condiție implică existența unui poligon caracteristic care să conțină muchia comună a celor două porțiuni:

$$Q_{3i} = R_{0i} \quad i = 0, \dots, 3 \quad (\text{II.4.6})$$

În proiectarea asistată de calculator, această restricție este gravă atunci când o suprafață compusă este construită dintr-o mulțime de porțiuni Bézier. De exemplu, dacă o suprafață compusă este creată pornind de la o porțiune și adăugând altele în jurul acesteia, alăturarea a două porțiuni pe o muchie comună implică faptul că opt dintre punctele de control ale celei de-a doua porțiuni sunt deja fixate, iar alăturarea unei porțiuni la altele două deja existente presupune existența a douăsprezece puncte de control deja fixate [142], [147].

S-a găsit însă o altă condiție, mai puțin restrictivă prin care în colțuri, porțiunile pot avea o continuitate pozițională dar nu și continuitate a gradientului (figura II.25). Totuși, vectorii tangenți la muchiile ce se întâlnesc într-un colț trebuie să fie coplanari. Chiar și cu această flexibilitate marginală mult mai mare, apar încă probleme la designul suprafețelor compuse, o soluție fiind folosirea porțiunilor de suprafață de grad mai mare decât trei [167], [171], [174], [177].

Trebuie menționat și faptul că porțiunile de suprafață de forma dreptunghiurilor curbilini nu pot fi folosite în reprezentarea oricărei suprafețe. Pentru forme complexe de corpuri porțiunile dreptunghiulare trebuie să fie înlocuite cu triunghiuri, care pot fi obținute prin degenerare.



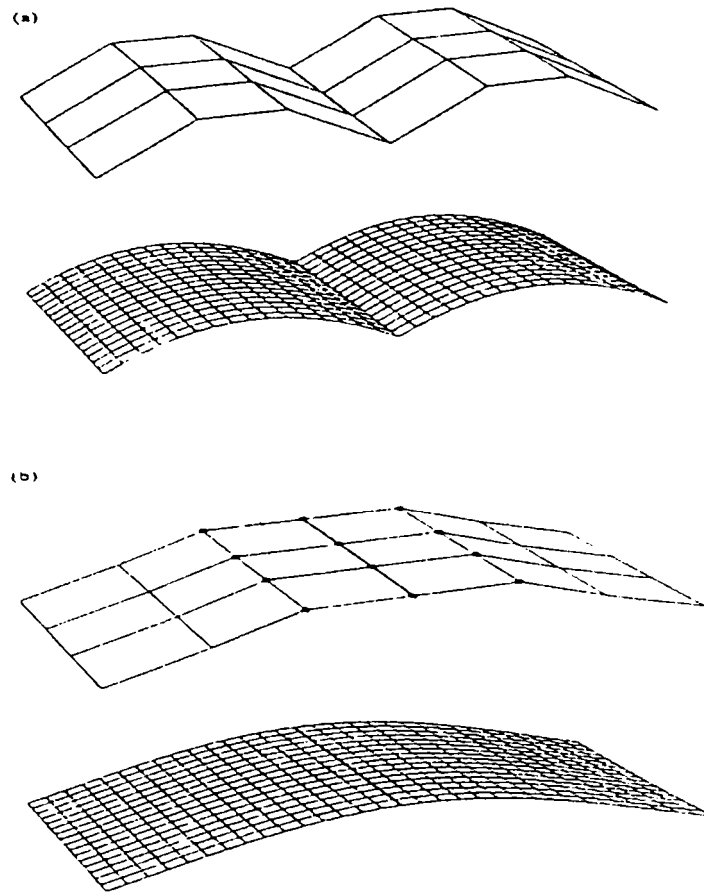


Fig.II.25 a) *Continuitate pozițională între porțiuni bicubice Bézier*
 b) *Continuitate tangențială între porțiuni bicubice Bézier*

Se consideră că principalul motiv pentru care porțiunile dreptunghiulare sunt predominante în sistemele CAD este faptul că prima lor aplicație a fost în designul automobilelor la definirea părților exterioare ale caroseriei. Aceste părți au o geometrie rectangulară și este natural să fie divizate în dreptunghiuri mai mici [178], [191].

II.4.3 Porțiuni de suprafață B-spline

Evaluarea unei porțiuni de suprafață B-spline bicubică se realizează prin relația:

$$Q(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{ij} \cdot B_{ij}(u, v) \quad (\text{II.4.7})$$

unde P_{ij} este un șir de puncte de control, iar $B_{i,j}(u,v)$ e o funcție de bază de două variabile. Aceasta poate fi generată prin:

$$B_{ij}(u, v) = B_i(u) \cdot B_j(v) \quad (\text{II.4.8})$$

unde $B_i(u)$ și $B_j(v)$ sunt funcțiile cubice B-spline anterior definite.

Astfel:

$$Q(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{ij} \cdot B_i(u) \cdot B_j(v) \quad (\text{II.4.9})$$

La fel ca și la curbele B-spline, se consideră că o porțiune de suprafață B-spline este formată din mai multe segmente dreptunghiulare. În spațiul parametrilor există două secvențe de noduri, funcții de u și respectiv, de v , care considerate împreună formează un tablou [150], [151].

Cel mai simplu mod de abordare consideră porțiuni de suprafață B-spline uniforme, la care grila valorilor de nod prezintă intervale egale pe direcțiile parametrilor u și v .

Fie un segment dintr-o porțiune de suprafață, noțiune ce descrie entitatea din spațiul biparametric analogă unui segment de curbă din spațiul monoparametric.

Astfel, o porțiune de suprafață este formată din mai multe segmente. În cazul unui segment de curbă B-spline era nevoie de patru puncte de control pentru a-l defini. Extinzând în spațiul biparametric, este necesar un tablou, P_{ij} , de 4x4 puncte de control pentru a forma un singur segment de porțiune. Aceste puncte de control sunt precizate prin 4x4 funcții de bază de două variabile. Astfel, un singur segment de porțiune necesită un tablou sau o matrice de noduri de 8x8 valori (figura II.26). Funcțiile de bază bivariate au, în fiecare nod marcat, valori bine precizate.

În figura II.27 se prezintă un segment de porțiune B-spline determinat prin șaisprezece puncte de control, segmentul fiind marcat în regiunea din apropierea celor patru puncte de control centrale. La fel ca și la curbele B-spline, care nu își interpolatează punctele de control, un segment de porțiune B-spline nu interpolatează nici

cele patru puncte de control interioare, nici vreunul din cele douăsprezece puncte exterioare.

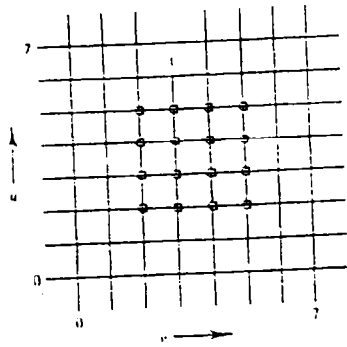


Fig.II.26 16 funcții B-spline bivariate precizează un segment

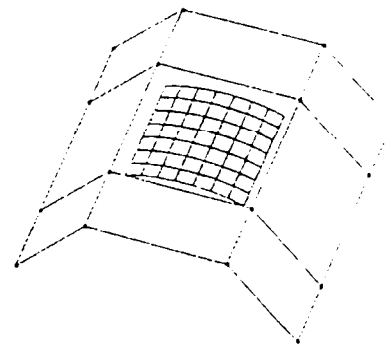


Fig.II.27 Un segment de porțiune B-spline

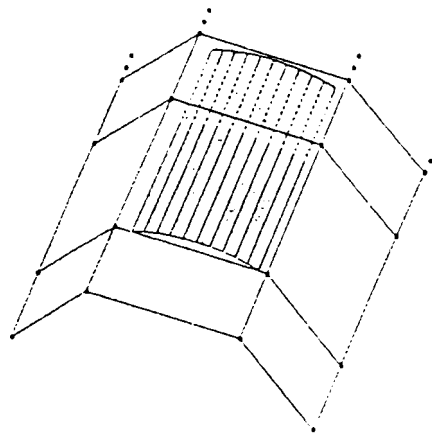


Fig.II.28 Modificarea porțiunilor de la marginile poliedrului de control

Se poate modifica comportarea porțiunilor la marginile poliedrului de control folosind muchii multiple (la fel cum s-au controlat curbele folosind puncte finale multiple). Aceasta se demonstrează cu ușurință prin exemplul din figura II.28 în care s-a triplat o parte din muchiile marginale, rezultând o matrice de control cu 24 de puncte. Astfel se formează o porțiune de suprafață din trei segmente care este atrasă către muchiile multiple. Se remarcă și faptul că nici una dintre muchiile din margine nu este interpolată.

În figura II.29 s-au triplat două seturi de muchii marginale, dintre care pe una din direcții există coliniaritate, obținându-se astfel o porțiune de suprafață cu nouă

segmente la care muchiile coliniare sunt interpolate. Dacă se triplează toate muchiile marginale rezultă o suprafață cu 25 de segmente, care va interpola doar punctele din colțuri.

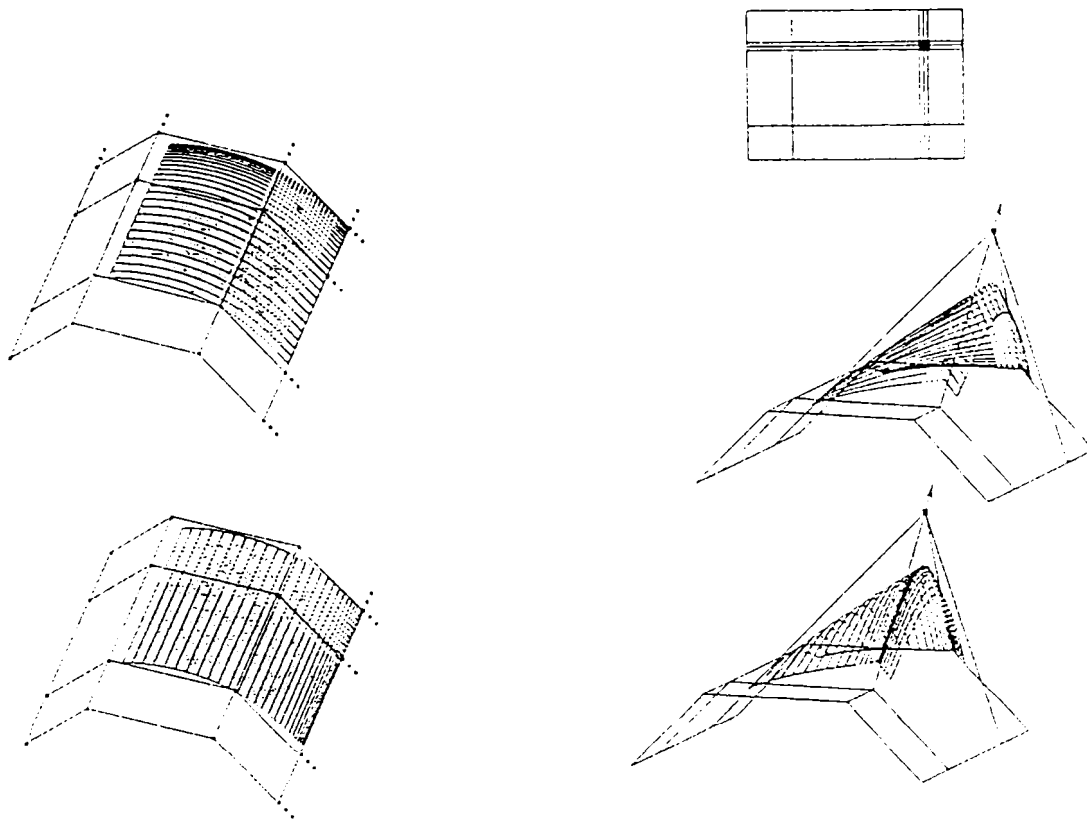


Fig.II.29 O porțiune de suprafață cu nouă segmente formată prin triplarea unui rând și a unei coloane de puncte de control

Fig.II.30 Triplarea unui punct interior de control

Dublarea sau triplarea punctelor de control interioare se folosește la producerea unor efecte de modelare mult mai puternice decât cele disponibile pentru o porțiune de suprafață Bézier. În figura II.30, s-a reprezentat noua formă a porțiunii de suprafață în care poliedrul de control este același ca și în exemplele anterioare, dar un punct interior a fost ridicat.

Si în acest caz, un rând și o coloană de puncte de control au fost triplate. În cazul coloanei, trei dintre muchiile de control sunt coliniare, ceea ce face ca suprafața să aibă o îndoitură de-a lungul muchiei corespunzătoare din poliedrul de control. La rândul triplat punctele nu sunt coliniare, iar efectul de îndoire este mai puțin precis.

În mod evident, aceste efecte pot fi cuprinse într-un program de modelare în care rândurile și coloanele sunt triplate și toate marginile poliedrului de control sunt mutate interactiv pentru a crea sau a muta o cută a suprafeței [159].

II.4.4. Editarea suprafețelor parametric

Algoritmii ce editează suprafețe reprezentate prin porțiuni parametric bicubice se împart în două categorii:

- algoritmi ce redau direct suprafața utilizând relațiile matematice de descriere parametrică
- algoritmi ce aproximează suprafața prin pași poligonali folosind poligoane plane pentru a reda această aproximație

În prezent, a doua variantă este mai des folosită, fiind cea mai ușor de implementat și mai puțin costisitoare din punct de vedere al calculelor.

Editarea directă a suprafețelor definite parametric este dificilă, deoarece proprietățile pasului poligonal, folosit la reprezentarea prin cea de-a doua metodă, nu sunt disponibile la porțiunile definite parametric [160], [161], [163]. Aceste proprietăți sunt:

- coordonatele maxime și minime pe direcția Oy se obțin ușor din lista vârfurilor ce conține mulțimile de coordonate (x,y,z) pentru fiecare vârf;
- pentru a scrie fiecare latură a poligonului ca o funcție de y pot fi folosite relații de incrementare;
- tot relații de incrementare pot fi folosite și pentru a calcula adâncimea reprezentării z ca o funcție de x .

O suprafață definită parametric nu posedă nici una din aceste proprietăți. Coordonatele y maxime și minime nu vor fi, în general, situate pe marginile suprafeței, iar o porțiune de suprafață va prezenta adesea o margine obținută prin scheletizare. O margine scheletizată e definită de acele puncte de pe suprafață în care normala la suprafață are componentă nulă pe direcția Oz . Marginile scheletizate nu sunt neapărat marginile reale ale porțiunii de suprafață, ci se obțin prin tehnici specifice, putând chiar acoperi o margine reală [153], [156], [157].

La suprafețele definite parametric trebuie trasate atât marginile reale cât și cele scheletizate apărând complicații atunci când cele două categorii de margini, definite diferit, se intersectează.

În general, o suprafață definită parametric, sau o porțiune de suprafață, este specificată prin trei funcții de câte două variabile:

$$\begin{aligned} x &= X(u, v) \\ y &= Y(u, v) \\ z &= Z(u, v) \end{aligned} \tag{II.4.10}$$

unde atât u cât și v variază între 0 și 1. Limitele unei porțiuni de suprafață sunt astfel definite de valorile $u=0$, $u=1$, $v=0$ și $v=1$.

În cazul porțiunilor de suprafață parametrică bicubice se poate considera un algoritm ce operează cu segmente de curbe formate prin intersectarea planului xOz cu suprafața. În general, un segment de curbă se află între două puncte ce o limitează. La poligoanele plane această curbă este o dreaptă, fiind suficientă stocarea punctelor de capăt. În cazul suprafețelor parametrică oarecare, trebuie însă determinate toate, sau aproape toate, punctele de pe curba de intersecție.

În figurile II.31 și II.32 sunt prezentate suprafețe obținute prin utilizarea curbelor parametrică a căror reprezentare este descrisă prin funcțiile prezentate în capitolul II.3 completate cu:

```
void CCurbeView::Demo()
{
    // Definește punctele de control pentru segmentele de curbă ce formează sfera
    // și apelează funcția de desenare a acestor segmente și a cilindrului în care este
    // înscrisă
}

void CCurbeView::PicturaDemo(int par, int k)
{
    // Desenează segmentele de curbă ce formează sfera și cilindrul în care este
    // înscrisă
}
```

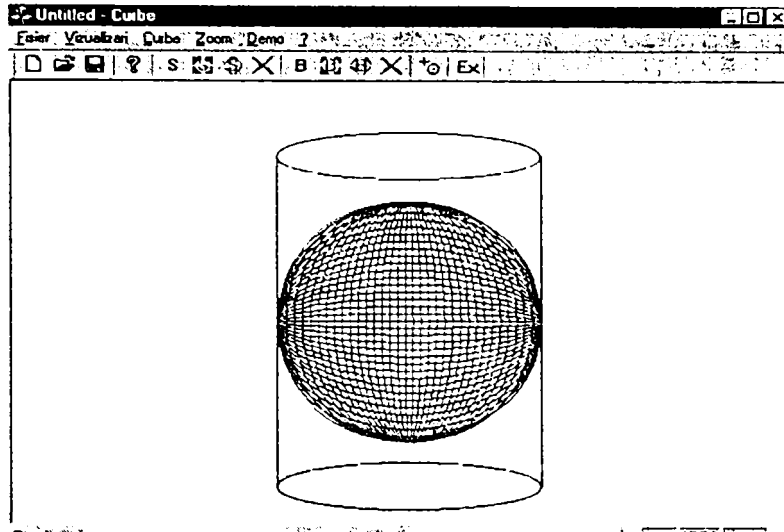


Fig.II.31.Reprezentarea a două suprafețe prin curbe Bézier

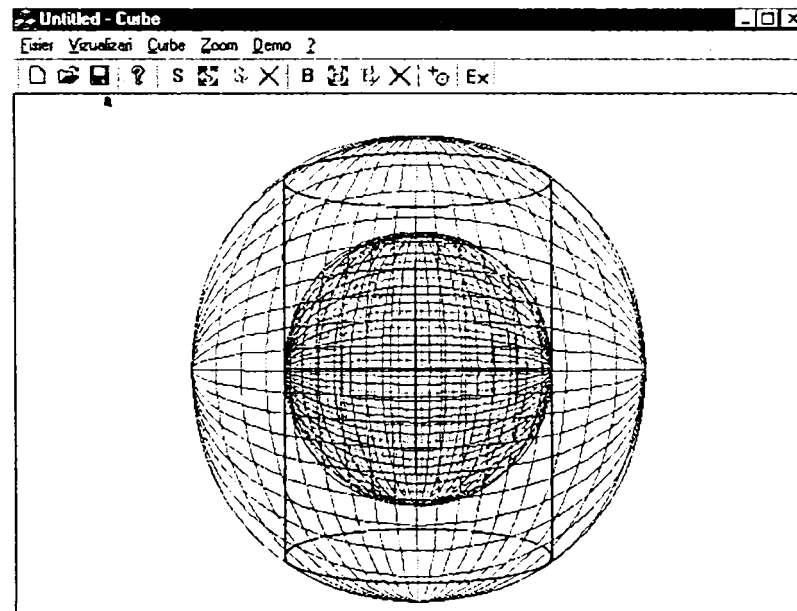


Fig.II.31.Reprezentarea a trei suprafețe prin curbe Bézier

II.5. Reprezentarea parametrică a spațiilor de lucru proprii ale roboților prin utilizarea Matlab

II.5.1. Considerații generale

Una din cele mai importante aplicații ale reprezentării parametrice a sistemelor este modelarea spațiilor descrise de roboți în timpul efectuării sarcinii. Deoarece problema care se pune este în esență geometrică, abordarea modelării acestor

suprafețe 3D se bazează pe considerarea modelului geometric (cinematic) al roboților, scrierea ecuațiilor cinematice fiind o operație fundamentală în dezvoltarea oricărei modelări ulterioare [27], [38], [39], [73], [86], [179], [180], [181], [182] .

S-au abordat mai multe tipuri de roboți, pentru care prin aplicarea convenției Denavit-Hartenberg s-a stabilit modelul geometric și s-au scris matricele de transfer, prin a căror înmulțire rezultă matricea ce exprimă poziția și orientarea efectorului final. Elementele matricei de transfer reprezintă cele 12 ecuații cinematice ale robotului care sunt funcții de variabilele articulare. Acestea la rândul lor fiind funcții de timp, rezultă că ecuațiile cinematice ale robotului sunt ecuațiile parametrice ce exprimă poziția și orientarea efectorului final. Reprezentarea lor în spațiul 3D, conduce la modelarea suprafețelor în interiorul cărora se află în permanență efectorul robotului în timpul executării sarcinii sale de lucru.

Pentru aceste reprezentări parametrice s-a utilizat pachetul de programe *Matlab* (Matrix LABoratory), dedicat calculului numeric și reprezentărilor grafice în domeniul științific. Performanțele deosebite și timpul de instruire redus fac din acest pachet de programe unul din cele mai agreate medii de lucru pentru o mare diversitate de utilizatori [18], [123], [131], [135].

El integrează analiza numerică, calculul matriceal, procesarea semnalelor și reprezentările grafice, într-un mediu ușor de învățat și folosit, în care enunțurile problemelor și rezolvările acestora sunt exprimate în modul cel mai natural posibil, așa cum sunt scrise matematic, fără a fi necesară programarea tradițională [29].

Elementul de bază cu care operează Matlab-ul este matricea. Cu el se pot aborda și rezolva probleme fără a fi necesară scrierea unui program într-un limbaj specific de programare.

Cea mai importantă caracteristică a Matlab-ului este ușurința cu care poate fi extins. De asemenea Matlab-ul include aplicații specifice, numite TOOLBOX-uri. Acestea sunt colecții extinse de funcții Matlab (fișiere cu extensia M) care dezvoltă mediul de programare de la o versiune la alta, pentru a rezolva probleme din domenii variate. Structural, Matlab-ul este realizat sub forma unui nucleu de bază, cu interpretor propriu, în jurul căruia sunt construite TOOLBOX-urile. Până la această versiune există foarte multe TOOLBOX-uri puse în circulație de firma The MathWorks Inc.

Versiunea Matlab utilizată în această lucrare este MATLAB 4.2 sub WINDOWS.

MATLAB 4.2 rulează sub Windows într-o prezentare nouă, orientată spre ferestre și meniuri. Această nouă versiune, care face posibilă vizualizarea mai multor ferestre grafice în același timp, adaptată stilului de operare sub sistemul Windows, menține compatibilitatea cu programe scrise în versiuni anterioare. Față de versiunea Dos, versiunile sub Windows au un număr semnificativ de funcții noi, atât pentru calcul, cât mai ales pentru reprezentările grafice. Acestea sunt atât de mult perfecționate, încât au dezvoltat mediul de programare Matlab până la posibilitatea de a-l utiliza la procesarea semnalelor bidimensionale și a imaginilor. Partea de documentare, inclusă în structura Help a produsului este aliniată sistemului Windows. Programul se lansează în execuție din mediul Windows, prin selecția pictogramei Matlab.

Matlab-ul lucrează cu două tipuri de ferestre: o fereastră de comenzi și o fereastră pentru reprezentări grafice.

Fereastra de comenzi este prezentată în figura II.32. Fiecare comandă din meniul principal furnizează un meniu specific, în cadrul acestuia selecția putându-se face fie cu mouse-ul fie cu ajutorul săgeților prin, deplasarea în zone active.

Selectând comanda **F**ile din meniul principal se poate deschide o fereastră de editare sau o fereastră grafică nouă selectând **N**ew sau se poate deschide o fereastră dialog pentru a selecta un fișier care va deveni fișier de lucru selectând **O**pen **M**-file.

Look **F**or **S**electe și **O**pen **S**electe analizează fișierele pentru selectare și deschide pe cel selectat. **S**ave **W**orkspace **A**s... deschide o fereastră de dialog pentru a salva datele din spațiul de lucru într-un fișier. **P**rint și **P**rinter **S**etup sunt comenzi de tipărire a documentului.

Exit **M**atlab are ca efect părăsirea aplicației.

Pot exista mai multe ferestre grafice deschise în același timp, dar numai o singură fereastră de comenzi.

Selectarea comenzii **F**ile urmată de comenzile **N**ew și **F**igure sau numai de **N**ew **F**igure, determină trecere într-o fereastră grafică nouă.

Selectând comanda **E**dit din meniul principal al ferestrei grafice, apoi selectând comanda **C**opy **M**etafile sau **C**opy **B**itmap se copiază figura curentă în format *meta* respectiv *bitmap* în Clipboard.

În fereastra grafică pot fi reprezentate funcții 2D sau 3D, imagini, animație etc.

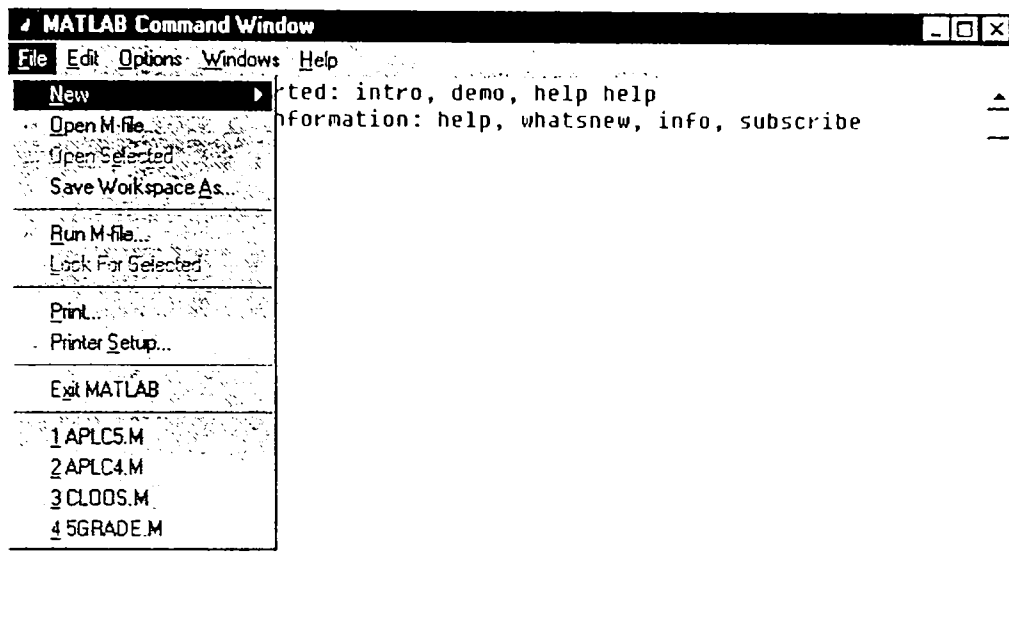


Fig.II.32 Fereastra de comenzi Matlab

Matlab-ul lucrează fie în modul linie de comandă, fie cu programe conținute în fișiere. Aceste două moduri formează împreună un mediu de programare. Fișierele ce conțin instrucțiuni Matlab se numesc fișiere-M și sunt programe Matlab. Un fișier-M constă dintr-o succesiune de instrucțiuni Matlab, cu posibilitatea apelării altor fișiere-M și a apelării recursive. Un fișier Script este un fișier extern care conține o secvență de comenzi Matlab care se execută prin apelarea numelui. Fișierele Script sunt folosite pentru rezolvarea unor probleme care cer comenzi succesive atât de lungi, încât ar putea deveni greoaie pentru lucrul în mod interactiv, adică în modul linie de comandă.

II.5.2. Funcții utilizate în programele Matlab scrise pentru generarea reprezentărilor grafice

Inițial se declară variabilele folosite în program. În cazul abordat, aceste variabile sunt variabilele articulare q_i ale roboților, adică unghiul θ_i la cuplele de rotație și deplasările d_i la cuplele de translație.

Matlab-ul este un limbaj de expresii. Expresiile tipărite de utilizator sunt interpretate și evaluate. Instrucțiunile Matlab sunt de cele mai multe ori, de forma:

variabilă = expresie

sau, mai simplu:

expresie

Expresiile sunt compuse din operatori sau alte caractere speciale, din funcții și nume de variabile. Dacă ultimul caracter al instrucțiunii este ";", aceasta este executată, dar tipărirea este suprimată.

Deoarece operațiile cu vectori și matrice sunt executate în Matlab mai repede cu un ordin de mărime decât operațiile compilate/interpretate, se obține o viteză de lucru mai mare dacă algoritmi înscrisi în fișierele-M sunt vectorizați (de exemplu: $t=0:0.01:10;$).

Pentru simplificarea reprezentării unele dintre variabilele articulare sunt vectorizate (de exemplu $\theta_l=0:\pi/10:2*\pi;$), iar celelalte sunt ciclitate cu ajutorul instrucției **for**. Instrucțiunea **for** permite repetarea unui grup de instrucțiuni din corpul buclei, de un anumit număr de ori. Are următoarea structură generală:

```
for index=expresie
grupul_de_instrucțiuni
end
```

unde :

- index este numele contorului
- expresie este o matrice, un vector sau un scalar;
- grupul_de_instrucțiuni este orice expresie Matlab.

În aplicații *expresie* este de cele mai multe ori de forma:

$k = \text{valoare inițială} : \text{pas} : \text{valoare finală}.$

Graficele sunt reprezentate prin trasarea unor curbe în 3D, conținând toate punctele pe care le generează tripleta p_x, p_y, p_z din soluția modelului geometric direct al robotului. Reprezentarea este de fapt de același tip ca cea exprimată prin relația (II.2.1) întrucât grupul de ecuații parametrică $p_x = p_x(t), p_y = p_y(t), p_z = p_z(t)$ reprezintă traiectoria originii sistemului atașat efectorului final.

Reprezentarea liniilor în spațiu se face cu funcția **plot3**, care se apelează cu sintaxa:

```
Plot3(x, y, z)
```

unde: x, y, z sunt vectori de aceeași dimensiune, ce reprezintă grafic o linie în spațiul 3D, linie care trece prin punctele ale căror coordonate sunt triplete (x, y, z) . Tripletele (x, y, z) pot fi urmate de perechile parametru/valoare, pentru a specifica proprietăți suplimentare ale liniilor.

Funcția **Hold** păstrează graficul curent.

HOLD ON păstrează graficul curent și toate proprietăților axelor în așa fel încât comenzile următoare de desenare sunt adăugate graficului existent.

HOLD OFF restabilește modul original în care o comandă plot șterge graficul precedent și resetează toate proprietățile axelor înainte de desenarea noului grafic.

Toate aceste considerații sunt prezentate pentru a justifica din ce motive uneori utilizarea unui software profesional, așa cum este Matlabul, este preferată creării unor programe originale. Utilizarea celor din prima categorie menționată asigură pe lângă caracterul de generalitate al aplicației și interschimbabilitatea rezultatelor, absolut necesară în orice cercetare interdisciplinară.

II.5.3. Modelarea suprafețelor descrise de efectorul final

În proiectarea asistată de calculator se folosesc modelele geometrice (cinematice) ale structurii mecanice a roboților pentru analiza caracteristicilor structurale, a dimensiunilor geometrice, a mișcării efectorului final, etc. În comanda mișcării roboților se folosesc aceste modele la transformarea traiectoriilor din coordonate operaționale (coordonate în care se descrie sarcina robotului - în general carteziane) în coordonate robot (cilindrice, sferice sau carteziane, care depind de structura mecanică a respectivului robot) [28], [30], [34], [35], [37], [40], 46], [49], [50].

Indiferent de utilizare și de tipul de robot, modelarea presupune cunoașterea arhitecturii robotului, aplicarea convenției de alegere a sistemelor de referință atașate elementelor acestuia, scrierea matricelor de transfer și înmulțirea lor de la bază către efectorul final. Ultima coloană a matricei ce exprimă poziția și orientarea sistemului

de referință atașat efectorului final în raport cu cel atașat bazei robotului reprezintă ecuațiile de mișcare $p_x = p_x(t)$, $p_y = p_y(t)$, $p_z = p_z(t)$ ce servesc la reprezentarea parametrică a suprafețelor pe care efectorul final se poate deplasa cu orientare variabilă [51], [56], [57], [59], [60], [61], [62], [63], [66], [83], [85], [90], [93], [102], [103], [113], [120], [129], [132], [155], [162], [165], [175], [188], [191], [196], [201].

Se prezintă în cele ce urmează mai multe tipuri de roboți cu sistemele de referință atașate și ale căror ecuații parametrice au fost utilizate pentru reprezentarea 3D a spațiului de lucru propriu al robotului folosind facilitățile Matlab.

➤ **Robotul RT**

Schema cinematică a robotului **RT** este prezentată în figura II.33.

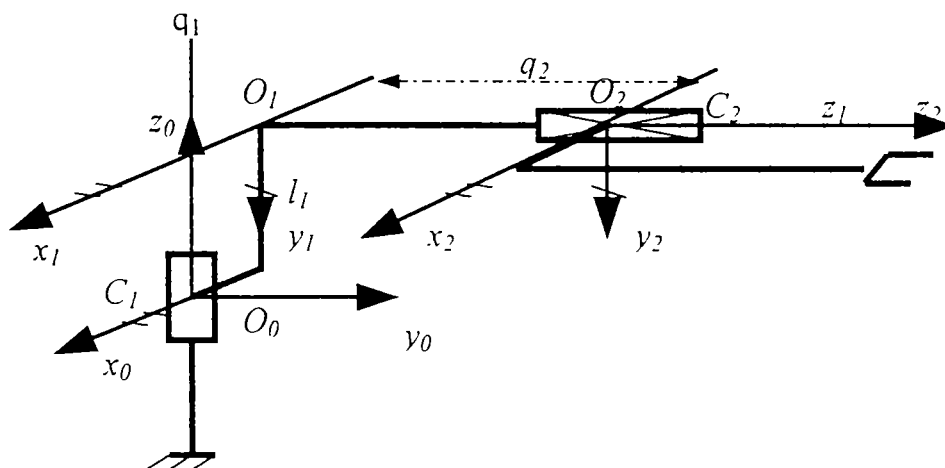


Fig.II.33. Schema cinematică a robotului **RT**

Caracteristicile geometrice și variabilele articulare ale robotului se prezintă în tabelul II.

Tabelul II.1

Nr.	Variabile articulare	l_i	θ_i grade	d_i cm	α_i grade	$\cos \alpha_i$	$\sin \alpha_i$
1	$q_1 = \theta_1$	0	θ_1	l_1	-90°	0	-1
2	$q_2 = d_1$	0	0°	0	0°	1	0

Matricea ce exprimă poziția și orientarea sistemului de referință atașat efectorului final este:

$$G_2 = {}^0T_1 \cdot {}^1T_2 = \begin{bmatrix} c_1 & 0 & -s_1 & -q_2 s_1 \\ s_1 & 0 & c_1 & q_2 c_1 \\ 0 & -1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Deci, ecuațiile cinematice ale robotului **RT** sunt:

$$n_x = \cos q_1$$

$$n_y = \sin q_1$$

$$n_z = 0$$

$$o_x = 0$$

$$o_y = 0$$

$$o_z = -1$$

$$a_x = -\sin q_1$$

$$a_y = \cos q_1$$

$$a_z = 0$$

$$p_x = -q_2 \sin q_1$$

$$p_y = -q_2 \cos q_1$$

$$p_z = l_1$$

Considerând lungimea $l_1 = 10$ cm, s-a reprezentat în figura II.34 suprafața pe care se află traiectoriile originii sistemului atașat efectorului final, descrisă prin instrucțiunile:

```

l1=10;
q1=0:pi/30:2*pi;
plot3(0,0,0)
hold on
for q2=10:2:50
plot3(-q2*sin(q1),q2*cos(q1),0*q1+l1)
end
hold off

```

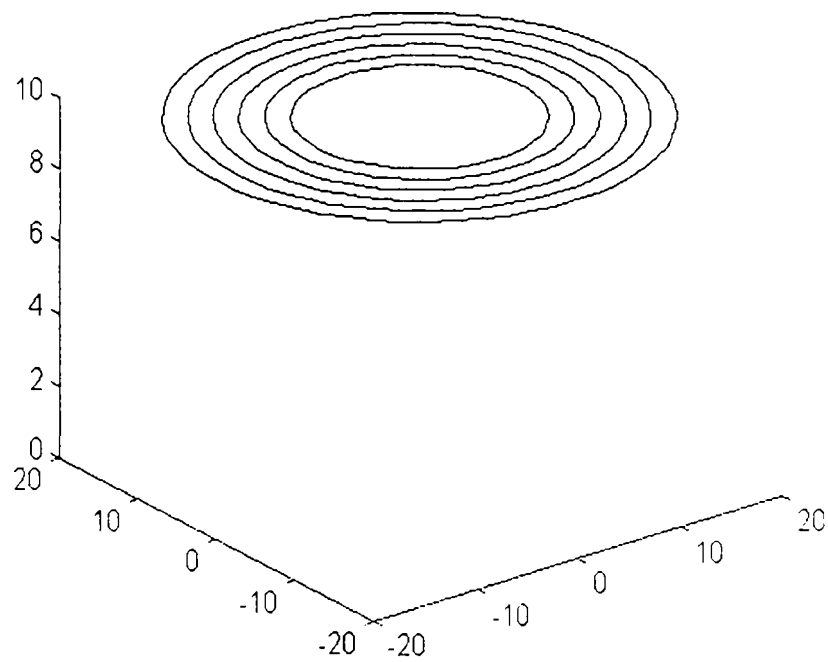


Fig.II.34. *Suprafața proprie de lucru a robotului RT*

$q_1 = \text{var}, q_2 = \text{var}.$

➤ Robotul RRT

Schema cinematică a robotului **RRT** este prezentată în figura II.35 iar caracteristicile geometrice și variabilele articulare ale robotului sunt indicate în tabelul II.2.

Tabelul II.2

Nr.	Variabile articulare	l_i cm	θ_i grade	d_i cm	α_i grade	$\cos \alpha_i$	$\sin \alpha_i$
1	$q_1 = \theta_1$	l_1	θ_1	0	0°	1	0
2	$q_2 = \theta_2$	0	θ_2	0	-90°	0	-1
3	$q_3 = d_3$	0	0°	d_3	0°	1	0

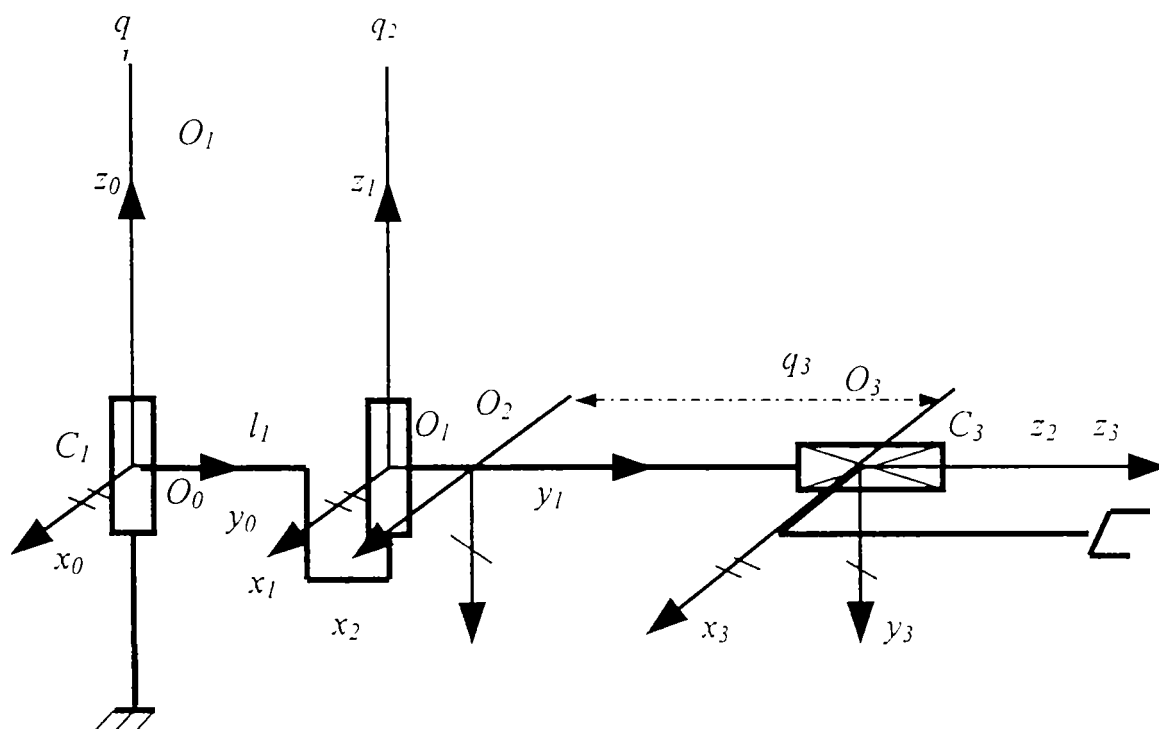


Fig.II.35. Schema cinematică a robotului RRT

Matricea generală ce exprimă poziția și orientarea efectorului final în raport cu sistemul de referință fix este:

$$G_3 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 = \begin{bmatrix} c12 & 0 & -s12 & -q_3s12 + l_1c1 \\ s12 & 0 & c12 & q_3c12 + l_1c1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

și deci, ecuațiile cinematice ale robotului RRT sunt:

$$n_x = \cos(q_1 + q_2)$$

$$n_y = \sin(q_1 + q_2)$$

$$n_z = 0$$

$$o_x = 0$$

$$o_y = 0$$

$$o_z = -l_1$$

$$a_x = -\sin(q_1 + q_2)$$

$$a_y = \cos(q_1 + q_2)$$

$$a_z = 0$$

$$p_x = -q_3 \sin(q_1 + q_2) + l_1 c_1$$

$$p_y = -q_3 \sin(q_1 + q_2) + l_1 s_1$$

$$p_z = 0$$

Pentru valorile $l_1 = 5$ cm și respectiv, $l_1 = 0$ cm, variind pe rând coordonatele articulare s-au reprezentat, în figurile II.36 și II.37, două variante diferite de suprafețe proprii de lucru ale efectorului final, conform listelor de instrucțiuni Matlab:

❖ Varianta 1

```
l1=0;
for q3=0
q1=0:pi/30:2*pi;
plot3(0,0,0)
hold on
for q2=0:2:20
plot3(-q3*sin(q1+q2)+l1*cos(q1),q3*cos(q1+q2)+l1*sin(q1),0*q1)
end
hold off
```

❖ Varianta 2

```
l1=0;
q1=0:pi/20:2*pi;
plot3(0,0,0)
hold on
for q2=0:2:20
for q3=0:pi/20:2*pi
plot3(-q3*sin(q1+q2)+l1*cos(q1),q3*cos(q1+q2)+l1*sin(q1),0*q1)
end
end
hold off
```

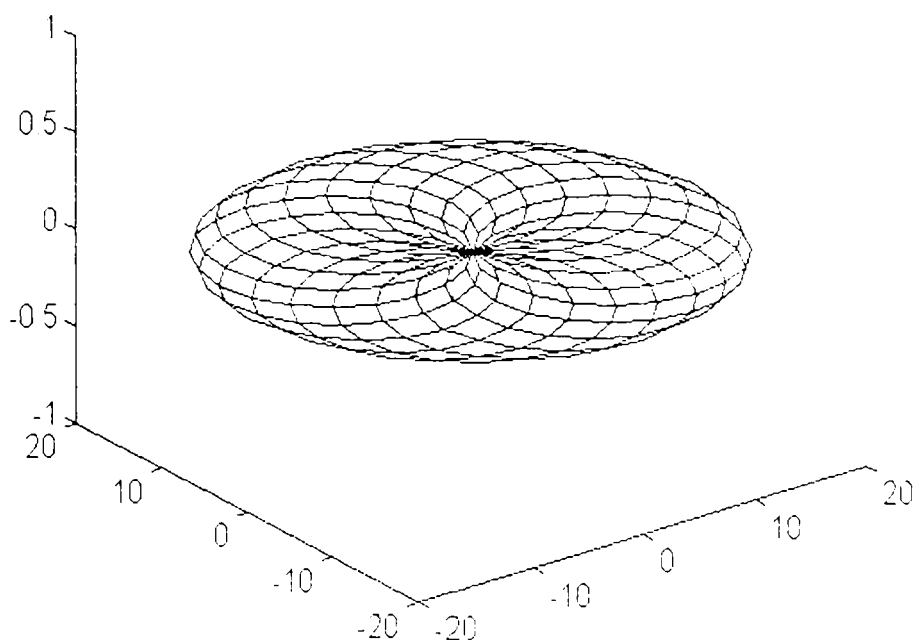


Fig. II.36. Suprafața proprie de lucru a robotului **RT** (Varianta 1)
 $q_1 = \text{var.}$, $q_2 = \text{var.}$, $q_3 = 0$

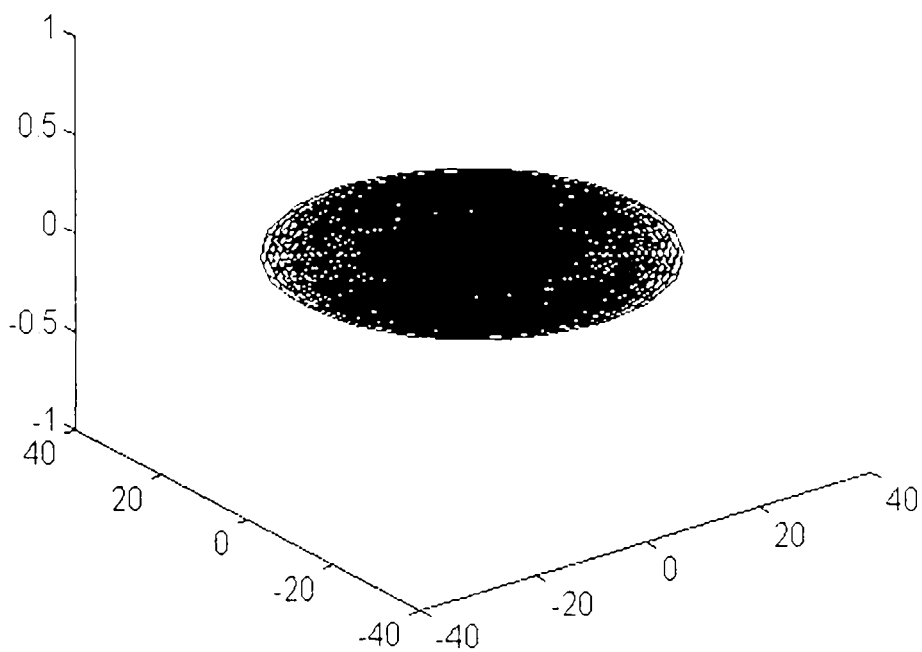


Fig. II.37. Suprafața proprie de lucru a robotului **RT** (Varianta 2)
 $q_1 = \text{var.}$, $q_2 = \text{var.}$, $q_3 = \text{var.}$

➤ Robotul TRTRT

Schema cinematică a robotului **TRTRT** este prezentată în figura II.38 iar caracteristicile sale geometrice și variabilele articulare sunt indicate în tabelul II.3.

Tabelul II.3

Nr.	Variabile articulare	l_i cm	θ_i grade	d_i cm	α_i grade	$\cos \alpha_i$	$\sin \alpha_i$
1	$q_1=d_1$	0	0°	d_1	-90°	0	-1
2	$q_2=\theta_2$	0	θ_2	l_1+l_2	0°	1	0
3	$q_3=d_3$	0	90°	l_3+d_3	90°	0	1
4	$q_4=\theta_4$	0	θ_4	l_4+l_5	0°	1	0
5	$q_5=d_5$	0	0°	d_5	0°	1	0

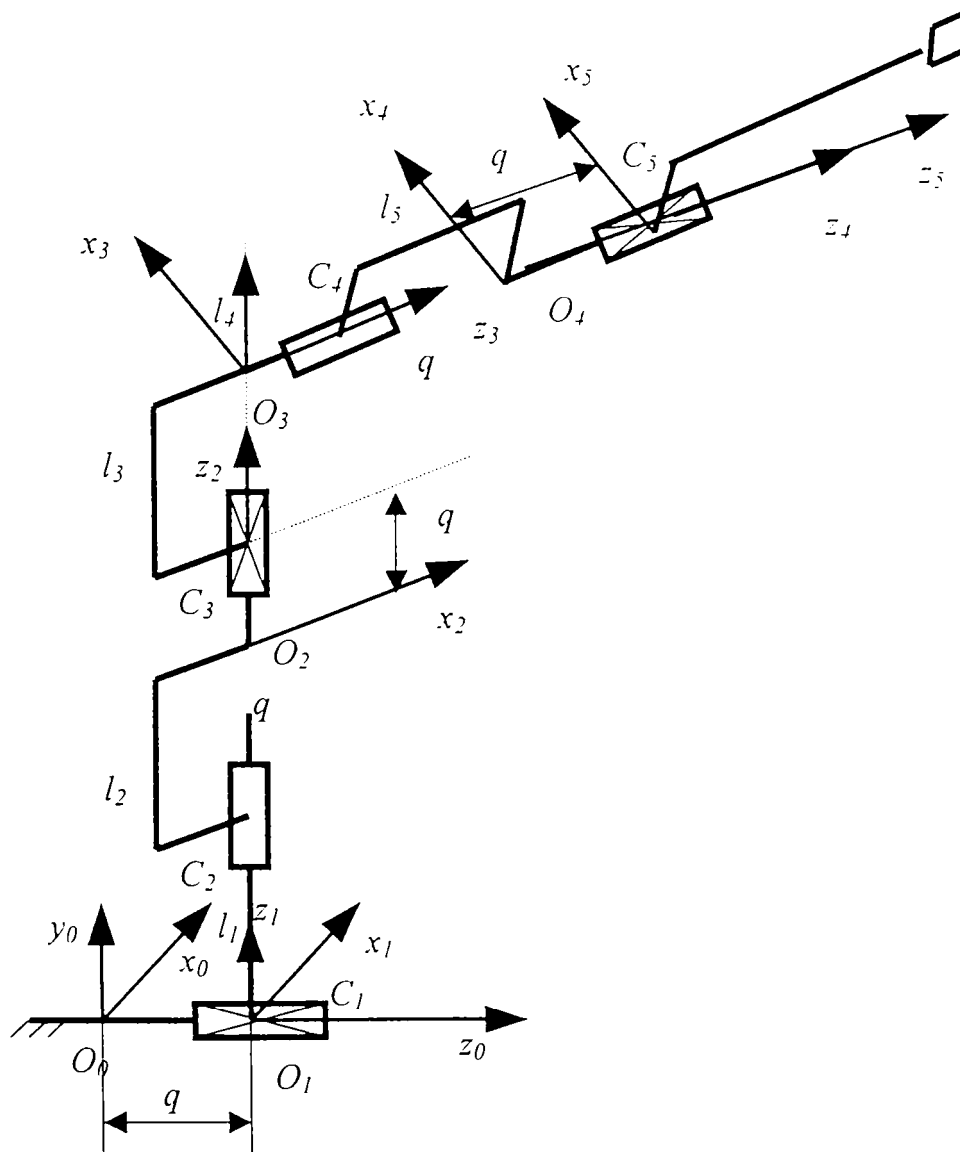


Fig. II.38. Schema cinematică a robotului **TRTRT**

Matricea generală ce exprimă poziția și orientarea efectorului final în raport cu sistemul de referință fix este:

$$G_5 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 = \begin{bmatrix} -s_2c_4 & s_2s_4 & c_2 & (l_4 + l_5 + q_5)c_2 \\ s_4 & c_4 & 0 & l_3 + l_1 + q_3 + l_2 \\ -c_2c_4 & c_2s_4 & -s_2 & -(l_4 + l_5 + q_5)s_2 + q_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

și astfel ecuațiile sale cinematice sunt:

$$n_x = -\sin q_2 \cos q_4$$

$$n_y = \sin q_4$$

$$n_z = -\cos q_2 \cos q_4$$

$$o_x = \sin q_2 \sin q_4$$

$$o_y = \cos q_4$$

$$o_z = \cos q_2 \cos q_4$$

$$a_x = \cos q_2$$

$$a_y = 0$$

$$a_z = -\sin q_2$$

$$p_x = (l_4 + l_5 + q_5) \cos q_2$$

$$p_y = l_1 + l_2 + l_3 - q_3$$

$$p_z = q_1 - (l_4 + l_5 + q_5) \sin q_2$$

Pentru diferite valori constante ale lungimilor l_i ale elementelor, variind pe rând coordonatele articulare s-au reprezentat, în figurile II.39, II.40, II.41 și II.42, patru variante diferite de suprafețe proprii de lucru ale efectorului final, conform listelor de instrucțiuni Matlab:

❖ Varianta 1

```

l=5;
q2=0:pi/25:pi;
q3=0;
plot3((l4+l5+10)*cos(q2), l1+l2+l3-q3)*q2, l0+
(l4-l5+10)*sin(q2));
hold on;
for q4=0:0.5:10;
for q5=0:0.5:10;
plot3((l4+l5-q5)*cos(q2), l1+l2+l3-q3-q4)*q2, q1-
(l4-l5-q5)*sin(q2));
end;
end;
hold off;

```

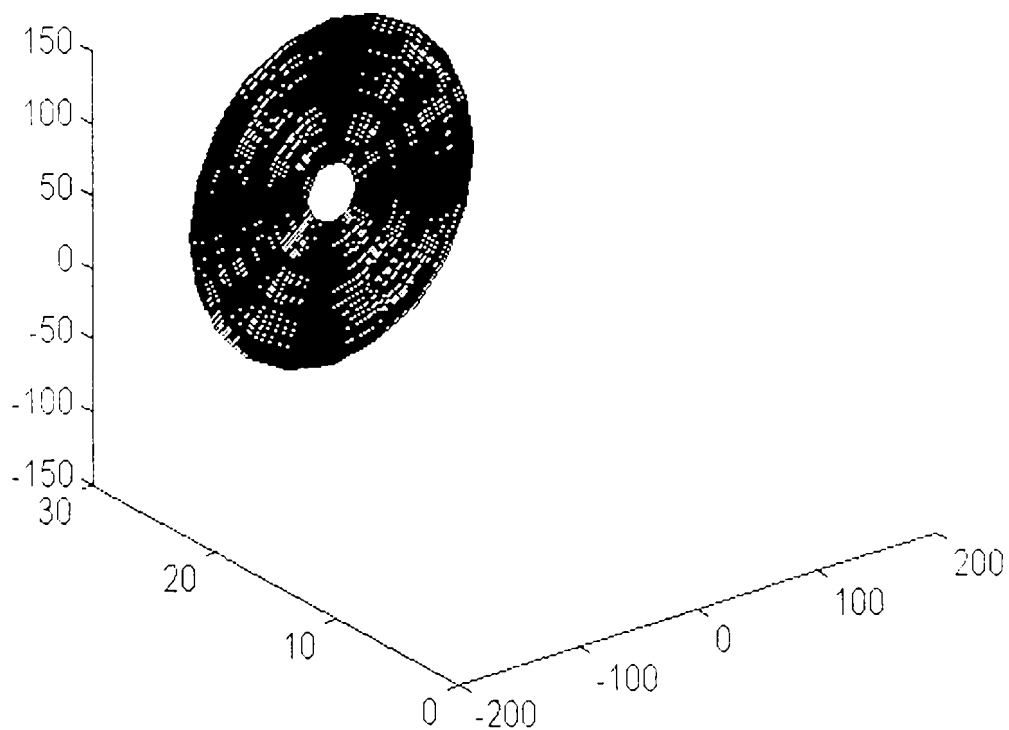


Fig.II.39. Suprafața proprie de lucru a robotului *TRTRT* (Varianta 1)
 $q_1 = 0$, $q_2 = \text{var}$, $q_3 = 0$, $q_4 = \text{var}$, $q_5 = \text{var}$.

❖ Varianta 2

```

l=5;
q2=0:pi/25:pi;
plot3(\sqrt{14-15+10}\*cos(q2) , 11+12+13-14+15\*q2 , \sqrt{14-15+10}\*sin(q2))
hold on
for q1=0:0.5:10
for q3=0:0.5:10
plot3(\sqrt{14-15-q5}\*cos(q2) , (11+12+13-q3+0\*q2) , q1-\sqrt{14-15-q5}\*sin(q2))
end
end
hold off

```

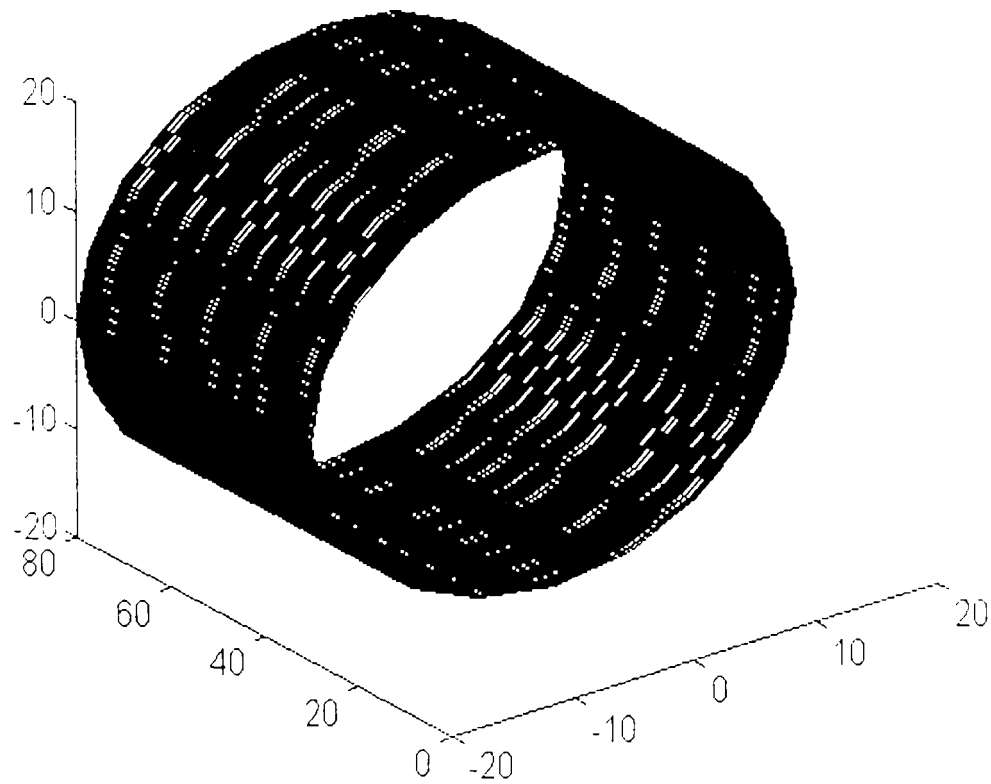


Fig. II.40. Suprafața proprie de lucru a robotului *TRTRT* (Varianta 2)
 $q_1 = \text{var}$, $q_2 = \text{var}$, $q_3 = \text{var}$, $q_4 = 0$, $q_5 = 0$.

