



UNIVERSITATEA DIN CRAIOVA

**FACULTATEA DE AUTOMATICĂ,
CALCULATOARE ȘI ELECTRONICĂ**

**TEZĂ DE DOCTORAT
- REZUMAT -**

**TEHNICI SOFTWARE
AVANSATE PENTRU
CONDUCEREA PRIN VEDERE
ARTIFICIALĂ A ROBOȚILOR**

Conducător științific:

PROF. DR. ING. MIRCEA IVĂNESCU

Doctorand:

ASIST. DRD. ING. RĂZVAN TUDOR TĂNASIE

2008

Robotica este un domeniu de știință în curs de formare, cu un pronunțat caracter multidisciplinar. Izvoarele sale se găsesc în domeniile de: teoria mecanismelor, transmisii mecanice, dinamica mașinilor, tehnologie, dispozitive, electronică industrială, calculatoare, automată, inteligență artificială, biomecatronică, economie, inginerie industrială, management, ergonomie, psihosociologia muncii.

Robotica se ocupă de cercetări fundamentale și fundamentale orientate privind automatizarea operațiilor de manipulare a obiectelor în mediu industrial obișnuit, în medii inaccesibile sau greu accesibile operatorilor umani, în medicina și îngrijirea / recuperarea bolnavilor, în activități de servicii și casnice, în activități militare.

Structurile robotice hyper-redundante, identificate și prin sintagmele roboți tentaculari sau roboți continui, reprezintă o clasă de roboți de mare interes și în egală măsură de mare complexitate. Modelarea matematică a acestui tip de robot are în vedere atât abordarea cinematică cât și pe aceea dinamică. Modelele cinematice directe se bazează pe conceptul de segment curb și concatenarea acestor segmente pentru a forma un multisegment continuu.

Teza prezintă un ansamblu de sisteme software avansate pentru conducerea prin visual servoing a roboților, cu exemplificare pentru cazul particular al roboților tentaculari. Aceste sisteme cuprind un mediu grafic de dezvoltare și simulare pentru algoritmi de conducere, un set de algoritmi de prelucrare de imagini și calibrare adaptați pentru aplicațiile realizate, precum și o serie de algoritmi de conducere – algoritmi de cinematică, de dinamică și algoritmi de conducere prin visual servoing ce fac uz de și de ceilalți algoritmi proiectați și implementați.

Dintre avantajele multiple ale acestui mediu de simulare, trebuie amintite: costurile extrem de reduse pentru a putea lucra cu diverse configurații de roboți hyper-redundanți; posibilități nelimitate de test a diferitelor legi de conducere fără riscuri de a deteriora roboții; folosirea în procesul de calibrare stereo a sistemului real de visual servoing; urmărirea evoluțiilor modelelor de cinematică inversă sau de dinamică dezvoltate folosind metode bazate pe diferențe unghiulare; și nu în ultimul rând, avantaje educaționale.

Aceste aplicații grafice reprezintă unelte pentru îmbunătățirea calității procesului educațional în procesul de predare a conducerii roboților. Pentru tinerii studenți un simulator grafic este mai atractiv, mai ușor de urmărit și un stimulent pentru o învățare mai rapidă și mai corectă a comportamentului roboților în diverse aplicații de conducere. De asemenea, aceste aplicații pot fi folosite și în cadrul instruirii la distanță, fiind vorba de sisteme software ce pot fi ușor portate pe un alt calculator.

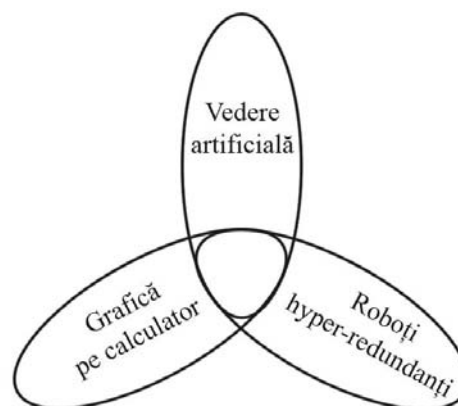


Figura 1. Domeniul lucrării de doctorat.

