

CONTRIBUȚII LA DEZVOLTAREA METODELOR ȘI ALGORITMILOR PENTRU CALCULUL POZIȚIEI BRAȚELOR ROBOTICE ÎN SPAȚIU FOLOSIND RECUNOAȘTERE DE CULORI. IMPLEMENTARE ÎN FPGA

dipl.-ing. Roland SZABÓ

Rezumatul tezei de doctorat

CAPITOLUL 1. INTRODUCERE

Teza de doctorat prezintă o contribuție la calculul poziției brațelor robotice în spațiu. Metoda utilizată este una optică, care se bazează pe detecție de marcheri [39].

Se va prezenta în continuare controlul unui braț robotic în spațiu utilizând două camere video. În industrie cei mai mulți roboți nu au un sistem propriu de vedere, ei doar se mișcă pe căile învățate sau predefinite anterior. Ei în principiu nu iau nici o decizie, neavând inteligență artificială în software-ul lor de control [12].

Metoda care va fi prezentată nu dorește a înlocui tehnicile consacrate de calcul a poziției brațului robotic (cinematica directă și cinematica inversă), ci dorește să adauge plus valoare a unui sistem tradițional cu brațe robotice preprogramate. Calculul optic automat al poziției brațului robotic poate fi adăugat metodelor consacrate, astfel încât robotul să fie programat punct cu punct și modulul de calcul optic automat să fie utilizat pentru corecții sau pentru autocalibrare. Desigur, pentru aplicații complexe și speciale, o metodă optică este mult mai flexibilă, decât una tradițională [40].

1.3. Structura tezei

Teza este organizată astfel:

- **Capitolul 2** prezintă resursele necesare pentru implementare și modul cum au fost utilizate. Tot în acest capitol este inclusă o comparație între diferitele plăci de dezvoltare utilizate.
- **Capitolul 3** prezintă descrierea metodei, descrierea părții experimentale, modelul matematic propus și explicația algoritmului implementat. Acest capitol reprezintă partea teoretică a lucrării.
- **Capitolul 4** prezintă schemele bloc, rezultatele obținute, prezentarea aplicațiilor dezvoltate, prezentarea diferitelor experimente și măsurări. Acest capitol reprezintă partea practică a lucrării. Tot în acest capitol sunt comparate aplicațiile realizate în diferite limbaje de programare și sunt trase concluzii, ierarhizând implementările propuse.
- **Capitolul 5** prezintă o comparație între versiunile de implementare și expunerea avantajelor FPGA-urilor. Sunt făcute comparații între versiuni de implementare și grafice statistice pentru evaluarea metodei de implementare. Este prezentată influența diversilor parametri asupra performanțelor sistemului. Sunt comparate și măsurări efectuate la diferite fluxuri luminoase, temperaturi de culoare și distanțe. Rezultatele proprii sunt comparate cu alte versiuni de implementare realizate de alți cercetători.
- **Capitolul 6** prezintă contribuțiile și concluziile finale ale lucrării. În final sunt creionate direcțiile viitoare de cercetare și articolele personale publicate, legate de această teză de doctorat.

CAPITOLUL 2. RESURSE. METODE

În fig. 2.2 se observă arhitectura SoC-ului Zynq-7000 creat pe placa ZYBO sau ZedBoard. Pe schemă s-au evidențiat cele două nuclee ARM Cortex – A9 cu 512 Kocteți L2 cache și memorie de 256 Kocteți pe chip.

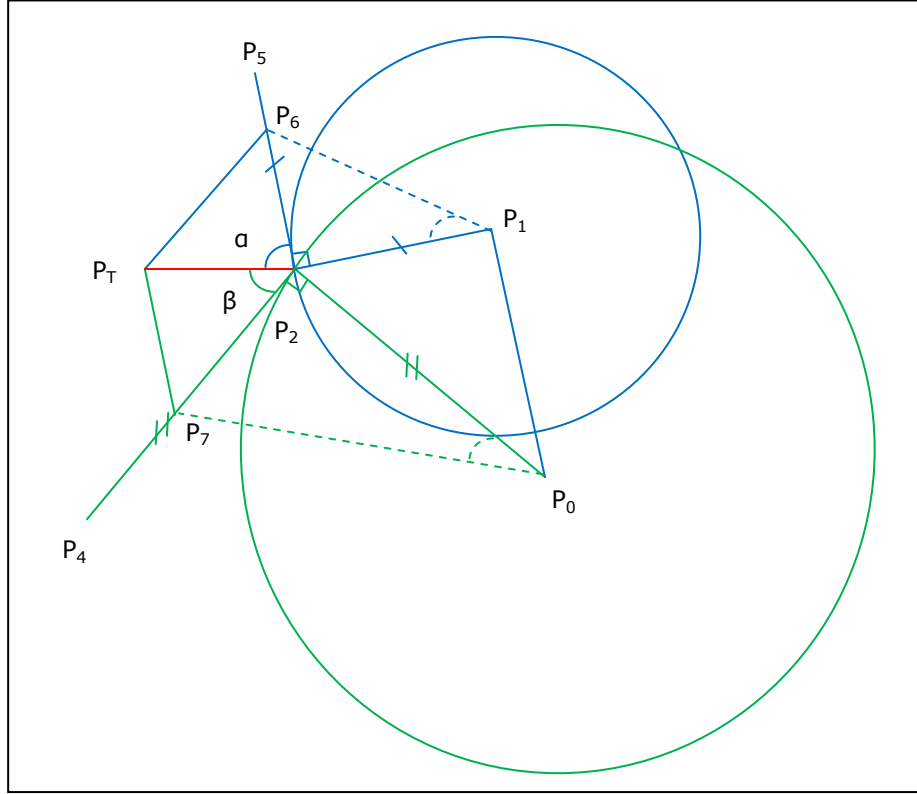


Fig. 3.6 Desene ajutătoare pentru calculul de distanțe 2D imprimate pe imaginea video

Cunoscând lungimea segmentelor P_2P_6 și P_2P_1 (lungimile catetelor triunghiului dreptunghic albastru) pentru care se dorește să se afle unghiul $P_2P_1P_6$ de mișcare a motorului cotului brațului robotic.

Acest desen se reîmprospătează pe imaginea cu brațul robotic în mod dinamic de fiecare dată când acesta își schimbă poziția. Dimensiunile diferă, însă simetria se păstrează, laturile paralelogramului $P_2P_6P_1P_7$ vor fi tot timpul paralele între ele. Anumite porțiuni pot fi ascunse (pentru lizibilitate) pe imaginea finală (de exemplu segmentele P_5P_6 , P_4P_7 , P_7P_0 , P_6P_1).