❖ Varianta 3

```

l=5;
q2=0:pi/25:pi;
plot3(((14+15+10)*cos(q2)), (l1+l2+l3-10+0*q2), (10-
(14+15+10)*sin(q2)))
hold on
for q1=0:0.5:10
for q3=0:0.5:10
for q5=0:0.5:10
plot3(((14+15+q5)*cos(q2)), (l1+l2+l3-q3+0*q2), (q1-
(14+15+q5)*sin(q2)))
end
end
end
hold off

```

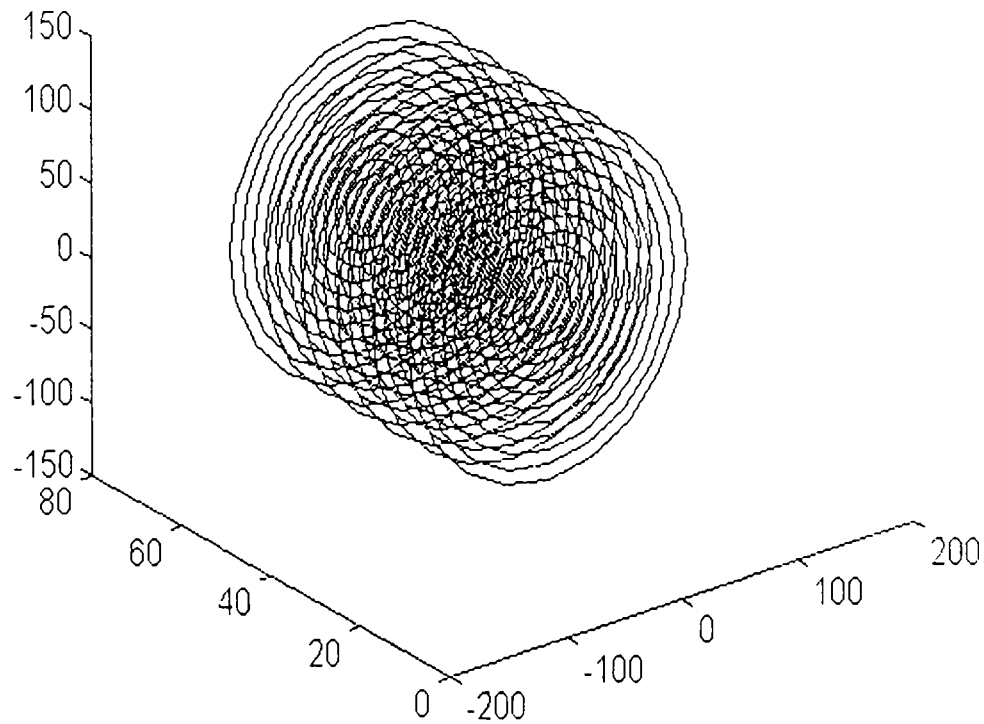


Fig. II.41. Suprafața proprie de lucru a robotului TRTRT (Varianta 3)
 $q_1 = \text{var}$, $q_2 = \text{var}$, $q_3 = \text{var}$, $q_4 = 0$, $q_5 = \text{var}$.

❖ Varianta 4

```

l=5;
q2=0:pi/25:pi;
q4=0:pi/25:pi;
plot3( (l4-l5+l0)*cos(q2), (l1+l2+l3-l0-0*q2), (l0-
(l4-l5+l0)*sin(q2)))
hold on
for q1=0:0.5:10
for q3=0:0.5:10
for q5=0:0.5:10
plot3( ((l4+l5+q5)*cos(q2)), (l1+l2+l3-q3-0*q2), (q1-
(l4+l5-q5)*sin(q2)))
end
end
end
end
end
hold off

```

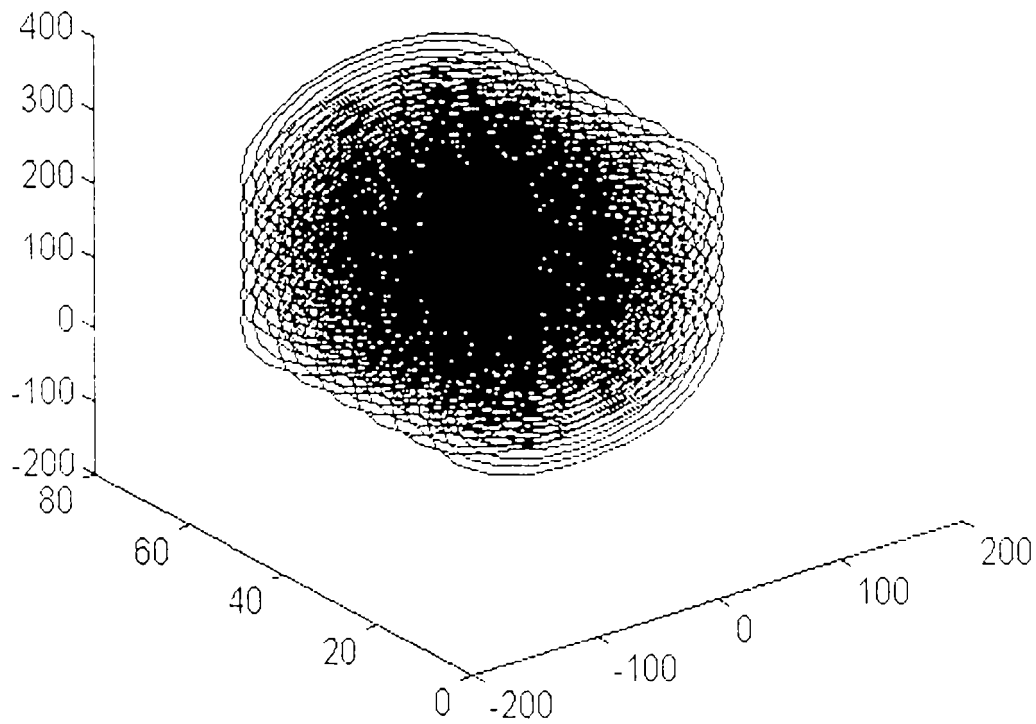


Fig. II.42. Suprafața proprie de lucru a robotului *TRTRT* (Varianta 4)
 $q_1 = \text{var}$, $q_2 = \text{var}$, $q_3 = \text{var}$, $q_4 = \text{var}$, $q_5 = \text{var}$.

➤ Robotul RRRRRR

Schema cinematică a robotului RRRRRR este prezentată în figura II.43 iar valorile parametrilor articulari în tabelul II.4.

Tabelul II.4

Nr.	Variabile articulare	l_i cm	θ_i grade	d_i cm	α_i grade	$\cos \alpha_i$	$\sin \alpha_i$
1	$q_1 = \theta_1$	0	θ_1	0	-90°	0	-1
2	$q_2 = \theta_2$	l_2	θ_2	0	0°	1	0
3	$q_3 = \theta_3$	l_3	θ_3	0	0°	1	0
4	$q_4 = \theta_4$	l_4	θ_4	0	90°	0	1
5	$q_5 = \theta_5$	0	θ_5	0	-90°	0	-1
6	$q_6 = \theta_6$	0	θ_6	0	0°	1	0

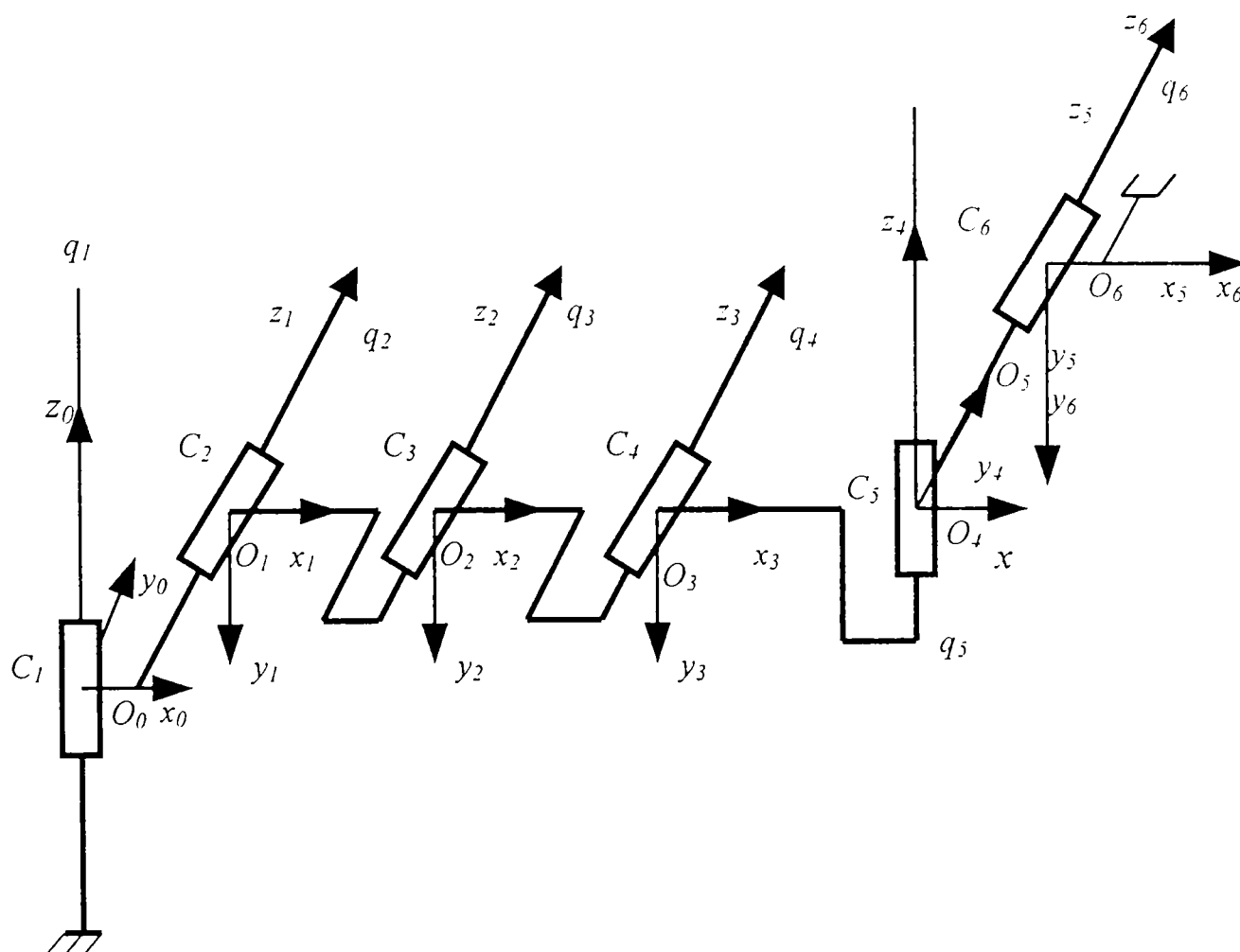


Fig.II.43. Schema cinematică a robotului RRRRRR

Efectuând produsul matricelor de transfer rezultă ecuațiile cinematice ale robotului :

$$n_x = c1(c234 c5 c6 - s234 s6) - s1 s5 c6$$

$$n_y = s1(c234 c5 c6 - s234 s6) + c1 s5 c6$$

$$n_z = -s234 c5 c6 - c234 s6$$

$$o_x = -c1(c234 c5 s6 + s234 c6) + s1 s5 s6$$

$$o_y = -s1(c234 c5 s6 + s234 c6) - c1 s5 s6$$

$$o_z = s234 c5 s6 - c234 c6$$

$$a_x = -c1 c234 s5 - s1 c5$$

$$a_y = -s1 c234 s5 + c1 c5$$

$$a_z = s234 s5$$

$$p_x = c1(l_4 c234 + l_3 c23 + l_2 c2)$$

$$p_y = s1(l_4 c234 + l_3 c23 + l_2 c2)$$

$$p_z = -(l_4 s234 + l_3 s23 + l_2 s2)$$

Pentru diferite valori constante ale lungimilor l_i ale elementelor, variind pe rând coordonatele articulare s-au reprezentat, în figurile II.44, II.45, II.46, II.47, II.48 și II.49, șase variante diferite de suprafețe proprii de lucru ale efectorului final, conform listelor de instrucțiuni Matlab:

❖ Varianta 1

```

l4=5
l3=10
l2=20
q4=0;
q1=0:pi/10:2*pi;
plot3(0,0,0)
hold on
for q2=pi:pi/10:2*pi
for q3=pi:pi/10:2*pi
plot3(cos(q1)*(l4*cos(q3+q2+q4)+l3*cos(q3+q2)+l2*cos(q2)),sin(q1)*(l4*cos(q3+q2+q4)+l3*cos(q3+q2)+l2*cos(q2)),0*q1-
,l4*sin(q3+q2+q4)+l3*sin(q3+q2)+l2*sin(q2))
end
end
end
hold off

```

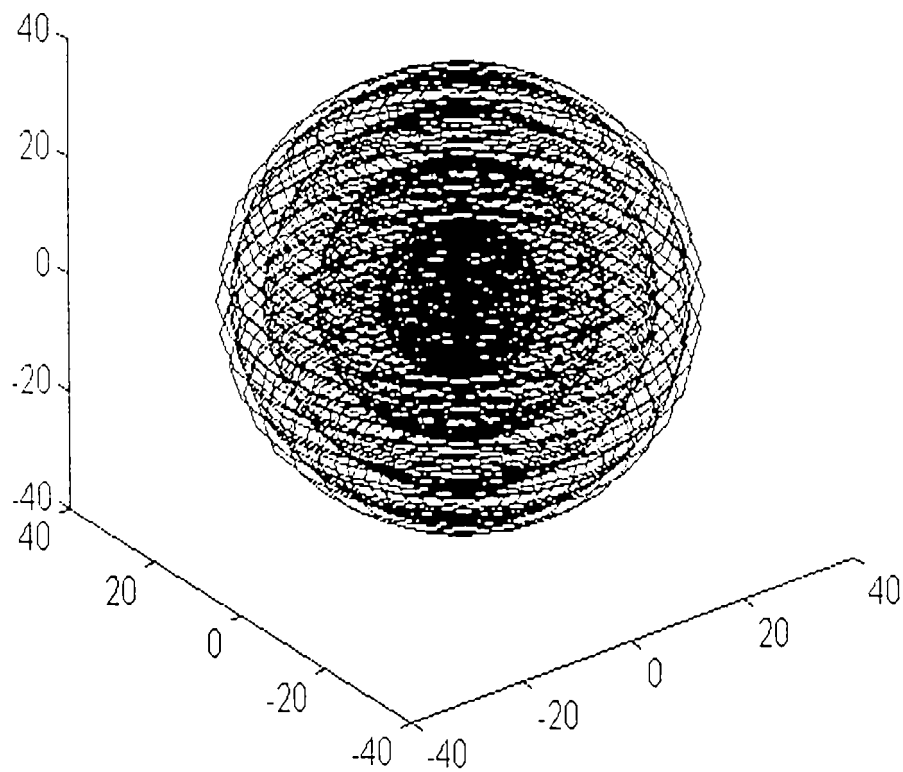


Fig.II.44. Suprafața proprie de lucru a robotului **RRRRRR** (Varianta 1)
 $q_1 = \text{var}$, $q_2 = \text{var}$, $q_3 = \text{var}$, $q_4 = 0$, $q_5 = 0$, $q_6 = 0$

❖ Varianta 2

```

l4=5
l3=10
l2=20
q3=0;
q1=0:pi/10:2*pi;
plot3(0,0,0)
hold on
for q2=pi:pi/10:2*pi
for q4=pi:pi/10:2*pi
plot3(cos(q1)*(l4*cos(q3+q2+q4)+l3*cos(q3+q2)+l2*cos(q2)),sin(q1)*l4
*cos(q3+q2+q4)+l3*cos(q3+q2)+l2*cos(q2)),0*q1-
(l4*sin(q3+q2+q4)+l3*sin(q3+q2)+l2*sin(q2)))
end
end
end

```

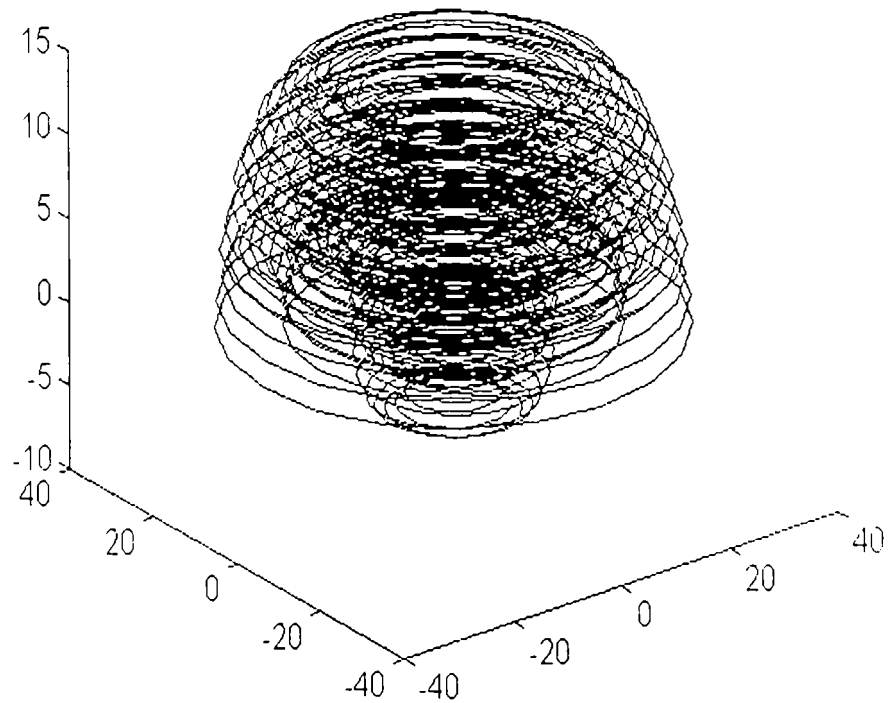


Fig. II.45. Suprafața proprie de lucru a robotului **RRRRRR** (Varianta 2)
 $q_1 = \text{var}$, $q_2 = \text{var}$, $q_3 = 0$, $q_4 = \text{var}$, $q_5 = 0$, $q_6 = 0$.

❖ Varianta 3

```

l4=5
l3=10
l2=20
q2=0;
q1=0:pi/10:2*pi;
plot3(0,0,0)
hold on
for q3=pi:pi/10:2*pi
for q4=pi:pi/10:2*pi
plot3(cos(q1)*(l4*cos(q3+q2+q4)+l3*cos(q3+q2)+l2*cos(q2)),sin(q1)*(l4*cos(q3+q2+q4)+l3*cos(q3+q2)+l2*cos(q2)),0*q1-
(l4*sin(q3+q2+q4)+l3*sin(q3+q2)+l2*sin(q2)))
end
end
end
hold off

```

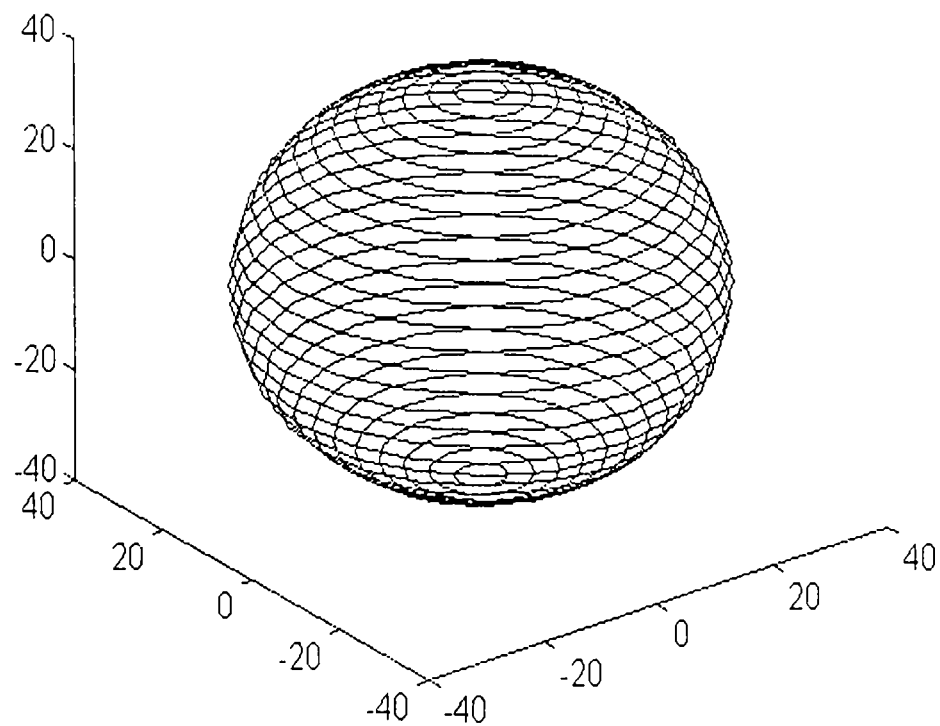


Fig.II.46. Suprafața proprie de lucru a robotului **RRRRRR** (Varianta 3)
 $q_1 = \text{var}$, $q_2 = 0$, $q_3 = \text{var}$, $q_4 = \text{var}$, $q_5 = 0$, $q_6 = 0$.

❖ Varianta 4

```

l4=5
l3=10
l2=20
q1=0;
q2=0:pi/10:2*pi;
plot3(0,0,0)
hold on
for q3=pi:pi/10:2*pi
for q4=pi:pi/10:2*pi
plot3(cos(q1)*(l4*cos(q3+q2+q4)+l3*cos(q3+q2)+l2*cos(q2)),sin(q1)*(l4*cos(q3+q2+q4)+l3*cos(q3+q2)+l2*cos(q2)),0*q1-
(l4*sin(q3+q2+q4)+l3*sin(q3+q2)+l2*sin(q2)));
end
end
end
hold off

```

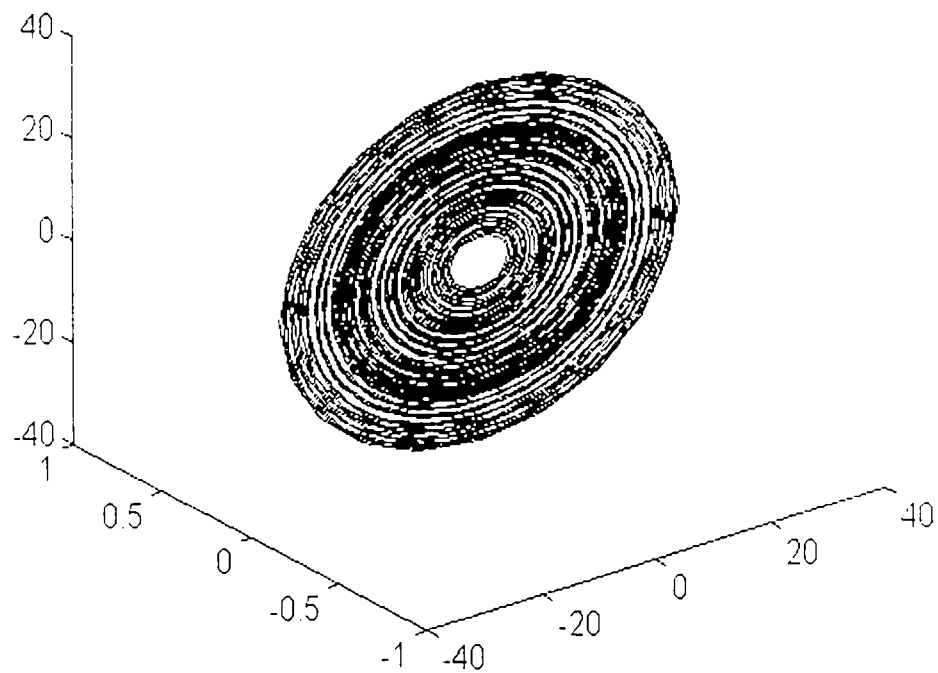


Fig. II.47. Suprafața proprie de lucru a robotului **RRRRRR** (Varianta 4)
 $q_1 = 0$, $q_2 = \text{var}$, $q_3 = \text{var}$, $q_4 = \text{var}$, $q_5 = 0$, $q_6 = 0$.

❖ Varianta 5

```

l4=5;
l3=10;
l2=20;
q3=0;
q4=0;
q1=0:pi/10:2*pi;
plot3(0,0,0)
hold on
for q2=0:pi/30:2*pi
plot3(cos(q1)*(l4*cos(q3+q2+q4)+l3*cos(q3+q2)+l2*cos(q2)),sin(q1)*(l4*cos(q3+q2+q4)+l3*cos(q3+q2)+l2*cos(q2)),0*q1-(l4*sin(q3+q2+q4)+l3*sin(q3+q2)+l2*sin(q2)))
end
hold off

```

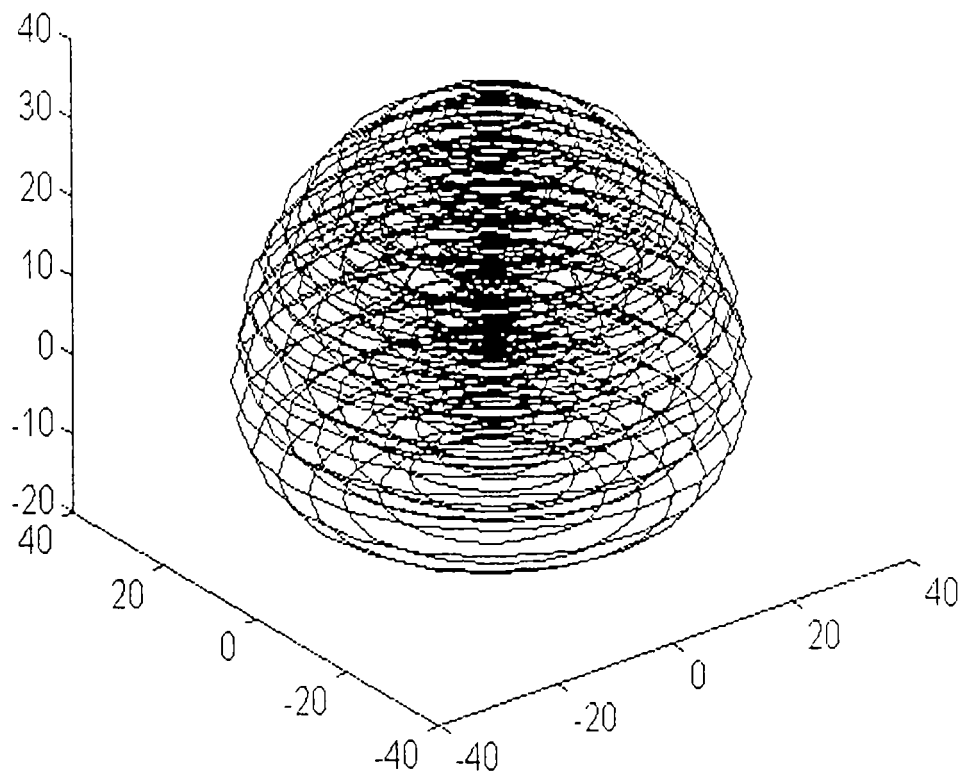


Fig. II.48. Suprafața proprie de lucru a robotului RRRRRR (Varianta 5)
 $q_1 = \text{var}$, $q_2 = \text{var}$, $q_3 = 0$, $q_4 = 0$, $q_5 = 0$, $q_6 = 0$.

❖ Varianta 6

```

l4=5;
l3=10;
l2=20;
q4=0;
q1=0:pi/10:2*pi;
plot3(0,0,0)
hold on
for q2=0:pi/20:2*pi
for q3=0:pi/5:2*pi
plot3(cos(q1)*(l4*cos(q3+q2+q4)+l3*cos(q3+q2)+l2*cos(q2)),sin(q1)*(l4*cos(q3+q2+q4)+l3*cos(q3+q2)+l2*cos(q2)),0*q1-(l4*sin(q3+q2+q4)+l3*sin(q3+q2)+l2*sin(q2)))
end
end
hold off

```

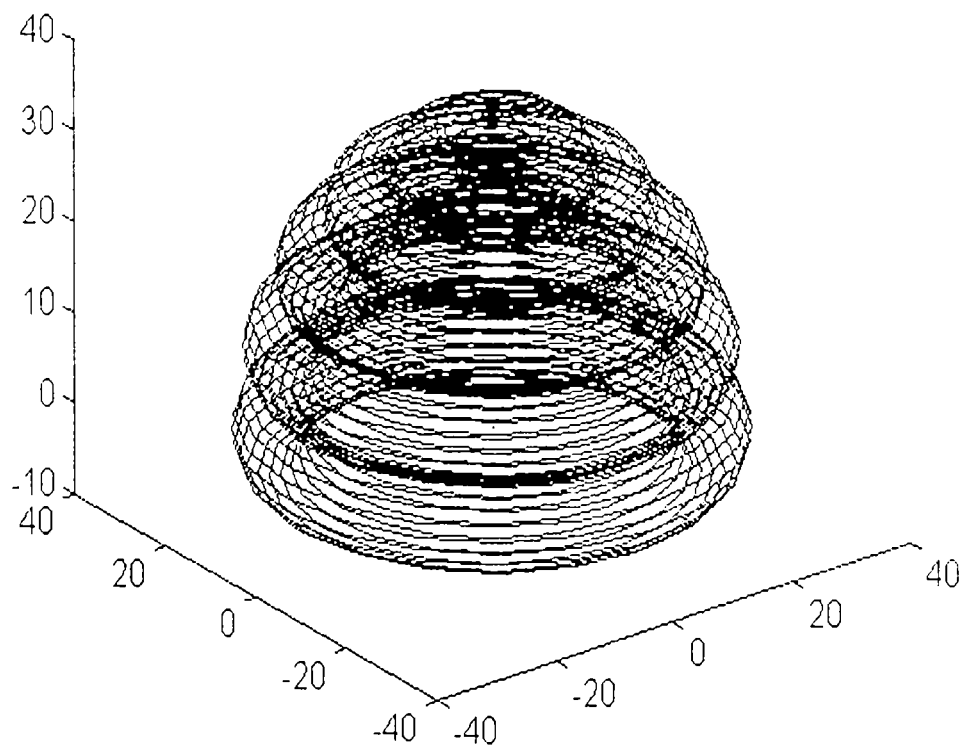


Fig. II.49. Suprafața proprie de lucru a robotului **RRRRRR** (Varianta 6)
 $q_1 = \text{var}$, $q_2 = \text{var}$, $q_3 = \text{var}$, $q_4 = 0$, $q_5 = 0$, $q_6 = 0$.

III. Reprezentarea geometrică

III.1. Generalități

Motivația pentru acest tip de reprezentare este facilitarea unui mod interactiv de modelare a solidului. Ideea pe care se bazează acest tip de reprezentare este aceea că obiectele se realizează prin anumite procese tehnologice al căror produs final sunt corpuri de forme complexe dar care pot fi construite combinând obiecte elementare numite primitive geometrice. Aceste primitive sunt sfere, conuri, cilindri sau diverse poliedre, combinate între ele prin folosirea operatorilor booleeni și a transformărilor liniare [14], [15], [29], [36], [88], [91].

Editarea formei complexe este posibilă deoarece reprezentarea definește nu numai forma obiectului dar și istoria modelării sale. Reprezentarea prin utilizarea de primitive geometrice este mai simplu de realizat prin metode numerice și de aceea multe din programele profesionale de construcție a corpurilor 3D utilizează această metodă în detrimentul metodelor de reprezentare parametrică. De exemplu, creșterea diametrului unei găuri într-un paralelipiped este o modificare neimportantă, ea implicând doar creșterea razei primitivei, care în acest caz este un cilindru. Această operație se realizează mai simplu decât în reprezentarea parametrică prin porțiuni de suprafață care se definesc complicat și care trebuie să asigure acuratețea și continuitatea reprezentării (Capitolul II).

Operatorii booleeni sunt folosiți atât ca formă de reprezentare cât și ca interfață. Utilizatorul specifică solidele primitive și le combină folosind operații booleene. Astfel reprezentarea și modelarea nu sunt operații separate, ci activitatea de modelare conduce la reprezentare [100], [111], [118], [122], [127], [133], [134], [146].

Intr-un fel, această metodă de modelare este o metodă clasică în mecanică, deoarece principiile sale sunt aceleași cu cele utilizate la calculul poziției centrului de masă al unui corp complex și al momentelor sale de inerție.

Figura III.1 prezintă o posibilitate de construcție geometrică a solidului prin 3 primitive geometrice: două paralelipipede și un cilindru. Paralelipipelele sunt combinate folosindu-se operatorul de unire și se realizează corpul asamblat într-o fază primară (asemănătoare unui semifabricat). Prin definirea acestuia și a unui cilindru se va putea realiza corpul final prin extragerea corpului cilindric din semifabricatul realizat ca ansamblu celor două paralelipipede.

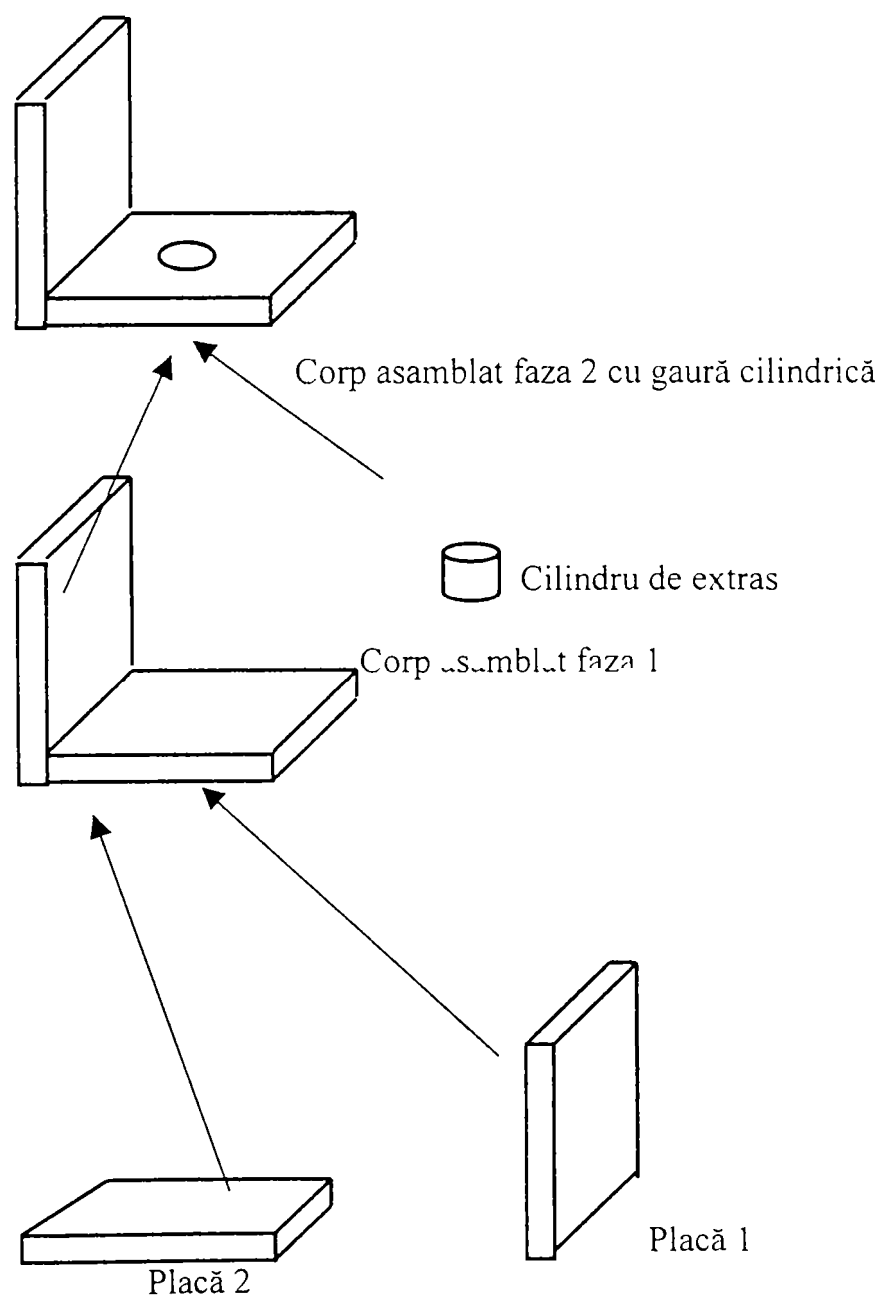


Fig.III.1. Realizarea unui corp complex din primitive geometrice

Deși există avantaje substanțiale în reprezentarea corpurilor solide prin primitive geometrice, există și dezavantaje. Acestea s-ar putea rezuma prin:

- timpul relativ lung necesar producerii imaginii ce reprezintă obiectul
- limitările impuse de metodă relativ la operațiile ce pot fi folosite pentru a crea și modela un solid; operațiile booleene sunt globale, ele afectând întregul corp și din acest motiv, operațiile locale cu efect limitat nu pot fi ușor implementate.

Strategiile de proiectare pentru modelele de corp solid sunt diferite. Caracteristica distinctivă a unui astfel de model constă în faptul că ea nu rezultă prin construcția suprafețelor care mărginesc corpul. Baza de date în această reprezentare este constituită dintr-o mulțime de primitive și de operații booleene. Această reprezentare facilitează o interactivitate puternică cu utilizatorul, dar strategia de proiectare este complexă și scumpă.

III.2. Reprezentarea prin sfere

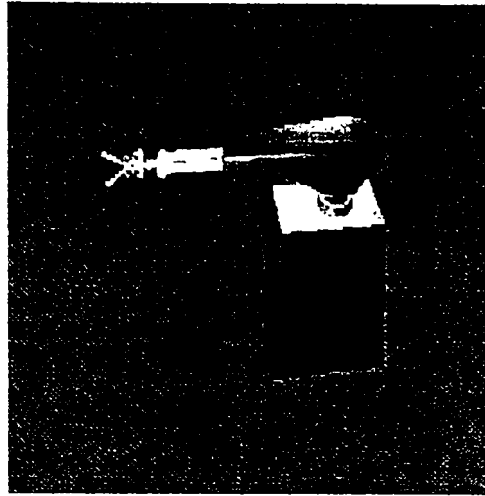
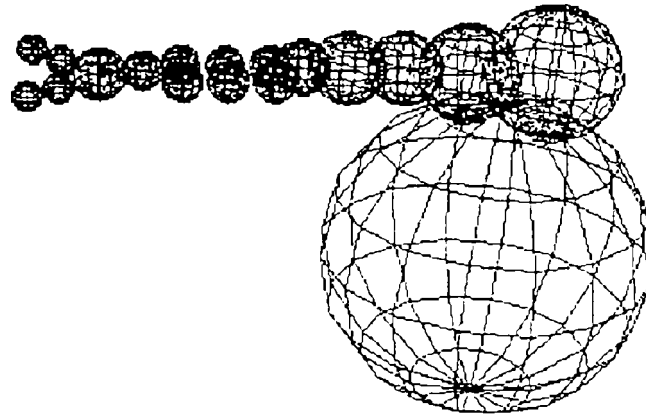
III.2.1. Generalități

Sfera este cel mai simplu obiect geometric, de aceea, multe probleme pot fi rezolvate numai cu sfere. Două caracteristici ale sferei îi denotă simplitatea:

- cunoscând poziția centrului, ea poate fi definită cu ajutorul unui singur parametru, raza;
- fiind precizată raza, mișcarea sferei este descrisă de mișcarea unui singur punct, centrul său.

Se pot prezenta avantajele multiple ale reprezentării spațiale prin sfere în robotică, dar în primul rând această reprezentare este utilizată pentru că are un caracter general și este extrem de simplă. De exemplu: un robot ca cel din figura III. 2 poate fi reprezentat ca o mulțime de sfere în figura III. 3.

Orice model este compus din două mulțimi de sfere: o *reprezentare exterioară*, care face o aproximare excedentară a formei sistemului real învăluindu-l cu sfere și o *reprezentare interioară*, compusă din sfere plasate în interiorul sistemului real care realizează o aproximare prin deficit.

Fig. III. 2. *Un robot real*Fig. III. 3. *Reprezentarea prin sfere a robotului din Fig. III. 2*

Prima modalitate de reprezentare este în mod uzual utilizată la modelarea corpurilor, cea de-a doua dovedindu-se foarte utilă în procesul de detectare a coliziunilor la problemele de planificare a mișcărilor. Problema inițială este de a realiza modelul spațial al roboților și al obstacolelor și aceasta se rezolvă prin reprezentare exterioară, iar ulterior aceste modele la care se adaugă reprezentarea interioară sunt aplicate în problemele de mișcare, mai precis în detectarea coliziunilor și planificarea mișcărilor în 3D.

Posibilitatea utilizării aproximării duble a reprezentării unui corp bazată pe sfere, a fost concepută observându-se că multe probleme particulare pot fi corect și exact rezolvate în acest mod, dar că este destul de dificil de utilizat în cazul general.

Astfel au fost implementate câteva tehnici euristice noi care au crescut gradul de precizie al reprezentării și au sporit posibilitățile de control curent.

Principalele caracteristici ale modelării prin sfere sunt:

- Singurele elemente ce iau parte la reprezentare sunt sferele; orice problemă se reduce la sfere; natura particulară a obiectelor poate fi ignorată.
- Modelul se compune întotdeauna din două grupuri de sfere: sferele din prima mulțime, ce constituie *reprezentarea exterioară* care acoperă suprafața obiectului, iar sferele din a doua mulțime ce constituie *reprezentarea interioară*, umplând obiectul.
- Pentru fiecare reprezentare este introdus câte un grup de parametrii care se reevaluează pe măsura creșterii gradului de precizie a modelului; această reprezentare ierarhizată conduce practic la o modelare fără erori.

Primul pas al modelării astfel abordate, este *sfericizarea* tuturor obiectelor relevante pentru problemă. Aceasta constă în stabilirea mulțimii de sfere exterioare care include în ea suprafețele exterioare ale obiectului și a mulțimii de sfere interioare care trebuie să se afle în interiorul obiectului reprezentat.

În plus fiecare reprezentare trebuie să aibă anumite proprietăți:

- *să fie perfectibilă* adică să poată fi realizată oricând o reprezentare mai bună ce poate înlocui o alta anterior realizată;
- *să fie echilibrată*, adică erorile pe care le introduce să nu fie concentrate în anumite zone ale modelului, ci să fie distribuite;
- *să fie optimă* în raport cu un anumit criteriu, adică folosind același număr de sfere să nu se poată obține o reprezentare mai bună.

Eficiența nu este un factor foarte important în acest stadiu. Toate reprezentările sunt calculate o singură dată și apoi stocate. Considerând procesul de sfericizare, timpul necesar pentru generarea reprezentărilor unui grup de obiecte este neglijabil comparat cu timpul+efortul+cheltuielile pentru orice altă metodă de modelare geometrică.

Este important de menționat faptul că procesul sfericizării are o componentă euristică importantă din punct de vedere conceptual. Acest lucru se datorează complexității inerente a problemei. Pentru același număr de sfere se pot găsi diferite soluții de reprezentare a modelului. Aceste soluții vor fi valide în măsura în care definesc corect, ca reprezentări interioare și exterioare, obiectul de modelat.

III.2.2. Considerații matematice asupra reprezentării prin sfere

III.2.2.1. Noțiuni și relații fundamentale

Problemele cu caracter aplicativ abordate în teză, impun ca într-un spațiu determinat populat cu obiecte de forme și dimensiuni date, indiferent că sunt obstacole sau roboți, să se găsească o modalitate unică de reprezentare a acestora care să satisfacă la anumite criterii și care să poată aproxima cu o încredere impusă în prealabil aceste corpuri, oricât de complexe ar fi ele.

Se vor prezenta, în cele ce urmează, câteva noțiuni și relații fundamentale de geometrie în spațiu precum și proprietăți fundamentale referitoare la sfere, necesare aplicațiilor și constituind baza teoretică a programelor realizate pentru sfericizare [108], [182].

➤ Aria laterală a unui trunchi de con

Teorema III.1 *Aria generată de un segment de dreaptă, care se rotește în jurul unei axe situate în același plan cu el și care nu îl intersectează, are ca măsură produsul dintre proiecția segmentului pe axă și lungimea cercului cu centru pe axă și tangent segmentului în mijlocul său.*

Demonstrație: Aria generată de un segment de dreaptă AB (figura III.4), care se rotește în jurul unei axe Ox , situată în același plan cu el și care nu îl intersectează, este în general aria unui trunchi de con, ale cărui cercuri de bază au razele egale respectiv cu perpendicularele AA' , BB' , coborâte pe axă din punctele A și B .

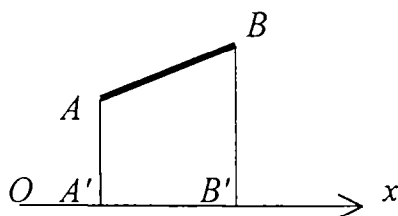


Fig. III.4. Segment de dreaptă generând un trunchi de con

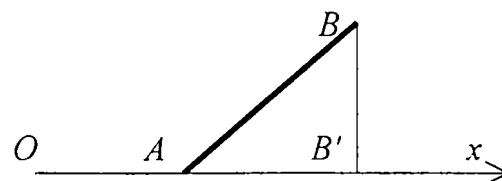


Fig. III.5. Segment de dreaptă generând un con

În caz excepțional, trunchiul de con se reduce la un con dacă segmentul are o extremitate pe axă (figura III.5), sau la un cilindru dacă segmentul este paralel cu axa (figura III.6).

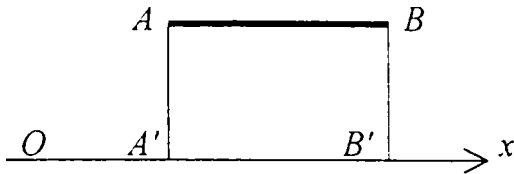


Fig. III.6. Segment de dreaptă generând un cilindru

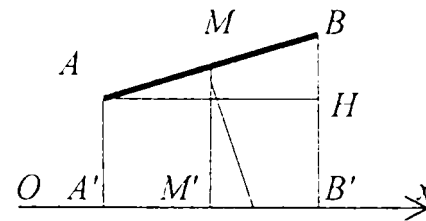


Fig. III.7. Definierea elementelor pentru calculul ariei laterale a trunchiului de con

Fie M mijlocul lui AB (figura III.7) și MM' perpendiculara coborâtă din acest punct pe axă. Aria laterală a trunchiului de con are ca măsură produsul $2\pi \cdot AB \cdot MM'$, expresie care rămâne valabilă și atunci când trunchiul de con este înlocuit fie printr-un con, fie printr-un cilindru, fie printr-o coroană circulară sau un cerc (dacă $AB \perp Ox$).

Expresia $2\pi \cdot AB \cdot MM'$ a ariei se poate înlocui cu alta care este mai avantajoasă. Se duce prin punctul M perpendiculara pe AB , care întâlnește axa de rotație în O . Pe de altă parte, se consideră paralela dusă prin A la axă, care intersectează BB' în H . Segmentul AH este congruent cu proiecția $A'B'$ a segmentului AB pe axă. Triunghiurile dreptunghice ABH și $MM'O$ sunt asemenea, de unde rezultă:

$$\frac{AB}{OM} = \frac{AH}{MM'} \Leftrightarrow AB \cdot MM' = OM \cdot AH \quad (\text{III.2.1})$$

astfel încât aria generată de AB este:

$$S_{tr.con} = 2\pi \cdot AB \cdot MM' = 2\pi \cdot OM \cdot AH = 2\pi \cdot OM \cdot A'B' \quad (\text{III.2.2})$$

➤ Aria zonei sferice

Definiții. Se numește *zonă sferică* porțiunea de suprafață cuprinsă între două plane paralele (figura III.8). Cercurile situate în cele două plane și care mărginesc

zona se numesc *baze*. Distanța dintre cele două plane ale bazelor este *înălțimea zonei*. Una dintre cele două porțiuni determinate pe o suprafață sferică prin intersecția cu un plan se numește *calotă sferică*.

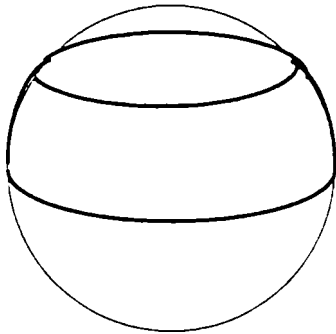
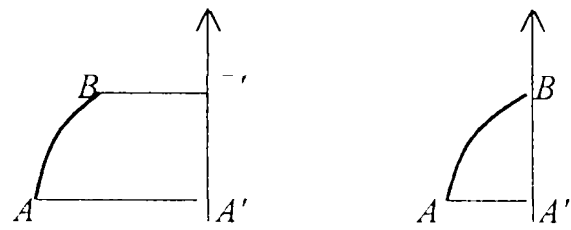


Fig.III.8. Zonă sferică



a)

b)

Fig.III.9. Generarea unei zone sferice

În mod evident, se poate considera calota sferică drept o zonă în care unul dintre planele de bază este tangent la sferă.

Zona (sau calota) mai poate fi definită ca suprafața generată prin rotația unui arc de cerc AB (figura III.9) în jurul unui diametru care nu are puncte comune cu el (cazul zonei, figura III.9,a) sau care trece prin una dintre extremitățile lui (cazul calotei, figura III.9,b). Înălțimea este atunci proiecția $A'B'$ a arcului AB pe axă.

Pentru a defini aria zonei, se înlocuiește întâi arcul AB printr-o linie frântă $ACDEB$ (figura III.10), înscrisă în acel arc.

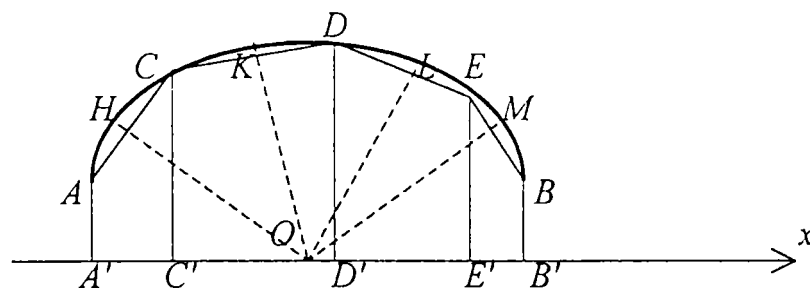


Fig.III.10. Aria generată de un sector poligonal

Prin definiție, aria zonei va fi limita către care tinde aria generată de această linie prin rotație în jurul axei, dacă numărul de laturi ale liniei crește indefinit, astfel ca fiecare dintre ele să tindă către zero. Existența acestei limite și expresia ei rezultă din următoarele două teoreme:

Teorema III.2. *Aria generată de o linie frântă înscrisă într-un cerc, care se rotește în jurul unui diametru cu care nu are puncte comune, este egală cu proiecția liniei frânte pe diametru, înmulțită cu lungimea unui cerc a cărui rază este cuprinsă între cea mai mare și cea mai mică dintre distanțele de la centru la laturile liniei frânte.*

Demonstrație. Fie linia frântă $ACDEB$ (figura III.10), înscrisă într-un cerc de centru O și care se rotește în jurul diametrului Ox care nu o taie (una dintre extremitățile liniei frânte sau ambele putând fi pe Ox). Fie A', C', D', E', B' , proiecțiile vârfurilor pe Ox .

Perpendicularele pe mijloacele H, K, L, M , ale laturilor liniei frânte sunt concurente în punctul O , de unde, conform relațiilor (III.2.2), rezultă:

$$\begin{aligned} S_{gener.AC} &= 2\pi \cdot A'C' \cdot OH \\ S_{gener.CD} &= 2\pi \cdot C'D' \cdot OK \\ S_{gener.DE} &= 2\pi \cdot D'E' \cdot OL \\ S_{gener.EB} &= 2\pi \cdot E'B' \cdot OM \end{aligned} \tag{III.2.3}$$

Cazul de excepție al teoremei III.1 nu poate fi prezentat în raționamentul de față, o coardă a unui cerc neputând fi perpendiculară pe un diametru care nu o intersectează.

Aria generată de linia $ACDEB$ are deci ca măsură:

$$S_{gener.ACDEB} = 2\pi \cdot (A'C' \cdot OH + C'D' \cdot OK + D'E' \cdot OL + E'B' \cdot OM) \tag{III.2.4}$$

Expresia cuprinsă între paranteze este egală cu produsul dintre suma $A'C' + C'D' + D'E' + E'B'$, adică $A'B'$ și o cantitate cuprinsă între cea mai mare și cea mai mică dintre cantitățile OH, OK, OL, OM , ceea ce demonstrează teorema.

Dacă linia frântă este regulată, distanțele OH, OK, OL, OM sunt toate egale cu apotema acestei linii frânte.

Corolar. *Aria generată de o linie frântă regulată, care se rotește în jurul unei axe din planul ei, care trece prin centrul ei și nu o intersectează, este egală cu proiecția acestei linii pe axă, înmulțită cu lungimea cercului înscris.*

Teorema III.3. *Aria generată de un arc de cerc care se rotește în jurul unui diametru ce nu-l taie, este egală cu proiecția arcului pe diametru, înmulțită cu lungimea cercului întreg.*

Sau, *aria zonei este egală cu produsul dintre înălțimea ei și lungimea unui cerc mare.*

Demonstrație. Fie arcul AB , cu centrul O , care se rotește în jurul diametrului determinat de Ox , $A'B'$ proiecția lui AB pe Ox . Dacă în arcul AB se înscrie linia frântă $ACDEB$, aceasta, rotindu-se în jurul lui Ox , va genera o arie egală cu produsul dintre $2\pi \cdot A'B'$ și o cantitate intermediară între cea mai mare și cea mai mică dintre distanțele de la centru la laturile liniei frânte.

Dacă va crește nemărginit numărul de laturi, astfel ca fiecare dintre ele să tindă către zero, toate distanțele precizate mai sus vor tinde către raza $OA = R$ a cercului.

Așadar, aria generată are o limită, independentă de legea după care crește numărul de laturi ale liniei frânte (cu condiția numai ca toate aceste laturi să tindă către zero), iar această limită este:

$$S_{gener. lim} = 2\pi \cdot A'B' \quad (III.2.5)$$

Observație. Raționamentul de mai sus și concluzia sa se aplică și cazului unei calote sferice.

Corolar. *Din teorema precedentă rezultă că ariile a două zone ale aceleiași sfere sunt proporționale cu înălțimile lor.*

➤ Aria sferei

Teorema III.4. *Aria sferei cu raza R este $4\pi R^2$.*

Demonstrație. Raționamentul din teorema III.3 rămâne valabil și când arcul AB devine un semicerc. Proiecția $A'B'$ are ca măsură mărimea diametrului $2R$; zona generată este sfera întreagă. Suprafața acestei sfere are deci ca măsură produsul:

$$S_{sfera} = 2\pi \cdot R \cdot 2R = 4\pi \cdot R^2 \quad (III.2.6)$$

Corolar. Aria sferei este de 4 ori aria unui cerc mare.

Observație. Raportul dintre ariile a două sfere este egal cu raportul pătratelor razelor.

➤ Volumul generat de un triunghi

Teorema III.5. Volumul generat de un triunghi care se rotește în jurul unei axe din planul său, care trece prin unul din vârfurile sale și nu-l traversează, are ca măsură produsul dintre aria descrisă de latura opusă vârfului situat pe axă și o treime din măsura înălțimii corespunzătoare.

Demonstrație. Dacă A este vârful situat pe axă, iar BC latura opusă, aceasta din urmă va genera, în rotația ei, un trunchi de con (care poate degenera, evident, în con dacă B sau C este pe axă, sau în cilindru, dacă BC este paralelă cu axa). Volumul V , care trebuie evaluat, este delimitat de suprafața laterală S a acestui trunchi de con și de suprafețele laterale ale celor două conuri generate respectiv de AB și de AC .

În trunchiul de con considerat, se înscrie un trunchi de piramidă regulată. În care $BB'C'C$, de exemplu (figura III.11) va fi o față și în care aria laterală S' va tinde către S , dacă numărul fețelor va crește nemărginit. ABC și $AB'C'$ vor fi două poziții succesive ale triunghiului care se rotește.

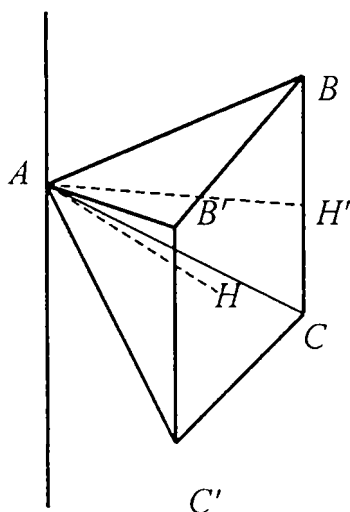


Fig.III.11. Volumul generat de un triunghi

Suma volumelor diverselor piramide, în general patrulate, având vârful A comun și ca baze respectiv trapezele (sau eventual triunghiurile sau dreptunghiurile) $BB'C'C$, dă un volum V' , care va tinde către volumul V căutat.

Însă volumul fiecărei piramide patrulate de felul lui $ABCC'B'$ are ca măsură produsul dintre aria bazei și o treime din înălțimea AH , astfel că suma lor va fi o treime din produsul ariei laterale S' cu valoarea comună a tuturor înălțimilor:

$$V' = S' \cdot \frac{1}{3} AH \quad (\text{III.2.7})$$

Pe de altă parte, triunghiurile ABC , $AB'C'$ sunt simetrice în raport cu planul bisector al diedrului format de planele lor; acest plan bisector, fiind un plan de simetrie pentru piramida patrulateră, va fi perpendicular pe planul bazei $BB'C'C$ și va conține deci înălțimea AH . Dacă numărul fețelor trunchiului de piramidă S' crește nemărginit, triunghiul $AB'C'$ va tinde către ABC , planul bisector va tinde către planul ABC , iar segmentul AH , care este mereu perpendicular pe BC , va tinde către înălțimea AH' a triunghiului dat.

Deci, volumul căutat:

$$V = S \cdot \frac{1}{3} AH' \quad (\text{III.2.8})$$

➤ Volumul sectorului sferic

Definiție. Se numește *sector sferic* figura generată de un sector circular care se rotește în jurul unui diametru care nu-l traversează. În această mișcare, arcul care este baza sectorului circular generează o zonă, numită *baza sectorului sferic*.

Pentru a defini volumul sectorului sferic, se înlocuiește sectorul circular printr-un sector poligonal înscris. Prin definiție, volumul sectorului sferic va fi limita către care tinde volumul generat de acest sector poligonal, dacă numărul de laturi ale liniei frânte care îi servește ca bază crește nemărginit, astfel ca fiecare dintre ele să tindă către zero.

Următoarele două teoreme arată existența acestei limite și dau expresia ei, independentă de legea după care se construiesc liniile frânte înscrise.