Prin tematica abordată, lucrarea de față se află la intersecția a trei mari domenii (Figura 1):

- Roboți hyper-redundanți,
- Vedere artificială – probleme legate de visual servoing și procesare de imagini,
- Grafică pe calculator.

Lucrarea este structurată pe șase capitole:

- Capitolul 1 – Roboți hyper-redundanți;
- Capitolul 2 - Visual servoing;
- Capitolul 3 - Mediu de dezvoltare pentru aplicații de conducere pentru roboți hyper-redundanți – aplicații de cinematică;
- Capitolul 4 - Mediu de dezvoltare pentru aplicații de conducere pentru roboți hyper-redundanți – aplicații de dinamică;
- Capitolul 5 - TEROB6 – Sistemul de vedere artificială;
- Capitolul 6 - TEROB6 – Sistemul de conducere prin visual servoing.

Teza este deschisă printr-o introducere unde sunt prezentate motivația și tema tezei, precum și direcțiile de cercetare investigate. De asemenea este prezentată pe scurt structura tezei de doctorat și subiectele discutate în fiecare capitol.

Capitolul 1 prezintă stadiul actual în domeniul roboților hyper-redundanți. Se prezintă o clasificare a manipuloarelor și se insistă asupra roboților continui și a avantajelor oferite de aceștia. Pentru aceștia se prezintă un model cinematic, prezentându-se modul de calcul al Jacobian-ului. Odată prezentat modelul cinematic, se descrie un model dinamic pentru această clasă de roboți și se identifică parametrii pentru acest model.

În capitolul 2 este realizată o prezentare a ceea ce înseamnă visual servoing și aplicații în acest domeniu. Sunt prezentate arhitecturile de visual servoing și caracteristicile fiecăreia din acestea, precum și Jacobian-ul imaginii. Apoi sunt prezentate mai multe aplicații de visual servoing, insistându-se pe aspectele de procesare de imagini.

Capitolul 3 prezintă mediul grafic de dezvoltare pentru aplicații de conducere pentru roboți hyper-redundanți și aplicațiile proiectate și dezvoltate în cadrul acestuia pentru problemele de cinematică directă și inversă. Este prezentat motorul grafic realizat pentru această aplicație și tehnicile folosite în cadrul acestuia. Sunt prezentați algoritmi proiectați și implementați, atât în domeniul grafic (randare și animație), cât și în cadrul modelelor cinematice. Este proiectat și implementat un evaluator de expresii și un derivator de expresii pentru modelul cinematic 3D. În finalul fiecărui subcapitol sunt prezentate testele experimentale realizate și sunt trase concluzii.

Principalele contribuții prezentate în acest capitol sunt mediul grafic de dezvoltare și simulare realizat, dar și modelele de cinematică testate și construite, precum și uneltele dezvoltate pentru acestea, cea mai importantă fiind GeDeE, un tool pentru generarea, derivarea și evaluarea expresiilor matematice.

Au fost elaborate axiome și teoreme în sprijinul modelelor matematice dezvoltate și implementate. A fost creat un model cinematic 2D pentru roboți hyper-redundanți. S-a creat un mediu de dezvoltare pentru aplicații de conducere pentru roboți tentaculari. Acest mediu include o bibliotecă proprie de funcții matematice caracteristice roboților tentaculari, un motor grafic complex pentru reprezentarea de animații realiste spre a ilustra mișcarea plană a

roboților hyper-redundanți, precum și o bibliotecă de funcții de programare pentru crearea, accesarea, modificarea și distrugerea instanțelor diverselor clase implementate.

Motorul grafic cuprinde un modul de desenare și randare, un motor de iluminare globală și un modul de texturare ce permite o serie variată de tehnici de texturare, atât uni-strat cât și multi-strat, folosind hărți de texturare pentru a crea o iluzie cât mai realistă a animației finale.

A fost proiectată o aplicație pentru a testa comportamentul roboților hyper-redundanți în cadrul unor probleme de cinematică directă 2D și a fost implementată în Microsoft Visual C++.NET 2005. Au fost folosite bibliotecile de funcții grafice Microsoft DirectX SDK.