În final coordonatele lui P_6 s-au obținut în felul următor:

$$\begin{cases} x_6 = \frac{(m_a x_2 - m_\beta x_T + y_T - y_2)}{m_a - m_\beta} \\ y_6 = m_a(x_6 - x_2) + y_2 \end{cases}$$

Executând aceleași calcule pentru y_7 , coordonatele pentru P_7 s-au obținut cu (3.12) și (3.13).

$$m_\beta x_7 - m_a x_7 = y_T - m_a x_T - y_2 + m_\beta x_2 \quad (3.12)$$

$$\begin{cases} x_7 = \frac{(m_\beta x_2 - m_a x_T + y_T - y_2)}{m_\beta - m_a} \\ y_7 = m_\beta(x_7 - x_2) + y_2 \end{cases} \quad (3.13)$$

În continuare se prezintă o metodă, prin care se calculează distanțe în spațiu utilizând două camere și un singur algoritm (fig. 3.9).

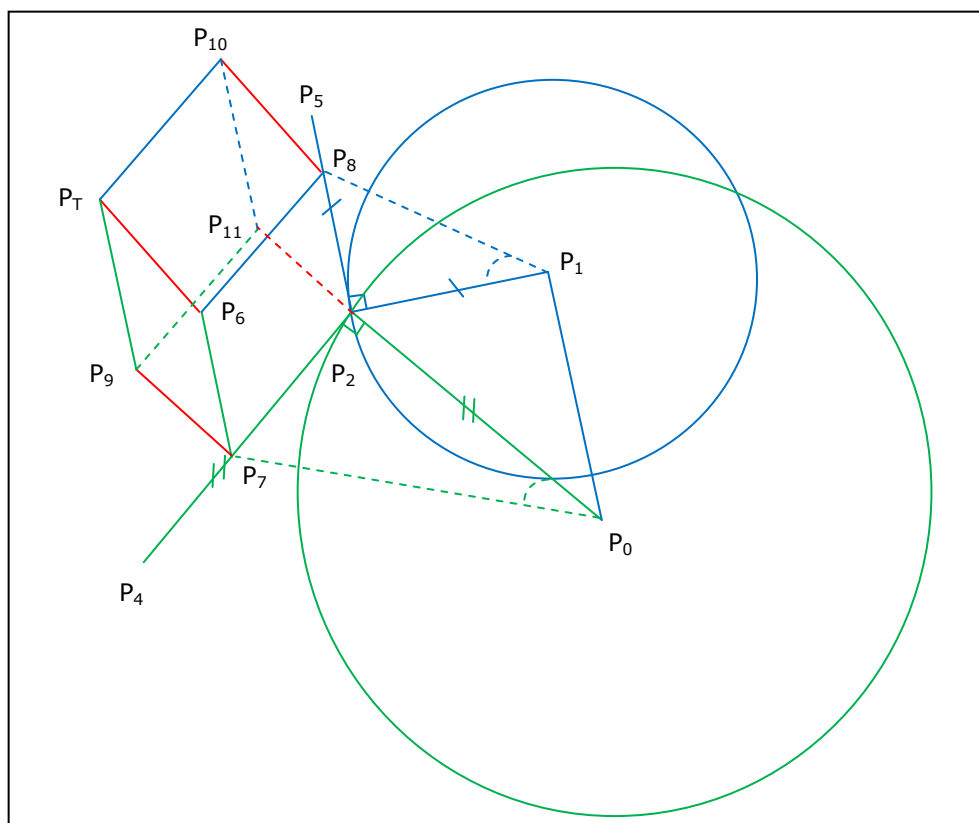


Fig. 3.9 Desene ajutătoare pentru calculul de distanțe 3D imprimate pe imaginea video

Punctele sunt similare ca cele din fig. 3.6, doar că în cazul de față în loc de paralelogram este un paralelipiped, totul fiind tridimensional. Punctul P_T nu este unit cu punctul P_2 , deoarece mai trebuie trasat un paralelogram paralel cu cel vechi (paralelogramul $P_8P_TP_9P_{11}$ este paralel în spațiu cu paralelogramul $P_8P_6P_7P_2$).

Acest lucru se poate realiza în felul următor. Din nou se trasează tangente la cercuri cu lungime egală cu razele acestora, iar punctele $P_4P_2P_5$ formează un plan. Din punctul țintă P_T se calculează perpendicularul pe planul $P_4P_2P_5$, care stă pe o dreaptă și intersectează planul în punctul P_6 , care este colțul celălalt al paralelogramului $P_8P_6P_7P_2$ (este de fapt sistemul de coordonate în 2D, pentru axele OY și OZ). Se generează paralelogramul $P_8P_6P_7P_2$ cu metoda prezentată în fig. 3.6).

Se calculează intersecția dreptei care trece prin punctul P_6 și care este paralelă cu dreapta verde care trece prin punctele P_2 și P_4 . Acesta sigur va intersecta dreapta care trece prin punctele P_5 și P_2 . Intersecția este făcută prin punctul de interes P_8 .

3.3. Concluzii

În acest capitol au fost prezentate metoda și ideea de bază al algoritmului utilizat. S-au comparat studiile similare din literatura de specialitate cu modelul de personal original și s-au reliefat diferențele și avantajele metodei proprii.

Am insistat pe deducerea formulelor utilizate ulterior în teză în diferite programe de control al brațelor robotice, pe diferite platforme de implementare (FPGA și PC).

Sunt prezentați mai mulți algoritmi de calcul ai distanțelor, în plan și în spațiu.

Deoarece masa roboților utilizați la partea experimentală nu este semnificativă, iar sarcina utilă este redusă, aspect coroborat de controlul prin servomotoare, problemele de dinamică nu au fost luate în considerare, considerând autocalibrarea optică suficientă.

CAPITOLUL 4. REZULTATE EXPERIMENTALE

4.2. Controlul poziției brațului robotic în spațiu (3D)

4.2.1. Implementarea în LabWindows/CVI și RoboRealm

În fig. 4.22 se observă rezultatul, imaginea 3D a brațului robotic Lynxmotion AL5B generată în MATLAB. Această imagine este o imagine 3D dinamică, care se poate roti în sistemul de coordonate 3D, astfel se poate vedea imaginea din toate părțile ca și un obiect 3D introdus în calculator.