Teorema III.6. *Volumul generat de un sector poligonal care se rotește în jurul unei axe din planul său, ce trece prin centrul acestuia și nu-l traversează, este egal cu o treime din aria generată de linia frântă care servește ca bază sectorului, înmulțită cu o mărime cuprinsă între cea mai mare și cea mai mică dintre distanțele de la centru la laturile acestei linii.*

Demonstrație. Fie sectorul poligonal $OACDEB$ (figura III.12) care se rotește în jurul unei axe Ox ce trece prin centrul său O , axă ce este situată în planul său și nu-l întretaie.

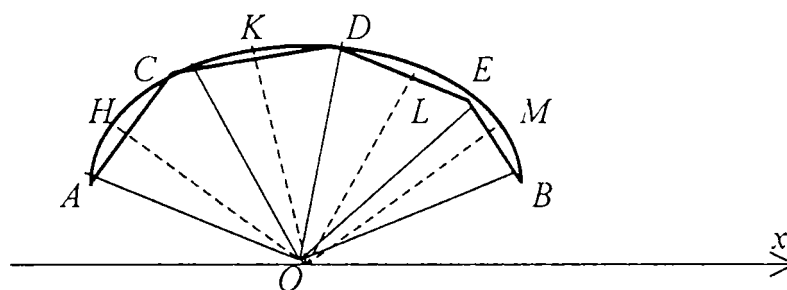


Fig.III.12. Volumul generat de un sector poligonal

Triunghiurile OAC , OCD , ODE , OEB generează fiecare volume a căror expresie este dată de teorema III.5. Notând cu OH , OK , OL , OM , distanțele de la laturile AC , DC , DE , EB la centrul O , rezultă:

$$\begin{aligned}
 V_{gener.\Delta OAC} &= \frac{1}{3} S_{gener.AC} \cdot OH \\
 V_{gener.\Delta OCD} &= \frac{1}{3} S_{gener.CD} \cdot OK \\
 V_{gener.\Delta ODE} &= \frac{1}{3} S_{gener.DE} \cdot OL \\
 V_{gener.\Delta OEB} &= \frac{1}{3} S_{gener.EB} \cdot OM
 \end{aligned}
 \tag{III.2.9}$$

Efectuând sumarea volumelor exprimate prin relațiile (III.2.9), se observă că volumul generat de sector este egal cu o treime aria generată de linia $ACDEB$,

înmulțită cu o cantitate cuprinsă între cea mai mare și cea mai mică dintre cantitățile OH, OK, OL, OM .

Corolar. Dacă sectorul poligonal este regulat, liniile OH, OK, OL, OM sunt toate egale cu apotema liniei frânte regulate $ACDEB$.

Așadar, volumul generat de un sector poligonal regulat care se rotește în jurul unei axe situate în planul său, ce trece prin centrul său și nu-l intersectează, este egal cu aria generată de linia frântă care este baza sectorului, înmulțită cu o treime din apotemă.

Teorema III.7. Volumul sectorului sferic este egal cu aria zonei care-i servește ca bază, înmulțită cu o treime din rază.

Demonstrație. Dacă numărul de laturi al unei linii frânte înscrise în arcul AB (figura III.12) care generează zona de bază, crește nemărginit și anume astfel ca fiecare dintre ele să tindă către zero, aria generată de această linie frântă înscrisă tinde către aria zonei, în timp ce distanțele la centru OH, OK, \dots ale diferitelor laturi tind toate către rază. Deci, volumul generat de sectorul poligonal respectiv tinde către limita menționată în enunț.

Corolar. Dacă R este raza sferei, h înălțimea zonei care este baza sectorului, atunci volumul acestuia este egal cu $\frac{2}{3} \pi \cdot R^2 \cdot h$.

Într-adevăr, această expresie este produsul dintre $R/3$ și aria zonei, obținută în teorema III.3.

➤ Volumul sferei

Teorema III.8. Volumul sferei cu raza R este $\frac{4}{3} \pi \cdot R^3$.

Demonstrație. Raționamentele anterioare sunt aplicabile figurii generate de un semicerc care se rotește în jurul diametrului său, adică volumului sferei. Trebuie atunci, în expresia indicată în corolarul teoremei III.7 să se înlocuiască înălțimea h prin $2R$; această operație va conduce la rezultatul enunțat.

Corolar. Volumul sferei cu diametrul D este $\frac{1}{6} \pi \cdot D^3$.

Observație. Volumele a două sfere sunt proporționale cu cuburile razelor lor (sau cu cuburile diametrelor lor).

➤ Volumul inelului sferic

Definiție. Se numește *inel sferic* solidul generat prin rotația unui segment de cerc în jurul unui diametru care nu-l traversează.

Teorema III.9. Volumul generat de un segment de cerc care se rotește în jurul unui diametru ce nu-l traversează (volumul inelului sferic) este a șasea parte din volumul cilindrului a cărui bază este coarda segmentului, înălțimea fiind proiecția acestei coarde pe axă.

Demonstrație. Fie segmentul de cerc, cuprins între arcul AB al cercului O (figura III.13) și coarda acestuia, care se rotește în jurul axei Ox ce trece prin O . Fie $A'B'$ proiecția lui AB pe Ox .

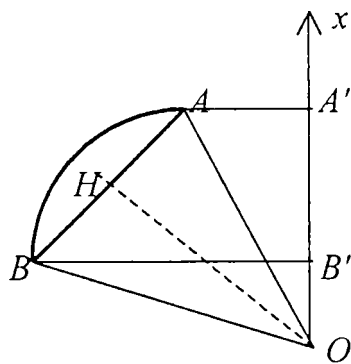


Fig.III.13. Generarea volumului unui inel sferic

Dacă figura se rotește în jurul lui Ox , sectorul de cerc OAB generează un volum având ca măsură:

$$V_{\text{gener.sect.}OAB} = \frac{2}{3} \pi \cdot OA^2 \cdot A'B' \tag{III.2.10}$$

Pe de altă parte, triunghiul OAB generează, în aceleași condiții, un volum egal cu:

$$V_{gener.\Delta OAB} = S_{gener.AB} \cdot \frac{1}{3} OH = 2\pi \cdot A'B' \cdot OH \cdot \frac{1}{3} OH = \frac{2}{3} \pi \cdot OH^2 \cdot A'B' \quad (\text{III.2.11})$$

Diferența dintre volumele (III.2.10) și (III.2.11) constituie, evident, volumul inelului sferic. Acesta are deci ca măsură:

$$V_{inelsferic} = \frac{2}{3} \pi \cdot OA^2 \cdot A'B' - \frac{2}{3} \pi \cdot OH^2 \cdot A'B' = \frac{2}{3} \pi \cdot A'B' (OA^2 - OH^2) \quad (\text{III.2.12})$$

Din triunghiul OAH se observă:

$$OA^2 - OH^2 = AH^2 = \frac{AB^2}{4} \quad (\text{III.2.13})$$

Deci:

$$V_{inelsferic} = \frac{2}{3} \pi \cdot A'B' \cdot \frac{AB^2}{4} = \frac{1}{6} \pi \cdot AB^2 \cdot A'B' \quad (\text{III.2.14})$$

➤ Volumul segmentului sferic

Definiție. Se numește *segment sferic* porțiunea de sferă cuprinsă între două plane paralele; acest volum este mărginit, în general, de o zonă și de două (interioare de) cercuri ale căror plane sunt paralele (figura III.8), cercurile fiind numite *bazele* segmentului. Segmentul sferic poate avea și o singură bază, adică să fie mărginit de un cerc și de o calotă sferică (figura III.14); în aceste condiții, planul celeilalte baze trebuie considerat ca fiind tangent la sferă. **Înălțimea** segmentului este distanța dintre planele bazelor.

Volumul segmentului sferic este egal cu semisuma volumelor celor doi cilindri care au ca înălțime comună înălțimea segmentului, iar ca baze, bazele respective ale segmentului, la care se adună volumul sferei având ca diametru înălțimea.

Fie segmentul sferic generat prin rotația trapezului mixtiliniu $A'B'B$ (figura III.15) în jurul axei $A'B'$ și mărginit, pe de o parte, de două cercuri având respectiv razele AA' și BB' , iar pe de altă parte, de zona pe care o generează arcul AB . Acest solid este, evident, suma trunchiului de con generat de trapezul rectiliniu $AA'B'B$ și a inelului sferic AB .

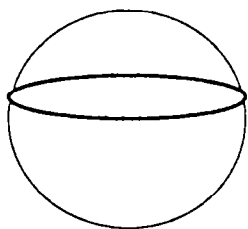


Fig.III.14. Segment sferic
cu o singură bază

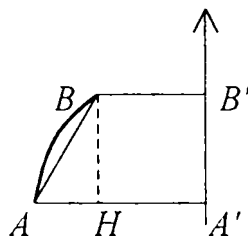


Fig.III.15. Generarea unui
segment sferic

Trunchiul de con are volumul:

$$V_{tr.con} = \frac{1}{3} \pi \cdot A' B' (AA'^2 + BB'^2 + AA' \cdot BB') \quad (III.2.15)$$

iar inelul sferic:

$$V_{inelsf.} = \frac{1}{6} \pi \cdot A' B' \cdot AB^2 \quad (III.2.16)$$

Suma va fi deci:

$$V_{segm.sf} = \frac{1}{6} \pi \cdot A' B' (AB^2 + 2 \cdot AA'^2 + 2 \cdot BB'^2 + 2 \cdot AA' \cdot BB') \quad (III.2.17)$$

Însă AB^2 se poate exprima cu ajutorul segmentelor AA' , BB' și $A'B'$. Astfel, din triunghiul dreptunghic ABH se obține:

$$AB^2 = AH^2 + BH^2 \quad (III.2.18)$$

iar:

$$AH = A'B' \quad BH = BB' - HB' = BB' - AA' \quad (III.2.19)$$

Înlocuind (III.2.19) în (III.2.18) și expresia obținută în volumul segmentului sferic, relația (III.2.17) devine:

$$\begin{aligned} V_{\text{segm.sf}} &= A'B'[A'B'^2 + (BB' - AA')^2 + 2 \cdot AA'^2 + 2 \cdot BB'^2 + 2 \cdot AA' \cdot BB'] = \\ &= A'B'(A'B'^2 + AA'^2 - 2 \cdot AA' \cdot BB' + BB'^2 + 2 \cdot A'B'^2 + 2 \cdot AA' \cdot BB' + 2 \cdot BB'^2) \end{aligned}$$

de unde, prin efectuarea reducerilor:

$$V_{\text{segm.sf}} = \frac{1}{6} \pi \cdot A'B'^3 + \frac{1}{2\pi} \cdot A'B'(AA'^2 + BB'^2) \quad (III.2.20)$$

ceea ce reprezintă efectiv suma volumelor sferei și a celor doi semicilindri indicați în enunț.

În particular, dacă A se află pe axă ($A \equiv A'$), atunci volumul segmentului este dat de:

$$V_{\text{segm.sf}} = \frac{1}{6} \pi \cdot A'B'^3 + \frac{1}{2\pi} \cdot A'B' \cdot BB'^2 \quad (III.2.21)$$

➤ Volumul intersecției a două sfere

Teorema III.10. *Volumul intersecției a două sfere incidente, având centrele la distanța d , se calculează prin relația (III.2.26), unde H și h sunt respectiv înălțimile segmentelor determinate pe cele două sfere de planul de intersecție, iar z este raza cercului de intersecție.*

Demonstrație. Fie date două sfere $S(O, R)$ și $s(o, r)$, $R > r$, pentru care distanța centrelor $d = |Oo|$ verifică relația: $R - r \leq d \leq R + r$. Volumul intersecției dintre cele două sfere este constituit din volumele a două segmente sferice, determinate de planul ce trece prin AA' și este perpendicular pe Oo .

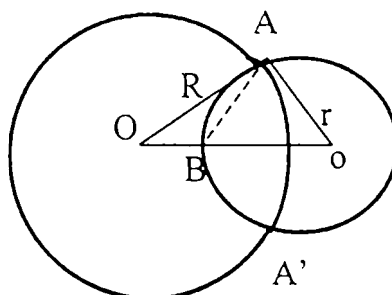


Fig.III.16. Volumul intersecției a două sfere

Fie x mărimea segmentului OB , y a segmentului Bo și z mărimea segmentului AB (figura III.16). Se pot scrie relațiile:

$$x + y = d \quad z^2 = R^2 - x^2 \quad z^2 = r^2 - y^2 \quad (\text{III.2.22})$$

de unde:

$$x = \frac{d}{2} + \frac{R^2 - r^2}{2d} \quad y = \frac{d}{2} - \frac{R^2 - r^2}{2d} \quad (\text{III.2.23})$$

Astfel, înălțimile segmentelor sunt:

$$H = R - \left(\frac{d}{2} + \frac{R^2 - r^2}{2d} \right) \quad h = r - \left(\frac{d}{2} - \frac{R^2 - r^2}{2d} \right) \quad (\text{III.2.24})$$

iar ultima din relațiile (III.2.22) devine:

$$z^2 = \frac{1}{2}(R^2 + r^2) - \frac{1}{4d^2}(R^2 - r^2) - \frac{d^2}{4} \quad (\text{III.2.25})$$

Volumul intersecției este:

$$V_{\text{inters.sf}} = \frac{1}{6}\pi(H^3 + h^3) + \frac{1}{2}\pi \cdot z^2(H + h) \quad (\text{III.2.26})$$

unde H și h sunt înălțimile segmentelor sferice iar z este mărimea segmentului AB .

Teorema III.11. *Volumul intersecției a două sfere de raze egale și cu centrele la distanța $d < 2R$, este egal cu volumul unui con de rază $R - d/2$ și înălțime $4R + d$.*

Demonstrație. În particular, dacă cele două sfere au razele egale, atunci volumul intersecției este:

$$V_{\text{inters.sf}} = \frac{1}{3}\pi \cdot H^3 + \pi \cdot H \cdot z^2 \quad (\text{III.2.27})$$

unde:

$$H = R - \frac{d}{2} \quad z^2 = R^2 - \left(\frac{d}{2}\right)^2 \quad (\text{III.2.28})$$

astfel încât volumul rezultă:

$$V_{\text{intres.sf}} = \frac{1}{3}\pi \left(R - \frac{d}{2}\right)^2 \cdot (4R + d) \quad (\text{III.2.29})$$

Dacă, în acest caz particular, se face precizarea că: $z \geq 0$, $-2r \leq x \leq 2r$, volumul (III.2.29) devine:

$$V(x) = V_{inters.sf} = \frac{1}{3}\pi\left(r - \frac{x}{2}\right)^2 + \pi\left(r - \frac{x}{2}\right) \cdot \left(r^2 - \frac{x^2}{4}\right) = \frac{\pi}{12}x^3 - 3r^2x + \frac{4\pi}{3}r^3 \quad (\text{III.2.30})$$

unde x este distanța dintre centre. Se observă că:

$$V(0) = \frac{4}{3}\pi \cdot r^3 \quad V(2r) = 0 \quad (\text{III.2.31})$$

Punctele de extrem ale graficului se vor obține prin anularea derivatei:

$$V'(x) = \frac{\pi}{4}x^2 - \pi \cdot r^2 = 0 \quad \Rightarrow \quad x = \pm 2r \quad (\text{III.2.32})$$

Derivata a doua:

$$V''(x) = \frac{\pi}{2}x \quad (\text{III.2.33})$$

este pozitivă pe intervalul $(0, 2r)$.

Graficul funcției este:

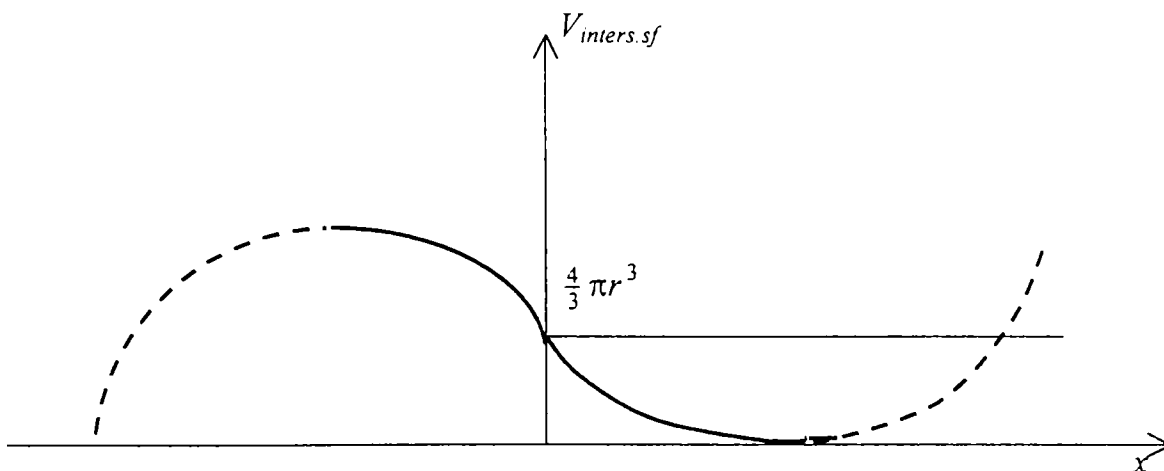


Fig.III.17. Reprezentarea grafică a volumului intersecției a două sfere

III.2.2.2. Reprezentarea exterioară a obiectelor prin sfere

În scopul concretizării modelelor de aproximare prin sfere a formei exterioare a corpurilor, se va trata matematic exact reprezentarea exterioară a unor corpuri de formă geometrică regulată (cilindrul, conul și trunchiul de con), rezultatele obținute stând la baza stabilirii algoritmilor de reprezentare și de realizare a programelor de modelare a corpurilor și sistemelor de corpuri cu care operează robotica și planificarea mișcărilor.

➤ Acoperirea unui cilindru cu sfere

Se urmărește realizarea unei *acoperiri optime* a unui cilindru cu sfere, astfel încât numărul sferelor de acoperire să nu fie prea mare iar diferența dintre volumul acoperirii și cel al cilindrului să fie suficient de mică. Acest deziderat se poate realiza prin sfere centrate pe axa cilindrului și cu raze egale a căror utilizare este evident avantajoasă din punct de vedere al algoritmilor de acoperire [**bibliogr. Referat oana**]

Teorema III.12. *Fie un cilindru circular drept, de rază r și înălțime h cu proprietatea că $h > 2r$. Un astfel de cilindru poate fi acoperit cu o sferă de rază*

$$R_1 = \sqrt{r^2 + \frac{h^2}{4}} \text{ și cu centrul în centrul de simetrie al cilindrului.}$$

Teorema III.13. *Un cilindru de rază r și înălțime h ($h \geq 2r$) poate fi acoperit cu două sfere (cele mai mici) de raze $R_2 = \sqrt{r^2 + \frac{1}{2^2} \cdot \frac{h^2}{4}}$ cu centrele pe axa cilindrului și la înălțime $h/4$ și respectiv $3 \cdot h/4$, astfel încât suma volumelor inelelor sferice determinate să fie minimă.*

Demonstrație. Fie cilindrul de rază r și înălțime h (figura III.18) și o pereche de sfere $S_1(O_1, R_1)$ și $S_2(O_2, R_2)$, cu centrele pe axa de simetrie a cilindrului. Volumul ocupat de cele două sfere se compune din patru segmente sferice, dintre care două, cele născute prin rotirea arcelor AB și CD , formează două volume acoperitoare ale capetelor, iar alte două, cele generate respectiv prin rotația arcelor BF și FC , care

acoperă partea laterală a cilindrului. Suma acestor două ultime volume este minimă, simultan cu suma volumelor sectoarelor sferice generate de arcele BF și FC , precum și cu suma volumelor celor două inele sferice generate prin aceleași arce, întrucât oricare ar fi sferile acoperitoare, care se intersectează după un cerc situat pe cilindru, ele determină fie patru conuri cu bazele cercuri de rază r (raza cilindrului) în cazul sectoarelor, fie însăși cilindrul, în cazul inelelor sferice.

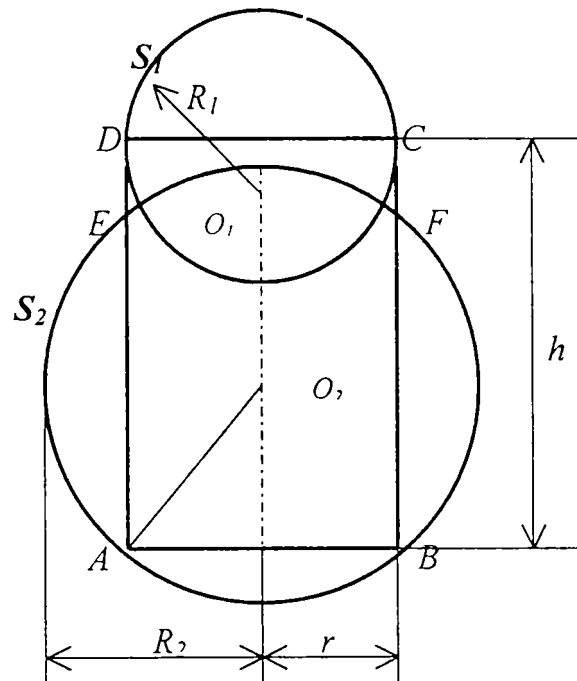


Fig.III.18. Acoperirea unui cilindru cu două sfere

În această din urmă metodă suma volumelor este dată de:

$$\sum V = \frac{\pi}{6}(x^3 + y^3) = \frac{\pi}{6} \left[x^3 + \left(\frac{h}{2} - x \right)^3 \right] \quad (\text{III.2.33})$$

și este minimă, dacă funcția:

$$f(x) = \frac{3}{2}h \cdot x^2 + \frac{3}{4}h^2 \cdot x - \frac{h^3}{8} \quad (\text{III.2.34})$$

este minimă, adică atunci când derivata sa:

$$(\text{III.2.35})$$

$$f'(x) = 3h \cdot x + \frac{3}{4} h^2 = 0$$

Aceasta se întâmplă dacă $x = y = h/4$, ceea ce demonstrează afirmația.

Volumul primelor două segmente sferice este dat, conform relației (III.2.26), de:

$$V_{\text{segm.sf}} = \frac{\pi}{3} (R_2 - x)^3 + \pi (R_2 - x) \cdot r^2 \quad (\text{III.2.36})$$

Mai general, are loc:

Teorema III.14. *Dacă un cilindru este acoperit cu n sfere, atunci cele n inele sferice care acoperă partea laterală au suma volumelor minimă, dacă sferile au raze*

$$\text{egale între ele și egale cu } R_n = \sqrt{r^2 + \frac{1}{n^2} \cdot \frac{h^2}{4}}$$

Demonstrație. Se face prin inducție matematică.

- În cazul $n = 2$, afirmația este adevărată, conform teoremei III.13.
- Se presupune că afirmația este adevărată în cazul a n sfere acoperitoare.
- Se presupune cilindrul acoperit cu $n+1$ sfere, dintre care primele n , conform ipotezei b), determină sectoare sferice care se proiectează respectiv pe axa cilindrului după segmente de lungime $2x$, al $(n+1)$ -lea după un segment de lungime $2y$.

Suma volumelor celor $n+1$ sectoare este:

$$V_{(n+1)\text{sect}} = n \cdot \frac{\pi}{3} \cdot x^3 + \frac{\pi}{3} \left(\frac{h}{2} - n \cdot x \right)^3 \quad (\text{III.2.37})$$

și este minimă dacă funcția:

$$f(x) = n(1 - n^2) \cdot x^3 + \frac{3}{2} n^2 \cdot h \cdot x^2 - \frac{3}{4} n \cdot h^2 \cdot x + \frac{1}{8} h^3 \quad (\text{III.2.38})$$

este minimă, ceea ce se întâmplă când derivata sa:

$$f'(x) = 3n(1 - n^2) \cdot x^2 + 3n^2 \cdot h \cdot x - \frac{3}{4}n \cdot h^2 = 0 \quad (\text{III.2.43})$$

de unde:

$$x = y = \frac{h}{2(n+1)} \quad (\text{III.2.40})$$

rezultat care demonstrează teorema.

Observație. În general pot fi considerate n sfere, de raze $R_n = \sqrt{r^2 + \frac{1}{n^2} \cdot \frac{h^2}{4}}$,

cu centrele pe axa cilindrului, respectiv la înălțimile $(2p+1) \cdot \frac{h}{2n}$, ($p=0, n-1$).

Printr-o astfel de acoperire interiorul cilindrului este cuprins în reuniunea interioarelor celor n sfere.

Definiție. Se numește numi *defect exterior (de acoperire)* și se va nota prin Def_{ext} diferența dintre volumul ocupat de cele n sfere și volumul cilindrului.

Expresia sa este:

$$Def_{ext} = \frac{4}{3}n \cdot \pi \cdot R_n^3 - (n-1) \cdot V_{int} - \pi \cdot r^2 \cdot h \quad (\text{III.2.41})$$

unde:

$$V_{int} = \frac{1}{3}\pi \left(R_n - \frac{d_n}{2} \right)^2 \cdot (4R_n + d_n) \quad (\text{III.2.42})$$

cu $d_n = h/n$.

Def_{ext} poate fi considerat ca fiind constituit din două componente, una corespunzătoare bazelor și alta părții laterale, notate respectiv cu: Def_{ext}^b și Def_{ext}^l . Acestea sunt definite prin:

$$Def_{ext}^b = V_{int} = \frac{1}{3} \pi \left(R_n - \frac{d_n}{2} \right)^2 \cdot (4R_n + d_n) \quad (\text{III.2.43})$$

și respectiv:

$$Def_{ext}^l = \frac{4}{3} n \cdot \pi \cdot R_n^3 - n \cdot V_{int} - \pi \cdot r^2 \cdot h \quad (\text{III.2.44})$$

Def_{ext}^b crește odată cu numărul sferelor și tinde către volumul uneia dintre sfere, dacă numărul n al acestora tinde către infinit.

Def_{ext}^l scade odată cu creșterea numărului sferelor și tinde către zero.

O problemă care se pune este de a determina numărul n astfel încât Def_{ext}^l să fie inferior unei anumite mărimi dinainte date.

Pentru aceasta, se calculează efectiv mărimea Def_{ext}^l :

$$Def_{ext}^l = \frac{4}{3} n \cdot \pi \cdot R_n^3 - n \cdot V_{int} - \pi \cdot r^2 \cdot h = \pi \cdot h \cdot \left(\frac{1}{n^2} \cdot \frac{h^2}{4} + \frac{1}{12} \cdot \frac{h^2}{n^2} \right) = \frac{\pi \cdot h^3}{3n^2} \quad (\text{III.2.45})$$

Considerând un ε dat și impunând $Def_{ext}^l \leq \varepsilon$ rezultă:

$$\frac{\pi \cdot h^3}{3n^2} \leq \varepsilon \quad \Rightarrow \quad n \geq \sqrt{\frac{\pi \cdot h^3}{3\varepsilon}} \quad (\text{III.2.46})$$

Volumele segmentelor sferice de acoperire a capetelor se exprimă prin relația (III.2.36).

Alături de șirul razelor $\{R_n\}_{n \in \mathbb{N}}$, cu $\lim_{n \rightarrow \infty} R_n \rightarrow r$, se construiește șirul volumelor

$$Def_{ext}^l, \text{ care este } \left\{ \frac{\pi \cdot h^3}{6} \cdot \frac{1}{n^2} \right\}.$$

Fie funcția $V(x) = \frac{\pi \cdot h^3}{6} \cdot \frac{1}{x^2}$ $x \in [1, \infty)$ a cărei reprezentare este dată în

figura III.19.

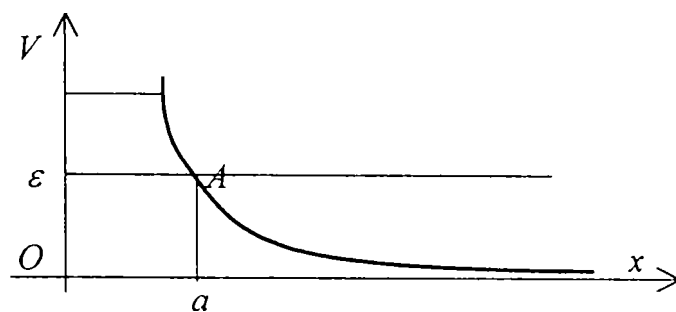


Fig.III.19. Reprezentarea funcției $V(x)$

Dacă se dă $\varepsilon \in \left(0, \frac{\pi \cdot h^3}{6}\right)$ și se construiește paralela la axa Ox , la distanța ε ,

aceasta intersectează curba $V_x=V(x)$ în punctul A , de abscisă a . Cel mai mic număr natural mai mare decât a este numărul de sfere necesar ca defectul să fie inferior lui ε .

Volumul segmentelor de acoperire crește de la V_1 la $V_\infty = \frac{4}{3} \pi \cdot r^3$.

➤ Acoperirea unui con cu sfere

❖ Sfera circumscrișă conului

Fie un con circular drept, cu cercul de bază de rază r și cu înălțimea h , (figurile III.19 și III.20).

Se consideră sfera circumscrișă conului, ce trece prin cercul de bază și prin vârf. Există două situații distincte în ceea ce privește posibilitatea de acoperire a conului cu o sferă depinzând de relația dintre h , a , r și R .

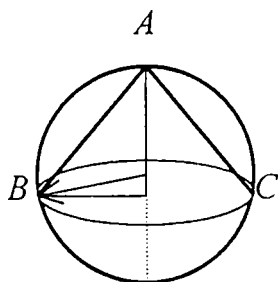


Fig.III.20. Sfera circumscrișă conului circular drept cu $h \geq r$

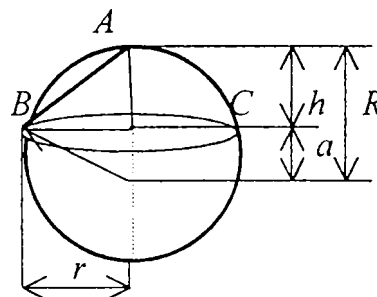


Fig.III.21. Sfera circumscrișă conului circular drept cu $h < r$

În cazul $h \geq r$, relațiile între elementele R și a , asociate sferei și, respectiv r și h asociate conului, sunt:

$$R = \sqrt{a^2 + r^2} \quad R = h - a \quad (\text{III.2.47})$$

de unde rezultă:

$$a = \frac{h^2 - r^2}{2h} \quad R = \frac{h^2 + r^2}{2h} \quad (\text{III.2.48})$$

În cazul $h \leq r$, atunci:

$$R = a + h \quad a = \frac{r^2 - h^2}{2h} \quad R = \frac{h^2 + r^2}{2h} \quad (\text{III.2.49})$$

În ambele cazuri:

$$a = |R - h| \quad a = \frac{|h^2 - r^2|}{2h} \quad (\text{III.2.50})$$

Pentru volume, sunt valabile relațiile:

$$V_{con} = \frac{1}{3} \pi \cdot r^2 \cdot h \quad V_{sfera} = \frac{4}{3} \pi \cdot R^3 = \frac{4}{3} \pi \cdot \left(\frac{h^2 + r^2}{2h} \right)^3 \quad (\text{III.2.51})$$

Dacă se calculează mărimea segmentului $R - a$, conform relațiilor (III.2.48), ea rezultă:

$$R - a = \frac{r^2}{h} \quad (\text{III.2.52})$$

Defectul de acoperire, ca diferență dintre volumul sferei acoperitoare și volumul conului acoperit, indicate în relațiile (III.2.51) este:

$$Def_{ext} = \frac{4}{3} \pi \cdot \left(\frac{h^2 + r^2}{2h} \right)^3 - \frac{1}{3} \pi \cdot r^2 \cdot h \quad (III.2.53)$$

și se compune din două părți, un defect bazic și unul lateral:

$$Def_{ext}^b = \frac{\pi \cdot r^4}{3h^3} (r^2 + 3h^2) \Rightarrow Def_{ext}^l = Def_{ext} - Def_{ext}^b \quad (III.2.54)$$

❖ Sferă circumscrisă unui trunchi de con

Fie trunchiul de con, cu raza bazei mari r_1 , raza bazei mici r_2 și înălțimea h . Se notează cu a_1 și a_2 distanțele de la baza mare, respectiv baza mică la centrul sferei.

Se pot scrie relațiile:

$$(III.2.55)$$

$$R^2 = a_1^2 + r_1^2 \quad R^2 = a_2^2 + r_2^2 \quad a_2 \pm a_1 = h$$

În ultima dintre relațiile (III.2.55) se consideră semnul + dacă centrul sferei este între planele de bază (figura III.22) și – dacă este în afară (figura III.23).

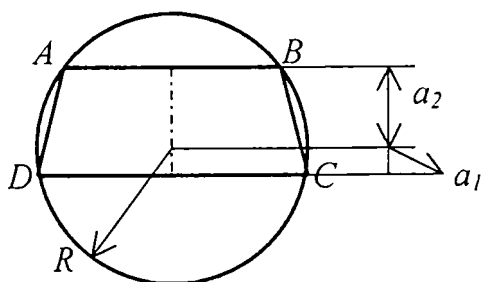


Fig.III.22. Sferă circumscrisă trunchiului de con ce conține centrul sferei

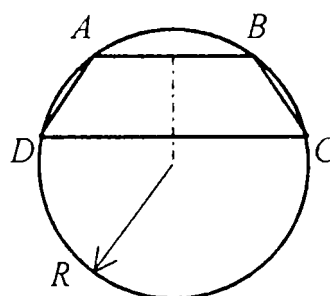


Fig.III.23. Sferă circumscrisă trunchiului de con ce nu conține centrul sferei

Din relațiile (III.2.55) rezultă:

$$a_1 = \pm \frac{h}{2} \pm \frac{r_1^2 - r_2^2}{2h} \quad a_2 = \frac{h}{2} + \frac{r_1^2 - r_2^2}{2h} \quad R^2 = r_1^2 + \left(\frac{h}{2} - \frac{r_1^2 - r_2^2}{2h} \right)^2 \quad (\text{III.2.56})$$

Volumul trunchiului de con este:

$$V_{tr\ con} = \frac{1}{3} \pi \cdot h \cdot (r_1^2 + r_2^2 + r_1 \cdot r_2) \quad (\text{III.2.57})$$

Defectul de acoperire se calculează similar cu cazul acoperirii conului:

$$Def_{ext} = V_{sfera} - V_{tr\ con} = \frac{4}{3} \pi \cdot R^3 - \frac{1}{3} \pi \cdot h \cdot (r_1^2 + r_2^2 + r_1 \cdot r_2) \quad (\text{III.2.58})$$

Se disting cele două cazuri posibile:

- Centrul sferei este situat între planele de bază (figura III.22), situație în care în relațiile (III.2.56) la parametrul a_1 apare primul termen cu semnul + și al doilea cu semnul -;
- Centrul sferei este situat în afara planelor de bază (figura III.23), situație în care în relațiile (III.2.56) la parametrul a_1 apare primul termen cu semnul - și al doilea termen cu semnul +.

Observație.

În cazul a.: Sfera minimă care acoperă trunchiul de con este sfera circumscrisă trunchiului. Diferența dintre volumul sferei și volumul trunchiului este defectul total. Volumul inelului sferic determinat de arcul AD este defectul lateral exterior iar suma volumelor segmentelor determinate de AB și CD este defectul bazic exterior.

În cazul b.: Sfera cea mai mică ce acoperă trunchiul este sfera de diametru CD . Defectul total exterior însă este cel mai mic atunci când sfera este circumscrisă trunchiului (ca și în cazul a.). Aceasta este situația care interesează în mod deosebit.

Defectul lateral este dat de volumul inelului sferic determinat de AD :

$$Def_{ext}^l = \frac{1}{6} \pi \cdot h \cdot [H^2 + (r_2 - r_1)^2] \quad (III.2.59)$$

iar raza sferei este exprimată prin ultima din relațiile (III.2.56).

Ținând seama de relația (III.2.58) în care se introduce expresia razei sferei, se obține defectul total:

$$Def_{ext} = \frac{4}{3} \pi \cdot \left[\left(\frac{h}{2} - \frac{r_1^2 - r_2^2}{2h} \right)^2 + r_1^2 \right]^{\frac{3}{2}} - \frac{1}{3} \pi \cdot h \cdot (r_1^2 + r_2^2 + r_1 \cdot r_2) \quad (III.2.60)$$

Defectul bazic, exprimat ca diferență dintre defectul total și cel lateral se obține prin utilizarea relațiilor (III.2.60) și (III.2.59):

$$Def_{ext}^b = \frac{4}{3} \pi \cdot R^3 - \frac{1}{2} \pi \cdot h \cdot (r_1^2 + r_2^2 + \frac{h^2}{3}) \quad (III.2.61)$$

❖ Trunchiul de con acoperit cu două sfere

Se consideră trunchiul de con de înălțime h cu raza bazei mari r_1 și cea a bazei mari r_2 acoperit prin două sfere incidente, una trecând prin cercul bazei mari și cealaltă prin cercul bazei mici. Ele se intersectează după un cerc de rază r situat pe con și paralel cu cercurile de bază (figura III.24).

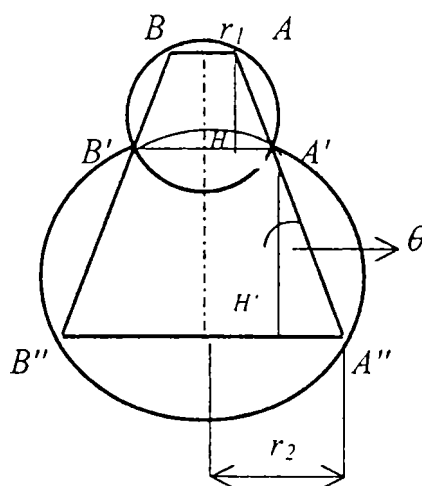


Fig.III.24. Trunchi de con
acoperit cu două sfere

Trunchiul de con dat se descompune în două trunchiuri de con, unul inferior de înălțime x și celălalt superior de înălțime y (evident că $x+y=h$).

Se pune problema determinării mărimii razei r în funcție de x . Ținând seama de notațiile menționate și de notațiile din figura III.24 se observă:

$$r = r_1 - (r_1 - r_2) \cdot \frac{x}{h} \quad (III.2.62)$$

cu respectarea relațiilor:

$$\frac{y}{x} = \frac{AA'}{A'A''} \quad AH = AA' \cdot \cos \theta \quad A'H' = A'A'' \cdot \cos \theta$$

deducându-se că volumele inelelor sferice sunt proporționale cu x^2 respectiv y^2 , suma lor fiind:

$$\sum V_{inel.sferic} = \frac{\pi}{6} (x^3 + y^3) \cdot \cos \theta \quad (III.2.63)$$

relație ce indică faptul că ea este minimă dacă $x = y$. Observația conduce la cazul cilindrului, care este de fapt un caz particular de trunchi de con.

Dacă se calculează acum defectul exterior lateral prin particularizarea relației (III.2.59) la cazul analizat, rezultă:

$$Def_{ext}^l = \frac{\pi}{2h^2} h \cdot (h^2 + (r_1 - r_2)^2) (x^2 - hx + \frac{1}{3} h^2) \quad (III.2.64)$$

care exprimat doar ca funcție de x devine:

$$f(x) = K \cdot (x^2 - hx + \frac{1}{3} h^2) \quad (III.2.65)$$

al cărei minim se obține din condiția de extrem:

$$f'(x) = K \cdot (2x - h) = 0 \quad \Rightarrow \quad x = \frac{h}{2} \quad (\text{III.2.66})$$

Deci, defectul exterior lateral este minim dacă cele două trunchiuri de con au înălțimi egale cu jumătate din înălțimea inițială a trunchiului de con, situație în care raza cercului lor de intersecție se obține prin particularizarea relației (III.2.62) și are, evident, valoarea:

$$r = r_1 - (r_1 - r_2) \cdot \frac{x}{h} = \frac{r_1 + r_2}{2} \quad (\text{III.2.67})$$

❖ Conul acoperit cu două sfere

Se consideră conul acoperit cu două sfere incidente, una trecând prin cercul de bază și alta prin vârf. Ele se intersectează după un cerc situat pe con și paralel cu cercul de bază (figura III.25).

Se notează raza cercului de bază cu r_1 și pe cea a acercului de intersecție cu r_2 .

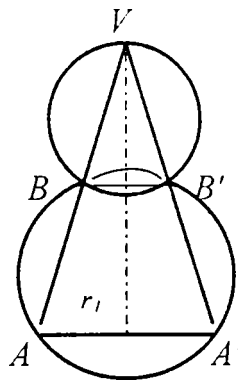


Fig.III.25. Acoperirea unui con
cu două sfere

Dacă R_1 și R_2 sunt razele sferelor de acoperire și h_1 , h_2 înălțimile segmentelor, atunci defectul de acoperire este:

$$Def_{ext} = \frac{4}{3}\pi \cdot (R_1^3 + R_2^3) - \left[\frac{\pi}{6}(h_1^2 + h_2^2) + \frac{\pi}{2}(h_1 + h_2) \cdot r_2^2 \right] - \frac{\pi}{3}r^2 \cdot h \quad (\text{III.2.68})$$

S-a obținut un caz particular de acoperire a unui trunchi de con în care baza superioară se reduce la un punct.

Observație finală

În cazul trunchiului de con, acoperit prin cel mai mic număr n de sfere, defectul exterior lateral minim este obținut din particularizarea relației (III.2.59) prin luarea în considerare a unghiului θ :

$$Def_{ext.l.r.con}^l = \frac{\pi}{6 \cdot \cos^2 \theta} \cdot h^3 \cdot \frac{1}{n^2} \quad (III.2.69)$$

rezultat care permite calcularea numărului n de sfere de acoperire necesar astfel încât să fie inferior unui număr ε dinainte dat. Relația este valabilă prin particularizare atât pentru con cât și pentru cilindru, deci are un caracter general.

III.2.2.3. Reprezentarea interioară a obiectelor prin sfere

Definiție. Se numește numi *defect interior (de umplere)* și se va nota prin Def_{int} diferența dintre volumul ocupat de cele n sfere de umplere și volumul corpului de reprezentat.

Se vor folosi toate expresiile utile determinate la paragraful III.2.2.2.

➤ Reprezentarea interioară a cilindrului prin sfere (Sfere înfășurate în cilindru)

Față de reprezentarea exterioară, reprezentarea interioară prezintă unele elemente noi, punându-se problema reprezentării interioare a unui cilindru circular drept, de rază r și înălțime h , printr-un număr de n sfere și determinarea relației între numărul de sfere și defectul interior, cu condiția ca acesta să fie minim.

Condiții de existență:

1. Raza r a sferei să fie egală cu raza cilindrului. Pentru realizarea unei condiții de minim a funcției Def_{int} , se presupune $R = r$, astfel încât se

elimină o mare clasă de cazuri, neinteresante; cu respectarea acestei condiții cele două volume sunt:

$$V_{cil} = \pi \cdot r^2 \cdot h \quad V_{sfera} = \frac{4}{3} \pi \cdot r^3 \quad (\text{III.2.70})$$

2. Înălțimea h a cilindrului să verifice inegalitatea $h \geq 2r$, iar sferile să fie disjuncte, astfel încât diferența de volume este, în cazul $n=1$:

$$V_{cil} - V_{sfera} = \frac{\pi \cdot r^3}{3} (3h - 4r) \geq 0 \quad (\text{III.2.71})$$

În cazul limită $h = 2r$, defectul interior:

$$Def_{int} = 2\pi \cdot r^3 - \frac{4}{3} \pi \cdot r^3 = \frac{2}{3} \pi \cdot r^3 = \frac{1}{2} V_{sfera} \quad \Rightarrow \quad 2V_{cil} = 3V_{sfera} \quad (\text{III.2.72})$$

Dacă h este suficient de mare, astfel încât n sfere să încapă în cilindru, fără să se întâlnească, evident $h \geq 2nr$. În acest caz defectul interior:

$$Def_{int} = V_{cil} - n \cdot V_{sfera} = \frac{1}{3} \pi \cdot r^3 (3h - 4n \cdot r) \geq \frac{2}{3} n \cdot \pi \cdot r^3 \quad (\text{III.2.73})$$

În general:

$$V_{cil} \geq \frac{n(2n+1)}{2} \cdot V_{sfera} \quad (\text{III.2.74})$$

În cazul $h = 2nr$, se realizează egalitatea:

$$Def_{int} = \frac{n}{2} \cdot V_{sfera} \quad (\text{III.2.75})$$

Dacă se realizează situația $2nr \leq h < 2(n+1)r$, atunci introducerea celei de-a $(n+1)$ -a sfere obligă ca cel puțin două dintre ele să fie incidente.

Se presupune că n este numărul maxim de sfere posibile din cilindru, cu intersecția a două câte două formată din cel mult un punct, adică cu respectarea condiției precizate anterior: $2nr \leq h < (2n+1)r$.

Pentru studiul acestor cazuri, este necesară demonstrarea următoarei teoreme.

Teorema III.15. *Fiind date două sfere cu centre la distanță $d < 4r$, o a treia sferă, plasată între cele două, determină defectul minim dacă centrul său este în mijlocul segmentului determinat de centrele celor două sfere.*

Demonstrație. Fie situația din figura III.27. Se notează cu Σ spațiul ce marchează suma defectelor interioare și care se obține reconsiderând expresia (III.2.33) și cocorbând-o cu (III.2.59) și (III.2.61), adică urmând să se determine minimul funcției:

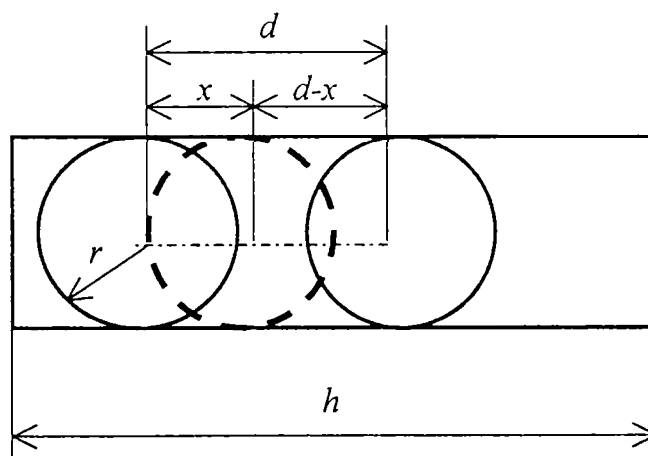


Fig.III.27. 3 sfere înfășurate de un cilindru

$$\begin{aligned} \Sigma = f(x) = & \left(2r^3 - \frac{3}{2}r^2 \cdot d + \frac{3}{4}r \cdot d^2 - \frac{d^3}{8} \right) + \frac{3d}{2} \cdot x \cdot \left(\frac{d}{4} - r \right) + \frac{3x^2}{2} \left(r - \frac{d}{4} \right) + \\ & + 3\sqrt{r^2 - x^2} \left(r - \frac{x}{2} \right) + 3\sqrt{r^2 - (x-d)^2} \left(r - \frac{d}{2} + \frac{x}{2} \right) \end{aligned} \quad (\text{III.2.76})$$

Derivata funcției (III.2.76) trebuie să fie nulă pentru ca defectul interior să fie minim:

$$f'(x) = 3\left(\frac{d}{4} - r\right) \cdot \left(\frac{d}{2} - x\right) + \frac{3}{2}(\sqrt{r^2 - (x-d)^2} - \sqrt{r^2 - x^2}) + \frac{3\left(r - \frac{d-x}{2}\right) \cdot (d-x) - 3x \cdot \left(r - \frac{x}{2}\right)}{\sqrt{r^2 - (x-d)^2} - \sqrt{r^2 - x^2}} = 0 \quad (\text{III.2.77})$$

care, evident, se anulează pentru $x = d/2$, ceea ce demonstrează teorema.

Un rezultat corespunzător celui de la reprezentarea exterioară a cilindriului cu sfere, ne spune că dacă un cilindru circular drept este umplut cu n sfere ($h < 2nr$), acestea determină un defect minim dacă distanța centrelor a două sfere consecutive este aceeași, oricare ar fi aceste sfere (observație confirmată și de teorema III.15).

Se consideră acum cazul cel mai important pentru aplicații, al cilindriului de rază r și înălțime h suficient de mare, care va fi reprezentat interior prin n sfere identice. Se consideră deci n sfere, care vor umple cilindrul, cele de la capete fiind tangente bazelor și întregul ansamblu având centrele distanțate cu pas constant d . Are loc:

Teorema III.16. Fie dat un cilindru circular drept, de rază r și înălțime h , volumul maxim ocupat de n sfere de rază r , interioare cilindriului, ($2r < h$), este dat prin relația:

$$V_{\text{ocup.max}} = \frac{4}{3}n \cdot \pi \cdot r^3 - (n-1) \left[\frac{1}{6} \pi \cdot \left(r - \frac{d}{2}\right)^3 + \frac{1}{2} \pi \cdot \left(r - \frac{d}{2}\right) \cdot \rho_n^2 \right]$$

Defectul interior de reprezentare a unui cilindru cu n sfere este minim dacă sferile sunt echivalente.

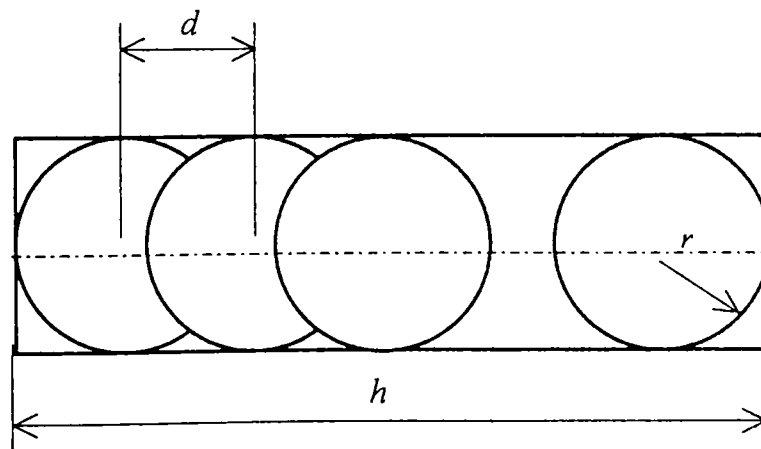


Fig.III.27. n sfere înfășurate într-un cilindru

Demonstrație. Se consideră n sfere interioare, dintre care două sunt tangente bazelor cilindrului, și având centrele distanțate cu un pas constant d (figura III.27).

Defectul său interior, Def_{int} este exprimat prin:

$$Def_{int} = V_{cil} - V_{ocup. max.} =$$

$$= \pi \cdot r^2 \cdot h - \frac{4}{3} n \cdot \pi \cdot r^3 - (n-1) \left[\frac{1}{6} \pi \cdot \left(r - \frac{d}{2} \right)^3 + \frac{1}{2} \pi \cdot \left(r - \frac{d}{2} \right) \cdot \rho_n^2 \right] \quad (III.2.78)$$

în care:

$$d = \frac{h-2r}{n-1} \quad \rho_n^2 = r^2 - \left(\frac{d}{2} \right)^2 = \frac{1}{2n-1} \sqrt{(2n \cdot r - h) \cdot (2n \cdot r - h - 4r)} \quad (III.2.79)$$

cu: $\lim_{n \rightarrow \infty} \rho_n = r$.

Similar cazului defectelor exterioare, în structura defectului interior (de umplere) apar două componente:

- defectul bazic: $Def_{int}^b = \frac{2}{3} \pi \cdot r^3$ constant în raport cu numărul n al sferelor;
- defectul lateral:

$$Def_{int}^l = \pi \cdot r^2 \cdot h - \frac{2(2n+1)}{3} \pi \cdot r^3 + \frac{n-1}{2} \pi \cdot \left(r - \frac{d}{2} \right) \cdot \left[\frac{1}{3} \left(r - \frac{d}{2} \right)^2 + \rho_n^2 \right].$$

Fiind dat un număr real ε , se poate determina un număr n de sfere, astfel încât Def_{int}^l să fie inferior lui ε .

➤ Reprezentarea interioară a unui cilindru fără baze

În multe situații este recomandabilă reprezentarea unui cilindru circular drept în așa fel încât defectul de baze, care este invariabil și mare, să nu apară. În acest scop, se consideră cilindrul de rază r și înălțime h , în capetele sale înscriindu-se câte o semisferă (Figura III.28).

Defectul de umplere, care acum este doar lateral și reprezintă volumul rămas neocupat, este:

$$Def_{int}^I = V_{rest} = \pi \cdot r^2 \cdot h - \frac{4}{3} \pi \cdot r^3 \quad (III.2.80)$$

Este important să se studieze variația defectului de volum în funcție de distanța dintre centrele unor sfere interioare și centrul unei semisfere de la capete.

Se notează cu x distanța dintre centrul uneia din semisferele de la baza cilindriului și centrul unei alte sfere înscrise în cilindru.

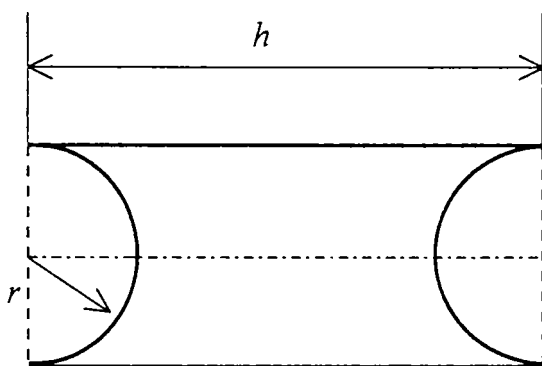


Fig. III.28. Cilindru cu două semisfere înscrise interior la capete

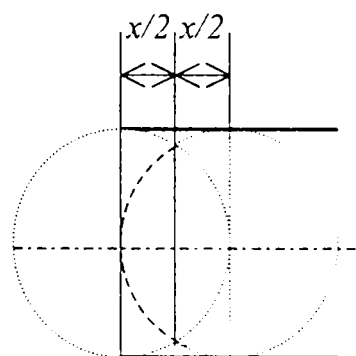


Fig. III.29. Intersecție nevidă dintre semisferă și sferă

Se consideră deci, o altă sferă (de rază r) înscrisă în cilindru, în următoarele situații:

a) $0 \leq x \leq 2r$

Cele două sfere se intersectează după un cerc (eventual de rază nulă) situat într-un plan la distanța $x/2$ de fiecare centru (Figura III.29).

Dacă se notează:

- Volumul sferei: $V_{sfera} = \frac{4}{3} \pi \cdot r^3$

- Volumul intersecției celor două sfere: $V_{inters} = \frac{1}{3} \pi \cdot H^3 + \pi \cdot H \cdot z^2$

unde: $H = r - \frac{x}{2}$ $z = \sqrt{r^2 - \frac{x^2}{4}}$

- Volumul semilunii determinată de cele două sfere: $V_{sl} = V_{sfera} - V_{inters}$

Se presupune h suficient de mare ($h \geq 4r$) și se impun: condiția de existență a intersecției $z \geq 0$, adică $x^2 \leq 4r^2$ ($-2r \leq x \leq 2r$) și condiția neinscriptibilitate a sferei în spațiul cilindrului (mai puțin cele două sfere) $x \geq 0$.

Calculul volumului intersecției conduce la:

$$V_{\text{inters}} = \frac{\pi}{12}(x^3 - 12r^2 \cdot x) + \frac{4}{3}\pi \cdot r^3 \quad (\text{III.2.81})$$

astfel încât volumul semilunii:

$$V_{sl} = \pi \cdot x \cdot \left(r^2 - \frac{x^2}{12} \right) \quad (\text{III.2.82})$$

ale cărui valori sunt:

$$V_{sl}(0) = 0 \quad V_{sl}(r) = \frac{11}{12}\pi \cdot r^3 \quad V_{sl}(2r) = \frac{4}{3}\pi \cdot r^3 \quad (\text{III.2.83})$$

Defectul de umplere lateral al reprezentării interioare, ca diferență între volumul cilindrului și cele ale semisferelor și semilunii, este:

$$Def_{\text{int}}^l = \pi \cdot r^2 \cdot h - \pi \cdot x \cdot \left(\frac{x^2}{12} - r^2 \right) \quad (\text{III.2.84})$$

b) $2r \leq x \leq h-2r$:

$$Def_{\text{int}}^l = V_{\text{cil}} - 2V_{\text{sfera}} = \pi \cdot r^2 \cdot h - \frac{8}{3}\pi \cdot r^3 = \text{const.} \quad (\text{III.2.85})$$

c) $h-2r \leq x \leq h$: volumul semilunii scade într-o manieră similară cu creșterea de la varianta a).

Reprezentarea grafică a variației defectului reprezentării interioare a cilindrului în funcție de x , ținând seama de diferitele variante studiate anterior pentru $x \in [0, h]$, conform relațiilor (III.2.80), (III.2.84) și (III.2.85), conduce la curba din figura III.30.

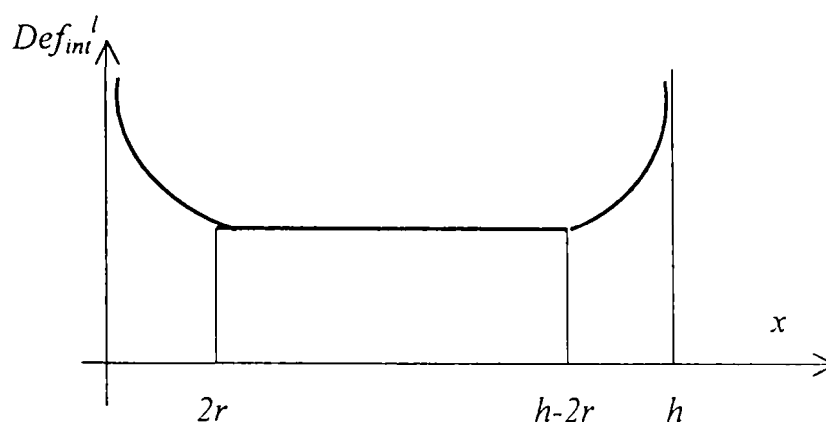


Fig.III.30.Reprezentarea grafică a defectului interior lateral la reprezentarea unui cilindru prin sfere

➤ Reprezentarea interioară a unui con prin sfere

Fie un con circular drept, având ca bază cercul de rază r_1 și înălțimea h_1 (figura III.31). Unghiul de la bază este α , dat de relația: $\operatorname{tg} \alpha = \frac{h_1}{r_1}$. Se înscrie în con o

sferă de rază R_1 a cărei valoare este evident: $R_1 = r_1 \cdot \operatorname{tg} \frac{\alpha}{2}$.