Avantajele unei astfel de aplicații sunt evidente, unul dintre cele mai importante fiind costurile. Folosind o aplicație grafică pentru simulare, un robot hyper-redundant poate avea cât de multe segmente dorește utilizatorul. Acesta își poate crea diverși roboți cu diverse configurații și poate testa comportamentele lor în diverse probleme de cinematică. Acest fapt poate fi realizat foarte greu în realitate, considerând doar faptul că un astfel de robot cu numai trei segmente costă aproximativ 10.000 euro. De asemenea, unele legi de conducere se pot dovedi a fi dăunătoare pentru integritatea roboților, astfel că testarea lor inițial pe simulator se poate dovedi foarte benefică.

Alte avantaje ale unei astfel de aplicații sunt folosirea sa în procesul de calibrare a camerelor, reglare a sistemului de conducere (cum se modifică cele două unghiuri folosite în conducere în funcție de ce cabluri sunt acționate), dar și pentru testarea modelelor cinematice inverse și dinamice (prin calcularea unor poziții finale ale gripper-ului ce pot fi atinse prin configurații realizabile, ce apoi devin date de intrare pentru aplicațiile de cinematică inversă sau dinamică. În final se poate calcula variația unghiulară pentru fiecare unghi. Această variație nu reprezintă o eroare, o poziție finală a gripper-ului putând fi atinsă prin mai multe configurații finale ale robotului hyper-redundant.

Un alt avantaj este cel educațional. Această aplicație este o unealtă pentru îmbunătățirea calității procesului educațional în procesul de predare a conducerii roboților hyper-redundanți. Pentru tinerii studenți un simulator grafic bine realizat este mai atractiv, mai ușor de urmărit și un stimulent pentru o învățare mai rapidă și mai corectă a comportamentului roboților hyper-redundanți în diverse aplicații de conducere. Aceasta aplicație poate fi folosită și în cadrul instruirii la distanță, fiind vorba de de un sistem software ce poate fi ușor portat pe un alt calculator.

Au fost proiectați și implementați algoritmi pentru calcularea pozițiilor punctelor ce reprezintă baza segmentelor robotului hyper-redundant. Forma circulară a segmentelor arcuite a fost aproximată folosind un algoritm de interpolare cubică ce trece prin punctele de control. În acest sens a fost ales algoritmul Catmull – Rom. Simularea mișcării a fost realizată prin intermediul unui algoritm de interpolare liniară ce interpolează pentru fiecare punct de la poziția sa inițială la cea finală ținând cont de numărul de pași de animație stabiliți.

A fost creat un model cinematic 3D pentru roboți hyper-redundanți. S-a creat un mediu de dezvoltare pentru aplicații de conducere pentru roboți tentaculari. Pentru această aplicație s-au folosit ca date de intrare: structura robotului, configurația sa inițială și cea finală. Fiecărui segment al robotului hyper-redundant i-a fost atașat un sistem de coordonate, sisteme de coordonate ce sunt aliniate în poziția inițială. Configurațiile exprimă orientarea segmentelor robotului hyper-redundant prin două unghiuri: unghiul de rotire și unghiul făcut de coardă cu

planul XOZ al propriului sistem de coordonate, unghi complementar unghiului de arcuire al segmentului.

Modulul matematic cuprinde lucrul cu matrici ce a fost folosit pentru a reține transformările, fie că este vorba de transformările segmentelor în sistemele de coordonate proprii, sau de transformările între sisteme de coordonate. S-a urmărit determinarea și trecerea, prin produs matriceal și vectorial-matriceal, a coordonatelor punctelor generatoare de cilindrii (bazele segmentelor) din sistemele de coordonate atașate propriilor segmente într-un sistem global de reprezentare.

Pentru iluminare au fost folosite, pe lângă lumina ambientală, două alte surse de lumină: una de tip direcțional atașată camerei care are componentă difuză și una de tip spot care este în permanență orientată pe robot. Această sursă are atât componentă difuză cât și componentă speculară. Toate sursele de lumină sunt predefinite albe și au valori parametrice implicite adaptate scenelor și materialelor folosite. Toți parametrii surselor de lumină (poziție, culoare, unghiuri etc.) pot fi modificați în timpul aplicației.

