

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

Teză destinată obținerii
titlului științific de doctor inginer
la
Universitatea Politehnica Timișoara
în domeniul INGINERIE ELECTRONICĂ ȘI
TELECOMUNICAȚII
de către

dipl.-ing. Roland Szabó

Conducător științific: prof.univ.dr.ing. Aurel Gontean
Referenți științifici: prof.univ.dr.ing. Paul Svasta
prof.univ.dr.ing. Dan Pitică
prof.univ.dr.ing. Radu Vasiu

Ziua susținerii tezei: 02.10.2015

Seriile Teze de doctorat ale UPT sunt:

- | | |
|---|--|
| 1. Automatică | 9. Inginerie Mecanică |
| 2. Chimie | 10. Știința Calculatoarelor |
| 3. Energetică | 11. Știința și Ingineria Materialelor |
| 4. Ingineria Chimică | 12. Ingineria sistemelor |
| 5. Inginerie Civilă | 13. Inginerie energetică |
| 6. Inginerie Electrică | 14. Calculatoare și tehnologia informației |
| 7. Inginerie Electronică și Telecomunicații | 15. Ingineria materialelor |
| 8. Inginerie Industrială | 16. Inginerie și Management |

Universitatea Politehnica Timișoara a inițiat seriile de mai sus în scopul diseminării expertizei, cunoștințelor și rezultatelor cercetărilor întreprinse în cadrul școlii doctorale a universității. Seriile conțin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susținute în universitate începând cu 1 octombrie 2006.

Copyright © Editura Politehnica – Timișoara, 2015

Această publicație este supusă prevederilor legii dreptului de autor. Multiplicarea acestei publicații, în mod integral sau în parte, traducerea, tipărirea, reutilizarea ilustrațiilor, expunerea, radiodifuzarea, reproducerea pe microfilme sau în orice altă formă este permisă numai cu respectarea prevederilor Legii române a dreptului de autor în vigoare și permisiunea pentru utilizare obținută în scris din partea Universității Politehnica Timișoara. Toate încălcările acestor drepturi vor fi penalizate potrivit Legii române a drepturilor de autor.

România, 300159 Timișoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Cuvânt înainte

Teza de doctorat a fost elaborată pe parcursul activității mele în cadrul Departamentului de Electronică Aplicată din Facultatea de Electronică și Telecomunicații, Universitatea Politehnică Timișoara.

Mulțumiri deosebite se cuvin conducătorului de doctorat, prof.dr.ing. Aurel Gontean, care m-a îndrumat de-a lungul pregătirii și elaborării tezei, cu răbdare și pasiune. Tot el m-a sprijinit în achiziționarea unor echipamente complexe necesare activității de cercetare desfășurate în ultimii ani.

Sunt recunoscător domnilor profesori din comisia mea de îndrumare: prof.univ.dr.ing. Vasile Gui, care m-a ajutat la partea de prelucrare de imagini, prof.univ.dr.ing. Dorina Isar, care m-a ajutat la partea de interfețe de comunicație și conf.univ.dr.ing. Ioan Jiveț, care m-a ajutat la partea de programare FPGA și mi-a dat idei de recunoaștere a brațelor robotice cu una sau mai multe camere.

Mulțumiri deosebite se cuvin și lui sl.dr.ing. Aurel Diaconu, care mi-a facilitat accesul la robotul Scorbot-ER III, atât în timpul săptămânii cât și în weekend-uri.

Aș dori să îi mulțumesc și domnului tehnician Cornel Crista, care m-a ajutat să realizez o multitudine de stative pentru camere, cârlige și alte subansamble mecanice pentru partea experimentală a tezei mele de doctorat.

De asupra de toate aș dori să îi mulțumesc mamei mele care m-a sprijinit în toți anii de facultate și doctorat. Fără marile sacrificii ale ei, nu aș fi ajuns aici. Pentru aceasta, toată viața îi voi fi recunoscător.

Fără ajutorul celor mai sus menționați, realizarea lucrării mele de doctorat nu ar fi fost posibilă.

Timișoara, septembrie 2015

dipl.-ing. Roland Szabó

Pentru mama mea

Szabó, Roland, dipl.-ing.

Contribuții la dezvoltarea metodelor și algoritmilor pentru calculul poziției brațelor robotice în spațiu folosind recunoaștere de culori. Implementare în FPGA

Teze de doctorat ale UPT, Seria 7, Nr. 81, Editura Politehnica, 2015, 148 pagini, 123 figuri, 13 tabele.

ISSN: 1842-7014

ISBN: 978-606-35-0009-1

Cuvinte cheie:

braț robotic, calculul de poziții, filtrare de culori, FPGA, interfețe de comunicație, Linux, detecție de marcheri, sisteme dedicate.

Rezumat:

Scopul acestei teze este de a dezvolta și implementa o metodă și un algoritm de calcul al poziției brațului robotic. Algoritmul a fost conceput pe durata tezei de doctorat din dorința de a simplifica actualele metode de calcul (cinematica directă și inversă), care presupun calcule matriciale complexe. Algoritmul propus se bazează doar pe detecția de marcheri și pe un sistem de ecuații simple pentru a afla poziția efectorului final al brațului robotic. Sistemul utilizează camere stereo pentru recunoașterea de culori și calculul distanței, prelucrând datele și calculând poziția brațului robotic în spațiu, astfel încât brațul robotic să poată fi controlat în modul dorit.

CUPRINS

Cuprins.....	5
Notații, abrevieri, acronime (în engleză).....	7
Denumiri.....	8
Lista de tabele.....	10
Lista de figuri.....	11
I. INTRODUCERE.....	16
1.1. Poziționarea lucrării în contextul actual.....	16
1.2. Prezentare generală.....	17
1.3. Structura tezei.....	18
II. RESURSE. METODE.....	20
2.1. Sisteme cu microprocesor.....	20
2.1.1. Placa dedicată Raspberry PI.....	20
2.2. Sisteme reconfigurabile.....	20
2.2.1. Comparație între plăci de dezvoltare cu FPGA.....	20
2.3. Brațe robotice.....	25
2.3.1. Prezentarea generală a brațelor robotice utilizate.....	25
2.3.2. Brațele robotice Lynxmotion AL5A și AL5B.....	26
2.3.3. Brațul robotic SCORBOT-ER III.....	28
2.4. Concluzii și contribuții.....	31
III. ALGORITM ȘI METODĂ.....	32
3.1. Tema de cercetare științifică.....	32
3.1.1. Prezentarea generală a temei de cercetare științifică....	32
3.1.2. Justificarea alegerii temei de cercetare.....	32
3.1.3. Stadiul actual al cercetării în tematica tezei propuse...	33
3.2. Metode de calcul.....	37
3.2.1. Prezentarea algoritmului propus.....	37
3.2.2. Prezentarea algoritmilor și formulelor alternative.....	43
3.3. Concluzii și contribuții.....	48
IV. REZULTATE EXPERIMENTALE.....	49
4.1. Controlul poziției brațului robotic în plan (2D).....	49
4.1.1. Utilizarea bulinelor albastre la articulații.....	49
4.1.2. Utilizarea bulinelor multicolore la articulații.....	54
4.2. Controlul poziției brațului robotic în spațiu (3D).....	57
4.2.1. Implementarea în LabWindows/CVI și RoboRealm.....	57
4.2.2. Implementarea în LabVIEW și NI Vision Development Module.....	61
4.2.3. Implementarea în LabWindows/CVI și NI Vision Development Module.....	63
4.2.4. Implementarea în C și OpenCV sub Linux.....	66
4.3. Controlul robotului industrial SCORBOT-ER III.....	71
4.4. Analiza comparativă a performanțelor implementărilor.....	83
4.5. Concluzii și contribuții.....	88
V. ANALIZA ȘI COMPARAȚIA REZULTATELOR.....	90
5.1. Analiza măsurării distanțelor folosind camere video.....	90
5.1.1. Analiza măsurării de distanțe la diferitele	

	echipamente utilizate.....	90
	5.1.2. Analiza măsurării distanțelor la diferite fluxuri luminoase.....	97
	5.1.3. Analiza măsurării distanțelor la diferite temperaturi de culoare.....	101
	5.1.4. Analiza măsurării distanțelor în diferite poziții.....	105
	5.2. Comparația metodei prezentate cu alte metode din literatură...	116
	5.3. Concluzii și contribuții.....	118
VI.	CONCLUZII FINALE ȘI CONTRIBUȚII.....	120
	6.1. Concluzii finale.....	120
	6.2. Rezumatul principalelor contribuții.....	122
	6.3. Direcții viitoare de cercetare.....	123
	6.4. Lista publicațiilor persoanele.....	124
	Bibliografie.....	127
	Anexa A.....	140
	Anexa B.....	146
	Anexa C.....	147

NOTAȚII, ABREVIERI, ACRONIME (ÎN ENGLEZĂ)

AES	Advanced Encryption Standard
DCM	Digital Clock Manager
EUI-48/64	Extended Unique Identifier – 48/64 bit
FMC	FPGA Mezzanine Card
I ² S	Integrated Interchip Sound
MSPS	Million Samples Per Second
OTG	On-The-Go
PHY	Physical Layer
PL	Programmable Logic
PS	Processor System
PTP	Point to Point
RIOS	Robotic arm Interactive Operating System
RSA	Ron Rivest, Adi Shamir and Leonard Adleman crypting
SCPI	Standard Commands for Programmable Instruments
SHA	Secure Hash Algorithm
SIPO	Serial Input Parallel Output
SoC	System on a Chip
SSC-32	Serial Servo Controller on 32 bits
VHDCI	Very-high-density Cable Interconnect
VISA	Virtual Instrument Software Architecture
XADC	Xilinx Analog to Digital Converter

DENUMIRI

P_n	punctele utilizate pentru desenul aplicat pe imaginea achiziționată de camere pentru calculul de distanțe, n este numărul punctului, unde $n = \{0, 1, 2, 3, 4, 5, 6, 7\}$, T este punctul țintă la prima metodă
$\Delta\omega$	diferența de unghi al poziției motorului în valori robot calculate în pulsuri la prima metodă
x_n, y_n	coordonatele x și y ale unui punct, n este numărul punctului, unde $n = \{0, 1, 2, 3, 4, 5, 6, 7\}$, T este punctul țintă la prima metodă
$\Delta x, \Delta y$	diferența de coordonate la prima metodă
$\ l\ $	lungimea unui vector la prima metodă
x, y	coordonate carteziene la prima metodă
$\tilde{x}_{ort}, \tilde{y}_{ort}$	coordonatele vectorului ortogonal la prima metodă
$\Delta x_{0x}, \Delta x_{0y}$	diferența de coordonate x de pe axele OX și OY la prima metodă
$\Delta y_{0x}, \Delta y_{0y}$	diferența de coordonate y de pe axele OX și OY la prima metodă
m_α, m_β	pantele unghiurilor α și β ale paralelogramului sistemului de coordonate desenat pe imaginea achiziționată de camere la prima metodă
b	membrul liber al ecuației dreptei de forma $y = mx + b$ la prima metodă
$\hat{\alpha}, \hat{\beta}$	unghiurile din triunghiurile pentru calculul distanței stereo
x_{nR}, x_{nL}	coordonata x al punctului n de la camera dreaptă ($R = \text{right}$) respectiv de la camera stângă ($L = \text{left}$), unde $n = \{0, 1, 2, 3, 4, 5, 6, 7\}$, T este punctul țintă pentru calculul distanței stereo
X, Y, Z	coordonatele spațiului tridimensional la calculul distanței prin triangulare
d	distanța între cele două camere, camera dreaptă (R), camera stângă (L) la calculul distanței prin triangulare
f	distanța focală la calculul distanței prin triangulare
θ	unghiul de rotație în jurul axei OY la calculul distanței prin triangulare
x_1, x_2	intersecția dreptei distanței focale f cu LZ_0 respectiv cu RZ_0 la calculul distanței prin triangulare
P	punctul țintă la calculul distanței prin triangulare
Z_0	intersecția dreptei care trece prin punctul R (camera dreaptă) cu axa OZ la calculul distanței prin triangulare
P, Q, R	denumirea coordonatelor din spațiu la metoda alternativă

\vec{PQ}, \vec{PR}	vectorii coordonatelor din spațiu la metoda alternativă
i, j, k	vectori pentru formarea determinantului pentru calculul produsului vectorial la metoda alternativă
u, v, w	parametrii din ecuația planului în forma parametrică la metoda alternativă
a, b, c, d	membrii din ecuația planului de forma $ax + by + cz + d = 0$ la metoda alternativă

LISTA DE TABELE

Tabel 2.1	Familii de FPGA-uri, plăci de dezvoltare, modele.....	20
Tabel 2.2	Comparație între plăcile de dezvoltare cu FPGA.....	21
Tabel 2.3	Resurse utilizate din FPGA pentru crearea SoC-ului Zynq-7000 (în engleză, rezumat).....	24
Tabel 2.4	Specificațiile brațului robotic SCORBOT-ER III, preluat după [31]....	29
Tabel 4.1	Resurse utilizate din placa FPGA ATLYS pentru crearea sistemului de control al brațului robotic cu recunoașterea bulinelor albastre de la articulații (în engleză, rezumat).....	50
Tabel 4.2	Comparația implementărilor sistemului de control al brațului robotic folosind camere video.....	83
Tabel 5.1	Studiul Gage R&R pentru măsurarea distanței de 1 m cu șase tipuri de camere video și pe trei tipuri de platforme.....	90
Tabel 5.2	Măsurarea distanței de 1 m folosind camere video la diferite fluxuri luminoase.....	97
Tabel 5.3	Măsurarea distanței de 1 m folosind camere video la diferite temperaturi de culoare.....	101
Tabel 5.4	Măsurarea distanței folosind camere video la diferite distanțe [mm] cu evidențierea erorii de măsurare [mm].....	105
Tabel 5.5	Comparația între metoda prezentată și metode similare publicate în diferite articole.....	116
Tabel 5.6	Erorile de distanță pentru cele două spoturi de laser de pe perete după procesul de calibrare (experiment), preluat după [46].....	117
Tabel 5.7	Performanțele la diferiți algoritmi, preluat după [43].....	118

LISTA DE FIGURI

Fig. 1.1	Cele patru etape ale revoluției industriale, preluat după [33].....	17
Fig. 2.1	Grafic de comparație între plăcile de dezvoltare.....	23
Fig. 2.2	Arhitectura SoC-ul Zynq-7000 creat în Xilinx Platform Studio.....	23
Fig. 2.3	Resurse folosite din SoC Zync 7000 care este configurat pentru a rula sistem de operare Linux.....	25
Fig. 2.4	Brațul robotic Lynxmotion AL5B.....	26
Fig. 2.5	Captarea comenzilor SCPI cu ajutorul programului HHD Free Serial Port Monitor.....	27
Fig. 2.6	Brațul robotic SCORBOT-ER III.....	29
Fig. 3.1	Localizarea brațului robotic în spațiu utilizând buline colorate, preluat după [37], apoi modificat.....	33
Fig. 3.2	Hârtii colorate aplicate în puncte cheie ale brațului robotic Lynxmotion AL5A.....	34
Fig. 3.3	Dopuri colorate aplicate în puncte cheie ale brațului robotic Lynxmotion AL5B (fiind tridimensional sunt vizibile și când brațul robotic este rotit în jurul bazei).....	34
Fig. 3.4	Dopuri mai mari colorate aplicate în puncte cheie și la punctul țintă ale brațului robotic SCORBOT-ER III (dopul este vizibil și când brațul robotic este perpendicular pe camerele video).....	35
Fig. 3.5	Desenele ajutătoare pe modelul 3D al brațului robotic, preluat după [36], apoi modificat.....	37
Fig. 3.6	Desene ajutătoare pentru calculul de distanțe 2D imprimate pe imaginea video.....	38
Fig. 3.7	Calculul distanței stereo, preluat după [32], apoi modificat.....	41
Fig. 3.8	Calculul distanței stereo prin triangulare, preluat după [33].....	44
Fig. 3.9	Desene ajutătoare pentru calculul de distanțe 3D imprimate pe imaginea video.....	45
Fig. 4.1	Diagrama bloc pentru sistemul de control al brațului robotic cu FPGA ALTY5 prin recunoașterea bulinelor albastre de la articulații – programare în VHDL.....	50
Fig. 4.2	Grafic de utilizare a resurselor pentru sistemul de control al brațului robotic cu FPGA ALTY5 prin recunoașterea bulinelor albastre de la articulații.....	52
Fig. 4.3	Imagine inițială captată de brațul robotic cu FPGA ATLY5 – programare în VHDL.....	52
Fig. 4.4	Recunoaștere culorii albastre cu ajutorul FPGA ATLY5 – programare în VHDL.....	52
Fig. 4.5	Diagrama bloc pentru experimentul de detectare a brațului robotic în planul 2D – realizat cu LabWindows/CVI și RoboRealm cu recunoașterea culorii albastre.....	53
Fig. 4.6	Imaginea inițială cu brațul robotic cu buline de culoare albastră aplicate la încheieturi.....	53
Fig. 4.7	Trasarea scheletului robotului de asupra imaginii inițiale prin recunoașterea articulațiilor cu ajutorul algoritmului de recunoaștere de culori.....	53
Fig. 4.8	Diagrama bloc pentru sistemul de achiziție de imagine și control	

12 Lista de figuri

	braț robotic cu placa de FPGA ATLYS – programare în C sub Linux.....	54
Fig. 4.9	Imagine captată cu placa FPGA ATLYS pe care rulează sistem de operare Linux.....	55
Fig. 4.10	Interfața grafică în LabWindows/CVI a programului de conversie a comenzilor UART de la microUSB la DB9.....	55
Fig. 4.11	Diagrama bloc pentru sistemul de detectare al brațului robotic în planul 2D – realizat cu LabWindows/CVI și RoboRealm cu recunoaștere de culori multiple.....	56
Fig. 4.12	Imaginea inițială a brațului robotic cu indicarea punctului țintă cu mouse-ul.....	56
Fig. 4.13	Trasarea liniilor ajutătoare pentru calculul distanțelor pentru ca efectorul final al brațului robotic să ajungă în punctul țintă.....	56
Fig. 4.14	Imaginea inițială a brațului robotic filmată de camera din partea stângă.....	57
Fig. 4.15	Imaginea inițială a brațului robotic filmată de camera din partea dreaptă.....	57
Fig. 4.16	Imaginile inițiale suprapuse în MATLAB pentru ochelari 3D folosind culorile turcoaz și roșu.....	58
Fig. 4.17	Harta de adâncime generată de MATLAB.....	58
Fig. 4.18	Identificarea de blocuri în MATLAB.....	58
Fig. 4.19	Identificarea de blocuri piramidale de nivelul 4 în MATLAB.....	58
Fig. 4.20	Identificarea de blocuri piramidale de nivelul 3 în MATLAB.....	58
Fig. 4.21	Identificarea de blocuri piramidale prin programare dinamică în MATLAB.....	58
Fig. 4.22	Imaginea generată 3D în MATLAB.....	59
Fig. 4.23	Achiziția imaginii stereo cu placa FPGA ATLYS – programare în VHDL.....	59
Fig. 4.24	Imaginea stereo a brațului robotic pe monitor captat cu placa FPGA ATLYS și camera stereo VmodCAM.....	59
Fig. 4.25	Imaginea cu trasarea liniilor ajutătoare de calcul a poziției brațului robotic în 3D cu două algoritme combinate: calculul distanței în planul 2D și calculul distanței stereo.....	60
Fig. 4.26	Imaginea cu trasarea liniilor ajutătoare de calcul a poziției brațului robotic în 3D utilizând un singur algoritm.....	60
Fig. 4.27	Diagrama bloc pentru experimentul de detectare a brațului robotic în 3D – realizat cu LabVIEW și NI Vision Development Module.....	62
Fig. 4.28	Panoul frontal în LabVIEW pentru controlul brațului robotic în 3D.....	62, 137
Fig. 4.29	Imagine stereo pentru brațul robotic Lynxmotion AL5B cu desene ajutătoare pentru calculul poziției în 3D [imaginea mărită din interfața LabVIEW].....	63
Fig. 4.30	Diagrama bloc pentru experimentul de detectare a brațului robotic în 3D – realizat cu LabWindows/CVI și NI Vision Development Module.....	64
Fig. 4.31	Interfața grafică în LabWindows/CVI pentru controlul brațului robotic în 3D.....	65
Fig. 4.32	Trasarea liniilor ajutătoare pentru controlul brațului robotic în 3D în LabWindows/CVI.....	66
Fig. 4.33	Detectarea articulațiilor brațului robotic cu metoda cercurilor lui Hugh – program creat în C sub Linux.....	67
Fig. 4.34	Imagine mărită din program pentru detectarea articulațiilor cu	

	ajutorul metodei cercurilor lui Hugh.....	67
Fig. 4.35	Diagrama bloc pentru experimentul de detectare a brațului robotic în 3D – realizat în Linux în C și OpenCV.....	68
Fig. 4.36	Experimentul pentru controlul brațului robotic Lynxmotion AL5B în 3D cu detecție de marcheri folosind camera IP în C cu OpenCV sub Linux.....	69
Fig. 4.37	Aplicația pentru controlul brațului robotic Lynxmotion AL5B în planul 3D cu detecție de contur în C cu OpenCV sub Linux.....	70
Fig. 4.38	Imaginea mărită din aplicație de controlul a brațului robotic Lynxmotion AL5B în C cu OpenCV sub Linux cu trasarea liniilor pentru calculul poziției brațului robotic.....	70
Fig. 4.39	Experimentul de control al brațului robotic SCORBOT-ER III în 3D în C cu OpenCV sub Linux utilizând camere IP.....	71
Fig. 4.40	Aplicația pentru controlul brațului robotic SCORBOT-ER III în planul 3D cu detecție de contur în C cu OpenCV sub Linux.....	71
Fig. 4.41	Imaginea mărită din aplicație de controlul a brațului robotic SCORBOT-ER III în C cu OpenCV sub Linux cu trasarea liniilor pentru calculul poziției brațului robotic.....	72
Fig. 4.42	Diagrama bloc al experimentului de control al brațului robotic Lynxmotion AL5B cu Raspberry PI și camere web.....	73
Fig. 4.43	Interfața grafică pentru controlul brațului robotic Lynxmotion AL5B cu Raspberry PI – interfață grafică realizată în Python cu OpenCV.....	74
Fig. 4.44	Imagine mărită din interfața grafică pentru controlul brațului robotic Lynxmotion AL5B realizată pe Raspberry PI în Python cu OpenCV.....	74
Fig. 4.45	Diagrama bloc al experimentului de control al brațului robotic SCORBOT-ER III cu Raspberry PI și camere IP.....	75
Fig. 4.46	Interfața grafică pentru controlul brațului robotic SCORBOT-ER III cu Raspberry PI – interfață grafică realizată în Python cu OpenCV.....	76
Fig. 4.47	Imagine mărită din interfața grafică pentru controlul brațului robotic SCORBOT-ER III realizată pe Raspberry PI în Python cu OpenCV.....	76
Fig. 4.48	Diagrama bloc al experimentului de control al brațului robotic Lynxmotion AL5B cu placa FPGA ZYBO.....	77
Fig. 4.49	Diagrama bloc al experimentului de control al brațului robotic SCORBOT-ER III cu placa FPGA ZYBO.....	78
Fig. 4.50	Diagrama bloc al experimentului de control al brațului robotic Lynxmotion AL5B cu placa FPGA ZedBoard.....	79
Fig. 4.51	Diagrama bloc al experimentului de control al brațului robotic SCORBOT-ER III cu placa FPGA ZedBoard.....	80
Fig. 4.52	Interfața grafică al controlului brațului robotic Lynxmotion AL5B realizat pe plăcile ZYBO sau ZedBoard cu Linux – program creat în C cu OpenCV.....	81
Fig. 4.53	Imagine mărită din programul de control al brațului robotic Lynxmotion AL5B cu liniile ajutoare pentru calculul poziției brațului robotic în 3D.....	81
Fig. 4.54	Interfața grafică al controlului brațului robotic SCORBOT-ER III realizat pe plăcile ZYBO sau ZedBoard cu Linux – program creat în C cu OpenCV.....	82
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.....	83
Fig. 4.56	Comparația numărului de linii cod la diferite implementări.....	84
Fig. 4.57	Comparația mărimii codului pe disk la diferite implementări.....	85
Fig. 4.58	Comparația mărimii fișierului binar pe disk la diferite implementări...	85
Fig. 4.59	Comparația între cantitatea de memorie ocupată în timpul execuției la diferite implementări.....	86
Fig. 4.60	Comparația între procentul de utilizare UCP în timpul execuției la diferite implementări.....	86
Fig. 4.61	Comparația între resursele utilizate la diferite implementări.....	87
Fig. 5.1	Rezultatele studiului de Gage R&R – studiu de reproductibilitate.....	92
Fig. 5.2	Graficul de probabilitate normală pentru măsurarea distanței de 1 m folosind camere video.....	93
Fig. 5.3	Graficul de linii pentru cele șase tipuri de camere video și cele trei tipuri de platforme.....	93
Fig. 5.4	Graficul de probabilitate normală pentru cele trei tipuri de platforme.....	94
Fig. 5.5	Graficul de probabilitate normală pentru cele șase tipuri de camere.....	95
Fig. 5.6	Graficul de pătrate pentru cele șase tipuri de camere video.....	95
Fig. 5.7	Graficul de intervale pentru cele șase tipuri de camere video – ANOVA (analiză de variație).....	96
Fig. 5.8	Analiza valorilor reziduale pentru cele șase tipuri de camere video.....	97
Fig. 5.9	Analiza de capabilitate pentru măsurarea distanței de 1 m cu camere video la diferite fluxuri luminoase.....	98
Fig. 5.10	Histograma de capabilitate pentru măsurarea distanței de 1 m folosind camere video la diferite fluxuri luminoase.....	99
Fig. 5.11	Analiza I-MR (valoare individuală – limită mobilă) și testul de normalitate pentru măsurarea distanței de 1 m folosind camere video la diferite fluxuri luminoase.....	100
Fig. 5.12	Sumarul analizei de capabilitate pentru măsurarea distanței de 1 m folosind camere video la diferite fluxuri luminoase.....	101
Fig. 5.13	Analiza de capabilitate pentru măsurarea distanței de 1 m cu camere video la diferite temperaturi de culoare.....	102
Fig. 5.14	Histograma de capabilitate pentru măsurarea distanței de 1 m folosind camere video la diferite temperaturi de culoare.....	103
Fig. 5.15	Analiza I-MR (valoare individuală – limită mobilă) și testul de normalitate pentru măsurarea distanței de 1 m folosind camere video la diferite temperaturi de culoare.....	104
Fig. 5.16	Sumarul analizei de capabilitate pentru măsurarea distanței de 1 m folosind camere video la diferite temperaturi de culoare.....	104
Fig. 5.17	Histograma distanței reale și distanței calculate de sistemul cu camere video.....	106
Fig. 5.18	Graficul de probabilitate normală pentru distanța reală și distanța calculată de sistemul folosind camere video.....	107
Fig. 5.19	Graficul de probabilitate normală pentru eroare (delta).....	107
Fig. 5.20	Graficul de probabilitate normală pentru eroarea absolută (delta absolut).....	108
Fig. 5.21	Analiza de regresie pentru distanța reală și distanța calculată de sistemul folosind camere video.....	109

Fig. 5.22	Analiza valorilor reziduale pentru distanța reală și distanța calculată de sistemul folosind camere video.....	109
Fig. 5.23	Graficul de valori reziduale și valori potrivite pentru distanța reală și distanța calculată de sistemul folosind camere video.....	110
Fig. 5.24	Graficul de predicție pentru distanța reală și distanța calculată de sistemul folosind camere video.....	110
Fig. 5.25	Graficul de valori potrivite pe o dreaptă pentru distanța reală și distanța calculată de sistemul folosind camere video.....	111
Fig. 5.26	Sumarul analizei de regresie pentru distanța reală și distanța calculată de sistemul folosind camere video.....	112
Fig. 5.27	Analiza de capabilitate pentru distanța reală și distanța calculată de sistem folosind camere video.....	113
Fig. 5.28	Histograma de capabilitate pentru distanța reală și distanța calculată de sistemul folosind camere video.....	114
Fig. 5.29	Analiza I-MR (valoare individuală – limită mobilă) și testul de normalitate pentru distanța reală și distanța calculată de sistemul folosind camere video.....	114
Fig. 5.30	Sumarul analizei de capabilitate pentru distanța reală și distanța calculată de sistemul folosind camere video.....	115
Fig. 5.31	Grafic de prezentare distanța reală [m] și eroarea de măsurare (delta) [mm].....	116
Fig. A.1	Diagrama bloc în LabVIEW pentru controlul brațului robotic în 3D.....	141
Fig. A.2	Panoul frontal pentru calculul originii cercului.....	142
Fig. A.3	Diagrama bloc pentru calculul originii cercului.....	142
Fig. A.4	Panoul frontal pentru subprogramul de recunoaștere de culoare.....	142
Fig. A.5	Diagrama bloc pentru subprogramul de recunoaștere de culoare.....	143
Fig. A.6	Panoul frontal pentru ortogonalizare.....	143
Fig. A.7	Diagrama bloc pentru ortogonalizare.....	143
Fig. A.8	Panoul frontal pentru calculul de paralelogram.....	144
Fig. A.9	Diagrama bloc pentru calculul de paralelogram.....	144
Fig. A.10	Panoul frontal pentru calculul lungimii unui vector.....	144
Fig. A.11	Diagrama bloc pentru calculul lungimii unui vector.....	145
Fig. B.1	Interfața grafică în LabWindows/CVI pentru recunoașterea articulațiilor brațului robotic prin învățarea porțiunilor din imagine.....	146
Fig. B.2	Imaginea inițială a brațului robotic Lynxmotion AL5B.....	146
Fig. B.3	Porțiuni din imagine învățate de program.....	146
Fig. B.4	Articulațiile detectate din brațul robotic după ce au fost învățate de program.....	146

1. INTRODUCERE

1.1. Poziționarea lucrării în contextul actual

De-a lungul ultimelor decenii, în industrie au survenit o serie de modificări semnificative. Scopul urmărit a fost de a automatiza cât mai mult liniile de producție astfel încât omul să aibă munca lipsită de rutină și cât mai ușoară. Munca repetitivă și monotona, încet-încet a fost înlocuită de echipamente, apoi de roboți.

De curând a fost definit conceptul Industrie 4.0 (fig. 1.1), aferent celei de a 4-a revoluții industriale. Propus de specialiști germani [68], Industrie 4.0 propune ca prin utilizarea de sisteme inteligente, în primul rând roboți, automatizarea să fie atât de extinsă încât:

- linia de producție să se adapteze la noi metode fără influența omului;
- în cazul unui defect, rolul unui defect rolul unui robot să fie preluat dinamic de roboți adiacenți.

Prima revoluției industrială se consideră a fi perioada când au început să se introducă mecanisme propulsate de apă și aburi marcată de introducerea în industrie a războiului de țesut.

A doua revoluție industrială s-a realizat o dată cu apariția liniei de producție, prima dată la abatoarele din Cincinnati. Linia de producție este propulsată de curent electric. Munca omului este și mai mult simplificată și divizată pe etape, astfel încât operatorul să aibă un aport de muncă fizică redusă.

A treia revoluție industrială începe cu introducerea PLC-ului în industrie. Ideea de bază este introducerea electronicii și a PC-ului în producție. Fiind introdus robotul și calculatorul pe liniile de producție, astfel apărând o automatizare pronunțată a industriei [58].

A patra revoluție este revoluția bazată pe fuziunea dintre sistemele cibernetice și cele fizice, adică presupune introducerea telefoanelor mobile inteligente, a tabletelor și a altor sisteme de acest gen în industrie. Ideea de bază este conexiunea întregului sistem industrial cu Internetul [81].

Această lucrare se dorește o contribuție la o mică parte din a patra revoluție industrială, ideea de bază fiind controlul unui braț robotic cu o placă dedicată cu FPGA și cu două camere video. Camerele video sunt camere IP cu interfață Ethernet, dar se pot folosi și camere IP wireless. Placa de dezvoltare este o placă cu FPGA, care are port Ethernet, dar și port USB, deci conexiunea la internet pe fir se poate face la portul Ethernet sau pe wireless la portul USB cu un stick USB WiFi. Ideea de bază este ca operatorul uman să interacționeze cât mai puțin cu acest sistem (autonom), programarea brațului robotic se va executa cât mai rar, acesta fiind ghidat de camerele video, în acest fel nu mai este necesar se oprească linia de producție pentru calibrare, aceasta realizându-se în timpul execuției (on-the-go).

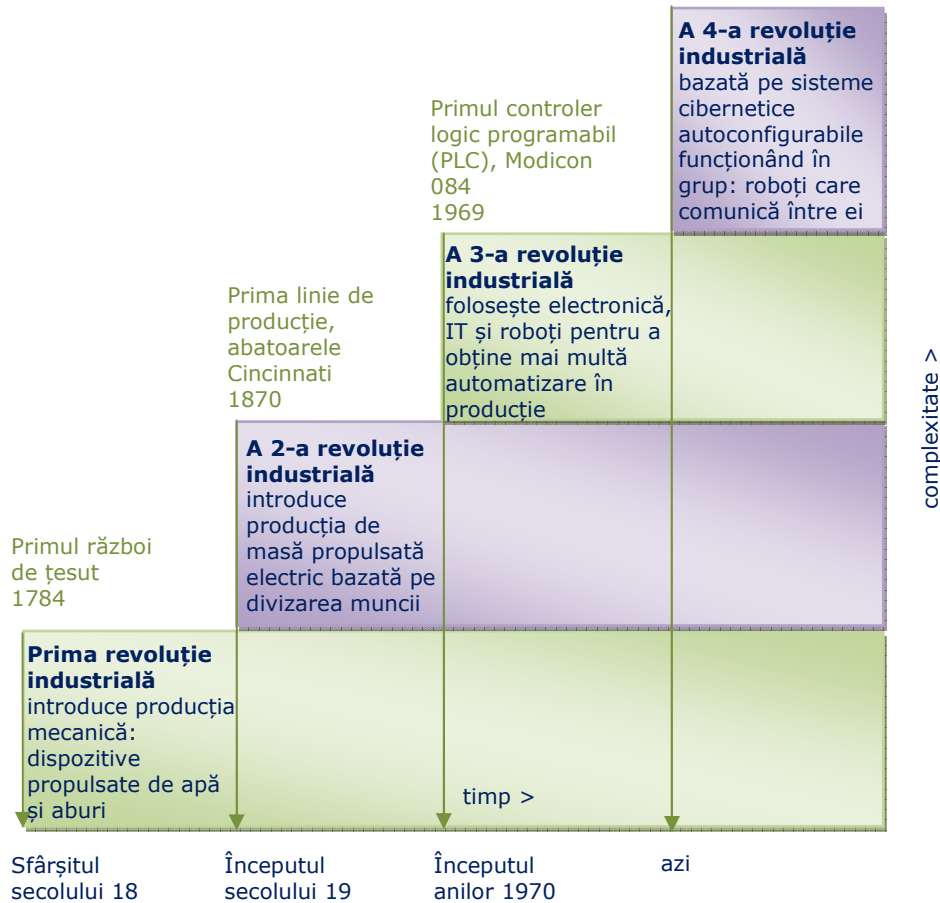


Fig. 1.1 Cele patru etape ale revoluției industriale, preluat după [34]

1.2. Prezentare generală

Brațul robotic este cel mai utilizat actuator în industrie [25], devenind una din cele mai utile și versatile invenții [27].