Generatoarea conului are mărimea: $g = \sqrt{h_1^2 + r_1^2}$, iar din: $\frac{g}{r_1} = \frac{h_1 - R_1}{R_1}$, rezultă:

$$R_1 = \frac{r_1 h_1}{r_1 + \sqrt{h_1^2 + r_1^2}} = \frac{r_1 \sin \alpha}{\sqrt{1 + \cos \alpha}} \quad (\text{III.2.86})$$

Un plan paralel la bază, la distanța $2R_1$ de aceasta, determină în con un alt con, de înălțime $h_2 = h_1 - 2R_1$ și de rază $r_2 = r_1 \frac{h_2}{h_1} = r_1 \frac{h_1 - 2R_1}{h_1}$.

Continuând procedeul, se pot înscrie oricâte sfere S_i se dorește, ele formând un șir, evident convergent, ai căror termeni sunt volumele:

$$V_i = \frac{4}{3} \pi \cdot R_i^3 = V_{i-1} \cdot \operatorname{tg}^3 \frac{\alpha}{2} \quad (\text{III.2.87})$$

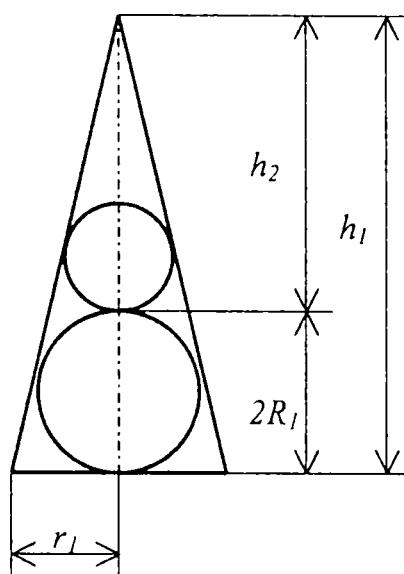


Fig.III.31. *Reprezentarea interioară a conului prin sfere*

Defectul de reprezentare interioară se va aborda la fel ca în cazul cilindrului, dar ținându-se seama de faptul că sferile de umplere nu mai sunt identice ci, volumele lor se află în relația (III.2.87) dacă ele sunt tangente. Dacă se intersectează, problema devine mult mai complicată pentru că volumele de intersecție ale sferelor de raze diferite nu mai sunt identice ca în cazul cilindrului, chiar dacă pasul de așezare de-a lungul axei conului este același.

➤ **Reprezentarea interioară a unui trunchi de con prin sfere**

❖ **O sferă în interiorul trunchiului de con**

Fie trunchiul de con circular drept, cu cercurile de bază de raze r_1 și r_2 și cu înălțimea h . Fie α unghiul format de generatoare, cu planul bazei (figura III.32). Se pune condiția ca: $h > 2r_1 \cdot \operatorname{tg} \frac{\alpha}{2}$. Este evidentă relația: $h = (r_1 - r_2) \cdot \operatorname{tg} \alpha$.

Fie o sferă de rază R , tangentă generatoarelor, al cărei centru este situat la distanța x de baza mare și y de baza mică, r fiind raza cercului de contact dintre sferă și trunchi iar u distanța, măsurată pe axa trunchiului, dintre centrul cercului de contact și centrul sferei.

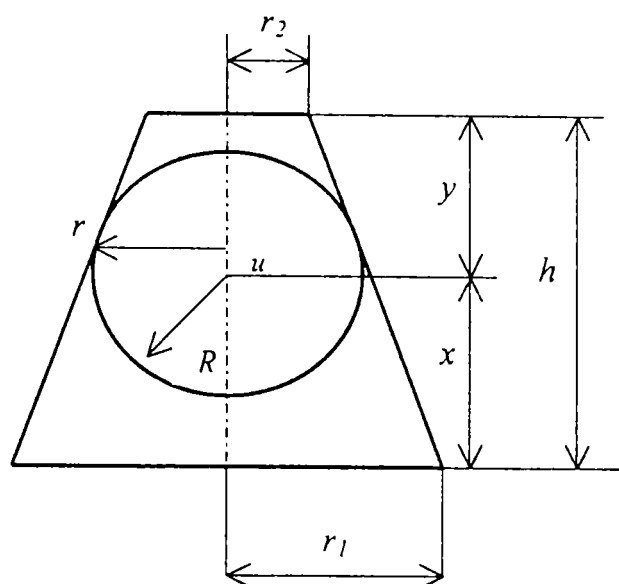


Fig.III.32. O sferă înscrisă într-un trunchi de con

Au loc relațiile:

$$\begin{aligned}
 x + y = h \quad r = R \sin \alpha \quad u = R \cos \alpha \quad \frac{r}{u} = \frac{x + u}{r_1 - r} &\Rightarrow \\
 x = \frac{r_1 \sin \alpha - R}{\cos \alpha} \quad y = \frac{R - r_2 \sin \alpha}{\cos \alpha} \quad x + y = (r_1 - R \sin \alpha) \cdot \operatorname{tg} \alpha & \quad \text{(III.2.88)}
 \end{aligned}$$

Defectul de umplere se obține ca diferență între volumul trunchiului de con și cel al sferei:

$$\begin{aligned}
 Def_{int} = V_{tr.con} - V_{sfera} &= \frac{1}{3} \pi \cdot h \cdot (r_1^2 + r_2^2 + r_1 \cdot r_2) - \frac{4}{3} \pi \cdot R^3 = \\
 &= \frac{1}{3} \pi \cdot x \cdot (r_1^2 + r^2 + r_1 \cdot r) + \frac{1}{3} \pi \cdot y \cdot (r^2 + r_2^2 + r \cdot r_2) - \frac{4}{3} \pi \cdot R^3 & \quad \text{(III.2.89)}
 \end{aligned}$$

În care dacă se utilizează relațiile (III.2.88) se poate obține expresia defectului ca funcție de distanța variabilă x . Ca în alte cazuri precedente se poate calcula distanța x pentru care defectul interior este minim, prin anularea derivatei funcției $Def_{int}(x)$.

❖ Sfera înscrisă în trunchiul de con

În acest caz $x = R, y = R$ și deci:

$$R = \frac{r_1 + r_2}{2} \sin \alpha \quad (\text{III.2.90})$$

expresia defectului interior obținându-se ca particularizare a relației (III.2.89).

❖ Două sfere înscrise în trunchiul de con

Se înscriu două sfere în trunchiul de con cu cercurile de bază de raze r_1 și r_2 și înălțimea h în care α este unghiul format de generatoare cu planul bazei. Una dintre sfere are raza R_1 fiind tangentă bazei mari și generatoarelor, iar cealaltă are raza R_2 și este tangentă bazei mici și generatoarelor. Centrele sferelor sunt plasate în O_1 și respectiv, O_2 , pe axa trunchiului de con (figura III.33).

Este evident:

$$R_1 = r_1 \cdot \operatorname{tg} \frac{\alpha}{2} \quad R_2 = r_2 \cdot \operatorname{tg} \frac{\beta}{2} \quad \alpha + \beta = \pi \quad h = (r_1 - r_2) \cdot \operatorname{tg} \alpha \quad (\text{III.2.91})$$

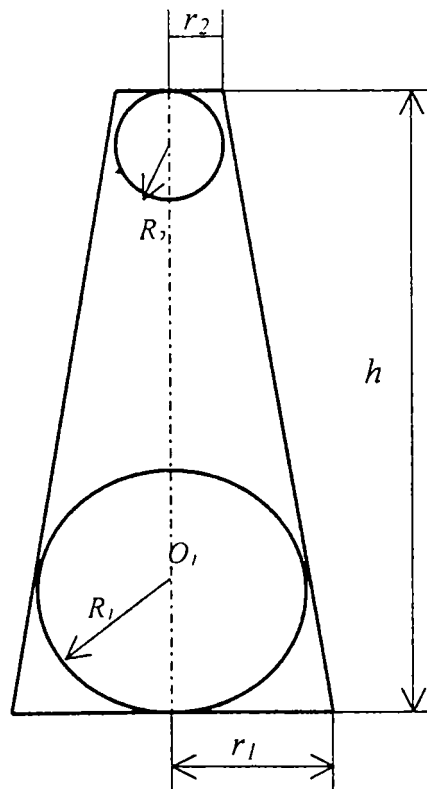


Fig.III.33. Două sfere înscrise într-un trunchi de con

Defectul de umplere:

$$Def_{int} = V_{tr.con} - V_{sfera1} - V_{sfera2} = \frac{1}{3}\pi \cdot h \cdot (r_1^2 + r_2^2 + r_1 \cdot r_2) - \frac{4}{3}\pi \cdot R_1^3 - \frac{4}{3}\pi \cdot R_2^3 \quad (III.2.92)$$

putând fi exprimat doar în funcție de parametrii conului și fiind o mărime constantă.

III.2.3. Considerații topologice asupra reprezentării prin sfere

Abordarea topologică a problemei reprezentării prin sfere se realizează pe baza considerațiilor matematice expuse în capitolul III.2.2. și în scopul de a detecta algoritmi de reprezentare ce vor fi utilizați la programare [16], [36], [76], [77], [78].

III.2.3.1. Reprezentarea exterioară

Fie o mulțime de m obiecte geometrice și un număr întreg n . Se dorește definirea unei mulțimi de n sfere astfel încât să nu existe nici un punct P aparținând frontierei unui obiect ce nu este conținut în interiorul a cel puțin unei sfere din reprezentare.

Adică:

$$\forall P \in BO_i, \quad \exists S_j \in S \mid P \in S_j \quad (III.2.93)$$

unde:

$O = \{O_i, i=1, \dots, n\}$ este mulțimea tuturor obiectelor;

$S = \{S_j, j=1, \dots, n\}$ este mulțimea tuturor sferelor din reprezentare;

BO_j este frontiera (marginea) obiectului O_i .

➤ Cazul plan

Se consideră cazul mai simplu de reprezentare a corpurilor în spațiul 2D. Obiectele vor fi poligoane convexe pentru că există metode cunoscute de a rezolva problema împărțirii unui poligon oarecare într-o mulțime de poligoane convexe. Problema devine însă complexă dacă se urmărește împărțirea optimă, spre exemplu, obținerea unui număr minim de poligoane convexe.

Dat fiind obiectivul urmărit, de reprezentare a corpurilor printr-un corp de formă unică, în particular prin sfere, se studiază modul în care se pune problema acestei reprezentări la structuri plane, adică la acoperirea poligoanelor convexe prin cercuri.

❖ Acoperirea unui poligon cu un cerc

Se pune problema găsirii unui cerc ce acoperă toate vârfurile poligonului. Pentru a fi optim, cercul trebuie să fie cât mai mic posibil. Această definiție corespunde problemei cu un singur centru ce poate fi rezolvată cu un algoritm ce folosește diagrama Voronoi și care este mult utilizat în problemele de modelare pentru planificarea mișcărilor [70], [71], [72], [110], [112].

Se poate însă folosi o metodă mai simplă bazată pe faptul că cel mai mic cerc trece prin cel puțin două vârfuri ale poligonului. Astfel, se caută cel mai mic cerc ce include un poligon, existând, prin aplicarea acestei variante, posibilitatea de accelerare a procesului de căutare (figura III.34).

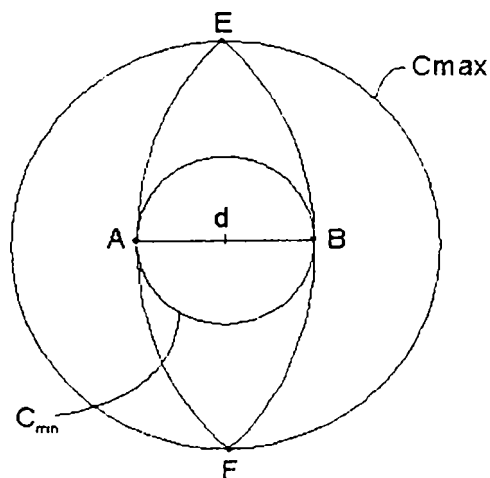


Fig.III. 34. Limitele razei celui mai mic cerc circumscris

Într-un prim pas este calculată perechea cea mai îndepărtată de mulțimi de vârfuri. Această pereche definește *diametrul* d al poligonului și evident, dacă cercul trece prin două vârfuri, diametrul va fi și cel al cercului și se va limita la punctele A și B ce realizează perechea respectivă.

Fie $R = \frac{d}{2}$ raza cercului. Dacă celelalte vârfuri sunt conținute de cerc, problema este rezolvată. Dacă nu, cel mai mic cerc circumscris va tăia cel puțin trei vârfuri, ale poligonului iar raza sa va avea valori în intervalul $(R\sqrt{3}, R)$.

Limita inferioară este dată de diametrul poligonului. Limita superioară corespunde razei unui cerc C_{max} cu centrul în mijlocul dreptei AB și care trece prin punctele E și F . Acest cerc acoperă toate punctele dar nu este optim.

Este evident că un vârf al poligonului în exteriorul cercului nu poate să existe dacă distanța de la A la B este mai mare decât d pentru că în felul acesta ar fi contrazisă definiția diametrului poligonului.

În faza următoare, se începe căutarea unei soluții din cadrul tuturor mulțimilor de trei de vârfuri posibile (triade). Fiecare triadă definește un cerc ce poate fi cel optim. Mai întâi se verifică dacă raza r îndeplinește condiția: $R < r < R_{max}$ în care $R_{max} = R\sqrt{3}$.

Dacă nu se îndeplinește această condiție, soluția este respinsă. Dacă se îndeplinește, atunci se verifică dacă toate vârfurile sunt acoperite. În acest din urmă caz R_{max} ia valoarea razei ultimului dintre cercurile acceptate, care va reprezenta *cercul soluție* cu cea mai mică dintre razele găsite până la etapa respectivă. Repetarea acestui proces de mai multe ori va conduce la soluția finală.

În principiu, pentru valori $n < 20$ numărul de triade de vârfuri care se cercetează pe parcurs are expresia: $(n^3 - 3n^2 + 2n)/16$. Pentru valoarea $n=20$ se obțin 1140 de operații, ceea ce este perfect acceptabil.

❖ Acoperirea unui poligon cu mai multe cercuri

Problema care se urmărește este *gradul de acoperire*. Problema se poate formula astfel:

Se dau n puncte într-un plan. Să se găsească p puncte înlocuitoare, oriunde în plan, astfel ca să se minimizeze distanța maximă de la un punct dorit la cel mai apropiat înlocuitor al său.

Rezolvarea ia în considerare modificarea enunțului, pe baza unei probleme mai restrictive din teoria grafurilor și anume:

La o mulțime dată de puncte și cercuri în plan să se determine modul de aranjare a cercurilor astfel încât să acopere toate punctele.

Se urmărește deci găsirea unei reprezentări exterioare pentru un poligon prin intermediul unei mulțimi de n cercuri. Razele și localizarea cercurilor trebuie alese astfel încât reprezentarea rezultată să acopere marginile poligonului, adică laturile sale. Pentru aceasta se vor împărți laturile poligonului pentru a se obține n zone disjuncte și se va acoperi fiecare zonă cu un singur cerc. O astfel de zonă a frontierei este definită printr-o mulțime de puncte numite *listă de puncte*. Fiecare punct al listei trebuie să fie vârf al poligonului cu excepția capetelor listei care pot fi puncte oriunde pe o latură.

Pentru a acoperi una din aceste liste cu un singur cerc, tot ce trebuie făcut este de a aplica algoritmul descris în cazul cercului unic.

Metoda prezentată este secvențială și urmărește calculul reprezentării exterioare a unui poligon folosind un număr n de cercuri. Această reprezentare se face prin modificarea reprezentării a $n-1$ cercuri. Lista cea mai defavorabilă este selectată și împărțită în alte două subliste. Noua reprezentare, a n -a, este formată prin înlocuirea cercului reprezentat de lista veche necorespunzătoare cu cele două cercuri corespunzând noilor subliste. Fiecare nouă reprezentare va trebui să fie mai bună decât cea precedentă (figura III.35).

Problema alegerii celei mai defavorabile liste este o problemă de decizie ce implică definirea unui criteriu care să evalueze reprezentarea și să decidă continuarea divizării în alte subliste sau nu.

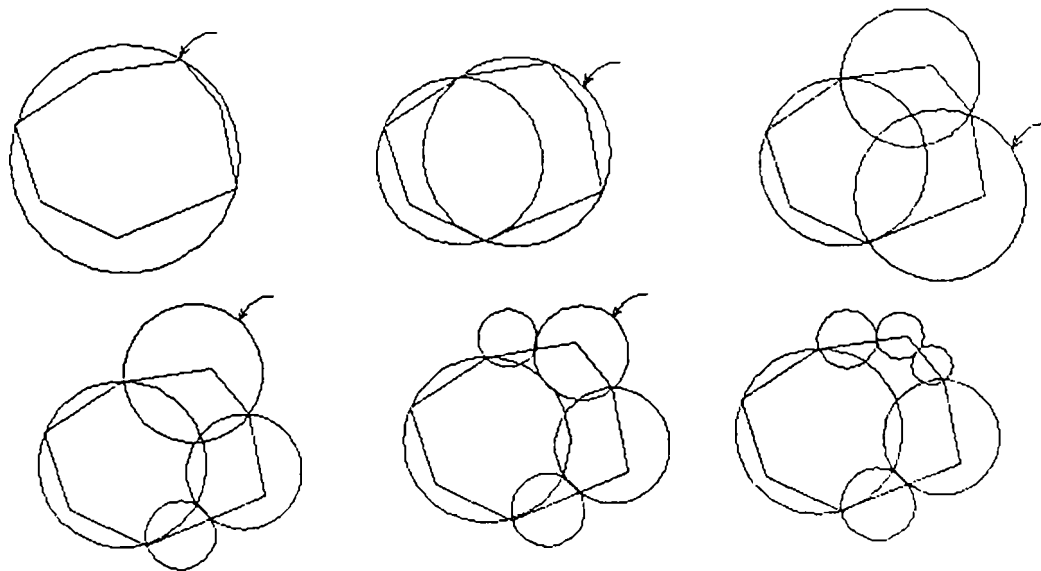


Fig. III.35. Șirul reprezentărilor celor mai defavorabile liste

❖ Algoritmul acoperirii poligonului

Odată realizată reprezentarea exterioară prin includerea frontierelor, metoda se poate extinde la întregul poligon, ce poate fi acoperit cu o mulțime de cercuri.

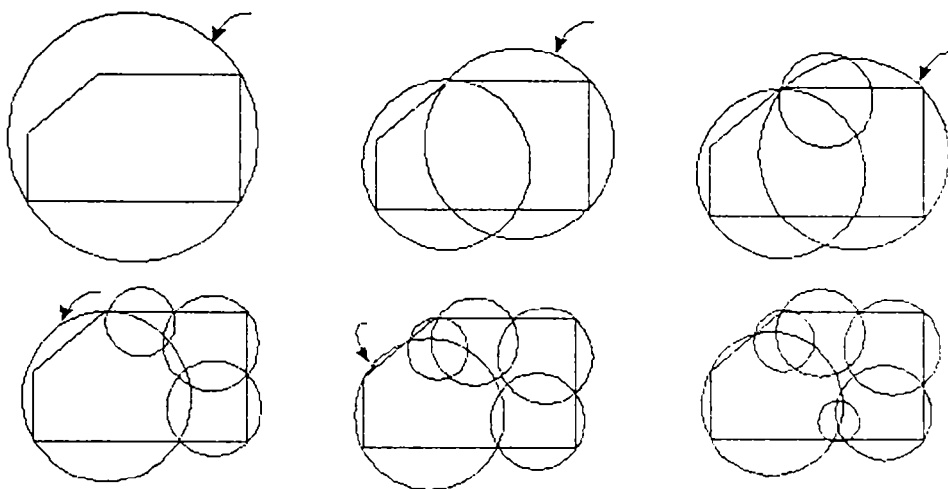


Fig.III.36. Șirul reprezentărilor celor mai defavorabile cercuri

Pentru aceasta, se împarte întregul poligon în n zone convexe. Fiecare zonă va fi acoperită cu un cerc prin algoritmul cercului unic. Realizarea reprezentării cu un număr dat de n cercuri urmează aceeași procedură secvențială ca în cazul anterior: n cercuri se obțin îmbunătățind varianta cu $n-1$ cercuri.

Se începe cu un singur cerc, se împarte poligonul în 2 părți, se obțin 2 sub-poligoane ce sunt acoperite cu 2 cercuri formând Apoi, se selectează cel mai defavorabil dintre aceste sub-poligoane, care se divide în alte două sub-poligoane, și se continuă în același mod.

În figura III.36 se poate urmări acest proces pentru un poligon oarecare. În fiecare desen este indicat cercul care acoperă cel mai defavorabil poligon.

Se observă că acest caz este tratat ca și cel anterior pentru reprezentarea exterioară. Doar că în cazul precedent algoritmul de referă la *liste* iar în acest caz, la *poligoane*. Metodele se diferențiază atât prin criteriul de alegere a listei, respectiv al poligonului cel mai defavorabil cât și prin metoda de divizare folosită.

➤ **Extensia algoritmului de acoperire în 3D**

Problema acoperirii spațiului prin intermediul sferelor se referă, în principal, la acoperirea completă a spațiului și nu la utilizarea unei sfere ca mijloc de reprezentare aproximativă a unui obiect geometric. Din studiul geometric al acoperirii diferitelor tipuri de corpuri s-a observat că cel mai general caz a fost cel al cilindrului, din care prin relații specifice se pot studia și alte forme de corp. De aceea studiul topologic cu cel mai înalt grad de generalitate se referă doar la cilindru. De altfel și programele create pentru reprezentarea prin sfere dezvoltă alte forme de corpuri pornind de la cilindru [148], [149].

❖ **Acoperirea unui corp solid cu un cilindru**

În teoria mulțimilor, sistemul de mulțimi S_1, S_2, \dots se presupune că acoperă mulțimea O , dacă:

$$\bigcap S_i \supset O \quad (\text{III.2.94})$$

adică, dacă orice element al lui O aparține cel puțin uneia din mulțimile S_1, S_2, \dots

Se consideră O o mulțime de puncte în spațiul euclidian tridimensional și se restrânge conceptul de acoperire obligând mulțimile S_1, S_2, \dots , să formeze un sistem finit de *translatate* ale unei mulțimi unice K . Mulțimea tuturor punctelor formei $k+a$, unde $k \in K$ iar a este un vector fix, este translatata lui K de vector a . Se nota aceasta translatată cu $K+a$.

Definiție. *Un sistem de translatate $K+a_i$ ($i=1, 2, \dots$) al unei mulțimi K prin șirul de vectori a_1, a_2, \dots formează acoperirea unei mulțimi O , dacă fiecare punct al lui O se află în cel puțin o mulțime a sistemului.*

Dacă vectorii a_1, a_2, \dots sunt o enumerare de puncte ale unei rețele, acoperirea se numește *rețea de acoperire* sau *rețea*. Pentru simplificare se numește o *acoperire cu K* orice acoperire cu translatate ale lui K .

S-a demonstrat [7] că: *O rețea de acoperire în spațiul $(n+1)$ dimensional se poate obține dintr-o rețea în spațiul n -dimensional.*

Dacă acoperirea în acest spațiu este alcătuită din translatate ale mulțimii C , acoperirea în spațiul $(n+1)$ dimensional este construită prin intermediul cilindrilor generalizați având-o pe C ca bază. Ulterior cilindrii sunt înlocuiți cu sfere și astfel se obține o acoperire având ca mulțime generatoare pe K , compusă numai din sfere.

Conceptul de cilindru vine de la conceptul de produs cartezian a două mulțimi. Astfel, dacă F este o mulțime în spațiul m -dimensional, iar G o mulțime în spațiul n -dimensional produsul cartezian între F și G este definit în spațiul $(n+m)$ -dimensional, se notează cu $F \times G$, iar mulțimea punctelor sale este: $(x_1, x_2, \dots, x_m, y_1, y_2, \dots, y_n)$ unde $(x_1, x_2, \dots, x_m) \in F$ și $(y_1, y_2, \dots, y_n) \in G$.

Se poate numi acest produs cartezian, *cilindru generalizat*, deoarece urmează procedeul de construire a unui cilindru.

Se presupune că mulțimea corpurilor ce urmează a fi reprezentate se poate descompune într-o nouă mulțime formată din obiecte mai simple: $\{O_i, i=1, \dots, m\}$, astfel încât fiecare obiect poate fi generat de translația prin baleiere a unei mulțimi convexe plane.

Astfel, un obiect O_i este produsul cartezian $B_i \times H_i$, unde B_i este un poligon convex, iar H_i este un segment de dreaptă perpendicular pe planul lui B_i și având un capăt în acest plan. Se subînțelege că se atribuie numele de *obiect*, acestui tip particular de construcție.

Conform definiției, o reprezentare exterioară a obiectului solid O trebuie să acopere suprafața care îl limitează. În cazul translației prin baleiere, această suprafață numită *acoperire* constă din două categorii de suprafețe: suprafața mulțimii generatoare 2D notată cu B și noile suprafețe laterale generate în timpul translației prin baleiere a lui B . Cele două suprafețe formatoare sunt denumite: *baza* obiectului, respectiv *marginile*.

O primă observație legată de soluția constructivă poate fi interpretată astfel:

Marginile obiectului sunt rezultatul translației laturilor poligonului generator; deci, dacă există o mulțime de cercuri care acoperă toate laturile în plan, procesul de baleiere va produce acoperirea marginilor obiectului tridimensional prin cilindrii ce au aceste cercuri ca baze.

Astfel, prin generalizare se poate stabili, că fiind dat un poligon convex B și o acoperire C pentru frontierele sale, solidul generat prin translația de baleiere a acestui poligon realizează o acoperire a suprafețelor laterale a corpului prin acoperirea

reprezentată de cilindrii ce au pe C ca bază. În mod identic se poate găsi o acoperire pentru întregul volum corespunzător lui B .

❖ Acoperirea cilindrilor cu sfere

S-a reușit acoperirea frontierelor unui obiect prin cilindrii, dar scopul final este să se utilizeze numai sfere. În plus, apare o altă problemă legată de faptul că cele două capete ale obiectului, inferior și superior, nu sunt încă acoperite.

Se urmărește modalitatea de acoperire a cilindrilor cu sfere.

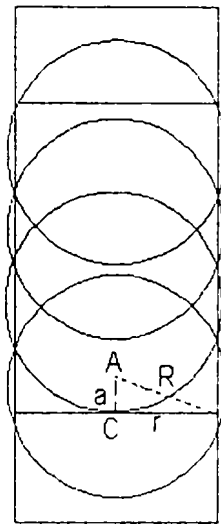


Fig.III.37. Acoperirea unui cilindru cu sfere

O mulțime de cilindri a fost definită pentru a acoperi suprafețele laterale ale unui obiect. Obiectul are ca elemente generatoare un poligon B și un segment de dreaptă H . Toți cilindrii au același segment generator H , dar fiecare își are propriul cerc generator C_i . Pentru a acoperi doar un singur cilindru cu un număr n de sfere, se poate proceda astfel conform figurii III.37:

Fie r_i raza cercului de bază C_i , h înălțimea sa (care reprezintă lungimea segmentului H), A_i un punct localizat la distanța $d = \frac{h}{2n}$ de centrul C_i pe segmentul H , iar \mathbf{a} vectorul cu originea în C_i și capătul în A_i .

Astfel poate fi definită sfera S_i cu centrul în A_i și raza R_i , ce satisface relația: $R_i^2 = r_i^2 + d^2$. Se poate ușor demonstra că sistemul de translate $S_i + \mathbf{a}_j$ ($j=1, 2, \dots, n$)

formează o rețea ce acoperă cilindrul cu S_i , unde șirul de vectori $\{a_j\}$ este definit de vectorul a ca: $a_j = (2j-1)a$, ($j=1, 2, \dots, n$).

Ca și corolar al acestui rezultat, se poate stabili următoarea consecință:

Frontiera laterală a unui obiect traslatat prin baleiere este acoperită de sistemul $K+a_j$, unde K este mulțimea compusă în întregime din sfere $\{S_i, i=1, 2, \dots, m\}$. Vectorul a_j și sfera S_i sunt definite ca mai sus, fiecare S_i fiind asociat cu un cilindru și un cerc C_i din mulțimea $\{C_i, i=1, 2, \dots, m\}$ care formează acoperirea 2D a poligonului generator B .

Se observă că a și secvența a_j sunt aceleași pentru toți cilindrii chiar dacă sferile S_i au raze diferite. Numărul total de sfere acoperitoare este $m.n$.

Pentru cazul când se utilizează o singură sferă pentru acoperirea marginilor unui obiect ($m=n=1$), se poate arăta simplu că această sferă include integral obiectul. Prin definiție, ea acoperă marginile obiectului. De fapt, aceasta înseamnă că sfera include cilindrul definit de C_1 , iar acesta include toate suprafețele laterale. Dar, deoarece în plus, C_1 include poligonul generator al obiectului, înseamnă că el include întreg obiectul, chiar și cele două capete inferior și superior, ceea ce revine la a afirma că sfera acoperă întregul solid.

❖ Reprezentarea capetelor (superior și inferior)

Problema care se pune este de a acoperi suprafețele de bază ale obiectului, cea superioară și cea inferioară. Acestea fiind echivalente din punct de vedere al construcției, referirea se va face doar la cea superioară.

Frontiera superioară a unui volum traslatat este suprafața închisă de laturile poligonului convex generator. Cum o acoperire cu un set de cercuri $\{C_i, i=1, 2, \dots, m\}$ este valabilă pentru întregul poligon, o acoperire pentru frontiera superioară a solidului poate fi simplu obținută cu un set de sfere, în locul fiecărui cerc C_i considerându-se sfera cu același centru și aceeași rază.

Acoperirea frontierelor superioară și inferioară, alături de acoperirea celor laterale, completează reprezentarea exterioară.

Dar, în anumite cazuri, pentru a păstra cele trei caracteristici ale acoperirii și anume să fie *perfectibilă - echilibrată - optimă*, un nou tip de acoperire trebuie introdusă.

În locul acoperirii frontierelor unui obiect cu un singur sistem de traslatate de $K+a_j$, se împart fețele laterale în trei zone și se acoperă independent fiecare dintre ele.

Se acordă mai multă atenție zonelor adiacente cu cele două capete ale corpului, împărțindu-se obiectul prin tăiere cu două plane paralele cu cele două capete sale și pe care le vor închide. Astfel, se definesc două fâșii, care vor fi numite *capetele obiectului*. Se va studia doar unul singur dintre ele, presupunând că ceea ce este valabil pentru unul este valabil și pentru celălalt.

Astfel, reprezentarea exterioară este compusă din trei acoperiri diferite (figura III.38), câte una pentru cele două capete (superior și inferior) și una pentru suprafețele laterale. Reprezentarea pentru capete este identică cu cele descrise deja pentru obținerea reprezentării frontierelor laterale. Este evident că cele două acoperiri diferă între ele prin sistemele $K+a_j$. Adică, mulțimile lor de sfere $\{S_i, i=1, \dots, m\}$, mulțimile lor de cercuri $\{C_i, i=1, \dots, m\}$ și vectorii lor din rețea a_j sunt diferiți.

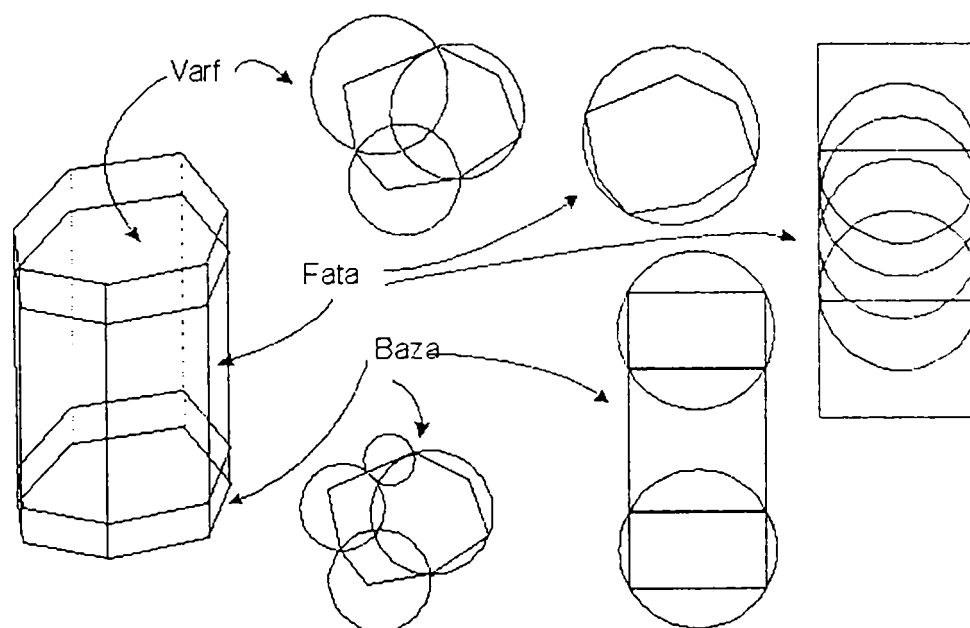


Fig. III.38. Exemplu de reprezentare exterioară

➤ Calitatea unei reprezentări

În teoria matematică singurul parametru pentru determinarea calității acoperirii este *densitatea*. Pentru acoperirea unui spațiu prin sistemul $K+a_j$, densitatea $\rho(K+a_j)$ este descrisă ca limita raportului dintre suma măsurilor acelor mulțimi

aparținând sistemului de translatate cere se afla într-un cub uriaș și măsura acestui cub, dacă el tinde spre infinit..

Poate fi arătat că: *dacă mulțimea K este mărginită și are măsură pozitivă, și dacă sistemul de translatate $K+a_j$ formează o acoperire, atunci: $\rho(K+a_j) \geq 1$.*

Densitatea oferă însă, doar informații despre cât de optimă este acoperirea, adică, cu cât densitatea este mai mare, acoperirea este mai compactă, iar numărul sferelor este mai mare.

Problema care interesează este și cât de bine este aproximată *forma* obiectului prin intermediul sferelor. Ideea de bază, pe care se sprijină toate noțiunile și definițiile ce urmează este aceea că: *calitatea unei reprezentări se poate aprecia din măsura mulțimii eroare E .*

Această mulțime se compune din punctele Q ce aparțin unei anumite sfere a acoperirii, dar nu sunt conținute în nici unul din obiectele reale. Adică: $Q \in E$ dacă și numai dacă:

$$\exists S_j \in S / Q \in S_j \text{ și } \forall i (i=1, 2, \dots, m) Q \notin Q_i \quad (\text{III.2.95})$$

Sau mai simplu: $Q \in S$ și $Q \notin O$

Sau doar: $E=S-O$

unde:

$O=(O_i, i=1, \dots, m)$ este mulțimea tuturor obiectelor

$S=(S_j, j=1, \dots, n)$ este mulțimea tuturor sferelor din reprezentare.

Ideea intuitivă pe care se bazează această definiție a lui E este aceea că mulțimea erorilor caracterizează discrepanța dintre reprezentare și obiectele reale, astfel încât o măsură a dimensiunii sale servește la cuantificarea calității reprezentării. Este evident că cu cât reprezentarea conține mai multe puncte care nu aparțin în realitate obiectului, cu atât este mai grosolană calitatea reprezentării. Cea mai exactă reprezentare ar fi cea în care mulțimea erorilor ar fi nulă [6], [8], [15], [16].

Când se caută o măsură a mulțimii E , foarte intuitiv ar fi un volum potrivit necesităților reale. Totuși, unele volume nu oferă siguranța că s-a realizat o reprezentare cu un grad înalt de confidență. Astfel, pot exista două reprezentări cu același volum al erorii, dar care să difere semnificativ în privința calității reprezentării.

Spre exemplu, în cazul (a) din figura III.39, forma obiectelor este mai puțin bine aproximată decât în cazul (b), deși mulțimea erorilor are volume similare.

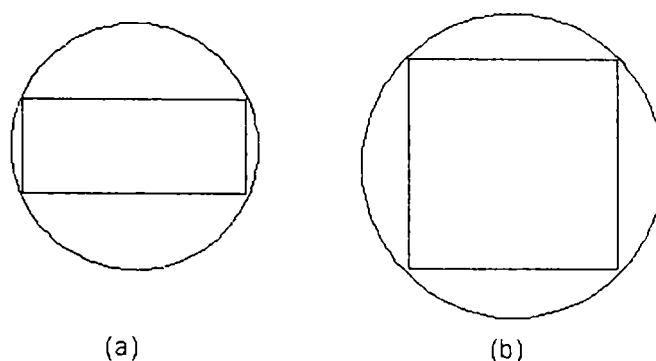


Fig.III.39. Volumul, ca măsură a erorii

Motivul acestui aparent paradox este că, deși dimensiunile obiectelor sunt comparabile, formele lor sunt total diferite, ceea ce face ca reprezentarea lor, cu un singur cerc în ambele cazuri, să aibă calități diferite.

În locul volumului, ca măsură a mulțimii erorii, s-a introdus coeficientul δ având unități de lungime. Coeficientul de acoperire δ poate fi descris ca raportul dintre volumul mulțimii eroare și suprafața frontierei acoperite a obiectului:

$$\delta = \frac{Vol(E_i)}{Supraf(BO_i)} \quad (III.2.96)$$

unde:

E_i este mulțimea erorilor corespunzătoare obiectului BO_i

BO_i este frontiera exterioară a obiectului ce se acoperă.

Această definiție corespunde erorii medii a distanței când se aproximează frontiera exterioară a obiectului real față de frontiera reprezentării.

Definiția (III.2.96) a lui δ poate fi folosită și pentru acoperirea parțială a unui obiect. Dacă se consideră fiecare porțiune în care obiectul a fost împărțit așa cum s-a prezentat anterior, și se determină acoperirea pentru fiecare din ele, atunci în fiecare caz în parte se poate aplica conceptul δ pentru a obține câte o *acoperire parțială*.

În plus, dacă se asociază fiecărei porțiuni din suprafața solidului câte o mulțime de sfere care o acoperă, atunci se poate ușor defini mulțimea eroare pentru această acoperire locală prin intermediul coeficientului δ care dă informații asupra calității sale. Această capacitate de a defini parțial calitatea reprezentării este esențială atunci când se urmărește realizarea unei reprezentări echilibrate, adică oferă posibilitatea de a decide care dintre părțile reprezentării contribuie mai mult la eroarea de reprezentare .

O altă problemă care se pune este de a preciza de ce nu s-a considerat o mărime adimensională, în locul coeficientului de acoperire δ , care să nu depindă de dimensiunile reale ale corpului. Această alegere poate fi argumentată prin faptul că δ , care are dimensiuni de lungime, poate oferi informații absolute și nu relative asupra mărimii obiectului. Deci, un obiect mai mare va fi asociat cu un δ de valoare mai mare decât un obiect mai mic, chiar dacă ambele reprezentări sunt la fel de precise în raport cu formele reale ale fiecărui obiect în parte.

Ceea ce se dorește nu este realizarea unei reprezentări absolute a unui obiect izolat, ci mărirea preciziei reprezentării unui obiect, acest lucru depinzând de legăturile sale cu restul obiectelor înconjurătoare. Dacă se privește din punct de vedere al planificării mișcării, rezultă că în reprezentare sunt foarte importante distanțele absolute dintre obiectele-roboti și obiectele-obstacole, aceste distanțe influențând reprezentarea unui anumit obiect la un anumit moment.

Totuși pentru a ști în ce măsură obiectul se apropie de reprezentarea sa fără să se ia în considerare dimensiunile sale, ar trebui definit un parametru de apreciere adimensional prin împărțirea lui δ cu o distanță care să fie caracteristică mărimii obiectului în cauză (această distanță ar putea fi considerată diametrul celei mai mici sfere circumscrise).

❖ Erorile reprezentărilor plane

Este evident că definiția mulțimii erorilor E , introdusă pentru cazul acoperirilor prin sfere, poate fi adaptată oricărui tip de obiecte, indiferent care este natura lor și figura geometrică ce stă la baza mulțimii de acoperire. Deci, dacă definiția lui δ este extinsă la cazul plan, trebuie considerată o suprafață ca măsură a mulțimii erorilor și să se împartă mărimea sa la lungimea frontierei acoperite, rezultând o relație similară cu (III.2.96) dar având la numărător o arie iar la numitor o

lungime. Coeficientul de acoperire astfel introdus pentru un obiect plan O_i care are frontiera BO_i și mulțimea erorilor E_i pentru reprezentare, se notează ε :

$$\varepsilon = \frac{\text{Supraf}(E_i)}{\text{Lung}(BO_i)} \quad (\text{III.2.97})$$

Un coeficient local de calitate ε_{local} poate fi considerat pentru a măsura calitatea unei acoperiri parțiale prin intermediul unui singur cerc pentru o porțiune a frontierei poligonului (figura III.40):

$$\varepsilon_{local} = \frac{F}{\text{Lung.listei}(ABCD)} \quad (\text{III.2.98})$$

unde:

F este suprafața din figură

$AB+BC+CD$ este lungimea listei de puncte

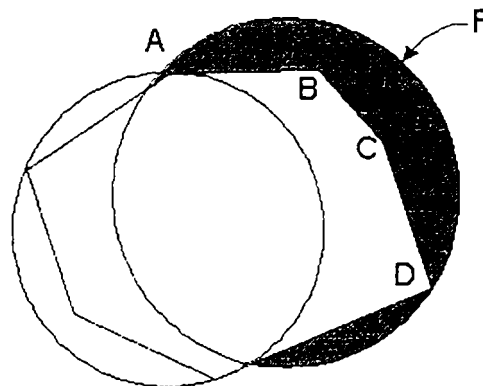


Fig.III.40. Coeficientul de calitate ε_{local}

Pentru un poligon întreg, coeficientul se calculează în mod identic (figura.III.41), adunând diferitele suprafețe eroare și împărțind cu perimetrul poligonului. Deoarece se folosește această măsură atât pentru reprezentarea frontierelor laterale cât și pentru extremități, se vor folosi în continuare pentru acestea notații similare cu cele de la reprezentarea geometrică: ε^l_{ext} , respectiv ε^b_{ext} .

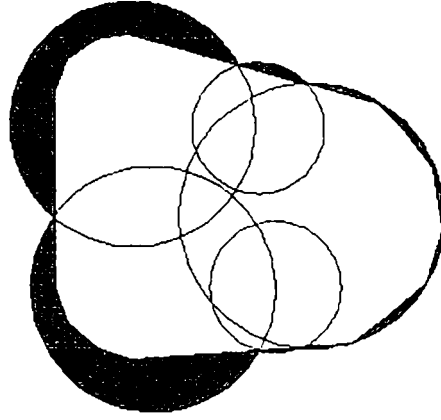


Fig.III.41. Coeficientul de calitate ε pentru un poligon complet

❖ Erorile reprezentării suprafețelor laterale și capătului superior

Acoperirea suprafeței laterale pentru un obiect tridimensional s-a prezentat anterior. Pentru calculul calității acestei acoperiri se consideră coeficientul δ_{local} pentru o mulțime de sfere aparținând unui singur sistem de translatate $S_i + \mathbf{a}_j$ ($j=1, 2, \dots, n$) corespunzând unui cilindru cu cercul de bază C_i .

Sunt însă necesari și alți doi parametrii:

- Erorile datorate cilindrului ce acoperă frontierele laterale (figura III.42);
- Erorile datorate sferelor sistemului $S_i + \mathbf{a}_j$, ce acoperă cilindrul (figura III.43).

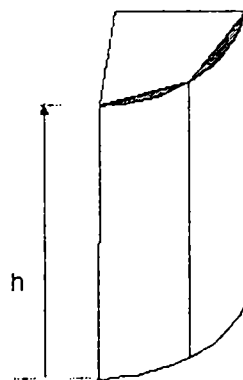


Fig.III.42. Erori datorate cilindrului ce acoperă suprafețele laterale

Eroarea de reprezentare este cuantificată prin *coeficientul local al calității*. Acest coeficient este definit ca raportul dintre mulțimea erorilor cilindrului (figura

III.42) și suprafața laterală a corpului. Se poate simplu percepe că acest coeficient este coeficientul ε pentru C_i . In acest caz particular, adăugând dimensiunea h relației de definiție a lui δ , aceasta conduce la ε . Astfel, acesta devine ε'_{ext} .

Un alt parametru local, numit δ'_{ext} se introduce pentru a calcula eroarea care există între sfera acoperitoare și cilindrul acoperit (similar studiului geometric al acoperirii). Acesta poate fi de asemenea definit ca raport dintre mulțimea de erori a sferelor (considerate acoperirea cilindrului) și suprafața laterală a cilindrului acoperit (figura III.43):

$$\delta'_{ext} = \frac{F \cdot L}{h \cdot l} = \frac{R^2 - r^2}{3r} \quad (III.2.99)$$

în care: F este suprafața prezentată în figură

L este lungimea curbei descrisă de centrul său de greutate G

l este lungimea segmentului de cerc prezentat

R este raza sferelor S_i

r este raza lui C_i .

De menționat că acest δ'_{ext} care este un parametru cu caracter local, nu se modifică dacă se iau în considerare toate sferile din sistemul $S_i + a_j$.

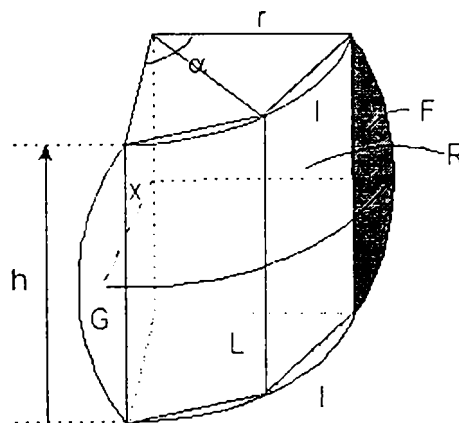


Fig.III.43. Erori datorate sferelor din $S_i + a_j$, ce acoperă cilindrul

Pentru întreaga acoperire a frontierelor laterale δ'_{ext} este practic echivalent cu ε din cazul plan, fiind dat de:

$$\delta_{ext}^l = \frac{\sum F_i \cdot l_i}{\sum h_i \cdot l_i} = \frac{1}{3} \cdot \frac{\sum \alpha_i (\sqrt{R_i^2 - r_i^2})^3}{\sum \alpha_i \cdot r_i \cdot \sqrt{R_i^2 - r_i^2}} \quad (III.2.100)$$

ambele sume din relația (III.2.100) fiind extinse la întregul sistem de mulțimi $\{S_i + a_j, i=1, 2, \dots, m, j=1, 2, \dots, n\}$.

Toți acești coeficienți pot fi definiți exact în același mod pentru capătul superior. Ei caracterizează în egală măsură reprezentarea diferitelor porțiuni. Este și aceasta o cale de a aprecia contribuția componentelor reprezentării la calitatea totală a acesteia.

Se vor utiliza astfel pentru reprezentarea frontierelor laterale δ_{ext}^l iar pentru capătul superior δ_{ext}^{bsup}

❖ Eroarea reprezentării capătului inferior

Acoperirea capătului inferior este caracterizată printr-un parametru distinct numit δ_{ext}^{binf} ce cuantifică calitatea reprezentării sale. Se poate determina în mod similar cu definirea generală a lui δ . Uneori acest parametru poate conduce la rezultate incorecte. Spre exemplu, dacă acoperirea frontierei laterale sau a capătului superior al obiectului acoperă parțial și capătul inferior, atunci se produce un efect nedorit.

În acest caz, eroarea de reprezentare este dată de acea parte din reprezentarea acoperirilor pentru suprafețele laterale sau/și a capătului superior care acoperă și capătul inferior și nu din reprezentarea propriu-zisă a acestuia. Pentru a rezolva acest inconvenient, se introduce un nou cuplu de coeficienți locali δ_{ext}^{bsup} *virtual* și δ_{ext}^{binf} *virtual*, care își mențin definițiile generale dar care se referă doar la sferele care acoperă parțial capătul inferior.

Procesul descris justifică introducerea reprezentării specifice pentru capătul superior și poate fi perceput din figura III.44. Îmbunătățirea acoperirii frontierelor laterale prin folosirea mai multor sfere conduce la înrăutățirea acoperirii capătului inferior.

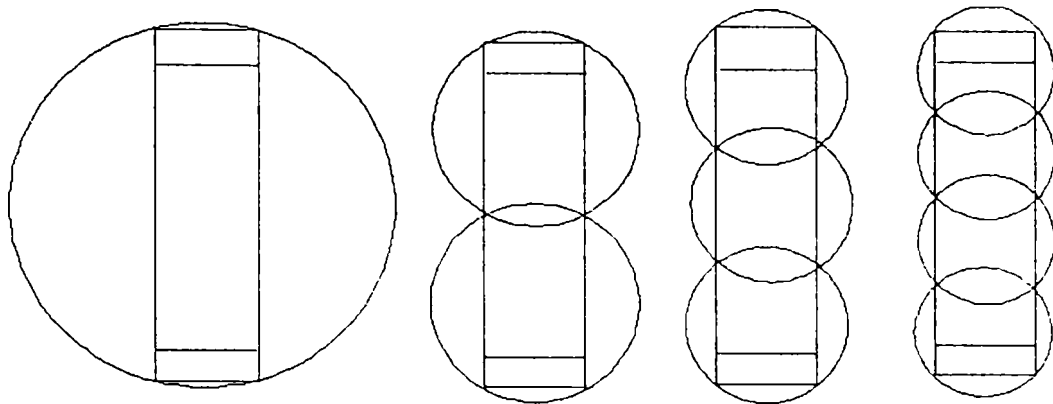


Fig.III.44. Îmbunătățirea acoperirii suprafețelor laterale prin folosirea mai multor sfere înrăutățește acoperirea capătului inferior

De fapt, cu cât mai bună este prima acoperire, cu atât mai deficientă este cea de a doua.

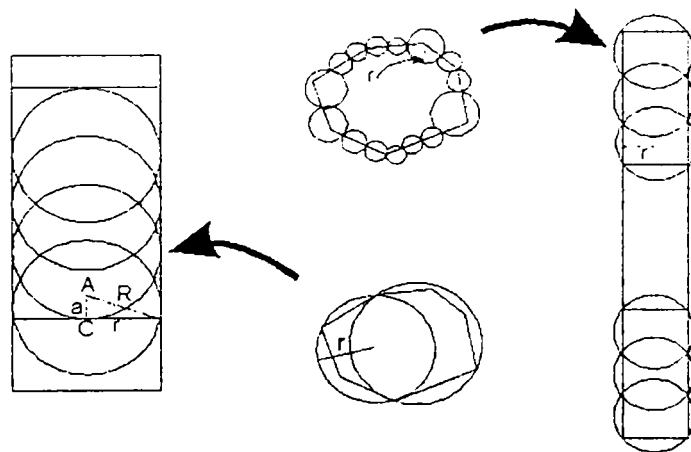


Fig.III.45. Acoperirea suprafețelor laterale și a capetelor

Unica soluție pentru remedierea acestei deficiențe este adăugarea mai multor sfere mulțimii generatoare K pentru sistemul $K+a_j$, dar astfel s-ar produce o falsă creștere a numărului sferelor din reprezentare. Soluția ar putea fi utilizarea unor acoperiri speciale pentru capete, caz în care numărul sferelor s-ar reduce, dar acoperirea ar fi îmbunătățită (figura III.45).

❖ Eroarea totală

Este evident că este necesar un parametru ce poate evalua calitatea ansamblului reprezentării. În acest scop este utilizat coeficientul δ_{total} , definit ca raportul dintre mulțimea erorilor pentru toate sferile reprezentării și întreaga suprafață exterioară a obiectului. Acest parametru poate fi utilizat și pentru a verifica convergența unui șir de reprezentări care tinde spre limita de eroare zero.

Definițiile mărimilor anterior introduse pentru caracterizarea unei acoperiri sunt destul de simple cu excepția calculului volumului reuniunii sectoarelor de sferă, fapt care a fost scos în evidență la prezentarea geometrică din capitolul III.2.2. Pentru acesta a fost elaborată o metodă simplă de calcul al coeficienților de acoperire.

Și anume, pentru acoperirea frontierei laterale și a capătului superior problema a fost rezolvată prin împărțirea mulțimii eroare în două submulțimi: una care conține toate punctele din interiorul cilindrilor și cealaltă care conține restul punctelor. Volumul mulțimii acoperire poate fi simplu calculat pentru că se observă că ultima din cele două submulțimi este formată din sectoare sferice care nu se intersectează între ele și deci calculul volumului se realizează simplu.

Pentru calculul coeficienților de calitate a acoperirii pentru capătul inferior se folosește o metodă aproximativă. În loc să se lucreze cu mulțimea eroare reală, se calculează volumul unei mulțimi noi imaginare. Aceasta este definită ca o *compactare* extrem de densă a acelorași sectoare care compun mulțimea originală. Volumul său este mai mic decât reuniunea volumelor sectoarelor componente reale, dar mai mare decât suma lor. Este important de menționat că noțiunea de *compactare* exclude prin definiție intersecțiile și consideră densitatea unitară, ceea ce este echivalent cu neglijarea totală a spațiilor inter-sectoare. Prin acest procedeu, suprafața acoperire este în mod corespunzător lărgită, dar păstrează valoarea lui δ_{total} .

➤ Sistemul expert de sfericizare

Se pune problema modului în care o reprezentare exterioară poate fi îmbunătățită astfel încât să rezulte o precizie mai mare și o reprezentare mai

echilibrată dar să se mențină în același timp numărul sferelor constant. Procedul este euristic în ceea ce privește tehnica utilizată pentru împărțirea poligoanelor.

Se utilizează un principiu secvențial în sensul că fiecare nouă reprezentare o îmbunătățește pe cea precedentă. Efectul unei astfel de acțiuni este de a substitui unei mulțimi de sfere o alta, cu un număr mai mare de sfere noi, care să micșoreze porțiunea din reprezentare căreia îi era asociată prima mulțime și astfel să producă o reprezentare mai corectă.

Deci, problema se pune în principiu astfel: considerând pentru început o reprezentare caracterizată de o mulțime de coeficienți de calitate, să se aleagă aceea dintre toate acțiunile posibile, care va duce la cele mai bune reprezentări. Soluția propusă se bazează pe un sistem euristic, numit *sistem expert de sfericizare* a cărui sarcină este să decidă ce acțiune este mai eficientă într-o situație dată.

❖ Cazul bidimensional

Au fost prezentate două metode de calcul a acoperirilor prin cercuri ale marginilor unui poligon și a întregii sale suprafețe. Ambele metode sunt bazate pe divizarea în porțiuni și sub-divizarea în continuare a acelor porțiuni considerate cele mai defavorabile.

În cazul împărțirii conturului poligonului, detalierea acestui procedeu presupune împărțirea acestuia în *laturi eficace*, care înlocuiesc laturile lor reale. Dacă unghiul dintre două laturi vecine este mai mare decât o anumită valoare (spre exemplu 135° în exemplul considerat) aceste laturi sunt considerate ca făcând parte din aceeași latură eficace.

Pentru a justifica această decizie trebuie să se considere ca valabilă afirmația că: *Dacă se atribuie o anumită eroare suprafeței unei părți dintr-un poligon în 2D, va trebui atribuită aceeași eroare fiecărei laturi a poligonului.*

De asemenea trebuie să se țină seama de faptul că atunci când este manevrat un obiect operatorul nu se informează despre toate laturile, laturile sau fețele sale, ci îi este suficient dacă are informații despre un model global al obiectului respectiv.

Este important să se mai facă o observație: *procesul de construire a unei imagini pentru un obiect dat este un proces ierarhizat*, ceea ce revine la a hotărî dacă este necesară o reprezentare foarte complexă sau doar una care satisface necesităților reale.

Pentru exemplul din figura III.46, poligoanele (a) și (b) au forme foarte asemănătoare, dar (a) este un triunghi iar (b) are unsprezece laturi, ceea ce indică faptul că nu există o relație foarte strânsă între forma obiectului și numărul laturilor sale. Rezultă că pare a fi un criteriu bun acela de a asocia o zonă de eroare pentru fiecare latură a triunghiului ca în figura III.46 (c), dar se pune problema eficacității desemnării acestor zone pentru toate cele unsprezece laturi din figura III.46 (b).

Prin introducerea noțiunii de latură eficace vor rezulta numai trei astfel de laturi, (figura III.46 (e)).

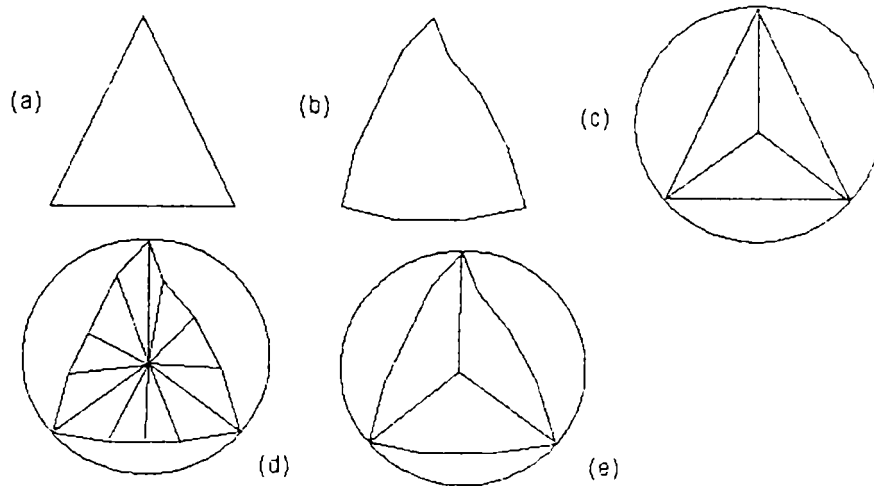


Fig.III.46. Laturi eficace

Deci, conceptul de latură eficace permite dezvoltarea noțiunii de *zonă laterală* a unui poligon, care este independentă de numărul laturilor sale.

Această problemă devine esențială în cazul poligoanelor generalizate (ale căror laturi sunt linii curbe), deoarece laturile curbe se vor aproxima printr-o mulțime de laturi drepte și vârfuri.

Un grup de laturi eficace definesc o *listă*, fiecare din acestea fiind acoperită de un singur cerc. Pentru fiecare latură eficace există o zonă circulară asociată, în sensul că această zonă acoperă latura respectivă.

Existând o anumită acoperire pentru întregul poligon, este definită lista cea mai favorabilă ca fiind aceea care conține cea mai rea latură eficace. Acestei laturi îi este asociată porțiunea de cerc de suprafață maximă. Lista selectată va fi împărțită prin tăiere, prin acel punct al laturii celei mai defavorabile care este definit de jumătatea

unghiului corespunzător arcului de cerc subîntins de latura respectivă în calitate de coardă.