A fost proiectat și implementat un modul interactiv dinamic care să permită utilizatorului să aleagă poziția și orientarea sistemului de coordonate. Astfel, în orice moment pe parcursul animației, utilizatorul poate stabili poziția din care să privească. Sensul a fost prestabilit spre originea sistemului, deoarece acolo este poziționat inițial robotul hyper-redundant.

De asemenea, pentru a permite o vizionare cât mai realistă și care să poată oferi cât mai multe informații utile despre conducerea robotului, utilizatorului i s-a oferit posibilitatea să aleagă care este direcția de privire, și implicit, orientarea sistemului de coordonate de vedere. Dacă inițial sistemul de coordonate este poziționat cu axa O-Y în jos și se privește în lungul axei O-Z, spre sensul pozitiv, utilizatorul poate, prin apăsarea unei singure taste, să privească de-alungul oricăreia dintre axele sistemului de coordonate.

Desenarea este mai complexă în aplicațiile tridimensionale, segmentele robotice trebuind ilustrate prin cilindrii. Astfel, pentru fiecare punct generat de algoritmul de interpolare Catmull-Rom s-a construit nu un patrulater (precum în cazul aplicațiilor 2D), ci un cilindru. Pentru determinarea punctelor ce compun cercurile – bazele segmentelor s-a folosit o ecuație vectorială parametrică a cercului în spațiu. Această ecuație determină coordonatele unui punct ce aparține unui cerc tridimensional, rezultat ce poate fi verificat prin calcularea distanței de la punct la plan și de la punct la cerc.

Un aspect în plus care trebuie luat în considerare în acest caz este că, desenarea făcându-se pe un ecran digital (matrice de pixeli) nu se pot afișa curbe, și implicit nici cilindrii. De aceea s-a apelat la o aproximare pentru a ilustra cercurile ce reprezintă bazele cilindrilor prin intermediul unor polilinii închise. Acest proces care încearcă simularea suprafețelor curbe prin linii poartă numele de rafinare.

În alegerea gradului de rafinare trebuie luat în considerare faptul că o valoare ridicată va ilustra mult mai realist suprafețele curbe, pe de o parte, dar, pe de altă parte, procesul de rafinare este intens consumator de resurse. În acest scop trebuie găsit un echilibru între resursele alocate și gradul de realism necesar. Astfel s-a ales un sistem dinamic de ajustare a gradului de rafinare bazat pe un raport de invers proporționalitate cu modificarea distanței cameră – robot hyper-redundant. S-au stabilit valori inițiale predefinite pentru distanța cameră – robot și gradul de rafinare corespunzător acesteia.

A fost creat un model de cinematică inversă 2D pentru roboți hyper-redundanți. S-a creat un mediu de dezvoltare pentru aplicații de conducere pentru roboți tentaculari. Pentru determinarea unghiurilor finale ale robotului hyper-redundant s-a folosit un algoritm iterativ.

Pe baza Jacobian-ului numeric determinat, s-a calculat pseudo-inversa numerică a Jacobian-ului. Algoritmul folosit în aceste aplicații verifică întâi dacă matricea este inversabilă. Dacă este cazul unei matrici inversabile, se permite calcularea inversei sale. În caz contrar, se folosește o altă matrice pe post de pseudo-inversă.

Utilizatorul poate alege între a folosi inversa (în rarele cazuri când există), transpusa Jacobian-ului (ce are erori de calcul mari, de multe ori atinge greu soluția dorită, dar este mult mai economicoasă din punct de vedere al resurselor utilizate) și pseudo-inversa Moore-Penrose (cea mai indicată variantă) prin setarea valorii variabilei de inversare.

Recalcularea coordonatelor se poate face atât înainte de pornirea simulării, sau în timp real (caz în care se modifică algoritmul de animație) după cum se setează utilizatorul aplicația.

Toți algoritmi proiectați și implementați pleacă de la premisa că unghiurile sunt valori date în intervalul $[0, 2\pi]$, în caz contrar realizându-se deplasări succesive până când unghiurile se află în intervalul primar.

S-a impus o condiție de integritate. Chiar dacă în spațiul virtual, teroretic, segmentele unui robot hyper-redundant se pot arcui oricât, un segment al unui robot hyper-redundant real nu poate face acest lucru. De aceea, pentru a păstra realismul aplicației, au fost impuse valori maxime pentru unghiurile de arcuire. Aceste valori nu pot fi depășite. În cazul care una din valorile obținute depășește totuși unghiul de arcuire maxim, aceasta este restricționată la acea valoare.