Designul și controlul unui braț robotic nu este o sarcină ușoară, deoarece este dificil pentru acesta să urmărească calea geometrică asignată într-o manieră precisă și cu mare acuratețe [15].

Ariile recente de cercetare sunt reprezentate de roboți de inspirație biologică. Mulți oameni sau animale se mișcă și umblă ușor fără a controla în mod explicit mișcărilor lor [16]. Roboții au nevoie de logică fuzzy pentru ca să facă mișcări cât mai umanoide. Rețele neuronale artificiale au fost incluse în mod tradițional pentru a învăța și calcula cinematica inversă a unui braț robotic [17].

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 [29].

Având în vedere conceptul Industrie 4.0, este important ca nu numai roboții umanoizi să aibă inteligență artificială pentru a putea lua decizii, ci și brațele robotice din industrie. Astfel, operatorul uman sau inginerul de linie va fi nevoit să intervină mai rar în procesul de producție. Un alt avantaj este că linia de producție va trebui să fie oprită mai rar, astfel eliminând astfel pierderi în producție [51].

Este necesară echiparea brațului robotic cu una sau mai multe camere video pentru a da vedere acestor mecanisme, astfel munca operatorilor se va diminua [52].

O utilizare foarte utilă ar fi implementarea de autocalibrare optică a brațelor robotice [53]. Se știe că după multe execuții repetate, un braț robotic deviază de la traiectorie, astfel trebuie oprită linia de producție și brațele robotice trebuie recalibrate [69]. O metodă optică permite mici ajustări la fiecare execuție, anulându-se astfel (teoretic) necesitatea altor recalibrări. O altă utilizare ar fi dacă obiectul care trebuie manipulat își schimbă poziția. Acest lucru în sistemele tradiționale de brațe robotice nu s-ar putea implementa fără a reprograma brațul robotic punct cu punct (PTP), operație consumatoare de timp [54]. Utilizarea sistemelor optice implică detectarea noii poziție a obiectului și controlul brațului robotic spre noul punct fără a fi necesară reprogramarea acestuia [59].

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 [60].

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 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 parametrii 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.

2. RESURSE. METODE

2.1. Sisteme cu microprocesor

2.1.1. Placa dedicată Raspberry PI

Placa dedicată Raspberry PI este un mini calculator de mărimea unei cărți de credit. Are niște caracteristici care permit instalarea unui sistem de operare cu interfață grafică din gama Linux, sporind astfel foarte mult domeniile de utilizare.

Sistemul este echivalent ca putere de calcul cu un telefon mobil din ziua de azi, numai că este destinat pentru educație, fiind realizat în forma unei plăci de dezvoltare.

Motivul utilizării acestei plăci de dezvoltare sunt dimensiunile reduse, performanțele tehnice și accesibilitatea. Pe placă se poate instala un sistem de operare Linux cu ajutorul căreia se poate face controlul unui braț robotic și se pot implementa algoritmi de detecție de marcheri. Sistemul mi-a fost de mare ajutor pentru a-mi însuși cunoștințele necesare pentru dezvoltarea sistemului cu FPGA. După crearea cu succes a arhitecturii de microprocesor pe FPGA, am instalat un sistem de operare Linux și am lucrat similar ca pe un Raspberry PI.

2.2. Sisteme reconfigurabile

2.2.1. Comparație între plăci de dezvoltare cu FPGA

FPGA-urile utilizate în această lucrare de cercetare au diferite arhitecturi și aparțin diferitelor familii de FPGA-uri. În tabelul 2.1 se observă modelele de FPGA-uri utilizate. Plăcile de dezvoltare utilizate sunt produse de firma Digilent [83].

Tabel 2.1 Familii de FPGA-uri, plăci de dezvoltare, modele

Familie	Placă de dezvoltare	Model FPGA
Spartan-6	ATLYS	XC6SLX45-CSG324C
Zynq	ZYBO	XC7Z2010-1CLG400C
	ZedBoard	XC7Z020-CLG484

Prima placă de dezvoltare utilizată este ATLYS, care oferă 4 porturi HDMI, un port USB, un controller UART pe port micro USB, 4 canale de sunet, un port Ethernet și un port VHDCI – pentru dispozitive care transmit sau recepționează imagini.

O altă placă de dezvoltare utilizată este placa ZYBO, care conține un SoC Zynq XC7Z2010, alcătuit dintr-un microprocesor ARM Cortex-9 Dual-core de 650 MHz și un FPGA de tip Atrix-7.

O altă placă de dezvoltare este placa ZedBoard, care conține un SoC Zynq XC7Z020, alcătuit dintr-un microprocesor ARM Cortex-9 Dual-core cu tact între 866 și 1000 MHz și un FPGA de tip Atrix-7.

În tabelul 2.2 s-a efectuat o comparație a celor trei plăci de dezvoltare amintite anterior. Comparația este făcută în sistem semafor: cel mai bun verde = 3 puncte, apoi galben = 2 puncte, apoi roșu = 1 punct. La sfârșit vedem și ordinea: locul I: placa ZedBoard cu 42 puncte, locul II: ZYBO cu 37 puncte și locul III: ATLYS cu 33 puncte.

Tabel 2.2 Comparație între plăcile de dezvoltare cu FPGA

Placă	ATLYS	ZYBO	ZedBoard
FPGA sau SoC	Xilinx Spartan-6 XC6SLX45-CSG324C, capsulă BGA cu 324-pini – 1 punct	SoC Zynq XC7Z2010-1CLG400C – 2 puncte	SoC Zynq-7000 XC7Z020-CLG484 – 3 puncte
CPU	– 1 punct	procesor ARM Cortex-A9 Dual-core până la 650 MHz – 3 puncte	procesor ARM Cortex-A9 Dual-core până la 667 MHz – 3 puncte
FPGA	Spartan-6 LX45: <ul style="list-style-type: none"> 6822 felii fiecare conținând patru LUT-uri cu 6 intrări și opt flip-flop-uri 2.1 Mbiți RAM rapid 4 plăcuțe de tact (8 DCM-uri & 4 PLL-uri) 6 bucle blocate pe fază 58 de felii de DSP frecvență de tact de viteze 500 MHz+ – 2 puncte	Artix-7: <ul style="list-style-type: none"> Celulele logice de 28K 240 Kocteți RAM 80 de DSP Convertor analog numeric de 1 MSPS, 12 biți și două canale (XADC) – 3 puncte	Artix-7: <ul style="list-style-type: none"> Celulele logice de 28K 240 Kocteți RAM 80 de DSP Convertor analog numeric de 1 MSPS, 12 biți și două canale (XADC) – 3 puncte
Memorie RAM	128 Mcocteți DDR2 pe 16-biți – 2 puncte	512 Mcocteți DDR3 x32 cu lățime de bandă de 1050 Mbps – 3 puncte	512 Mcocteți DDR3 – 3 puncte
Ethernet	Ethernet 10/100/1000 – 3 puncte	Ethernet PHY cu trei viteze (10 Mbit/100 Mbit/1 Gbit) – 3 puncte	Ethernet 10/100/1000 – 3 puncte
Programare	convertor USB2 la JTAG pentru programare cu mufă microUSB – 3 puncte	convertor USB2 la JTAG pentru programare cu mufă microUSB – 3 puncte	USB2-JTAG de programare – 3 puncte
USB	convertor USB-UART cu mufă microUSB și USB-HID (pentru mouse/tastatură) – 3 puncte	convertor UART la USB cu mufă microUSB și USB OTG 2.0 PHY (suportă postul de gazdă sau dispozitiv) – 2 puncte	USB OTG 2.0 și USB-UART – 1 punct
Video digital	două porturi video HDMI de intrare și două porturi video HDMI de ieșire (1080p) – 3 puncte	port HDMI bidirecțional de intrare/ieșire la 720p – 1 punct	port HDMI de ieșire (1080p60 + audio) – 2 puncte
Video analogic	– 1 punct	port de ieșire VGA cu 16 biți pe pixel – 3 puncte	port de ieșire VGA cu 8 biți – 2 puncte
Card memorie	– 1 punct	slot card microSD care suportă sistemul de fișiere Linux – 3 puncte	slot card SD care suportă sistemul de fișiere Linux și cu un card de 4 GB inclus – 2 puncte

Audio	codec audio AC-97 care conține canal de intrare, canal de ieșire, microfon și căști – 2 puncte	codec audio care conține canal de intrare, microfon și căști – 1 punct	codec audio I ² S de 24 de biți stereo care conține canal de intrare, canal de ieșire, microfon și căști – 3 puncte
EEPROM	– 1 punct	EEPROM extern (programat la nivel global cu un identificator unic de 48 biți compatibil cu EUI-48/64™) – 3 puncte	– 1 punct
Memorie Flash	16 Mcoțeți Flash SPI x4 pentru configurare și stocarea datelor – 1 punct	128 Mcoțeți Flash serial cu interfață QSPI – 2 puncte	256 Mb SPI Flash x4 – 3 puncte
Oscilator	oscilator CMOS de 100 MHz – 2 puncte	oscilator CMOS de 50 MHz – 1 punct	oscilatori CMOS de 33.333 MHz (PS) 100 MHz (PL) – 3 puncte
Canale intrare/ieșire	48 de canale de intrare/ieșire conectate la conectorii de expansiune (VHDCI + 1 conector Pmod) – 2 puncte	6 conectori Pmod – 48 de canale de intrare/ieșire (1 dedicat la procesor, 1 dublu analog/digital, 3 diferențiale de viteză mare, 1 dedicat pentru logică digitală) – 1 punct	5 conectori Pmod (40 de canale de intrare/ieșire) și un conector FMC (68 de canale de intrare/ieșire) – 3 puncte
GPIO	canalele de intrare/ieșire de uz general (GPIO) care include 6 butoane, 8 comutatoare și 8 LED-uri – 2 puncte	GPIO: 6 butoane, 4 comutatoare, 5 LED-uri – 1 punct	GPIO: 7 butoane, 8 comutatoare și 9 LED-uri, 128 x 32 OLED – 3 puncte
Alimentare	pachetul include un alimentator de 20W la 5V și un cablu USB – 3 puncte	pachetul nu include alimentator, dar se poate alimenta de la portul USB la unui PC – 2 puncte	pachetul include un alimentator la 12V și un cablu USB – 1 punct
Punctaj	6*1+6*2+5*3=33	5*1+4*2+8*3=37	3*1+3*2+11*3=42

În fig. 2.1 este redat graficul punctajelor fiecărei caracteristici ale plăcilor de FPGA utilizate. Placa ZedBoard obține cel mai mare scor, fiecare competitor are însă punctul lui forte. O caracteristică neprezentată în grafic este prețul, s-a ținut cont exclusiv de performanțe.

Plăcile ZYBO și ZedBoard sunt suficient de puternice pentru a instala un sistem de operare de tip Linux cu interfață grafică, devenind foarte versatile din punct de vedere a aplicațiilor putând rula aproape orice aplicație PC. Sistemul de operare utilizat este un Ubuntu Linux 12.04 LTS.

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 Koțeți L2 cache și memorie de 256 Koțeți pe chip. Sistemul conține și DDR3 respectiv PICE. Are criptare a comunicației cu protocoale AES, SHA și RSA. Printre porturi se pot aminti 2 x SPI, 2 x I²C, 2 x CAN, 2 x UART, 2 x USB, 2 x GigE și alte porturi digitale (2 x SDIO și GPIO).

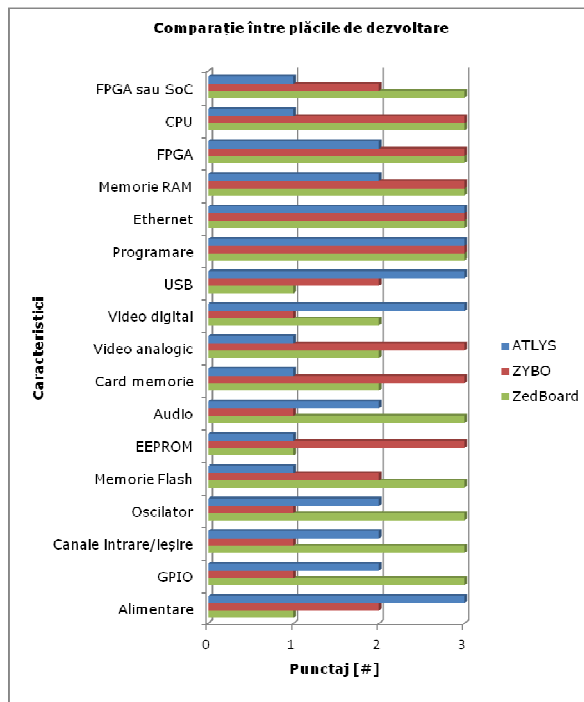


Fig. 2.1 Grafic de comparație între plăcile de dezvoltare

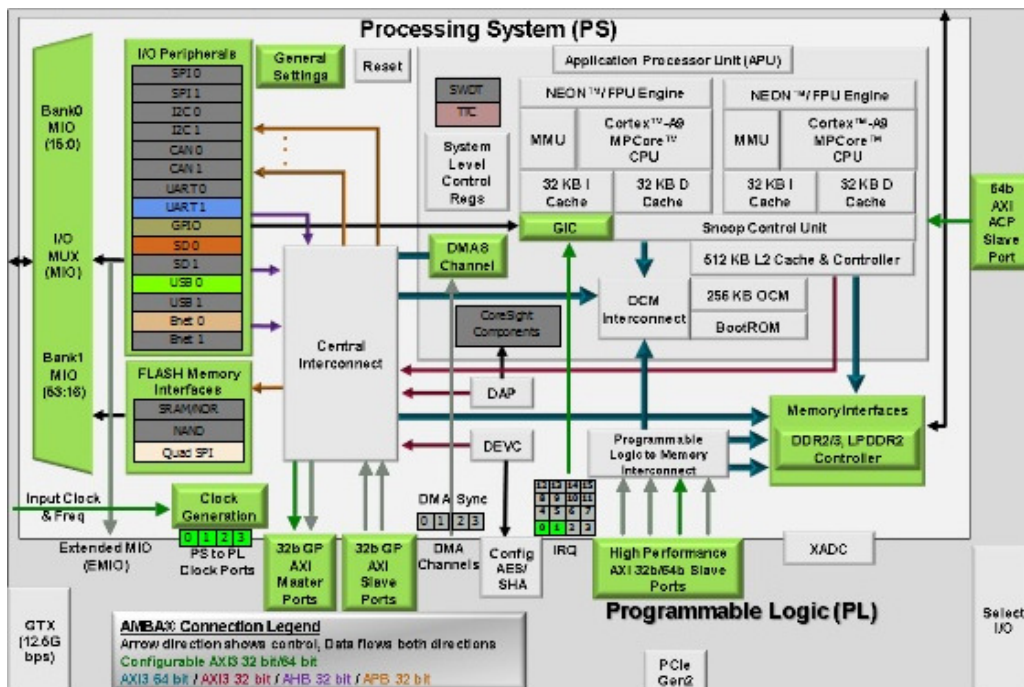


Fig. 2.2 Arhitectura SoC-ului Zynq-7000 creat în Xilinx Platform Studio

În tabelul 2.3 se văd resursele FPGA utilizate pentru crearea SoC-ului Zynq-7000 creat pe placa ZYBO sau ZedBoard și folosit pe parcursul experimentelor. Tabelul a fost generat în Xilinx ISE. Se poate vedea că resursele folosite sunt minimale, în ciuda complexității sistemului.

Tabel 2.3 Resurse utilizate din FPGA pentru crearea SoC-ului Zynq-7000 (în engleză, rezumat)

Device Utilization Summary				[-]
Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	4,040	106,400	3%	
Number used as Flip Flops	4,026			
Number used as AND/OR logics	14			
Number of Slice LUTs	3,894	53,200	7%	
Number used as logic	3,513	53,200	6%	
Number using O6 output only	2,039			
Number using O5 output only	258			
Number using O5 and O6	1,216			
Number used as Memory	257	17,400	1%	
Number used as Dual Port RAM	232			
Number using O6 output only	180			
Number using O5 and O6	52			
Number used as Single Port RAM	20			
Number using O5 and O6	20			
Number used as Shift Register	5			
Number using O6 output only	5			
Number used exclusively as route-thrus	124			
Number with same-slice register load	100			
Number with same-slice carry load	24			
Number of occupied Slices	1,871	13,300	14%	
Number of LUT Flip Flop pairs used	5,271			
Number with an unused Flip Flop	1,643	5,271	31%	
Number with an unused LUT	1,377	5,271	26%	
Number of fully used LUT-FF pairs	2,251	5,271	42%	
Number of unique control sets	335			
Number of slice register sites lost to control set restrictions	1,445	106,400	1%	
Number of bonded IOBs	85	200	42%	
Number of LOCed IOBs	85	85	100%	
Number of bonded IOPAD	130	130	100%	
IOB Flip Flops	19			

Number of RAMB18E1/FIFO18E1s	9	280	3%	
Number using RAMB18E1 only	9			
Number of BUFG/BUFGCTRLs	3	32	9%	
Number used as BUFGs	3			
Number of ILOGICE2/ILOGICE3/ISERDESE2s	3	200	1%	
Number used as ILOGICE2s	3			
Number of OLOGICE2/OLOGICE3/OSERDESE2s	72	200	36%	
Number used as OLOGICE2s	72			
Number of PLLE2_ADVs	1	4	25%	
Number of PS7s	1	1	100%	
Average Fanout of Non-Clock Nets	3.49			

În fig. 2.3 se observă un grafic generat de Xilinx ISE pentru resursele utilizate pentru a crea SoC-ul Zync 7000 creat pe placa ZYBO sau ZedBoard, care constă de microprocesorul ARM Cortex – A9 și un FPGA de generația a 7-a, de obicei Atrix 7. Se remarcă un procentul redus de utilizare din FPGA.

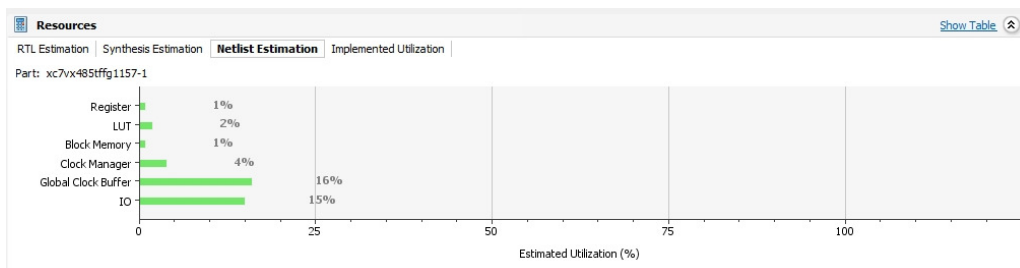


Fig. 2.3 Resurse folosite din SoC Zync 7000 care este configurat pentru a rula sistem de operare Linux

2.3. Brațe robotice

2.3.1. Prezentarea generală a brațelor robotice utilizate

Sistemul, care va fi prezentat ulterior, poate controla un braț robotic în spațiul 3D, astfel se pot controla trei motoare ale brațului robotic (mișcarea bazei în direcțiile stânga-dreapta și față-spate și mișcarea cotului în direcția sus-jos) folosind doar două camere video. Comandând trei motoare ale brațului robotic este suficient pentru a putea apuca un obiect din apropierea brațului robotic. Sistemul se poate extinde pentru a controla un număr mai mare de motoare utilizând mai multe camere video. Sistemul este similar brațului uman, neavând mișcarea încheieturii. Baza robotului copiază mișcarea umărului uman și cotul robotului copiază mișcarea cotului uman.

Pentru aceste experimente s-au folosit două brațe robotice. Unul educațional în două versiuni (Lynxmotion AL5A de înălțime aproximativă de 146 mm și AL5B de înălțime aproximativă de 191 mm, ambele având greutatea aproximativă de 1 kg) și

unul industrial (SCORBOT-ER III cu rază de acțiune de 610 mm și cu greutate de 16 kg).

Scopul urmărit a fost demonstrarea funcționării pe un model de robot educațional, iar mai apoi să se valideze ideea pe un robot industrial.

Sistemul de control al brațului robotic este general, astfel cu mici ajustări se poate folosi la orice braț robotic.

2.3.2. Brațele robotice Lynxmotion AL5A și AL5B

Brațele robotice AL5A și AL5B fac parte din familia de roboți AL5x, x = A, B, C, D. Singura diferență între acești roboți este următoarea: dacă ultima literă este mai departe în alfabet, atunci și brațul robotului este mai lung, materialele de construcție a motoarelor sunt de duritate mai mare, brațul robotic are putere mai mare de ridicare greutate și prețul este mai ridicat. Roțile dințate din motoarele robotului la versiunea AL5A sunt majoritatea din plastic, la varianta AL5D ele sunt din metal extradur tratat cu carbon [51].

În fig. 2.4 este prezentat robotul AL5B, utilizat în experimente.

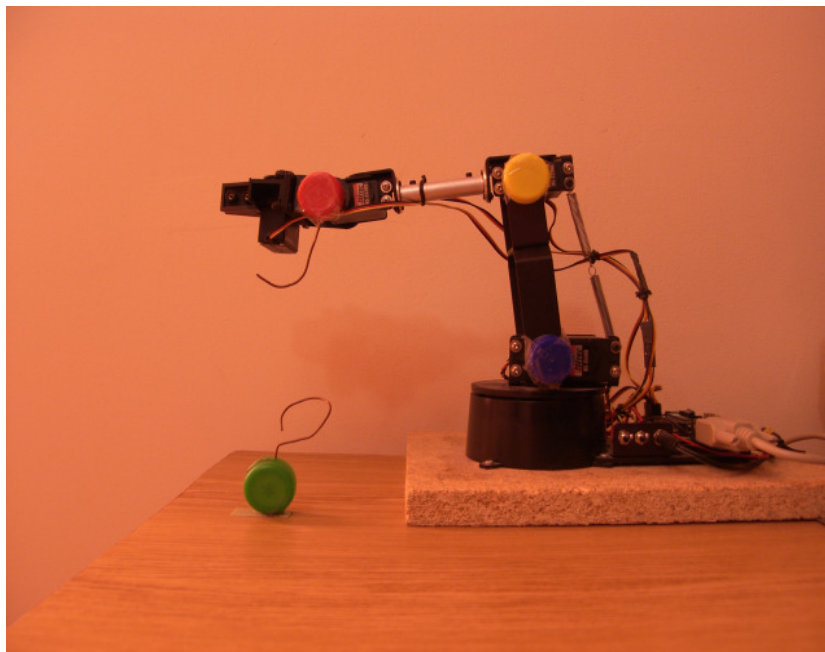


Fig. 2.4 Brațul robotic Lynxmotion AL5B

Partea electronică de comandă a brațului robotic este placa SSC-32 cu microcontrolerul ATmega168.

Circuitul mai conține pe lângă microcontrolerul ATmega168, o memorie EEPROM pentru salvare programe, driverul serial ST202EC cu mufa DB9 și 4 registre 74HC595MTC de deplasare cu intrare serială și ieșire paralelă (SIPO).

Portul serial RS-232 este utilizat pentru comunicație, iar registrele de deplasare pentru a demultiplexa semnalele pe 32 de canale digitale.

Firma Lynxmotion la roboții săi oferă programe (LynxTerm, SSC-32 Sequencer, RIOS/SSC-32 Arm Control Software), prin care se pot comanda roboții

precum cu ajutorul plăcii SSC-32, care se pot descărca de pe internet gratuit sau contra cost. Ele s-au dovedit a fi insuficiente pentru scopul tezei și s-au utilizat doar pentru a obține comenzile SCPI. Rulând aceste programe și cu ajutorul programului de monitorizare a portului serial (HHD Free Serial Port Monitor – fig. 2.5) au fost capturate comenzile SCPI pentru a realiza aplicațiile de comandă a brațelor robotice.

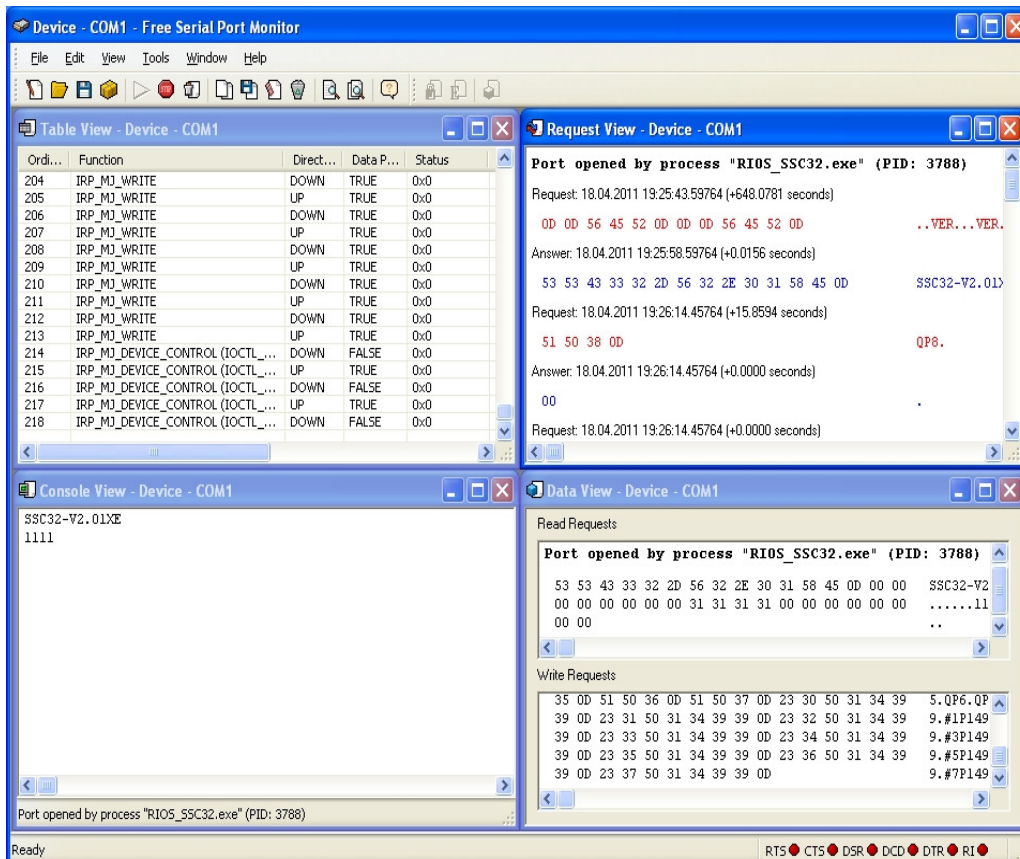


Fig. 2.5 Captarea comenzilor SCPI cu ajutorul programului HHD Free Serial Port Monitor

Cu aceste comenzi se poate controla orice servomotor conectat la această placă.

```
// VERSIUNE SSC-32
\r\rVER\r

// INIȚIALIZARE MOTOARE
QPL0\rQP0\r
QP1\r
QP2\r
//...
QP31\r
// TOATE SERVO MOTOARELE ÎN POZIȚIA 1500 (POZIȚIA DE MIJLOC)
#OP1500S0\r#1P1500S0\r#2P1500S0\r#3P1500S0\r#4P1500S0\r#5P1500S0\r
```

```
// EFECTOR FINAL
#4P1500S1000\r

// ROTIRE ÎNCHEIETURĂ
#5P1500S1000\r

// ÎNCHEIETURĂ
#3P1500S1000\r

// COT
#2P1500S1000\r

// ŞOLD
#1P1500S1000\r

// BAZĂ
#0P1500S1000\r
```

Comenzile SCPI pentru brațul robotic reflectă următoarea sintaxă prezentată în cele ce urmează. Simbolul “#” semnifică numărul motorului. Litera “P” semnifică poziția servomotorului, care este între 500 și 2500; 1500 fiind poziția de mijloc, adică 90°, deoarece servomotoarele sunt blocate mecanic să poată face o mișcare doar între 0° și 180°. Pentru aceste valori s-a dedus și o formulă care va fi prezentată în capitolele care urmează. După litera “S” urmează timpul în milisecunde, care standard este de 1000 ms, ceea ce înseamnă că brațul robotic încearcă să execute mișcarea sub 1000 ms indiferent de distanța de mișcare.

Comanda de mișcarea motorului:

```
#aPbcdeSfghi↵
```

a este un număr între 0 și 31 și specifică numărul motorului operat;
P este un parametru de **poziție motor**;
bcde este un număr între 500 și 2500, care de fapt semnifică poziția de mișcare a motorului, care este calculat în pulsuri;
S este un parametru de **viteză (speed) motor**;
fghi viteza de mișcare a motorului, calculat în ms, 0 semnifică viteza maximă.
↵ <CR> (Carriage Return).

2.3.3. Brațul robotic SCORBOT-ER III

SCORBOT-ER III (fig. 2.6) este un braț robotic articulată vertical. Are o bază care se rotește în plan orizontal și trei articulații care se rotesc în plan vertical, și o încheietură care rotește efectorul final [31].

Cadrul robotului este care conține cinci din cele șase motoare [31].

Partea superioară a brațului, antebrațul și articulațiile permite mișcarea efectorului final [31].

Efectorul final este la capătul brațului robotic. Pentru a ajunge la poziția dorită, doar câteva sau toate axe trebuie puse în mișcare [31].

Degetele efectorului final se deschid și se închid în paralel. Diferite tipuri de tampoane sau efectori, cum ar fi o perie de aer sau un dispozitiv de prindere cu vid, pot fi atașate la efectorul final, prin intermediul găurilor cu diametru de 4 mm [31].

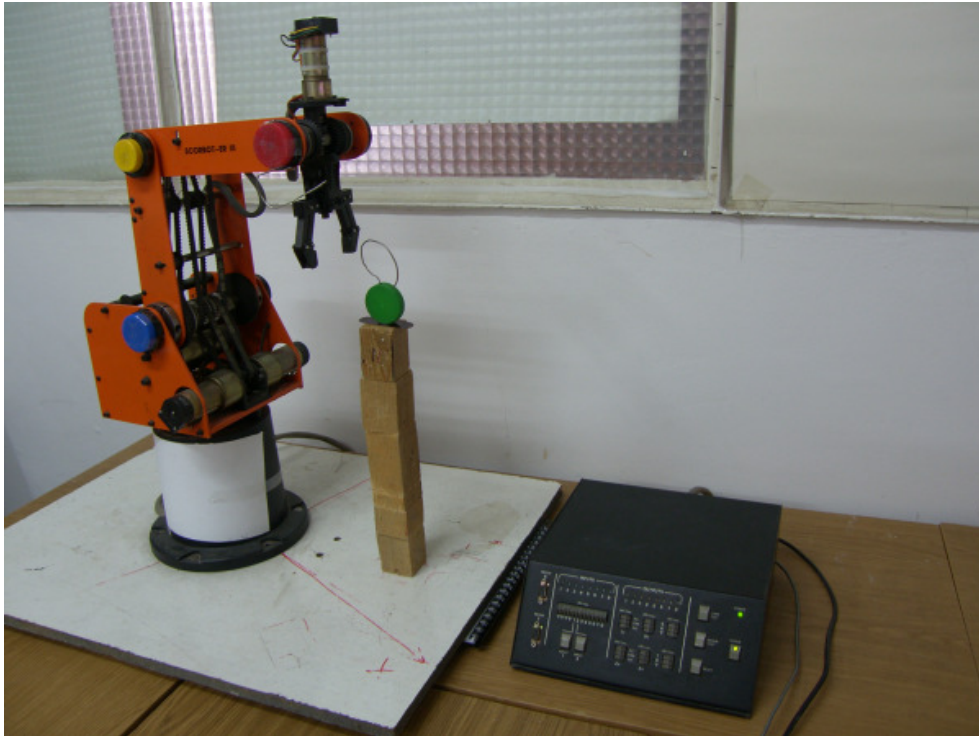


Fig. 2.6 Brațul robotic SCORBOT-ER III

Specificațiile tehnice ale brațului robotic SCORBOT-ER III sunt cele din tabelul 2.4.

Tabel 2.4 Specificațiile brațului robotic SCORBOT-ER III, preluat după [31]

Element	Specificație
Structura mecanică	Articulată pe verticală 5 axe plus efector final Control de 8 axe simultane
Raza de acțiune: Axa 1: rotație de bază Axa 2: rotație de umăr Axa 3: rotație de cot Axa 4: mișcare sus-jos încheietură Axa 5: rotație încheietură	310° +130°/-35° ±130° ±130° nelimitat
Raza maximă de lucru	610 mm
Deschiderea efectorului final	75 mm fără tampoane de cauciuc 65 mm cu tampoane de cauciuc
Sarcină maximă de lucru	1 kg

Transmisie	Roți dințate, curele de sincronizare și șurub
Actuatori	6 servomotoare DC care au control de circuit închis
Control al mișcării (feedback)	Codificatoare optice pe toate axele
Poziția inițială	Poziție de referință pe toate axele
Repetabilitate	$\pm 0,5$ mm
Viteză maximă	330 mm/s
Masă	
Brațul robotic	11 kg
Controller	5 kg

Motoare:

Cele cinci axe și sistemul de prindere de pe robot sunt operate de servomotoare DC. Direcția de rotație a motorului este determinată de polaritatea tensiunii de funcționare: tensiune pozitivă de curent continuu învârtă motorul într-o direcție, în timp ce tensiunea negativă de curent continuu învârtă motorul în direcția opusă. Fiecare motor de are control în buclă închisă, un circuit codificator oferă feedback regulatorului despre viteza și direcția de mișcare a motorului [31].

Micro comutatoare:

Cinci micro comutatoare sunt montate pe brațul mecanic. Când robotul este pornit, fiecare micro comutator este deconectat, această poziție este cunoscută ca și poziția inițială, poziția de repaus. Acesta este punctul de referință pentru operarea robotului. Ori de câte ori sistemul este pornit, robotul ar trebui să fie resetat în această poziție inițială [31].

Transmisie:

Mai multe tipuri de transmisie sunt folosite pentru a opera părțile mecanice ale brațului robotic. Roți dințate pun în mișcare baza robotului. Roți dințate și curele de sincronizare pun în mișcare cotul robotului. Curele sincronizate și o unitate de diferențială de roți dințate de la capătul brațului pune în mișcare încheietura robotului. Un cablu și un șurub cuplat direct la un servomotor DC determină efortul final să se deschidă și să se închidă [31].