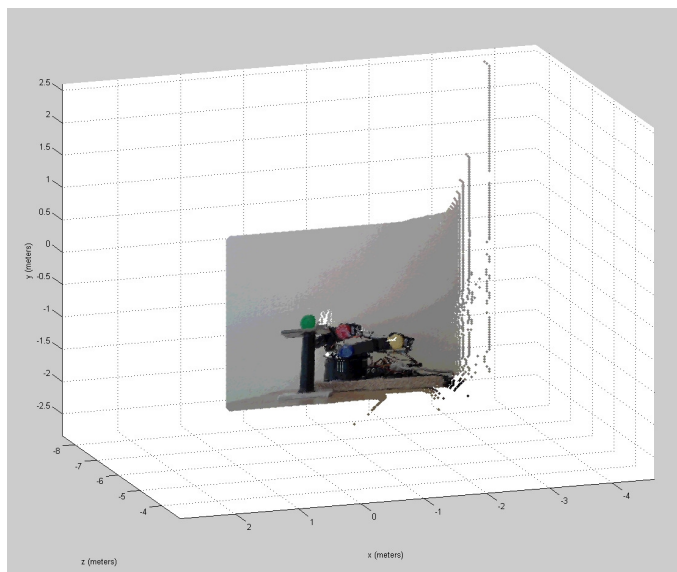


Fig. 4.22 Imaginea generată 3D în MATLAB

În fig. 4.36 se observă experimentul de control al brațului robotic Lynxmotion AL5B cu ajutorul recunoașterii dopurilor colorate plasate la articulațiile brațului robotic, utilizând camere IP conectate pe interfața Ethernet la PC printr-un switch. Codul este scris în C/C++ utilizând biblioteca OpenCV pe sistemul de operare Linux.

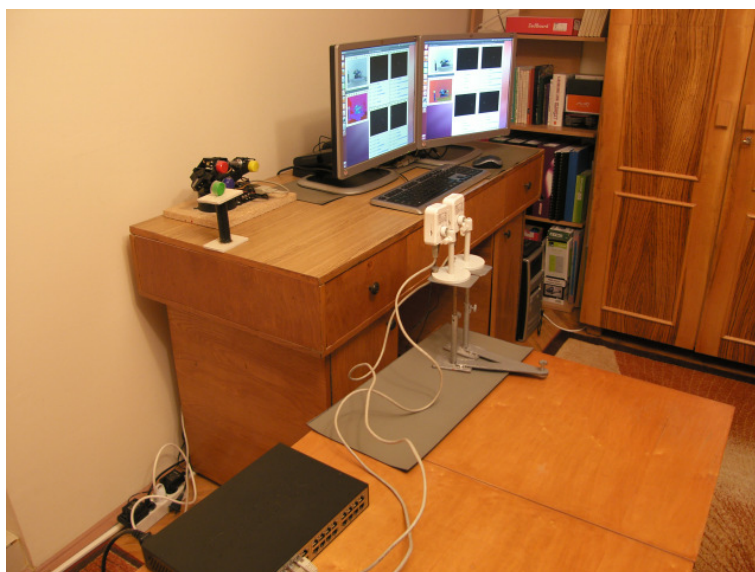


Fig. 4.36 Experimentul pentru controlul brațului robotic Lynxmotion AL5B în 3D cu detecție de marcheri folosind camera IP în C/C++ cu OpenCV sub Linux

În fig. 4.55 se observă imaginea mărită al brațului robotic SCORBOT-ER III, achiziționată cu cele două camere, stângă și dreaptă.

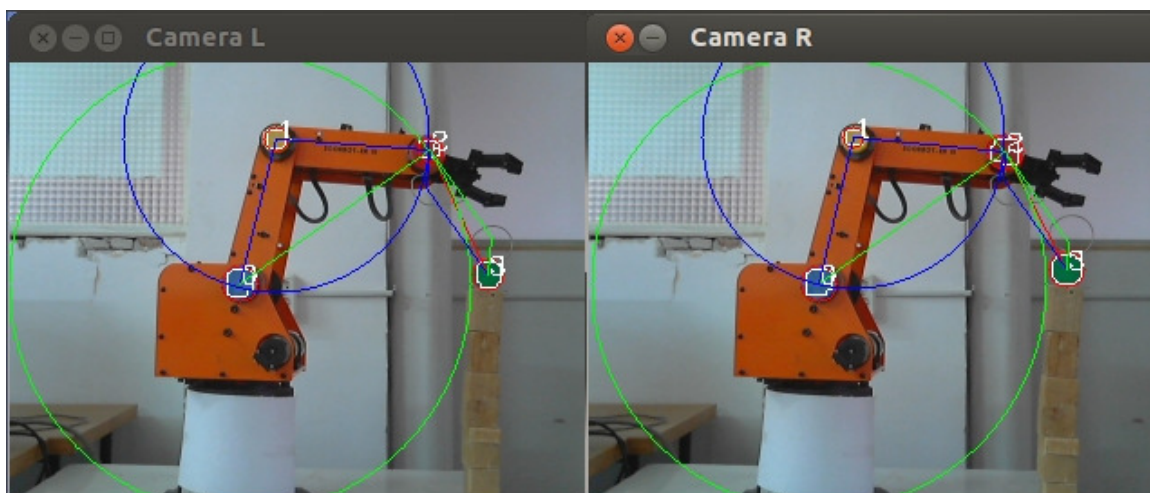


Fig. 4.55 Imagine mărită din programul de control al brațului robotic SCORBOT-ER III cu liniile ajutătoare pentru calculul poziției brațului robotic în 3D

4.4. Analiza comparativă a performanțelor implementărilor

În figura 4.61 este redată o comparație a resurselor folosite la diferitele implementări.

4.5. Concluzii și contribuții

Pornind de la controlul cu succes utilizând un robot educațional, matricea implementărilor conține practic toată combinația de resurse, limbaje de programare și sisteme de operare disponibilă în acest moment. Rezultatele au fost validate și pe robotul industrial SCORBOT-ER III.

Din comparația diverselor implementări rezultă următoarele:

- În cazul VHDL, sistemul ocupă foarte puține resurse, însă codul este foarte lung și foarte complicat, astfel necesită un efort sporit din partea celui care face implementarea.
- La implementările în Linux resursele ocupate în timpul execuției sunt relativ mari, însă nu trebuie uitat că implementarea necesită un efort redus față de alte implementări, biblioteca OpenCV fiind cea mai flexibilă, dar consumul de resurse este relativ ridicat.
- Având în vedere că în ziua de azi și sistemele dedicate sunt relativ puternice, folosirea de resurse multe nu ar fi o problemă, mai ales că pe un sistem care controlează un braț robotic nu se mai rulează nimic în paralel.
- Cel mai bun compromis este Python pe sistemul de operare Linux, care este relativ simplu și nici din resurse nu ocupă prea mult.
- Nu trebuie uitat nici faptul că implementările pe Linux sunt de asemenea atractive, deoarece aceste se pot porta cu ușurință pe multe sisteme dedicate.