Valorile unghiurilor sunt recalulate la fiecare iterație, și după ce toate unghiurile au fost calculate (la fiecare pas) se verifică (și se aplică modificările necesare dacă este cazul) condițiile de interval primar și de integritate.

Trecerea de la o poziție la poziția următoare este dereminată de algoritmul de conducere pentru cinematică inversă pe baza erorii de poziție (diferența dintre poziția actuală și cea dorită a gripper-ului).

Însă un pas de mișcare mare poate genera erori mari și automat, mișcarea să "sară" peste poziția finală dorită. Astfel s-a introdus un factor de control al mișcării dependent de Jacobian și de pseudo-inversa sa.

Acesta este factorul de eroare permis - φ (dependent de ε - eroarea maximă admisă pentru poziția finală, și având valori mai mici decât aceasta) și, distanța de mișcare se va înjumătăți atât timp cât eroarea depășește acest factor. Această "corectare" a erorii garantează convergența algoritmului (chiar dacă în unele cazuri mai lent, dar mai sigur dacă este posibil) către poziția țintă dorită a gripper-ului cu o marjă minimă de eroare acceptată.

S-a dezvoltat un algoritm iterativ de cinematică inversă în nouă pași, ce folosește inversa Jacobian-ului pentru determinarea coordonatelor generalizate ce situează gripper-ul robotului hyper-redundant în poziția țintă dorită. Încheierea algoritmului semnalează că robotul a ajuns

într-o configurație în care poziția gripper-ului se află situat în cercul cu centrul în poziția țintă dorită și de rază eroarea maximă admisă ε .

Algoritmul are și un mecanism de siguranță pentru cazurile când poziția țintă dorită se află în afara spațiului de operare al robotului hyper-redundant și nu poate fi atinsă. În acest sens, dacă numărul de iterații depășește valoarea de prag impusă, utilizatorul va fi înștiințat că acea poziție a gripper-ului nu poate fi atinsă de robotul selectat, plecând de la configurația inițială dată. Algoritmul poate fi rulat atât on-line cât și off-line, în funcție de setările impuse de utilizator.

Forma circulară a segmentelor arcuite a fost aproximată folosind un algoritm de interpolare cubică ce trece prin punctele de control. În acest sens a fost ales algoritmul Catmull – Rom. Simularea mișcării se poate realiza, după dorința utilizatorului, fie prin interpolare între pozițiile inițiale date și cele finale calculate cu algoritmul de cinematică inversă, fie în timp real folosind pozițiile intermediare, generate pe parcursul găsirii soluției, de către algoritmul de cinematică inversă.

A fost creat un model cinematic de cinematică inversă 3D pentru roboți hyper-redundanți. S-a creat un mediu de dezvoltare pentru aplicații de conducere pentru roboți tentaculari.

S-au folosit ca date de intrare: structura robotului, configurația sa inițială și poziția țintă dorită a gripper-ului. Fiecărui segment al robotului hyper-redundant i-a fost atașat un sistem de coordonate, sisteme de coordonate ce sunt aliniat în poziția inițială. Configurațiile exprimă orientarea segmentelor robotului hyper-redundant prin două unghiuri: unghiul de rotire și unghiul făcut de coardă cu planul XOZ al propriului sistem de coordonate, unghi complementar unghiului de arcuire al segmentului.

A fost construit un generator de expresii pentru a reține valorile coordonatele operaționale și apoi pentru a determina un Jacobian de expresii. Structurile folosite sunt ilustrate în Figura 2. Aceste structuri (atom, termen, element) au folosit clase template pentru a reține liste de obiecte sau pentru a le accesa prin iteratori.

Au fost proiectați și implementați algoritmi pentru generarea expresiilor, derivarea acestora și decriptarea lor folosind un interpretor. În scopul realizării tuturor acestor operații a fost necesară supraîncărcarea mai multor operatori și crearea unor funcții de optimizare.