Comenzile SCPI pentru brațul robotic SCORBOT-ER III sunt diferite de cele de la brațul robotic de tipul Lynxmotion AL5x. Cea mai mare diferență o găsim în faptul că acest braț robotic nu are poziții fixe, ci numai poziții relative, deci distanța de deplasare tot timpul va depinde de poziția anterioară a robotului. Este necesară o rutină de revenire sau resetare a brațului robotic în poziția inițială, astfel ca brațul robotic tot timpul să aibă un reper fix cunoscut.

Comenzile SCPI al brațului robotic SCORBOT – ER III sunt [31]:

Comanda de mișcarea motorului:

aM±bcde.┘

a	este un număr între 1 și 8 și specifică numărul motorului operat;
M	semnifică că este o comandă de mișcare motor ;
±	este direcția de rotire a motorului, fără semn se consideră direcție pozitivă;
bcde	este un număr între -8000 și 8000, care de fapt semnifică cantitatea de mișcare a motorului, care este calculat în pulsuri;
␣	<CR> (Carriage Return).

Comanda pentru setarea vitezei motorului:

aVb

a	este un număr între 1 și 8 și specifică numărul motorului operat;
V	semnifică că este o comandă de viteză motor ;
b	este un număr între 0 și 9, viteza crește de la 1 la 9, iar 0 semnifică viteza maximă.

2.4. Concluzii și contribuții

Pe parcursul acestui capitol 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 [83]), 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. Interfețele ce comunicare folosite au fost: USB, Ethernet, VHDCI (pentru camerele video), RS-232 (pentru controlul brațelor robotice) și HDMI (pentru afișare pe un monitor).

Contribuții

1. **S-a implementat arhitectura microprocesorului OpenRISC-32 pe FPGA-ul Spartan-6 de pe placa ATLYS și s-a instalat Linux-ul cu linie de comandă pe memoria FLASH a acestei plăci.**
2. **S-a fuzionat arhitectura ARM Cortex – A9 Dual-core cu FPGA-ul Atrix-7 pe SoC-ul Zynq-7000 de pe plăcile ZYBO și ZedBoard și s-a instalat Ubuntu Linux 12.04 LTS și OpenCV pe cardurile de memorie (microSD respectiv SD) care vin cu aceste plăci cu FPGA.**
3. **S-au comparat performanțele celor trei plăci cu FPGA ATLYS, ZYBO și ZedBoard – tabelul 2.2 și fig. 2.1.**

3. ALGORITM ȘI METODĂ

3.1. Tema de cercetare științifică

3.1.1. Prezentarea generală a temei de cercetare științifică

Scopul acestei lucrări dezvoltarea și implementarea unui algoritm de detecție de marcheri destinat controlului unui braț robotic folosind camere video conectate la un FPGA utilizat pentru toate funcțiile implementate.

Pentru a face posibilă funcționarea acestui sistem, au trebuit să fie dezvoltate driverele FPGA pentru a interfața perifericele.

Metoda propusă necesită aplicarea bulinelor colorate în puncte cheie ale brațului robotic, astfel ca aceste culori să fie recunoscute de sistem. După acesta se va desena scheletul brațului robotic, astfel fiind posibilă introducerea coordonatelor brațului robotic în FPGA. Acest pas va fi urmat de un proces de prelucrare date, ca mai apoi brațul robotic să fie controlat pe baza informațiilor obținute de la camerele video.

Pe FPGA s-au utilizat două metode de implementare, una tradițională, totul fiind realizat în hardware și o implementare în limbaj de programare de nivel înalt pe un sistem de operare instalat pe FPGA. La cea din urmă am dezvoltat o arhitectură de microcontroler în FPGA, apoi am instalat sistemul de operare Linux și biblioteca de recunoaștere de imagini OpenCV. Varianta Linux are avantajul că are create drivere pentru porturi, astfel la acest sistem cu FPGA se pot conecta cu ușurință camerele pe interfețele Ethernet sau USB.

Sistemul a fost implementat și în sistemul de operare Windows în limbajele de programare LabWindows/CVI, LabVIEW și MATLAB cu bibliotecile de recunoaștere imagini specifice acestor limbaje de programare. De asemenea am dezvoltat sistemul de detecție de marcheri și în sistemul de operare Ubuntu Linux în limbajul de programare C cu biblioteca de recunoaștere de imagini OpenCV. După acesta se va crea sistemul de detecție de marcheri și pe o placă dedicată cu microprocesor ARM, pe Raspberry PI cu sistemul de operare Raspbian Linux în limbajul de programare Python cu biblioteca de recunoaștere de imagini OpenCV.

3.1.2. Justificarea alegerii temei de cercetare

Este interesantă integrarea acestor sisteme într-un sistem unic, complex, (un robot autonom), care poate lua decizii în funcție de imaginile achiziționate cu două sau mai multe camere conectate la un sistem de calcul.

Sistemul propus nu dorește să fie unul de sine stătător, ci mai degrabă un sistem complementar, care să completeze într-un întreg unitar sistemele deja existente în industrie.

3.1.3. Stadiul actual al cercetării în tematica tezei propuse

Controlul unui braț robotic folosind camere video este o sarcină de interes în lumea robotică, dar controlul brațului robotic cu două sau mai multe camere are un număr redus de implementări în industrie [38].

Introducerea coordonatelor poziției unui obiect în mișcare utilizând buline colorate sau puncte luminoase se practică în testarea și perfecționarea tehnicii atleților. De asemenea se mai utilizează în crearea mișcărilor similare oamenilor a personajele 3D în jocuri pe PC și console [60], [65]. Toate aceste metode se fac off-line [68], adică se face prelucrarea datelor după achiziția coordonatelor. Implementările în literatura de specialitate în timp real sunt relativ puține, deoarece puterea de procesare trebuie să fie considerabilă pentru această sarcină [128]. La un braț robotic lucrurile se simplifică, deoarece, brațul robotic nu are atâtea articulații ca un om.

Sistemul de recunoaștere nu este de o complexitate mare, deoarece trebuie recunoscute doar culori, apoi acestea sunt unite cu linii drepte și astfel se poate introduce poziția unui braț robotic în întreg sistemul de calcul. Ulterior se urmărește întreaga mișcare a acestui braț robotic (fig. 3.1).

Aportul personal este algoritmul și calculul matematic utilizat în detecția poziției brațului robotic și în calculul poziției viitoare al brațului robotic.

Se utilizează cercuri care materializează traseele de mișcare a fiecărui motor în parte. Se vor trasa tangente la cercuri, care vor crea un sistem coordonate prin care se calculează poziția curentă și poziția viitoare dorită. Din acest sistem de coordonate se pot trasa triunghiuri, iar prin calculul arctangentei unghiurilor acestora, se poate afla exact unghiul de mișcare al fiecărui motor.

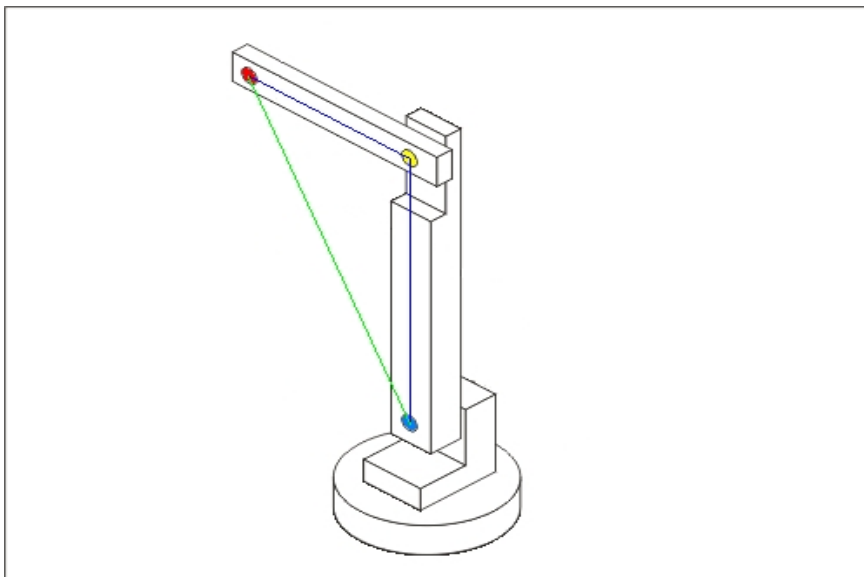


Fig. 3.1 Localizarea brațului robotic în spațiu utilizând buline colorate, preluat după [37], apoi modificat

În fig. 3.2 se observă brațul robotic Lynxmotion AL5A cu buline colorate aplicate la încheieturi pentru a recunoaște brațul folosind camere video.

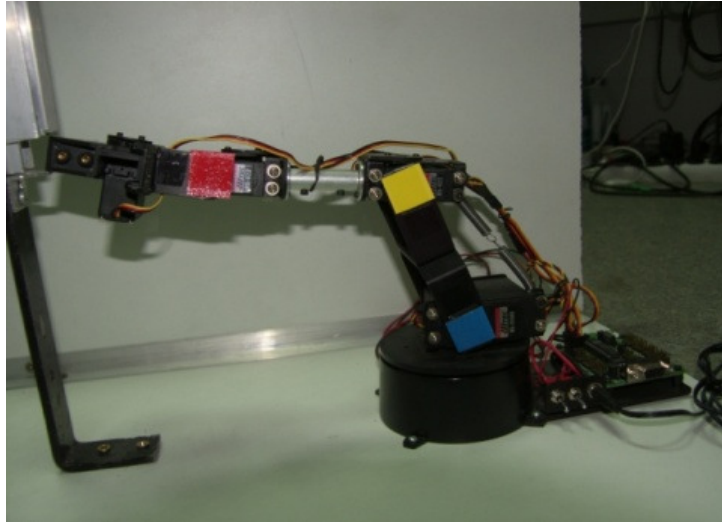


Fig. 3.2 Hârtii colorate aplicate în puncte cheie ale brațului robotic Lynxmotion AL5A

În fig. 3.3 se observă brațul robotic Lynxmotion AL5B cu dopuri colorate aplicate la încheieturi pentru a recunoaște brațul robotic folosind camere video. Dopurile colorate sunt utile și când brațul robotic este rotit în jurul bazei, deoarece dopul este vizibil, acesta având o formă tridimensională (bulinele colorate nu se văd bine când brațul robotic este rotit în jurul bazei).

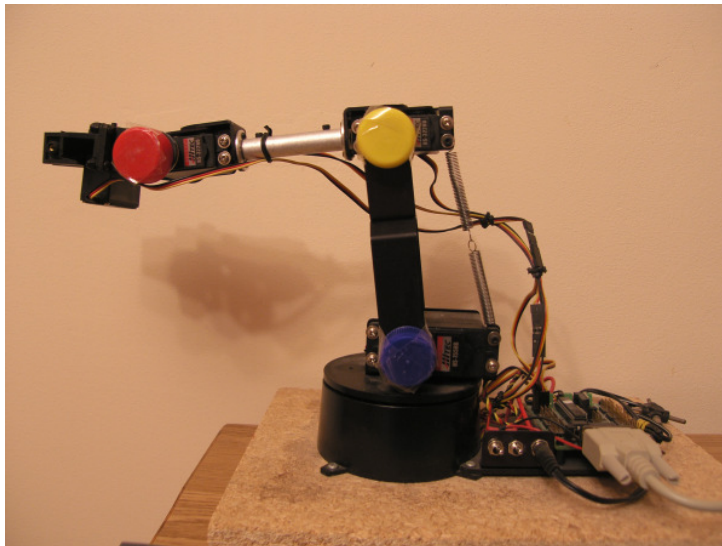


Fig. 3.3 Dopuri colorate aplicate în puncte cheie ale brațului robotic Lynxmotion AL5B (fiind tridimensional sunt vizibile și când brațul robotic este rotit în jurul bazei)

În fig. 3.4 se observă brațul robotic SCORBOT-ER III cu dopuri colorate mai mari aplicate la încheieturi pentru a recunoaște brațul folosind camere video. S-au utilizat dopuri colorate mai mari, deoarece și brațul robotic este mai mare.



Fig. 3.4 Dopuri mai mari colorate aplicate în puncte cheie și la punctul țintă ale brațului robotic SCORBOT-ER III (dopul este vizibil și când brațul robotic este perpendicular pe camerele video)

În literatura de specialitate se pot întâlni mai multe experimente, care folosesc brațe robotice controlate de camere video.

Într-un articol au fost conectate camerele video cu un braț robotic care avea ca sarcină acoperirea cu plasmă a anumitor obiecte. Scopul sistemului era de a reduce timpul și costul de a învăța brațul robotic. Brațul robotic cu șase grade de libertate avea conectate camere video și putea manipula mai multe obiecte [38]. Sistemul acesta este similar cu cel prezentat în această teză, se dorește ca brațul robotic să poată fi mișcat fără să se mai învețe punct cu punct pozițiile intermediare ale acestuia. Se mai dorește și evitarea calibrării brațului robotic, astfel ca acest lucru să se desfășoare automat în timp ce brațul robotic execută mișcările necesare de a manipula obiectul țintă.

O altă modalitate de a crea un ansamblu de braț robotic și cameră este de a monta camera exact pe efectorul final al brațului robotic. Această metodă putea fi avantajoasă, deoarece camera filma obiectele în măsură ce se apropia de ele. Sistemul nu are nevoie de calcule de cinematică inversă sau calcule cu matrice Jacobiene inverse. Este nevoie doar de o cameră montată pe robot, care poate estima distanțe [39]. Metoda este interesantă, având doar o singură cameră montată pe efectorul final, însă este mai degrabă o metodă complementară celei prezentate în această teză. Dacă sistemul este format din camerele fixe montate lângă robot, aceasta poate avea o imagine de ansamblu atât al brațului robotic cât și al obiectului pe care acesta dorește să îl manipuleze.

O modalitate interesantă, similară cu experimentul anterior este format dintr-un braț robotic cu mai multe degete și o cameră video care filmează mediul înconjurător. Sistemul de implementare este complex, deoarece mai există un sistem paralel, care poate reconstrui obiectul din imaginea captată, astfel este calculat cum poate fi manipulat acesta de brațul robotic [40]. Sistemul necesită o putere de calcul foarte mare și plasarea unui astfel de sistem pe o platformă dedicată ar fi o adevărată provocare, însă probabil nici nu s-a propus acest lucru. Pe

de altă parte, rata în care procesoarele dedicate și FPGA-urile evoluează, este foarte mare. Plasarea unui sistem complex pe o platformă dedicată, este doar o chestiune de timp, dacă nu este deja posibilă. Acest sistem este mult mai complex ca și cinematica inversă sau directă; sistemul care va fi prezentat în cadrul tezei de doctorat fiind mult mai simplu din punct de vedere al complexității calculului.

O modalitate similară cu cele prezentate anterior este tot un sistem care are camera montată la efectorul final, însă în acest experiment se mai ia în considerare și dinamica camerei video. Sistemul face mai multe iterații în continuu și calculează toate traiectoriile posibile aranjându-le într-o structură arborescentă. După ce a făcut un anumit număr de calcule, acesta selectează traiectoria optimă de deplasare [41]. Sistemul este interesant, dar resursele necesare pentru a funcționa sunt destul de mari, din cauza numărului mare de iterații. Plasarea unei camere la efectorul final al robotului dă un spectru de viziune mult mai îngust, decât mai multe camere fixe plasate în apropierea brațului robotic.

Un alt sistem prezintă controlul unui braț robotic cu șase grade de libertate cu ajutorul camerelor video. Sistemul are la bază proiecția sferică în trei puncte ale distanțelor carteziane. A doua parte constă în reprezentarea unghi-axă a matricii de rotație obținut din două puncte măsurate de pe imagine [42]. Sistemul este foarte interesant și este apropiat de cel prezentat în continuare, însă calculul matematic utilizat este complex.

Au fost prezentate și sisteme mai complexe unde sistemul de camere s-a inspirat din biologie. Tot sistemul se bazează pe rețele neuronale, s-a încercat să se imite coordonarea între cameră și mișcarea brațului robotic similar cu cea a omului, acesta având o coordonare precisă între ochi și braț. Sistemul acesta a fost apoi comparat cu sisteme convenționale, care se bazează pe calibrare. Acesta sistem este mai performant decât cele existente până acum în literatură [43]. Fiind o rețea neuronală complexă, implementarea acestora necesită un efort pronunțat. Sistemul este eficient, dar mult prea complicat. Sistemul care va fi prezentat a avut în vedere simplitatea și posibilitatea de a fi implementat pe FPGA.

Un sistem de control foarte interesant care dorește să copieze mișcările brațului uman utilizează, de asemenea, un set de camere stereo, care captează imaginile [44]. Sistemul poate controla un braț robotic cu șase grade de libertate. Sunt utilizate ecuații diferențiale, autocorelație, covarianță, procese Wiener, filtre Kalman și alte ecuații și formule specifice din teoria semnalelor. Partea hardware este similară cu sistemul realizat de mine.

Un alt sistem asemănător cu cel din teză controlează servomotoare cu ajutorul unei camere video. Sistemul ia în considerare doar viteza, ceea ce este suficient pentru multe brațe robotice clasice, însă la multe sisteme trebuie luată în considerare și dinamica sistemului, ceea ce de altfel menționează și autorii [45]. Sistemul are și alți senzori, care împreună cu o cameră video fac ca tot sistemul să se comporte în modul dorit. Sensorii adiționali cresc complexitatea sistemului.

Un grup de cercetători [46] se ocupă de erorile de cinematică ale brațului robotic. Sistemul este similar cu unul clasic, dar are o cameră în plus care verifică dacă poziția diferă de cea dorită. Ca sistemul să fie precis, erorile au fost măsurate cu un laser. Camera verifică dacă laserul este direcționat pe toată durata funcționării sistemului în punctul dorit. Sistemul pe care va fi prezentat în teză are o funcție de autocalibrare, în sensul că brațul robotic va face mici ajustări în timpul mișcării până când se va poziționa în punctul țintă. Brațul robotic iese din bucla de execuție doar atunci când a ajuns exact în punctul țintă, acest lucru fiind indicat și de marcajele trasate pe imaginea captată în timp real.

3.2. Metode de calcul

3.2.1. Prezentarea algoritmului propus

Algoritmul utilizat presupune aplicarea unor linii și cercuri ajutătoare pe imaginea captată în timp real de camerele web. Acestea se utilizează pentru calculul distanțelor.

În fig. 3.5 se observă cum vor fi desenate liniile și cercurile ajutătoare pe imaginea video cu brațul robotic. Aici este redată o imagine 3D generată al brațului robotic de tip Lynxmotion AL5x cu desenele de ghidaj pentru calculul poziției.

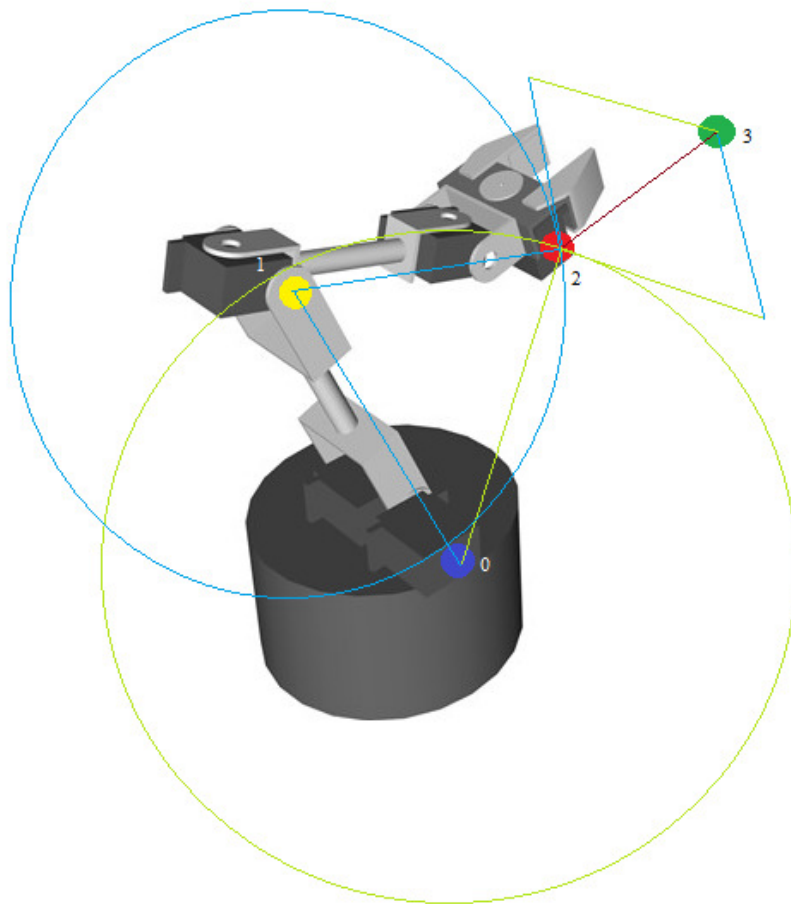


Fig. 3.5 Desenele ajutătoare pe modelul 3D al brațului robotic, preluat după [36], apoi modificat

În fig. 3.6 se observă desenele imprimate pe imaginea video.

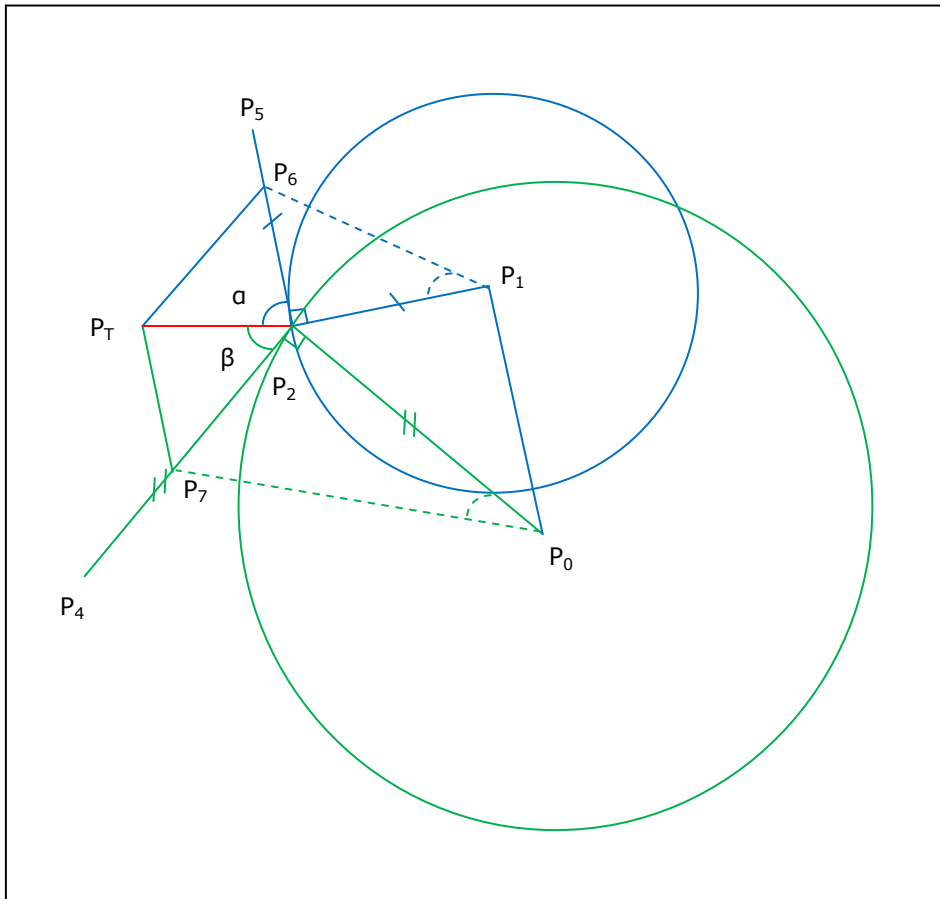


Fig. 3.6 Desene ajutătoare pentru calculul de distanțe 2D imprimate pe imaginea video

Punctul P_0 se referă la bulina albastră de pe brațul robotic (baza robotului), punctul P_1 se referă la bulina galbenă pe brațul robotic (cotul robotului), iar punctul P_2 se referă la bulina roșie (efectorul final al brațului robotic). Punctul P_T este punctul țintă (punctul unde trebuie să ajungă efectorul final), acesta este marcat cu bulina verde.

Unind punctele P_1 și P_2 și P_1 cu P_0 cu linii albastre, iar P_2 cu P_0 cu linie verde, obținem scheletul brațului robotic. Liniile albastre sunt trasate pe brațul robotic (acestea sunt linii fixe din punct de vedere al lungimii) iar linia verde arată deschiderea brațului robotic (această linie este variabilă în lungime). În final s-a unit punctul P_2 cu punctul P_T cu linie roșie (și această linie este variabilă în lungime în funcție de unde este plasat punctul țintă).

În jurul punctului P_1 se trasează un cerc albastru, iar în jurul punctului P_0 se trasează un cerc verde. Cercul albastru are rază fixă, însă are o origine variabilă, cercul verde are origine fixă, dar are o rază variabilă. După acestea se trasează tangente la cercurile albastru și verde. Metoda cea mai simplă este să se calculeze

vectorul ortogonal la segmentul P_1P_2 albastru și la segmentul P_0P_2 verde, trasând segmente perpendiculare de lungime egală cu razele cercurilor. Segmentul verde P_4P_2 este egal și perpendicular cu segmentul verde P_2P_0 , iar segmentul albastru P_5P_2 este egal și perpendicular cu segmentul albastru P_2P_1 .

S-a obținut un sistem de coordonate, care nu este ortogonal, dar potrivit scopului urmărit. Punctele de interes sunt P_6 și P_7 , acestea trebuie determinate.

Se poate trasa o dreaptă paralelă cu dreapta verde P_2P_4 ; această dreaptă trece și prin punctul P_T , care sigur va intersecta dreapta albastră pe care sunt P_2 și P_5 . Intersecția este realizată prin punctul P_6 . Se poate trasa o dreaptă paralelă, cu dreapta albastră pe care sunt punctele P_5 și P_2 , această dreaptă paralelă mai trece și prin punctul P_T , care sigur va intersecta dreapta verde pe care sunt punctele P_2 și P_4 . Intersecția este realizată prin punctul P_7 . Astfel se obține paralelogramul $P_2P_6P_7P_T$ de dimensiune variabilă. Acest paralelogram este de fapt sistemul de coordonate la care informația utilă este lungimea segmentelor P_2P_7 și P_2P_6 . Se știe lungimea segmentelor P_0P_2 și P_2P_1 (se cunoaște poziția punctelor P_0 , P_1 și P_2 , coordonatele cărora au fost aflate cu ajutorul unei singure camere). La fel se cunoaște și coordonatele punctului țintă P_T și a punctelor de intersecție P_6 și P_7 .

Se cunoaște lungimea segmentelor P_2P_7 și P_2P_0 , care sunt lungimile catetelor triunghiului dreptunghic verde, pentru care se dorește să se afle unghiul $P_2P_0P_7$ de mișcare al motorului de bază al brațului robotic folosind o funcție trigonometrică (arctangentă).

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 $P_2P_6P_7P_T$ 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).

Unghiurile astfel calculate trebuie convertite în valori robotice.

Extinderea tridimensională a prezentului algoritm 2D se realizează utilizând o cameră video secundară și calcule de segmente.

În continuare se vor prezenta formulele de calculul pentru trasarea desenelor ajutătoare aplicate pe imagine în timp real achiziționată de camerele web.

S-au determinat valorile pentru comanda robotului. Acestea s-au obținut prin convertirea unghiurilor de mișcare a motoarelor după cum urmează.

$$\text{constanța}_{\text{robot}} = \frac{\Delta\omega}{180^\circ - 0^\circ} = \frac{2500 - 500}{180^\circ - 0^\circ} = 11.(1)_{\text{valori}_{\text{robot}}}$$

Valorile comenzilor robotice fiind între 500 și 2500, aferent socotite unghiului maxim de 180° , astfel relația între unghiuri și valori robotice este.

$$1^\circ \cong 11.(1)_{\text{valori}_{\text{robot}}}$$

Diferența dintre vectori s-a calculat în felul următor (3.1):

$$\begin{cases} \Delta x = x_2 - x_0 \\ \Delta y = y_2 - y_1 \end{cases} \quad (3.1)$$

Lungimea vectorului cu norma euclidiană este:

$$\|l\| = \sqrt{x^2 + y^2}$$

Vectorul ortogonal s-a calculat în felul următor:

$$\begin{cases} \tilde{x}_{ort} = y \\ \tilde{y}_{ort} = -x \end{cases} \quad (3.2)$$

Combinând (3.1) cu (3.2), rezultă coordonatele punctelor P_4 și P_5 .

$$\begin{cases} x_4 = \Delta x_{0x} - 2 \cdot \Delta x_{0x} + x_2 \\ y_4 = \Delta x_{0y} - 2 \cdot \Delta x_{0y} + y_2 \end{cases} \quad (3.3)$$

$$\begin{cases} x_5 = \Delta y_{0x} - 2 \cdot \Delta y_{0x} + x_2 \\ y_5 = \Delta y_{0y} - 2 \cdot \Delta y_{0y} + y_2 \end{cases} \quad (3.4)$$

Paralelogramul $P_7P_T P_6P_2$ s-a calculat după cum urmează.
Panta (m) a celor două tangente la cercuri este:

$$\begin{cases} m_\alpha = \frac{y_5 - y_2}{x_5 - x_2} \\ m_\beta = \frac{y_4 - y_2}{x_4 - x_2} \end{cases}$$

Dreapta care trece prin punctul P_2 pentru efectorul final al brațului robotic este descrisă de:

$$y_2 = m_\alpha x_2 + b \quad (3.5)$$

Adică:

$$y = m_\alpha x + y_2 - m_\alpha x_2 \quad (3.6)$$

Utilizând (3.6) de două ori pentru x și de două ori pentru y , pentru ambele pante (m_α și m_β), s-au obținut expresiile (3.7-3.10).

$$y_6 = m_\alpha x_6 + y_2 - m_\alpha x_2 \quad (3.7)$$

$$y_7 = m_\beta x_7 + y_2 - m_\beta x_2 \quad (3.8)$$

$$y_6 = m_\beta x_6 + y_T - m_\beta x_T \quad (3.9)$$

$$y_7 = m_\alpha x_7 + y_T - m_\alpha x_T \quad (3.10)$$

Introducând y_6 în (3.7) și (3.9) s-a obținut (3.11).

$$m_a x_6 - m_\beta x_6 = y_T - m_\beta x_T - y_2 + m_a x_2 \quad (3.11)$$

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 este descris modul de calcul al distanței utilizând camerele stereo.

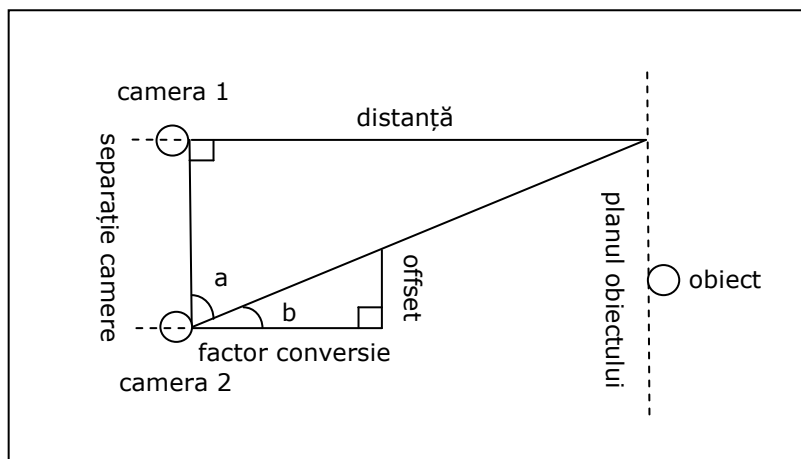


Fig. 3.7 Calculul distanței stereo, preluat după [32], apoi modificat

În fig. 3.7, triunghiul dreptunghic are o catetă care reprezintă distanța obiectului (*distanță*), iar cealaltă catetă care reprezintă distanța între camere, sau separația între camere (*separație camere*).

În triunghiul dreptunghic mai mic (de calibrare), o catetă e diferența între cele două imagini, iar cealaltă catetă este factorul de conversie (*factor conversie*), care este un parametru de calibrare. Sistemul oferă astfel o funcție de autocalibrare, deoarece factorul de conversie se recalculează de fiecare dată când este calculată distanța, eveniment care se petrece de fiecare dată când robotul își schimbă direcția.

S-au calculat două distanțe una la bulina albastră, adică la baza robotului, iar cealaltă la bulina verde, adică la punctul țintă, unde trebuie să ajungă efectul final. Aceste două distanțe sunt scăzute una din cealaltă, această diferență reprezentând cateta unui triunghi dreptunghic, iar cealaltă catetă este segmentul verde P_0P_2 din fig. 3.6.

Utilizând formulele ce urmează, valorile de distanțe măsurate se pot transforma din coordonate din lumea reală în coordonate de pixeli pentru a le putea sincroniza cu valorile obținute în fig. 3.6. Cu funcții trigonometrice se va calcula valoarea unghiului de mișcare al brațului robotic pe axa OX , iar după aceea această valoare va fi convertită într-o comandă înțeleasă de robot. Imaginea din fig. 3.6 este dublată pentru redundanță, pe lângă asta este utilizată pentru a obține parametrii distanței măsurate folosind camerele stereo. În fig. 3.6 sunt determinate distanțele pentru axele OY și OZ , iar în fig. 3.7 sunt determinate distanțele pentru axa OX . Poziția brațului robotic în spațiul 3D rezultă combinând cei doi algoritmi, cel de pe fig. 3.6 dublat, filmat cu două camere, iar cel din fig. 3.7, care folosește și parametrii obținuți cu algoritmul prezentat în fig. 3.6.

În continuare se vor prezenta formulele pentru calculul distanței stereo conform fig. 3.7.