Testarea sistemului a fost efectuată după ce s-a așezat punctul țintă (dopul verde) în toate cele opt colțuri în aria de acoperire a brațului robotic (stânga sus – jos, dreapta sus – jos, pe ambele părți ale brațului robotic) și brațul robotic a reușit să ajungă autonom în toate aceste colțuri fără a fi învățat punctele țintă, printr-un proces anterior de calibrare; controlul e în timp real, calculul poziției țintă fiind efectuat de algoritmul prezentat (și liniile ajutătoare pentru distanțe sunt retrasate în timp real pe imaginea achiziționată la fiecare deplasare al brațului robotic).

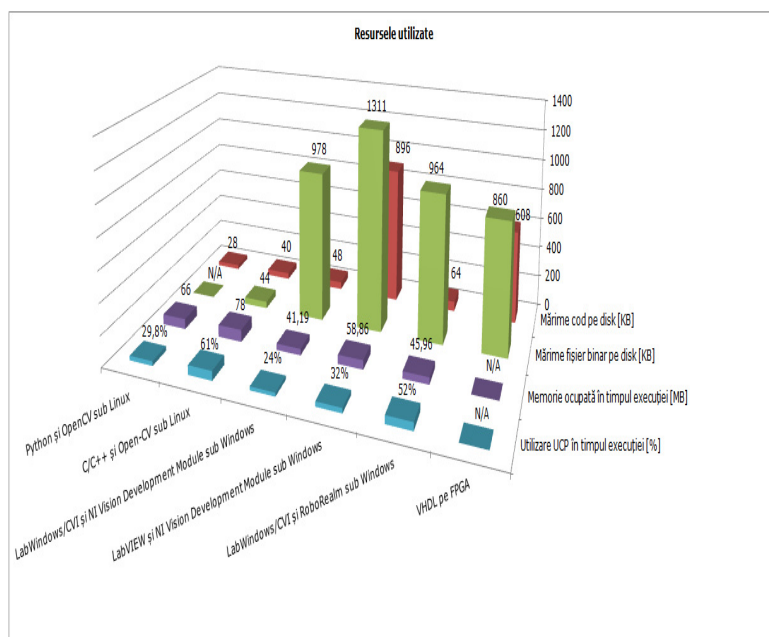


Fig. 4.61 Comparația între resursele utilizate la diferite implementări

CAPITOLUL 5. ANALIZA ȘI COMPARAȚIA REZULTATELOR

5.1.2. Analiza măsurării distanțelor la diferite fluxuri luminoase

În fig. 5.9 este redată analiza de capabilitate. Graficele I-MR sunt în limite, astfel graficul I (Individual Value = valoare individuală) și graficul MR (Moving Range = limită mobilă) sunt între limitele trasate cu roșu.. Deviația standard la întreaga măsurare este de 3,83 mm, iar la subgrup este de 4,285 mm, care sunt valori foarte bune. PPM-ul, adică eroarea la parte pe milion, este de asemenea bun cu valoarea 40,94 la subgrup și valoarea de doar 4,87 la toate măsurătorile.

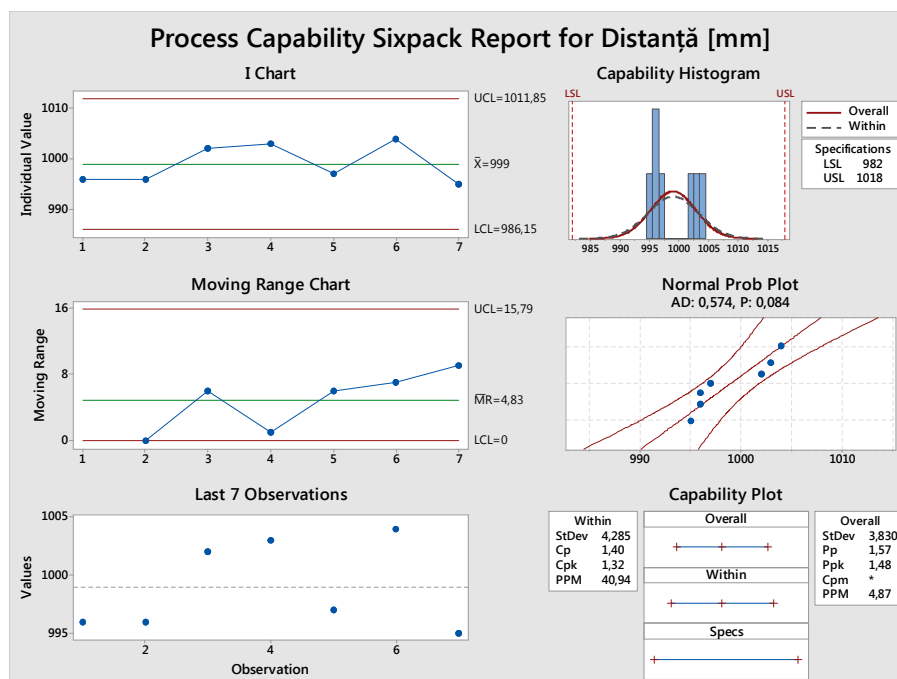


Fig. 5.9 Analiza de capabilitate pentru măsurarea distanței de 1 m folosind camere video la diferite fluxuri luminoase

5.1.3. Analiza măsurării distanțelor la diferite temperaturi de culoare

Tabel 5.3 Măsurarea distanței de 1 m folosind camere video la diferite temperaturi de culoare

Lumină	Temperatură de culoare [K]	Distanță [mm]
Caldă	2700	995
Ambientală	5000	998
Rece	5500	1003

În fig. 5.31 se observă fluctuația erorii delta, măsurată în mm, în funcție de distanța reală, măsurată cu ruleta laser, în m. Se poate vedea o tendință de fluctuație, de valori mai mari, la distanțe mici, sub 1 m, se observă o tendință de fluctuație, de valori proporționale, la distanțe medii, între 1 m și 2 m și se observă o tendință de fluctuație, de valori mai mici, la distanțe mari, între 2 m și 3 m.