Algoritmii proiectați și implementați sunt intens computaționali, din acest motiv optându-se pentru implementarea lor folosind fire de execuție, fapt ce va paraleliza mare parte din procese, permițând un timp de execuție mult mai scurt. Paralelizarea proceselor a fost realizată astfel încât să permită utilizatorului să poate alege câte fire de execuție să fie create, în funcție de puterea de calcul aflată la dispoziția acestuia.

Pentru aplicațiile 3D, pentru un robot hyper-redundant cu n segmente, Jacobian-ul are 3 linii (câte una pentru fiecare coordonată operațională – X , Y și Z) și $2n$ coloane (câte una pentru fiecare coordonată generalizată – n unghiuri de arcuire $\alpha_{arcuire}^i$ și n unghiuri de rotire α_V^i), adică, în total, $6 \cdot n$ elemente ce trebuie determinate. Astfel, algoritmul de derivare a fost aplicat de $6 \cdot n$ ori, rezultând un efort de calcul destul de important. Din acest motiv optimizările, dar mai ales paralelizarea folosind fire de execuție s-au dovedit esențiale în obținerea unui algoritm rapid.

Pentru a determina valorile Jacobian-ului se folosește un interpretor pentru a descrie expresiile și pentru a obține valorile numerice ale Jacobian-ului. Acest interpretor are un limbaj relativ restrâns și două hărți de mapare ce permit "traducerea" în format matematic a atomilor, plecând de la tipurile și valorile acestora. Această unealtă reprezintă o contribuție importantă, fapt ilustrat și de structura GeDeE, structură ce permite o adaptare ușoară la o gamă variată de expresii pentru diferite aplicații, după cum s-a arătat.