S-a calculat tangenta la unghiurile \hat{a} și \hat{b} în felul următor [32].

$$\begin{cases} \operatorname{tg}(a) = \frac{\text{dis\ tan\ ta}}{\text{separatie_camere}} \\ \operatorname{tg}(b) = \frac{\text{offset}}{\text{factor_conversie}} \end{cases}$$

Ofsetul s-a calculat ca și diferența dintre punctele inițiale de pe imaginea stângă și dreaptă [32].

$$\text{offset} = |x_{0R} - x_{0L}|$$

Factorul de conversie rezultă [32]:

$$\text{factor_conversie} = \frac{\text{offset} \cdot \text{dis\ tan\ ta_initiala}}{\text{separatie_camere}}$$

Ofsetul s-a calculat relativ la punctul P_2 după cum apare mai jos [32].

$$\text{offset} = |x_{2R} - x_{2L}|$$

Distanța finală s-a calculat astfel [32]:

$$dis\ tant a_ finala = \frac{factor_ conversie \cdot separatie_ camere}{offset}$$

Unghiul în grade devine:

$$unghi = \frac{180^\circ}{\pi} \cdot arctg\left(\frac{lungime_ tangenta}{lungime_ raza}\right)$$

Valorile pentru comenzile robotice finale s-au calculat în felul următor.

$$valori_ robot_ finale = unghi \cdot 11.(1)_ valori_ robot$$

Coordonatele din lumea reală s-au convertit în pixeli după cum urmează. Monitorul utilizat este un monitor de 19", iar rezoluția pentru imaginea capturată este 320x240 rezultând [35]:

$$\begin{cases} rezolutie_ orizontala = 320 \\ rezolutie_ verticala = 240 \\ diagonala = 19" \end{cases} \quad (3.14)$$

Densitatea de pixeli s-a obținut din (3.14). Din (3.15) s-a obținut datele pentru mărimea imaginii, pentru o mărime de monitor anume, care în cazul experimentului a fost 21.05 PPI [35].

$$\begin{aligned} marime_ monitor &= 15.2" \cdot 11.4" = 173.28_ in^2 \\ la_ 21.05PPI \end{aligned} \quad (3.15)$$

Distanța (în pixeli) rezultă [35]:

$$scala[pixeli] = dis\ tant a_ lume_ reala[cm] \cdot \frac{21.05[PPI]}{2.54[cm]} \quad (3.16)$$

3.2.2. Prezentarea algoritmilor și formulelor alternative

În continuare se va prezenta o altă metodă de calcul a distanței prin triangulație (fig. 3.8), deoarece la anumite aplicații s-a folosit această de metodă de calcul al distanței stereo.

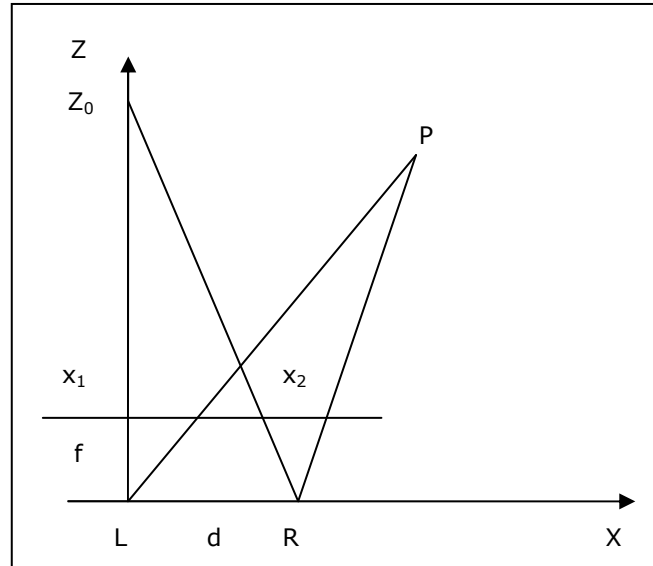


Fig. 3.8 Calculul distanței stereo prin triangulare, preluat după [33]

Această metodă presupune ca anumite elemente să fie cunoscute. L și R se referă la cele două camere stânga (left) și dreapta (right), P este punctul în spațiu unde se află obiectul, la acest punct se măsoară distanța folosind camerele video, d este distanța între cele două camere, această informație trebuie să fie cunoscută. Tot sistemul este proiectat în spațiul 3D, astfel L și R sunt pe axa OX . Dreapta care trece prin R se intersectează cu axa OZ prin punctul Z_0 . L și R sunt unite și cu punctul P, care este punctul țintă în spațiu. Distanța focală este notată cu f , la distanța focală dacă trasăm o dreaptă, aceasta intersectează LZ_0 prin x_1 și RZ_0 prin x_2 . Cu aceste puncte putem să calculăm distanța cu două camere prin triangulare.

Coordonata Z s-a calculat cum se observă mai jos [33].

$$Z = \frac{d \cdot f}{x_1 - x_2 + f \cdot \operatorname{tg}\left(\theta \cdot \frac{\pi}{180}\right)}$$

Coordonata X s-a calculat în felul următor [33].

$$X = x_1 \cdot \frac{Z}{f}$$

Coordonata Y s-a calculat după cum urmează [33].

$$Y = y_1 \cdot \frac{Z}{f}$$

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

Segmentele P_7P_2 și P_2P_0 sunt catetele triunghiului dreptunghic verde, cu ajutorul căruia se poate calcula unghiul $P_2P_0P_7$ cu o funcție trigonometrică (arctangentă).

Segmentele P_8P_2 și P_2P_1 sunt catetele triunghiului dreptunghic albastru, cu ajutorul căruia putem calcula unghiul $P_2P_1P_8$.

Pentru controlul 3D suntem interesați de lungimea perpendicularei din punctul țintă P_T pe planul $P_4P_2P_5$.

Segmentul perpendicular pe planul $P_4P_2P_5$ din punctul țintă P_T intersectează planul în punctul P_6 . Segmentul P_6P_T este perpendicular pe planul $P_4P_2P_5$. Se vor trasa alte perpendiculare paralele cu acest segment P_6P_T , dar care vor intersecta colțurile paralelogramului $P_8P_6P_7P_2$, astfel vor fi patru segmente paralele și perpendiculare pe planul $P_4P_2P_5$, care intersectează colțurile paralelogramului $P_8P_6P_7P_2$, iar acestea sunt P_7P_9 , P_6P_T , P_8P_{10} și P_2P_{11} .

Paralelogramul $P_8P_TP_9P_{11}$ este paralel în spațiu cu paralelogramul $P_8P_6P_7P_2$. Lungimea vectorului de interes este P_TP_6 , care se obține calculând distanța din punctul țintă P_T la planul $P_4P_2P_5$.

Segmentul $P_{11}P_2$ este paralel și are aceeași lungime cu segmentul P_TP_6 . $P_{11}P_2P_0$ formează un triunghi dreptunghic în spațiul 3D util pentru calcularea unghiului pentru mișcarea pe axa OX .

S-a calculat ecuația planului $P_4P_2P_5$. După aceea s-a calculat distanța sau vectorul ortogonal din punctul țintă P_T la plan.

S-au notat coordonatele în spațiu cu P , Q , R , deoarece s-a dorit să se scrie ecuația planului care trece prin trei puncte în felul următor.

$$\underbrace{(x_1, y_1, z_1)}_P \underbrace{(x_2, y_2, z_2)}_Q \underbrace{(x_3, y_3, z_3)}_R$$

Al doilea pas este calculul vectorilor \overrightarrow{PQ} și \overrightarrow{PR} :

$$\overrightarrow{PQ} = \langle x_2 - x_3, y_2 - y_3, z_2 - z_3 \rangle$$

$$\overrightarrow{PR} = \langle x_1 - x_3, y_1 - y_3, z_1 - z_3 \rangle$$

Apoi s-a calculat vectorul normal prin produsul vectorial în felul următor.

$$\overrightarrow{PQ} \times \overrightarrow{PR}$$

Produsul vectorial s-a calculat după cum urmează.

$$\begin{vmatrix} i & j & k \\ x_2 - x_3 & y_2 - y_3 & z_2 - z_3 \\ x_1 - x_3 & y_1 - y_3 & z_1 - z_3 \end{vmatrix}$$

S-a simplificat în felul următor ca să se obțină determinanți de ordin doi, care se pot calcula cu o putere de procesare mai mică ca și determinantul de ordinul trei:

$$i \begin{vmatrix} y_2 - y_3 & z_2 - z_3 \\ y_1 - y_3 & z_1 - z_3 \end{vmatrix} - j \begin{vmatrix} x_2 - x_3 & z_2 - z_3 \\ x_1 - x_3 & z_1 - z_3 \end{vmatrix} + k \begin{vmatrix} x_2 - x_3 & y_2 - y_3 \\ x_1 - x_3 & y_1 - y_3 \end{vmatrix}$$

S-au calculat determinantele de ordin doi. S-a înmulțit diagonala principală și s-a scăzut diagonala secundară, de asemenea s-a notat și fiecare membru cu literele a, b și c (elementele ecuației planului), rezultând (3.17).

$$\begin{aligned} & \underbrace{i[(y_2 - y_3)(z_1 - z_3) - (y_1 - y_3)(z_2 - z_3)]}_a \\ & - j \underbrace{[(x_2 - x_3)(z_1 - z_3) - (x_1 - x_3)(z_2 - z_3)]}_b \\ & + k \underbrace{[(x_2 - x_3)(y_1 - y_3) - (x_1 - x_3)(y_2 - y_3)]}_c \end{aligned} \quad (3.17)$$

S-au scris ecuațiile generale pentru plan care au incluse puncte P_0 și P_2 după cum se observă la (3.18).

$$\begin{cases} a(x - x_0) + b(y - y_0) + c(z - z_0) = 0 \\ a(x - x_2) + b(y - y_2) + c(z - z_2) = 0 \end{cases} \quad (3.18)$$

Ecuația parametrică a unui punct prin care trece o dreaptă perpendiculară pe un plan este:

$$\begin{cases} x_0 = u - a \frac{au + bv + cw + d}{a^2 + b^2 + c^2} \\ y_0 = v - b \frac{au + bv + cw + d}{a^2 + b^2 + c^2} \\ z_0 = w - c \frac{au + bv + cw + d}{a^2 + b^2 + c^2} \end{cases}$$

S-a scris și ecuația generală a planului:

$$ax + by + cz + d = 0 \quad (3.19)$$

S-a înlocuit (3.17) în (3.18) și s-a obținut (3.20).

$$\begin{aligned} & [(y_2 - y_3)(z_1 - z_3) - (y_1 - y_3)(z_2 - z_3)] \cdot (x - x_2) \\ & - [(x_2 - x_3)(z_1 - z_3) - (x_1 - x_3)(z_2 - z_3)] \cdot (y - y_2) \\ & + [(x_2 - x_3)(y_1 - y_3) - (x_1 - x_3)(y_2 - y_3)] \cdot (z - z_2) \end{aligned} \quad (3.20)$$

S-au recunoscut membrii a, b și c ale ecuației planului cu ajutorul (3.19) și s-a obținut (3.21) după calcule.

$$\begin{cases} a = y_2 z_1 - y_2 z_3 - y_3 z_1 - y_1 z_2 + y_1 z_3 + y_3 z_2 \\ b = -(x_2 z_1 - x_2 z_3 - x_3 z_1 - x_1 z_2 + x_1 z_3 + x_3 z_2) \\ c = x_2 y_1 - x_2 y_3 - x_3 y_1 - x_1 y_2 + x_1 y_3 + x_3 y_2 \end{cases} \quad (3.21)$$

S-a recunoscut și membrul d al ecuației planului cu ajutorul (3.19) și s-a obținut (3.22) după calcule.

$$d = x_2y_3z_1 - x_2y_1z_3 - y_2x_3z_1 + y_2x_1z_3 + z_2x_3y_1 - z_2x_1y_3 \quad (3.22)$$

Aceste metode alternative au fost folosite în anumite programe de calcul a distanței stereo. În capitolul următor se vor prezenta diferite implementări, astfel să poată compara metodele diferite de implementare și să poată alege cea mai bună metodă de calcul al distanțelor folosind camere video.

3.3. Concluzii și contribuții

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.

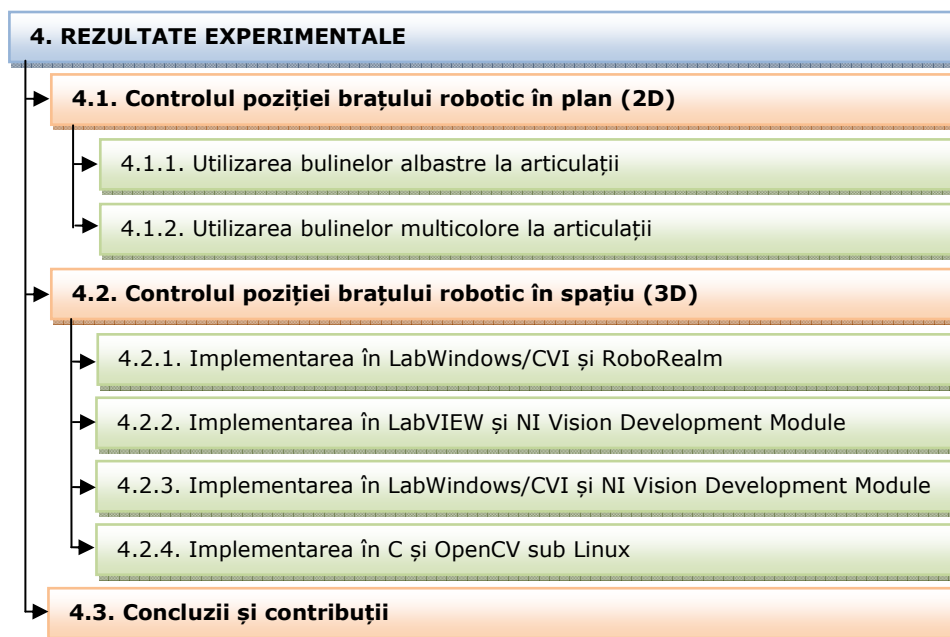
Deoarece masa roboților utilizați la partea experimentală nu este semnificativă, iar sarcina utilă este redusă, aspect coroborat de controlul prin servomotoare, problemele de dinamică nu au fost luate în considerare, considerând autocalibrarea optică suficientă.

Contribuții

Metoda de calcul și algoritmul de control al mișcării sunt **originale**.

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. Toate formulele au fost simplificate și puse într-o formă implementabilă în cod sursă calculator.
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.
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.
5. A fost dezvoltat **algoritmul de trasare a linilor și cercurilor ajutătoare pentru calculul distanțelor** pentru brațul robotic, având ca date de intrare doar coordonate obținute din poziția în spațiu a bulinelor colorate detectate.
6. Am demonstrat geneza tuturor formulelor și le-am validat prin mișcarea controlată a robotului.

4. REZULTATE EXPERIMENTALE



4.1. Controlul poziției brațului robotic în plan (2D)

4.1.1. Utilizarea bulinelor albastre la articulații

Poziția brațului robotic se poate recunoaște cu ajutorul camerelor în mai multe moduri. Una din metode este de a distinge numai anumite puncte cheie ale brațului robotic și restul să se completeze cu algoritmi de prelucrare de imagini. Metoda utilizată în această lucrare este de a plasa buline de o anumită culoare la articulațiile robotului, acestea să fie recunoscute cu ajutorul filtrării de culoare, apoi acestea să se unească cu linii, astfel să se obțină scheletul robotului.

În fig. 4.1 se observă diagrama bloc pentru recunoașterea brațului robotic în plan cu ajutorul recunoașterii de culoare. Bulinele colorate sunt așezate în puncte cheie ale brațului robotic pentru a se putea introduce scheletul robotului în unitatea centrală de procesare al sistemului dedicat. Tot sistemul constă din placa cu FPGA ATLYS de la Digilent la care este conectată camera stereo VmodCAM prin cablul VHDCI. La placă este conectat un monitor pe port HDMI și un Pmod RS232 pentru a comunica cu brațul robotic pe interfața RS-232. Tot sistemul este dezvoltat în limbaj VHDL, deci **toată implementarea este realizată în hardware**. Am implementat în VHDL inclusiv comunicarea pe porturile VHDCI, HDMI și RS-232; de asemenea tot în VHDL este implementat și algoritmul de recunoaștere de culoare. Pentru a

recunoaște poziția în spațiu a articulațiilor brațului robotic, s-au aplicat buline colorate, care sunt recunoscute folosind camerele video.

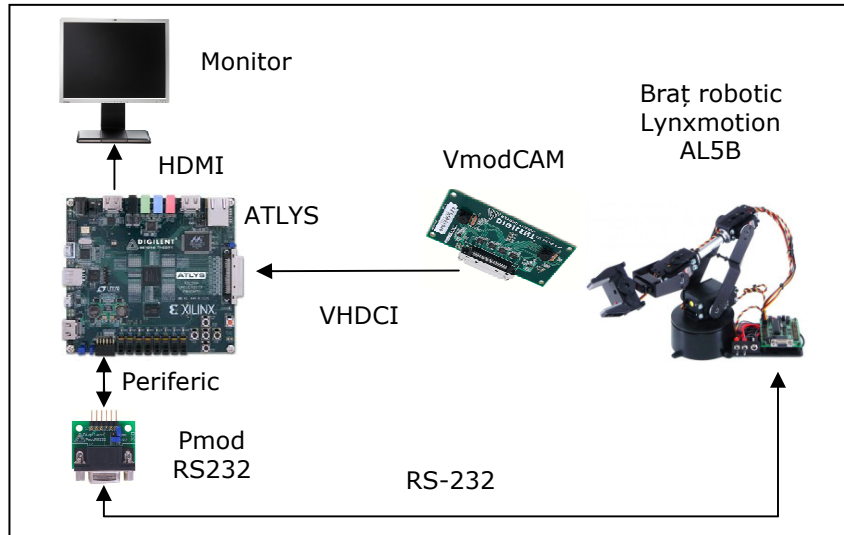


Fig. 4.1 Diagrama bloc pentru sistemul de control al brațului robotic cu FPGA ATLYS prin recunoașterea bulinelor albastre de la articulații – programare în VHDL

În tabelul 4.1 se observă resursele utilizate de placa cu FPGA Spartan-6 ATLYS. Se poate vedea că nu au fost utilizate toate resursele plăcii, însă numărul de elemente logice utilizate este mai mare față de sistemele cu SoC Zynq-7000. Acest lucru este dat și de faptul că placa ATLYS are resurse mai puține, ca și plăcile cu SoC Zynq-7000, deci din punct de vedere al numărului de elemente logice ocupate poate fi mai mare aria de acoperire, dar ca și procent de ocupare din FPGA implementările sunt similare.

Tabel 4.1 Resurse utilizate din placa FPGA ATLYS pentru crearea sistemului de control al brațului robotic cu recunoașterea bulinelor albastre de la articulații (în engleză, rezumat)

Device Utilization Summary				[-]
Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	1,103	54,576	2%	
Number used as Flip Flops	1,103			
Number of Slice LUTs	1,692	27,288	6%	
Number used as logic	1,602	27,288	5%	
Number using O6 output only	960			
Number using O5 output only	315			
Number using O5 and O6	327			
Number used as Memory	42	6,408	1%	
Number used as Single Port RAM	34			
Number using O6 output only	2			

4.1 – Controlul poziției brațului robotic în plan (2D) 51

Number using O5 and O6	32		
Number used as Shift Register	8		
Number using O6 output only	8		
Number used exclusively as route-thrus	48		
Number with same-slice register load	31		
Number with same-slice carry load	17		
Number of occupied Slices	591	6,822	8%
Number of LUT Flip Flop pairs used	1,799		
Number with an unused Flip Flop	799	1,799	44%
Number with an unused LUT	107	1,799	5%
Number of fully used LUT-FF pairs	893	1,799	49%
Number of unique control sets	98		
Number of slice register sites lost to control set restrictions	287	54,576	1%
Number of bonded IOBs	108	218	49%
Number of LOCed IOBs	108	108	100%
IOB Flip Flops	2		
IOB Master Pads	4		
IOB Slave Pads	4		
Number of BUFIO2/BUFIO2_2CLKs	1	32	3%
Number used as BUFIO2s	1		
Number of BUFG/BUFGMUXs	9	16	56%
Number used as BUFGs	9		
Number of DCM/DCM_CLKGENs	2	8	25%
Number used as DCMs	1		
Number used as DCM_CLKGENs	1		
Number of IODELAY2/IODRP2/IODRP2_MCBs	24	376	6%
Number used as IODRP2s	2		
Number used as IODRP2_MCBs	22		
Number of OLOGIC2/OSERDES2s	55	376	14%
Number used as OLOGIC2s	2		
Number used as OSERDES2s	53		
Number of BUFPLLs	1	8	12%
Number of BUFPLL_MCBs	1	4	25%
Number of MCBs	1	2	50%
Number of PLL_ADVs	2	4	50%
Average Fanout of Non-Clock Nets	3.66		

În fig. 4.2 se observă graficul resurselor utilizate din FPGA-ul Spartan-6 de pe placa ATLYS. Se observă că **sistemul a utilizat aproape 50% din porțile FPGA-ului Spartan-6 de pe placa ATLYS**. Acest lucru este datorat și faptului că FPGA-ul Spartan-6 de pe placa ATLYS are porți mult mai puține decât SoC-ul Zynq-7000.

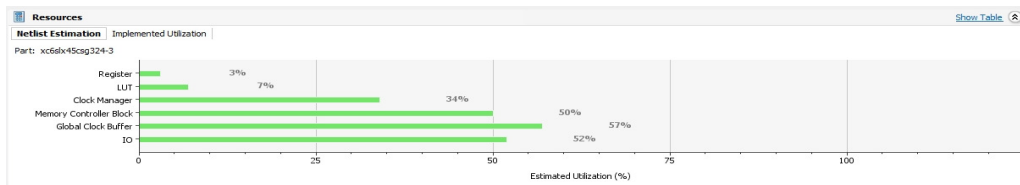


Fig. 4.2 Grafic de utilizare a resurselor pentru sistemul de control al brațului robotic cu FPGA ALTYs prin recunoașterea bulinelor albastre de la articulații

În fig. 4.3 se observă imaginea inițială captată de VmodCAM cu placa cu FPGA ATLYS, apoi afișată pe un monitor conectat la portul HDMI. Implementarea are un grad de dificultate ridicat, deoarece în primul rând a trebuit să se includă trei drivere de interfețe, dintre care nu toate erau comune. Au trebuit implementate interfețele VHDCI (pentru camera stereo), HDMI (pentru monitor) și RS-232 (pentru brațul robotic).

În fig. 4.4 se observă pixeli negrii la bulinele colorate aplicate la articulațiile brațului robotic. Acest lucru înseamnă că algoritmul de detecție de markeri din sistemul creat pe FPGA în VHDL funcționează, dat fiind și faptul că recunoașterea de culoare albastră funcționează.

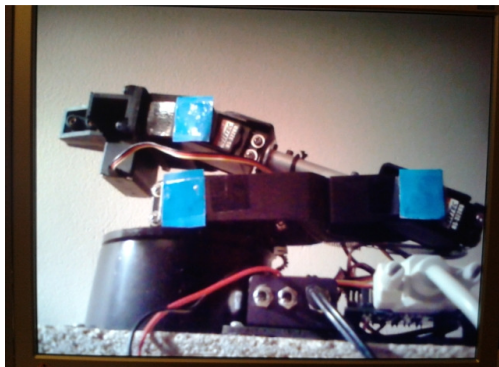


Fig. 4.3 Imagine inițială captată de brațul robotic cu FPGA ATLYS – programare în VHDL



Fig. 4.4 Recunoaștere culorii albastre cu ajutorul FPGA ATLYS – programare în VHDL

În fig. 4.5 se observă diagrama bloc pentru experimentul de detectare a brațului robotic în plan. La partea hardware este conectat un webcam pe portul USB și un braț robotic pe portul RS-232. La partea de software este folosit un filtru RGB pentru culoarea albastră, apoi se efectuează umplerea pixelilor lipsă după aplicarea de filtru pentru a avea pete de culoare uniforme. Apoi se folosește operația de deschidere, care presupune o eroziune urmată de o dilatare, care mai uniformizează extremitățile petelor de culoare. După aceste operații sunt efectuate calcule matematice și este realizată trasarea pe imagine a liniilor și cercurilor ajutatoare. Culoarele sunt unite cu linii și sunt trasate cercuri în jurul culorilor. După ce sunt efectuate calculele matematice, acestea sunt transformate în comenzi pentru robot

(comenzi SCPI). Aceste comenzi apoi se vor trimite pe portul serial utilizând driverul VISA de la National Instruments (NI).

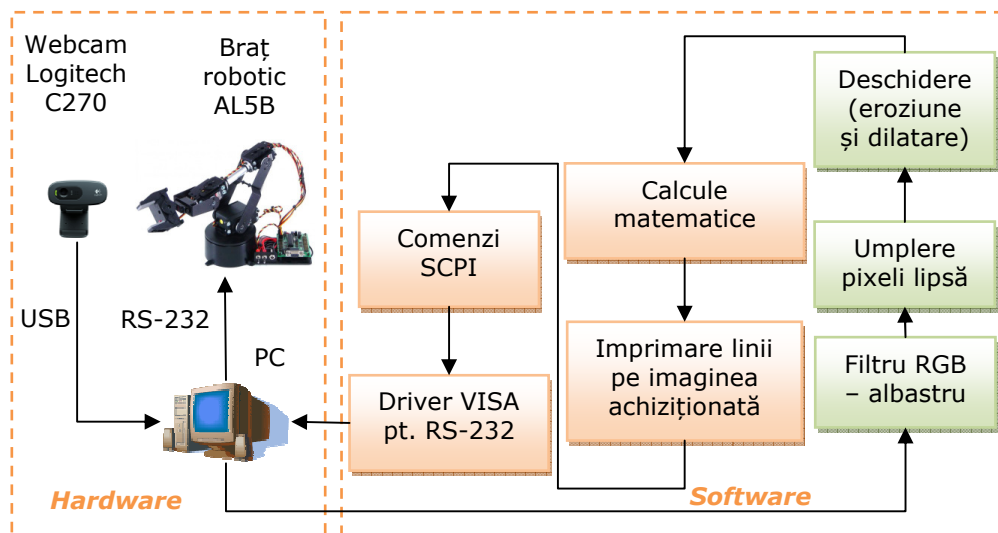


Fig. 4.5 Diagrama bloc pentru experimentul de detectare a brațului robotic în planul 2D – realizat cu LabWindows/CVI și RoboRealm cu recunoașterea culorii albastre

În fig. 4.6 se observă imaginea inițială a brațului robotic captat de camera web pentru a fi prelucrat cu sistemul RoboRealm.

În fig. 4.7 se observă cum se introduce scheletul brațului robotic în PC. În cadrul tezei s-a creat un program în LabWindows/CVI, care comunică în sistem pipeline cu RoboRealm pe portul Ethernet. Acesta numerotează articulațiile brațului robotic și le unește cu linii. Toate aceste desene sunt aplicate de asupra imaginii inițiale. În acest fel se poate recunoaște un braț robotic în spațiu și se poate introduce scheletul acestuia în PC. Această figură este salvată înainte de calibrare, adică încă nu se știe ordinea articulațiilor robotului, ordinea corectă ar fi 0 – bază, 1 – cot, 2 – efector final, iar distanța între bază și efectorul final este trasată cu verde, restul cu albastru. Pentru a afla ordinea articulațiilor brațului robotic se face un proces de calibrare [114], care va fi explicat în cele ce urmează (sau se folosesc culori diferite la articulații).

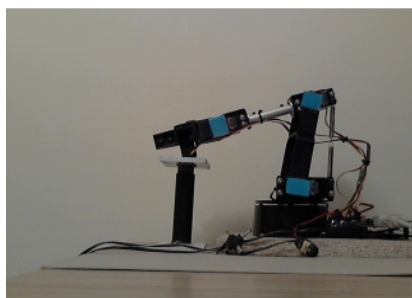


Fig. 4.6 Imaginea inițială cu brațul robotic cu buline de culoare albastră aplicate la încheieturi

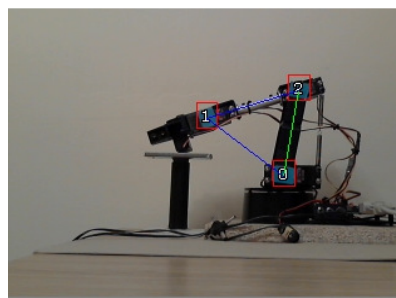


Fig. 4.7 Trasarea scheletului robotului de asupra imaginii inițiale prin recunoașterea articulațiilor cu ajutorul algoritmului de recunoaștere de culori

4.1.2. Utilizarea bulinelor multicolore la articulații

Pentru a deosebi fiecare articulație al unui braț robotic utilizând o singură culoare la articulații, este destul de dificil, deoarece necesită o calibrare anterioară. Prima oară mișcăm motorul de la cotul brațului robotic, astfel singurul punct care se va mișca este bulina de la efectorul final, deoarece mișcarea este exercitată în jurul punctului cotului brațului robotic și baza brațului robotic nu se mișcă. În a doua fază este mișcat motorul de la bază, se exclude bulina de la efectorul final, astfel singurul punct care se va mișca este bulina de la cotul brațului robotic, deoarece mișcarea este efectuată în jurul punctului de la baza brațului robotic [114].

Acest proces adesea poate fi laborios și trebuie repetat la fiecare pornire pentru a deosebi articulațiile între ele. O abordare mult mai simplă este de utiliza culori diferite pentru articulații, astfel se pot recunoaște articulațiile brațului robotic fără o calibrare prealabilă. Numărul de culori recunoscute nu este o problemă, deoarece sistemul poate recunoaște și până la 256 de culori diferite, (cele 8 culori de bază sunt arhisuficiente deoarece și cele mai complexe brațe robotice au 7 articulații, deci sistemul cu buline multicolore este potrivit și în acest caz). La robotul prezentat se utilizează doar 3 articulații. Sistemul se poate extinde și la mai multe articulații folosind algoritmul în mod repetat.

În fig. 4.8 observă diagrama bloc a sistemului cu placa cu FPGA ATLYS cu arhitectura de procesor OpenRISC-32 și sistem de operare Linux.

Se observă cum este conectată la placa dedicată cu FPGA tastatura pe USB, monitorul pe HDMI, camera video pe Ethernet și brațul robotic pe interfața RS-232 pe mufa microUSB cu un convertor microUSB la DB9.

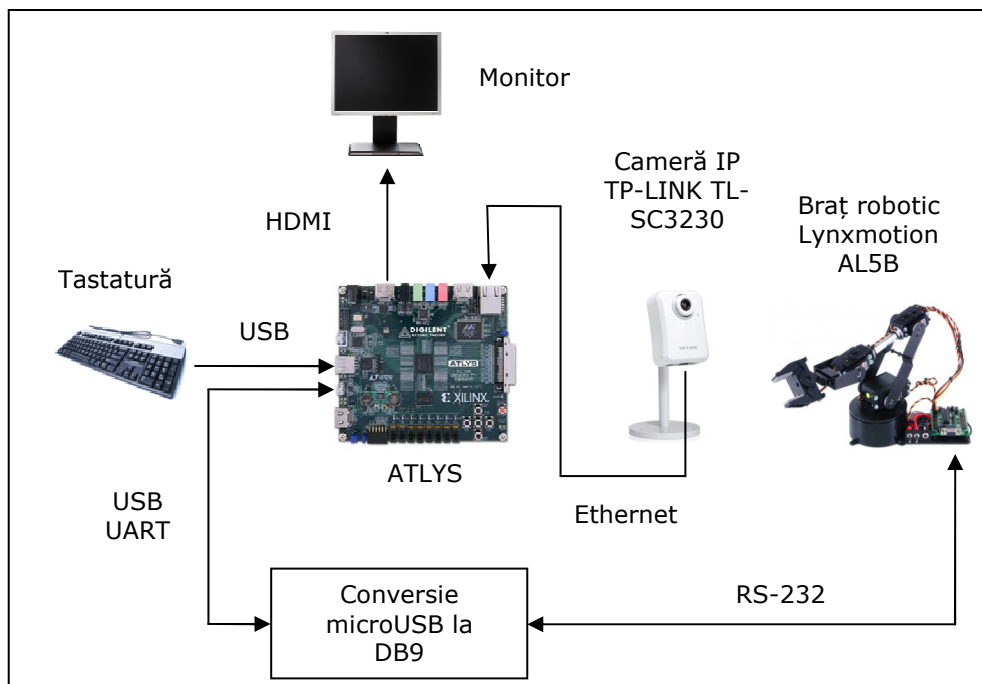


Fig. 4.8 Diagrama bloc pentru sistemul de achiziție de imagine și control braț robotic cu placa de FPGA ATLYS – programare în C sub Linux

În fig. 4.9 se observă imaginea cu brațul robotic Lynxmotion AL5B achiziționată de camera IP conectată la placa ATLYS. Imaginea a fost copiată de pe serverul FTP al plăcii ATLYS pe PC. Sistemul dedicat nu poate vizualiza imaginea, fiind un sistem de operare Linux cu linie de comandă, însă poate stoca și prelucra șirul de biți (informația despre imaginea achiziționată). Acest șir de biți se copiază pe un alt sistem de operare cu interfață grafică, astfel să fie posibilă vizualizarea imaginii ca și o poză, nu doar ca un șir de biți.

În fig. 4.10 se observă programul scris în LabWindows/CVI care ajută la conversia comenzilor RS-232 de la un tip de mufă la altul, în cazul de față de la microUSB (placa ATLYS) la DB9 (braț robotic). Programul ascultă o interfață și trimite comanda la o altă interfață, pe imagine se poate vedea clar comanda achiziționată (Read String) și comanda trimisă (Send String).

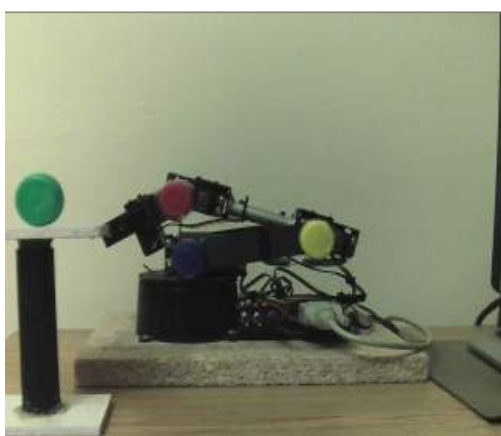


Fig. 4.9 Imagine captată cu placa FPGA ATLYS pe care rulează sistem de operare Linux

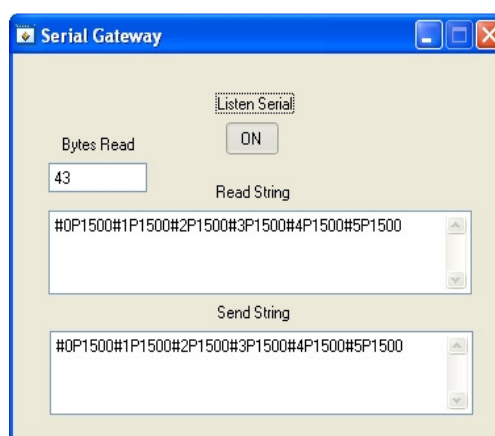


Fig. 4.10 Interfața grafică în LabWindows/CVI a programului de conversie a comenzilor UART de la microUSB la DB9

În fig. 4.11 se observă diagrama bloc pentru experimentul de detectare a brațului robotic în planul 2D. Precum se observă la partea de hardware este conectat un webcam pe portul USB și un braț robotic pe portul RS-232. La partea de software este un filtru RGB pentru culorile albastru, galben și roșu, apoi sunt puse limite la mărimea petelor de culoare pentru a filtra petele de culoare mici, astfel să se detecteze doar culorile puse în punctele cheie ale brațului robotic nu și alte culori asemănătoare cu cele folosite pentru robot, apoi se efectuează calculul punctului central sau punctul gravitațional al petei de culoare detectat. După aceste operații sunt efectuate calcule matematice și trasarea pe imagine a liniilor și cercurilor ajutătoare. Culorile sunt unite cu linii și sunt trasate cercuri în jurul culorilor. Centrul gravitațional ajută să fie unite culorile prin centrul lor. După sunt efectuate calculele matematice, acestea sunt transformate în comenzi pentru robot (comenzi SCPI). Aceste comenzi se vor trimite apoi pe portul serial utilizând driverul VISA.