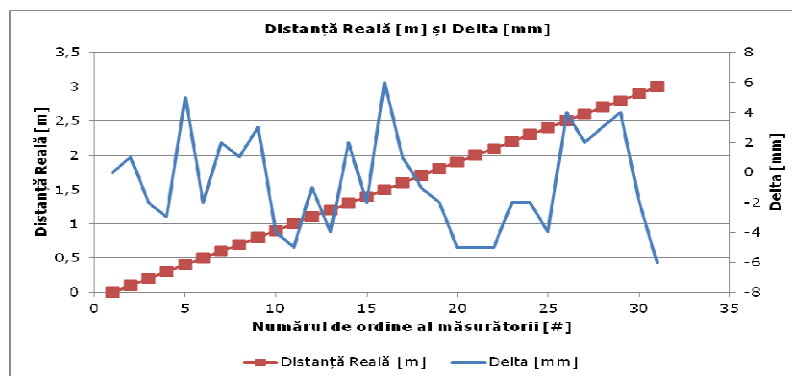


Fig. 5.31 Grafic de prezentare distanța reală [m] și eroarea de măsurare (delta) [mm]

Concluzii finale

Realizarea unui robot autonom este o sarcină dificilă, deoarece operatorul uman trebuie să intervină la anumite sarcini; tehnologia roboților fiind într-un progres continuu, evoluția acestor sisteme inteligente este din ce mai vizibilă.

Ideea de robot autonom în prima instanță se referă la un robot umanoid sau cel puțin un alt tip de robot bioinspirat (insectă). Roboții autonomi sunt întâlniți și la roboții de tip mașină, însă este mai rar întâlnit comportamentul autonom la brațele robotice, care sunt arhiprezente în sistemul industrial, din mai multe motive.

O primă cauză este că în industrie trebuie folosite sisteme bine testate și cunoscute. Altă cauză este că se dorește un sistem simplu. O altă cauză ar fi că de multe ori este destul de bun și un robot preprogramat, deoarece sarcina este repetitivă și nu sunt multe situații de decizie folosind o inteligență artificială.

Trebuie evidențiat faptul că prin algoritmul propus, brațul robotic funcționează total autonom și în timp real, o dată pornit acesta se va deplasa în orice poziție din aria de acoperire a brațului robotic fără a se mai face procesul de învățare a acestuia. Efectorul final tot timpul va urmări permanent punctul țintă. Tot sistemul funcționează în buclă, dacă efectorul final al brațului robotic nu a ajuns exact în punctul țintă (un comportament foarte rar întâlnit) sistemul va face o calibrare în timp real, astfel încât va recalcula distanța din noua poziție, se va face o ajustare fină, astfel încât brațul robotic să ajungă cât mai precis în punctul țintă.

În **capitolul 2** s-au analizat și evaluat performanțele sistemelor cu FPGA destinate controlului brațului robotic. Instalarea unui sistem de operare cu interfață grafică pe un FPGA (o mică performanță tehnică în sine, fiind raportate relativ puține astfel de implementări pe FPGA),

astfel a fost posibilă abordarea controlului la un nivel ridicat din punct de vedere al performanțelor, dar modic raportat la costul echipamentului.

Dificultatea dezvoltării constă în fuziunea unor sisteme complexe și separat în controlul mai multor echipamente pe diverse interfețe de comunicare în același timp. O dificultate suplimentară s-a concretizat în faptul și că aceste sisteme (brațe robotice educaționale și industriale) au fost controlate cu ajutorul plăcilor cu FPGA, pe care au trebuit implementate arhitecturi de procesor.

În **capitolul 3** s-au comparat studiile similare din literatura de specialitate cu modelul de personal original și s-au reliefat diferențele și avantajele metodei proprii.

Am insistat pe deducerea formulelor utilizate ulterior în teză în diferite programe de control al brațelor robotice, pe diferite platforme de implementare (FPGA și PC). Sunt prezentați mai mulți algoritmi de calcul ai distanțelor, în plan și în spațiu.

Deoarece masa roboților utilizați la partea experimentală nu este semnificativă, iar sarcina utilă este redusă, aspect coroborat de controlul prin servomotoare, problemele de dinamică nu au fost luate în considerare, considerând autocalibrarea optică suficientă.

În **capitolul 4** au fost prezentate rezultatele experimentale. Pornind de la controlul cu succes utilizând un robot educațional, matricea implementărilor conține practic toată combinația de resurse, limbaje de programare și sisteme de operare disponibilă în acest moment. Rezultatele au fost validate și pe robotul industrial SCORBOT-ER III.

Din comparația diverselor implementări rezultă următoarele:

- În cazul VHDL, sistemul ocupă foarte puține resurse, însă codul este foarte lung și foarte complicat, astfel necesită un efort sporit din partea celui care face implementarea.
- La implementările în Linux resursele ocupate în timpul execuției sunt relativ mari, însă nu trebuie uitat că implementarea necesită un efort redus față de alte implementări, biblioteca OpenCV fiind cea mai flexibilă, dar consumul de resurse este relativ ridicat.
- Având în vedere că în ziua de azi și sistemele dedicate sunt relativ puternice, folosirea de resurse multe nu ar fi o problemă, mai ales că pe un sistem care controlează un braț robotic nu se mai rulează nimic în paralel.
- Cel mai bun compromis este Python pe sistemul de operare Linux, care este relativ simplu și nici din resurse nu ocupă prea mult.
- Nu trebuie uitat nici faptul că implementările pe Linux sunt de asemenea atractive, deoarece aceste se pot porta cu ușurință pe multe sisteme dedicate.

Testarea sistemului a fost efectuată după ce s-a așezat punctul țintă (dopul verde) în toate cele opt colțuri în aria de acoperire a brațului robotic (stânga sus – jos, dreapta sus – jos, pe ambele părți ale brațului robotic) și brațul robotic a reușit să ajungă autonom în toate aceste colțuri fără a fi învățat punctele țintă, printr-un proces anterior de calibrare; controlul e în timp real, calculul poziției țintă fiind efectuat de algoritmul prezentat (și liniile ajutoare pentru distanțe sunt retrasate în timp real pe imaginea achiziționată la fiecare deplasare al brațului robotic).

În **capitolul 5** au fost prezentate graficele pentru evaluarea sistemului de măsură a distanței implementat.

Măsurarea în sistemul 2D (axe 0Y, 0Z) are precizie mai mare.

Dacă se efectuează măsurări cu două camere, și pe axa 0X, intervin erori, care trebuie analizate și corectate. Sistemul de măsurătoare a distanței folosind camere a fost analizat folosind metode șase sigma (6σ). S-au realizat grafice în Minitab prin care s-a evaluat cât de precis este sistemul de măsurătoare. Rezultatul final a fost un proces de măsurare 4σ , care este foarte bun pentru un proces real. O astfel de precizie este regăsită și în industria automotive în fabrici.

Pe de altă parte a mai fost implementat și un sistem de corecție pentru brațul robotic. Dacă imaginea video finală nu este cea dorită de brațul robotic, algoritmul se reia din noua poziție a acestuia și se face o recalculare pentru poziția finală. Algoritmul se reia în buclă până când se obține o precizie cât mai bună. De multe ori brațul robotic reușește să găsească poziția finală din prima încercare, dar sunt și excepții, când algoritmul este repetat pentru o ajustare fină a poziției finale.