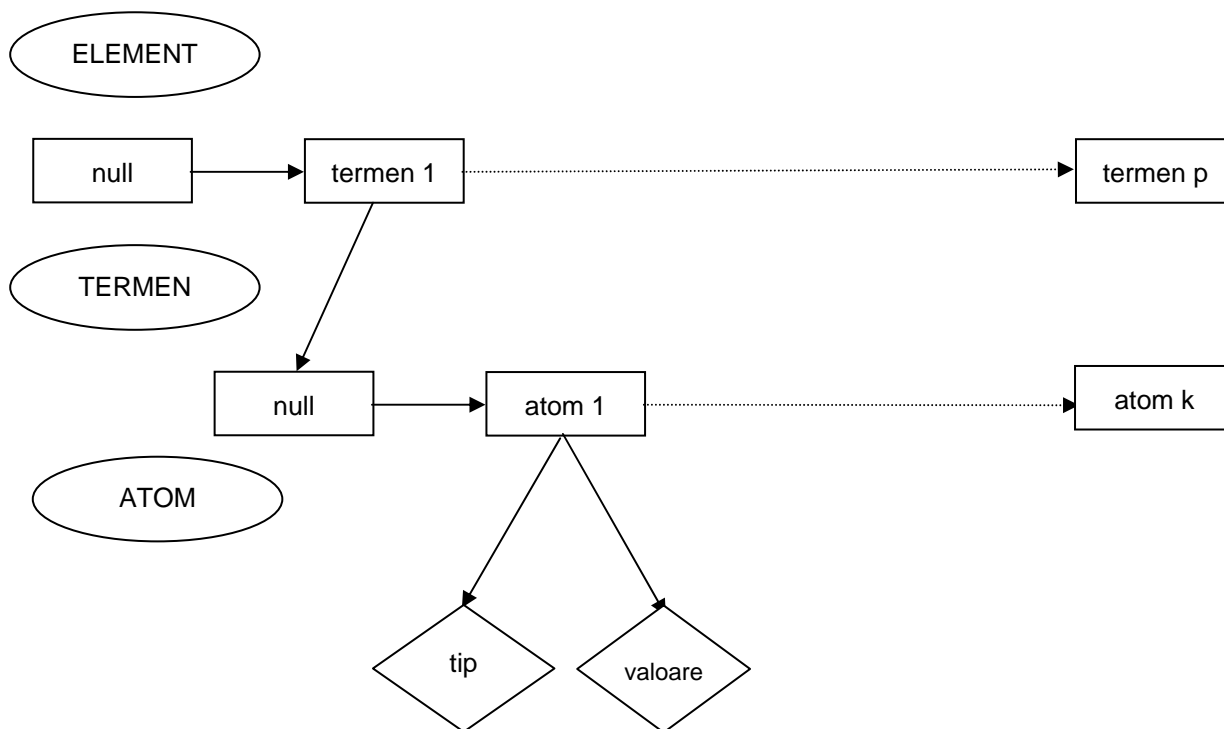


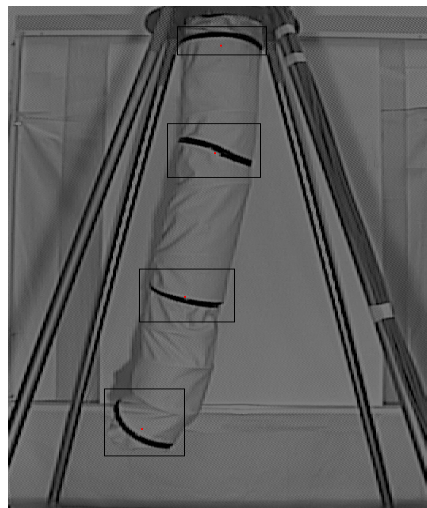
Figura 2. Arhitectura unei expresii generate cu sistemul pentru generarea de expresii.

Capitolul 4 prezintă mediul grafic de dezvoltare pentru aplicații de conducere pentru roboți hyper-redundanți și aplicațiile proiectate și dezvoltate în cadrul acestuia pentru problemele de dinamică. Este prezentat modelul dinamic folosit pentru un manipulator atât cu încărcare, cât și fără încărcare și algoritmul de conducere aferent. Sunt prezentate funcțiile adiționale aduse la motorul grafic pentru a simula aceste modele.

Contribuția adusă prin realizarea mediului grafic de dezvoltare și simulare prezentat în capitolul anterior este îmbunătățită prin adăugarea unor funcții ce permit realizarea și testarea de aplicații de dinamică. În acest sens a fost implementat un model dinamic ce ulterior a fost testat în cadrul simulatorului. Modulul matematic cuprinde întregul modul matematic folosit pentru cinematica inversă 3D (lucrul cu matrici, aplicații de derivare, formule geometrice, calcul trigonometric), dar au fost adăugate și alte funcționalități, cum ar fi un modul de integrare și evaluare de expresii. În final sunt prezentate testele experimentale realizate cu acest simulator și în MATLAB și sunt trase concluzii. În Figura 3 a) este prezentată o captură din simularea unei aplicații de dinamică în mediul de dezvoltare și simulare creat.



a)



b)

Figura 3. a) Imagine din mediul de dezvoltare și simulare; b) Imagine achiziționată și procesată în timpul conducerii robotului tentacular.

Capitolul 5 prezintă sistemul de vedere artificială folosit. Sunt prezentate noțiunile de vedere artificială folosite. Este descris sistemul de vedere artificială ce a fost construit. De asemenea sunt descriși doi algoritmi de calibrare (poziționare și calibrare a camerelor) ce au fost proiectați și implementați, unul folosind obiecte special construite pentru acest proces și un al doilea care face uz de mediul grafic de dezvoltare prezentat în capitolele anterioare.

Algoritmii de poziționare și orientare ai camerelor reprezintă o contribuție originală pentru lucrarea de doctorat și sunt, de asemenea de o importanță ridicată pentru algoritmul de conducere prin visual servoing implementat, deoarece acesta necesită poziționarea și orientarea camerelor astfel încât să aibă direcții de privire ortogonale.

În acest sens, în acest capitol este prezentat un algoritm de corelare optică proiectat și implementat, algoritm necesar pentru ca cele două camere să capteze obiectul de calibrare la aceeași scară și astfel calculele unghiulare să nu fie influențate.

În final sunt prezentate rezultatele obținute la scenariile de test realizate pentru a demonstra validitatea și eficacitatea algoritmilor proiectați și implementați.

În capitolul 6 este prezentat sistemul de conducere realizat prin visual servoing și sunt prezentate funcțiile de procesare de imagini proiectate, adaptate și implementate. Sunt prezentate aspectele teoretice necesare, modul de control al camerelor, aplicația de achiziție și procesare de imagini și algoritmul de conducere prin visual servoing.