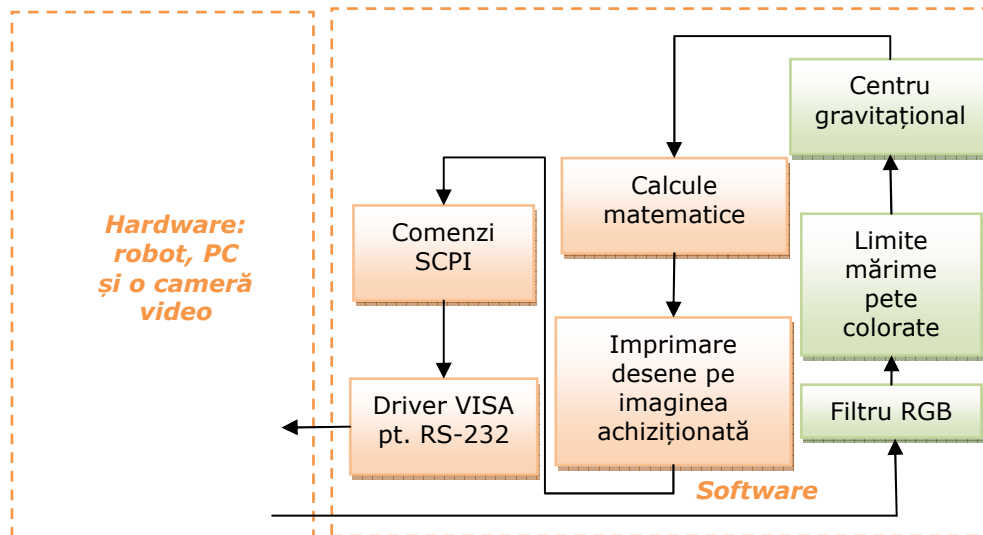


Fig. 4.11 Diagrama bloc pentru sistemul de detectare al brațului robotic în planul 2D – realizat cu LabWindows/CVI și RoboRealm cu recunoaștere de culori multiple

În fig. 4.12 se observă imaginea inițială cu brațul robotic achiziționată cu programul RoboRealm. Se observă și punctul țintă unde trebuie să ajungă efectul final al brațului robotic. Punctul țintă a fost indicat cu ajutorul mouse-ului și este marcat cu o cruce roșie. În această situație fiecare articulație al brațului robotic are o altă culoare. Acest lucru ajută la deosebirea articulațiilor, astfel nu mai este necesar de o calibrare prealabilă, precum era în cazul când toate articulațiile erau marcate cu aceeași culoare.

În fig. 4.13 se observă trasarea liniilor și cecurilor ajutătoare pentru a calcula cât trebuie mișcat fiecare motor al brațului robotic, astfel să ajungă efectul final în punctul țintă, la crucea roșie, indicată de mouse.

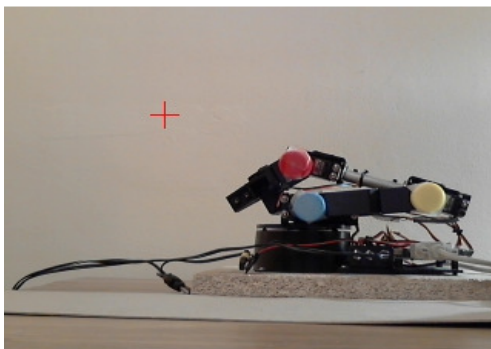


Fig. 4.12 Imaginea inițială a brațului robotic cu indicarea punctului țintă cu mouse-ul

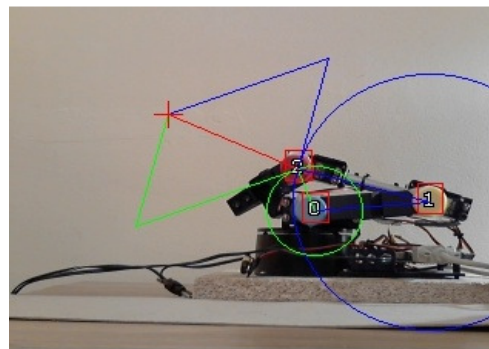


Fig. 4.13 Trasarea liniilor ajutătoare pentru calculul distanțelor pentru ca efectul final al brațului robotic să ajungă în punctul țintă

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

4.2.1. Implementarea în LabWindows/CVI și RoboRealm

Pentru recunoașterea brațului robotic în 3D, sistemul este similar ca și pentru 2D, numai algoritmul trebuie duplicat pentru ambele imagini. O diferență majoră este însă, că în loc de buline colorate, s-au folosit dopuri colorate. Motivul este destul de simplu, în momentul când brațul robotic este rotit, bulinele nu mai sunt vizibile, dar dopurile sunt, deoarece acestea sunt tridimensionale. O bulină aplicată la punctele cheie ale brațului robotic nu s-ar vedea în momentul când brațul robotic nu mai este paralel cu planul camerelor.

Pe figurile următoare s-a generat în MATLAB imaginea 3D al brațului robotic din cele două imagini filmate de cele două camere.

În fig. 4.14 se observă imaginea filmată de camera stânga, în fig. 4.15 putem vedea imaginea filmată de camera dreaptă. Se poate observa că cele două imagini sunt puțin decalate.

În fig. 4.16 se văd imaginile stânga și dreapta suprapuse cu folosirea culorile turcoaz și roșu. S-a generat o poză, apropiată de imaginile 3D care se pot vizualiza cu ochelari speciali.

În fig. 4.17 se observă harta de adâncime din identificarea blocurilor de bază generată cu programul MATLAB din cele două imagini.

În fig. 4.18 este prezentată identificarea de blocuri cu programare dinamică generată de MATLAB. În fig. 4.19 se observă identificarea de blocuri piramidă de nivelul patru generată de MATLAB. În fig. 4.20 este prezentată piramida de nivelul trei cu programare dinamică generată de MATLAB. În fig. 4.21 se observă piramida cu programare dinamică și acuratețe mai fină ca un pixel.

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.

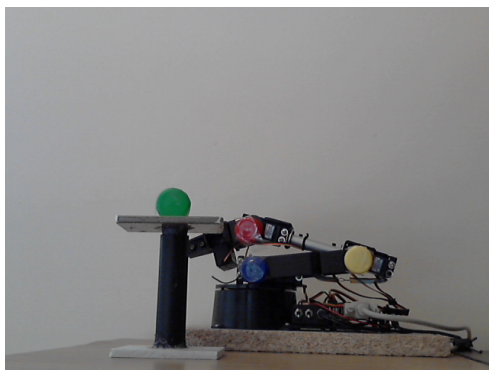


Fig. 4.14 Imaginea inițială a brațului robotic filmată de camera din partea stângă

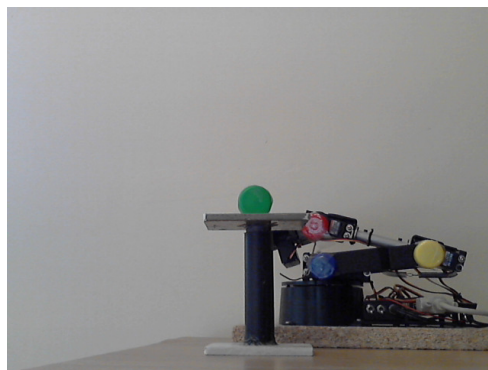


Fig. 4.15 Imaginea inițială a brațului robotic filmată de camera din partea dreaptă

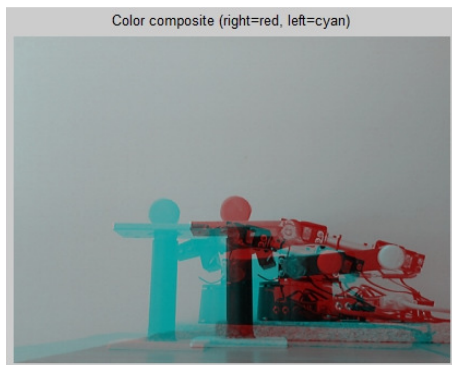


Fig. 4.16 Imaginile inițiale suprapuse în MATLAB pentru ochelari 3D folosind culorile turcoaz și roșu

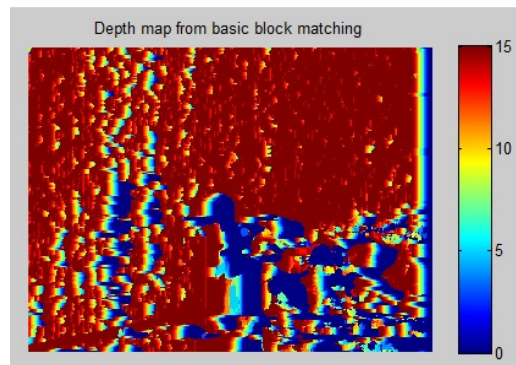


Fig. 4.17 Harta de adâncime generată de MATLAB

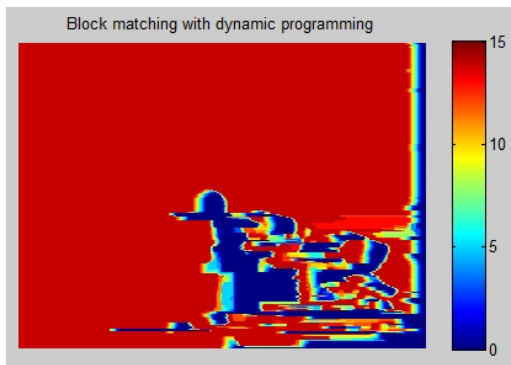


Fig. 4.18 Identificarea de blocuri în MATLAB

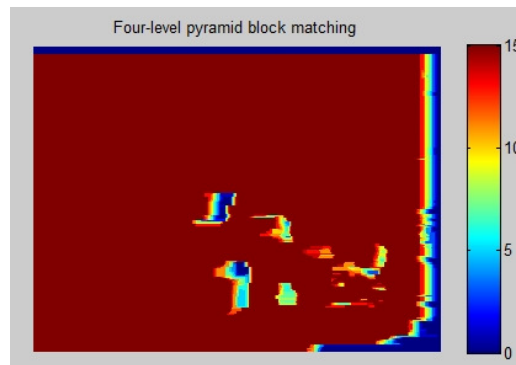


Fig. 4.19 Identificarea de blocuri piramidale de nivelul 4 în MATLAB

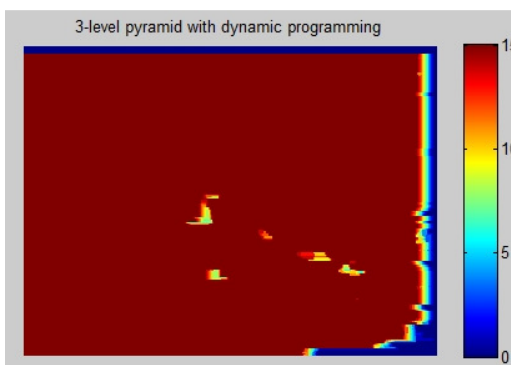


Fig. 4.20 Identificarea de blocuri piramidale de nivelul 3 în MATLAB

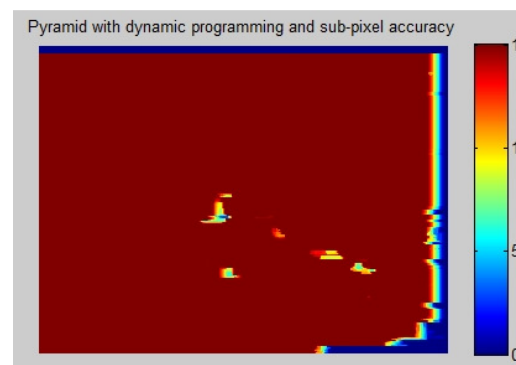


Fig. 4.21 Identificarea de blocuri piramidale prin programare dinamică în MATLAB

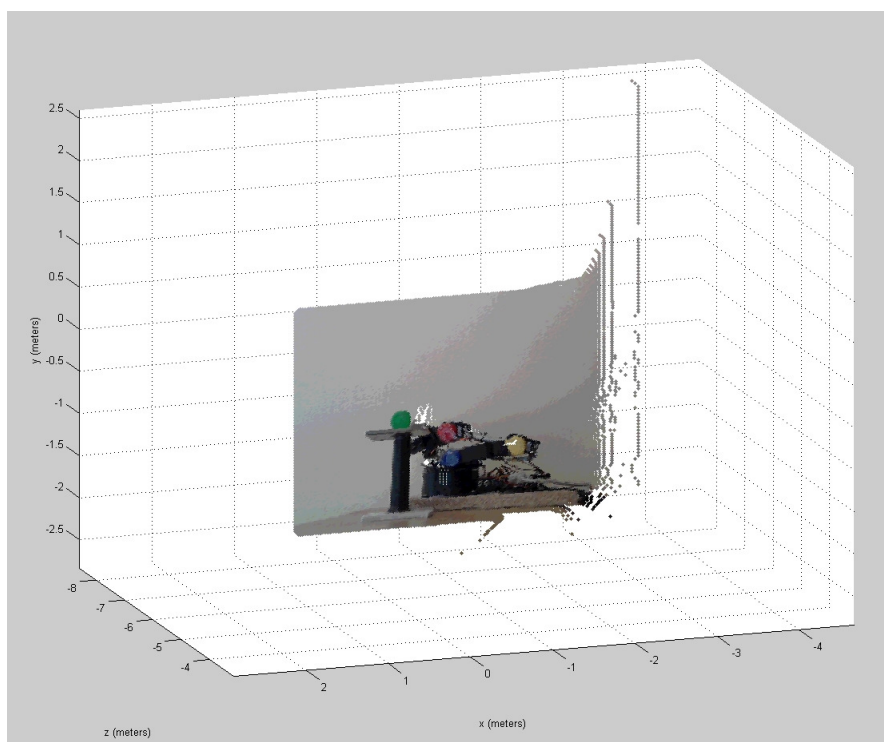


Fig. 4.22 Imaginea generată 3D în MATLAB

În fig. 4.23 se observă imaginea stereo captată de camera VmodCAM conectată la placa ATLYS cu FPGA-ul Spartan-6. Sistemul are diagrama bloc similară cu cea din fig. 4.1. Sistemul este realizat în cod VHDL. Placa are conectată camera VmodCAM pe portul VHDCI, monitor pe port HDMI, iar robotul este controlat pe portul RS-232 utilizând un Pmod RS232.

În fig. 4.24 se observă imaginea monitorului pe care e captat brațul robotic cu ajutorul camerei stereo. Se observă destul de bine decalarea între cele două imagini, care este proporțională cu distanța dintre camere.



Fig. 4.23 Achiziția imaginii stereo cu placa FPGA ATLYS – programare în VHDL



Fig. 4.24 Imaginea stereo a brațului robotic pe monitor captat cu placa FPGA ATLYS și camera stereo VmodCAM

În fig. 4.25 se observă imaginea stereo cu desenele ajutătoare aplicate direct pe imaginea inițială, cu ajutorul acestora se poate calcula cât trebuie mișcat fiecare motor astfel încât brațul robotic să ajungă la punctul țintă, adică efectul final (marcat cu dopul roșu) să ajungă în proximitatea dopului verde. Această primă abordare folosește un algoritm combinat din algoritmul de calcul de distanțe în planul 2D al brațului robotic și algoritmul de calcul al distanțelor stereo. Algoritmul de calcul distanțe în planul 2D calculează distanța pentru mișcarea pe sistem de coordonate YOZ, adică în jurul punctului 0 pentru axa OY, iar în jurul punctului 1 pentru axa OZ. Pentru axa OX se folosesc două imagini și se calculează distanța în adâncime folosind algoritmul de calcul al distanțelor stereo.

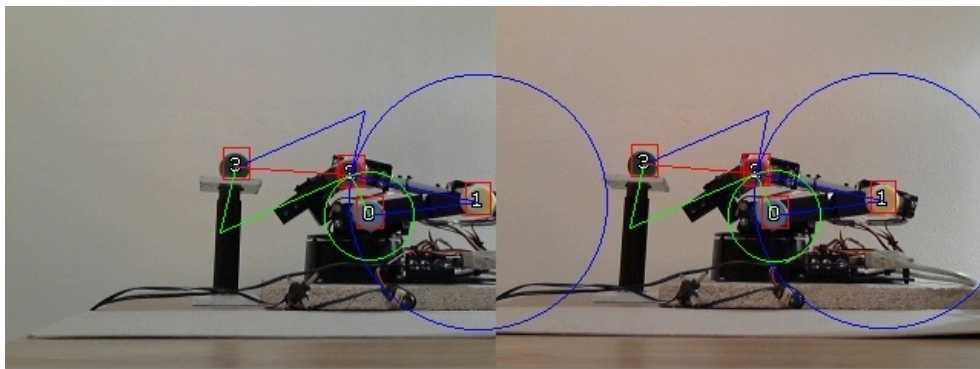


Fig. 4.25 Imaginea cu trasarea liniilor ajutătoare de calcul a poziției brațului robotic în 3D cu două algoritme combinate: calculul distanței în planul 2D și calculul distanței stereo

În fig. 4.26 este prezentat un sistem asemănător ca și cel anterior, numai că nu se mai folosesc doi algoritmi de calcul pentru calculul distanțelor, ci există un paralelipiped, prin care se poate calcula distanța de mișcare celor trei motoare, adică motorul de bază (0) pe axa OY, motorul de cot (1) pe axa OZ, iar motorul de bază (0) în adâncime pe axa OX.

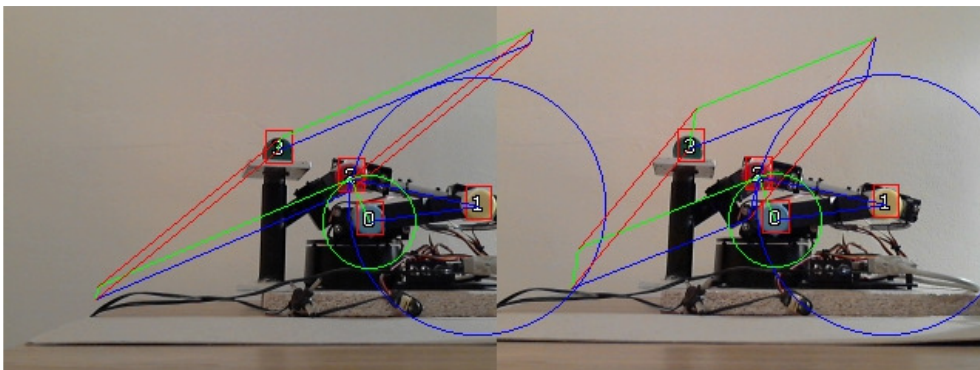


Fig. 4.26 Imaginea cu trasarea liniilor ajutătoare de calcul a poziției brațului robotic în 3D utilizând un singur algoritm

4.2.2. Implementarea în LabVIEW și NI Vision Development Module

În primă fază sistemul a fost implementat în LabVIEW. Ca și sistem de detecție de markeri a fost utilizat NI Vision Development Module.

În fig. 4.27 este prezentată diagrama bloc pentru experimentul de detecție al brațului robotic în spațiul 3D. La partea de hardware sunt utilizate două camere web conectate la porturi USB și brațul robotic conectat la portul RS-232. La partea de software sunt definite limite pentru detectarea culorii albastre, galbene, roșii și verzi, apoi sunt înlăturate particulele prea mici pentru eliminarea zgomotelor sau a detecțiilor pozitive false, apoi particulele rămase sunt analizate și sunt calculate centrele de masă sau centrele culorilor detectate. Se fac din nou calcule matematice și se trasează pe imagine cercuri și linii ajutătoare pentru calcule. Culorile sunt unite cu linii și sunt trasate cercuri în jurul culorilor. Centrul de masă ajută să fie unite culorile prin centrul lor. După calcule matematice sunt convertite valorile de deplasare al fiecărui motor în comenzi pentru robot (comenzi SCPI), apoi sunt trimise la robot cu ajutorul driverului VISA.

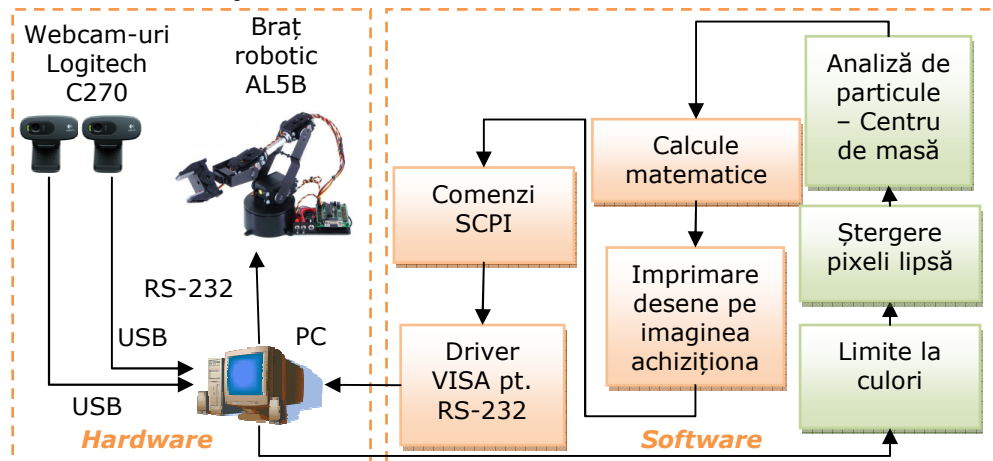


Fig. 4.27 Diagrama bloc pentru experimentul de detecție a brațului robotic în 3D – realizat cu LabVIEW și NI Vision Development Module

În fig. 4.28 se observă interfața programului realizat în LabVIEW, care recunoaște și controlează brațul robotic cu ajutorul detecției de buline colorate plasate la articulațiile acestuia. Pe interfață sunt butoanele pentru controlul brațului robotic de tip Lynxmotion AL5x. Lângă aceste butoane sunt parametrii de setare a portului serial pentru comunicarea cu brațul robotic, astfel este setată comunicația pe COM1, rata de transfer 115200 baud, 8 biți de date, fără paritate, 1 bit de stop și fără control al cursului datelor. Mai jos se poate selecta camerele video, în caz că sunt conectate la PC mai mult de două camere. Pe primele imagini poate vedea imaginea stereo cu brațul robotic și liniile ajutătoare trasate pe imagine, iar apoi mai jos se poate vedea filtrarea de culori albastre, galbene, roșii și verzi pentru detectarea dopurilor aplicate la articulațiile robotului. Cu butoanele glisante se poate seta pentru fiecare culoare valorile minime și maxime ale filtrului HSV (hue = nuanță, saturation = saturație, value = valoare). Se folosește acest filtru HSV, deoarece în multe cazuri funcționează mai bine decât filtrul RGB.

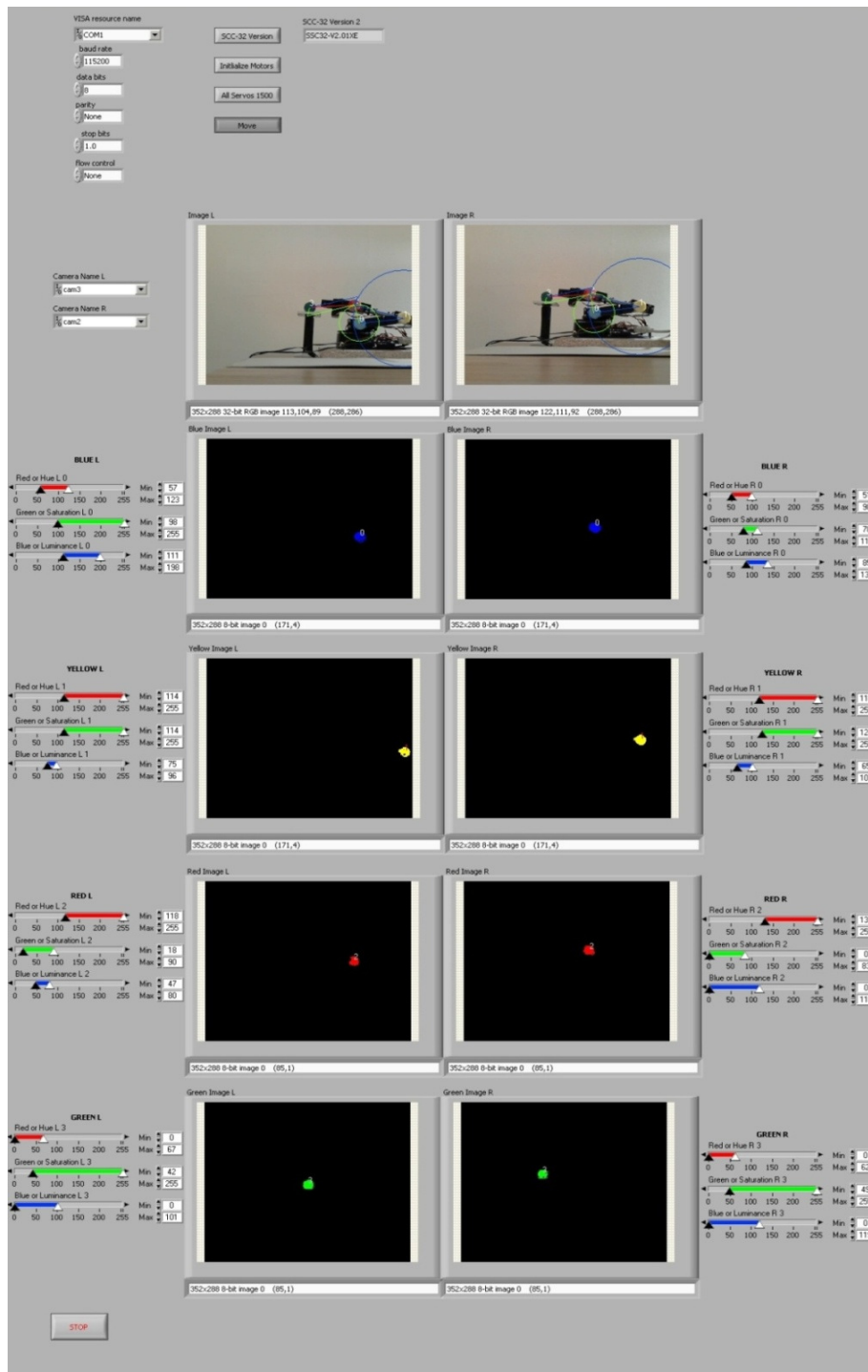


Fig. 4.28 Panoul frontal în LabVIEW pentru controlul brațului robotic în 3D

În fig. 4.29 se observă imaginea stereo mărită al brațului robotic. Imaginea fost mărită pentru a scoate în evidență cum sunt detectate culorile dopurilor colorate aplicate pe articulații și cum sunt trasate liniile și cercurile ajutătoare pentru a calcula cât trebuie mișcat fiecare motor, astfel încât efectorul final al brațului robotic să ajungă în punctul țintă, adică dopul roșu în proximitatea dopului verde.

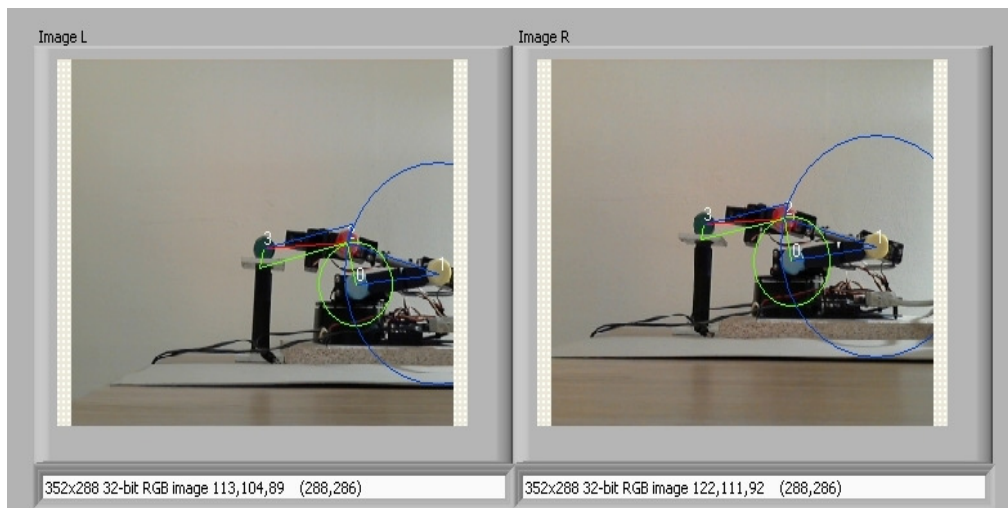


Fig. 4.29 Imagine stereo pentru brațul robotic Lynxmotion AL5B cu desene ajutătoare pentru calculul poziției în 3D [imaginea mărită din interfața LabVIEW]

4.2.3. Implementarea în LabWindows/CVI și NI Vision Development Module

O altă abordare este ca sistemul să fie implementat în LabWindows/CVI. Sistemul de detecție de marcheri a fost implementat similar ca și în aplicația LabVIEW, folosind NI Vision Development Module, numai că nu se realizează o diagramă bloc ca și în LabVIEW (acest fiind un limbaj de programare grafic), ci se scrie codul în ANSI C.

În fig. 4.30 este prezentată diagrama bloc pentru experimentul de detecție al brațului robotic în spațiul 3D. La partea de hardware sunt două camere web conectate pe porturi USB și brațul robotic este conectat pe portul RS-232. La partea de software sunt implementate limite pentru culorile albastru, galben, roșu și verde, mai este implementat și un filtru pentru mărimea petelor de culoare ca să se elimine particulele mici (zgomot sau detecție pozitivă falsă), apoi se numără particulele de culori detectare, apoi particulele sunt măsurate. Se fac din nou calcule matematice și se trasează pe imagine cercuri și linii ajutătoare pentru calcule. Culorile sunt unite cu linii și sunt trasate cercuri în jurul culorilor. După calcule matematice se convertesc valorile, care definesc cantitatea de mișcare al fiecărui motor, în comenzi pentru roboti (comenzi SCPI) apoi sunt trimise acestea la robot cu ajutorul driverului VISA.

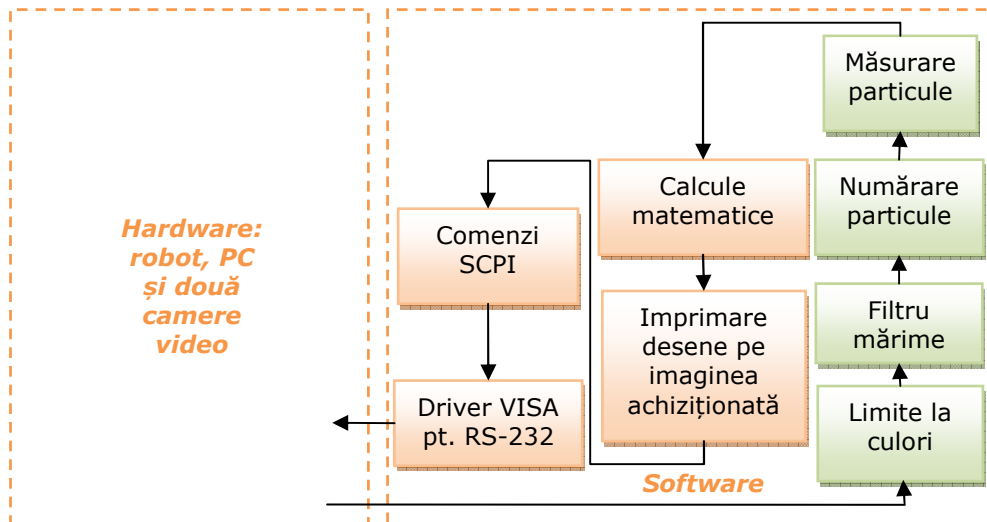


Fig. 4.30 Diagrama bloc pentru experimentul de detectare a brațului robotic în 3D – realizat cu LabWindows/CVI și NI Vision Development Module

În fig. 4.31 se observă sistemul implementat și în LabWindows/CVI. Practic este similar cu cel din fig. 4.28, numai în loc de realizarea unei diagrame bloc în LabVIEW, a fost scris codul în ANSI C, folosind mediul de dezvoltare LabWindows/CVI. Sistemul de prelucrare de imagini folosește aceleași biblioteci NI Vision Development Module, însă sunt apelate în mod diferit.

Pe interfața grafică din programul realizat în LabWindows/CVI, sunt prezente cele patru butoane pentru controlul brațului robotic, ca și pe panoul frontal din aplicația realizată în LabVIEW. Sunt prezente trei butoane glisante, pentru mișcarea manuală, al brațului robotic, într-o poziție dorită. Există posibilitatea de selectare a camerelor de la care se face achiziția, iar cu butoanele "Grab L" și "Grab R" se pornește achiziția pe cele două camere video, stânga (L = left) și dreapta (R = right). Sunt prezente mai multe butoane glisante pentru setarea valorilor minime și maxime ale filtrului HSV pentru fiecare culoare și pentru fiecare cameră în parte, stânga și dreapta. Precum se observă există un număr dublu de butoane glisante, față de aplicația realizată în LabVIEW, deoarece, spre deosebire de LabVIEW, în LabWindows/CVI nu există buton glisant dublu prin care se poate seta valoarea minimă și maximă la un parametru HSV, astfel a trebuit să se folosească un buton glisant pentru valoarea minimă și alta pentru valoare maximă al fiecărui parametru HSV pentru fiecare culoare și pentru fiecare cameră în parte.