În analiza de șase sigma (6σ) s-au analizat și platformele utilizate și camerele video utilizate. Precum era și de așteptat la platforme nu s-a văzut o diferență clară, deoarece toate platformele foloseau sisteme de operare din aceeași familie de Linux: Debian Linux, respectiv Ubuntu Linux. Totuși trebuie subliniat că dacă este privită flexibilitatea sistemelor, pe ultimul loc ar fi PC-ul, pe locul doi microprocesorul dedicat (μP), deoarece acesta este portabil, iar pe primul loc este microprocesorul creat în FPGA. FPGA-ul pe lângă că e portabil este și reprogramabil, adică se poate realiza și o altă arhitectură de microprocesor pe el, deci este și flexibil.

S-au comparat și camerele video între ele, cel mai bine comportându-se camerele web Logitech C270 și Trust WB-3400T.

Rezumatul principalelor contribuții

1. Datorită noutății temei propuse, studiul stadiul actual al soluțiilor din domeniu prezente în literatura de specialitate a fost strâns legat și de analiza uneltelor și instrumentelor disponibile
 - 1.1. S-au comparat rezultatele similare din literatura de specialitate cu modelul implementat în cadrul acestei teze de doctorat, precizând avantajele sistemului propus.
 - 1.2. S-au analizat, comparat și ierarhizat performanțele celor trei plăci cu FPGA disponibile: ATLYS, ZYBO și ZedBoard și s-a evaluat posibilitatea utilizării lor în pentru controlul brațului robotic.
 - 1.3. S-au comparat implementările în diferite limbaje de programare (18 implementări) din punct de vedere al lungimii codului, al memoriei ocupate, al utilizării UCP și timpului de execuție.
 - 1.4. S-au comparat camerele video între ele și separat cu cele utilizate în alte cercetări similare, evidențiind avantajele contextuale ale camerelor IP.
2. **Metoda de calcul și algoritmul de control al mișcării sunt originale.**
 - 2.1. Aceste formule au fost gândite pentru a obține anumite distanțe în spațiu având ca **informație doar anumite coordonate a unor buline colorate folosite ca ghidaj** pentru recunoașterea articulațiilor brațului robotic.
 - 2.2. Toate formulele au fost simplificate și puse într-o formă implementabilă în cod sursă calculator.
 - 2.3. S-a prezentat și ideea ca **pentru mișcare în spațiu (3D) să se folosească dopuri colorate**, deoarece când brațul robotic se întoarce perpendicular pe planul camerelor să se mai vadă o porțiune din aceste marcaje colorate puse la articulațiile robotului.
 - 2.4. Deoarece un braț robotic industrial nu are în general mai mult de 7 articulații, metodele descrise (2D și 3D) se pot aplica cu succes.
 - 2.5. A fost dezvoltat **algoritmul de trasare a linilor și cercurilor ajutatoare pentru calculul distanțelor** pentru brațul robotic, având ca date de intrare doar coordonate obținute din poziția în spațiu a bulinelor colorate detectate.
 - 2.6. Am demonstrat geneza tuturor formulelor și le-am validat prin mișcarea controlată a robotului.
3. **S-a realizat un sistem complet de control al brațelor robotice cu ajutorul camerelor video.** Sistemul dezvoltat este autonom, nu necesită o programare anterioară, obiectul țintă este recunoscut de camerele video, apoi sistemul calculează în timp real pozițiile brațului robotic până la obiectul țintă și este controlată deplasarea brațului robotic. Sistemul nu necesită programarea unor poziții intermediare sau a unor puncte în spațiu.
 - 3.1. Am implementat **o funcție de autocalibrare și una de corecție a erorilor**. Mișcarea se execută în buclă, fără intervenția operatorului uman (exceptând pornirea sa).
4. Pentru rularea pe FPGA a algoritmului propus a fost necesară **implementarea unor noi arhitecturi pe aceste circuite reconfigurabile:**
 - 4.1. S-a implementat arhitectura microprocesorului OpenRISC-32 pe placa ATLYS cu FPGA și s-a instalat Linux în linie de comandă pe memoria FLASH a acestei plăci.
 - 4.2. S-a fuzionat arhitectura ARM Cortex – A9 Dual-core cu FPGA-ul Atrix-7 de pe SoC-ul Zynq-7000 de pe plăcile ZYBO (650 MHz) și ZedBoard (între 866 MHz și 1000 MHz) și s-a instalat Ubuntu Linux 12.04 LTS și OpenCV pe card-urile de memorie (microSD respectiv SD) care vin cu aceste plăci cu FPGA.
5. S-a dezvoltat un **model tridimensional MATLAB pentru brațul robotic.**

6. **Studiul rezultatelor obținute a fost realizat utilizând metode șase sigma (6σ)**, cum ar fi: Gage R&R, analiză de regresie, analiză de capabilitate, analiză ANOVA (analiză de variație), grafice I-MR, grafice Xbar & R, histograme și grafice pe probabilitate normală.
 - 6.1. Analiza șase sigma (6σ) a fost completată de grafice Minitab și Excel.
 - 6.2. A fost analizată influența condițiilor de iluminare, a temperaturii de culoare și a modului distanței până la obiectul țintă.

Activitatea de cercetare desfășurată pe perioada doctoratului s-a materializat prin **14** articole indexate **ISI**, din care sunt **10** sunt direct legate de această teză de doctorat și **3** lucrări **BDI** de asemenea direct legate de subiectul tezei. Dintre cele 14 articole ISI, unul este publicat într-o **revistă** indexată **ISI** cu **factor de impact 0.642**. Publicațiile personale au în total **22 citări** din care **4** sunt indexate **BDI**, citările publicațiilor personale au un **h-index** de **2** (excluzând autocitările).

Statistica tezei: pe parcursul a 148 pagini, au fost realizate 123 figuri (dintre care 118 originale), 13 tabele (10 originale), 156 titluri bibliografice, 3 anexe și un CD cu programele dezvoltate.

Lista de publicații ISI personale legate de teza de doctorat

Revistă indexată ISI

- [1] **R. Szabó**, A. Gontean, "Robotic Arm Control Algorithm Based on Stereo Vision Using RoboRealm Vision," AECE 2015 – Advances in Electrical and Computer Engineering, vol. 15, nr. 2, mai, 2015, pp. 65-74, factor de impact 0.642.