S-a avut în vedere studiul unei metode de servoing bazat pe imagine pentru un robot hyper-redundant. Acesta are la bază vederea binoculară. Măsurarea continuă a parametrilor brațului, derivată din calculul în timp real al fluxului optic binocular al celor două imagini, a fost comparată cu poziția dorită pentru braț. Funcția de control al erorilor a fost construită în spațiul cartezian 3D utilizând informația video obținută de la cele două camere de luat vederi.

Errorile 2D obținute în planele celor două imagini au fost determinate de diferențele dintre valorilor existente și dorite ale unghiurilor ce definesc proiecțiile brațelor. Errorile planelor pot fi considerate erorile brațelor. Aceste erori au fost utilizate pentru calculul erorii spațiale, fiind în același timp sintetizată o lege de control. Pentru sistemul de conducere cu circuit închis și

legea de control, stabilitatea este demonstrată prin aplicarea celei de-a doua metode a lui Lyapunov. Funcția eroare este calculată în spațiul virtual al imaginilor iar calibrarea nu este necesară dând astfel posibilitatea sintezării unor legi de control mai robuste.

Pentru a testa algoritmi proiectați și implementați au fost realizate o serie de experimente. Rezultatele acestora, însoțite de imagini capturate, datele obținute și grafice comparative sunt prezentate în finalul lucrării. Rezultatele obținute pe modelul real sunt comparate cu cele obținute prin simulare. Rezultatele obținute și prezentate în acest capitol validează algoritmi proiectați și implementați și mediul de dezvoltare și simulare construit. În Figura 3 b) este prezentată o imagine achiziționată în timpul conducerii robotului tentacular și apoi procesată pentru a se determina centrele bazelor segmentelor robotice (care sunt evidențiate pe imagine prin algoritmul de clustering aplicat).

În final o serie de concluzii sunt trase raportat la motorul grafic și la modelele matematice dezvoltate. Sunt prezentați pe scurt algoritmi proiectați și implementați pentru modelele de cinematică directă, inversă și dinamică, motorul grafic creat și tehnicile folosite pentru randare și animație, modelele matematice dezvoltate, algoritmi de calibrare realizați, metodele de achiziție și prelucrare de imagini, și, bineînțeles, algoritmul de conducere prin visual servoing.

În urma experimentelor efectuate s-a reușit o calibrare corectă a echipamentelor și obținerea unor rezultate bune. Pentru a elimina problemele întâlnite, dar și pentru a obține rezultate superioare din punct de vedere calitativ am considerat mai multe variante de îmbunătățiri ulterioare dintre care amintim:

- Modificarea structurii mecanice a robotului prin realizarea unor segmente cilindrice cu razele cilindrilor de dimensiuni din ce în ce mai mici, pentru a ușura manevrarea segmentelor terminale, scăzând astfel sarcina de încărcare a motoarelor ce acționează segmentele inferioare ale robotului. O altă alternativă ar fi realizarea unor segmente sub formă de trunchi de con, model apropiat de varianta biologică a trompei de elefant;
- Îmbunătățirea algoritmilor de conducere a motoarelor pas cu pas pentru micșorarea vibrațiilor și creșterea preciziei de mișcare. În acest scop se are în vedere folosirea informațiilor furnizate de o cameră de luat vederi de mare viteză. O astfel de cameră este în curs de achiziție din fondurile unuia din contractele de cercetare în care doctorandul este membru. Acest fapt va crea premisele îmbunătățirii legilor de conducere și, ca o consecință, a reducerii semnificative a erorii de poziționare, iar în final reducerea posibilității de a „sări” peste poziția țintă.

Rezultatele experimentale efectuate au demonstrat validitatea și eficacitatea algoritmilor proiectați și implementați. Compararea rezultatelor obținute în cadrul mediului grafic de simulare proiectat și implementat cu cele obținute din conducerea robotului au validat atât sistemul de conducere construit cât și valorile obținute de aplicațiile construite în mediul grafic de simulare.

Apreciem ca principală realizare fundamentarea, proiectarea și implementarea unui mediu de dezvoltare de aplicații (echipamente, metode, tehnici și algoritmi) care să permită continuarea cercetărilor pentru dezvoltarea unor structuri tentaculare din ce în ce mai performante.