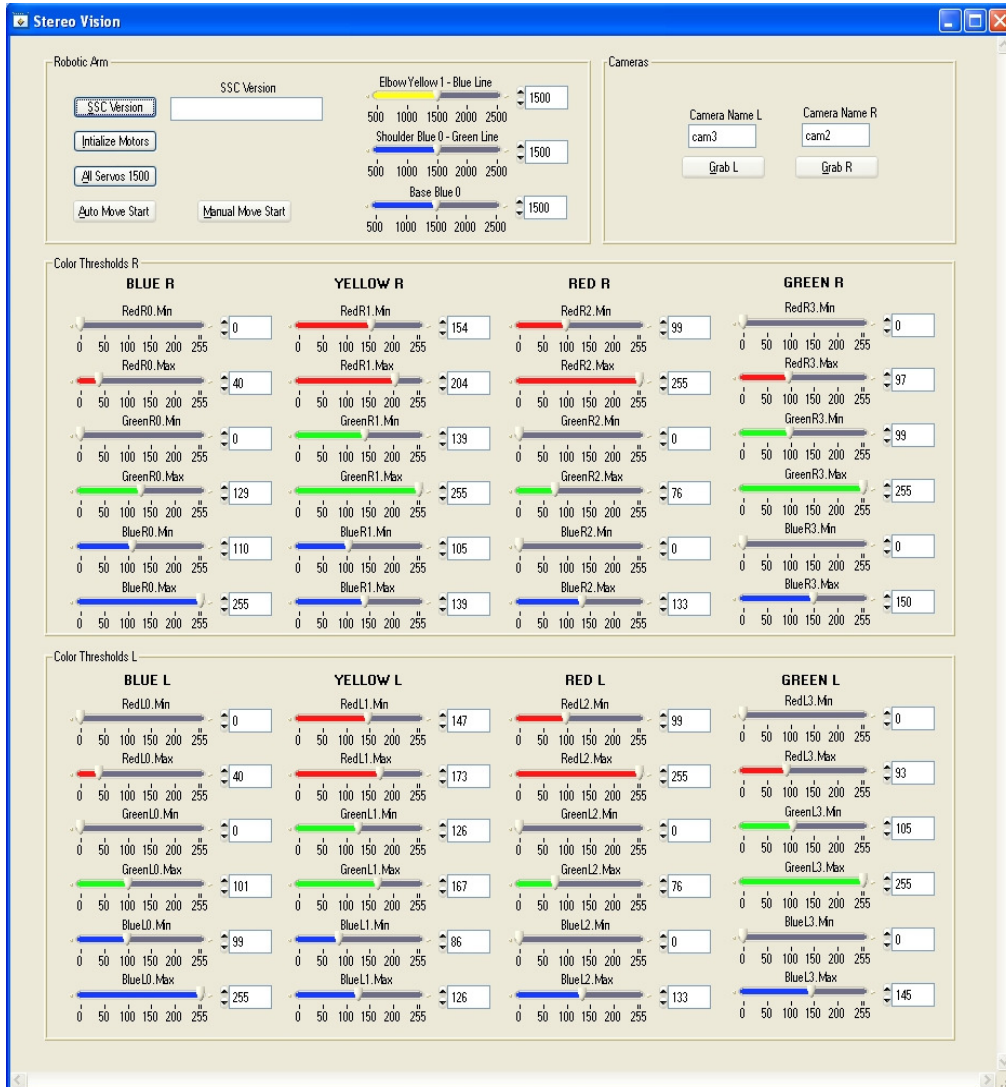


Fig. 4.31 Interfața grafică în LabWindows/CVI pentru controlul brațului robotic în 3D

În fig. 4.32 se observă imaginea stereo achiziționată cu cele două camere cu ajutorul mediului de programare LabWindows/CVI. Este o imagine asemănătoare cu imaginea 4.29, numai că de data asta este realizat în LabWindows/CVI. Aici la fel este o imagine stereo cu liniile ajutătoare pentru a calcula cât trebuie mișcat fiecare motor în parte, astfel încât efectorul final, adică dopul roșu, să ajungă în proximitatea punctului țintă, adică dopul verde.

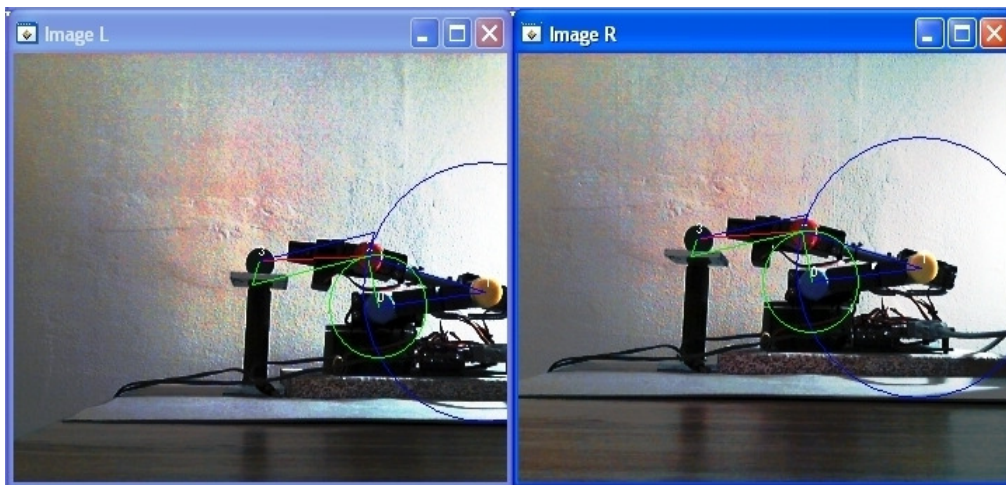


Fig. 4.32 Trasarea liniilor ajutătoare pentru controlul brațului robotic în 3D în LabWindows/CVI

4.2.4. Implementarea în C și OpenCV sub Linux

Sistemul de control al brațului robotic, care folosește recunoașterea dopurilor colorate plastice la articulațiile robotului cu ajutorul camerelor video se poate implementa și în limbaj C, integrând biblioteca OpenCV, apoi aplicația este executată pe sistemul de operare Linux. Acest lucru este util deoarece pe multe plăci dedicate se poate instala un sistem Linux chiar și cu interfață grafică. Sistemul de operare Linux cu interfață grafică se poate instala atât pe un sistem dedicat cu microprocesor, de genul Raspberry PI, sau pe un sistem dedicat cu SoC Zynq-7000, care conține un microprocesor de tip ARM Cortex-9 Dual-core și un FPGA de generația a 7-a, care este de obicei de tip Atrix-7.

În fig. 4.33 se observă aplicația scrisă în C cu biblioteca OpenCV și interfață QT pe Linux. Sistemul de detecție de marcheri folosește metoda cercurilor Hugh, care este o funcție în OpenCV utilizată pentru detectarea formelor circulare. Sistemul este similar cu cel de detecție pixeli rămași după aplicarea filtrului de culoare, însă în cazul de față după filtrul HSV se detectează petele cele mai mari circulare, apoi sunt numerotate. Se observă că și pentru acest sistem, sunt implementate butoane glisante, pentru a seta valorile minime și maxime ale parametrilor HSV. Se pot vedea culorile detectate pe ferestrele cu butoane glisante, iar în imaginea inițială sunt numerotate dopurile colorate de la articulațiile recunoscute.

În fig. 4.34 este prezentată imaginea mărită, unde este vizibil că dopurile colorate de la articulații au fost recunoscute. Acestea sunt marcate cu cercuri, ca urmare a folosirii algoritmului Hugh.

Ca să se obțină un sistem de control 3D, întregul sistemul trebuie duplicat.

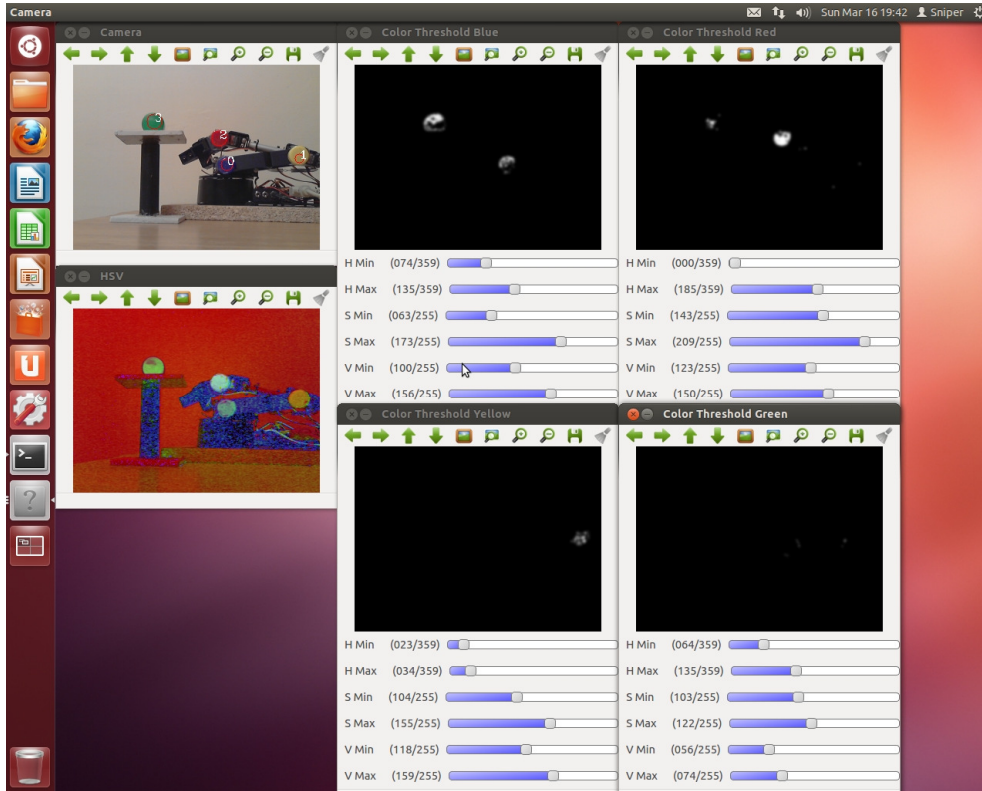


Fig. 4.33 Detectarea articulațiilor brațului robotic cu metoda cercurilor lui Hugh – program creat în C sub Linux

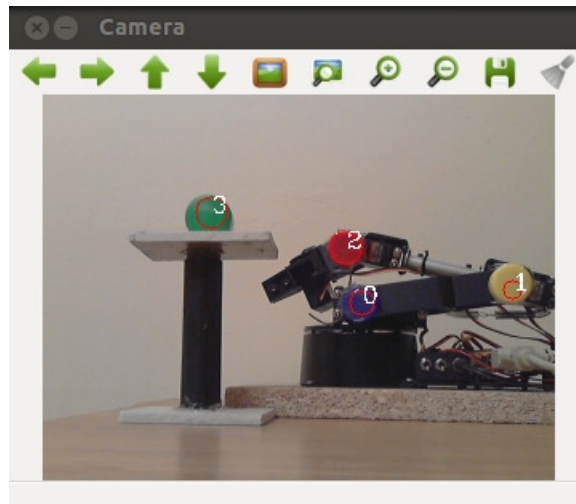


Fig. 4.34 Imagine mărită din program pentru detectarea articulațiilor cu ajutorul metodei cercurilor lui Hugh

În fig. 4.35 este prezentată diagrama bloc pentru experimentul de detectare al brațului robotic în spațiu. La partea de hardware sunt două camere web conectate pe portul USB și brațul robotic conectat pe portul RS-232. La partea de software se face egalizare de histogramă pentru uniformizarea imaginii, apoi s-a folosit un filtru HSV (hue = nuanță, saturation = saturație, value = valoare), apoi sunt detectate contururile petelor de culoare, apoi sunt calculate centrul cercurilor, care să fie trasate în jurul petelor de culoare detectate. Se fac din nou calcule matematice și se trasează pe imagine cercuri și linii ajutătoare pentru calcule. Culoarele sunt unite cu linii și sunt trasate cercuri în jurul culorilor. Centrul cercurilor din jurul culorilor detectate ajută să fie unite culorile prin centrul lor. După calcule matematice sunt convertite valorile de deplasare al fiecărui motor în comenzi pentru roboti (comenzi SCPI) apoi sunt trimise la robot cu ajutorul driverului RS-232 din Linux.

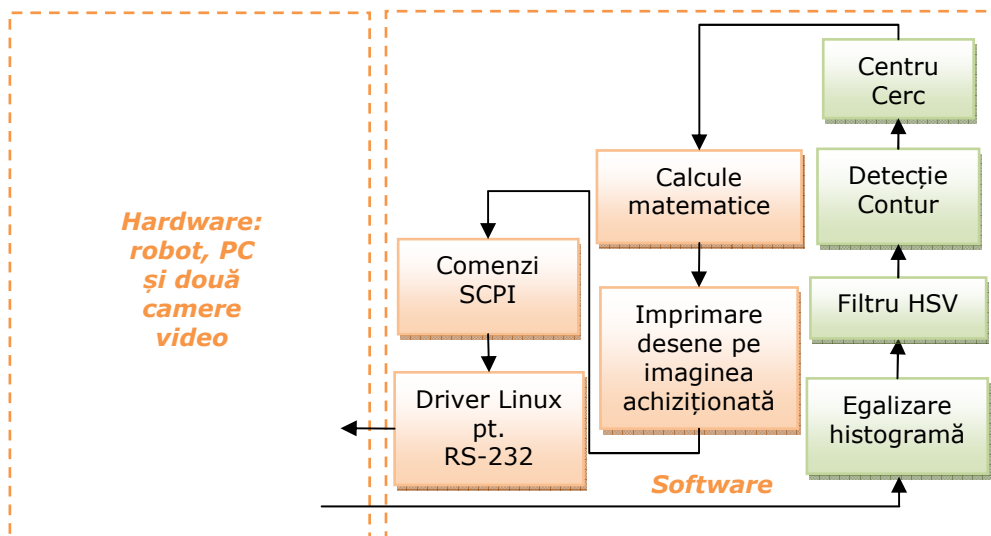


Fig. 4.35 Diagrama bloc pentru experimentul de detectare a brațului robotic în 3D – realizat în Linux în C și OpenCV

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 utilizând biblioteca OpenCV pe sistemul de operare Linux.



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

În fig. 4.37 se observă aplicația de control al brațului robotic în 3D, scrisă în C cu biblioteca OpenCV pe sistemul de operare Linux utilizând metoda de detecție de contur. Se observă sistemul duplicat, deoarece imaginea este captată de pe două monitoare puse în paralel. Pe imagine se observă imaginea inițială pentru camera stângă și dreaptă, sub acesta este egalizarea de histogramă, apoi este filtrarea HSV, iar apoi este detecția de contur pentru fiecare culoare în parte cu butoane glisante pentru setarea parametrilor minimi și maximi al filtrului HSV. La detecția de contur este memorată coordonata centrului cercului care înconjoară conturul dopului colorat recunoscut. Această coordonată este transferată pe imaginea inițială, iar apoi primește un număr de ordine, de obicei, albastru – 0, galben – 1, roșu – 2 și verde – 3. După aceea aceste coordonate sunt unite cu linii, apoi sunt trasate cercurile în jurul bazei (dopul albastru marcat cu 0) și în jurul cotului (dopul galben marcat cu 1) brațului robotic. După aceea sunt trasate și alte linii pentru a calcula cât de mult trebuie rotit fiecare motor în parte, astfel ca efectorul final al brațului robotic să ajungă în punctul țintă.

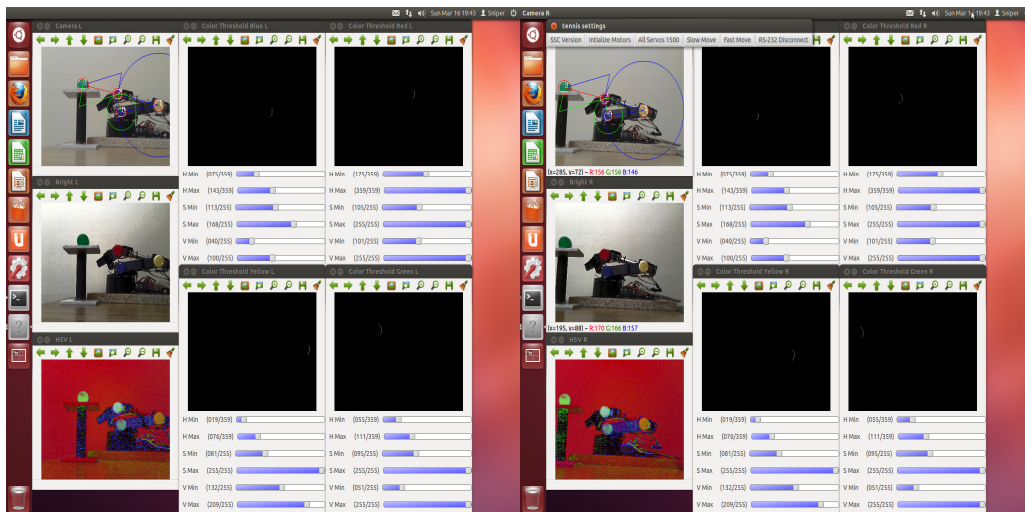


Fig. 4.37 Aplicația pentru controlul brațului robotic Lynxmotion AL5B în 3D cu detecție de contur în C cu OpenCV sub Linux

În fig. 4.38 este prezentată imaginea stereo al brațului robotic Lynxmotion AL5B cu desenele ajutătoare pentru calculul distanțelor de mișcare, astfel ca efectorul final al brațului robotic să ajungă în punctul țintă. Această imagine este o imagine mărită din figura anterioară, este de asemenea asemănătoare cu fig. 4.29 și cu fig. 4.32 numai că e scrisă în C cu biblioteca OpenCV pe sistemul de operare Linux.

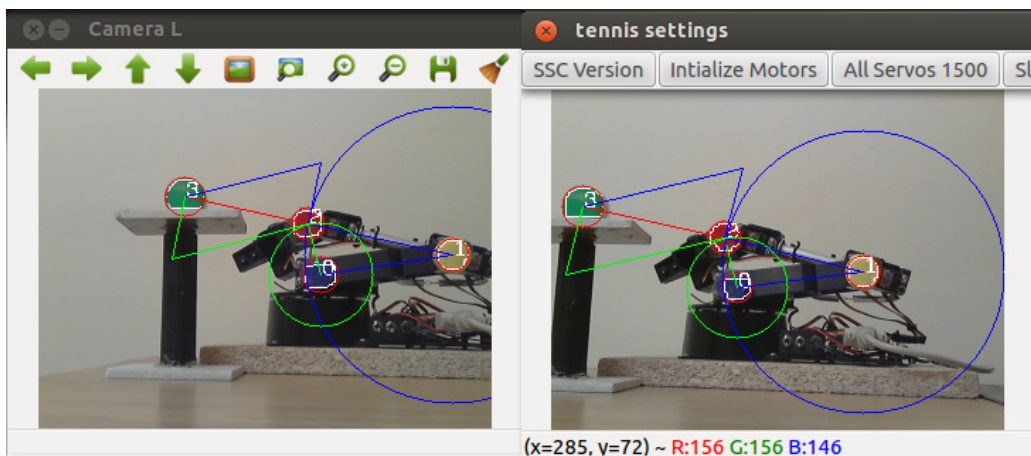


Fig. 4.38 Imaginea mărită din aplicație de controlul a brațului robotic Lynxmotion AL5B în C cu OpenCV sub Linux cu trasarea liniilor pentru calculul poziției brațului robotic

În fig. 4.39 se observă experimentul de control al brațului robotic SCORBOT-ER III 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 utilizând biblioteca OpenCV pe sistemul de operare Linux.

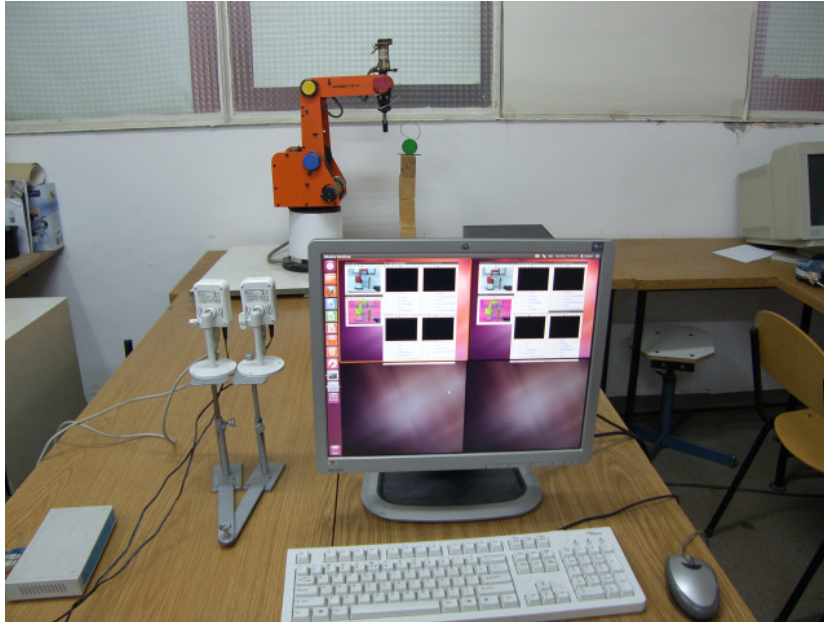


Fig. 4.39 Experimentul de control al brațului robotic SCORBOT-ER III în 3D în C cu OpenCV sub Linux utilizând camere IP

4.3. Controlul robotului industrial SCORBOT-ER III

În fig. 4.40 se observă programul de control al brațului robotic SCORBOT-ER III cu ajutorul camerelor video, folosind recunoașterea de culoare a dopurilor colorate plasate la articulații. Practic este un program similar cu cel din fig. 4.37, numai că este realizat pentru robotul SCORBOT-ER III.

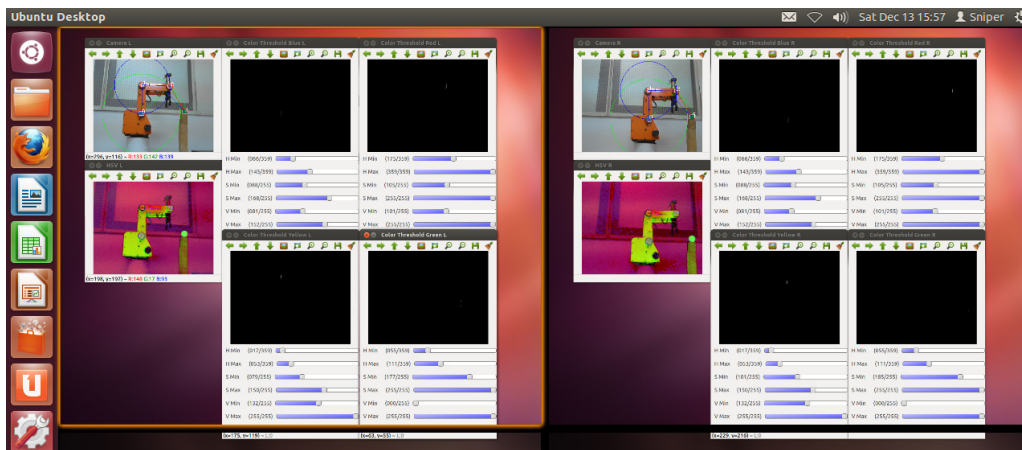


Fig. 4.40 Aplicația pentru controlul brațului robotic SCORBOT-ER III în 3D cu detecție de contur în C cu OpenCV sub Linux

La fel ca și la alte programe, s-a mărit partea de recunoaștere a dopurilor colorate și de trasare a liniilor ajutătoare pentru calculul distanțelor. Această figură (fig. 4.41) este similară cu fig. 4.38, numai că este făcută pentru brațul robotic SCORBOT-ER III. Această figură evidențiază și faptul că algoritmul este general, funcționează și la un robot mai mare, cum ar fi SCORBOT-ER III și funcționează și atunci când direcția robotului este din partea stângă spre partea dreaptă (ca și în cazul de față). Pe figurile anterioare robotul Lynxmotion AL5B era montat mecanic astfel încât să se miște de pe partea dreaptă spre partea stângă.

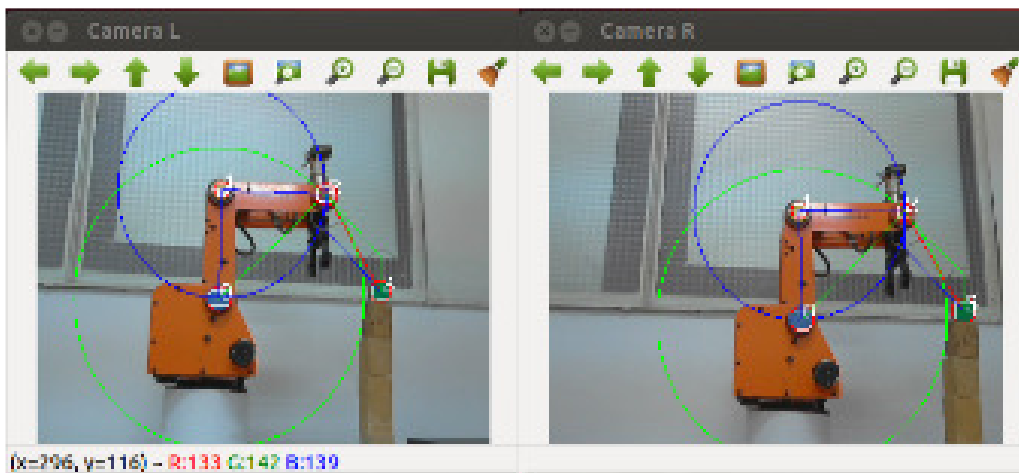


Fig. 4.41 Imaginea mărită din aplicație de controlul a brațului robotic SCORBOT-ER III în C cu OpenCV sub Linux cu trasarea liniilor pentru calculul poziției brațului robotic

În fig. 4.42 se observă diagrama bloc al sistemului de control al brațului robotic cu Raspberry PI și camere web. S-a utilizat Raspberry PI modelul B, care are două porturi USB, astfel a trebuit să se folosească și un HUB USB să se multiplexeze porturile USB. Singurul lucru la care a necesitat o atenție sporită era să se folosească un HUB USB cu alimentare externă și care suportă un curent suficient de mare, pentru a nu mai sustrage curent prin USB de la Raspberry PI. Un asemenea HUB USB este D-Link H4. Cu HUB-ul utilizat s-au obținut în total cinci porturi USB, astfel s-au putut conecta două camere web, un mouse, o tastatură și un convertor USB la serial. Cu acest convertor USB la serial s-au putut trimite comenzile pentru controlul brațului robotic. Raspberry PI are port serial la pini, însă acolo mai trebuie și un transceiver, iar astfel se complică mult prea mult experimentul. O alegere bună a fost acest convertor USB la serial.

Placa mai are și un port HDMI prin care s-a putut conecta monitorul. De obicei doar televizoarele au port HDMI, monitoarele au doar port DVI, pentru care se găsește convertor HDMI la DVI. Cei de la fundația Raspberry PI au făcut și un adaptor PI-VIEW, cu ajutorul căreia semnalul HDMI se poate transforma și la VGA, pentru monitoare analogice sau pentru proiectoare, mai vechi. Adaptorul este unul activ, deci se alimentează din portul HDMI, astfel consumul pe Raspberry PI va crește (trebuie ținut cont de acest aspect).

Un lucru foarte interesant în experiment este și folosirea Pmod-ului RS232, care de fapt conține un circuit integrat, care este un driver pentru comunicația serială. Pentru o comunicație serială trebuie să avem un nivel de tensiune între 3.3V și 12V. În situația de față brațul robotic funcționează fiind comandat doar de PC,

care are niveluri de tensiune suficient de mari. Când brațul robotic este comandat de o placă dedicată, funcționează o perioadă, apoi se resetează, deoarece pe serial placa dedicată nu poate menține nivelul de tensiune suficient de mare pentru funcționare. Placa de comandă SSC-32 al brațului robotic are un driver pentru serial ST202EC de la ST, acesta a fost recablă și s-a introdus plăcuța Pmod RS232, care are un driver de serial ADM3232E de la Analog Devices, care pare să fie mai bun, decât cel de la brațul robotic de pe placa SSC-32, deoarece acum brațul robotic poate fi comandat direct cu orice placă dedicată, nu numai cu un PC.

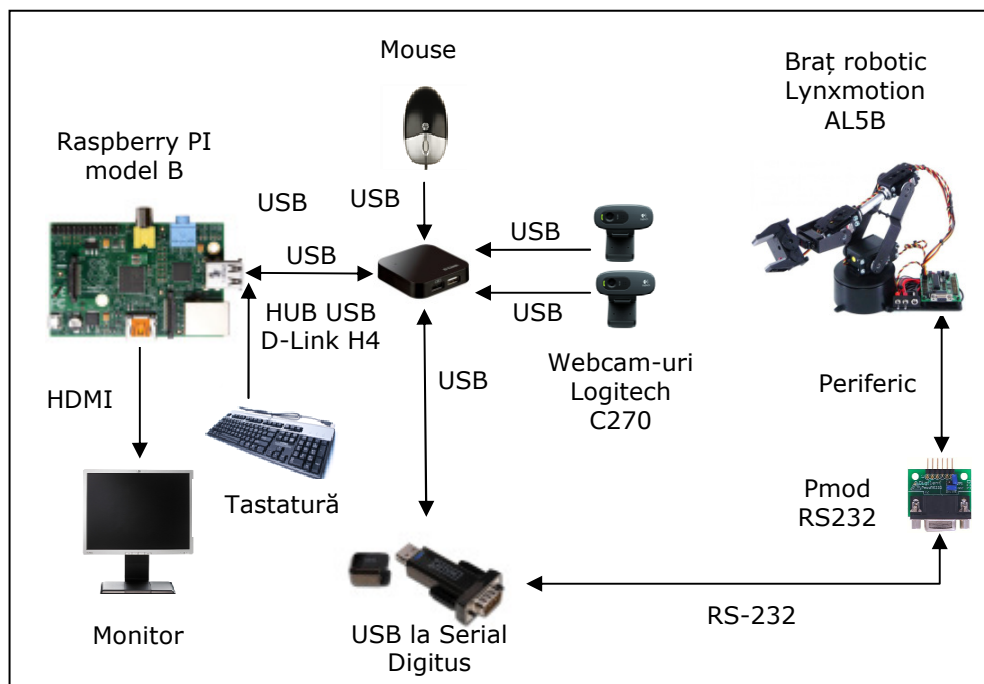


Fig. 4.42 Diagrama bloc al experimentului de control al brațului robotic Lynxmotion AL5B cu Raspberry PI și camere web

În fig. 4.43 se observă rezultatul programului pe Raspberry PI, scris în Python cu biblioteca OpenCV. La fel ca și în programele anterioare, există poza inițială cu desenele ajutătoare pentru calculul distanțelor. A doua imagine este cu brațul robotic după aplicarea filtrului HSV, iar la următoarele imagini este implementată detecția de contur cu setarea parametrilor minimi și maximi de la filtrul HSV pentru fiecare culoare. Nici în această interfață grafică nu există buton glisant dublu, deci trebuie să se utilizeze un buton glisant separat pentru valoarea minimă și maximă. Pentru controlul 3D al brațului robotic, tot programul trebuie dublat.

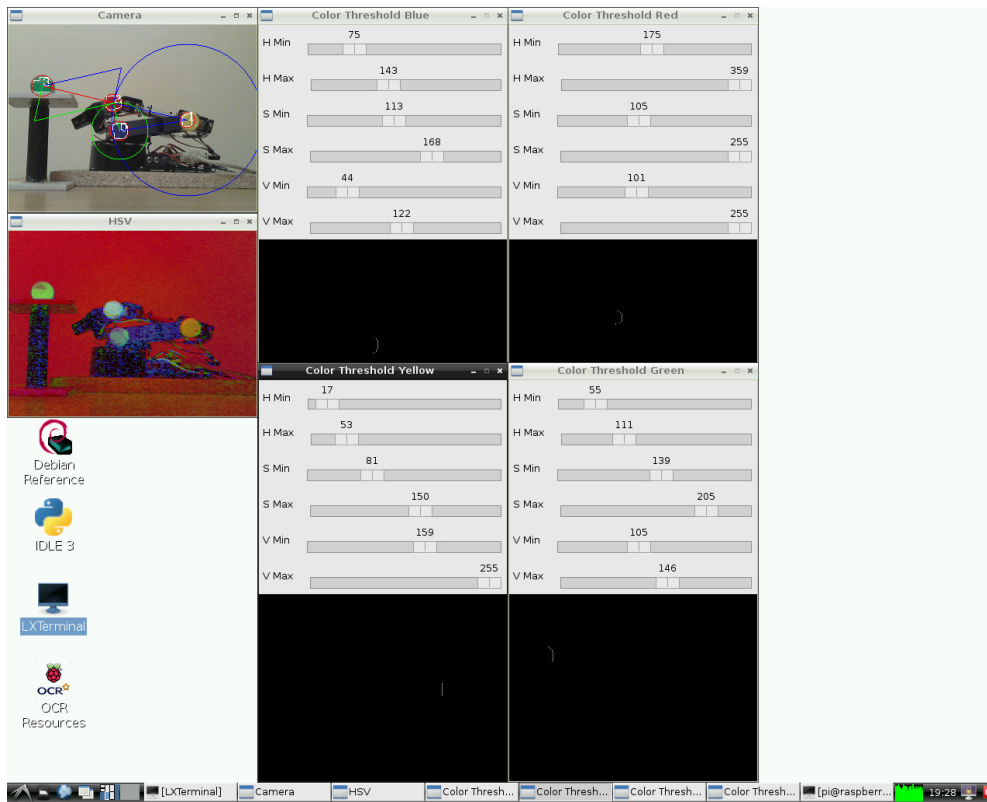


Fig. 4.43 Interfața grafică pentru controlul brațului robotic Lynxmotion AL5B cu Raspberry PI – interfață grafică realizată în Python cu OpenCV

În fig. 4.44 este prezentată imaginea mărită al brațului robotic Lynxmotion AL5B cu desenele ajutătoare pentru calculul distanțelor. Imaginea a fost achiziționată și prelucrată pe Raspberry PI în limbajul de script Python utilizând biblioteca OpenCV.

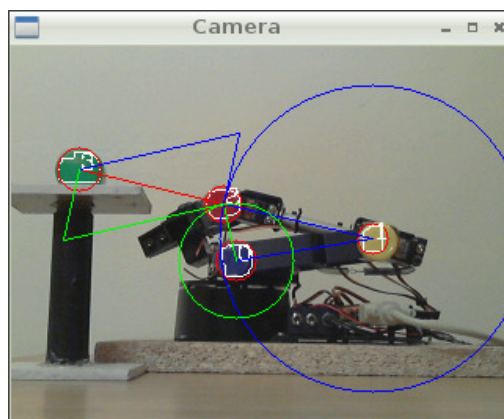


Fig. 4.44 Imagine mărită din interfața grafică pentru controlul brațului robotic Lynxmotion AL5B realizată pe Raspberry PI în Python cu OpenCV

În fig. 4.45 este prezentată diagrama bloc al experimentului de control al brațului robotic SCORBOT-ER III folosind Raspberry PI și camere video. În acest experiment se pot utiliza atât camerele pe interfața USB cât și camerele IP. Utilizând camerele IP sistemul se apropie mai mult de un sistem industrial și astfel se mai eliberează două porturi USB. Sunt ocupate trei porturi USB: de mouse, de tastatură și de convertorul USB la serial. Pe portul Ethernet de la Raspberry PI este conectat un switch, acesta este utilizat pentru a conecta camerele IP.