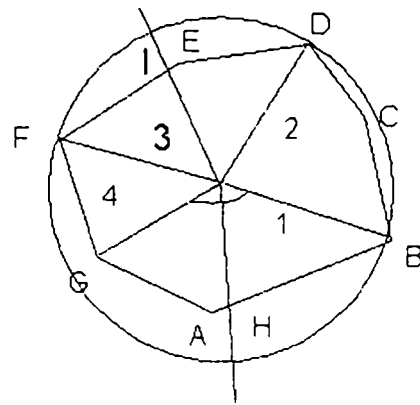


Fig.III.47. Alegerea primelor două puncte de tăiere

În figura III.47 există o singură listă, cele mai defavorabile laturi eficace fiind nr. 1 (G, A, B) și nr.3 (D, E, F). Cum prima listă este circulară, sunt necesare două puncte de tăiere: H și I .

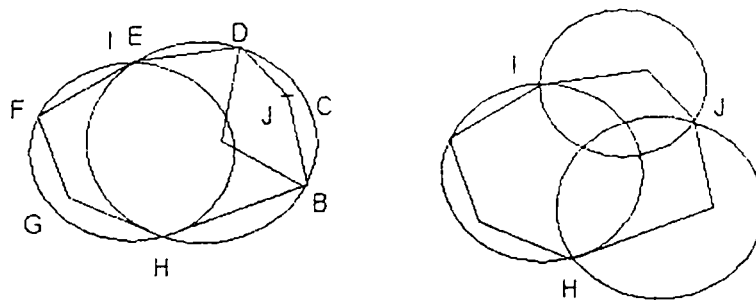


Fig.III.48. Noua selectare și împărțire

Astfel rezultă noi sub-liste (figura III.48): (H, B, C, D, E, I) și (I, F, G, A, H) acoperite cu 2 cercuri. Cea mai defavorabilă listă este acum (I, F, G, H) deoarece cea mai defavorabilă latură eficace este (B, C, D). Punctul J definește o nouă subdiviziune și deci o nouă reprezentare.

Algoritmul laturii eficace dă în general rezultate bune. S-a demonstrat că se implementează în mod nesatisfăcător pentru poligoane regulate cu numărul de laturi $n > 8$. La un astfel de poligon este simplu de observat că toate unghiurile dintre două laturi adiacente sunt mai mari ca 135° .

Deci, conform definiției laturii eficace, întregul perimetru al poligonului, formează o singură astfel de latură. Ori, pentru a implementa algoritmul este necesar să existe cel puțin două laturi pentru a putea începe procesul de împărțire.

Ar putea fi analizată posibilitatea de a micșora valoarea acestui unghi limită de 135° dar aceasta ar însemna să se considere un unghi $\beta = 180^\circ - \frac{360^\circ}{n}$ care ar conduce la un număr de laturi eficace egal cu n , fiecare cu mulțimea eroare aferentă. În orice caz, poligoanele regulate cu multe laturi se adaptează cel mai bine reprezentării prin cerc unic, ceea ce face să nu fie necesară utilizarea mai multor cercuri.

Pentru stabilirea algoritmului acoperirii suprafeței unui poligon, metoda este identică (figura III.49).

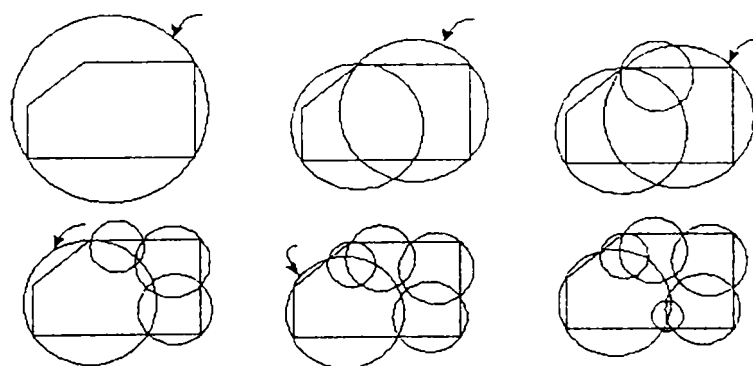


Fig. III.49. Procesul împărțirii în algoritmul acoperirii suprafeței poligonului

În acest caz cercul cel mai defavorabil este cel cu raza cea mai mare. Fiecare sub-poligon este împărțit printr-un segment definit de două puncte. Aceste două puncte se află pe cele mai defavorabile laturi eficace, care sunt acoperite de cercurile cele mai defavorabile.

❖ Cazul tridimensional

Cazul este evident mult mai complex, începându-se cu o singură sferă exterioară și generându-se o secvență de acoperiri pentru fiecare reprezentare.

S-a constatat la analiza anterioară a reprezentării exterioare a obiectelor în 3D că fiecare corp se reprezintă prin mai multe zone corespunzătoare frontierelor laterale și capetelor, ceea ce s-a realizat și geometric în capitolul III.2.2. Astfel sistemul expert de sfericizare trebuie să ia decizii referitoare la secvența reprezentărilor.

Există următoarele tipuri de acoperiri:

1. *Acoperirea frontierelor laterale* exprimată prin sistemul de translate:

$$\{K_{ext}^l + \mathbf{a}_j^l; j=1, 2, \dots, n_l\} \quad (\text{III.2.101})$$

unde: K_{ext}^l este mulțimea sferelor de acoperire exterioară $K_{ext}^l = \{S_i^l; i=1, 2, \dots, m_l\}$, iar vectorul \mathbf{a}_j^l este definit ca:

$$\mathbf{a}_j^l = (2j-1)\mathbf{a}_1 + h_l \cdot \mathbf{u} \quad (\text{III.2.102})$$

în care: $\mathbf{a}_1 = \frac{h_l}{2n_l} \mathbf{u}$.

În aceste expresii, \mathbf{u} este versorul direcției segmentului H prin a cărui translație de baleiere se generează obiectul iar h_l reprezintă înălțimea suprafeței laterale.

În plus, mulțimea de cercuri $\{C_i^l; i=1, 2, \dots, m_l\}$ reprezintă acoperirea frontierelor poligonului care servește drept bază pentru construcția acoperirii 3D.

N_l este numărul total de sfere din acoperirea laterală cu sistemul $\{K_l + \mathbf{a}_j^l\}$. Este evident că :

$$N_l = m_l \cdot n_l \quad (\text{III.2.103})$$

2. *Acoperirea capătului superior* se realizează în mod similar prin:

$$\{K_s + \mathbf{a}_j^s; j=1, 2, \dots, n_s\} \quad (\text{III.2.104})$$

unde: $K_s = \{S_i^s; i=1, 2, \dots, m_s\}$, iar:

$$\mathbf{a}_j^s = (2j-1)\mathbf{a}_s + h_s \cdot \mathbf{u} \quad (\text{III.2.105})$$

$$\text{cu: } \mathbf{a}_s = \frac{h_s}{2n_s} \mathbf{u}$$

Mulțimea de cercuri de acoperire este: $\{C^s_i; i=1, 2, \dots, m_s\}$. N_s este numărul total de sfere din acoperirea capătului superior cu sistemul $\{K_s+\mathbf{a}^s_j\}$. Este evident că :

$$N_s = m_s \cdot n_s \quad (\text{III.2.106})$$

3.Acoperirea capătului inferior se realizează în mod similar cu cel descris pentru capătul superior prin:

$$\{K_i+\mathbf{a}^i_j; j=1, 2, \dots, n_i\} \quad (\text{III.2.107})$$

unde: $K_i = \{S^i_i; i=1, 2, \dots, m_i\}$, iar:

$$\mathbf{a}^i_j = (2j-1)\mathbf{a}_i + h_i \cdot \mathbf{u} \quad (\text{III.2.108})$$

$$\text{cu: } \mathbf{a}_i = \frac{h_i}{2n_i} \mathbf{u} \quad h_s = h_i + h_t$$

Mulțimea de cercuri de acoperire este: $\{C^i_i; i=1, 2, \dots, m_i\}$. N_s este numărul total de sfere din acoperirea capătului superior cu sistemul $\{K_i+\mathbf{a}^i_j\}$. Este evident că :

$$N_i = m_i \cdot n_i \quad (\text{III.2.109})$$

În figura III.38 sunt exemplificate diferite mulțimi de sfere care pot compune o reprezentare, rezultând că pentru realizarea unei reprezentări oarecare este necesar să se deruleze următoarea succesiune de acțiuni:

- **Introducerea acoperirilor pentru capătul inferior.** În prima reprezentare o singură sferă definește K_i iar $N_s = N_i = 0$ cu $n_i = m_i = 1$. După un anumit timp poate începe procesul de acoperire $\{K_k+\mathbf{a}^k_j\}$ (cu semnificația $k=i$ pentru capătul inferior și $k=s$ pentru capătul superior).
- **Introducerea acoperirilor pentru capătul superior.** În mod similar, la un anumit moment acoperirea $\{K_s+\mathbf{a}^s_j\}$ poate fi introdusă. Aceasta revine la

împărțirea frontierei exterioare a obiectului în trei regiuni diferite: două capete și o parte centrală; $\{K_l + \mathbf{a}_j^l\}$ va fi redefinită.

- **Modificarea mulțimii K_l .** Aceasta constă în mărirea numărului de sfere din mulțimea generată pentru acoperirea frontierelor laterale. Această acțiune este echivalentă cu modificarea mulțimii $\{C^l_j\}$.
- **Modificarea mulțimii K_s .** Aceasta constă în mărirea numărului de sfere din mulțimea generată pentru acoperirea capătului superior. Această acțiune este echivalentă cu modificarea mulțimii $\{C^s_j\}$.
- **Modificarea mulțimii K_l .** Aceasta constă în mărirea numărului de sfere din mulțimea generată pentru acoperirea capătului inferior. Această acțiune este echivalentă cu modificarea mulțimii $\{C^i_j\}$.
- **Modificarea secvenței $\{\mathbf{a}_j^l\}$.** O nouă secvență de acest tip este definită, crescând numărul vectorilor ce compun sistemul de translatate $\{K_l + \mathbf{a}_j^l; j=1, 2, \dots, n_l\}$.
- **Modificarea secvenței $\{\mathbf{a}_j^s\}$.** O nouă secvență de acest tip este definită, crescând numărul vectorilor ce compun sistemul de translatate $\{K_s + \mathbf{a}_j^s; j=1, 2, \dots, n_s\}$.

❖ Structura sistemului expert de sfericizare

Realizarea unei reprezentări echilibrate și optime se poate obține prin detectarea zonei cu cea mai necorespunzătoare reprezentare în scopul îmbunătățirii sale și a asigurării posibilității de a realiza o reprezentare cât mai exactă. Este de menționat faptul că în timp ce reprezentarea unei zone se îmbunătățește, se poate înrăutăți reprezentarea altei zone a obiectului. De aceea sfericizatorul trebuie să funcționeze în structura unui sistem cu reguli riguroase.

Sfericizatorul folosește principiul detectării ariei reprezentării cu cea mai slabă calitate, pentru ca ulterior reprezentarea să fie redefinită. Coeficienții de calitate care contribuie la luarea deciziilor sunt: δ_{ext}^l și ε_{ext}^l pentru frontierele laterale, δ_{ext}^{bsup} și ε_{ext}^{bsup} pentru capătul superior, respectiv δ_{ext}^{binf} și ε_{ext}^{binf} pentru capătul inferior și δ_{total} care caracterizează întreaga reprezentare exterioară.

Procesul de sfericizare folosește un algoritm pas cu pas, realizând generații de noi reprezentări ce sunt îmbunătățiri ale celor anterioare. Prima reprezentare este cea

în care întregul obiect este inclus într-o singură sferă. A doua reprezentare se va obține din aceasta aplicând una dintre reguli.

III.2.3.2. Reprezentarea interioară

O sferă interioară este definită ca fiind complet conținută de obiectul ce este reprezentat. Adică:

$$\forall P \in S_j \quad \exists O_i \in O \mid P \in O_i \quad (\text{III.2.110})$$

unde:

$O = \{O_i, i = (1, \dots, m)\}$ este mulțimea obiectelor

$S = \{S_j, j = (1, \dots, n)\}$ este mulțimea sferelor de reprezentare.

Se urmărește definirea unei mulțimi n de sfere interioare astfel încât ele să poată fi privite ca o reprezentare interioară a mulțimii de m obiecte O . Reprezentarea poate fi îmbunătățită, astfel ca umplerea să devină echilibrată și cea mai potrivită pentru sistemul dat.

Această definiție poate fi considerată o generalizare a conceptului de acoperire cu sfere, dacă este înlăturată condiția de intersecție:

$$\bigcap S_j \supset O \quad S_j \cap S_k = C_{jk} \quad (\text{III.2.111})$$

unde mulțimea C_{jk} poate fi vidă sau nu.

Se vor presupune aceleași restricții asupra naturii corpurilor ca și în cazul reprezentării exterioare. Deci, în mod similar se va preciza sistemul de translate $\{K + \mathbf{a}_j; j = 1, 2, \dots\}$ pentru a fi definită reprezentarea interioară generalizată, căreia i se asociază noțiunea de densitate pentru un obiect : $\rho(K + \mathbf{a}_j)$.

Tinând seama de aceleași restricții ca în cazul acoperirii se poate scrie că:

$\rho(K + \mathbf{a}_j) \leq 1$ fiind definită ca raport al măsurii reuniunii mulțimilor de umplere și măsura obiectului.

➤ Cazul plan

Problema mai simplă este cea a umplerii în spațiul bidimensional. Se poate preciza existența unei umpleri generalizate pentru poligoane prin intermediul cercurilor, printr-un procedeu similar cu cel al acoperirii.

❖ Segmentele Voronoi și cercurile maxime

Pentru a realiza reprezentării interioare este necesară prezentarea conceptului de *Diagramă Voronoi*. Inițial, o astfel de diagramă a fost realizată numai pentru distribuirea punctelor în spațiu în felul următor:

Se presupune că se studiază mulțimea de puncte a_1, a_2, \dots și că există un număr pozitiv R astfel încât pentru orice punct x din spațiu se poate preciza un punct a_i din șirul anterior, a cărui distanță $|x-a_i|$ față de x este mai mică decât R .

Fiecărui punct a_i i se asociază mulțimea $\Pi(a_i)$ a tuturor punctelor x a căror distanță de la a_i este egală cu distanța minimă la punctele lui $\{a_j\}$. În acest caz $\Pi(a_i)$ este mulțimea punctelor x care satisfac relația:

$$|x - a_i| < |x - a_j| \quad j \neq i \quad (\text{III.2.112})$$

Se poate arăta că mulțimea $\Pi(a_i)$ astfel definită, reprezintă un poliedru convex, iar fiecare punct din spațiu care aparține cel puțin unui astfel de poliedru și cel mult la două sau mai multe poliedre, este situat pe frontierele lor.

Astfel, diagrama Voronoi precizată de șirul $\{a_j\}$ definește o împărțire a spațiului în regiuni $\Pi(a_i)$ care sunt asociate fiecărui punct a_i astfel încât regiunea respectivă conține toate acele puncte din spațiu care sunt mai apropiate de a_i decât orice punct aparținând șirului $\{a_j\}$ (figura III.50).

Utilitatea diagramei Voronoi este evidentă pentru numeroase aplicații. Astfel pentru n puncte plasate într-un plan real, a fost găsit pentru determinarea sa un algoritm simplu, bazat pe metrica euclidiană, precizând cercul de diametru minim care le cuprinde. Prin intermediul său se pot rezolva și alte probleme de geometrie: cea mai apropiată pereche de puncte, cel mai apropiat punct vecin, etc.

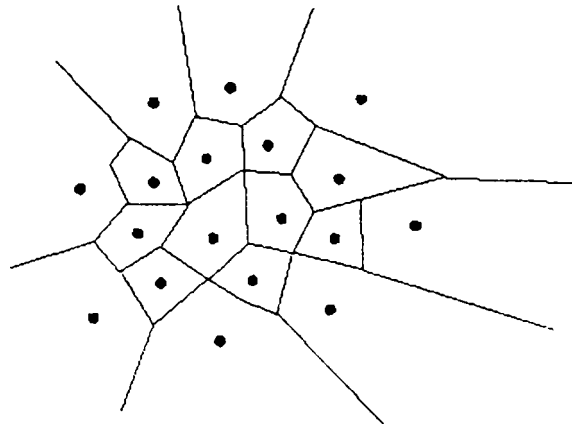


Fig.III.50. Diagrama Voronoi pentru o mulțime de puncte

În domeniul planificării mișcării roboților mobili, aplicațiile bazate pe diagrama Voronoi sunt multiple. Pe baza sa s-a rezolvat problema drumului optim al unui punct între obstacole poliedrice; de asemenea ea este folosită la planificarea traiectoriilor unui obiect într-un plan aplicând criterii euristice pentru a decide dacă mișcarea optimă este o translație, o rotație sau o mișcare plan-paralelă.

În domeniul reprezentării corpurilor important este tipul de *diagramă Voronoi generalizată*. Pentru n laturi ale unui poligon convex, aceasta poate fi definită ca împărțire a poligonului în n regiuni disjuncte Π_i , fiecare dintre ele fiind asociată unei anumite laturi L_i a poligonului astfel încât toate punctele P care aparțin acestei regiuni să se afle pe cea mai apropiată latură. Adică:

$$\Pi_i = \{P; d(P, L_i) \leq d(P, L_j), j=1, 2, \dots, n\} \quad (\text{III.2.113})$$

în care: $d(P, L_j)$ reprezintă distanța euclidiană de la punctul P la dreapta suport a laturii L_j .

Trebuie notat că în cazul general unde succesiunea $\{L_j\}$ poate reprezenta orice mulțime de segmente, se poate defini $d(P, L_i)$ ca distanța dintre P și proiecția sa Q pe dreapta suport a lui L_i , dacă Q aparține lui L_i ; în caz contrar ea trebuie definită ca:

$$d(P, L_i) = \min(d(P, A), d(P, B)) \quad (\text{III.2.114})$$

unde: A și B sunt capetele segmentului L_i .

Dacă segmentele succesiunii L_1, L_2, \dots, L_n formează laturi adiacente ale unui poligon convex, definiția dată pentru $d(P, L_i)$ este suficientă.

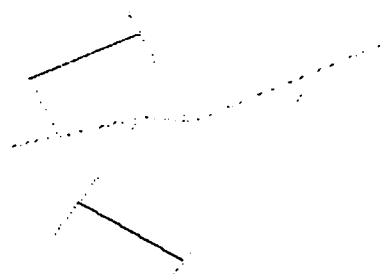


Fig.III.51. Diagrama Voronoi pentru două segmente

O altă diferență între cazul general și cel al poligoanelor este aceea că în ultimul caz regiunile Voronoi vor fi poligoane convexe, iar în cazul general anumite porțiuni ale frontierelor vor fi curbe. Conform definiției parabolei, este evident că porțiunile de diagramă corespunzătoare unui segment de dreaptă aparținând unui poligon și unui vârf al altui poligon sunt segmente de parabolă (figura III.51).

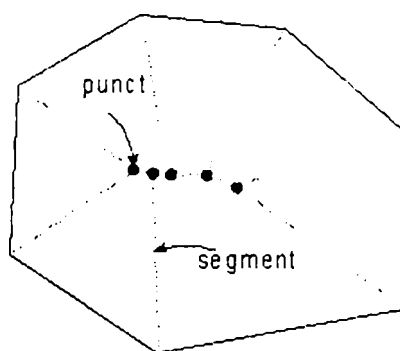


Fig.III.52. Diagrama Voronoi generalizată pentru un poligon convex

Dreapta comună a două regiuni Voronoi se numește un *segment Voronoi* (V_i) iar punctul comun a trei laturi *punct Voronoi* (figura III.52). În cazul unui poligon convex toate V_i sunt poligoane convexe

Diagrama Voronoi are o proprietate importantă: *mulțimea finită de puncte Voronoi este locul geometric al centrelor tuturor cercurilor tangente la cel puțin trei laturi ale poligonului.*

Aceste cercuri sunt denumite *cercuri maxime* C_i . Fiind calculate aceste centre, segmentele Voronoi sunt ușor de găsit prin conectarea lor cu alte segmente de dreaptă.

Teorema III.17: *Fiind dat un poligon convex, o mulțime finită de segmente din interiorul poligonului poate fi definită astfel încât să fie verificate condițiile:*

1. Pentru fiecare punct P aparținând unuia dintre segmente, există un cerc $C(P)$ cu centrul în P și tangente la cel puțin două laturi ale poligonului; în plus acest cerc se afla complet în interiorul poligonului (cu excepția punctelor de tangență).
2. Mulțimea infinită de cercuri obținute prin precizarea câte unui cerc $C(P)$ pentru fiecare punct P al segmentelor verifică faptul că există o acoperire a poligonului prin intermediul cercurilor care sunt interioare poligonului.
3. Aceasta este reprezentarea exterioară (acoperirea) perfectă și ea poate fi considerată limita unei reprezentări interioare (umpleri) generalizate cu cercuri, când numărul acestora tinde la infinit.

Definiție: *Segmentele Voronoi reprezintă locul geometric al tuturor punctelor interioare ce se află la aceeași distanță de cel puțin două laturi.*

Segmentele Voronoi au următoarele proprietăți:

1. Toate sunt porțiuni ale bisectoarei a două laturi de poligon.
2. Două laturi de poligon adiacente definesc un segment Voronoi cu originea în vârful comun, care se va numi *segment terminal*.
3. Dacă două laturi nu sunt adiacente și deci nu se intersectează într-un vârf comun, bisectoarea lor poate sau nu să conțină un segment Voronoi.
4. Segmentele Voronoi se intersectează în punctele unde se întâlnesc cel puțin *trei* segmente. Aceste puncte reprezintă centrele cercurilor tangente la cel puțin trei laturi de poligon.

Algoritm:

Pentru a construi un segment Voronoi se poate utiliza un algoritm bazat pe următoarele considerente:

Fie dat un poligon convex definit de cele p laturi ale sale L_1, L_2, \dots, L_p .

1. Pentru toate combinațiile posibile de trei laturi se caută cercul tangent la laturile respective. Centrul său este punctul de intersecție al celor trei bisectoare construite pentru trei perechi de laturi.
2. Se verifică cercurile astfel determinate pentru a se stabili care dintre ele se află în interiorul poligonului și se elimină cele care nu sunt. Această primă mulțime de cercuri interioare va fi *mulțimea de cercuri maxime* $C^m = \{C_i; i=1, 2, \dots, m_m\}$ și cu ajutorul ei se vor defini segmentele Voronoi. Centrele acestor cercuri coincid cu intersecțiile a câte trei segmente Voronoi.
3. Deoarece fiecare centru al unui cerc maxim este asociat la trei bisectoare ce-l definesc, iar un segment corespunde întotdeauna unei porțiuni de bisectoare, fiecare centru al unui cerc maxim corespunde intersecției a trei segmente Voronoi.
4. O bisectoare nu poate conține mai mult de un segment Voronoi, existând două situații posibile:
 - a) Dacă bisectoarea începe la un vârf al poligonului, segmentul mărginit de un centru al unui cerc maxim are proprietatea că cercurile tangente la ambele laturi L_1 și L_2 nu pot fi mărite, fiind mărginite de latura a treia L_3 (figura III.53)

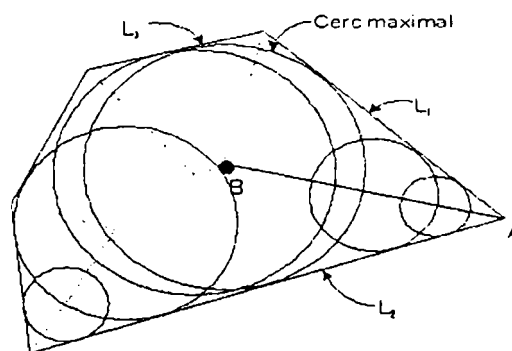


Fig.III.53. Segment mărginit de centrul unui cerc maxim B și vârful A al poligonului

b) Dacă bisectoarea corespunde la două laturi ne-adiacente, originea lor va fi în punctul de intersecție Q care nu este un vârf al poligonului iar cercurile tangente ambelor laturi vor fi cu atât mai mari cu cât centrul este mai departe de originea menționată. Mulțimea acestor cercuri va avea două limite: una reprezentând cercul cel mai mic și una reprezentând cercul cel mai mare (figura III.54). Convexitatea poligonului garantează faptul că pot exista numai cercurile interioare cu centrele pe un interval continuu aparținând unei bisectoare.

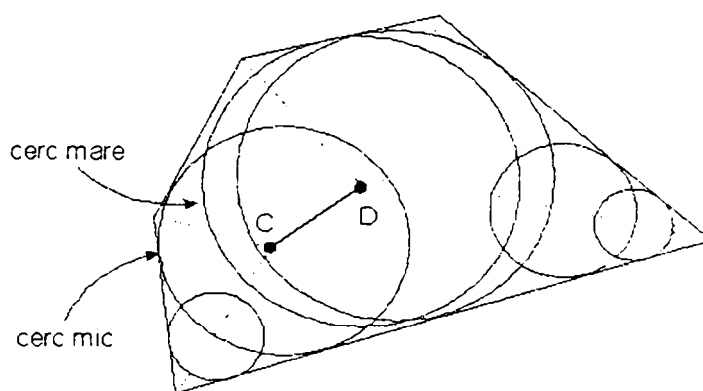


Fig.III.54. Segment mărginit de centrele a două cercuri C și D

5. Conform ultimului rezultat, bisectoarea ce corespunde laturilor ne-adiacente va avea ori două centre de cercuri maximale ori nici unul. Dacă are două, ele vor defini un segment care nu va fi ultimul. Un caz limită poate apărea când ambele centre coincid într-un singur punct. Pot apărea de asemenea cazuri ale unor poligoane regulate cu multe laturi când diferite cercuri maximale au același centru și atunci toate trebuie calculate ca fiind unic.
6. Bisectoarea a două laturi adiacente ale unui poligon va conduce la un segment terminal ale cărui capete sunt vârful comun al laturilor și respectiv, centrul cercului maximal asociat bisectoarei.
7. Bisectoarea a două laturi ne-adiacente va defini un segment numai în cazul când pe ea există două centre de cercuri maximale.

Astfel, pentru un poligon dat există o mulțime de segmente Voronoi și o mulțime de cercuri maxime. Figura III.55 prezintă exemple pentru diferite poligoane. Este clar că diagrama Voronoi obținută prin reprezentarea segmentelor este în mod sigur digrama generalizată pentru laturile poligonului și pentru cercurile maxime care au centrele chiar în punctele Voronoi.

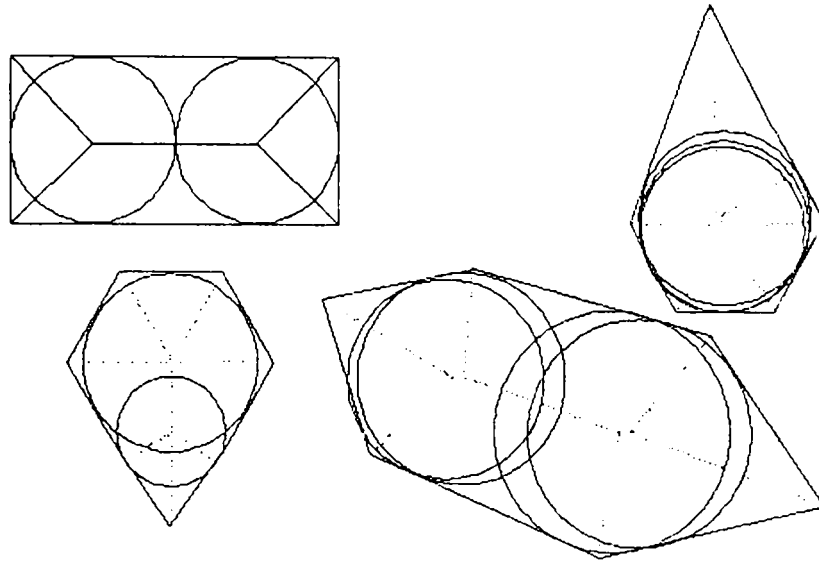


Fig.III.55. Exemple de digrame Voronoi și cercuri maxime

Alte rezultate importante pot fi deduse din situațiile analizate anterior:

1. Nu poate exista nici un cerc interior unui poligon mai mare decât cel cu cea mai mare rază conținut în mulțimea $C^m = \{C_i; i=1, 2, \dots, m\}$.
2. Pentru un segment dat V_i , elementele mulțimii de cercuri cu centrele pe segmentul dat și tangente la laturile ce definesc segmentul $\{C(P); P \in V_i\}$, au o limită superioară și inferioară $C(A_i)$ și $C(B_i)$, luând razele ca măsuri ale cercurilor, unde A_i și B_i sunt capete ale segmentului V_i . Când segmentul este terminal, A_i va coincide cu un vârf al poligonului și raza cercului $C(A_i)$ este nulă. O primă reprezentare interioară este obținută prin acoperirea generalizată formată de mulțimea de cercuri maxime C^m . Această reprezentare are o caracteristică interesantă: ea *umple* interiorul poligonului astfel încât toate cercurile care realizează această reprezentare interioară sunt tangente la cel puțin trei laturi.

❖ Umplerea perfectă

Dacă un cerc $C(P)$ este atribuit fiecărui punct al unui segment Voronoi, astfel încât centrul său să fie în acest punct și el să fie tangent la două laturi ale poligonului, se obține o *umplere perfectă* a poligonului prin cercuri interioare definite de mulțimea completă C_∞ a tuturor cercurilor atribuite punctelor de pe toate segmentele Voronoi (figura III.56):

$$C_\infty = \{C(P); P \in V_i; i = 1, 2, \dots, v\} \quad (\text{III.2.115})$$

unde v este numărul de segmente Voronoi.

C_∞ este o acoperire corespunzătoare pentru poligon deoarece fiecare punct al poligonului este conținut cel puțin într-unul din cercuri. Densitatea sa este unitară, reprezentarea fiind în același timp și exterioară și interioară (și acoperire și umplere). Se observă că umplerea generalizată este o acoperire când atinge perfecțiunea, adică în cazul în care densitatea sa este unitară.

Intr-adevăr, dacă se definește mulțimea C_∞ , ca un șir $C_\infty = C_1, C_2, \dots$ și o submulțime a sa $C(m) = \{C_1, C_2, \dots, C_m\}$, atunci definițiile valabile pentru C_∞ și, respectiv, densitatea sa sunt :

$$C_\infty = \lim_{m \rightarrow \infty} C(m) \quad \text{și} \quad \rho(C_\infty) = \lim_{m \rightarrow \infty} \rho(C(m)) = 1 \quad (\text{III.2.116})$$

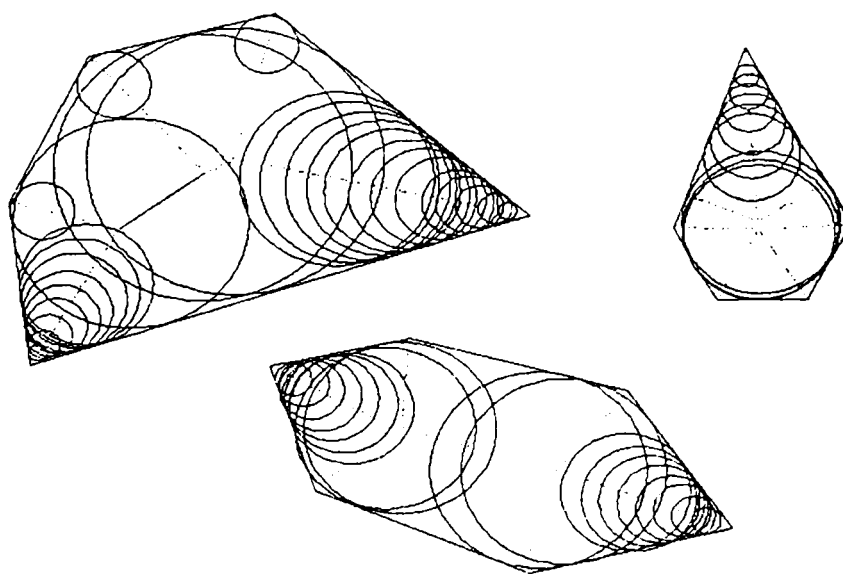


Fig.III.56. Trei variante ale unor umpleri aproape perfecte

❖ Construcția reprezentării interioare

Umplerea perfectă definită prin relațiile (III.2.115) nu este totuși foarte potrivită pentru scopuri practice, deoarece ea se compune dintr-un număr infinit de cercuri. Este necesar un procedeu euristic pentru a selecta în cadrul acestei mulțimi infinite, un număr m de cercuri ce reprezintă cel mai bine poligonul. Figura III.57 propune o schemă constructivă.

Problema aceasta poate fi enunțată astfel:

Este necesară o astfel de submulțime $C_1, C_2, \dots, C_m, \dots$ din cadrul lui C_∞ încât împachetarea $C(m)$ compusă din primele m elemente C_1, C_2, \dots, C_m ale lui C_∞ să devină o reprezentare interioară cu m cercuri care să fie echilibrată și optimă pentru numărul respectiv de cercuri, dar care se poate îmbunătăți prin adăugarea elementelor C_{m-1}, C_{m-2}, \dots .

Primele elemente ale lui C_∞ vor fi luate din mulțimea de cercuri maximale $C^m = \{C_i; i=1, 2, \dots, m_n\}$ astfel încât $C(m_n) = C^m$. În plus, trebuie menționat: $r_i \geq r_j$ ($i < j \leq m_m$), unde r_i este raza cercului C_i .

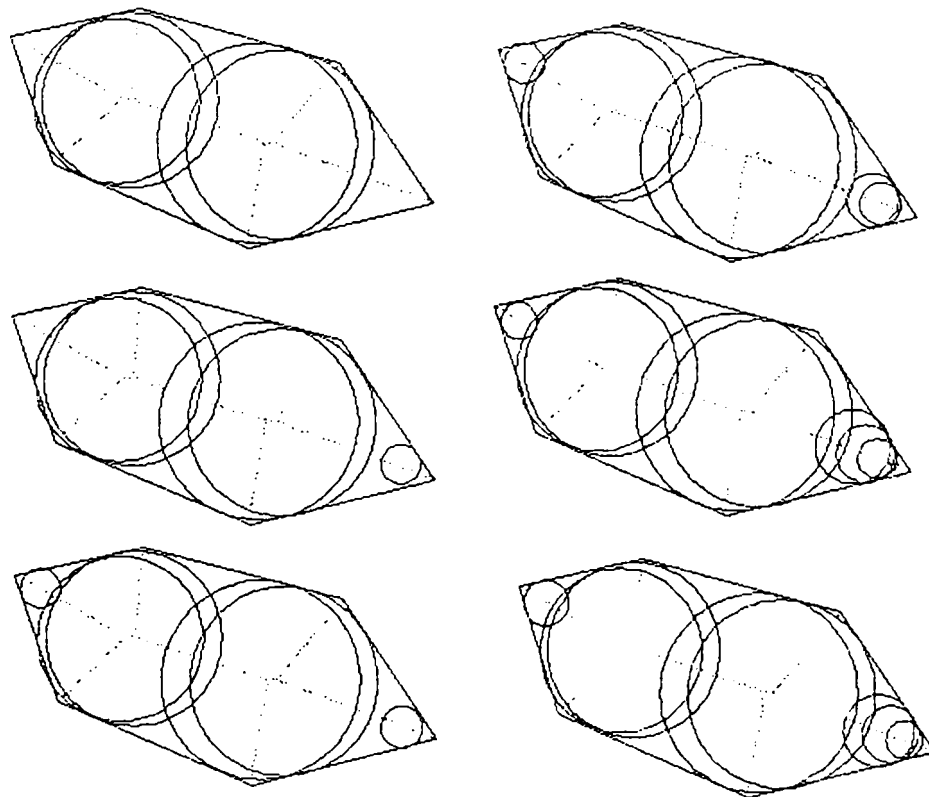


Fig.III.57. Construcția reprezentării interioare

Pentru $m > m_n$, dacă se numește $C(V_i)$ submulțimea lui C_∞ compusă din acele cercuri ce au centrul în V_i , excluzându-le pe cele maxime, atunci:

$$C_\infty = C^m \cup \{C(V_i); i = 1, 2, \dots, v\} \quad (\text{III.2.117})$$

Elementele infinite ale $C(V_i)$ sunt notate cu C_{i1}, C_{i2}, \dots astfel încât elementele în fiecare din acestea sunt selectate convenabil.

Pentru a defini $C(m)$ se pornește de la $C(m_n)$ și $m=m_n$ și se selectează cel mai nefavorabil segment V_i , adăugând primul element C_{i1} din mulțimea sa de cercuri asociate $C(V_i)$ pentru a obține $C(m_m+1)$ astfel încât: $C_{m+1} = C_{i1}$.

În această etapă, cel mai nefavorabil segment V_i este selectat din nou și un anumit număr k de cercuri din $C(V_i)$ sunt adăugate la $C(m)$ pentru a se obține $C(m+k)$. Procedura se repetă pentru orice $C(m)$.

Rezultă astfel că o reprezentare interioară se reduce la următoarele probleme:

- Cum să se aleagă elementele în mulțimea $C(V_i)$;
- Cum să se selecteze cel mai nefavorabil segment într-un moment dat al procesului;
- Ce număr k de cercuri din $C(V_i)$ să se selecteze astfel încât să aparțină la $C(m+k)$.

Este astfel evidentă diferența dintre reprezentarea exterioară și cea interioară: la cea din urmă îmbunătățirea reprezentării nu se face prin substituirea unor cercuri cu altele având alte raze, ci prin adăugarea de cercuri noi.

Trecerea la cazul tridimensional se face pe aceleași principii la umplere ca și la acoperire. Se începe cu generarea câte unui cilindru pentru fiecare cerc C_i aparținând umplerii generalizate pentru poligonul generator al obiectului spațial. Apoi, pentru fiecare astfel de cilindru se realizează o umplere cu sfere, astfel ca densitatea împachetării să fie cât mai mică. Criteriile de realizare a unei împachetări echilibrate și optime sunt aceleași ca și la acoperire.

Un criteriu euristic adițional corespunzător schemei de reprezentare se bazează pe faptul că fiecare cerc din mulțimea C_∞ este tangent la cel puțin două laturi ale poligonului. Atunci există două sau trei cercuri care sunt tangente la aceleași laturi ale

conturului poligonului generator original, ceea ce constituie o garanție că împachetarea are calitatea corespunzătoare.

➤ **Îmbunătățirea reprezentării**

Reprezentarea interioară se îmbunătățește prin adăugarea de noi sfere (sau cercuri) interioare la umplerea deja realizată într-o etapă precedentă. Metodologia folosită pentru selectarea celor mai potrivite sfere (sau cercuri) din ansamblul tuturor posibilităților, este prezentată separat pentru cazul 2D și, respectiv 3D.

❖ **Cazul bidimensional**

Fiecărui segment Voronoi V_i i se va asocia un număr întreg m_i , astfel încât inițial $m_i = 1, i = 1, 2, \dots, v$. Aceste m_i vor fi funcție de valoarea lui m pentru ultima reprezentare calculată astfel încât pentru un m dat:

$$C_{ij} \in C(m), \quad j = 1, 2, \dots, \frac{m_i(m_i - 1)}{2} \quad (\text{III.2.117})$$

unde C_{ij} este elementul j din $C(V_i)$.

Dacă V_i este un segment terminal, un alt parametru d_i este de asemenea asociat cu el (figura III.58) fiind definit ca:

$$d_i = \min [d(A_i, D_{ij}) - r_{ij}; \quad j = 1, 2, \dots, m-1] \quad (\text{III.2.118})$$

unde: A_i este capătul lui V_i care este și un vârf al poligonului

r_{ij} este raza lui C_{ij}

D_{ij} este centrul lui C_{ij} .

Segmentul Voronoi cel mai defavorabil este ales utilizând următoarea succesiune de operații:

1. Este selectat segmentul terminal cu cea mai mare valoare pentru raportul $\frac{d_i}{m_i}$:

$$\delta = \max \left[\frac{d_i}{m_i}; \quad i = 1, 2, \dots, p \right] \quad (\text{III.2.119})$$

unde p este numărul de segmente terminale. Fie V_i un segment astfel selectat, ca fiind cel mai defavorabil ($i = t$).

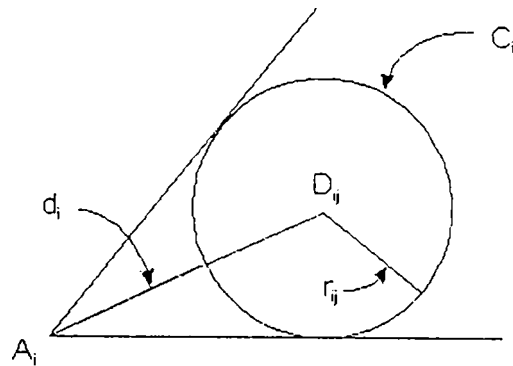


Fig.III.58. Parametrul d_i asociat segmentului terminal

2. Se selectează segmentul ne-terminal cu cea mai mare valoare pentru $\frac{l_i}{m_i}$:

$$\varepsilon = \max \left[\frac{l_i}{m_i}; \quad i = p + 1, p + 2, \dots, v - p \right] \quad (\text{III.2.120})$$

unde: $l_i = d(A_i B_i)$ este lungimea lui V_i . Fie V_s cel mai defavorabil segment ne-terminal ($i = s$).

3. În final, dacă:

$$\frac{l_i}{n_i} \geq k \cdot \varepsilon \quad (\text{III.2.121})$$

atunci V_i este cel mai defavorabil segment Voronoi; dacă nu, atunci este V_s .

k este un coeficient subunitar care favorizează alegerea cu prioritate a unui segment terminal, datorită faptului că segmentele neterminale devine mai târziu aparente decât cele terminale.

O dată selectat cel mai defavorabil V_i , se majorează cu o unitate m_i și astfel $k = m_i - 1$ cercuri sunt adăugate la $C(m)$. Centrele lor sunt distribuite într-un anumit mod de-a lungul lui V_i între A_i și B_i . Pentru a defini această distribuție, trebuie precizate două cazuri:

- Dacă segmentul este V_s , neterminal, centrele sunt distribuite omogen de-a lungul segmentului la distanțele $\varepsilon = \frac{l_s}{n_s}$.
- Dacă segmentul este V_t , terminal, centrele sunt localizate la distanțe succesive date de regula:

$$\delta_k = l_r \cdot \log\left(1 + \frac{9k}{m_i}\right) \quad k = 1, 2, \dots, m-1 \quad (\text{III.2.122})$$

aceste distanțe fiind măsurate începând de la B_t , care este capătul segmentului și reprezintă un punct Voronoi (figura III.59).

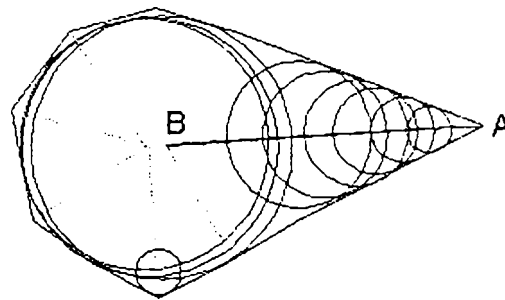


Fig.III.59. Distribuție de cercuri în segmentul terminal

Pentru a justifica ultimele afirmații este necesar să se revadă criteriul euristic care le motivează. Distincția între segmentele terminale și neterminale este o consecință a faptului că numărul de cercuri ce trebuie plasate pe un segment neterminal pentru a obține o reprezentare bună este mai mic decât cel pentru un segment terminal cu aceeași lungime pentru că:

- Segmentele terminale sunt mereu în interiorul poligonului, departe de vârfuri și laturi;
- Ca o consecință a afirmației anterioare, cercurile centrate pe ele vor avea comparativ razele mai mari.

Rezultă că vor fi îndeajuns de puține cercuri pentru umplerea unei importante părți a poligonului. Factorul k folosește la selectarea celor mai defavorabile segmente terminale; în plus, criteriile pentru selectarea celui mai defavorabil segment sunt:

$$\frac{d_i}{m_i} \text{ pentru cel terminal și } \frac{l_i}{n_i} \text{ pentru cel neterminal.}$$

O ultimă observație este legată de criteriul euristic utilizat pentru a evidenția cercurile localizate în apropierea vârfurilor poligonului (figura III.59), deoarece în aceste zone cercurile trebuie să fie mai mici și vor umple o porțiune mai mică din interiorul poligonului. Aceasta se produce deoarece s-a utilizat o distribuție uniformă pentru cercurile segmentelor neterminale și una logaritmică, mai potrivită, pentru cazul celor terminale.

❖ Cazul tridimensional

Pentru o cât mai corectă reprezentare a fost ales un criteriu ce favorizează umplerile distribuite uniform în volumul corpului. S-a dat însă atenție și faptului că zonele înconjurătoare vârfurilor și laturilor obiectului trebuie să fie corect reprezentate.

S-a ales o schemă mai puțin complexă decât cea pentru reprezentarea exterioară, realizând o împărțire a corpului în diferite zone și atribuind câte o umplere parțială fiecăreia, astfel încât mai târziu, prin utilizarea sistemului expert de sfericizare, să se poată decide care dintre acțiunile posibile este cea mai potrivită.

Acest mod de a aborda problema a permis să se aleagă pentru reprezentare cea mai puțin complexă metodă. Nu trebuie pierdut din vedere faptul că reprezentarea interioară este oarecum subordonată celei exterioare. Această afirmație este susținută și de observația că sferele exterioare sunt cele care servesc la garantarea evitării coliziunilor. Pentru a împiedica obiectele să se lovească unul de altul, este nevoie să se asigure inexistența coliziunilor posibile, ceea ce va accelera luarea unei decizii.

Conform celor afirmate, în primul rând este calculată o mulțime $C(m)$ corespunzătoare. Sistemul de translate: $\{S_i + \mathbf{a}_{ij}; j = 1, 2, \dots, n_i; i=1, 2, \dots, m\}$ este generat din el; această acțiune va realiza o împachetare completă pentru obiect. Pentru o valoare dată a lui i , sistemul $\{S_i + \mathbf{a}_{ij}\}$ va fi compus din n_i sfere.

Așa cum s-a prezentat în capitolul anterior, n_i depinde de h , r_i (este raza lui S_i) și λ . Deoarece h nu se schimbă dacă λ este menținut constant, numărul de sfere pentru sistemul generat de S_i este invariabil.

Deci, pentru a îmbunătăți o reprezentare interioară tridimensională, se aplică un procedeu de rafinare umplerii generalizate pentru poligonul generator al corpului conform schemei prezentate. Adică, se mărește $C(m)$ prin pași succesivi în C_∞ . Pentru fiecare S_i nou, sunt generate sferele din sistemul $\{S_i + \mathbf{a}_{ij}\}$. Coeficientul de densitate λ se menține invariabil pentru toate reprezentările obiectului, schimbându-se doar în cazul în care se dorește o optimizare a sistemului.

➤ Convergență și ierarhizare

Dubla reprezentare a oricărui obiect are importanta proprietatea de a fi *convergentă*. Acest concept trebuie înțeles în cadrul schemei secvențiale ce permite definirea unei succesiuni de reprezentări astfel încât fiecare o îmbunătățește pe cea precedentă. Rezultatul este că o reprezentare cu o eroare atât de mică cât se dorește, se poate obține cu ajutorul altor elemente din succesiunea de reprezentări. Astfel se poate defini o dublă *ierarhizare* a reprezentărilor pentru a descrie corpul cu diferite nivele de precizie.

Convergența e foarte importantă, deoarece ea este trăsătura caracteristică a modului de reprezentare cu sfere care îl diferențiază de alte moduri de reprezentare. Este însă, obligatoriu să se observe că reprezentarea sferică este interesantă din multe puncte de vedere, dar nu este exhaustivă, pentru că nu se pretează la absolut toate corpurile. Totuși, cu anumite limite, modelul reprezentării prin sfere, este aproape general, practic orice obiect putând fi astfel abordat cu precizia impusă. Toată problema este cât de multe sfere implică reprezentarea. Această afirmație este probată de modul în care a fost conceput programul de sfericizare pentru corpuri geometrice de diferite forme clasice, ce în fond pot aproxima suficient de bine corpurile reale, astfel încât o problemă de reprezentare în vederea planificării mișcărilor fără coliziuni să fie corect rezolvabilă.

❖ Convergența reprezentării exterioare

Fiecare pas din procesul de sfericizare produce două serii de date. Prima serie cuprinde trei categorii de date ce descriu acțiunea desfășurată:

- a) numărul reprezentării sau al pasului din proces;
- b) codul ce identifică regula aplicată;
- c) scurta descriere a acțiunii și scopului regulii;

A doua serie, include date ce caracterizează fiecare reprezentare: δ_{total} , $\delta_{lateral}$, N_b , N_s , m_b , m_s , n_b , n_s , δ_{ext}^l , δ_{ext}^{bsup} , δ_{ext}^{binf} , ε_{ext}^{bsup} , ε_{ext}^{binf} , ε_{ext}^l , etc.

Din exemple practice, pentru care algoritmul de reprezentare a fost aplicat ca atare atât pentru acoperire cât și pentru împachetare, se poate concluda că secvența de reprezentări obținută este rezultatul unei aplicări repetate a regulilor sistemului expert de sfericizare care converge spre o reprezentare practic lipsită de erori.

Adică, dacă un șir de numere reale este definit ca fiind compus din valori ale lui δ_{total} pentru reprezentări succesive: $\{\delta_{total\ n}\} = \{\delta_{tot1}, \delta_{tot2}, \delta_{tot3}, \dots\}$, atunci acest șir este la limită convergent spre zero: $\lim_{n \rightarrow \infty} \delta_{total\ n} = 0$

Șirul $\{\delta_{total\ n}\}$ nu este totuși monoton deoarece nu este îndeplinită condiția: $\delta_{toti} \geq \delta_{toti+1}$, $\forall i = 1, 2, \dots$. Acest fapt nu este important deoarece el este în mod clar un șir Cauchy. Într-adevăr pentru orice număr real ε oricât de mic, exista cu siguranță o reprezentare n_o astfel încât:

$$|\delta_{totali} - \delta_{totalj}| < \varepsilon, \quad \forall i \geq n_o, j \geq n_o \quad (\text{III.2.123})$$

Acesta este și motivul pentru care în succesiunea de reprezentări, valorile lui δ_{total} devin din ce în ce mai apropiate.

S-a constatat prin încercări de reprezentare prin sfere ale diferitelor corpuri că se poate obține o eroare mai mică de 10% cu o reprezentare exterioară compusă din mai puțin de 20 sfere. Acesta reprezentare corespunde primului minim al coeficientului de calitate δ .

De aceea de cele mai multe ori un obiect va fi reprezentat prin mai puțin de 20 de sfere. Doar în cazuri speciale e nevoie de mai multe sfere, deoarece modelul nu limitează numărul maxim de sfere.

La realizarea efectivă a programului de reprezentare se impune o eroare de reprezentare și rezultă numărul de sfere pentru acoperire și umplere, astfel încât să fie satisfăcut acest criteriu.

❖ Convergența reprezentării interioare

Un rezultat similar se poate stabili pentru sferele interioare. În cazul reprezentării bidimensionale convergența este evidentă ca un corolar direct al faptului

că generarea în acest caz conduce la o umplere perfectă (cu densitatea de împachetare egală cu unitatea).

Mulțimea eroare E poate fi definită pentru a caracteriza o reprezentare interioară, la fel ca în cazul exterior: $Q \in E$ dacă și numai dacă $\exists O_i \in O / Q \in O_i$ și $Q \notin S_j, \forall j (j = 1, 2, \dots)$. Sau, mai simplu: $Q \in O$ și $Q \notin S$, în care:

$O = \{O_i, i=1, \dots, m\}$ este mulțimea tuturor obiectelor iar $S = \{S_j, j = 1, \dots, n\}$ este mulțimea tuturor sferelor din reprezentare.

În cazul plan convergența este evidentă, deoarece s-a arătat deja existența unei reprezentări perfecte: $C_\infty = \lim_{m \rightarrow \infty} C(m)$.

Dacă se notează cu $E[C(m)]$ mulțimea eroare asociată cu cea de-a m -a reprezentare este evident că:

$$\lim_{m \rightarrow \infty} E[C(m)] = E[\lim_{m \rightarrow \infty} C(m)] \quad E[C_\infty] = \emptyset \quad (\text{III.2.124})$$

rezultând astfel un șir care este și monoton: $E[C(m)] > E[C(m+1)] \quad \forall m = 1, 2, \dots$

Concluzia imediată este că adăugarea de sfere suplimentare conduce la o umplere finală aproape perfectă.

Extensia la cazul tridimensional indică din nou convergență, dar reprezentarea capetelor necesită un tratament special similar acoperirii.

III.2.4. Program de sfericizare

Programul realizează modele spațiale ale roboților și obstacolelor. Acestea vor fi ulterior aplicate în probleme de mișcare, mai precis în detectarea coliziunilor și planificarea mișcărilor în 3D [205].

Planificarea mișcărilor fiind o aplicație fundamentală în robotică a principiilor inteligenței artificiale, are drept principal scop înzestrarea unui robot autonom cu capacitatea primare de generare a propriilor mișcări [13], [21], [54], [106], [144]. O primă condiție ar fi evitarea coliziunii cu posibile obstacole ce se pot afla pe traiectorie.

În cazul reprezentării robotului și obstacolelor prin mulțimi de sfere, detectarea coliziunilor se realizează simplu, prin intersecții ale spațiilor astfel

construite, ceea ce reduce numărul calculelor la minim posibil și se bazează pe o evaluare euristică a funcțiilor și a procesului de cercetare [142], [145], [163], [168], [176], [198], [199].

III.2.4.1. Prezentarea interfeței și utilizarea programului

Interfața programului se prezintă sub forma unei ferestre plutitoare de dimensiuni 640x480 pixeli ce conține o suprafață de desenare, butoane și zone de afișare a indicațiilor și a informațiilor cu privire la starea programului, dispuse în modul prezentat în figura III.60.

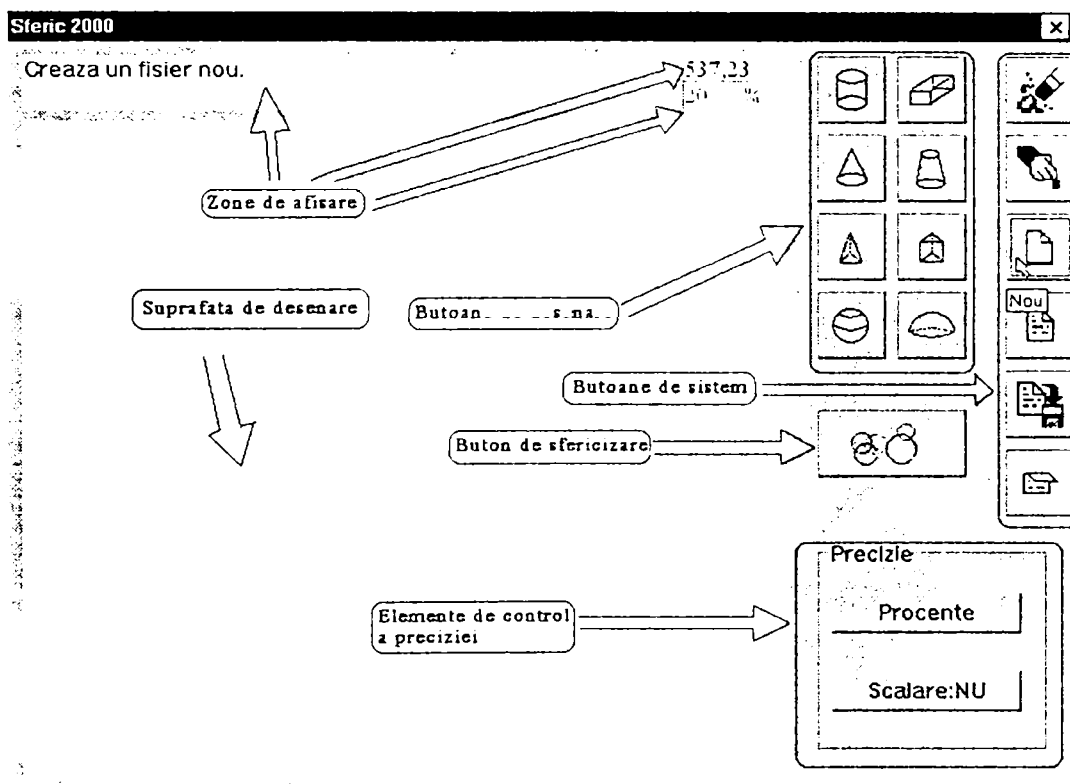


Fig.III.60. Aspectul interfeței grafice

Suprafața de desenare este zona în care se sintetizează obiectul, cu ajutorul butoanelor de desenare, din corpuri geometrice regulate reprezentate prin secțiunea transversală centrală. Desenul ce se obține constituie un plan de reprezentare a obiectului.

Programul conține algoritmi de sfericizare pentru opt corpuri geometrice prestabilite din care se pot compune obiecte de formă neregulată. Aceste corpuri sunt: cilindrul, paralelipipedul, conul, trunchiul de con, tetraedrul, prisma triunghiulară, sfera

și calota sferică. Fiecărui corp geometric îi corespunde câte un buton de desenare, cu ajutorul căruia se adaugă corpul la obiectul deja existent. Pentru aceasta, trebuie să se selecteze un punct pe suprafața de desenare, după ce în prealabil s-a creat un fișier special cu butonul *Nou*, ce va reprezenta centrul bazei inferioare a corpului, apoi se activează butonul de desenare corespunzător. Va apărea o succesiune de casete de dialog ce permit introducerea dimensiunilor corpului și unghiul de înclinare a obiectului. Numărul acestora depinde de corpul geometric în cauză.

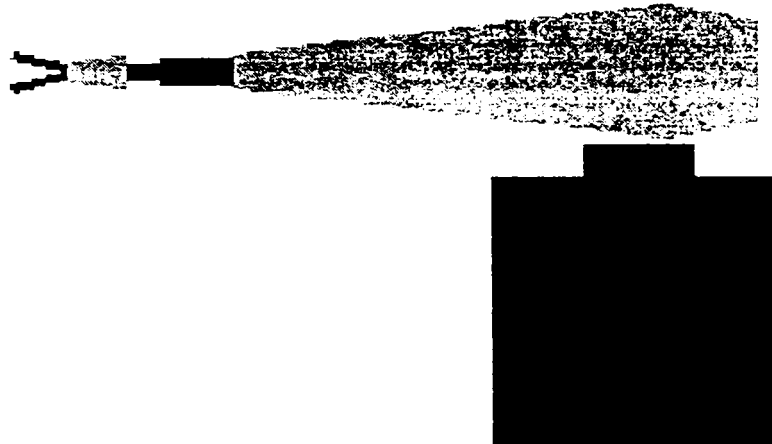


Fig.III.61. *O reprezentare 2D prin sfericizare a robotului din fig.III.2*

Odată încheiată această operațiune, dacă dimensiunile au fost corect corelate, va apărea pe ecran varianta 2D a obiectului, reprezentată prin secțiunea sa transversală. Convenția de culori utilizată pentru reprezentare este:

- verde închis = cilindru
- albastru închis = paralelipiped
- galben = con
- albastru deschis = trunchi de con
- maro = tetraedru
- mov = prisma triunghiulară
- verde deschis = sferă
- roșu = calota sferică.