Conferințe indexate ISI

- [2] **R. Szabó**, A. Gontean, "Robotic Arm Detection in the 2D Space," SIITME 2014 – 20th International Symposium for Design and Technology in Electronic Packaging, București, România, 23-26 octombrie, 2014, pp. 215-220.
- [3] **R. Szabó**, A. Gontean, "Robotic Arm Movement Using Color Detection with FPGA Vision," SIITME 2014 – 20th International Symposium for Design and Technology in Electronic Packaging, București, România, 23-26 octombrie, 2014, pp. 117-122.
- [4] **R. Szabó**, A. Gontean, "Pong Game on FPGA with CRT or LCD Display and Push Button Controls," FedCSIS 2014 – Federated Conference on Computer Science and Information Systems, Varșovia, Polonia, 7-10 septembrie, 2014, pp. 735-740.
- [5] **R. Szabó**, A. Gontean, "Controlling a Robotic Arm in the 3D Space with Stereo Vision," TELFOR 2013 – 21th Telecommunications Forum, Belgrad, Serbia, 26-28 noiembrie, 2013, pp. 916-919.
- [6] **R. Szabó**, A. Gontean, "Remotely Commanding the Lynxmotion AL5 Type Robotic Arms," TELFOR 2013 – 21th Telecommunications Forum, Belgrad, Serbia, 26-28 noiembrie, 2013, pp. 889-892.
- [7] **R. Szabó**, A. Gontean, "Creating a Serial Driver Chip for Commanding Robotic Arms," FedCSIS 2013 – Federated Conference on Computer Science and Information Systems, Cracovia, Polonia, 8-11, septembrie 2013, pp. 671-674.
- [8] **R. Szabó**, A. Gontean, "Creating a Programming Language for the AL5 Type Robotic Arms," TSP 2013 – 36th International Conference on Telecommunications and Signal Processing, Roma, Italia, 2-4 iulie, 2013, pp. 62-65.
- [9] **R. Szabó**, I. Lie, "New Automatic SPI Decoding Algorithm," ISETC 2012 – International Symposium on Electronics and Telecommunications, Tenth Edition, Timișoara, România, 15-16 noiembrie, 2012, pp. 327-330.
- [10] **R. Szabó**, I. Lie, "Automated Colored Object Sorting Application for Robotic Arms," ISETC 2012 – International Symposium on Electronics and Telecommunications, Tenth Edition, Timișoara, România, 15-16 noiembrie, 2012, pp. 95-98.

Citată de:

- [10.1] T. L. Nguyen, R. Boukezzoula, D. Coquin, E. Benoit, S. Perrin, "Interaction between humans, NAO robot and multiple cameras for colored objects recognition using information fusion," HSI 2015 – 8th International Conference on Human System Interactions, Varșovia, Polonia, 25-27 iunie, 2015, pp. 322-328, **indexată BDI**.
- [10.2] N. R. Vange, A. V. Nar, D. B. Surve, A. P. Trimukhe, M. M. Patil, R. A. Patil, "Pic-O-Place Robotic Arm Object Sorting Robotic Arm Based on Color Sensing Mechanism," International Journal of Emerging Technology and Advanced Engineering, vol. 5, nr. 5, mai 2015, pp. 141-145.

- [10.3] P. Prashanth, M. P. Saravanan, V. Nandagopal, "Vision Based Object's Dimension Identification To Sort Exact Material," IOSR-JEEE 2015 – IOSR Journal of Electrical and Electronics Engineering, vol. 10, nr. 1, ver. III, ianuarie-februarie 2015, pp. 11-15.
- [10.4] N. Rai, B. Rai, "Active Vision Approach for Controlling Educational Robotic Arm with Autonomous Object Manipulation," Journal of Advancements in Robotics, vol. 1, nr. 3, 2015, pp. 1-10.
- [10.5] V. Pereira, V. A. Fernandes, J. Sequeira, "Low cost object sorting robotic arm using Raspberry Pi," GHTEC-SAS 2014 – Global Humanitarian Technology Conference – South Asia Satellite, Trivandrum, India, 26-27 septembrie, 2014, pp. 1-6, **indexată BDI**.
- [10.6] C. C. Mouli, P. Jyothi, K. P. J. Pradeep, K. N. Raju, Design, "Implementation and Comparative Study of Supervised Classification Algorithms for Object Sorting," IOSR Journal of Electrical and Electronics Engineering, vol. 9, nr. 4, ver. IV, iulie-august 2014, pp. 51-60.
- [10.7] C. C. Mouli, P. Jyothi, K. N. Raju, "Design and Implementation of an Intelligent Virtual Instrumentation System for Vision based Object Sorting," International Journal of Advanced Scientific and Technical Research, vol. 4, nr. 4, iulie-august 2014, pp. 545-556.
- [10.8] N. I. Giannoccaro, L. Spedicato, A. Lay-Ekuakille, "A robotic arm to sort different types of ball bearings from the knowledge discovered by size measurements of image regions and RFID support," International Journal on Smart Sensing and Intelligent Systems, vol. 7, nr. 2, iunie 2014, pp. 674-700.
- [10.9] V. S. Zilpe, S. Pathan, P. Choudhary, "A Smart Robotic Arm for Automatic Sorting Of Objects With Different RFID Tags," Discovery, vol. 18, nr. 54, 10 mai 2014, pp. 111-114.
- [10.10] C.C. Mouli, P. Jyothi, K.N. Raju, "Comparative Study of Supervised Classification Algorithms for WOSRAS," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 3, nr. 2, februarie 2014, pp. 7188-7193.
- [10.11] C. C. Mouli, K. N. Raju, "A Review on Wireless Embedded System for Vision Guided Robot Arm for Object Sorting," International Journal of Scientific & Engineering Research, vol. 4, nr. 7, iulie 2013, pp. 33-38.
- [10.12] N. I. Giannoccaro, L. Spedicato, A. Lay-Ekuakille, "A Smart Robotic Arm For Automatic Sorting of Objects With Different Tags," 4th Imeko TC19 Symposium on Environmental Instrumentation and Measurements Protecting Environment, Climate Changes and Pollution Control, Lecce, Italia, 3-4 iunie 2013, pp. 95-98.