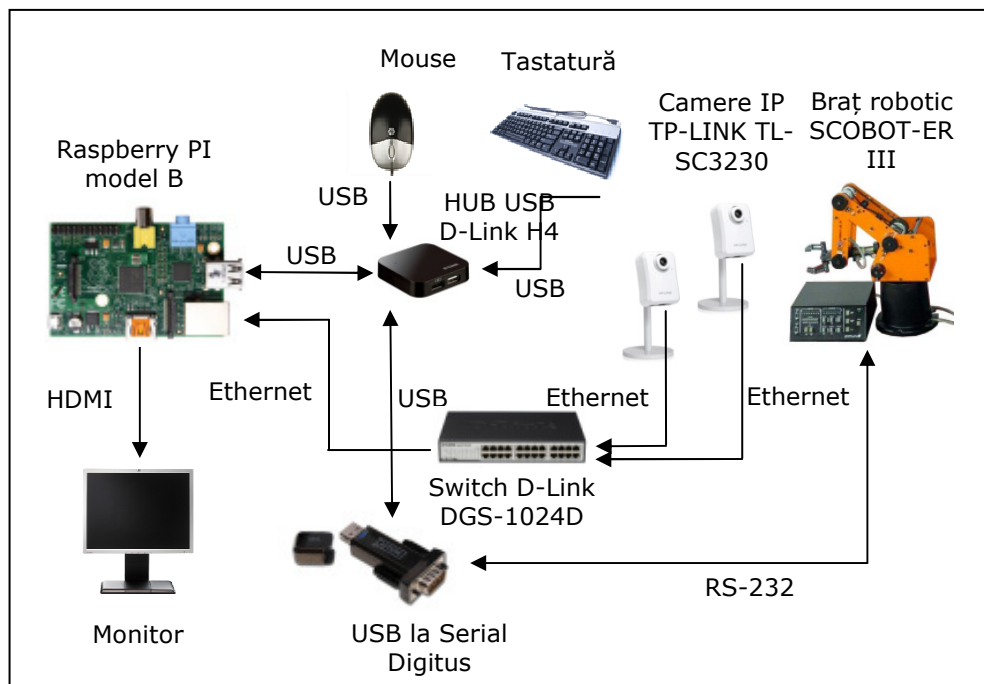


Fig. 4.45 Diagrama bloc al experimentului de control al brațului robotic SCORBOT-ER III cu Raspberry PI și camere IP

În fig. 4.46 este prezentat rezultatul programului pentru controlul brațului robotic SCORBOT-ER III cu Raspberry PI și camere video. Programul este similar cu cel din fig. 4.43, numai că de data aceasta este utilizat un robot industrial, brațul robotic SCORBOT-ER III și s-au utilizat și camerele IP.

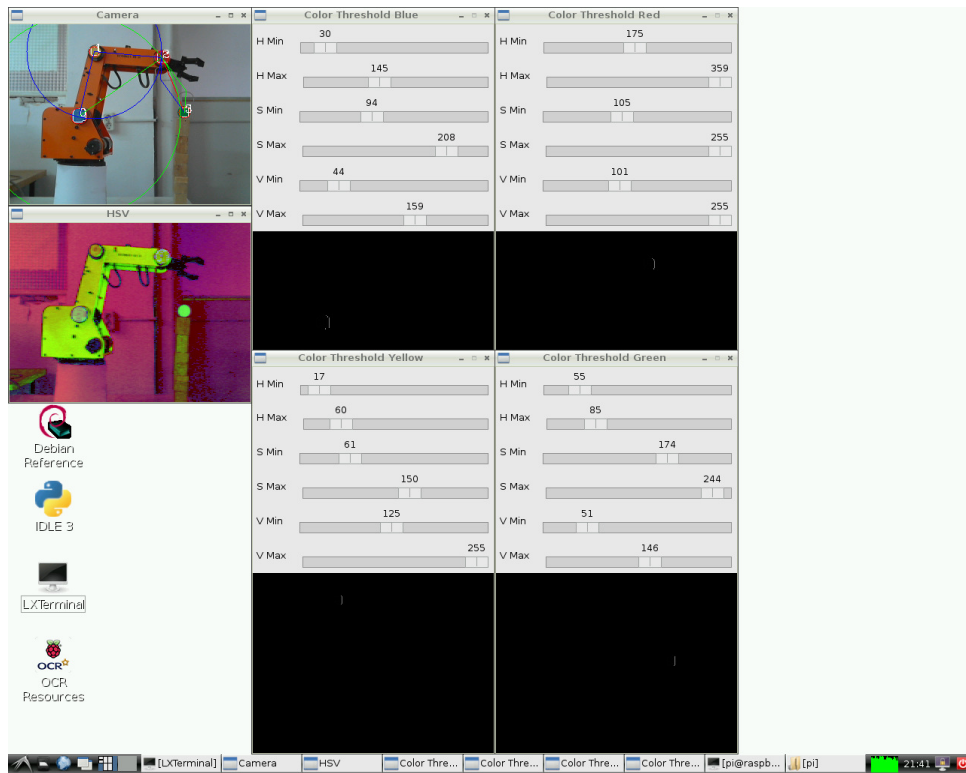


Fig. 4.46 Interfața grafică pentru controlul brațului robotic SCORBOT-ER III cu Raspberry PI – interfață grafică realizată în Python cu OpenCV

Fig. 4.47 este o porțiune mărită din figura anterioară. Această figură scoate în evidență faptul că algoritmul funcționează corect și pe alt braț robotic (SCORBOT-ER III), care acționează în altă direcție. Sistemul utilizează și alte camere video (camere IP), rulează pe un al sistem dedicat (Raspberry PI) cu un alt sistem de operare (Raspbian Linux) și este scris într-un alt limbaj de script (Python), cu aceeași bibliotecă OpenCV.

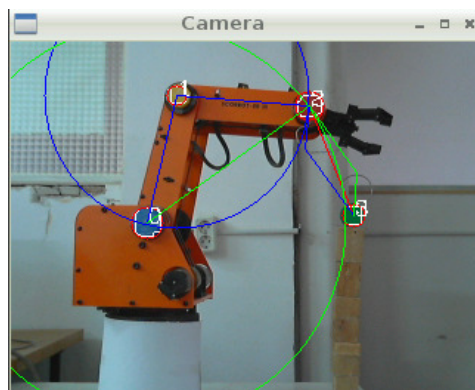


Fig. 4.47 Imagine mărită din interfața grafică pentru controlul brațului robotic SCORBOT-ER III realizată pe Raspberry PI în Python cu OpenCV

În fig. 4.48 se observă diagrama bloc al sistemului de control al brațului robotic Lynxmotion AL5B cu placa ZYBO și camerele web. Singurul lucru, care este diferit la această placă de dezvoltare utilizată (ZYBO) față de Raspberry PI este că are doar un singur port USB, astfel cu HUB-ul USB D-Link H4 avem doar 4 porturi USB, deci cu unul mai puțin cât s-ar dori. Acest sistem este puțin mai performant din punct de vedere al procesorului ca și Raspberry PI, deoarece are un microprocesor ARM Cortex – A9 Dual-core de 650 MHz, iar Raspberry PI are un microprocesor ARM1176JZF-S Single Core de 700 MHz.

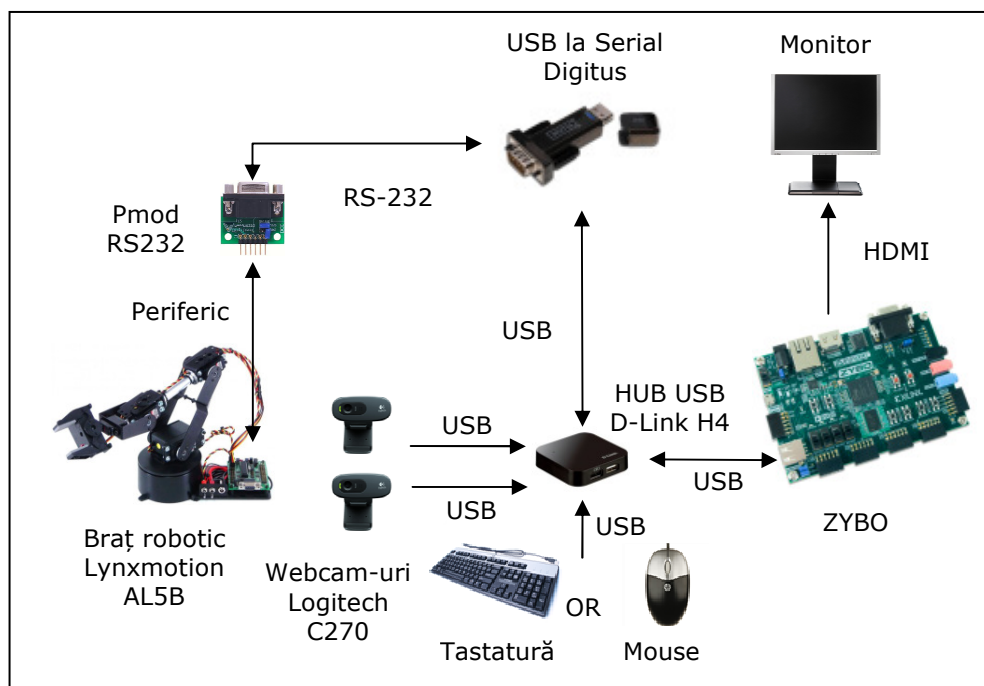


Fig. 4.48 Diagrama bloc al experimentului de control al brațului robotic Lynxmotion AL5B cu placa FPGA ZYBO

În fig. 4.49 este prezentată diagrama bloc pentru controlul brațului robotic SCORBOT-ER III cu placa ZYBO și camere IP. Diagrama bloc este similară cu cea din fig. 4.48, numai că la partea brațului robotic avem brațul robotic SCORBOT-ER III și pentru captarea imaginilor se folosesc camere IP. Sistemul dă posibilitatea să se folosească și camere web pe USB.

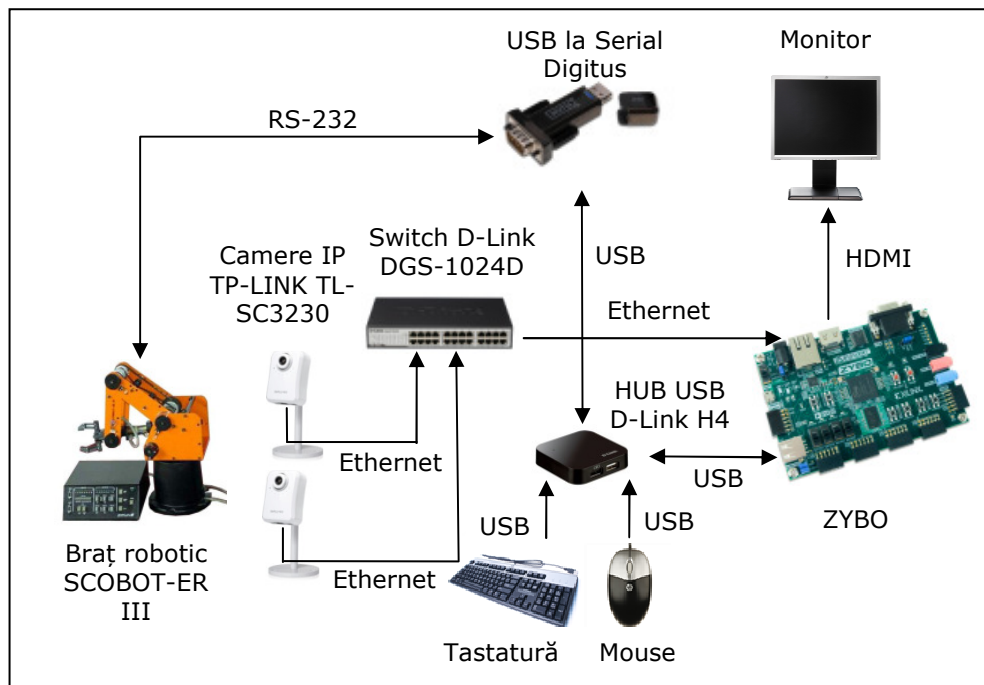


Fig. 4.49 Diagrama bloc al experimentului de control al brațului robotic SCORBOT-ER III cu placa FPGA ZYBO

În fig. 4.50 este prezentată diagrama bloc pentru experimentul de control al brațului robotic Lynxmotion AL5B cu placa ZedBoard și camere web pe interfața USB. Experimentul este foarte asemănător cu cel din fig. 4.48, numai că placa de dezvoltare este mult mai performantă. Placa ZedBoard conține un SoC Zynq-7000, care are un microprocesor ARM Cortex – A9 Dual-core tactat la o frecvență cuprinsă între 866 MHz și 1000 MHz, față microprocesorul de pe placa de dezvoltare ZYBO, care are un microprocesor ARM Cortex – A9 Dual-core cu semnal de tact la o frecvență de 650 MHz. SoC-ul Zynq-7000 are un FPGA de tip Atrix-7 atât pe placa ZedBoard cât și pe placa ZYBO. Placa ZedBoard are anumite diferențe față de placa ZYBO. Placa ZedBoard are portul USB cu mufă microUSB, ZYBO are portul USB cu mufă USB (mare) de tip A. La placa ZedBoard pentru a conecta o tastatură sau un mouse pe portul USB mai trebuie un cablu adaptor microUSB tată – USB A mamă. Pe placa ZYBO interfața serială UART este împreună cu interfața de programare cu mufă microUSB, la placa ZedBoard sunt mufe separate. Placa ZedBoard are un modul audio mai complex decât placa ZYBO, deoarece placa ZYBO are doar trei porturi audio analogice: intrare linie, microfon și ieșire căști; în afară de aceste porturi audio analogice, placa ZedBoard mai are un port audio analogic: ieșire linie. Partea de sunet nu este folosită în acest experiment, însă e un punct în plus, partea de sunet analogic mai complexă, pentru placa ZedBoard. O diferență mare, care se face simțită la cele două plăci este că ieșirea video este setată în sistemul de operare Ubuntu Linux 12.04 LTS la placa ZedBoard pe portul VGA, iar pe placa ZYBO

pe portul HDMI. Utilizarea portului VGA este destul de vizibilă la calitatea imaginii, deoarece microprocesorul nu are alăturat și un procesor grafic. Sistemul nu are GPU, astfel când este utilizat portul analogic VGA, se văd pixelii mult mai pronunțați, față de când se utilizează portul digital HDMI. Având în vedere că nu este GPU pe aceste sisteme, sistemul de detecție de markeri funcționează fără probleme. Sistemul Raspberry PI are GPU, sistemele cu SoC Zynq-7000 au fost rezultatele experimentelor mai bune deoarece prezența procesorului Dual-core este mai importantă, în aceste experimente, decât prezența unui GPU. Practic pozele sunt achiziționate la mai mult de 256 de culori, numai vizualizarea nu este așa de clară din cauza portului analogic VGA și a lipsei de GPU.

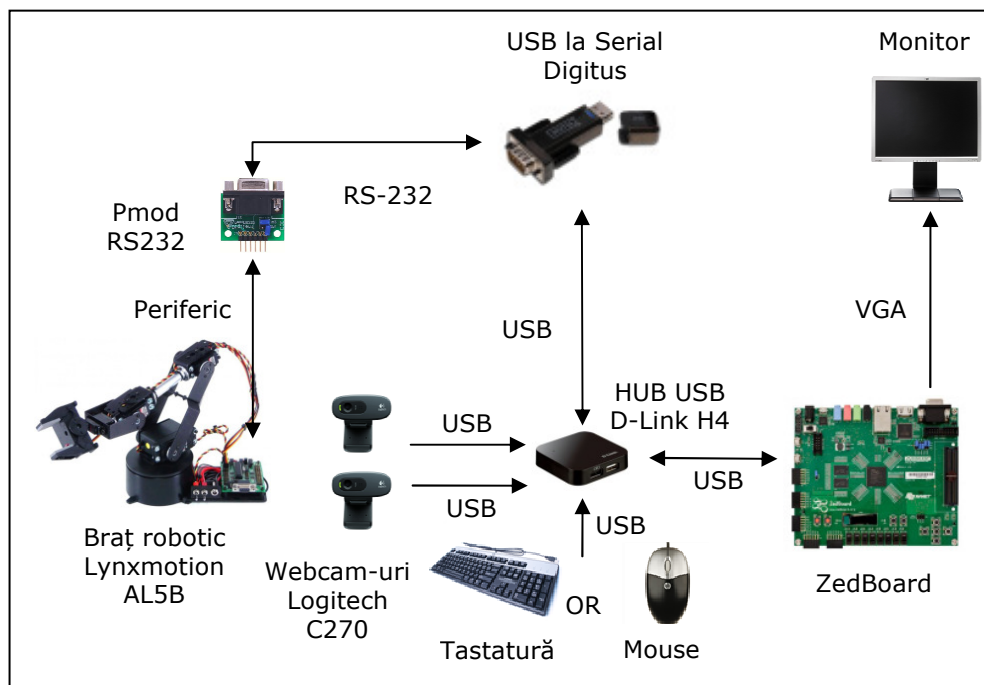


Fig. 4.50 Diagrama bloc al experimentului de control al brațului robotic Lynxmotion AL5B cu placa FPGA ZedBoard

În fig. 4.51 este prezentată diagrama bloc pentru controlul brațului robotic SCORBOT-ER III cu placa ZedBoard și camere IP. Diagrama bloc este similară cu cea din fig. 4.49, numai aici este prezentată versiunea industrializată cu robotul SCORBOT-ER III și camere IP.

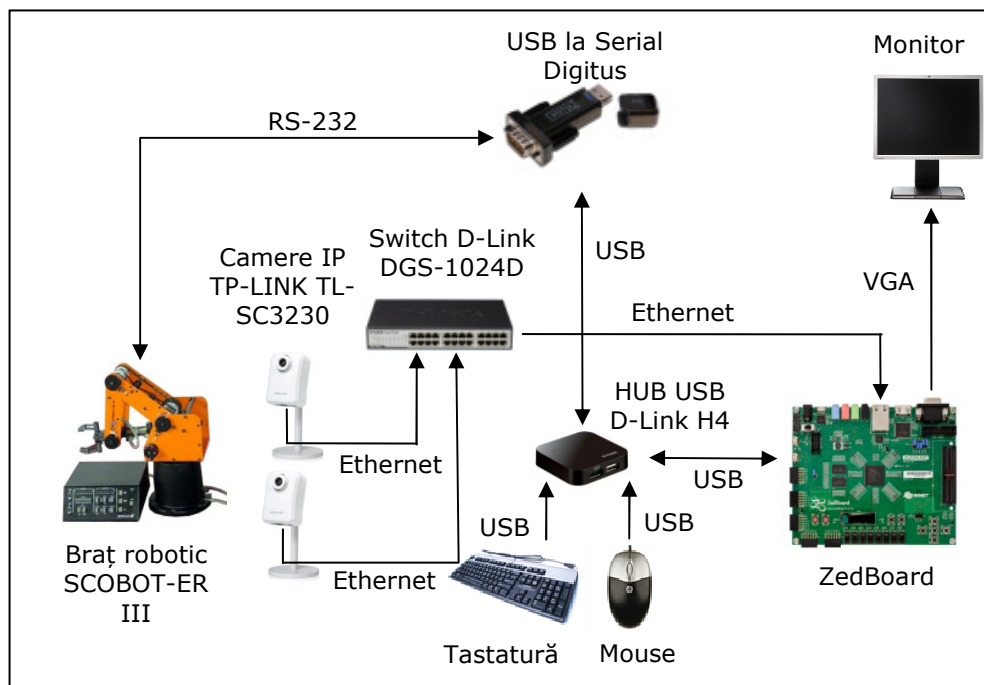


Fig. 4.51 Diagrama bloc al experimentului de control al brațului robotic SCORBOT-ER III cu placa FPGA ZedBoard

În fig. 4.52 este prezentat rezultatul programelor de pe plăcile ZYBO și ZedBoard, care controlează brațul robotic Lynxmotion AL5B cu ajutorul camerelor web pe interfața USB. Programul se poate porta de pe PC foarte ușor (fig. 4.37) pe plăcile dedicate cu SoC-ul Zynq-7000 format dintr-un microprocesor și un FPGA. Pentru portare trebuie realizate doar mici modificări, cea mai mare schimbare este de a comprima imaginile să încapă pe un singur monitor, deoarece la plăcile dedicate nu sunt instalate două monitoare ca și la unele PC-uri. Astfel este afișată imaginea stângă și dreaptă al brațului robotic Lynxmotion AL5B cu trasarea liniilor ajutătoare pentru calculul distanțelor, apoi este realizată egalizarea de histogramă pentru ambele imagini și este implementată și filtrarea HSV. Pentru fiecare culoare este utilizată o detecție de contur cu reglarea parametrilor minim și maxim ale filtrului HSV. Multe imagini sunt suprapuse din cauza că nu este destul spațiu pe un singur monitor. **Rezultatele pe plăcile de dezvoltare cu SoC Zynq-7000 au fost comparabile cu rezultatele obținute pe un PC performant.** Captarea de imagine este copiată, pe PC, de pe plăcile de dezvoltare care rulează Ubuntu Linux 12.04 LTS.

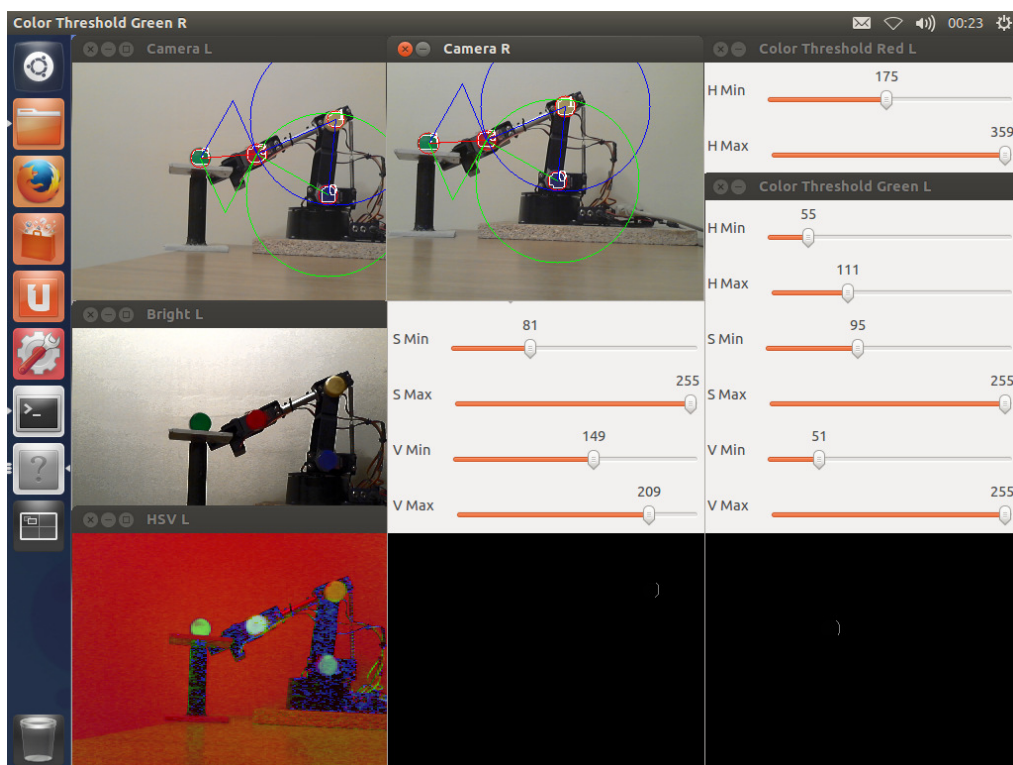


Fig. 4.52 Interfața grafică al controlului brațului robotic Lynxmotion AL5B realizat pe plăcile ZYBO sau ZedBoard cu Linux – program creat în C cu OpenCV

În fig. 4.53 este prezentată imaginea mărită cu brațul robotic Lynxmotion AL5B filmată de cele două camere web pe interfața USB pentru a evidenția funcționarea corectă al algoritmului de trasare a liniilor ajutătoare pentru calculul de distanțe.

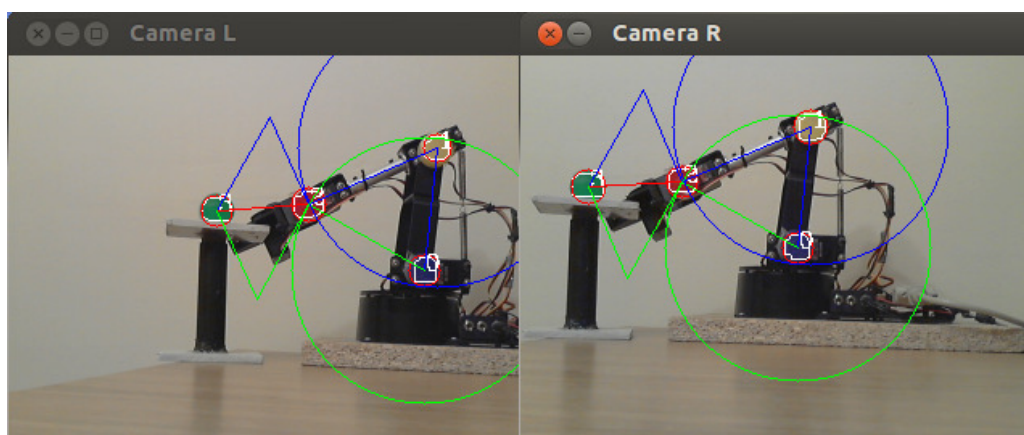


Fig. 4.53 Imagine mărită din programul de control al brațului robotic Lynxmotion AL5B cu liniile ajutătoare pentru calculul poziției brațului robotic în 3D

În fig. 4.54 se observă imaginea copiată, pe un PC, de pe plăcile de dezvoltare ZYBO și ZedBoard în momentul când se execută programul pentru controlul brațului robotic SCORBOT-ER III cu ajutorul camerelor video. Programul este similar cu cel din fig. 4.52, numai că este controlat brațul robotic SCORBOT-ER III. Mai este diferită și posibilitatea sistemului industrial de a avea camere IP pentru captarea imaginilor. În acest caz achiziția imaginilor se face pe portul Ethernet, nu pe portul USB ca și la webcam-uri.

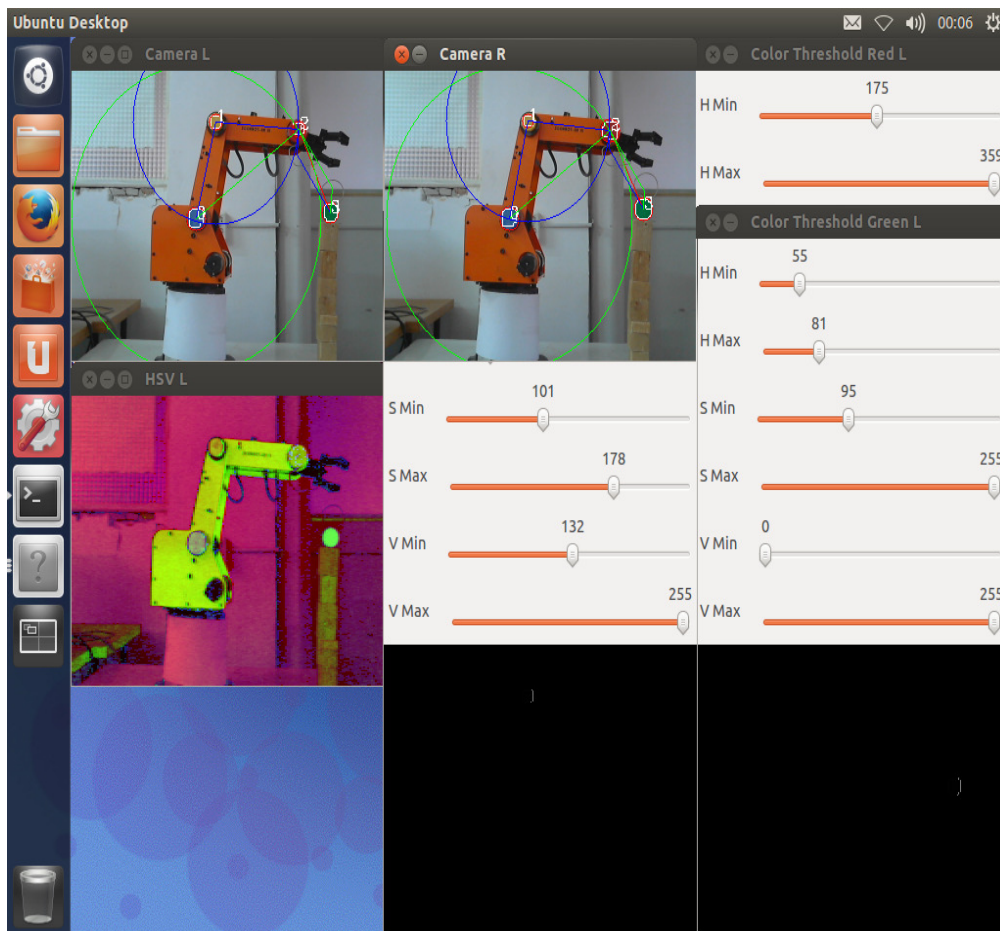


Fig. 4.54 Interfața grafică al controlului brațului robotic SCORBOT-ER III realizat pe plăcile ZYBO sau ZedBoard cu Linux – program creat în C cu OpenCV

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

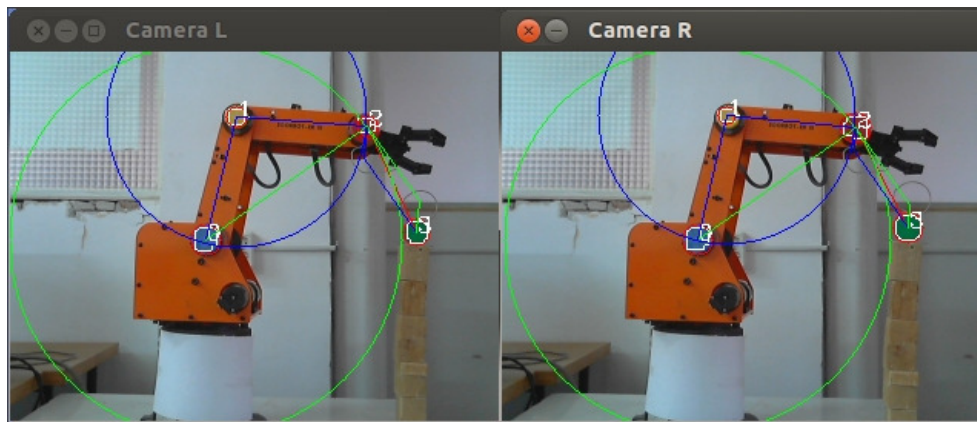


Fig. 4.55 Imagine mărită din programul de control al brațului robotic SCORBOT-ER III cu liniile ajutătoare pentru calculul poziției brațului robotic în 3D

4.4. Analiza comparativă a performanțelor implementărilor

În tabelul 4.2 este redată o comparație a resurselor folosite la diferitele implementări. A fost adunat fiecare parametru al diferitelor implementări, astfel să se poată arăta, care implementarea este optimă din punct de vedere al resurselor ocupate.

Tabel 4.2 Comparația implementărilor sistemului de control al brațului robotic folosind camere video

Resurse utilizate	VHDL pe FPGA	LabWindows/CVI și Robo-Realm sub Windows	LabVIEW și NI Vision Development Module sub Windows	LabWindows/CVI și NI Vision Development Module sub Windows	C și Open-CV sub Linux	Python și OpenCV sub Linux
Număr linii cod	13722	2091	N/A (limbaj grafic)	1214	1030	539
Mărime cod pe disk	608 KB (13 fișiere *.vhd)	64 KB (2 fișiere *.c și 2 fișiere *.h)	896 KB (6 fișiere *.vi)	48 KB (1 fișier *.c și 1 fișier *.h)	40 KB (1 fișier *.cpp)	28 KB (1 fișier *.py)
Mărime fișier binar pe disk	860 KB (1 fișier *.bit)	964 KB (1 fișier *.exe)	1,28 MB (1 fișier *.exe)	978 KB (1 fișier *.exe)	44 KB (1 fișier fără extensie)	N/A (script)

Memorie ocupată în timpul execuției	N/A (1103 din 54576 Flip Flop)	47064 KB	60268 KB	42180 KB	79872 KB (din 2 GB RAM în PC) 50332 KB (din 512 MB RAM pe placa ZYBO și Zed-Board)	67584 KB (din 2 GB RAM în PC) 54002 KB (din 512 MB RAM pe placa Raspberry PI)
Utilizare UCP în timpul execuției	N/A	52% (din Intel Core 2 Duo @ 2,4 GHz)	32% (din Intel Core 2 Duo @ 2,4 GHz)	24% (din Intel Core 2 Duo @ 2,4 GHz)	61% (din Intel Core 2 Duo @ 2,4 GHz) 76,6 % (din ARM Cortex – A9 Dual-core @ 650 MHz)	29,8% (din Intel Core 2 Duo @ 2,4 GHz) 46% (din ARM 11 @ 700 MHz)

În fig. 4.56 sunt reprezentate numărul liniilor de cod, la care minimul este scriptul în limbajul Python și codul în limbajul C pe sistemul de operare Linux. Aceste implementări care folosesc biblioteca OpenCV. Pe partea opusă stă codul VHDL, unde totul trebuie implementat de la zero.

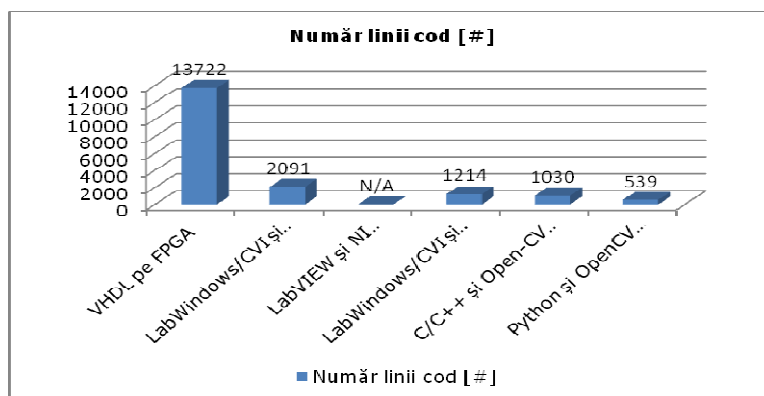


Fig. 4.56 Comparația numărului de linii de cod la diferite implementări

În fig. 4.57 este reprezentată mărimea codului pe disk la diferite implementări, unde campionii sunt scriptul în limbajul Python și codul în limbajul C pe sistemul de operare Linux. Pe cealaltă parte este implementarea în LabVIEW, care generează un fișier (neașteptat de) mare.