Aceste culori se păstrează și la vizualizarea 3D a obiectelor sfericizate. Un exemplu este prezentat în figura III.61 ce reprezintă un robot compus dintr-un paralelipiped, cinci cilindri și trei trunchiuri de con.

După ce s-a creat obiectul dorit în 2D, se stabilesc parametrii sfericizării reprezentăți de procentul de acoperire (minim 20% și maxim 100%) și de o mărime booleană scalară care are două valori *DA* și *NU*. Cu cât procentul de acoperire este mai mare cu atât sfericizarea este mai bună. Setarea lui *DA* conduce la obținerea unei sfericizări bune din punct de vedere ai parametrilor de calitate menționați la capitolul III.2.3. Cu desenul creat și parametrii setați la valoarea dorită, se apasă butonul de sfericizare .

Pentru a crea imaginea 3D a obiectului s-a folosit setul de instrucțiuni *VRML* pentru citirea căruia se folosește programul *GLVIEW*. Aceste precizări îi sunt ascunse utilizatorului, programul scriind automat într-un fișier codul *VRML* pentru obiectul respectiv, apoi apelând interpretorul de *VRML*. Utilizatorul va observa pe ecran doar imaginea 3D a obiectului.

În limita resurselor disponibile ale sistemului, obiectul poate fi rotit și mărit apelând la funcțiile puse la dispoziție de *GLVIEW* prin meniurile și butoanele acestuia.

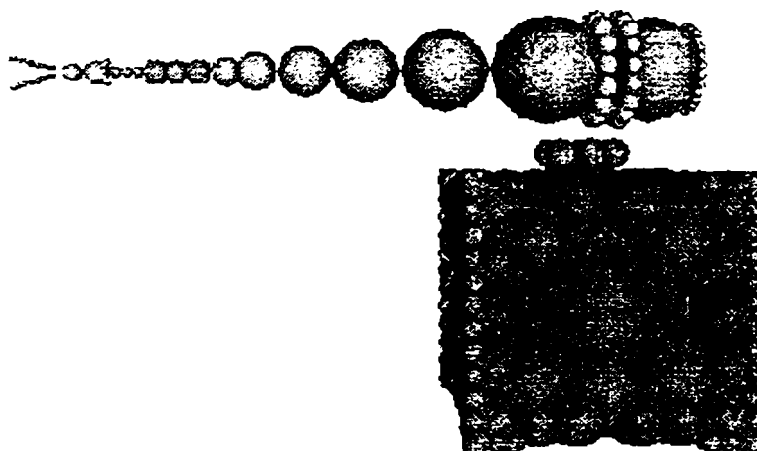


Fig.III.62. Robotul plan din fig.III.61 reprezentat 3D cu o precizie de 20%

Pentru a reveni la interfața programului de sfericizare se închide fereastra *GLVIEW* ca oricare altă fereastră Windows. În figurile III.62 și III.63 sunt prezentate sfericizări 3D vizualizate în *GLVIEW*, la o precizie de 20% și respectiv, 100%.

După finalizarea unei reprezentări, desenul se poate continua urmând a se apela ulterior o nouă sfericizare sau se poate salva și închide folosind butoanele sistem *Salvează* și *Închide* aflate în partea dreaptă a ecranului. O nouă reprezentare se poate realiza deschizând un fișier nou cu butonul *Nou* sau se poate redeschide un

fișier creat anterior și salvat, cu ajutorul butonului *Deschide*. În cazul în care se dorește eliminarea unor corpuri din desen se pot utiliza butoanele: *Șterge* pentru a șterge ultimul corp desenat și *Șterge tot* pentru a șterge toate corpurile desenate.

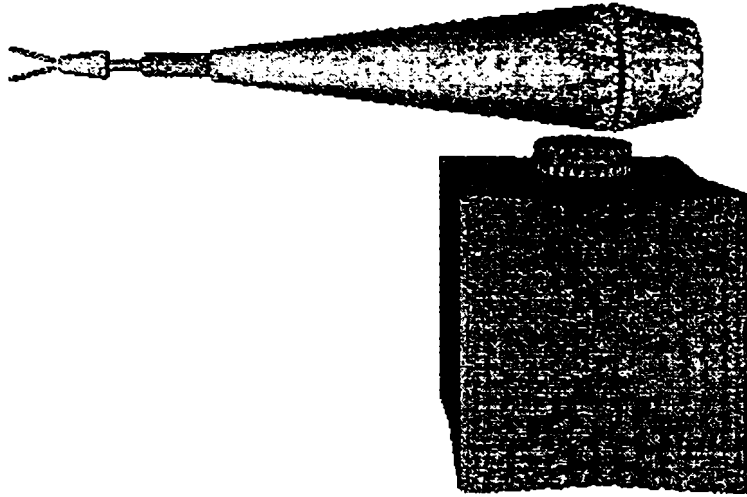


Fig.III.63. Robotul plan din fig.III.61 reprezentat 3D cu o precizie de 100%

Interfața mai conține încă trei zone de afișare pe lângă suprafața de desenare, butoanele de desenare, de ștergere și de sistem:

- Prima zonă joacă rolul de *Help* contextual completând informațiile oferite de program prin indicatori de butoane, titluri de ferestre, etc., explicând mai pe larg funcțiile diverselor butoane, operațiile ce trebuie efectuate sau erorile apărute;
- A doua zonă de afișare aflată în dreapta sus indică coordonatele la care se află mouse-ul pe suprafața de desenare. În cazul în care cursorul mouse-ului nu se află pe suprafața de desenare, această fereastră va conține coordonatele ultimului punct peste care a trecut mouse-ul când a ieșit din suprafața de desenare;
- Ultima zonă de desenare afișează valoarea curentă în procente a preciziei ce poate fi modificată de la butonul *Procente*.

III.2.4.2.Algoritmi de sfericizare

Pentru calcularea coordonatelor centrelor și razele sferelor s-au utilizat considerațiile teoretice prezentate în capitolul III.2.2. Calitatea sfericizării este funcție de precizia dorită și exprimată în procente, calitatea minimă fiind aleasă la precizia de

reprezentare de 20%, reprezentarea aproape perfectă corespunzând preciziei de 100%. Algoritmii permit și o precizie mai mare ca 100% dar este inefficient deoarece numărul de sfere crește foarte mult astfel încât nu mai pot fi reprezentate 3D, placa grafică nesuportând chiar zeci de mii de sfere.

De altfel, nici chiar algoritmii pur matematici nu permit o sfericizare perfectă, fapt care a fost scos în evidență prin calculul erorilor de reprezentare în capitolul III.2.2. În plus, la reprezentarea prin programul de sfericizare, perfecțiunea nu poate fi atinsă și datorită faptului că orice dimensiune este un număr întreg de pixeli. Astfel, între sfera de rază 3 și sfera de rază 2 nu pot exista sfere de rază intermediară și deci acestea se rotunjesc prin lipsă sau adaos.

Programul folosește ca unitate de măsură pixelul și astfel pot apărea erori de rotunjire vizibile dacă se urmărește reprezentarea unor corpuri de dimensiuni mici. Dacă dimensiunile depășesc 20 de pixeli erorile de aproximație devin tot mai puțin vizibile.

➤ Structurarea algoritmilor

Algoritmii de sfericizare sunt structurați pe procedurile:

- **procedure scriewrml**(yrot,zrot,xc,yc,zc,x,y,z,raza:integer; unghi:real; var fis:text; var nr:integer);

În timpul calculelor intermediare se folosesc coordonate relative la centrul bazei figurii. De aceea în procedura scriewrml se trece la coordonate absolute și se efectuează rotația, deoarece utilizatorul poate specifica înclinarea obiectelor față de axa Ox (teoretic algoritmii permit orice rotație în spațiu, dar reprezentarea inițială 2D a obiectelor impune limitarea doar la rotația în planul monitorului).

- **procedure IntCilindrubaza**(yrot,zrot,xc,yc,zc,h,r,e:integer; unghi:real; var fis:text; var nr:integer);

- **procedure IntCilindruLateral**(yrot,zrot,xc,yc,zc,h,r,e:integer; unghi:real; var fis:text; var nr:integer);

- **procedure IntConBaza**(xc,yc,zc,h,r1,dc0:integer; proc,alfa,unghi:real; var fis:text; var nr:integer);

- **procedure IntTrunchiConSus**(yrot,zrot,xc,yc,zc,h,r2,dc0:integer; proc, alfa, unghi: real; var fis:text; var nr:integer);

- **procedure IntTrunchiConMic**(xc,yc,zc,h,r1,r2:integer; proc,alfa,unghi:real; var fis: text; var nr:integer);
- **procedure IntTrunchiConLateral**(xc,yc,zc,h,r1,r2:integer; proc,unghi:real; var fis : text; var nr:integer);
- **procedure IntConLateral**(xc,yc,zc,h,r:integer; proc,unghi:real; var fis:text; var nr: integer);
- **procedure IntCalotasf**(yrot,zrot,xc,yc,zc,h,r,dc0:integer; proc,unghi:real; var fis : text; var nr:integer).

Următoarele trei proceduri nu sunt algoritmi de sfericizare, dar sunt folosite de aceștia aproape tot timpul, astfel încât sunt necesare pentru descifrarea codului.

```

procedure scriewrml(yrot,zrot,xc,yc,zc,x,y,z,raza:integer;
unghi:real; var fis:text; var nr:integer);
{x,y,z sunt coord relative la centru (xc,yc,zc) dar rotatia nu se
face totdeauna in jurul centrului bazei ci in jurul altui punct...de
ex pt baza de sus a cilindrului...}
var nrsfera:string[10];
    sferey:integer;
begin
    nr:=nr+1;
    str(nr,nrsfera);
    x:=x+xc;
    y:=y+yc;
    z:=z+zc;
    sferey:=y;
    y:=round(y*cos(unghi)- z*sin(unghi)+yrot-yrot*cos(unghi)
+zrot*sin(unghi));
    z:=round(sferey*sin(unghi)+ z*cos(unghi)+zrot-yrot*sin(unghi)-
zrot*cos(unghi));
    writeln(fis,'          DEF Sphere'+nrsfera+' Separator (');
    writeln(fis,'          Translation (');
    writeln(fis,'          translation ',y,' ',z,' ',x);
    writeln(fis,'          } #Translation');
    writeln(fis,'          Sphere (');
    writeln(fis,'          radius ',raza);
    writeln(fis,'          } #Sphere');
    writeln(fis,'          } #Sphere'+nrsfera+' Separator');
    {in wrml avem ox si oy n plan si oz perpendicular pe planul
monitorului, iar in calcule s-a considerat sistemul Oxyz clasic cu oz
sus..asa ca schimbam aici coord}

    {pt tiparirea in sectiunea transversala}
    circle(y+200,200-z,raza);
    {in sectiunea bazei}
    circle(y+400,200-x,raza);
end;

procedure aflapas(h:integer; var nr,pas:integer);
begin
    if (h<0) or (pas=0) then nr:=0
    else
        if h mod pas=0 then
            nr:=h div pas -1

```

```

else begin
  nr:=h div pas;
  pas:=h div (nr+1);
end;
end;

function arcsin(x:real):real;
begin
  if x=1
  then arcsin:=pi/2
  else if x=-1
  then arcsin:=-pi/2
  else
    arcsin:=arctan (x / sqrt (1-sqr(x)));
end;

```

❖ Rotatia figurilor și alegerea reperelor

Reprezentarea figurilor rotite se realizează prin operația de rotație în jurul unui centru în planul yOz :

$$\begin{aligned}
 xrot &= x; \\
 yrot &= y \cdot \cos(u) - z \cdot \sin(u) + yc - yc \cdot \cos(u) + zc \cdot \sin(u) \quad (\text{III.2.125}) \\
 zrot &= y \cdot \sin(u) + z \cdot \cos(u) + zc - yc \cdot \sin(u) - zc \cdot \cos(u)
 \end{aligned}$$

Pe parcursul implementării s-au folosit trei sisteme de coordonate, însă transformările de coordonate și relațiile de legătură dintre ele sunt realizate automat de program, astfel încât utilizatorul nu întâmpină nici o problemă. Au fost necesare mai multe sisteme pentru că s-au folosit mai multe limbaje de programare, iar programele sunt definite fiecare în propriul sistem.

Algoritmii de sfericizare au folosit sistemul cartezian de coordonate cu planul yOz planul monitorului. În acest sistem se scriu coordonatele în fișierul rezultat cu extensia *.sft*.

În cadrul interfeței Delphi la prezentarea 2D a obiectelor s-a folosit sistemul conventional al monitorului xOy (cu originea în colțul stânga sus).

Pentru reprezentarea spațială în *wrml* s-a folosit sistemul acestuia, adică sistemul cartezian cu planul xOy în planul monitorului și originea în colțul din stânga jos.

➤ Sfericizarea cilindrului

Așa cum s-a prezentat în cadrul capitolului III.2.2. parametrii ce definesc cilindrul sunt raza bazei r și înălțimea h .

❖ Acoperirea laterală

Pentru scrierea programului s-au luat în considerare:

- Pasul ca distanță dintre centrele sferelor: $pas = 2 \cdot r \cdot 20 / \text{precizie}$;
- Precizia minimă asigurată atunci când sferele sunt tangente exterior: $pas = 2 \cdot r$;
pasul scade și numărul de sfere crește proporțional cu precizia; ulterior pasul se adaptează astfel încât prima și ultima sferă să fie tangente bazelor cilindrului;
- Relațiile utilizate sunt:

$$\begin{aligned}
 rsfera &= rcilindru \\
 xsfera &= xcentru \\
 ysfera &= ycentru \\
 zsfera_0 &= zcentru + rcilindru \\
 zsfera_i &= zsfera_{i-1} + pas \\
 zsfera_n &= zcentru + h - rcilindru.
 \end{aligned}
 \tag{III.2.126}$$

Algoritmul implementat în Pascal este:

```

procedure IntCilindruLateral(yrot, zrot, xc, yc, zc, h, r, e: integer;
unghi: real; var fis: text; var nr: integer);

var i, nrlocal, pas: integer;
begin
  if h < 2 * r then
    IntCilindruBaza(yrot, zrot, xc, yc, zc, h, r, e div 2, unghi, fis, nr)
  else
    begin
      pas := e;
      aflapas(h - 2 * r, nrlocal, pas);
      z := r;
      scriewrml(yrot, zrot, xc, yc, zc, 0, 0, z, r, unghi, fis, nr);
      for i := nr + 2 to nr + nrlocal + 1 do
        begin
          z := z + pas;
          scriewrml(yrot, zrot, xc, yc, zc, 0, 0, z, r, unghi, fis, nr);
        end;
      z := h - r;
      scriewrml(yrot, zrot, xc, yc, zc, 0, 0, z, r, unghi, fis, nr);

      if e > 1 then
        begin
          IntCilindruBaza(yrot, zrot, xc, yc, z + zc, r, r, e div
2, unghi, fis, nr);
          IntCilindruBaza(yrot, zrot, xc, yc, zc, r, r, e div
2, unghi, fis, nr);
        end;
    end;
end;
end;

```

❖ Acoperirea bazei

Baza se consideră de la punctul de tangență al ultimei sfere de acoperire cu cilindrul, deci are parametrii: r și $h = r$, sau poate fi un cilindru cu $h < r$.

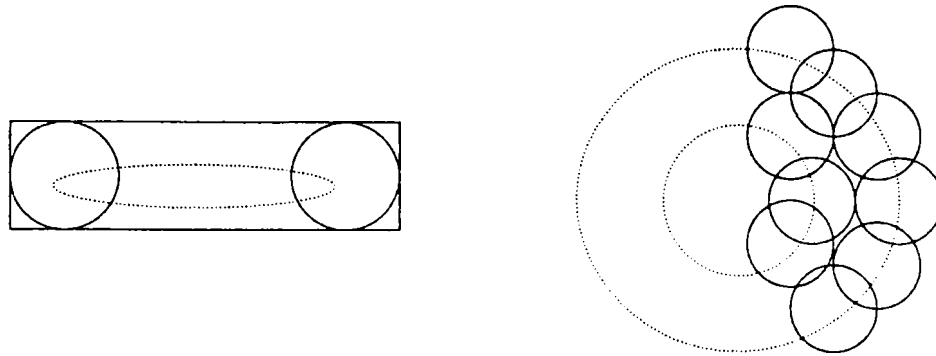


Fig.III.64. Primul pas al sfericizării bazelor cilindrilor

Procesul decurge astfel:

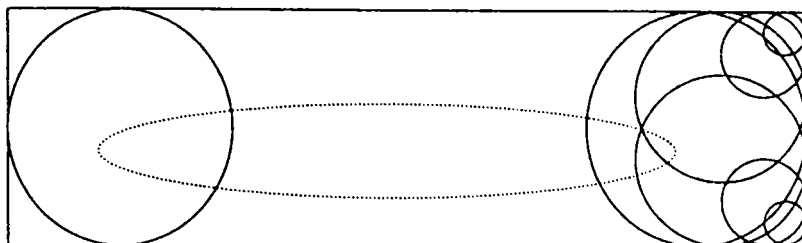
- Inițial se sfericizează raza bazei prin considerarea unor sfere de rază $h/2$ cu centre ce se plasează pe cercuri concentrice în funcție de precizie (figura III.64):

```

raza=h/2
pasr=raza*20/prec      (precizia bazei e dublă față de a suprafeței
laterale)
distcentru0=r-raza
distcentrui=distcentrui-1-pasr
arc=pasr              (arcul de deplasare pe cerc)
alfa=arc/distcentru  (unghiul la centru în radiani)
n=2*pi/alfa +1       (numarul de sfere cu centrele pe același cerc)
centrele sferelor : z=yc+h/2
                   x=xc+distcentru*cos(alfa*j)
                   y=zc+distcentru*sin(alfa*j)

```

- În pasul următor se sfericizează lateral baza împărțind înălțimea în funcție de pas, și de fiecare dată se deplasează sferele pe cercuri concentrice pentru a ocupa suprafața laterală (figura III.65).



```
rsfera=pash
zjos=zc+pash
zsus=zc+h-pash
```

Fig.III.65.Pasul al doilea în sfericizarea bazelor cilindrilor

```
Procedura IntCilindrubaza(yrot,zrot,xc,yc,zc,h,r,e:integer;
unghi:real; var fis:text; var nr:integer);
```

```
var i,j,K,n,dc,hlocal,pash,psr,nrstrat,nrcerc,raza:integer;
    alfa,aux:real;
begin
    raza:=h div 2;
    if e>2*raza then e:=e*2*raza div r;
    psr:=e; {si psr in functie de h}
    hlocal:=0+raza;
    { se traseaza pe linia mediana cercurile concentrice de sfere....
    pt a se acoperi bazele...apoi se acopera doar suprafata laterala}
    aflapas(r-raza,nrcerc,psr);
    nrcerc:=nrcerc+1;

    z:=hlocal;
    scriewrml(yrot,zrot,xc,yc,zc,0,0,z,raza,unghi,fis,nr);

    dc:=r-raza; {dist de la centru la centrul cercului}
    for k:=1 to nrcerc do
        if dc>0 then
            begin
                if e>2*raza then aux:=2*raza
                else aux:=e;
                alfa:=aux/dc;
                n:=round(2*pi/alfa)+1;
                alfa:=2*pi/n;
                for j:=1 to n do
                    begin
                        x:=round(dc*cos(alfa*j));
                        y:=round(dc*sin(alfa*j));
                        z:=hlocal;
                        scriewrml(yrot,zrot,xc,yc,zc,x,y,z,raza,unghi,fis,nr);
                    end;
                dc:=dc-psr; {pt cercurile concentrice de pe baza}
            end;
    pash:=e;
    aflapas(h div 2,nrstrat,pash);
    for i:=1 to nrstrat do
        begin
            hlocal:=i*pash;
            raza:=i*pash;
            dc:=r-raza;
            if e>2*raza then aux:=2*raza
            else aux:=e;
            alfa:=aux/dc;
            n:=round(2*pi/alfa)+1;
```

```

alfa:=2*pi/n;
for j:=1 to n do
begin
  x:=round(dc *cos(alfa*j));
  y:=round(dc *sin(alfa*j));
  z:=hlocal;
  scriewrml(yrot,zrot,xc,yc,zc,x,y,z,raza,unghi,fis,nr);
  scriewrml(yrot,zrot,xc,yc,zc,x,y,h-
i*pash,raza,unghi,fis,nr);
end;
end;
end;

```

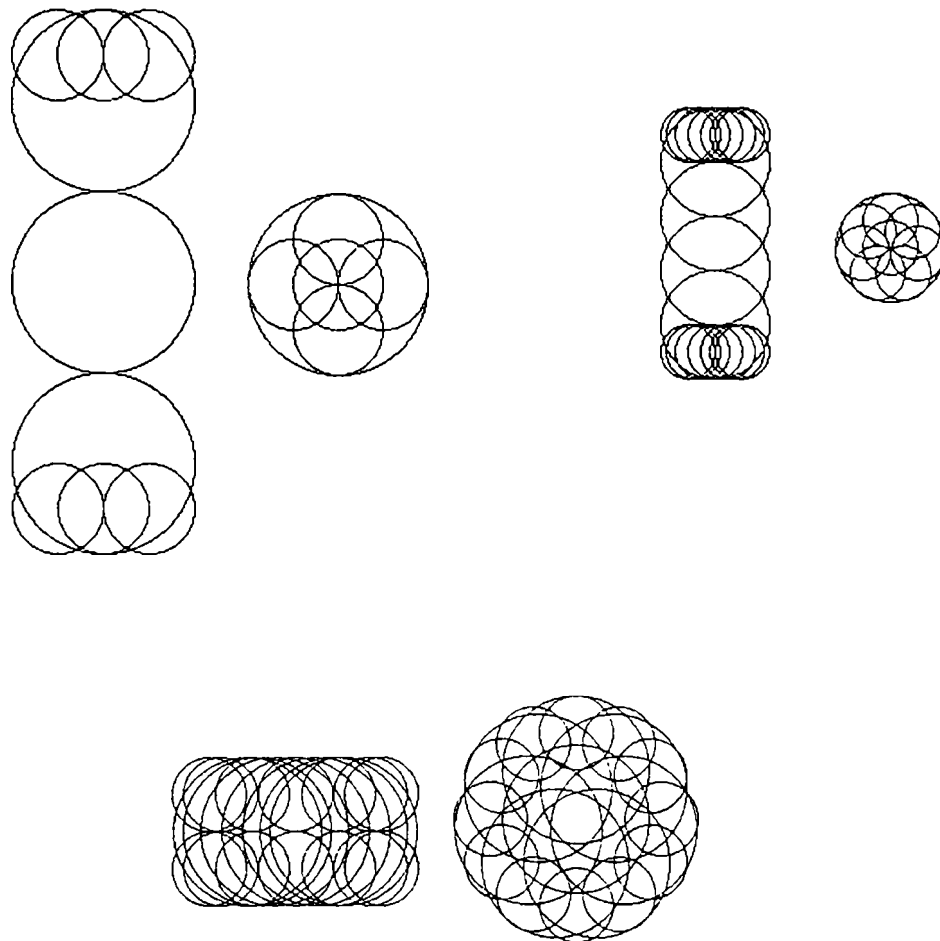


Fig.III.66. *Reprezentări de cilindri prin secționare*

Pe baza programului astfel realizat se reprezintă corpuri cilindrice la precizii diferite (figura III.66). Pentru o mai clară vizualizare a modului de sfericizare a suprafeței laterale și a bazei s-au reprezentat alături secțiunile prin cilindri realizate cu plane verticale și orizontale

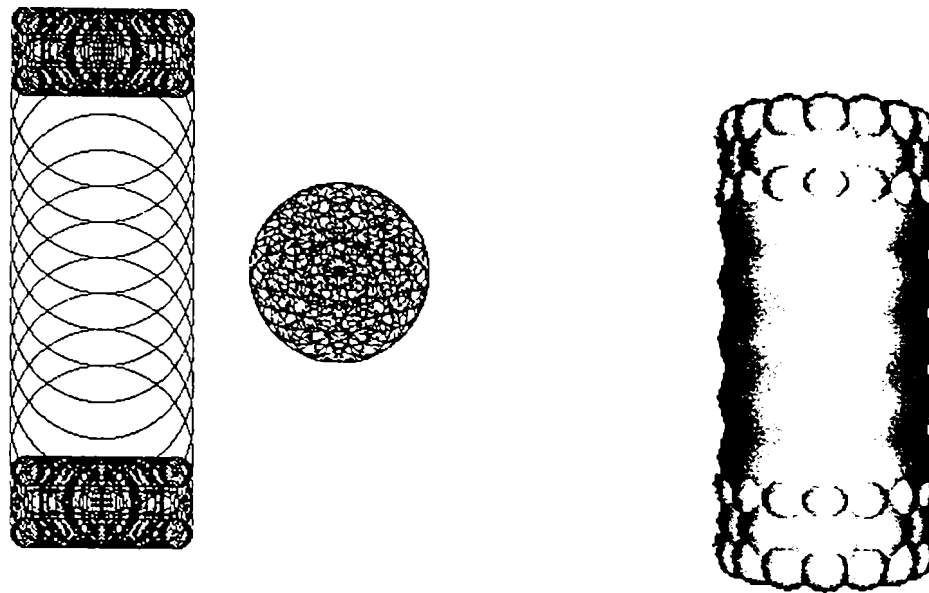


Fig.III.67. Diferite etape în reprezentarea 3D a cilindrilor

Pentru o vizualizare 3D este prezentat un model de cilindru obținut cu un player de *wrml*.

➤ Sfericizarea conului și a trunchiului de con

Parametrii geometrice ai conului sunt: raza bazei r și înălțimea h , iar parametrii trunchiului de con sunt: raza bazei mari $r1$, raza bazei mici $r2$ și înălțimea h .

❖ Reprezentarea exterioară laterală

În conformitate cu relațiile stabilite în capitolul III.2.2, programul de reprezentare exterioară laterală conține:

□ Pentru con:

```

procedure IntConLateral(xc,yc,zc,h,r:integer; proc,unghi:real; var
fis:text; var nr:integer);
var zn,znn,rn,htr,r2:integer;
    alfa,rnn:real;
begin
    yrot:=yc;
    zrot:=zc;
    alfa:=arctan (h/r);

```

```

zn:=trunc( r* tan(alfa/2));
rn:=zn;
{coord pt trunchiul de con prin care se va sfericiza baza mai
exact}
htr:=2*rn;

while (h-zn-rn>0) and (rn>0) do
begin
  scriewrml(yrot, zrot, xc, yc, zc, 0, 0, zn, rn, unghi, fis, nr);
  znn:=zn;
  rnn:= rn*((h-zn-rn) / (h-zn+rn));
  {daca sfera urm ar fi exact tang la actuala}
  zn:=round((rn+rnn)*20/proc);
  zn:=znn+zn;
  if zn=znn then zn:=zn+1;
  rn:=trunc( rn* ((h-zn)/(h-znn)) +0.25);
end;
IntConBaza(xc, yc, zc, htr, r, 0, proc, alfa, unghi, fis, nr);
end;

```

□ Pentru trunchiul de con:

```

procedure IntTrunchiConLateral(xc, yc, zc, h, r1, r2: integer;
proc, unghi: real; var fis: text; var nr: integer);
var zn, znn, rn, htr, hcon: integer;
    alfa, rnn: real;
begin
  {fiind un trunchi in general poate avea h>>r1}
  yrot:=yc;
  zrot:=zc;
  alfa:=arctan (h/(r1-r2));
  zn:=trunc( r1* tan(alfa/2));
  rn:=zn;
  htr:=2*rn;
  hcon:=trunc( r1*tan(alfa));
  if h<2*rn then
    IntTrunchiConMic(xc, yc, zc, h, r1, r2, proc, alfa, unghi, fis, nr)
  else
    begin
      IntConBaza(xc, yc, zc, htr, r1, 0, proc, alfa, unghi, fis, nr);
      { rn raza din varf}
      while (rn>0) and (h-zn-rn>0) do
        begin
          scriewrml(yrot, zrot, xc, yc, zc, 0, 0, zn, rn, unghi, fis, nr);
          znn:=zn;
          rnn:= rn*((hcon-zn-rn)/ (hcon-zn+rn));
          {daca sfera urm ar fi exact tang la actuala}
          zn:=zn+ round((rn+rnn)*20/proc);
          rn:=round( rn* ((hcon-zn)/(hcon-znn)));
        end;
          scriewrml(yrot, zrot, xc, yc, zc, 0, 0, h-rn, rn, unghi, fis, nr);

          IntTrunchiConsus(yrot, zrot, xc, yc, zc+h-
2*rn, 2*rn, r2, 0, proc, alfa, unghi, fis, nr);
        end;
    end;
end;

```

❖ Reprezentarea exterioară a bazelor□ Pentru con

```

procedure IntConBaza(xc,yc,zc,h,r1,dc0:integer; proc,alfa,unghi:real;
var fis:text; var nr:integer);
var dc,dh,n,j:integer;
    beta,aux,hnext,dr:real;
begin
    yrot:=yc;
    zrot:=zc;
    dh:=h div 2;
    dc:=dc0;
    hnext:= dh * (1-sin(alfa/2))/(1+sin(alfa/2));
    {daca urm sfera ar fi tangenta la anterioara}
    dr:=(dh+hnext)*20/proc;
    dh:=dh-round(dr*sin(alfa/2));
    dc:=dc+trunc(dr*cos(alfa/2));
    aux:=2*dh*20/proc;

    while (r1-dc-dh>0) and (dh>0) do
    begin
        if aux>2*dh then aux:=2*dh;
        beta:=aux/dc;
        n:=trunc(2*pi/beta)+1;
        beta:=2*pi/n;
        for j:=1 to n do
        begin
            x:=round(dc *cos(beta*j));
            y:=round(dc *sin(beta*j));
            z:=dh;
            scriewrml(yrot,zrot,xc,yc,zc,x,y,z,dh,unghi,fis,nr);
        end;
        hnext:= dh *(1-sin(alfa/2))/(1+sin(alfa/2));
        {daca urm sfera ar fi tangenta la anterioara}
        dr:=(dh+hnext)*20/proc;
        if (dh<h div 8) and (alfa<0.78) or (dh<h div 12)
            then if proc<40 then dr:=dh+hnext
                else dr:=dr*2;
        if trunc(dr*cos(alfa/2))=0 then dr:=dr+2;
        dc:=dc+round(dr*cos(alfa/2));
        dh:=round(dh- dr*sin(alfa/2));
    end;
end;

```

□ Pentru trunchiul de con

```

procedure IntTrunchiConSus(yrot,zrot,xc,yc,zc,h,r2,dc0:integer;
proc,alfa,unghi:real; var fis:text; var nr:integer);
var dc,dh,n,j:integer;
    beta,aux,hnext,dr:real;
begin
    dh:=h div 2;
    dc:=dc0;
    alfa:=pi-alfa;
    hnext:= dh * (1-sin(alfa/2))/(1+sin(alfa/2));
    {daca urm sfera ar fi tangenta la anterioara}
    dr:=(dh+hnext)*20/proc;
    dh:=round( dh-(dr*sin(alfa/2)));

```

```

dc:=dc+round(dr*cos(alfa/2));
aux:=2*dh*20/proc;

while (r2-dc-round(dh*cos(alfa/2))>0) and (dh>0) do
begin
  if aux>2*dh then aux:=2*dh;
  beta:=aux/dc;
  n:=trunc(2*pi/beta)+1;
  beta:=2*pi/n;
  for j:=1 to n do
  begin
    x:=round(dc*cos(beta*j));
    y:=round(dc*sin(beta*j));
    z:=h-dh;
    scriewrml(yrot,zrot,xc,yc,zc,x,y,z,dh,unghi,fis,nr);
  end;
  hnext:=dh*(1-sin(alfa/2))/(1+sin(alfa/2));
  {daca urm sfera ar fi tangenta la anterioara}
  dr:=(dh+hnext)*20/proc;
  if (dh<h div 8) and (alfa<0.78) or (dh<h div 12)
    then if proc<40 then dr:=dh+hnext
          else dr:=dr*2;
  if trunc(dr*cos(alfa/2))=0 then dr:=dr+2;

  dh:=round(dh-dr*sin(alfa/2));
  dc0:=dc;
  dc:=dc+round(dr*cos(alfa/2));

  if dc=dc0 then break;
end;
end;

```

```

procedure IntTrunchiConMic(xc,yc,zc,h,r1,r2:integer;
proc,alfa,unghi:real; var fis:text; var nr:integer);
var dc,dc0,raza,nrcerc,pasr,k,j,n:integer;
    beta,aux:real;
begin
  yrot:=yc;
  zrot:=zc;
  raza:=h div 2;
  dc:=trunc ( (r1+r2 - sqrt(sqr(r1-r2)+sqr(h)) ) /2) ;
  dc0:=dc;
  {dist de la centru la centrul cercului}

  pasr:=round(raza*2*20/proc);
  aflapas(dc,nrcerc,pasr);
  nrcerc:=nrcerc+1;

  scriewrml(yrot,zrot,xc,yc,zc,0,0,raza,raza,unghi,fis,nr);

  for k:=1 to nrcerc do
  if dc>0 then
  begin
    aux:=pasr;
    beta:=aux/dc;
    n:=round(2*pi/beta)+1;
    beta:=2*pi/n;
    for j:=1 to n do
    begin
      x:=round(dc*cos(beta*j));

```



```

y:=round(dc *sin(beta*j));
scriewrml(yrot,zrot,xc,yc,zc,x,y,raza,raza,unghi,fis,nr);
end;
dc:=dc-pasr;    {pt cercurile concentrice de pe baza}
end;
IntConBaza(xc,yc,zc,h,r1,dc0,proc,alfa,unghi,fis,nr);

IntTrunchiConsus(yrot,zrot,xc,yc,zc,h,r2,dc0,proc,alfa,unghi,fis,nr);

end;

```

În figura III.68 sunt prezentate diferite reprezentări de conuri în 2D și 3D cu diferite precizii de reprezentare.

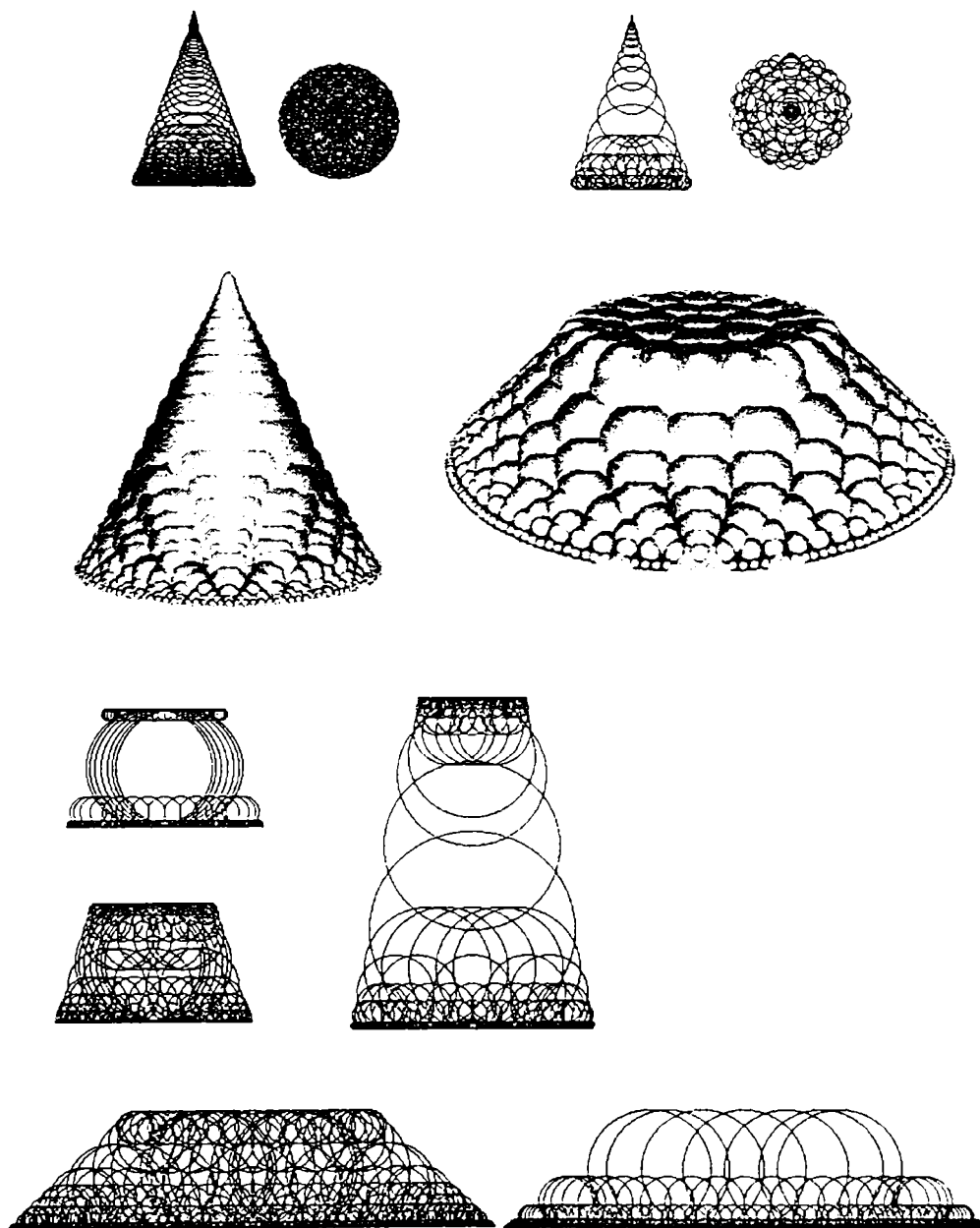


Fig.III.68.Diferite reprezentări de conuri și trunchiuri de con

➤ Sfericizarea calotelor sferice

Parametrii geometrice ai reprezentării sunt: raza calotei r și înălțimea calotei h .
 Relațiile utilizare pentru algoritmul de reprezentare sunt cele determinate la capitolul III.2.2.

```

procedure IntCalotasf(yrot,zrot,xc,yc,zc,h,r,dc0:integer;
proc,unghi:real; var fis:text; var nr:integer);
var n,j,rsf,dc,yp,zp:integer;
    beta,aux,t,arc,arcmax,tmax:real;
begin
  rsfera:=h div 2;
  rsf:=round(h/2+(r/(2*h))*r);
    {raza sferei din care provine calota}
  arc:=round(3/2*rsfera*20/proc);
  t:=arc/rsf;
  yp:=round(rsf*sin(t));
  zp:=round(rsf*cos(t))-(rsf-h);
  rsfera:=round(zp*tan(t/2)/sin(t));
  z:=rsfera;
  y:=dc0+round((rsf-rsfera)*sin(t));

  tmax:=arcsin(r/rsf);
  arcmax:=tmax*rsf;

  while (rsfera>1) and (arc-rsfera<arcmax) do
  begin
    aux:=2*rsfera*20/proc;
    beta:=aux/y;
    n:=trunc(2*pi/beta)+1;
    beta:=2*pi/n;
    dc:=y;
    for j:=1 to n do
    begin
      x:=round(dc*cos(beta*j));
      y:=round(dc*sin(beta*j));
      scriewrml(yrot,zrot,xc,yc,zc,x,y,z,rsfera,unghi,fis,nr);
    end;
    arc:=arc+round(3/2*rsfera*20/proc);
    t:=arc/rsf;
    yp:=round(rsf*sin(t));
    zp:=round(rsf*cos(t))-(rsf-h);
    rsfera:=round(zp*tan(t/2)/sin(t));
    z:=rsfera;
    y:=dc0+round((rsf-rsfera)*sin(t));
    if y=dc then break;
  end;
end;

```

În figura III.69 sunt prezentate realizări ale programului de sfericizare a calotelor sferice.

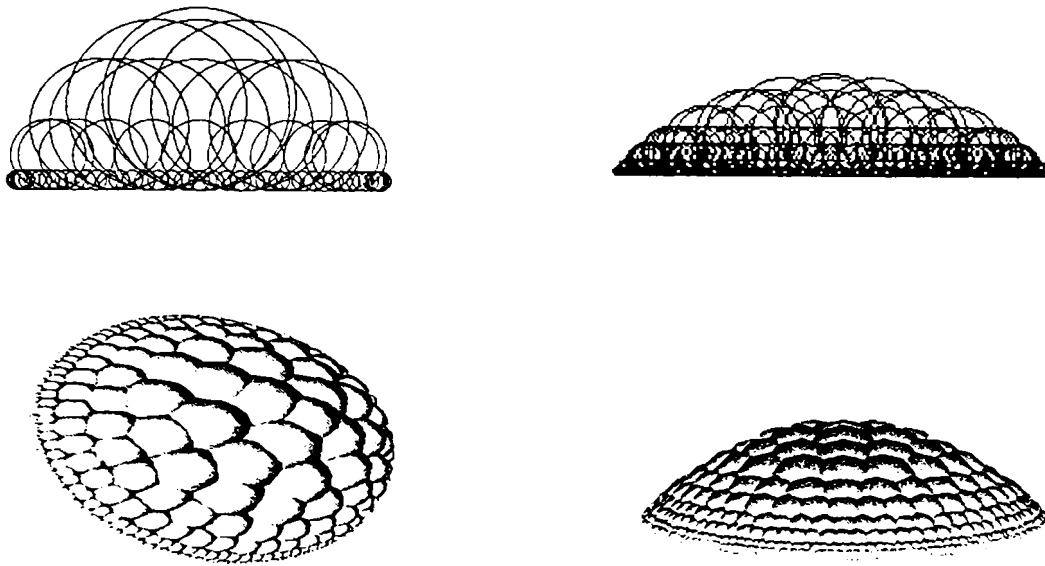


Fig.III.69. Sfericizari de calote sferice

➤ Sfericizarea prismelor drepte

Procesul de sfericizare constă în acoperirea prismelor cu cilindrii care ulterior se sfericizează conform metodologiei precizate anterior. Astfel, se acoperă cu cercuri baza prisme (triunghi, dreptunghi, pătrat, etc.) și pe fiecare cerc se ridică un cilindru ce se sfericizează. În aplicația prezentată, au fost sfericizate prisma triunghiulară regulată dreaptă și paralelipipedul drept, deci s-au folosit ca baze triunghiul echilateral și dreptunghiul.

```

procedure Tform1.intprismatriunghiulara(x,y,lat,h,un:longint);
var baza:textfile;
    pas,r,aux:longint;
    unghi:real;
    xc0,yc0,zc0:longint;
begin
    xc0:=0;
    yc0:=x-xcentru;
    zc0:=ycentru-y;
    unghi:=un*pi/180;
    bazaprismatriunghiulara(proc,lat);
    assignfile(baza,'triunghi.txt');
    reset(baza);
    while not eof(baza) do
    begin
        readln(baza,yc,xc,r);
        xc:=-xc;
    end
end

```

```

        xc:=xc0+xc;
        yc:=yc0+yc;
        zc:=zc0;
        yrot:=yc0;
        zrot:=zc0;
        pas:=round(2*r*20/proc);
        if pas>0 then
            IntCilindruLateral(yrot,zrot,xc,yc,zc,h,r,pas,unghi,fw,nr);
        end;
        closefile(baza);
    end;
end;

procedure Tform1.bazaprismatrionghiulara(proc:integer;n:longint);
var centx,centy,sx,sy,djx,djy,sjx,sjy,step:longint;
    fl:textfile;

procedure desen1;
begin
    writeln(fl,0,' ',0,' ',round(n*sqrt(3)/6));
end;

procedure desen2;
var i,raz,k:longint;
    aux,a:real;
begin
    i:=1;
    k:=0;
    aux:=0.99;
    repeat
        aux:=aux*aux;
        a:=1-aux;
        raz:=i div 2;
        if (raz<>0)and(raz<>k) then writeln(fl,0,' ',centy-sy-i,'
',raz);
        k:=raz;
        i:=i+round(a*step);
    until i>=round(n*sqrt(3)/3);
end;

procedure desen3;
var i,raz,x,y:longint;
    aux,a:real;
    k:longint;
begin
    i:=1;
    aux:=0.99;
    k:=0;
    repeat
        aux:=aux*aux;
        a:=1-aux;
        raz:=i div 2;
        x:=round(sjx+i*cos(pi/6));
        y:=round(sjy-i*sin(pi/6));
        if (raz<>0)and(raz<>k) then writeln(fl,x-centx,' ',centy-y,'
',raz);
        k:=raz;
        i:=i+round(a*step);
    until i>=round(n*sqrt(3)/3);
end;

```

```

procedure desen4;
var i,raz,x,y,k:longint;
    aux,a:real;
begin
    aux:=0.99;
    i:=1;
    repeat
        aux:=aux*aux;
        a:=1-aux;
        raz:=i div 2;
        x:=round(djx-i*cos(pi/6));
        y:=round(djy-i*sin(pi/6));
        if (raz<>0)and(raz<>k) then writeln(f1,x-centx,' ',centy-
y,' ',raz);
        k:=raz;
        i:=i+round(a*step);
    until i>=round(n*sqrt(3)/3);
end;

begin
    step:=10*round(500/proc);
    assignfile(f1,'triunghi.txt');
    rewrite(f1);
    centx:=640 div 2;
    centy:=(480 div 3)*2;
    sx:=centx;
    sy:=centy-round(n*sqrt(3)/3);
    djx:=centx+(n div 2);
    djy:=centy+round(n*sqrt(3)/6);
    sjx:=centx-(n div 2);
    sjy:=centy+round(n*sqrt(3)/6);
    desen1;
    desen2;
    desen3;
    desen4;
    closefile(f1);
end;

```

Cele două proceduri de mai sus reprezintă algoritmul de sfericizare a prismei triunghiulare și, respectiv, de acoperire cu cercuri a triunghiului echilateral. Pentru paralelipiped se procedează în mod similar, deci procedurile vor fi aproape identice.

➤ Sfericizarea tetraedrului regulat

Parametrul geometric al reprezentării este latura tetraedrului l .

Algoritmul de sfericizare constă din următorii pași:

- se plasează o sferă în centrul de greutate al tetraedrului, astfel încât ea să fie tangentă bazei acestuia;
- pornind din fiecare vârf al tetraedrului, se plasează sfere astfel încât acestea să fie tangente celor trei laturi ce se întâlnesc în respectivul vârf;

- măbind distanța dintre vârf și centrul sferei cu un pas calculat în funcție de precizia cerută de utilizator, sfera se deplasează spre interiorul tetraedrului;
- algoritmul a ajuns la sfârșit în momentul în care s-a realizat acoperirea necesară a suprafeței tetraedrului.

Reprezentări ale tetraedrelor sunt prezentate în figura III.70 iar procedura ce realizează operațiile de reprezentare este:

```
procedure Tform1.inttetraedru(x0,y0,n,un:longint;var fis:textfile);
var centx,centy,sx,sy,djx,djy,sjx,sjz,sjy,step:longint;
    alfa,beta,unghi:real;
    index:longint;
    nrsfera:string;
```

```
procedure scrievrml2(nrsfera:string;raza,x,y,z:integer);
begin
writeln(fis,'      DEF Sphere'+nrsfera+' Separator (');
writeln(fis,'          Translation (');
writeln(fis,'              translation ',y,' ',z,' ',x);
writeln(fis,'          )#Translation');
writeln(fis,'      Sphere (');
writeln(fis,'          radius ',raza);
writeln(fis,'          )#Sphere');
writeln(fis,'      )#Sphere'+nrsfera+' Separator');
    append(ft);
    writeln(ft,x div coefcon,' ',y div coefcon,' ',z div coefcon,'
',raza div coefcon);
    closefile(ft);
end;
procedure desen1;
var sferey:longint;
begin
    x:=xc;
    y:=yc;
    z:=zc+round(n/sqrt(6)/2);
    sferey:=y;
    y:=round(y*cos(unghi)- z*sin(unghi)+yrot-
yrot*cos(unghi)+zrot*sin(unghi));
    z:=round(sferey*sin(unghi)+ z*cos(unghi)+zrot-yrot*sin(unghi)-
zrot*cos(unghi));
    nr:=nr+1;
    str(nr,nrsfera);
    scrievrml2(nrsfera,round(n/sqrt(6)/2),x,y,z);
end;
```

```
procedure desen2;
var i,raz,x,y,z,k:longint;
    aux,a:real;
    sferey:longint;
begin
    i:=1;
    k:=0;
    aux:=0.99;
    repeat
        aux:=aux*aux;
        a:=1-aux;
        raz:=round(i*sin(beta));
```

```

        y:=round(-djx+i*cos(beta)*cos(pi/6));
        x:=round(djy-i*cos(beta)*sin(pi/6));
        z:=round(i*sin(beta));
    x:=x+xc;
    y:=y+yc;
    z:=z+zc;
    sferey:=y;
    y:=round(y*cos(unghi)- z*sin(unghi)+yrot-
yrot*cos(unghi)+zrot*sin(unghi));
    z:=round(sferey*sin(unghi)+ z*cos(unghi)+zrot-yrot*sin(unghi)-
zrot*cos(unghi));
        if (raz<>0)and(raz<>k) then
        begin
            inc(index);
            str(index,nrsfera);
            scrievrml2(nrsfera,raz,x,y,z);
            end;
            k:=raz;
            i:=i+round(a*step);
        until i>round(n/sqrt(6)/2/sin(beta));
end;

procedure desen3;
var i,raz,x,y,z:longint;
    aux,a:real;
    k:longint;
    sferey:longint;
begin
    i:=1;
    aux:=0.99;
    k:=0;
    repeat
        aux:=aux*aux;
        a:=1-aux;
        raz:=round(i*sin(beta));
        y:=round(djx-i*cos(beta)*cos(pi/6));
        x:=round(djy-i*cos(beta)*sin(pi/6));
        z:=round(i*sin(beta));
    x:=x+xc;
    y:=y+yc;
    z:=z+zc;
    sferey:=y;
    y:=round(y*cos(unghi)- z*sin(unghi)+yrot-
yrot*cos(unghi)+zrot*sin(unghi));
    z:=round(sferey*sin(unghi)+ z*cos(unghi)+zrot-yrot*sin(unghi)-
zrot*cos(unghi));
        if (raz<>0)and(raz<>k) then
        begin
            inc(index);
            str(index,nrsfera);
            scrievrml2(nrsfera,raz,x,y,z);
            end;
            k:=raz;
            i:=i+round(a*step);
        until i>round(n/sqrt(6)/2/sin(beta));
end;

procedure desen4;
var i,raz,x,y,k,z:longint;
    sferey:longint;
    aux,a:real;

```

```

begin
  aux:=0.99;
  i:=1;
  repeat
    aux:=aux*aux;
    a:=1-aux;
    raz:=round(i*sin(beta));
    x:=round(sy+i*cos(pi/6));
    z:=round(i*sin(beta));
    x:=x+xc;
    y:=yc;
    z:=z+zc;
    sferey:=y;
    y:=round(y*cos(unghi)- z*sin(unghi)+yrot-
yrot*cos(unghi)+zrot*sin(unghi));
    z:=round(sferey*sin(unghi)+ z*cos(unghi)+zrot-yrot*sin(unghi)-
zrot*cos(unghi));
    if (raz<>0)and(raz<>k) then
      begin
        inc(index);
        str(index,nrsfera);
        scrievrml2(nrsfera,raz,x,y,z);
      end;
    k:=raz;
    i:=i+round(a*step);
  until i>round(n/sqrt(6)/2/sin(beta));
end;

procedure desen5;
var i,raz,x,y,k,z,h:longint;
    sferey:longint;
    aux,a:real;
begin
  aux:=0.99;
  i:=1;
  h:=round(n*sqrt(6)/3);
  k:=0;
  repeat
    aux:=aux*aux;
    a:=1-aux;
    raz:=round(i*sin(beta));
    x:=xc;
    y:=yc;
    z:=h-i+zc;
    sferey:=y;
    y:=round(y*cos(unghi)- z*sin(unghi)+yrot-
yrot*cos(unghi)+zrot*sin(unghi));
    z:=round(sferey*sin(unghi)+ z*cos(unghi)+zrot-yrot*sin(unghi)-
zrot*cos(unghi));
    if (raz<>0)and(raz<>k) then
      begin
        inc(index);
        str(index,nrsfera);
        scrievrml2(nrsfera,raz,x,y,z);
      end;
    k:=raz;
    i:=i+round(a*step);
  until i>round(n/sqrt(6)/2/sin(beta));
end;

```



```

begin
  xcentru:=form1.imagel.width div 2;
  ycentru:=form1.imagel.height div 2;
  xc:=0;
  yc:=x0-xcentru;
  zc:=ycentru-y0;
  yrot:=yc;
  zrot:=zc;
  unghi:=un*pi/180;
  append(fis);
  step:=trunc(99/proc)+1;
  alfa:=arctan(sqrt(2))/2;
  beta:=arctan(1/sqrt(2)/2);
  centx:=640 div 2;
  centy:=(480 div 3)*2;
  sx:=centx;
  sy:=-round(n*sqrt(3)/3);
  djx:=n div 2;
  djy:=round(n*sqrt(3)/6);
  sjx:=centx-(n div 2);
  sjy:=centy+round(n*sqrt(3)/6);
  sjz:=centy+round(n*sqrt(3)/6);
  desen1;
  index:=nr;
  desen2;
  desen3;
  desen4;
  desen5;
  nr:=index;
  closefile(fis);
end;

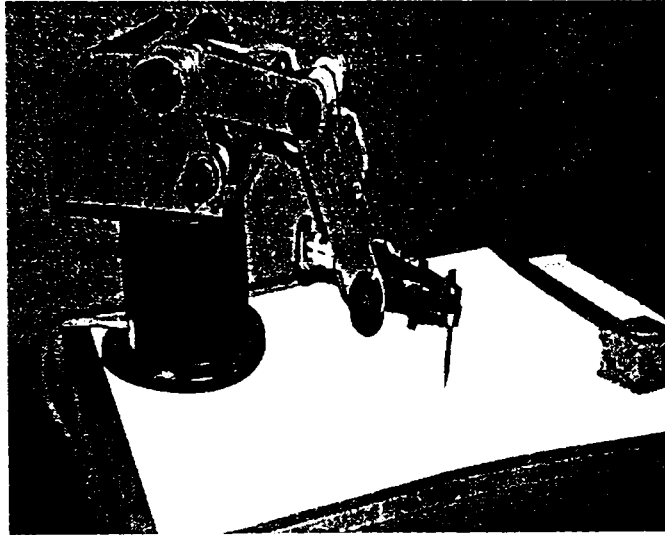
```

Tot din cadrul acestei proceduri se realizează transformarea fișierului .sfr în fișier .vrl, în ceea ce privește tetraedrele. În acest scop este scrisă procedura *scrievrlm2*.

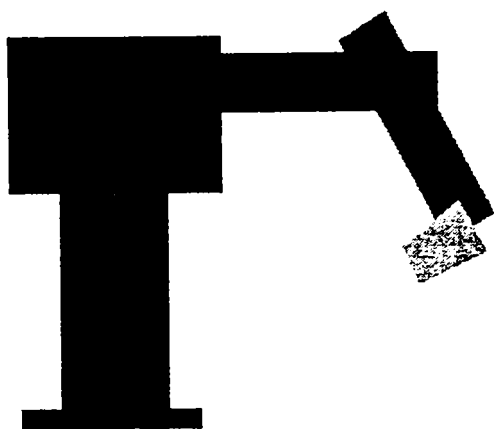


Fig.III.70. Sfericizarea tetraedrelor regulate

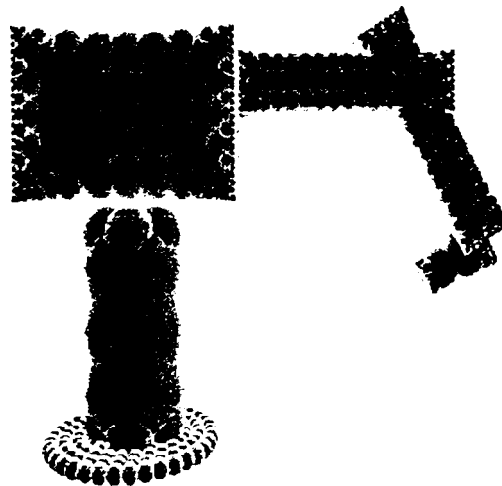
Un exemplu de reprezentare complexă prin sfericizare a robotului Scorbot din cadrul Laboratorului de Robotică al Catedrei OMM din Facultatea de Mecanică, prin programul de reprezentare astfel realizat, este prezentat în figura III.71, cu diferite precizii de reprezentare .



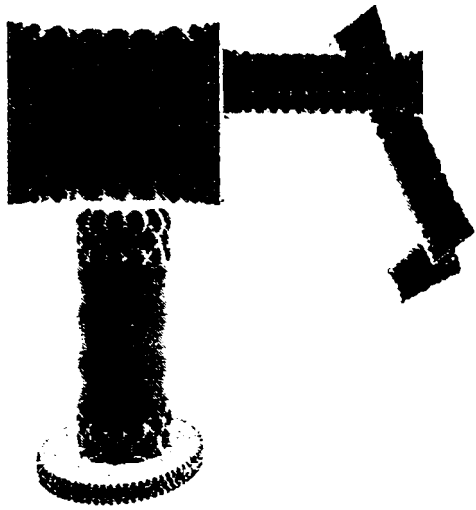
a. Robotul Scorbot în realitate



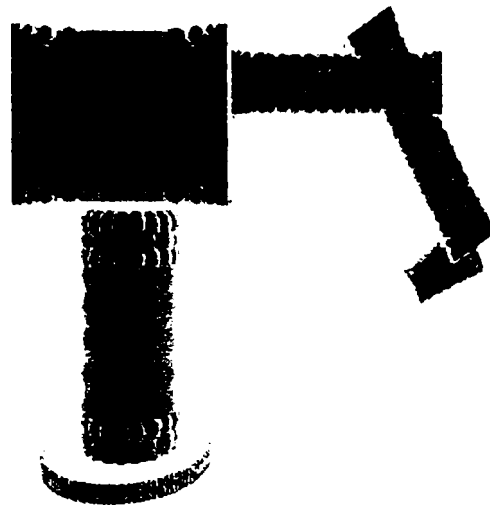
b.Reprezentarea 2D
a robotului Scorbot



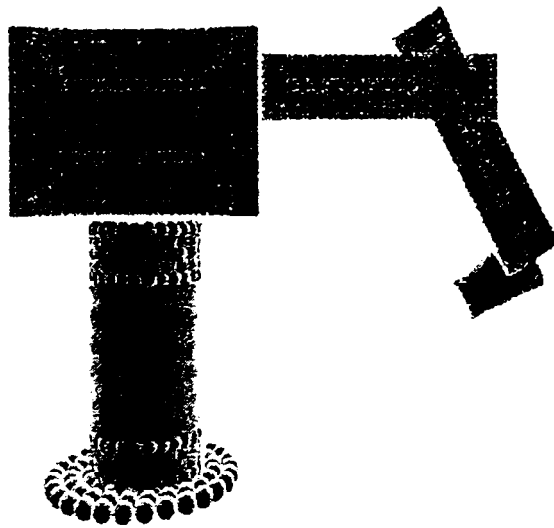
c.Reprezentarea 3D
a robotului Scorbot cu
precizie de 25%



c.Reprezentarea 3D
a robotului Scrobot
cu precizie de 50%



d.Reprezentarea 3D
a robotului Scrobot
cu precizie de 75%



d.Reprezentarea 3D
a robotului Scrobot cu
precizie de 100 %

Fig.III.71. Robotul Scrobot reprezentat prin sfericizare

IV. Reprezentarea prin subdivizare spațială

IV.1. Considerații generale

Metodele de reprezentare a sistemelor de corpuri prin tehnici de subdivizare spațială sunt utilizate frecvent pentru rezolvarea uneia dintre problemele foarte importante ale roboticii: planificarea mișcărilor [44], [45], [48], [70], [101], [104], [109], [110], [117], [140], [143].