Bibliografie

- [11] Wen-Chung Chang, Chih-Wei Cho, "Automatic Mobile Robotic Manipulation with Active Eye-to-Hand Binocular Vision," IECON 2007 – 33rd Annual Conference of the IEEE Industrial Electronics Society, Taipei, Taiwan, 5-8 noiembrie 2007, 2007, pp. 2944-2949.
- [12] P. C. Nunnally, J. M. Weiss, "An inexpensive robot arm for computer vision applications," Southeastcon '89 – IEEE Proceedings of the Energy and Information Technologies in the Southeast, Columbia, SC, SUA, vol. 1, 9-12 aprilie, 1989, pp. 1-6.
- [13] Manual de utilizare SCORBOT-ER III, <http://www.theoldrobots.com/book45/ER3-Manual.pdf>, vizitat ultima oară 20 octombrie, 2014.
- [14] Calcul distanță stereo, <http://www.alexandria.nu/ai/blog/entry.asp?E=32>, vizitat ultima oară 10 decembrie, 2014.
- [15] Stereo triangulare, <http://www.dis.uniroma1.it/~iocchi/stereo/triang.html>, vizitat ultima oară 9 decembrie, 2014.
- [16] Industrie 4.0, http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Material_fuer_Sonderseiten/Industrie_4.0/Final_report__Industrie_4.0_accessible.pdf, vizitat ultima oară 5 februarie, 2015.
- [17] Calcul PPI, <https://www.sven.de/dpi/>, vizitat ultima oară 20 februarie, 2015.
- [18] Model 3D braț robotic, <http://www.3dcadbrowser.com/download.aspx?3dmodel=17132>, vizitat ultima oară 30 septembrie, 2015.
- [19] Desen braț robotic, <https://www.cdli.ca/courses/ep/ds2/t06/02knowledge-skills/act-01a.htm>, vizitat ultima oară 30 septembrie, 2015.
- [20] M. Seelinger, E. Gonzalez-Galvan, M. Robinson, S. Skaar, "Towards a robotic plasma spraying operation using vision," IEEE Robotics & Automation Magazine, vol. 5, nr. 4, 1998, pp. 33-38, 49.
- [21] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, F. Reyes, "Stable visual servoing of camera-in-hand robotic systems," IEEE/ASME Transactions on Mechatronics, vol. 5, nr. 1, 2000, pp. 39-48.
- [22] V. Lippiello, F. Ruggiero, B. Siciliano, L. Villani, "Visual Grasp Planning for Unknown Objects Using a Multifingered Robotic Hand", IEEE/ASME Transactions on Mechatronics, vol. 18, nr. 3, 2013, pp. 1050-1059.
- [23] M. Kazemi, K. K. Gupta, M. Mehrandezh, "Randomized Kinodynamic Planning for Robust Visual Servoing", IEEE Transactions on Robotics, vol. 29, nr. 5, 2013, pp. 1197-1211.
- [24] R. T. Fomena, O. Tahri, F. Chaumette, "Distance-Based and Orientation-Based Visual Servoing From Three Points", IEEE Transactions on Robotics, vol. 27, nr. 2, 2011, pp. 256-267.
- [25] L. Behera, N. Kirubanandan, "A hybrid neural control scheme for visual-motor coordination", IEEE Control Systems, vol. 19, nr. 4, 1999, pp. 34-41.
- [26] J. J. Heuring, D. W. Murray, "Modeling and copying human head movements", IEEE Transactions on Robotics and Automation, vol. 15, nr. 6, 1999, pp. 1095-1108.
- [27] F. Chaumette, S. Hutchinson, "Visual servo control. II. Advanced approaches [Tutorial]", IEEE Robotics & Automation Magazine, vol. 14, nr. 1, 2007, pp. 109-118.

- [28] In-Won Park, Bum-Joo Lee, Se-Hyoung Cho, Young-Dae Hong, Jong-Hwan Kim, "Laser-Based Kinematic Calibration of Robot Manipulator Using Differential Kinematics", *IEEE/ASME Transactions on Mechatronics*, vol. 17, nr. 6, 2012, pp. 1059-1067.
- [29] Zhang Hanqi, "Hand/eye calibration for electronic assembly robots," *IEEE Transactions on Robotics and Automation*, vol. 14, nr. 4, 1998, pp. 612-616.
- [30] Zhuang Hanqi, Wu Wen-Chiang, Zvi S. Roth, "Camera assisted calibration of SCARA arms," *IEEE Robotics & Automation Magazine*, vol. 3, nr. 4, 1996, pp.46-53.
- [31] R. Kelly, "Robust asymptotically stable visual servoing of planar robots," *IEEE Transactions on Robotics and Automation*, vol. 12, nr. 5, 1996, pp. 759-766.
- [32] F. C. A. Groen, G. A. den Boer, A. van Inge, R. Stam, "A chess-playing robot: lab course in robot sensor integration," *IEEE Transactions on Instrumentation and Measurement*, vol. 41, nr. 6, 1992, pp. 911-914.
- [33] Aghili Farhad, "A Prediction and Motion-Planning Scheme for Visually Guided Robotic Capturing of Free-Floating Tumbling Objects With Uncertain Dynamics," *IEEE Transactions on Robotics*, vol. 28, nr. 3, 2012, pp. 634-649.
- [34] P. K. Allen, A. Timcenko, B. Yoshimi, P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *IEEE Transactions on Robotics and Automation*, vol. 9, nr. 2, 1993, pp. 152-165.
- [35] G. Silveira, "On Intensity-Based Nonmetric Visual Servoing," *IEEE Transactions on Robotics*, vol. 30, nr. 4, 2014, pp. 1019-1026.
- [36] Jiang Hairong, B. S. Duerstock, J. P. Wachs, "A Machine Vision-Based Gestural Interface for People With Upper Extremity Physical Impairments," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, nr. 5, 2014, pp. 630-641.
- [37] D. Kruse, J. T. Wen, R. J. Radke, "A Sensor-Based Dual-Arm Tele-Robotic System," *IEEE Transactions on Automation Science and Engineering*, vol. 12, nr. 1, 2015, pp. 4-18.
- [38] N. M. Garcia-Aracil, J. M. Azorin, J. M. Sabater, C. Perez Vidal, R. Saltaren Pazmino, "Visual Control of robots with changes of visibility in image features," *IEEE Latin America Transactions, (Revista IEEE America Latina)*, vol. 4, nr. 1, 2006, pp. 27-33.
- [39] D. Caldwell, A. Wardle, O. Kocak, M. Goodwin, "Telepresence feedback and input systems for a twin armed mobile robot," *IEEE Robotics & Automation Magazine*, vol. 3, nr. 3, 1996, pp. 29-38.
- [40] H. Hady-Abdelkader, Y. Mezouar, P. Martinet, F. Chaumette, "Catadioptric Visual Servoing From 3-D Straight Lines," *IEEE Transactions on Robotics*, vol. 24, nr. 3, 2008, pp. 652-665.
- [41] Shih-Jer Huang, Jian-Cheng Huang, "Vision guided dual arms robotic system with DSP and FPGA integrated system structure," *Journal of Mechanical Science and Technology*, vol. 25, nr. 8, 2011, pp. 2067-2076.