În general, robotul inteligent trebuie să fie capabil să-și planifice propriile mișcări, și anume să decidă automat ce mișcări să execute pentru a realiza o sarcină specificată prin aranjamentul inițial și final al obiectelor din spațiul de lucru. Crearea roboților autonomi este o problemă cu mari aplicații și, din acest punct de vedere constituie un obiectiv important în robotică. S-a constatat că este mai util și mai economic să se doteze roboții cu dispozitive de planificare automată a mișcărilor în sistemele de programare off-line, decât să i se încorporeze sisteme de recunoaștere, respectiv de vedere artificială. Pentru că, în fond, robotul trebuie să realizeze mișcarea dorită și să-și activeze diferitele mecanisme în acord cu sarcina cerută.

Planificarea mișcărilor unui robot prezintă o varietate neașteptată de aspecte dificile de calcul. De fapt, inteligența operativă pe care oamenii o utilizează inconștient pentru a interacționa cu mediul înconjurător, necesară percepției și planificării mișcării, este foarte dificil de reprodus într-un program de calcul. Astfel, o problemă importantă în planificarea mișcărilor este complexitatea algoritmilor de calcul [20], [33], [40], [41], [42], [43], [136], [137], [139], [141], [154], [163], [168], [169], [171], [172], [183], [188], [200], [201], [202].

Așa cum s-a precizat, planificarea mișcării este o problemă în funcționarea roboților autonomi, dar nu este singura. Aceasta se coroborează cu alte probleme importante, proprii structurii însăși a robotului, ca: analiza cinematică directă și inversă, modelarea dinamică directă și inversă, generarea traiectoriilor, controlul

mișcării în timp real, sistemul și planificarea sarcini, etc., dar mai cu seamă cu necesitatea de a modela cât mai corect spațiile în care evoluează robotul și spațiul propriu pe care îl descrie efectorul său final în timpul execuției sarcinii [53], [55], [94], [101], [104], [115], [116], [121], [124], [125], [128], [130].

În capitolele II și III se prezintă câteva principii și metode de modelare aplicabile atât structurii mecanice complexe care constituie robotul cât și spațiului său propriu de evoluție, reprezentat de suprafața înfășurătoare descrisă de efectorul final în timpul execuției sarcinii. Aceleași metode se pot aplica și pentru modelarea corpurilor care constituie obstacolele din spațiul în care robotul evoluează.

Această modelare a obstacolelor este unul din obiectivele ce trebuie rezolvate în orice metodă de planificare a mișcării. Deoarece planificarea implică o succesiune de etape, un timp de calcul relativ lung și operații complexe, se urmărește ca modelarea obstacolelor să se realizeze în cea mai simplă manieră posibilă pentru ca timpul de realizare a modelului să fie optimizat prin minim. S-a constatat că cea mai simplă modalitate de reprezentare se obține prin divizarea spațiului în celule, cu forme și dimensiuni opționale, dependente de:

- forma inițială a sistemului de corpuri și amplasarea lor în spațiu;
- timpul de lucru considerat potrivit pentru aplicația respectivă;
- necesitatea abordării bi sau tri-dimensionale;
- performanțele calculatoarelor pe care se rulează aplicația, etc.

Cea mai elementară problemă de planificare presupune că doar robotul se află în mișcare în spațiul de lucru, abordarea realizându-se doar cinematic, caracteristicile dinamice nefiind interesante pentru programator la această etapă de studiu. Se consideră de asemenea că robotul nu intră în contact cu obiectele înconjurătoare, evitând astfel problemele legate de interacțiunea mecanică dintre două obiecte fizice. Aceste considerații transformă problema planificării fizice a mișcării într-o problemă pur geometrică.

Pentru a satisface restricțiile menționate în problema de planificare (în special non-coliziunea), robotul trebuie fie modelat prin înfășurarea într-o suprafață care să-l cuprindă în întregime. Această operație se poate realiza pe două căi:

- robotul să fie înfășurat în întregime într-o suprafață sferică în 3D, care îl acoperă complet;

- robotul să fie modelat printr-o reuniune de primitive geometrice (posibilitate prezentată în capitolul III);
- reprezentarea robotului să coincidă cu suprafața descrisă în mod real de efectorul său final în mișcare (posibilitate prezentată în capitolul II.5).

Varianta cea mai costisitoare, din punct de vedere al timpului de lucru, este cea de modelare a robotului prin primitive geometrice, așa cum s-a prezentat în capitolul III.2 unde robotul a fost reconstruit geometric prin sfericizare. Indiferent de varianta de modelare, în problemele de planificare, acest model al robotului în ansamblu se consideră ca un singur solid rigid.

Problema este uneori extinsă, în sensul că reprezentarea realizată în modalitățile descrise, va fi învăluită într-o suprafață identică, dar mai mare, care să asigure certitudinea existenței unei zone tampon și deci să înlăture orice posibilă coliziune.

Pe de altă parte, obstacolele pot fi și ele modelate prin aceleași metode ca și roboții, dar timpul de lucru suplimentar impus de reprezentarea unei structuri complexe a spațiului de lucru, nu se justifică. Aceasta afirmație se datorează în primul rând faptului că în planificare condiția esențială este să nu existe coliziune între robot și obstacole și astfel problema se simplifică adoptând pentru spațiul populat cu obstacole o structură celulară, exactă sau aproximativă, fiecare obstacol fiind acoperit de o reuniune oarecare a celulelor respective. Important este să se realizeze o astfel de descompunere în celule a spațiului, încât să poată fi identificată o succesiune de celule libere în care să se înscrie traiectoria robotului.

Cu aceste precizări, problema de bază a planificării traiectoriilor unui robot se poate formula astfel:

- Fie robotul A un solid rigid (obținut prin modelare) care se mișcă într-un spațiu de lucru 2D sau 3D, mișcarea lui nefiind limitată de nici o restricție cinematică.
- Fie B_j , mai multe obstacole considerate obiecte rigide fixe (încapsulate în celule) distribuite în poziții bine determinate în spațiul de lucru.
- Cunoscând poziția și orientarea robotului, la momentul inițial și cel final, să se genereze o traiectorie, care să specifice o secvență continuă de poziții și

orientări ale lui A , pornind de la configurația (poziție și orientare) inițială, evitând contactul cu obstacolele B_j și oprindu-se în configurația finală.

- Dacă o astfel de traiectorie nu există, atunci fie se caută aplicarea unei alte metode de modelare care să permită efectuarea mișcării, fie se renunță la căutarea unei traiectorii, mișcarea fără coliziuni nefiind posibilă.

Este evident că, deși problema de bază a planificării este extrem de simplificată față de cazurile puse de realitate, ea este totuși o problemă dificilă cu mai multe soluții și cu extensii directe spre probleme mai complicate.

Formularea problemei de bază a planificării traiectoriilor se bazează pe anumite presupuneri care limitează semnificativ utilizarea soluțiilor. Este foarte dificil să se reducă o problemă reală de robotică la problema de bază, să se rezolve această problemă și să se adapteze soluțiile obținute astfel încât să se potrivească condițiilor problemei reale. De multe ori însă, soluția furnizată de aplicarea problemei de bază este satisfăcătoare, căci deseori doar robotul este unicul mobil într-un spațiu populat cu obstacole fixe cunoscute a priori.

Problema de bază a planificării traiectoriei presupune că robotul parcurge cu exactitate traiectoria generată de planificator. De asemenea, se presupune că, atât geometria robotului, cât și geometria și pozițiile obstacolelor sunt cunoscute cu exactitate.

În realitate, nici o problemă de planificare nu satisface aceste ipoteze. Mai mult, comanda roboților și modelele geometrice ale acestora nu sunt exacte. În condițiile în care robotul nu deține informații inițiale despre spațiul de lucru, acesta trebuie să se bazeze, în timpul execuției, pe sistemul său senzorial pentru înregistrarea informațiilor necesare realizării sarcinii. Se impune astfel explorarea spațiului de lucru și rezolvarea problemei de planificare cu incertitudini.

Deoarece obiectul acestei teze îl constituie modelarea sistemelor complexe de corpuri, etapă ce intervine ca prim obiectiv în planificarea mișcării, se prezintă considerații legate de planificare doar în măsura în care acestea ilustrează o altă cale aleasă pentru modelare.

IV.2. Modelarea spațiului cu obstacole

Stabilirea unor căi de navigare în spațiul de lucru se poate face fie independent de orice acțiune a robotului prin memorarea unei configurații date, fie pe baza

informațiilor înregistrate de sistemul senzorial. Se alege prima variantă de modelare obstacolelor și a spațiului liber rămas în spațiul de lucru, pentru că obiectivul tezei este prezentarea de modele.

Pe baza cunoștințelor acumulate despre spațiul de lucru, acesta se împarte în celule independente. Aceste celule reprezintă zonele admise respectiv interzise pentru robot. Această operație se numește *modelarea spațiului de lucru al robotului*, pe care se bazează procesul propriu-zis de planificare a mișcărilor.

Având reprezentate obstacolele în forma în care ele se iau în considerare, se poate proceda la împărțirea spațiului de lucru în zone accesibile și zone interzise. Există mai multe metode pentru realizarea modelării spațiului: metoda grilei omogene, metoda arborelui, metoda grilei neomogene, metoda poligoanelor convexe, metoda punctelor de trecere, etc. Toate metodele enumerate au ca principiu investigarea de tip arbore și luarea deciziilor de forma admis-respins, sau accesibil-inaccesibil.

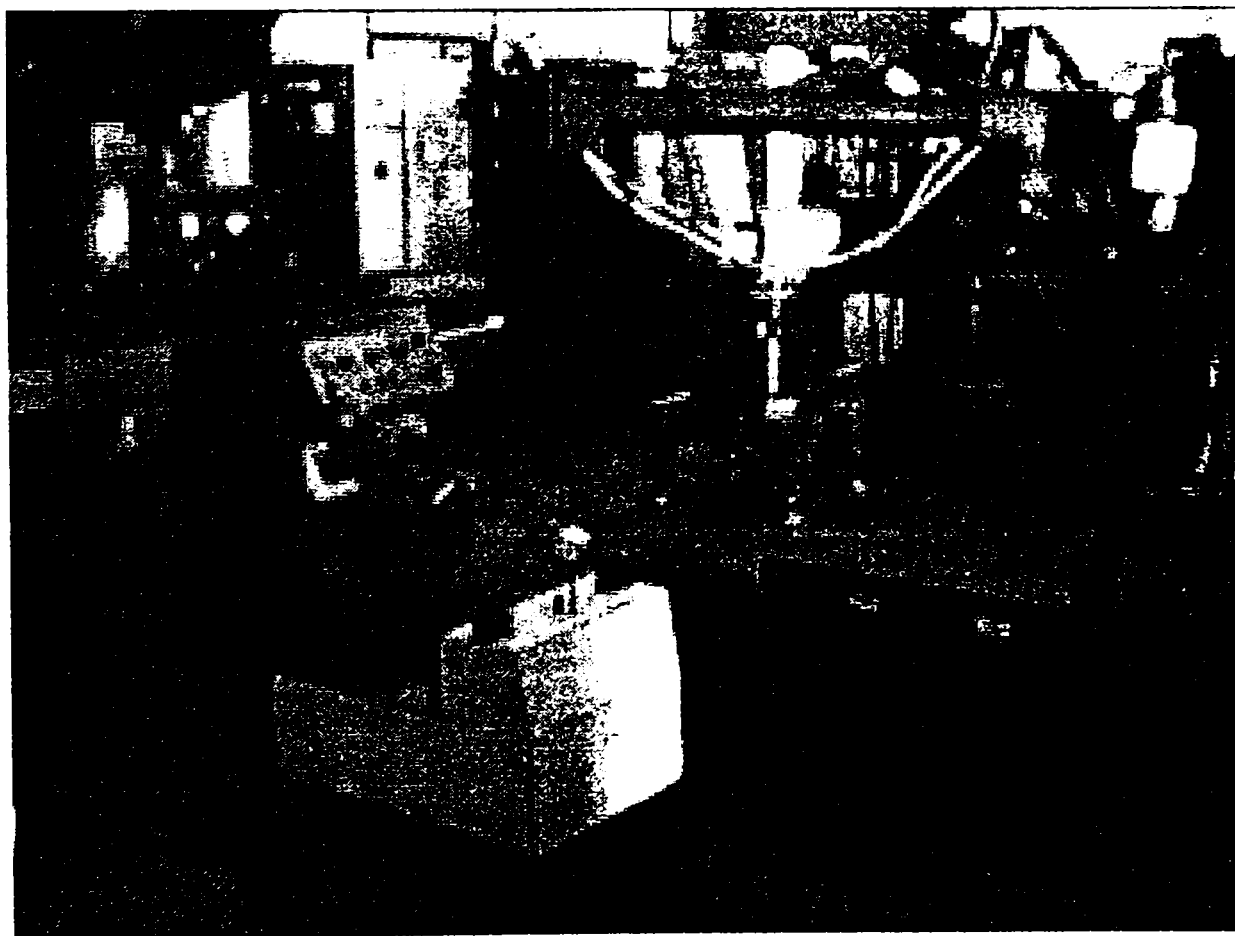


Fig.IV.1. Imagine din Laboratorul de Robotică pentru care se va studia modelarea spațiului liber în reprezentare 2D și 3D

Un grup de metode ce îmbină simplitatea reprezentării unui spațiu complex cu o viteză destul de mare de decizie în privința stabilirii traiectoriei, se bazează pe descompunerea aproximativă în celule a spațiului de lucru. Această descompunere poate fi realizată aproximativ utilizând celule de dimensiuni egale, motiv pentru care metoda de planificare a traiectoriei se numește *metoda grilei omogene*, sau în celule neegale, conducând la *metoda grilei neomogene*.

Ambele metode sunt exemplificate pentru spațiul de lucru existent în Laboratorul de Robotică din cadrul catedrei de Organe de Mașini și Mecanisme, din care este prezentată o imagine în figura IV.1.

IV.2.1. Modelarea 2D a spațiului de lucru

IV.2.1.1. Metoda grilei omogene

Metoda grilei omogene este o metodă de descompunere aproximativă a unui spațiu. Cele mai multe metode de descompunere celulară aproximativă permit ca dimensiunea celulelor să fie în concordanță cu geometria obstacolelor. A prestabili dimensiunea celulelor poate însă conduce la dificultăți: o dimensiune mare a celulei poate avea ca efect negăsirea vreunui drum, în timp ce o dimensiune prea mică are ca efect un timp de calcul prea mare. De aceea, cele mai multe metode se bazează pe un procedeu ierarhic, generând inițial o descompunere mai grosolană, cu precizie de reprezentare scăzută, pentru ca în etape ulterioare să se realizeze rafinarea celulelor și să se obțină o descompunere mai precisă [52], [63], [65], [70], [97], [198].

Rațiunile pentru standardizarea formei celulelor sunt:

- obținerea unei descompuneri a spațiului prin iterații simple;
- stabilirea unei metode de reprezentare insensibilă la aproximarea numerică;
- implementări simple;
- controlul direct al mărimii spațiului liber de-a lungul unei căi generate prin setarea unei dimensiuni minime impusă celulelor.

Metoda grilei omogene ține seama mai puțin de structura matematică a spațiului și chiar poate da greș la găsirea unui drum pentru robot, chiar și atunci când acesta există.

Considerând spațiul de lucru W , acesta se tratează ca o mulțime finită de celule pătrate $\{k_i\}_{i=1, \dots, r}$.

O celulă k_i poate fi:

1. *liberă*, dacă și numai dacă interiorul său aparține în întregime spațiului liber;
2. *ocupată*, dacă și numai dacă interiorul său aparține în întregime obstacolelor;
3. *mixtă*, al cărei interior aparține parțial unui obstacol și parțial unei zone libere.

Două celule sunt adiacente dacă și numai dacă au o latură comună.

Fiind dată o descompunere a lui W , se definește un *canal* ca o secvență de celule *libere* și/sau *mixte*, astfel încât orice două celule consecutive sunt adiacente. Un canal care conține numai celule *libere* este cu siguranță utilizabil pentru o traiectorie fără coliziuni, iar un canal care conține și celule *mixte*, mai trebuie studiat prin rafinarea descompunerii pentru a stabili dacă poate conține o traiectorie, sau nu.

Programul ce realizează descompunerea celulară aproximativă a unui spațiu dat, în celule pătrate identice, a fost scris în echivalentul sub mediu WINDOWS a limbajului de programare Borland Pascal, DELPHI.

S-a ales WINDOWS -ul ca mediu de dezvoltare a aplicației, pentru a satisface tendința normală de migrare spre medii mai performante care dispun de facilități grafice puternice precum și de o interfață prietenoasă cu utilizatorul.

Programarea în acest mediu a permis folosirea ferestrelor pentru dialogul cu utilizatorul, meniurile selectabile cu mouse-ul, un mini-help eficient bazat pe buline de help, care direcționează utilizatorul.

➤ Structura programului

Ca structură, programul se compune din fereastra principală care conține și meniul principal. Selectarea oricărei opțiuni a meniului principal are ca rezultat deschiderea altei ferestre în care se realizează funcția dorită. Astfel, opțiunile meniului principal sunt:

Info

Considerații teoretice

Crearea unui nou spațiu

Descompunerea unui spațiu dat

Ieșire

❖ *Info*

Selectarea acestei opțiuni duce la afișarea pe ecran a unei ferestre în care se dau unele informații suplimentare privitoare la program.

❖ *Considerații teoretice*

În urma activării cu mouse-ul a butonului cu această inscripție din meniul principal, se deschide o fereastră ce permite citirea considerațiilor teoretice care au stat la baza realizării programului [158].

❖ *Crearea unui nou spațiu*

Această opțiune a meniului lansează în execuție modulul de realizare grafică a spațiului de lucru pentru robot, care urmează a fi descompus celular. Spațiul gol inițial și care urmează a fi populat cu figuri geometrice pentru redarea suprafeței pe care robotul urmează a evolua, este reprezentat de o suprafață de desen având dimensiunile 320x320 pixeli. Pe aceasta se vor plasa obstacolele prin selectarea elementelor grafice din lista prezentată alăturat acestei suprafețe și prin indicarea cu mouse-ul a locului unde se dorește poziționarea obstacolului.

Spațiul poate fi populat cu obstacole poligonale și de aceea utilizatorul are la dispoziție următoarele elemente grafice:

- *Dreptunghi*, având laturile liber modificabile prin intermediul unor casete de introducere de date; Prin această metodă se pot desena și pătrate, alegând dimensiunile laturilor egale;
- *Triunghi* de dimensiuni date de către utilizator; pentru acesta se vor specifica dimensiunile pentru două laturi alăturate și unghiul dintre ele;
- *Poligon oarecare*, constructibil cu mouse-ul, având practic un număr nelimitat de laturi; se vor desena laturile, atâtea câte se doresc, apoi se va acționa asupra butonului drept al mouse-ului pentru închiderea și umplerea conturului poligonal;

- *Cerc*, de rază variabilă, care se poate introduce la valoarea dorită.

În cazul în care s-a făcut o greșeală de reprezentare, ea se poate corecta cu ajutorul gumei de șters care curăță de pe suprafața de lucru câte un pătrat de latura de 10 pixeli.

Ștergerea întregii imagini se face acționând butonul *Ștergere imagine*, resetând astfel suprafața de lucru.

În urma creării spațiului, imaginea va fi salvată prin acționarea butonului *Salvare*. În acest moment se deschide o fereastră de alegere a numelui fișierului și locului în care acesta se va salva. Se pot alege directorul și unitatea de disc dorită. Fișierul este de tip bitmap având extensia .BMP.

❖ *Descompunerea unui spațiu dat*

În cazul în care există spații create anterior, acestea pot fi încărcate și modificate prin apelarea butonului *Încărcare*. Procedura se derulează asemănător procedurii de salvare. Și aceste spații deja create, sunt salvate tot cu extensia .BMP.

❖ *Ieșire*

Prin selectarea butonului de ieșire se revine la meniul principal după ce, utilizatorul își dă acordul asupra acestei acțiuni, pentru a nu se ieși din greșeală și de a pierde eventual cele lucrate până în momentul respectiv.

În prima etapă de aplicare a algoritmului de modelare a spațiului prin metoda grilei omogene, s-a urmărit realizarea modelului unei secțiuni plane a unei porțiuni a spațiului Laboratorului de Robotică populat cu obstacole. S-a selectat opțiunea *Crearea unui nou spațiu* și s-a încercat o reprezentare cât mai apropiată de cea reală.

Pentru construirea spațiului, conform celor precizate, se utilizează figurile geometrice aflate alături de suprafața ce urmează a fi reprezentată (figura IV.2).

S-a realizat porțiunea din suprafața laboratorului care interesa pentru modelare și s-a salvat sub denumirea de *lab.bmp*. Apoi s-a activat opțiunea *Descompunere celulară* care până la această etapă era neselectabilă și care acum este gata de activare. Programul oferă facilitatea alegerii dimensiunii dorite a celulelor, adică a gradului de rafinare a reprezentării.

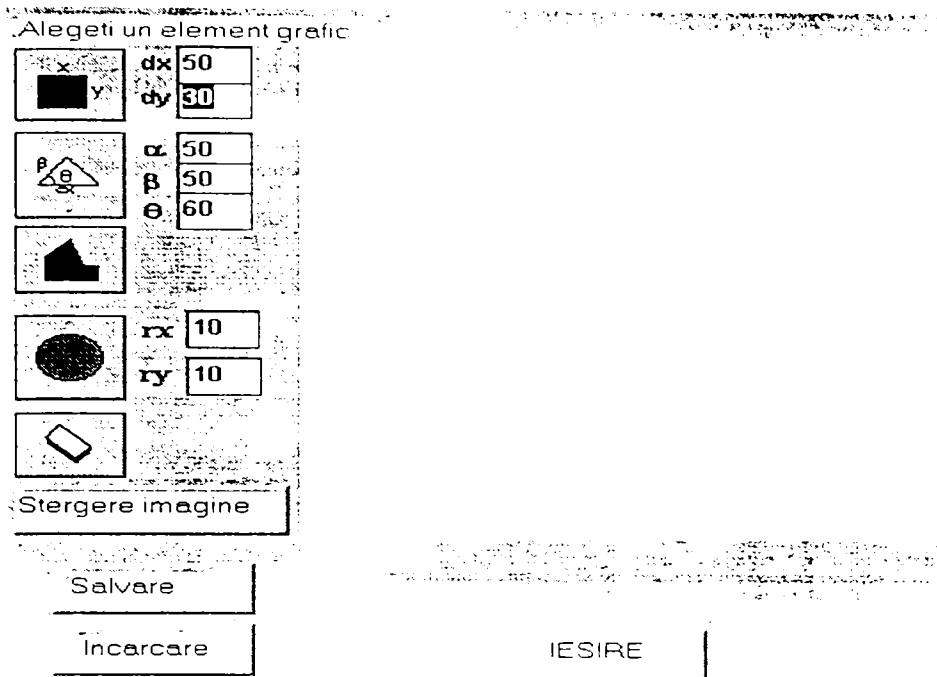


Fig.IV.2. Posibilitățile oferite de program pentru crearea unui spațiu ce urmează a fi descompus

În urma selectării cu mouse-ul, descompunerea celulară aproximativă propriu-zisă începe, în partea de jos a ferestrei apărând un mesaj care roagă utilizatorul să aștepte. Această așteptare este de fapt timpul în care se efectuează calculele necesare realizării descompunerii celulare aproximative și a reprezentării celulelor.

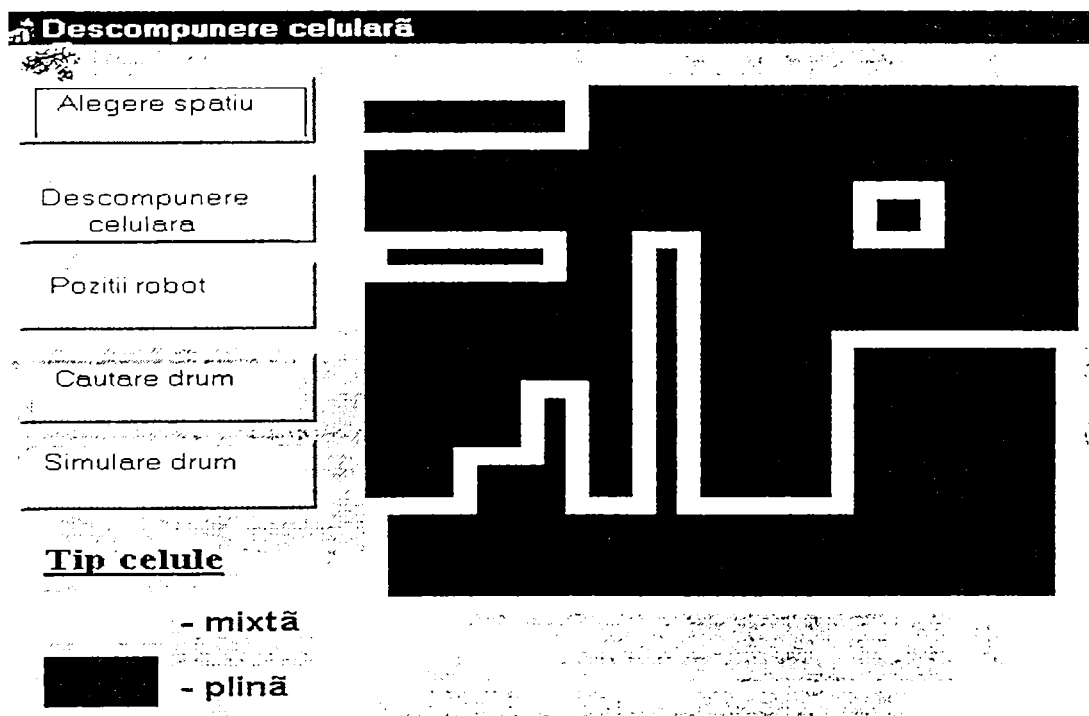


Fig.IV.3. Reprezentarea 2D a spațiului de lucru din Laboratorul de Robotică destinată aplicării metodei grilei omogene

În figura IV.3 este reprezentată porțiunea selectată a laboratorului unde se dorește modelarea obstacolelor și trasarea traiectoriei, alegându-se cea mai fină reprezentare a celulelor pe care o poate oferi programul. În timpul descompunerii celulare se pot observa celulele care apar pe măsura analizei spațiului. Celulele de diferite tipuri sunt reprezentate cu diferite culori:

- celulele goale sunt colorate în albastru;
- celulele pline sunt colorate în roșu;
- celulele mixte sunt hașurate cu alb, având chenar galben.

Finețea descompunerii poate fi modificată, o valoare mai mare având ca rezultat o eroare mai mică, adică o aproximare mai bună a împărțirii spațiului în celule pline și goale, în celulele mixte fiind cuprinsă o suprafață mai mică din spațiu.

➤ *Reprezentarea traiectoriei*

După ce descompunerea celulară a fost realizată, devine activabil butonul cu inscripția *Drum robot*. În urma selectării acestei opțiuni se indică cu mouse-ul celula inițială și cea de destinație a robotului, acesta urmând să-și găsească drumul numai în spațiul liber determinat de succesiunea celulelor goale. Între aceste două celule date, programul încearcă să găsească o succesiune numai de celule goale adiacente.

În cazul în care nu există un asemenea drum utilizatorului i se comunică acest lucru, iar în caz contrar se afișează drumul printr-un canal format din celule adiacente. În figura IV.4 este indicat un astfel de canal pentru spațiul de lucru descompus în figura IV.3 și pozițiile inițială și finală, menționate.

Canalul ales este cel cu lungime minimă, aceasta fiind soluția aleasă de program din mulțimea canalelor libere posibile între cele două celule de la capete.

Este evident că metoda de modelare prezentată presupune implicit că robotul are el însuși dimensiunea unei celule. Deci, un criteriu de alegere a dimensiunii celulei și implicit de precizie a metodei de modelare este cea impusă de dimensiunile robotului.

Apelarea butonului de *Ieșire* cere confirmarea utilizatorului pentru părăsirea descompunerii celulare și, în cazul acordului acestuia, se revine la meniul principal.

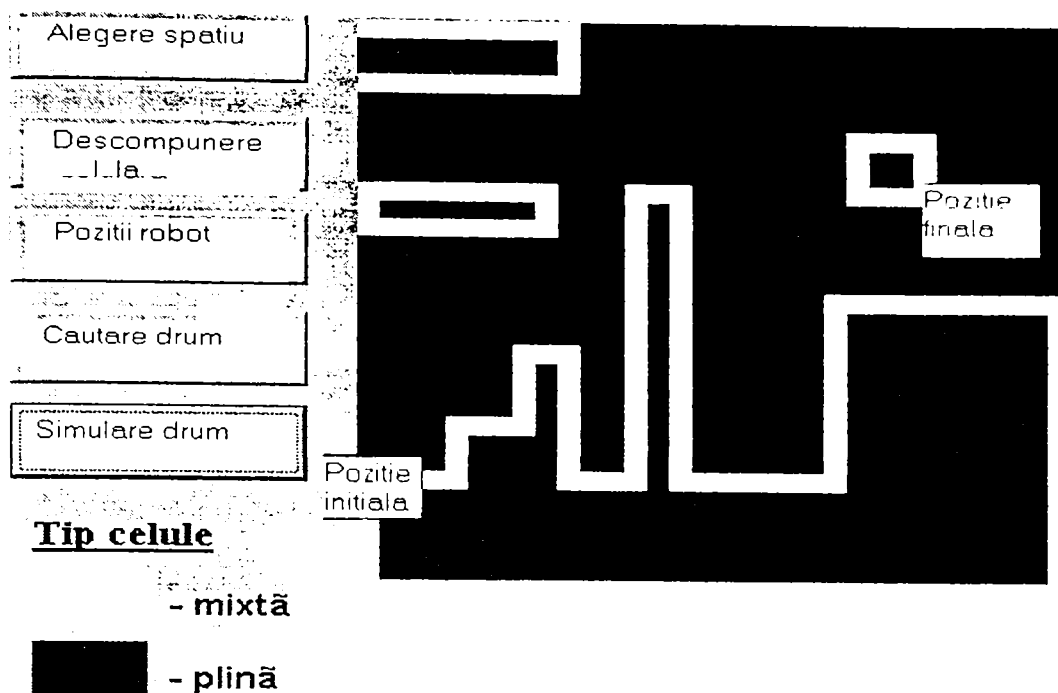


Fig.IV.4. Un canal în care se poate afla traiectoria robotului pentru pozițiile inițială și finală impuse

IV.2.1.2. Metoda grilei neomogene

Metoda presupune toate obstacolele de formă dreptunghiulară. Dacă ele nu au această formă se vor aproxima printr-un dreptunghi acoperitor. Existând n obstacole în spațiul de lucru, acesta se împarte printr-o grilă rezultată din prelungirea laturilor obstacolelor, obținându-se $(2n+1)^2$ dreptunghiuri sau, altfel spus, $2n+1$ benzi orizontale și $2n+1$ benzi verticale (figura IV.5).

Fiecare regiune din grilă este reprezentată prin două lanțuri binare de câte $2n+1$ biți, dintre care unul reprezintă poziția relativă pe orizontală, iar celălalt poziția relativă pe direcția verticală. Un bit are valoarea 1 dacă celula este liberă și 0 în caz contrar [67], [70], [184].

Modelarea se realizează în șase etape (exemplificările pe etape fiind valabile pentru spațiul prezentat în figura IV.5):

- ❖ *Etapa 1:* Reprezentarea fiecărei benzi orizontale prin două lanțuri de biți. Primul lanț conține $2n+1$ biți respectând succesiunea celulelor grilei și marcându-se cu valoarea 1 dacă celula este liberă și cu 0 în caz contrar. Un al doilea lanț conține un singur 1 care corespunde la poziția verticală a benzii.

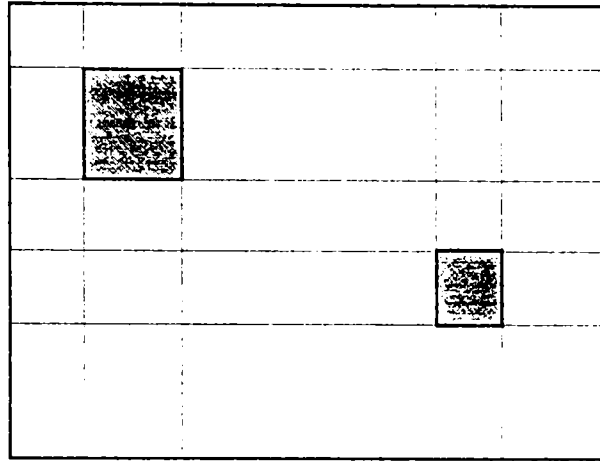


Fig.IV.5. Acoperirea unui spațiu cu două obstacole printr-o grilă neomogenă

- ❖ *Etapa 2:* Determinarea tuturor benzilor orizontale continue rezultate din separarea primului lanț în mai multe, fiecare având o suită continuă de 1.

De exemplu:

1 1 1 0 1 0 0 0 1 0

devine:

1 1 1 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0 1 0

- ❖ *Etapa 3:* Gruparea lanțurilor generate în etapa a doua, astfel:

- ele să fie regrupate în raport cu benzile care le-au generat;
- grupele să fie ordonate în raport cu poziția verticală a benzilor generatoare.

- ❖ *Etapa 4:* Generarea unei noi liste pornind de la cea precedentă conform următoarelor reguli:

- noua grupă i de lanțuri este generată prin combinarea fiecărui lanț din vechea grupă i cu fiecare lanț din vechea grupă $i+1$ ($i = 1, 2, \dots$);
- două lanțuri sunt combinate printr-un **ȘI** logic pentru lanțurile din stânga și **SAU** logic pentru lanțurile din dreapta; dacă toți biții sunt nuli, lanțul respectiv se elimină. dacă nu, se include pe listă;

- de fiecare dată când un lanț este adăugat la noua listă se elimină lanțurile de pe lista precedentă care sunt acoperite de această adăugare; un lanț S_1 este acoperit de către S_2 dacă un SAU logic între ele conduce la S_2 .

- ❖ *Etapa 5:* Se repetă etapa 4 dacă noua listă are două sau mai multe grupe.
- ❖ *Etapa 6:* Se stabilesc lanțurile neeliminate de pe liste care reprezintă o zonă fără obstacole

Pentru spațiul cu două obstacole din figura IV.5 derularea etapelor menționate conduce la stabilirea unor zone libere marcate în figura IV.6, prin ale căror intersecții se obțin porțiunile prin care traiectoria robotului poate fi trasată. Ca și în cazul precedent se realizează un canal de celule adiacente, iar rezultatul cercetării acestui canal și al trasării grafului de conexiuni va conduce, după modelare, la posibilitatea de trasare a traiectoriei.

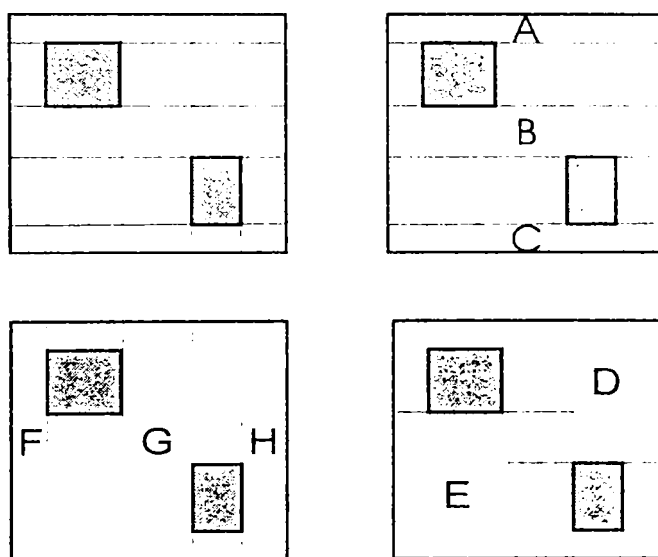


Fig.IV.6. Etape ale derulării metodei de modelare a spațiului 2D prin grilă neomogenă

Sucesiunea de liste de date și de etape ale calculului este:

Lista 1 (Etapa 1)

```

1 1 1 1 1 1 0 0 0 0
1 0 1 1 1 0 1 0 0 0
1 1 1 1 1 0 0 1 0 0
1 1 1 0 1 0 0 0 1 0
1 1 1 1 1 0 0 0 0 1
    
```

Lista 2 (Etapa 2+3)

1 1 1 1 1 1 0 0 0 0 (A)

1 0 0 0 0 0 1 0 0 0

0 0 1 1 1 1 0 1 0 0

1 1 1 1 1 1 0 0 1 0 0 (B)

1 1 1 0 0 0 0 0 1 0

0 0 0 0 1 0 0 0 1 0

1 1 1 1 1 1 0 0 0 0 1 (C)

Lista 3 (Etapa 4)

1 0 0 0 0 1 1 0 0 0

0 0 1 1 1 1 1 1 0 0 0

1 0 0 0 0 0 1 1 1 0 0

0 0 1 1 1 1 0 1 1 0 0

1 1 1 0 0 0 0 1 1 1 0

0 0 0 0 1 0 0 1 1 1 0

1 1 1 0 0 0 0 0 1 1 1

0 0 0 0 1 0 0 0 1 1 1

Lista 4 (Etapa 5)

1 0 0 0 0 1 1 1 1 0 0

0 0 1 1 1 1 1 1 1 0 0 (D)

1 0 0 0 0 0 1 1 1 1 0

0 0 1 0 0 0 1 1 1 1 0

0 0 0 0 1 0 1 1 1 1 0

1 1 1 0 0 0 0 1 1 1 1 (E)

0 0 0 0 1 0 0 1 1 1 1

Lista 5 (Etapa 5)

1 0 0 0 0 1 1 1 1 1 0

0 0 1 0 0 1 1 1 1 1 0

0 0 0 0 1 1 1 1 1 1 0

1 0 0 0 0 0 1 1 1 1 1

0 0 1 0 0 0 1 1 1 1 1

0 0 0 0 1 0 1 1 1 1 1

Lista 6 (Etapa 5)

1 0 0 0 0 1 1 1 1 1 1 (F)

0 0 1 0 0 1 1 1 1 1 1 (G)

0 0 0 0 1 1 1 1 1 1 1 (H)

Metodei astfel descrise i se poate asocia un graf ale cărui noduri sunt zonele fără obstacole (figura IV.7). La fiecare grup de liste unde s-a putut deja stabili cu exactitate celula liberă, notația corespunzătoare nodului din graf este prezentată alături de listele respective.

Arce	Noduri	Intersecții	
a	AF	10000	10000
b	AG	00100	10000
c	AH	00001	10000
d	HF	10000	00100
e	BG	00100	00100
f	BH	00001	00100
g	AD	00111	10000
h	BD	00111	00100
i	GD	00100	00111
j	HD	00001	00111
k	BE	11100	00100
l	CE	11100	00001
m	FE	10000	00111
n	GE	00100	00111
o	DE	00100	00100
p	CF	10000	00001
q	CG	00100	00001
r	CH	00001	00001

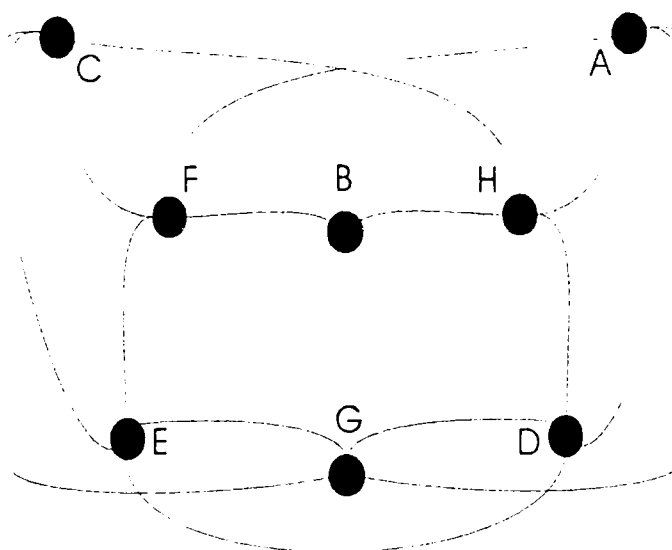


Fig.IV.7. Graful de conexiuni corespunzător spațiului modelat prin grilă neomogenă

➤ *Utilizarea programului*

Pentru a lucra în modul 2D se creează un nou spațiu bidimensional de lucru sau se încarcă unul deja existent.

Pentru un nou spațiu trebuie specificate câteva opțiuni fereastra de dialog din figura IV.8:

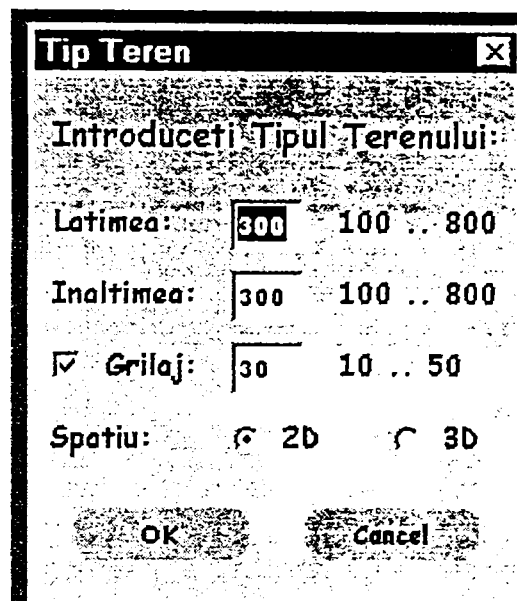


Fig.IV.8. *Fereastra de dialog pentru exprimarea opțiunilor de aplicare a descompunerii neomogene*

Lățimea și înălțimea specifică dimensiunile spațiului bidimensional, cuprinse între 100 și 800 de pixeli, iar grilajul presupune afișarea în cadrul terenului a unor puncte de referință din n în n pixeli, unde $10 \leq n \leq 50$.

După exprimarea dimensiunilor dorite și a opțiunii de reprezentare 2D sau 3D, programul afișează spațiul în care se vor plasa obstacolele, sub forma din figura IV.9.

Editarea obstacolelor se realizează cu mouse-ul prin fixarea unor puncte ce reprezintă colțurile unui obstacol printr-o apăsare pe butonul stâng. Unirea primului punct cu ultimul și definitivarea obstacolului se realizează prin apăsare pe butonul drept. Același spațiu bidimensional prezentat și în figura IV.3 și descompus prin grilă omogenă este prezentat în figura IV.10 în vederea descompunerii prin grilă neomogenă.

După terminarea editării există posibilitatea salvării spațiului sau începerii derulării etapelor procedurii de descompunere. Pentru a aplica metoda de găsimă a drumului minim între două puncte alese ale terenului trebuie acționat butonul de

forma unui bec, marcat printr-o curbă neregulată în figura IV.9, pe bara de unelte plasată sub meniul programului.

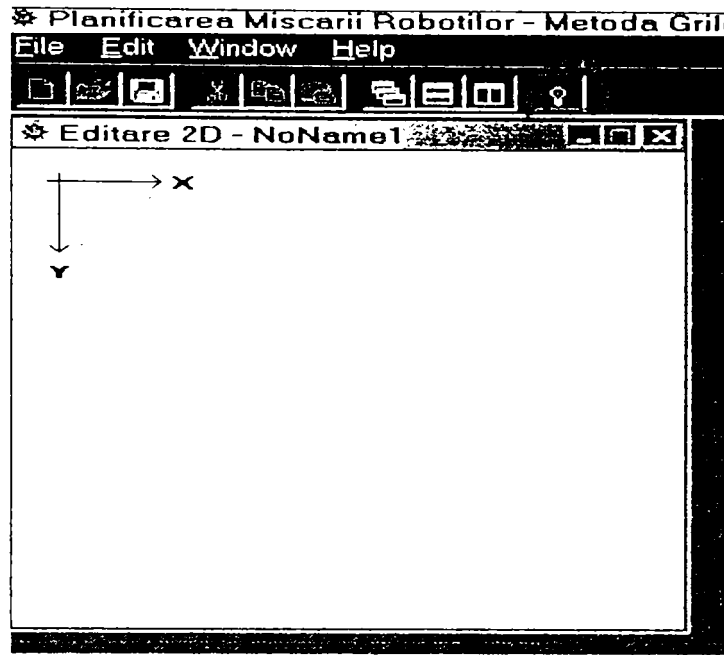


Fig.IV.9. Planul pe care se va reprezenta spațiul ce urmează a fi modelat

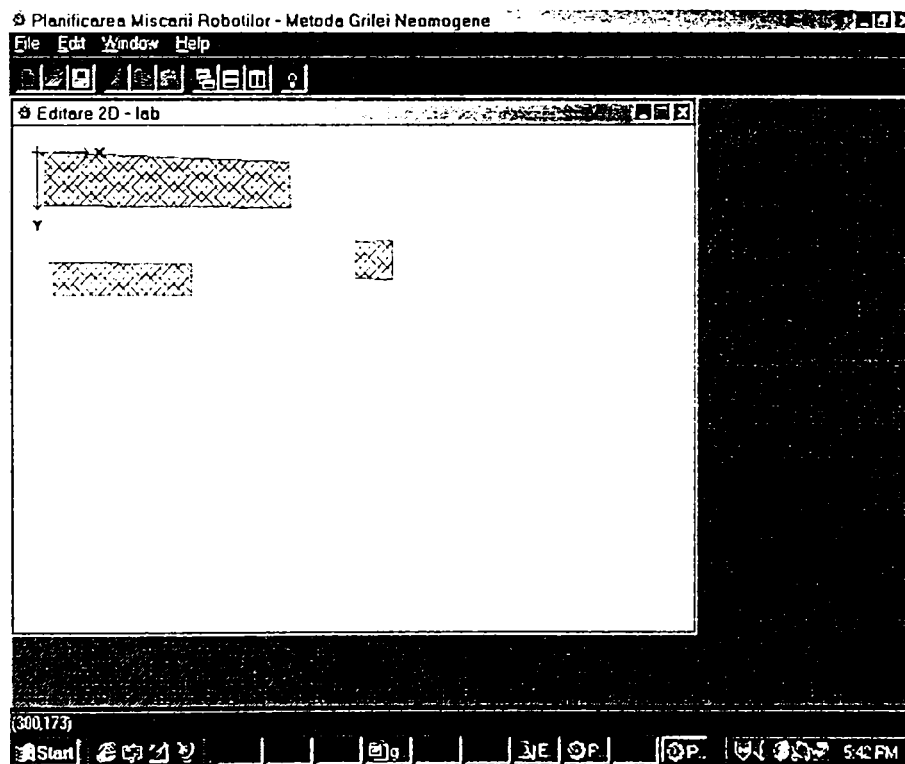


Fig.IV.10. Spațiul ce urmează a fi descompus prin grilă neomogenă

În continuare se vor urmări pașii metodei de calcul, urmând ca după fiecare pas să se apese butonul de continuare, aflat în dreapta ecranului alături de suprafața de descompus, din cadrul ferestrei de comandă a metodei. În figura IV.11 este reprezentată grila ce acoperă spațiul de studiat și de la care începe procesarea prin scrierea lanțurilor de biți.

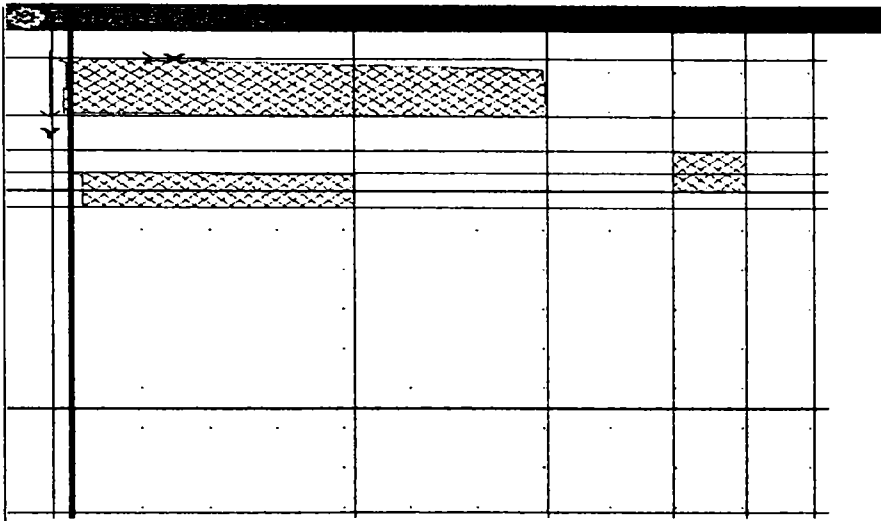


Fig.IV.11. Grila neomogenă pentru spațiul populat cu obstacole

Pasul următor al algoritmului de modelare, acoperă obstacolele cu dreptunghiuri formate din reuniuni de celule întregi, reprezentarea fiind prezentată în figura IV.12.

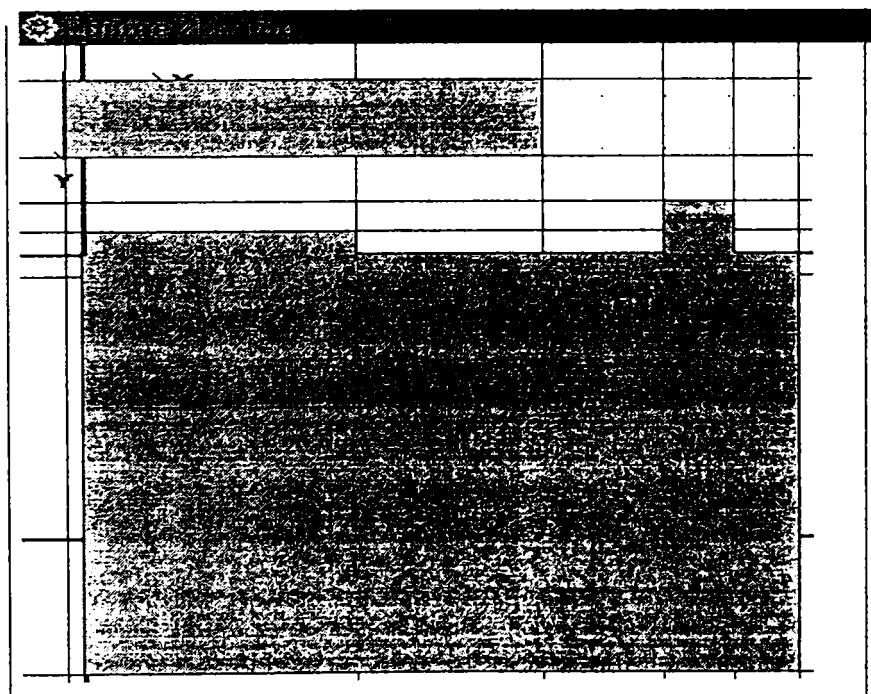


Fig.IV.12. Acoperirea obstacolelor cu dreptunghiuri formate din celulele ocupate ale grilei

În continuare se solicită introducerea punctului de plecare și a celui de sosire ale robotului, operație care se realizează printr-o simplă fixare a două puncte în spațiul de lucru bidimensional (figura IV.13).

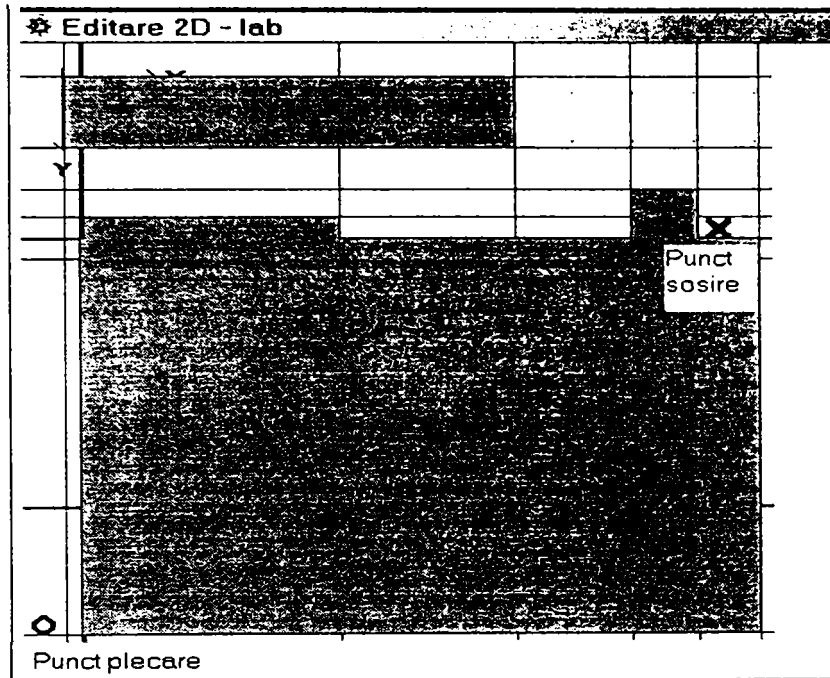


Fig.IV.13. Spațiul de lucru cu obstacolele acoperite cu celule având marcate punctele de capăt ale traiectoriei

Se urmăresc apoi benzile de liberă trecere a robotului printr-o animație a acestora, benzi ce sunt prezentate și sub forma lanțurilor într-o fereastră plasată sub spațiul de lucru și care se prezintă după cum urmează:

Lista : 1

1111111111	1000000000	0
1000000000	0100000000	
0000001111	0100000000	
1111111111	0010000000	1
1111111100	0001000000	
0000000011	0001000000	
1111100000	0000100000	
0000001100	0000100000	
0000000011	0000100000	
1100000000	0000010000	
0000000001	0000010000	
1100000000	0000001000	

0000000001	0000010000
1100000000	0000001000
0000000001	0000001000
1100000000	0000000100
0000000001	0000000100
1100000000	0000000010
0000000001	0000000010
1111111111	0000000001 2

Lista : 2

1000000000	1100000000
0000001111	1100000000
1000000000	0110000000
0000001111	0110000000
1111111000	0011000000 3
0000000011	0011000000
1111100000	0001100000
0000011000	0001100000
0000000011	0001100000
1100000000	0001100000
0000000001	0001100000
1100000000	0000110000
0000000001	0000110000
1100000000	0000110000
0000000001	0000110000
1100000000	0000011000
0000000001	0000011000
1100000000	0000001100
0000000001	0000001100
1100000000	0000000110
0000000001	0000000110
1100000000	0000000011
0000000001	0000000011

Lista : 3

1000000000	1110000000
0000001111	1110000000 4
1000000000	0111000000
0000001000	0111000000
0000000011	0111000000
1111100000	0011100000 5
0000011000	0011100000 6
0000000011	0011100000
1100000000	0001110000
0000000001	0001110000
1100000000	0001110000
0000000001	0001110000
1100000000	0000111000
0000000001	0000111000
1100000000	0000011100
0000000001	0000011100
1100000000	0000001110
0000000001	0000001110
1100000000	0000000111
0000000001	0000000111
1100000000	0000000011
0000000001	0000000011

Lista : 4

1000000000	1111000000
0000001000	1111000000
0000000011	1111000000
1000000000	0111100000
0000001000	0111100000
0000000011	0111100000
1100000000	0011110000
0000000001	0011110000
1100000000	0001111000
0000000001	0001111000
1100000000	0000111100
0000000001	0000111100
1100000000	0000011110
0000000001	0000011110
1100000000	0000001111
0000000001	0000001111

Lista : 5

1000000000	1111100000
0000001000	1111100000 7
0000000011	1111100000 8
1000000000	0111110000
0000000001	0111110000
1100000000	0011111000
0000000001	0011111000
1100000000	0001111100
0000000001	0001111100
1100000000	0000111110
0000000001	0000111110
1100000000	0000011111
0000000001	0000011111
1100000000	0000001111
0000000001	0000001111

Lista : 6

1000000000	1111110000
0000000001	1111110000
1000000000	0111111000
0000000001	0111111000
1100000000	0011111100
0000000001	0011111100
1100000000	0001111110
0000000001	0001111110
1100000000	0000111111
0000000001	0000111111
1100000000	0000011111
0000000001	0000011111

Lista : 7

1000000000	1111110000
0000000001	1111110000
1000000000	0111111000
0000000001	0111111000
1100000000	0011111100
0000000001	0011111100
1100000000	0001111110
0000000001	0001111110
1100000000	0001111111
0000000001	0001111111

Lista : 8

1000000000	1111111000
0000000001	1111111000
1000000000	0111111100
0000000001	0111111100
1100000000	0011111110
0000000001	0011111110
1100000000	0001111111
0000000001	0001111111

Lista : 9

1000000000	1111111100
0000000001	1111111100
1000000000	0111111110
0000000001	0111111110
1100000000	0011111111 9
0000000001	0011111111

Lista : 10

1000000000	1111111110
0000000001	1111111110
1000000000	0111111111
0000000001	0111111111

Lista : 11

1000000000	1111111111 10
0000000001	1111111111 11

Alături de lanțurile de biți din dreapta, sunt afișate și nodurile grafului asociat metodei de reprezentare.

In etapa următoare, va fi afișat graful și drumul minim pe graf (figura IV.14).

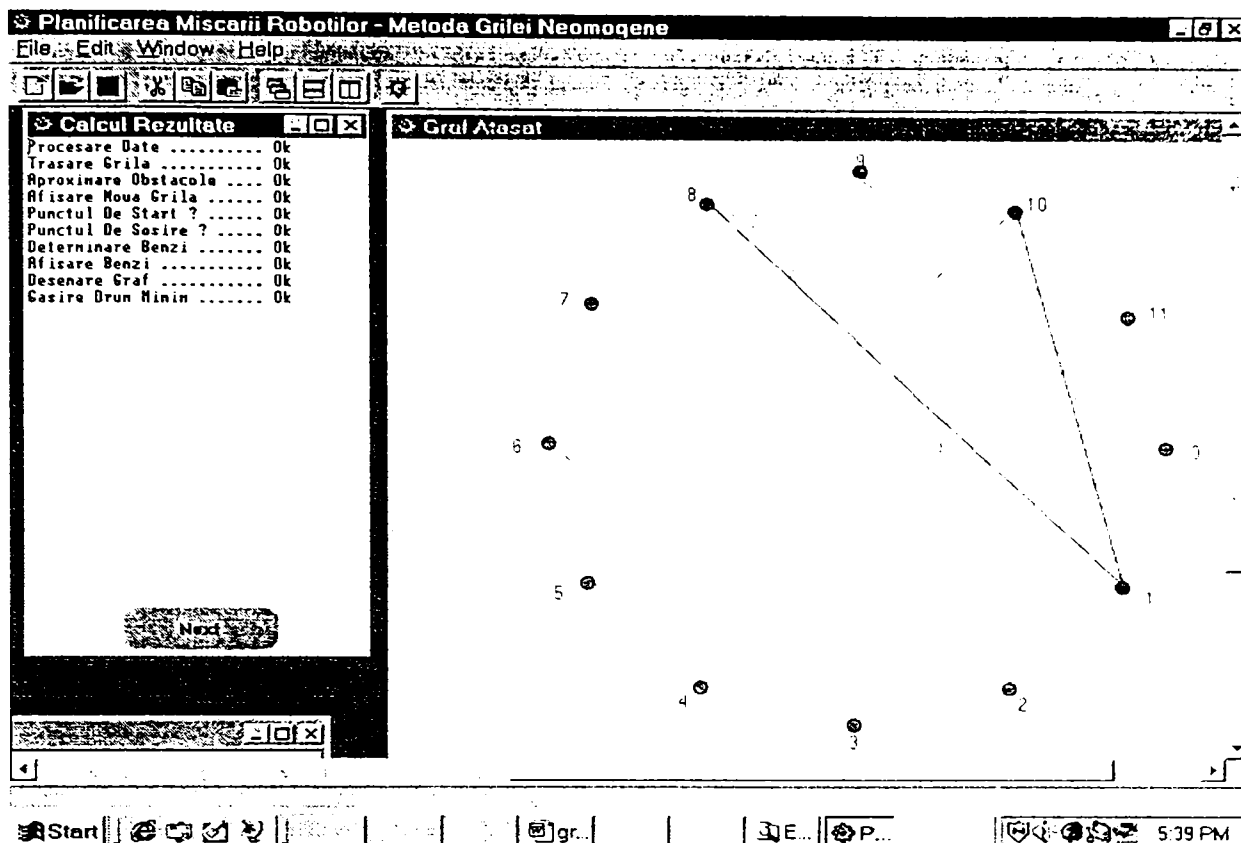


Fig.IV.14. Graful asociat metodei de descompunere prin grilă neomogenă
în 2D cu indicarea succesiunii de zone libere 8-1-10

După terminarea calculelor se solicita apăsarea pentru ultima dată a butonului de continuare care are ca efect închiderea tuturor ferestrelor deschise, inclusiv a spațiului bidimensional pe care s-a lucrat.

IV.2.2. Modelarea 3D a spațiului de lucru

În principiu modelarea se derulează însă la fel ca în spațiul 2D, cu aceleași etape, funcție de metoda adoptată. Se prezintă metoda descompunerii spațiului prin grila neomogenă, care se derulează la fel, dar în fereastra de dialog din figura IV.8 se va alege opțiunea 3D.

Pentru a lucra în modul 3D se creează un nou spațiu tridimensional de lucru sau se încarcă unul deja existent. În sistemul de referință plasat în colțul din stânga ecranului sus se afla un cub curent cu ajutorul căruia se realizează obstacolele reale. Scena de lucru este reprezentată în figura IV.15, unde se află o vedere parțială a spațiului de lucru din Laboratorul de Robotică, cu câteva obstacole din jurul unui stâlp de susținere a halei.

În colțul din stânga sus, se observa cubul curent cu ajutorul căruia se construiesc obstacolele.

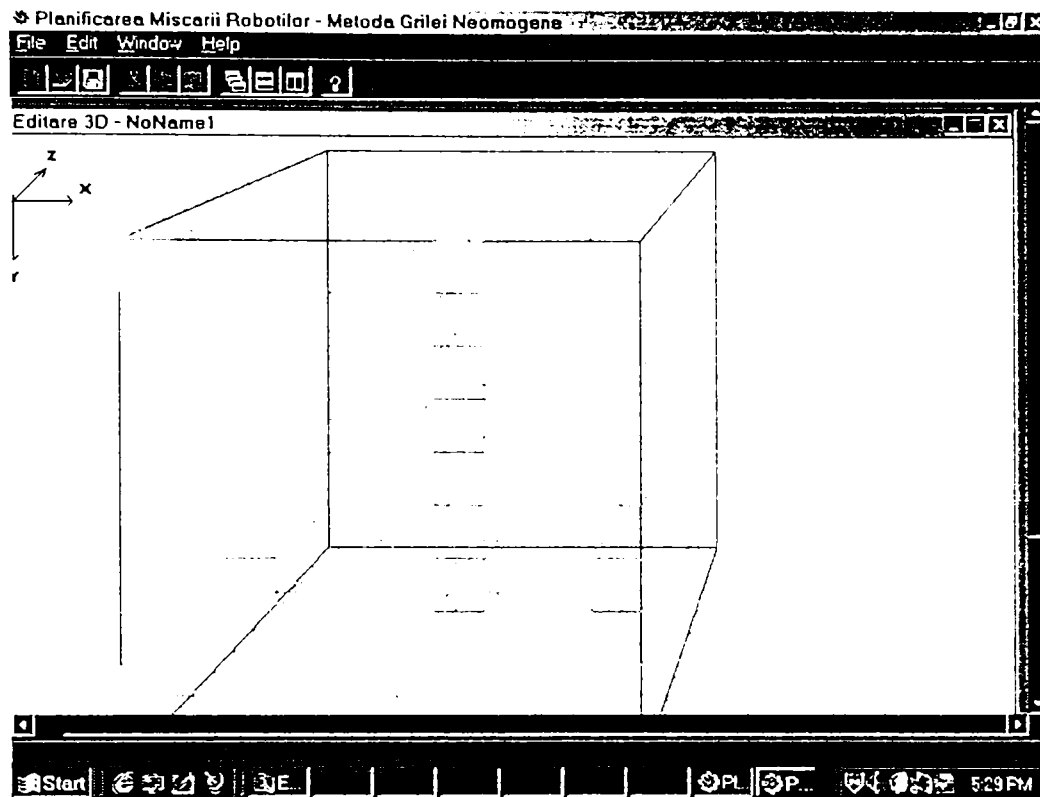


Fig.IV.15. O porțiune din spațiul de lucru modelat în 3D prin metoda grilei neomogene

Editarea obstacolelor se realizează cu mouse-ul și cu tastele, prin reuniunea mai multor cuburi până la realizarea obstacolelor reale, astfel:

- printr-o apăsare pe butonul stâng al mouse-ului se poziționează un nou cub curent la coordonatele 0,0,0 sau se aduce cubul curent la aceste coordonate;
- cubul curent se poate mișca pe axa Ox cu tastele *săgeată stânga* și *săgeată dreapta*;
- cubul curent se poate mișca pe axa Oy cu tastele *săgeată sus* și *săgeată jos*;
- cubul curent se poate mișca pe axa Oz cu tastele *PageUp* și *PageDown*;
- printr-o apăsare pe butonul drept al mouse-ului se validează cubul la coordonatele dorite ca obstacol.

După terminarea editării există posibilitatea salvării spațiului sau a începerii lucrului în acest spațiu. Pentru a aplica metoda descompunerii prin grilă neomogenă a spațiului trebuie apăsat același buton sub formă de bec de pe bara de unelte plasată sub meniul programului, ca în cazul spațiului 2D.

În continuare se urmează pașii metodei de calcul, urmând ca după fiecare pas să se apese butonul de continuare din cadrul ferestrei de control a metodei. În figurile IV.16 și IV.17 sunt prezentate două etape diferite de evidențiere a benzilor libere în spațiul creat.

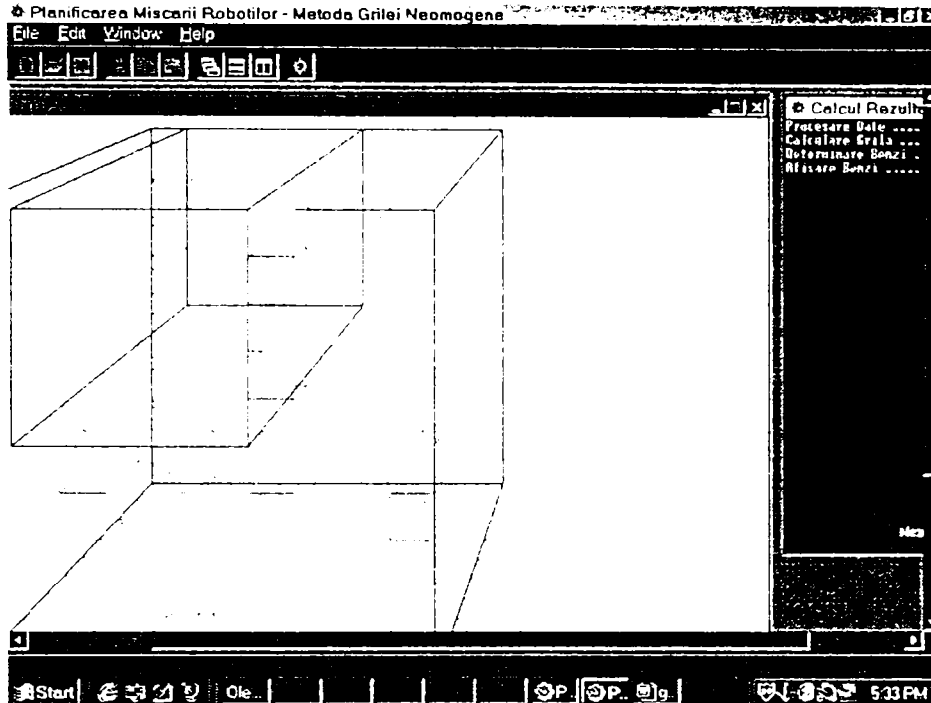


Fig.IV.16. Evidențierea unei benzi libere în partea de sus stânga a spațiului de lucru

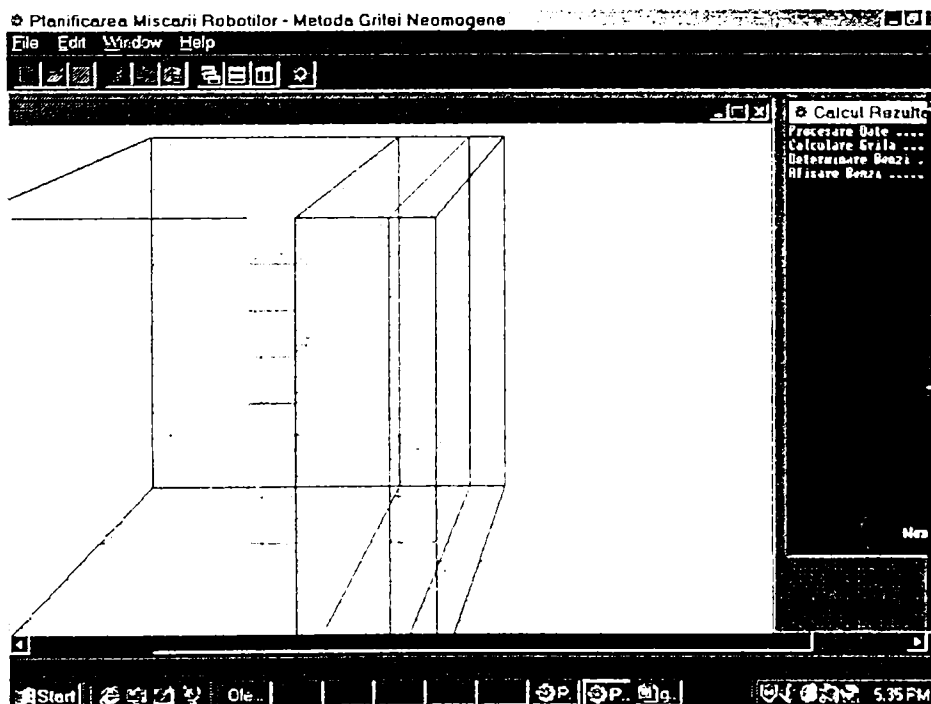


Fig.IV.17. Evidențierea unei benzi libere în partea din dreapta lateral a spațiului de lucru

Aceste benzi libere sunt puse în evidență printr-o structură de liste binare la fel ca cele prezentate în modelarea 2D.

În continuare se solicită introducerea punctului de plecare și a celui de sosire ale robotului, operație care se realizează printr-o simplă fixare a două noi cuburi în spațiul de lucru, în pozițiile dorite (figura IV.18).

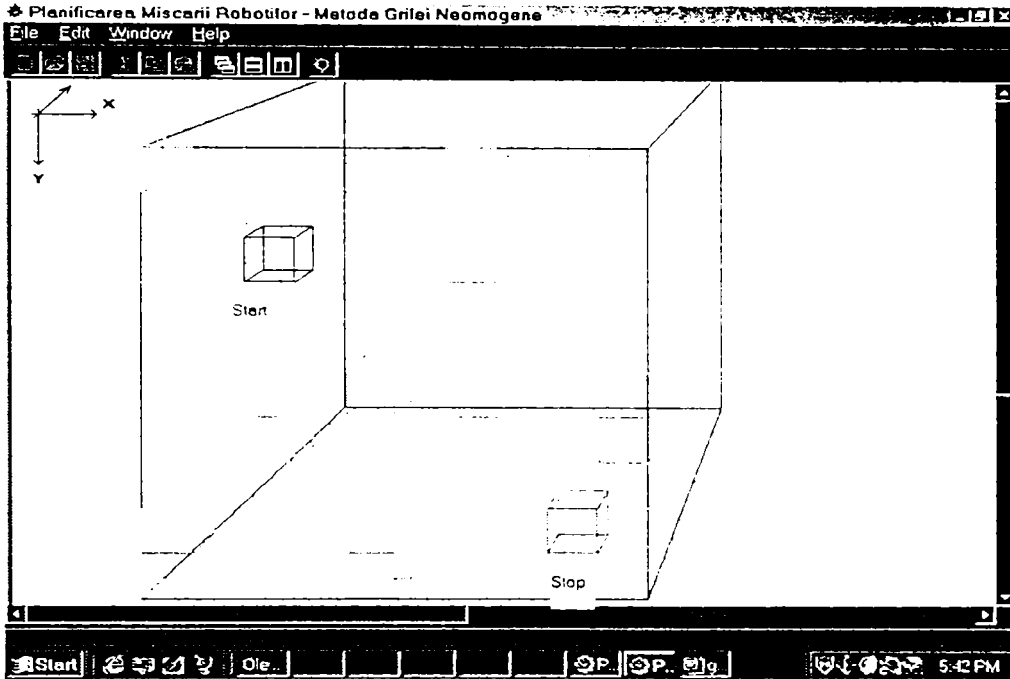


Fig.IV.18. Reprezentarea pozițiilor inițială și finală pentru traiectoria dorită

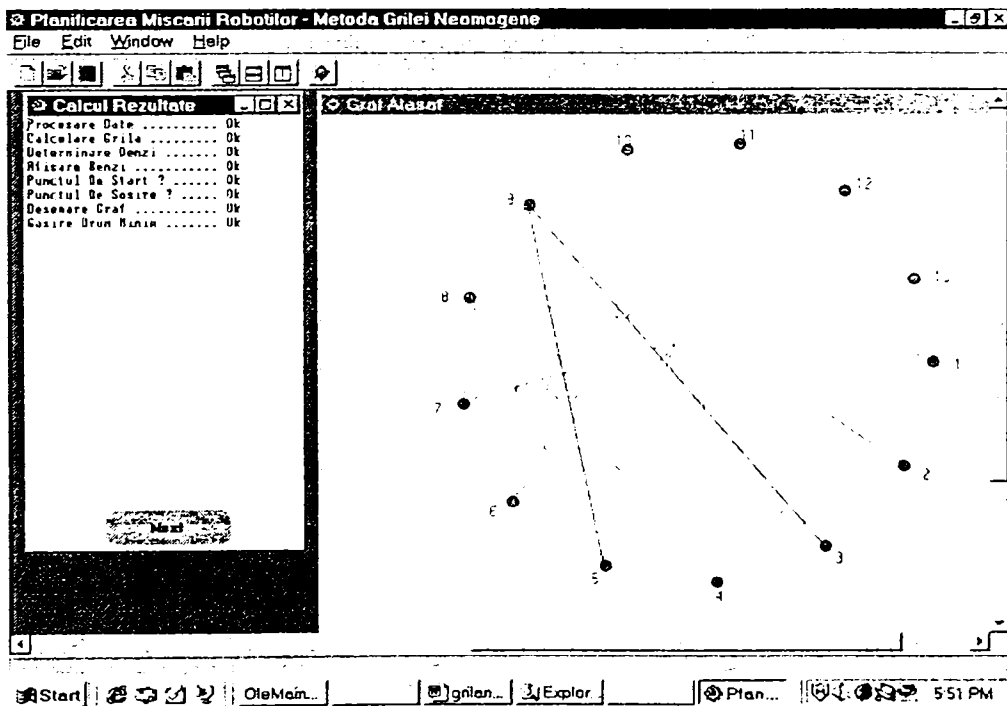


Fig.IV.19. Graful asociat descompunerii spațiului de lucru prin grilă neomogenă în 3D cu indicarea succesiunii de zone libere 3-9-5

După această reprezentare va fi afișat graful atașat descompunerii, cu indicarea celulelor libere prin care poate trece traiectoria unui robot aflat în întregime într-un cub curent (figura IV.19).

La terminarea calculelor se solicită apăsarea pentru ultima dată a butonului de continuare *Next*, care are ca efect închiderea tuturor ferestrelor deschise, inclusiv a spațiului în care s-a lucrat.

Toate metodele de reprezentare prin subdivizare sunt mari consumatoare de memorie, mai ales la spații cu configurații complexe când se urmărește o reprezentare cât mai apropiată de realitate.

De cele mai multe ori însă, pentru aplicații practice, reprezentările foarte exacte nu se justifică, deoarece oricum modelarea inițială este aproximativă și se lucrează cu toleranțe destul de largi pentru a păstra certitudinea non-coliziunii.

V. Concluzii și contribuții originale

Problema reprezentării obiectelor și a combinării acestor reprezentări prin diferite tehnici computerizate a luat practic, locul desenului tehnic, deoarece din ce în ce mai multe ramuri industriale folosesc aceste tehnici de reprezentare în locul metodelor clasice de proiectare și desenare. Este astfel evident că robotica, știință complexă și multidisciplinară, nu poate face abstracție de acest mod de abordare a reprezentării sistemelor complexe de corpuri, care permit dezvoltarea imaginativă a structurii roboților și a spațiilor în care aceștia evoluează.

În domeniul reprezentării spațiale, au fost folosite pe parcursul timpului mai multe tipuri de modele 3D și 2D, mai ales în aplicațiile grafice computerizate. În funcție de structurile de date și de informațiile deținute despre corpuri, modelele prefabricate de sisteme de corpuri solide sunt dificil de utilizat și prezintă limitări atât în ceea ce privește complexitatea reprezentării cât și a acurateții geometriei acesteia. De aceea se impune găsirea unor soluții cât mai elastice și universale de reprezentare a corpurilor indiferent de forma lor și de structura de date pe care reprezentarea respectivă o implică.

Factorii pe care orice reprezentare trebuie să-i aibă în vedere, sunt:

- structura de date impusă, forma algoritmilor de procesare și designul programelor fixe de hardware;
- costul procesării unui obiect și al unui sistem de obiecte 3D;
- gradul de aproximare al imaginii finale a unui obiect ;
- ușurința editării formei obiectului.

Teza de doctorat tratează modelarea sistemelor de corpuri de formă complexă prin metodele reprezentării parametrice, utilizării de primitive geometrice și prin subdivizare spațială, utilizând pentru modelare atât programe originale cât și software profesional. Aceste metode sunt folosite pentru reprezentarea unor corpuri complexe (roboții și spațiile proprii descrise de efectorul lor final), precum și pentru spațiul de

lucru în care un robot mobil evoluează, cu precizarea erorilor impuse pentru reprezentare.

Principalele contribuții originale ale autoarei se pot sintetiza pe capitole după cum urmează:

- ✓ **Capitolul II : Reprezentarea parametrică**, conține :
 - O sinteză bibliografică consistentă a metodelor de reprezentare a curbilor și suprafețelor în spațiul 3D prin utilizarea curbilor Bézier și B-spline;
 - Programe originale de reprezentare a curbilor Bézier și B-spline prin utilizarea punctelor și a muchiilor de control;
 - Programe originale de reprezentare a suprafețelor 3D prin intermediul curbilor parametrice;
 - Calculul modelului geometric direct și evidențierea ecuațiilor cinematice pentru 4 tipuri de roboți cu structură în lanț cinematic deschis, pe baza utilizării convenției Denavit-Hartenberg;
 - Reprezentarea suprafețelor proprii de lucru pentru tipurile de roboți la care s-au calculat ecuațiile cinematice, utilizând facilitățile pachetului de software Matlab; aceste reprezentări sunt utile la planificarea mișcării roboților cu bază fixă într-un spațiu de lucru de geometrie cunoscută.

- ✓ **Capitolul III : Reprezentarea geometrică**, conține:
 - Justificarea reprezentării corpurilor de forme complexe prin utilizarea sferei, ca primitivă unică;
 - Introducerea conceptului de sfericizare la reprezentarea exterioară și interioară a oricărui corp 3D;
 - Construirea unui sistem original și coerent de 16 teoreme, cu corolarele aferente, care să permită reprezentarea prin sfericizare a altor primitive geometrice (cilindru, con, trunchi de con, sector sferic, etc.) prin intermediul cărora se poate realiza construcția oricărui sistem complex de corpuri;

- Evidențierea, la fiecare tip de primitivă reprezentată prin sfere, a erorilor de reprezentare interioară și exterioară, urmărindu-se ca aceste erori să fie minime;
 - O sinteză bibliografică complexă a abordării sfericizării din punct de vedere topologic, cu prezentarea unor algoritmi utilizați la crearea de programe de reprezentare;
 - Realizarea de programe originale de sfericizare pentru primitivele geometrice menționate și reconstrucția, prin utilizarea acestor programe, a unor modele de roboți din Laboratorul de Robotică al Catedrei de Organe de Mașini și Mecanisme al facultății de Mecanică.
- ✓ **Capitolul IV : Reprezentarea prin subdivizare spațială**, conține:
- Abordarea problemei de bază a planificării roboților mobili într-un spațiu de lucru, prin prisma posibilității de modelare prin tehnica de subdivizare spațială și pe baza unor algoritmi de navigare din literatură;
 - Prezentarea a două metode de planificare compatibile cu tehnica de reprezentare aleasă pentru spațiul Laboratorului de Robotică în care evoluează un robocar automobil;
 - Realizarea unor programe de modelare 2D și 3D a spațiului Laboratorului de Robotică prin metoda grilei neomogene;
 - Realizarea unor programe de modelare 2D a spațiului Laboratorului de Robotică prin metoda grilei omogene.

Această prezentare succintă a contribuțiilor autoarei în domeniul abordat relevă faptul că prezenta teză reprezintă o etapă dintr-un domeniu vast, ce constă de mai mulți ani preocuparea colectivului de cercetare aflat sub conducerea dnei. prof. dr.ing. Doina Drăgulescu. Cercetările au început cu modelarea analitică a roboților (cinematică, diferențială și dinamică), continuând cu probleme complexe de planificarea mișcării roboților mobili, de interacțiune a roboților cu mediul prin vedere artificială și dezvoltându-se cu abordări de modelări din ce în ce mai complexe de roboți mobili printre obstacole mobile.

Etapa care se finalizează prin această teză are rolul de a oferi suportul matematic riguros problemelor de modelare a structurilor complexe, prezentând modalități de rezolvare aproape universal valabile pentru orice structură.

Teza deschide drumul unor studii de planificare a mișcării în laboratoare virtuale, pentru roboți virtuali, neexistând decât piedica propriei imaginații a utilizatorului.

Bibliografie

1. **A.Davidoviciu, G.H.Drăgănoiu, A.Moangă**: Modelarea, simularea și comanda manipuloarelor și roboților industriali, Editura Tehnică, București, 1986
2. **A.Dumitriu**: Tendințe și perspective în robotică, Vol. Simpozionul Național de Roboți Industriali, Ediția a XII-a, Timișoara, Vol.1, 1994
3. **A.C.Nearchou, N.A. Aspragathos**: Application of genetic algorithms for point to point motion of redundant manipulators, Mechanism and Machine Theory, Vol.31, No.3, 1996
4. **A.Davidoviciu, A.Moangă**: Roboți evoluți și automatizări flexibile, Seria: Tehnica la zi, Editura Tehnică, București, 1985
5. **A.Glasner** : An introduction to ray tracing. Academic Press, London , 1989
6. **A.Hatvany**: On conjugate curve schemes for quadratic and cubic curves, Proc. International Conference on Technical Informatics, Timisoara, Vol.2, 1994
7. **A.Lambert, N.Le Fort-Piat** : Robot tasks planning integrating uncertainties and local maps. AVCS'98, Amiens, France, 1998
8. **A.M.Arbib, A.Hanson**: Vision, brain and cooperative computation. MIT Press, Cambridge, MA, 1987
9. **A.M.Barry** : Visual intelligence. State University of New York Press, Ph.D Thesis,1997
10. **A.Pruski**: Robotique générale, Editions Ellipses, Paris, 1989
11. **A.Pugh** : Robot vision. Editions IFS Publications, Springle, New York, NY, 1983
12. **A.R.Weeks** : Fundamentals of electronic image processing. SPIE-International Society for Optical Engineering, 1996

13. **B. Bouilly, T.Simeon, R.Alami** : A numerical technique for planning motion strategies of a mobile robot in presence of uncertainty. IEEE International Conference on Robotics and Automation, 1995
14. **B. Epureanu, V.Tăbăcaru**: Algoritm pentru minimizarea timpului de mișcare a roboților manipulatori, Vol. Simpozionul Național de Roboți industriali, Timișoara, Ediția a XII-a, Vol.2, 1994
15. **B. Jahne** : Digital image processing. Springer-Verlag New York, 1997
16. **B.Rendi, O.Benghia**: The calculus of some special groups of coomology, Bulletins for Applied Mathematics BAM, Budapest, 994/94, LXXII, 1994
17. **B.Rendi, O.Benghia**: Numerical methods for linear differential equations with polinomial coefficients, Bulletins for Applied Mathematics BAM, Budapest, 993/94, LXXII, 1994
18. **C. Moler, J.Little, S.Bangert**: PRO-MATLAB user's guide. Math Works, Sherborn MA, 1987
19. **C.Canudas, H.Khennouf, C.Samson, O.J. Sordalen** : Nonlinear control design for mobile robots. World Scientific Series in Robotics and Automated Systems, Vol. 11, 1993
20. **C.I.Connolly, J.B.Burns, R.Weis** : Path planning using Laplace's equations. Proc. IEEE International Conference on Robotics and Automation, Cincinnati, Ohio, 1990
21. **C.J.Li, T.S.Sankar**: Fast inverse dynamics computation in real-time robot control, Mechanism and Machine Theory, Vol.27, No.6, 1992
22. **C.Rădulescu**: Oscilațiile punctului caracteristic la comanda punct cu punct a roboților industriali, Analele Universității din Oradea, Fascicola Mecanica, Oradea, 1994
23. **C.S.Lee, M.J.Chung**: An adaptive control strategy for mathematical manipulators, IEEE Transactions on Automation and Control, Vol.AC-29, No.9, 1984
24. **C.S.Lin, P.R.Chang, J. Y.Luh**: Formulation and optimization of cubic polynomial joint trajectories for industrial robots, IEEE Transactions on Automation and Control, Vol.AC-28, No.12, 1983.
25. **C.Strothotte, T.Strothotte** : Seeing between the pixels. Springer-Verlag New York, 1997

26. **C.Vibet**: Les robots. Principes et contrôle, Editions Marketing, Paris, 1987
27. **C.Watkins, R.Stevens** : Advanced graphics programming in C and C++. M&T Publishing, San Mateo, CA, 1991
28. **C.Z.Grudci, P.D. Lawrence** Iterative inverse kinematics with manipulator configuration control, IEEE Transactions on Robotics and Automation, Vol.9, No.4, 1993
29. **D. Larionescu**: Metode numerice, Editura Tehnică, București, 1989
30. **D.Manocha, J.F.Canny**: Efficient inverse kinematics for general 6R manipulators, IEEE Transaction on Robotics and Automation, Vol.10, No.5, 1994
31. **D.Andreiciuc**: Optimizarea conducerii vehiculelor ghidate automat, Teză de doctorat, Universitatea Politehnica Timișoara, 1999
32. **D.B.Reister, F.Pin**: Time-optimal trajectories for mobile robots with two independently driven wheels, International Journal of Robotics Research, Vol.13, No.1, 1994
33. **D.B.Reister, S.M.Lenhart**: Time-optimal paths for high-speed maneuvering, International Journal of Robotics Research, Vol.14, No.2, 1995
34. **D.Drăgulescu, F.Moldovan, M.Toth-Tașcău**: Soluționarea problemei cinematice inverse pentru un robot având șase grade de libertate, prin metoda separării matricelor, Analele Universității din Oradea, Fascicola Mecanica, 1995
35. **D.Drăgulescu, M.Toth-Tașcău**: Elemente de inginerie mecanică. Introducere în cinematica roboților. Introducere în dinamica roboților, Lito UPT, Timisoara, 1993
36. **D.Drăgulescu, M.Toth-Tașcău, C. Neagoe, V. Stănciulescu**: Metodă de identificare curentă a planelor autoascunse, Vol. Al XI-lea Simpozion Național de Roboți Industriali, Timișoara, 1992
37. **D.Drăgulescu, M.Toth-Tașcău, Serafin P**: Analiza spațiului de lucru al unui robot RTRTTR prin metoda modelării statistice, Vol. Al XI-lea Simpozion Național de Roboți Industriali, Timișoara, 1992
38. **D.Drăgulescu** : Dinamica Roboților. Ed. Didactică și Pedagogică, București, 1997
39. **D.Drăgulescu, C.Couturier** : Cours de modélisation des robots. Lito. Université d'Artois, Franța, 1995

40. **D.Drăgulescu, F. Moldovan:** Legile de mișcare ale unui robot RRT cu momente de inerție variabile. Vol.A VIII-a Conferință de Vibrații Mecanice, Timișoara, 1996
41. **D.Drăgulescu, F. Moldovan, H.Moldovan:** Metoda de determinare a marginilor unui C-obstacol. Vol. Al XII-lea Simpozion Național de Roboți Ind., Timișoara, 1994
42. **D.Drăgulescu, F.Moldovan, H.Moldovan:** Metoda descompunerii celulare exacte. Curbe critice. Vol. Al XII-lea Simpozion Național de Roboți Ind., Timișoara, 1994
43. **D.Drăgulescu, F.Moldovan, H.Moldovan:** Considerations about the critical curves at the exact cell decomposition method. International Conference on Technical Informatics, Conti'94, Timișoara, 1994
44. **D.Drăgulescu, F.Moldovan, M.Toth-Tașcău:** Metode de planificare a traiectoriilor la roboți. Sinteză. Analele Universității Eftimie Murgu, Reșița, 1995
45. **D.Drăgulescu, F.Moldovan, M.Toth-Tașcău:** Considerations about the critical curves at the exact cell decomposition method. Buletin Șt. Și Tehnic al UTT, seria Mecanică, Tom 40 (54), 1995
46. **D.Drăgulescu, F.Moldovan:** Trajectory control of RRT robot with timevarying inertia links. International Conference on Technical Informatics, Conti'96, Timișoara, 1996
47. **D.Drăgulescu, D.Iosif:** Using shape numbers for image description and recognition. Buletinul Șt. și Tehnic al UTT, seria Automatică și calculatoare, Tom 42(56), 1997
48. **D.Drăgulescu, F.Moldovan, M.Toth-Tașcău, C.Crivacucea** Metode de planificare a traiectoriilor la roboți. Sinteză. Analele universității Eftimie Murgu, Reșița, anul III, 1996
49. **D.Drăgulescu, M.Toth-Tașcău :** Determinarea ecuațiilor de mișcare pe baza modelului dinamic pentru un manipulator cu trei grade de libertate. Proc. The 25-th Conference on Production Engineering, Beograd, Jugoslavia, 1994
50. **D.Drăgulescu, M.Toth-Tașcău :** Comparative dynamic study of a serial robot. Mécanique-Industrie-Matériaux, vol. 6, Franța, 1996

51. **D.Drăgulescu, M.Toth-Tașcău** : Comparative dynamic study of a serial robot. Proc.The 26-th Conference on Mechanical Engineering, Technion City, Israel, 1996
52. **D.Drăgulescu, M.Toth-Tașcău** : Contributions to cell decomposition method for a two-dimensional work space. Proc. International Conference on Technical Informatics, Conti'96, Timișoara, 1996
53. **D.Drăgulescu, M.Toth-Tașcău**: Contributions to motion planning using potential field method. Proc. International Conference on Technical Informatics, Conti'96, Timișoara, 1996
54. **D.Drăgulescu, M.Toth-Tașcău**: Coordonarea cuplelor cinematice ale unui robot prin metoda timpului minimal. Vol. Al XII-lea Simpozion Național de Robotică, Reșița, 1996
55. **D.Drăgulescu, M.Toth-Tașcău**: Determinarea preciziei de parcurgere a traiectoriei unui robot. Vol.A XXVII-a Ses. de comunicări științifice cu part. intern., Ac. Tehnică Militară, București, 1997
56. **D.Drăgulescu, M.Toth-Tașcău**: Dynamic comparative study of a welding robot CLOOS. Proc. Third European Conference on Joining Technology EUROJOIN3, Berna, Elveția, 1998
57. **D.Drăgulescu, M.Toth-Tașcău**: Dynamic study of a welding robot. Proc.The 25-th Conference on Production Engineering, Beograd, Jugoslavia, 1994
58. **D.Drăgulescu, M.Toth-Tașcău**: Interpolarea traiectoriilor unui robot cu funcții spline și restricții cinematice și dinamice. Proc.The VII-th Symposium of Mathematics and its Applic., univ. Politehnica timișoara, 1997
59. **D.Drăgulescu, M.Toth-Tașcău**: Kinematic and dynamic study of a welding robot. Second European Conference on Joining Technology EUROJOIN2, Florența, Italia, 1994
60. **D.Drăgulescu, M.Toth-Tașcău**: Method to generate an imposed trajectory for a robot having 6 degrees of freedom. Proc. International Conference on Technical Informatics, Conti'94, Timișoara, 1994
61. **D.Drăgulescu, M.Toth-Tașcău**: Method to study the dynamical behaviour of a complex mechanical structure. Proc. International Conference, University of the West, Timișoara, 1997

62. **D.Drăgulescu, M.Toth-Tașcău:** Modelarea roboților cu structura arborescentă și lanț cinematic închis. Proc. The VII-th International Conference of Manufacturing Engineering Techno'95, Timișoara, 1995
63. **D.Drăgulescu, M.Toth-Tașcău:** Modelarea spațiului de lucru prin metoda arborelui. Workspace modelling by tree method. Analele Universității din oradea, Fascicola Mecanică, 1997
64. **D.Drăgulescu, M.Toth-Tașcău:** Motion planning for a movable robot by potential field method. Buletinul Șt. Și Tehnic al UTT, seria Mecanică, Tom 41(55), 1996
65. **D.Drăgulescu, M.Toth-Tașcău:** Path planning for a robot by exact cell decomposition method. Proc. The II-nd International Conference on Advanced Robotics, Viena, 1996
66. **D.Drăgulescu, M.Toth-Tașcău:** Position and orientation errors due to robot element deformations. Proc.The 30-th International Symposium on Automotive Technology and Automation ISATA'97, Florența, Italia, 1997
67. **D.Drăgulescu, M.Toth-Tașcău:** Workspace modelling by non-homogeneous grid and tree method. Buletinul Șt. Și Tehnic al UTT, seria Mecanică, Tom 42(56). 1997
68. **D.Drăgulescu, M.Toth-Tașcău, D.Iosif:** Dynamical study of the welding robot CLOOS with rigid links. Buletinul Șt. și Tehnic al UTT, seria Mecanică, Tom 42(56), 1997
69. **D.Drăgulescu, M.Toth-Tașcău, F.Moldovan:** Metodă și algoritmi de planificare a traiectoriilor plane. Vol. A II sesiune de comunicări științifice cu participare internațională "Realizări tehnice și cultural-științifice pe meleaguri arădene", Arad, 1994
70. **D.Drăgulescu, M.Toth-Tașcău, F.Moldovan:** Planificarea mișcării roboților, Ed. Helicon, Timișoara, 1995
71. **D.Drăgulescu, M.Toth-Tașcău, G.Pacso:** Path planning by the exactcell decomposition method. Buletinul Șt. Și Tehnic al UTT, seria Mecanică, Tom 41(55), 1996

72. **D. Drăgulescu, O. Căuș, H. Moldovan:** Geometrical considerations regarding the representation of objects with cylindrical form by sphere covering, în curs Proc. Conti'2000, Timișoara
73. **D. Drăgulescu, O. Căuș, H. Moldovan:** Program to represent cylindrical body systems using spheres, în curs Proc. Conti'2000, Timișoara
74. **D. Iosif:** Sinteza inteligenței artificiale pe baza interacțiunii prin vedere a robotului cu mediul, Teză de doctorat, Universitatea Politehnica Timișoara, 1999
75. **D.E.Koditschek:** Robot planning and control functions, The Robotics Review 1, MIT Press, 1989
76. **D.M. Himmelblau:** Applied nonlinear programming, McGraw-Hill, New-York, 1972.
77. **D.P. Dobkin, D.G. Kirkpatrick:** A linear algorithm for determining the separation of convex polyhedra, Journal of Algorithmus, No.6, 1985
78. **D. Rendi, O. Benghia, B. Rendi:** Despre conexitatea relativizată generală, Proc. Scientific communications meeting of A. Vlaicu University, Arad, vol.14B, 1996
79. **D. Rendi, O. Căuș:** Properties of relative (n,k) - connected subsets, Bulletins for Applied Mathematics BAM, Budapest, LXXXVI-A/98, 1998
80. **D. Rendi, O. Căuș:** About relative (n,k) connectedness, Bulletins for Applied Mathematics BAM-1487/98, Budapest, LXXXV, 1998
81. **D.Surdilovici, M.Vukobratovici:** One method for efficient dynamic modeling of flexible manipulators, Mechanism and Machine Theory, Vol.31, No.3, 1996
82. **D.W. Wloka :** Robotersysteme, Springer-Verlag, Berlin, 1992
83. **E. Dombre, W. Khalil :** Modélisation et commande des robots, Edition Hermés, Paris, 1988
84. **F. Pfeiffer, R.Johanni:** A concept for manipulator trajectory planning, Proc. IEEE Conference on Robotics and Automation (San Francisco), 1986
85. **F.Kovacs, C.Rădulescu:** Roboți industriali, Lito UTT, Timișoara, 1992
86. **F.Solc :** Robot modelling and control, Proc.Internationaler Sommerkurs, Technische Universität Wien, 1993
87. **G. Cojocaru, F. Kovacs:** Roboții în acțiune. Sistemele flexibile și fabricația de serie, Ed. Facla, Timișoara, 1985

88. **G.Feng**: A new adaptive control algorithm for robot manipulators in task space, IEEE Transactions on Robotics and Automation, Vol.11, No.3, 1995
89. **G.S. Chirikjian, J. Burdick**: 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, 1990
90. **G.X.Ritter, J.N.Wilson** : Handbook of computer vision algorithms in image algebra. CRC Press Inc. , New York, 1996
91. **Gh. Micula**: Funcții spline și aplicații, Editura Tehnică, 1978
92. **Gh.Silaș, I.Groșanu**: Mecanica, Editura Didactică și Pedagogică, București, 1981
93. **G.Gh.Savii**: Bazele proiectării asistate de calculator, Editura Mirton, Timișoara, 1997
94. **H. Asada, J. Slotine**: Robot analysis and control, New-York: Wiley & Sons, 1986
95. **H. Bruyninckx, S. Demey, S. Dutre, J. De Schutter**: Kinematic model for model-based compliant motion in the presence of uncertainty, International Journal of Robotics Research, Vol.14, No.5, 1995
96. **H. Diker** : Energy efficient sinusoidal path planning of robot manipulators, Mechanism and Machine Theory, Vol.29, No.6, 1994
97. **H.H.Ahlberg** : The theory of splines and their applications, New-York: Academic Press, 1967
98. **H.Li, M.A.Lavin, R.J.Le Master** : Fast Hough transform: A hierarchical approach. computational vision, Graphics Image Proc, vol. 36, 1986
99. **H.Ozaki, T.Shimadzu, A.Mohri**: Collision-free path generation for a mobile robot by an artificial transformation of obstacle space, Robotica, Vol.7, 1989
100. **H.Stark, Y.Yang** : Vector space projections. John Wiley & Sons, New York, 1998
101. **I. Bogdanov**: Problema conducerii unui robot industrial, Vol. Simpozionul Național de Roboți Industriali, Timișoara, Ediția a XII-a, Vol.3 , 1994
102. **I.J.Schoenberg**: Privești matematice, Centrul de Cercetări Matematice, Universitatea Wisconsin-Madison, Traducere-Editura Tehnică, București, 1989
103. **I.L.Aken, H.Brussel**: On-line robot trajectory control in joint coordinates by means of imposed acceleration profiles, Robotica, Vol.6, 1988.

104. **LNegrean, P.Popescu, N.Haiduc, LVuscan**: Influența erorilor geometrice ale parametrilor de tip D-H asupra preciziei cinematice a roboților industriali, Vol. Simpozion Național de Roboți Industriali, Ediția a XII-a, Timișoara, Vol.2, 1994
105. **I.Văcărescu, V.Văcărescu, E.Lovasz**: Optimizarea traiectoriei și a legilor de mișcare ale punctului caracteristic, Vol.Simpozion Național de Roboți Industriali, Ediția a XII-a, Timișoara, Vol.3, 1994
106. **J. Barraquand, B. Langlois, J.C.Latombe**: Robot motion planning with many degrees of freedom and dynamic constraints, Proc. The 5-th International Symposium on Robotics Research, Tokyo, 1989
107. **J. Barraquand, J.C. Latombe**: A Monte-Carlo algorithm for path-planning with many degrees of freedom, Proceedings IEEE International Conference on Robotics and Automation, Cincinnati, Ohio, 1990
108. **J.Albus** : Brain's behaviour and robotics, BYTE Publication Inc., 1981
109. **J.Borenstein, V.Koren**: The vector field histogram-fast obstacle avoidance for mobile robots, IEEE-Transactions on Robotics and Automation, Vol.7, No.3, 1991
110. **J. Hadamard**: Lecții de geometrie elementară; geometrie în spațiu, Editura tehnică, București, 1961
111. **J.C.Lamadrid**: Avoidance of obstacles with unknown trajectories: Locally optimal path and periodic sensor readings, International Journal of Robotics Research, Vol.13, No.6, 1994
112. **J.C.Latombe**: Robot motion planning. Kluwer Academic Publishers, Norwell-Massachusetts, USA, 1991
113. **J.G.Postaire** : De l'image à la decision. Editions Dunod-Informatique, Bordas, Paris, 1987
114. **J.Imura, T.Sugie, T.Yoshikawa**: Adaptive robust control of robot manipulators- Theory and Experiment, IEEE Transactions on Robotics and Automation, Vol.10, No.5, 1994
115. **J.J.Craig** : Introduction to robotics. Mechanics and control, 2nd edition, Addison Wesley, 1989.
116. **J.L.Starck, F.Murtagh, A.Bijaoui**: Image processing and data analysis. Cambridge University Press, 1998

117. **J.Lin, F.L.Lewis**: A symbolic formulation of dynamic equations for a manipulator with rigid and flexible links, *International Journal of Robotics Research*, Vol.13, No.5, 1994.
118. **J.Liu**: Spatial reasoning about robot compliant movements and optimal path in qualitatively modeled environments, *International Journal of Robotics Research*, Vol.15, No.2, 1996.
119. **J.P.Merlet**: Trajectory verification in the workspace for parallel manipulators, *International Journal of Robotics Research*, Vol.13, No.4, 1994
120. **J.R.Parker**: Algorithms for image processing and computer vision. John Wiley & Sons, New York, 1996
121. **J.R.Parker**: Practical computer vision using C. John Wiley & Sons, New York, 1993
122. **J.R.Singh, J.Rastegar**: Optimal synthesis of robot manipulators based on global kinematic parameters, *Mechanism and Machine Theory*, Vol.30, No.4, 1995
123. **J.Y.Luh, M.W.Walker**: Minimum-time control along the path for a mechanical arm, *Proc. CDC*, TP-5, 1977
124. **J.Y.Luh, M.W.Walker, R.Paul**: On-line computational scheme for mechanical manipulators, *ASME Transactions on Journal of Dynamic Systems, Measurement and Control*, Vol.102, 1980
125. **K.C.Gupta, V.K.Singh**: A numerical algorithm for solving robot inverse kinematics, *Robotica*, Vol.7, 1989
126. **K.G.Shin, N.D.McKay**: Open-loop minimum-time control of mechanical manipulators and its application, *Proc. of American Control Conference*, San Diego, California, 1984
127. **K.Kondo** : Motion planning with six degrees of freedom by multistrategic bidirectional heuristic free-space enumeration, *IEEE Transactions on Robotics and Automation*, Vol.7, No.3, 1991
128. **K.L.Ting, N.L.Lee, G.H.Braudan**: Synthesis of polynomial and other curves with the Bézier technique, *Mechanism and Machine Theory*, Vol.29, No.6, 1994
129. **K.S.Fu, R.C.Gonzalez, C.S.G.Lee**: Robotics-Control, sensing, vision, and intelligence. Academic Press, Boston, 1990

130. **L.Marcé, M.Jullière, H.Place**: Stratégie de contournement d'obstacles pour un robot mobile, R.A.I.R.O. Automatique/Systems Analysis and Control, Vol.15, No.1, 1981
131. **L.Wang**: A decomposition theorem of motion, International Journal of Engineering Science, Vol.34, No.4, 1996
132. **M. Baghat, S. El-Shakery**: A compact kinematic synthesis procedure for robot's manipulator arms, Mechanisms and Machine Theory, Vol.28, No.3, 1993
133. **M. Brady**: Robot motion, planning and control, The MIT Press, Berlin, 1982
134. **M. Cai, A. Rovetta**: A new approach of joint trajectories for cartesian path approximation of mechanical manipulators, Meccanica, No.24, 1989
135. **M.Aldon**: Elaboration automatique du modèle dynamique des robots, RAIRO Automatique/Systems, Analysis and Control, Vol.16, No.3, 1982.
136. **M.Awad, J.Kuusela** : Object-oriented technology for real time systems. Prentice Hall, Cambridge, 1996
137. **M.Ghinea, V.Firețeanu** : MATLAB- Calcul numeric și grafică - Aplicații, Editura Teora, București, 1995
138. **M.Kircianski, M.Vukobratovici**: Contribution to control of redundant robotic manipulators in an environment with obstacles, International Journal of Robotics Research, Vol.5, No.4, 1986
139. **M.Papageorgoiu, Th.Bauschert**: Stochastic optimal control of moving vehicles in a dynamic environment, International Journal of Robotics Research, Vol.13, No.4, 1994
140. **M.Parent, C.Laurgeau**: Les Robots. Langages et méthodes de programmation, Hermès Publishing, Paris, 1983
141. **M.Toth-Tașcău, D.Drăgulescu**: Path planning for a car-like robot by potential field method. Proc. The 29-th International Symposium on Automotive Technology and Automation ISATA'96, Florența, Italia, 1996
142. **M.Toth-Tașcău, D.Drăgulescu**: Two approaches of potential field method in two and three-dimensional workspace for robot path planning, Robotica &Management, vol.3,nr.1,1998

143. **M.Toth-Tașcău, D.Drăgulescu** Precizia cinematică a parcurgerii traiectoriei unui robot. Acta Universitatis Cibiniensis, Vol XiX, Universitatea L. Blaga, Sibiu, 1995
144. **M.Vlada, A.Posea, I.Nistor, C.Constantinescu**: Grafică pe calculator în limbajele Pascal și C. Implementare și aplicații, Editura Tehnică, București, 1992
145. **M.Vukobratoviæ, R.Stojci**: Historical perspective of hibrid control in robotics: beginnings, evolution, criticism and trends, Mechanism and Machine Theory, Vol.30, No.4, 1995
146. **M.Vukobratovici, D.Stokici**: Control of manipulation robots, Berlin, Heidelberg, New York, Springer-Verlag, 1982
147. **N. Lungu N**: Matematici cu aplicații tehnice, Editura Tehnica, Bucuresti, 1991
148. **N.D.Kadianakis**: The kinematics of continua and the concept of connection on classical space-time, International Journal of Engineering Science, Vol.34, No.3, 1996
149. **O. Benghia**: Bounded Gets in vector spaces, Studia universitatis V. Goldiș, Arad, seria A, no.6, 1996
150. **O. Benghia, D. Deac**: Submulțimi relativ compacte, Studia universitatis V. Goldiș, Arad, seria A, no.6, 1996
151. **O. Benghia, D. Deac**: Despre compactitatea relativizată, Proc. Scientific communications meeting of A. Vlaicu University, vol.14 B, Arad, 1996
152. **P.Couffin, J.Herbin , M.Puzenat M**: C.A.O. en mécanique, Edition Armand Colin, 1987.
153. **P.E.Mattinson** : Practical digital video with programming examples in C. John Wiley & Sons, New York,1994
154. **P.Soueres, T.Hamel, V.Cadenat, A.Dzual** : A reactive path following controller for mobile robots. Proc. AVCS'98, Amiens, France, 1998
155. **P.Tournassoud**: Géometrie et intelligence artificielle pour les robots, Editions Hermès, Paris, 1982
156. **P.Tournassoud**: Planification et contrôle en robotique. Application aux robots mobiles et manipulateurs, Editions Hermès, Paris, 1992

157. **Ph. Coiffet**: Robotique. Principes et applications, Editions Hermès, Paris, 1986, 1992
158. **Ph. Coiffet**: Les Robots. Modélisation et commande, Hermes Publishing, Paris, 1987
159. **Ph.Coiffet** : Les Robots - Interaction avec l'environnement. Ed.Dunod, Paris, 1991
160. **R.Demeter, M.Jurca, W.Konnerth**: Metodă de descompunere celulară aproximativă a spațiului de lucru, proiect, Universitatea politehnica TimișoaraFac. de Automatizări și calculatoare, Timișoara, 1996
161. **R.G.Roberts, A.Maciejewski**: Singularities, stable surfaces and the repeatable behavior of kinematically redundant manipulators, International Journal of Robotics Research, Vol.13, No.1, 1994
162. **R.H. Bartels, J. Beaty, B. Barski**: Mathématique et C.A.O. Bêta-splines, Edition Hermès, Paris, 1988
163. **R.J.Schalkoff**: Artificial intelligence. An engineering approach, Mc.Graw-Hill Publishing Company, New-York, 1990
164. **R.J.Theodore, A.Ghosal**: Comparison of the assumed modes and finite element models for flexible multilink manipulators, International Journal of Robotics Research, Vol.14, No.2, 1995
165. **R.M.Haralic, L.G.Shapiro** : Computer and robot vision. Addison Wesley Logman, 1992
166. **R.M.Santis**: Path-tracking for a tractor-trailer-like robot, International Journal of Robotics Research, Vol.13, No.6, 1994
167. **R.Paul**: Robot manipulators: Mathematics, programming and control, Cambridge, The MIT Press, 1981, 1983, 1994
168. **R.Voinea, D.Voiculescu, V.Ceaușu**: Mecanica, Editura Didactică și Pedagogică, București, 1983
169. **S. Fleury, Ph. Souères, J-P.Laudmond, R.Chatila**: Primitives for smoothing mobile robot trajectories, IEEE Transactions on Robotics and Automation, Vol.11, No.3, 1995
170. **S. Hesse** : Industrie-roboter-peripherie. Verlag Technik, Berlin, 1989

171. **S. Mitsi, K.Bouzakis, G.Mansour**: Optimisation of robot links motion in inverse kinematics solution considering collision avoidance and joint limits, *Mechanism and Machine Theory*, Vol.30, No.5, 1995
172. **S.J.Lorenc, M.Stanisici**: Application of instantaneous invariants to the path tracking control problem of planar two degree-of-freedom systems: a singularity-free mapping of trajectory geometry, *Mechanism and Machine Theory*, Vol.30. No.6, 1995
173. **S.Jun, K.G.Shin**: Shortest path planning in discretized workspaces using dominance relation, *IEEE Transactions on Robotics and Automation*, Vol.7, No.3, 1991
174. **S.Sanjeevv, J.T.Wien**: A global approach to path planning for redundant manipulators, *IEEE Transactions on Robotics and Automation*, Vol.11, No.1, 1995
175. **T. Binford**: Discussion of trajectory calculation methods, Exploratory study of computer integrated assembly systems, Stanford Artificial Intelligence Laboratory, Progress Report, Memo AIM-285.4, Stanford, 1977
176. **T.Precup, V.Handra-Luca, D.Precup, C.Sipos**: Robot simulation using object-oriented programming, *Proc.International Conference on Technical Informatics*, Timisoara, Vol.2, 1994.
177. **T.Szkodny**: Forward and inverse kinematics of IRb-6 manipulator, *Mechanism and Machine Theory*, Vol.30, No.7, 1995
178. **T.Uhl, M.Szymkat**: Comparison of the classical and neural-based approach to control of manipulation robots, *Proc. The 17-th International Seminar on Modal Analysis-KU Leuven*, Belgium, 1993
179. **Th.Boiangiu, LDumitrache** *Intelligent manufacturing systems 1995*. Elsevier Science Press, 1997
180. **Tz.Y.Young, King-Sun Fu**: *Handbook of pattern recognition and image processing*. Academic Press, San Diego, 1986
181. **V. Barbu**: *Metode matematice în optimizarea sistemelor diferențiale*, Editura Academiei RSR, București, 1989
182. **V.Handra-Luca, V.Matieș, C.Brisan, T.Tiuca**: *Roboți: Structură, cinematică și caracteristici*, Editura Dacia, Cluj- Napoca, 1996

183. **V.Ispas**: Aplicațiile cinematicii în construcția manipuletoarelor și a roboților industriali, Editura Academiei Române, București, 1990
184. **V.Ispas, I.Pop, M.Bocu** : Roboți industriali, Editura Dacia, Cluj-Napoca, 1985.
185. **V.Lakshmikantham, S.Leela**: Differential and integral inequalities, New York: Academic Press, New York, 1969
186. **V.Lumelsky**: A comparative study on the path length performance of maze-searching and robot motion planning algorithms, IEEE Transactions on Robotics and Automation, Vol.7, No.1, 1991
187. **V.R.Dragomir, A.Valea, C.Bratu** : Proiect – Metoda grilei neomogene și metoda grafului vizibilității. Proiect, Facultatea de Automatică și Calculatoare, Universitatea Politehnica Timisoara, 1995
188. **V.Văcărescu, E.Lovasz**: Minimizarea timpului de deplasare a punctului caracteristic pe traiectorie, Analele Universității din Oradea, Fascicola Mecanica, Vol.II, Oradea, 1995
189. **V.Văcărescu, E.Lovasz**: Traectorii ale punctului caracteristic de accelerație minimă, Analele Universității din Oradea, Fascicola Mecanică, Vol.II, Oradea, 1995
190. **W. Armstrong**: A recursive solution to the equation of motion of an N-link manipulator, Proc. The 5th World Congress on Theory of Machines and Mechanisms, Vol.2, p. 1343-1346, Montreal, PQ, Canada, 1979.
191. **W.Jacak W**: Strategies of searching for collision-free manipulator motions: automata theory approach, Robotica, Vol.7, 1989
192. **W.Khalil, A.Liégeois**: Génération des mouvements optimaux en robotique, R.A.I.R.O. Automatique/Systems Analysis and Control, Vol.18, 1984
193. **X.Hu, N.Ahuja**: Motion estimation under orthographic projection, IEEE Transactions on Robotics and Automation, Vol.7, No.6, 1991
194. **Y. Koren** : La robotique pour ingénieurs, McGraw-Hill, Paris, 1986
195. **Y.Fisher** : Fractal image compression. Springer-Verlag Ney York, 1994
196. **Y.Shan, Y.Koren**: Obstacle accommodation motion planning, Transactions on Robotics and Automation, Vol.11, No.1, 1995
197. **Y.Shimon**: Handbook of industrial robotics, Krieger Publishing Company, Malabar, Florida, 1992

198. **Z.Shiller**: On singular time-optimal control along specified paths, IEEE Transactions on Robotics and Automation, Vol.10, No.4, 1994
199. **Z.Shiller, S.Dubowsky**: 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, 1991
200. *** Microsoft Visual C++ & Video for Windows Help.
201. ***Contract Nr.4004/1995, Tema 15B, CNCSU: Planificarea și generarea mișcării la roboți. Analiza comparativă a diferitelor metode de generare a mișcării roboților. Elaborarea programelor de calcul pentru generarea mișcării roboților prin metodele studiate. **O. Căuș** - în colectivul de lucru
202. *** Contract Nr.5004/1996, Tema 377, CNCSU: Planificarea și generarea mișcării la roboți. Simularea mișcării unui robot pe baza modelului dinamic al acestuia.
203. *** Contract Nr.7004/1997, Tema 29/870, CNCSU: Planificarea și generarea mișcării la roboți. Optimizarea generării traiectoriilor unui robot, **O. Căuș** - în colectivul de lucru
204. *** Contract Nr.36/1998, Tema 34/279, CNCSU: Planificarea și generarea mișcării roboților mobili într-un spațiu cu obstacole. Modelarea spațiului de lucru al unui robot, **O. Căuș** - în colectivul de lucru
205. *** Contract Nr.35034/1999, Tema 37/60, CNCSIS: Planificarea și generarea mișcării roboților mobili într-un spațiu cu obstacole. Metode de planificare a mișcării roboților în cadrul problemei de bază. Planificarea mișcării într-un spațiu cu mai mulți roboți și obstacole mobile, **O. Căuș** - în colectivul de lucru.