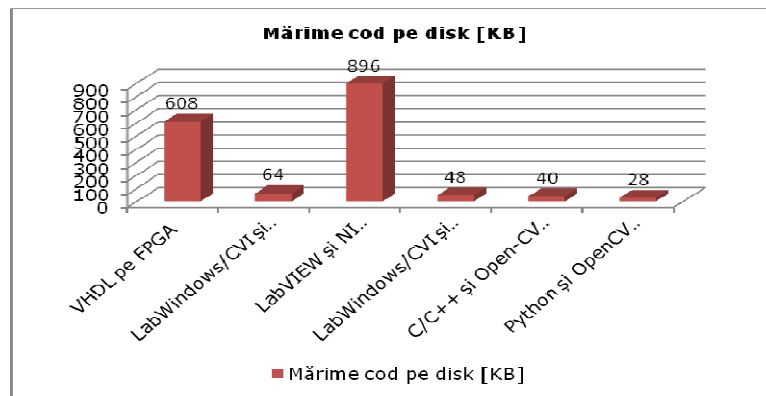


Fig. 4.57 Comparația mărimii codului pe disk la diferite implementări

În fig. 4.58 este prezentată mărimea fișierului binar pe disk, la care câștigătorul este codul în limbajul C pe sistemul de operare Linux, deoarece codul este foarte scurt, iar funcțiile din biblioteca OpenCV sunt alocate dinamic în timpul execuției, deci nu sunt compilate în executabil. La limbajul de script Python nu putem vorbi de executabil, deoarece nu se generează un executabil, fiind un limbaj de script, se rulează scriptul pe serverul Python. Pe partea cealaltă stă LabVIEW, unde totul se compilează în executabil. Astfel rezultând un fișier relativ mare.

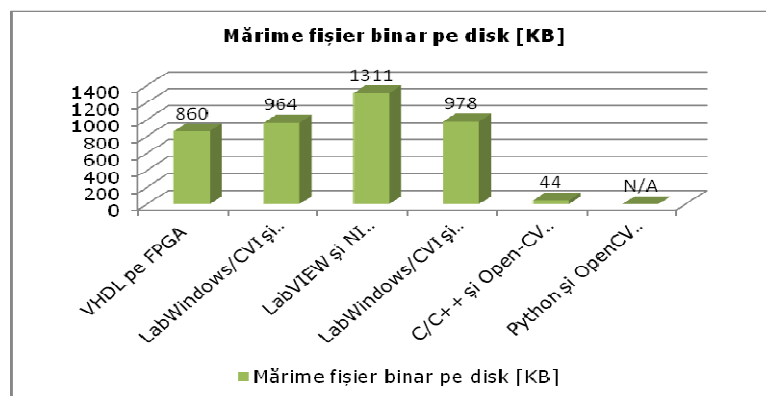


Fig. 4.58 Comparația mărimii fișierului binar pe disk la diferite implementări

În fig. 4.59 este prezentată comparația între memoria utilizată la implementările prezentate. Implementarea în VHDL, care chiar dacă nu are memorie, din sumarul de implementare, resursele ocupate au fost în proporție de 2%. Pe cealaltă parte o poziție bună au implementările care folosesc scriptul în limbajul Python și codul în limbajul C pe sistemul de operare Linux. Acestea din urmă ocupă o memorie mai mare datorată faptului că toată biblioteca OpenCV este încărcată dinamic în memorie în timpul execuției, iar această implementare este și

una eficace de utilizat, deoarece biblioteca OpenCV are cea mai mare diversitate de funcții de procesare de imagine.

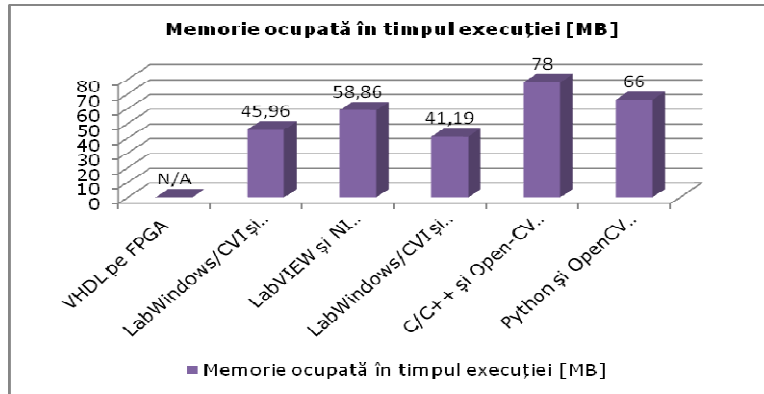


Fig. 4.59 Comparația între cantitatea de memorie ocupată în timpul execuției la diferite implementări

În fig. 4.60 este prezentată comparația între procentul de utilizare UCP (unitatea centrală de procesare) în timpul execuției la implementările prezentate în această muncă de cercetare. Din nou, câștigătorul cu cele mai puține resurse ocupate este implementarea în VHDL, iar pe latura cealaltă este implementarea în limbajul C, care rulează pe sistemul de operare Linux. Acesta trebuie să încarce dinamic toată biblioteca OpenCV, astfel acesta necesită o putere mai mare de procesare, de pe grafic se poate vedea că acest lucru se poate reduce dacă se utilizează un script în limbajul Python, care rulează pe server, iar astfel încărcarea de procesor este mult diminuată.

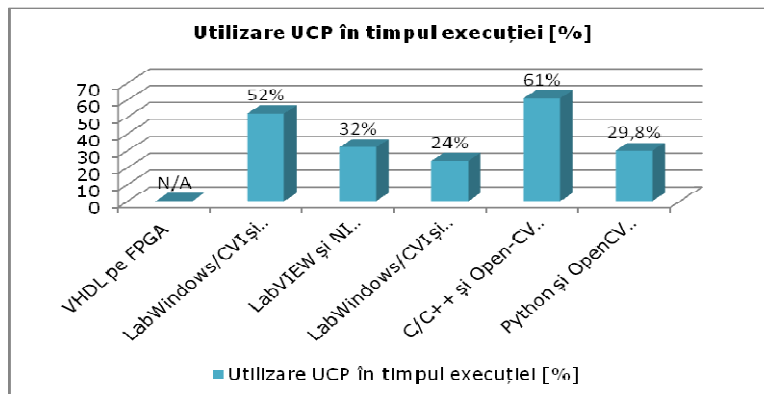


Fig. 4.60 Comparația între procentul de utilizare UCP în timpul execuției la diferite implementări

În final (fig. 4.61) se poate afirma că fiecare implementare are propriile particularități. La VHDL toată implementarea 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. Totuși câștigătorul ar fi implementarea în scriptul Python pe sistemul de operare Linux, care este relativ simplă 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.

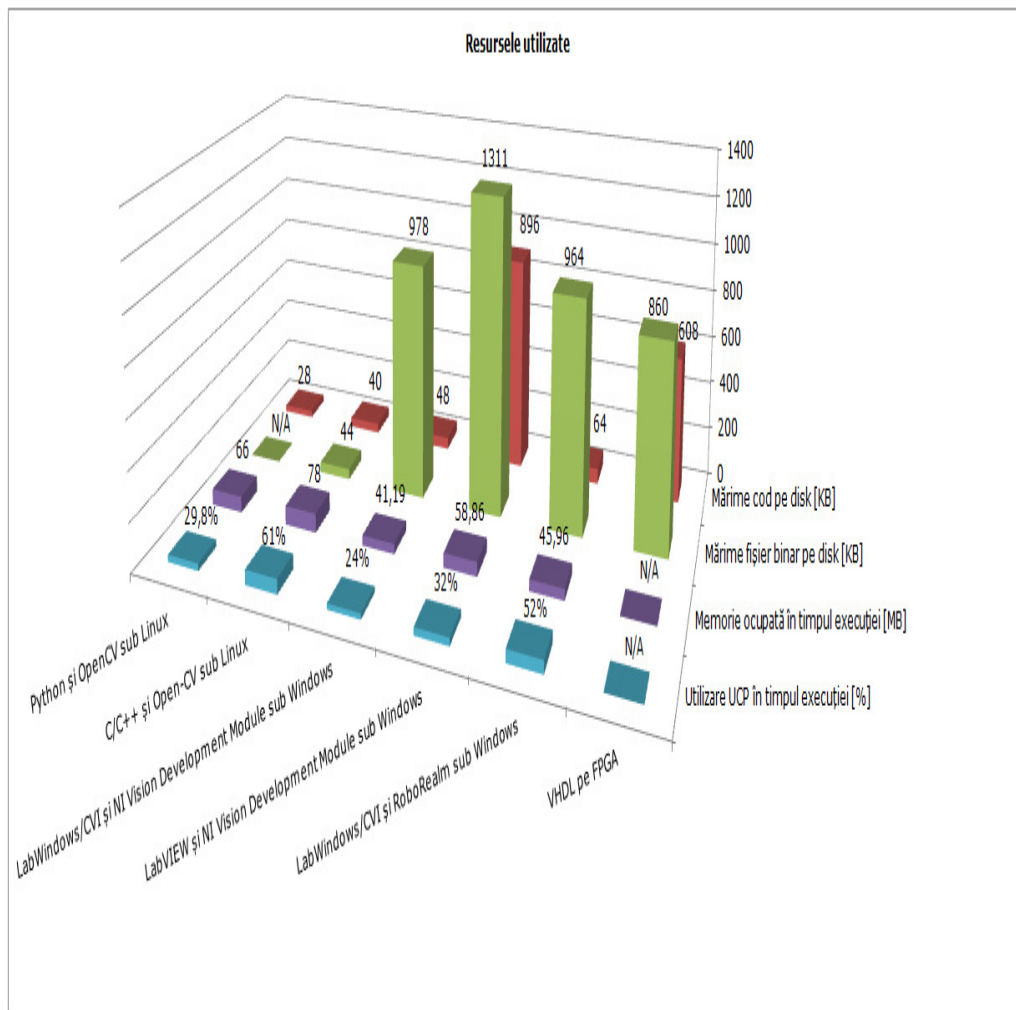


Fig. 4.61 Comparația între resursele utilizate la diferite implementări

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:

- a. 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.
- b. 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.
- c. 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.
- d. Cel mai bun compromis este Python pe sistemul de operare Linux, care este relativ simplu și nici din resurse nu ocupă prea mult.
- e. 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).

Contribuții

1. **Metoda de calcul și algoritmul de control al mișcării sunt originale.**
 - 1.1. 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.
 - 1.2. Toate formulele au fost simplificate și puse într-o formă implementabilă în cod sursă calculator.
 - 1.3. 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 vadă o porțiune din aceste marcaje colorate puse la articulațiile robotului.
 - 1.4. Deoarece un braț robotic industrial nu are în general mai mult de 7 articulații, metodele descrise (2D și 3D) se pot aplica cu succes.
 - 1.5. A fost dezvoltat **algoritmul de trasare a liniilor și cercurilor ajutătoare pentru calculul distanțelor** pentru brațul robotic, având

- ca date de intrare doar coordonate obținute din poziția în spațiu a bulinelor colorate detectate.
- 1.6. Am demonstrat geneza tuturor formulelor și le-am validat prin mișcarea controlată a robotului.
 2. **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.
 - 2.1. Am implementat **o funcție de autocalibrare și una de corecție a erorilor.** Mișcarea se execută în buclă, fără intervenția operatorului uman (exceptând pornirea sa).
 3. Pentru rularea pe FPGA a algoritmului propus a fost necesară **implementarea unor noi arhitecturi pe aceste circuite reconfigurabile:**
 - 3.1. 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.
 - 3.2. 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.
 4. S-a dezvoltat un **model tridimensional MATLAB pentru brațul robotic.**
 5. 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. Rezultatele au fost reprezentate grafic.

5. ANALIZA ȘI COMPARAȚIA REZULTATELOR

5.1. Analiza măsurării distanțelor folosind camere video

5.1.1. Analiza măsurării de distanțe la diferitele echipamente utilizate

Pentru a verifica precizia sistemului de măsurare a distanțelor folosind camere video s-au utilizat metode statistice șase sigma (6σ). Aceste metode de testare a robusteții sistemului se utilizează și în producție pentru ca produsele să corespundă standardelor ridicate din ziua de azi. Metodele șase sigma au fost implementate realizând grafice cu programul Minitab.

Precizia sistemului este foarte importantă, deoarece după aceste măsurări, folosind camere video, se calculează distanțele de deplasare al brațului robotic. Fiind o metodă optică, se operează uneori corecții - dacă brațul robotic nu a ajuns în punctul dorit, acesta mai face auto ajustări sau repetă mișcarea pentru a ajunge exact în punctul dorit, acest fapt fiind datorat sistemului care este executat într-o buclă.

Pentru axele 0Y și 0Z nu sunt probleme de precizie (deoarece, erori mari pot aprărea doar la măsurarea distanței în adâncime, axa 0X, unde măsurarea se face la nivel de cm, la axele paralele cu planul camerei, axele 0Y și 0Z, precizia e la nivel de pixel), însă pentru axa 0X există probleme de precizie, deoarece aceasta reprezintă măsurarea distanței în adâncime, adică măsurarea distanței cu cele două camere video, așezate pe post de camere stereo.

În tabelul 5.1 sunt prezentate o serie de măsurări la distanța de 1 m repetate de două ori cu șase tipuri de camere (TP-LINK, Logitech, A4Tech, Trust 1, Trust 2 și Trust 3) și trei platforme diferite (PC, FPGA și μ P). Camera notată TP-LINK este TP-LINK TL-SC3230, camera notată Logitech este Logitech C270, camera notată A4Tech este A4Tech PK-635M, camera notată Trust 1 este Trust WB-8600R, camera notată Trust 2 este Trust WB-3400T, iar camera notată Trust 3 este Trust WB-3350P. Tabelul fost creat pentru studiul Gage R&R, utilizat pentru reproductibilitatea măsurătorilor.

Tabel 5.1 Studiul Gage R&R pentru măsurarea distanței de 1 m cu șase tipuri de camere video și pe trei tipuri de platforme

Distanță [mm]	Cameră Video	Platformă
990	TP-LINK	PC
989	Trust 3	PC
995	A4Tech	PC
1005	Trust 2	PC
992	Trust 1	PC
1001	Logitech	PC

988	Trust 3	FPGA
1006	Trust 2	FPGA
991	TP-LINK	FPGA
991	Trust 1	FPGA
1002	Logitech	FPGA
994	A4Tech	FPGA
1003	Logitech	μ P
994	Trust 1	μ P
990	Trust 3	μ P
993	TP-LINK	μ P
996	A4Tech	μ P
1004	Trust 2	μ P
993	A4Tech	PC
992	Trust 3	PC
995	Trust 1	PC
1005	Trust 2	PC
999	Logitech	PC
994	TP-LINK	PC
997	A4Tech	FPGA
1003	Trust 2	FPGA
996	Trust 1	FPGA
1001	Logitech	FPGA
992	TP-LINK	FPGA
991	Trust 3	FPGA
993	A4Tech	μ P
991	TP-LINK	μ P
997	Trust 1	μ P
990	Trust 3	μ P
1004	Trust 2	μ P
1003	Logitech	μ P

În fig. 5.1 este prezentat studiul Gage R&R, care are rezultate foarte bune. La contribuție s-a obținut 27,9%, **ceea ce este sub limita de acceptabilitate de 30%**, iar la variația de studiu s-a obținut 7,79%, **ceea ce este sub limita de acceptabilitate de 10 %**. Graficele R și Xbar (Xbarat sau \bar{x}) arată tendința dorită, graficul R este între limite, iar graficele Xbarat sunt aproape similare la toate cele trei platformele.

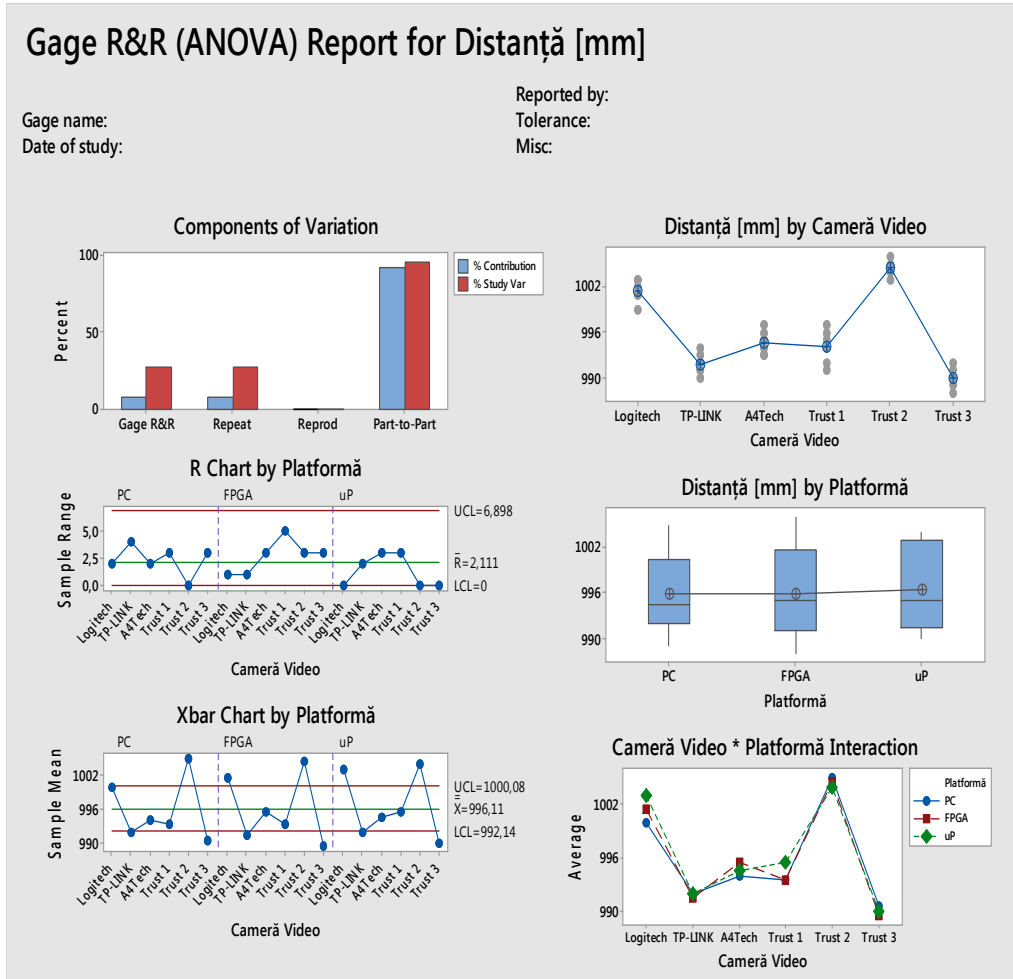


Fig. 5.1 Rezultatele studiului de Gage R&R – studiu de reproductibilitate

În fig. 5.2 se observă graficul de probabilitate normală, la intervalul de încredere de 95%, pentru $\alpha = 0,05$, la distanța de 1 m. Se poate vedea că valorile nu sunt dispersate, nu sunt valori ieșite prea mult în afara liniilor care marchează limite. Media este la 996,1 mm la 36 de măsurări ale distanței de 1 m, la aceste valori deviația standard este de 5,46 mm. Valoarea AD (Anderson-Darling) este 1,199, iar valoarea p este $< 0,005$, deci mai mică ca $\alpha = 0,05$, deci ipoteza nulă este rejectată, aceasta înseamnă că există diferență între valori, ceea ce era și de așteptat, deoarece în această situație sunt comparate toate valorile, nu doar valorile în funcție de platforme sau valorile în funcție de camerele utilizate.

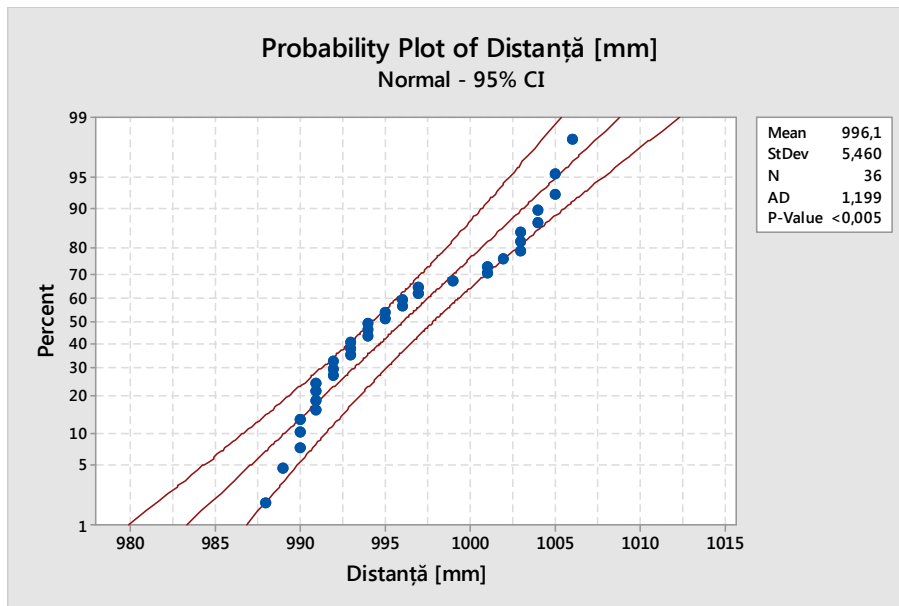


Fig. 5.2 Graficul de probabilitate normală pentru măsurarea distanței de 1 m folosind camere video

În fig. 5.3 se observă graficul cu linii, pentru măsurare distanței de 1 m, pe cele trei platforme și cu cele șase camere video. Din grafic reiese că cele mai bune măsurători (cel mai aproape de 1 m) se obțin folosind camerele Logitech și Trust 2. La platforme nu se observă clar, care este mai bună, toate par similare, liniile trasate fiind relativ orizontale.

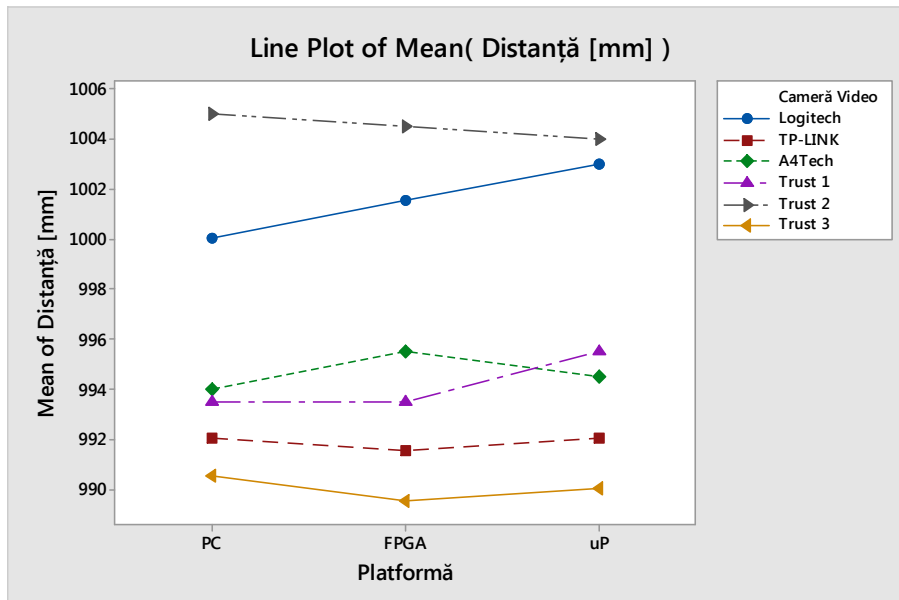


Fig. 5.3 Graficul de linii pentru cele șase tipuri de camere video și cele trei tipuri de platforme

În fig. 5.4 se observă graficul de probabilitate normală la intervalul de încredere de 95%, pentru $\alpha = 0,05$, la distanța de 1 m, la diferite platforme. Se observă o medie destul de apropiată de 1 m, pentru cele 12 măsurări, la fiecare platformă în parte. Deviația standard este asemănătoare la fiecare platformă în parte. Se pot observa valorile AD (Anderson-Darling) și valorile p, care sunt mai mari de $\alpha = 0,05$, deci ipoteza nulă nu este rejectată, astfel se remarcă că nu sunt diferențe majore între implementările pe diferite platforme. Precum se observă, graficele se suprapun aproape perfect, ceea ce evidențiază încă o dată că nu există diferențe prea mari între platforme, ceea ce era și de așteptat, deoarece măsurarea de distanțe, folosind camere video, nu ar trebui să fie influențată, de platformă, mai degrabă de camera video.

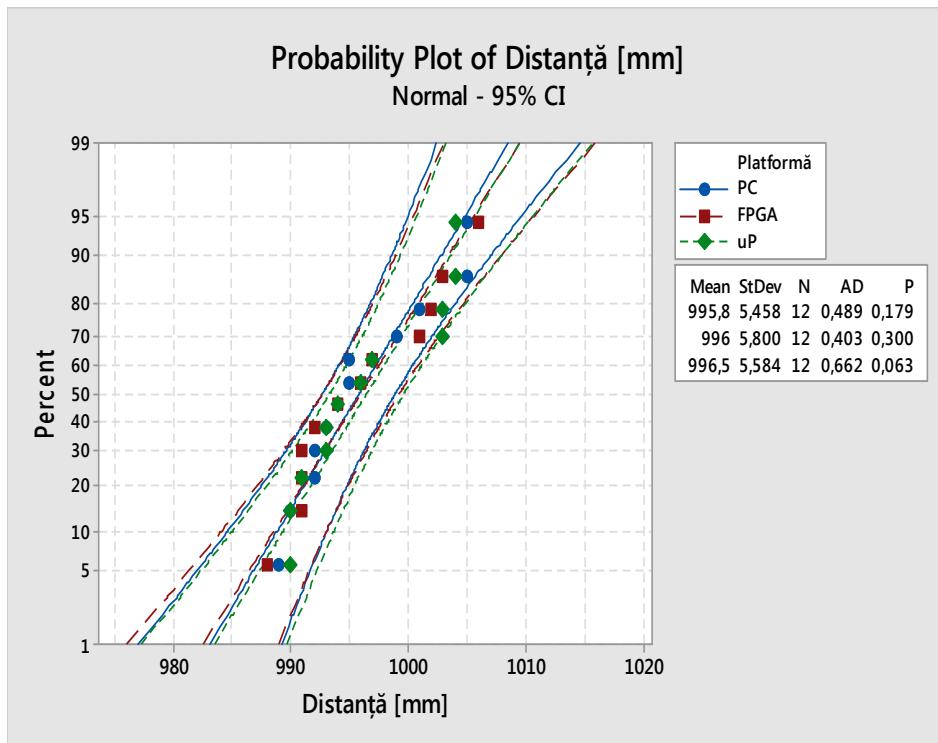


Fig. 5.4 Graficul de probabilitate normală pentru cele trei tipuri de platforme

În fig. 5.5 se observă graficul de probabilitate normală, la intervalul de încredere de 95%, pentru $\alpha = 0,05$, la distanța de 1 m la diferite camere video. Media este destul de bună la fiecare, poate cea mai bună valoare se regăsește, la camerele Logitech și Trust 2, din cele șase măsurători. Deviația standard este destul de asemănătoare la fiecare cameră. Se observă valorile AD (Anderson-Darling) și valorile p, care sunt mai mari de $\alpha = 0,05$, de ipoteza nulă nu este rejectată, astfel se poate spune că nu există diferențe majore nici între măsurătorile cu diferite camere video. Precum se vede, graficele nu se suprapun, deci există o diferență mai pronunțată între camere, dar conform testului de ipoteză această diferență nu este așa de mare încât să influențeze măsurările.

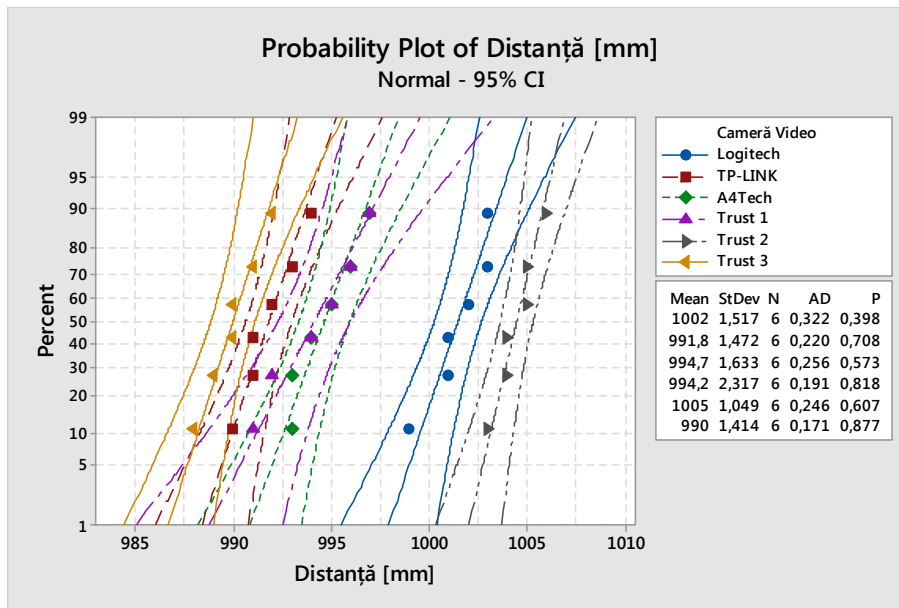


Fig. 5.5 Graficul de probabilitate normală pentru cele șase tipuri de camere

În fig. 5.6 se observă graficul cu pătrățele, la măsurarea distanței de 1 m, cu diferite camere video. Acest grafic evidențiază așa numita coadă și mustață (tail and whisker), care sunt liniile care ies din pătrățele. Mai este evidențiată media, care este linia din mijlocul pătrățelelor, deci se poate vedea valoarea medie și valorile limită. Din acest grafic reiese, de asemenea, că cele mai bune camere video, sunt Logitech și Trust 2, care au măsurat valori cel mai apropiate de 1 m.

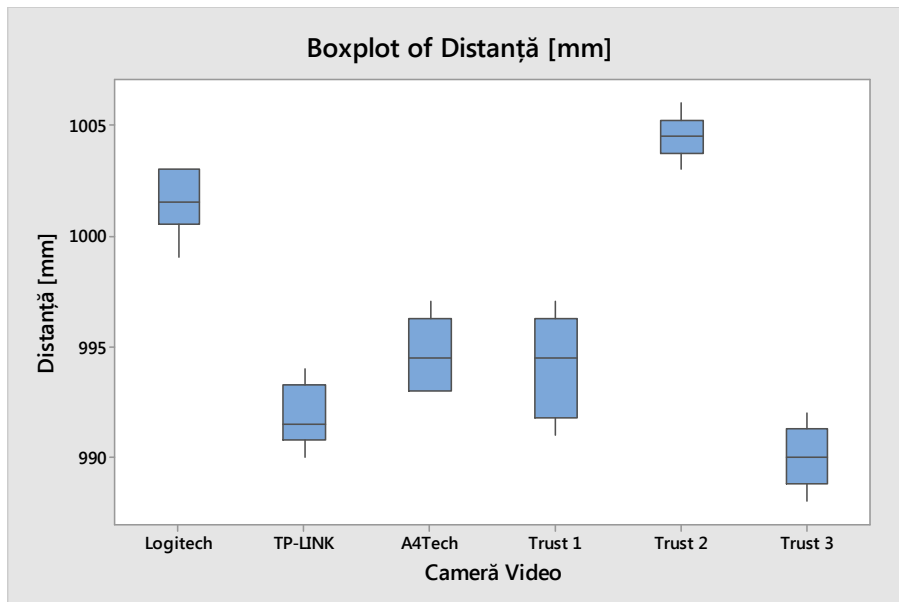


Fig. 5.6 Graficul de pătrate pentru cele șase tipuri de camere video

În fig. 5.7 se observă graficul de intervale, pentru intervalul de încredere de 95%, la măsurarea distanței de 1 m, cu diferite camere video. Acest grafic este o metodă ANOVA (analiză de variație) cu o singură direcție. Se poate vedea, că și din acest grafic reiese, că cele mai bune camere, sunt Logitech și Trust 2, având măsurători foarte apropiate de 1 m, însă se poate vedea și că variația la celelalte camere nu este foarte mare.

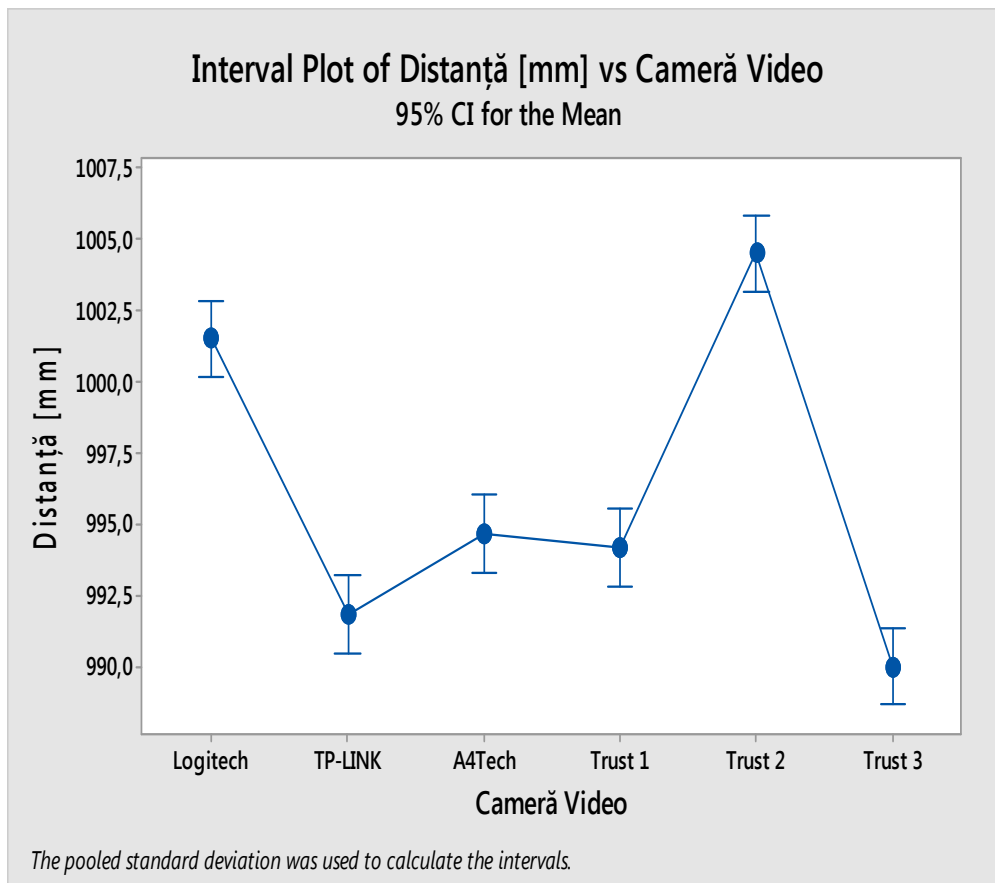


Fig. 5.7 Graficul de intervale pentru cele șase tipuri de camere video – ANOVA (analiză de variație)

După analiza ANOVA rezultă și o analiză a valorilor reziduale, care este prezentată din fig. 5.8. Se poate vedea că valorile reziduale sunt aproape toate pe line la graficul de probabilitate normală, valoarea AD (Anderson-Darling) este 0,191, valoarea p este 0,89, deci mai mare ca $\alpha = 0,05$, deci ipoteza nulă nu este rejectată, deci se poate spune că nu există diferențe majore între valorile reziduale. Acest lucru se poate observa și pe histogramă. În rest se poate vedea dispersia valorilor, pe graficele versus, o dispersie, care nu este foarte mare.

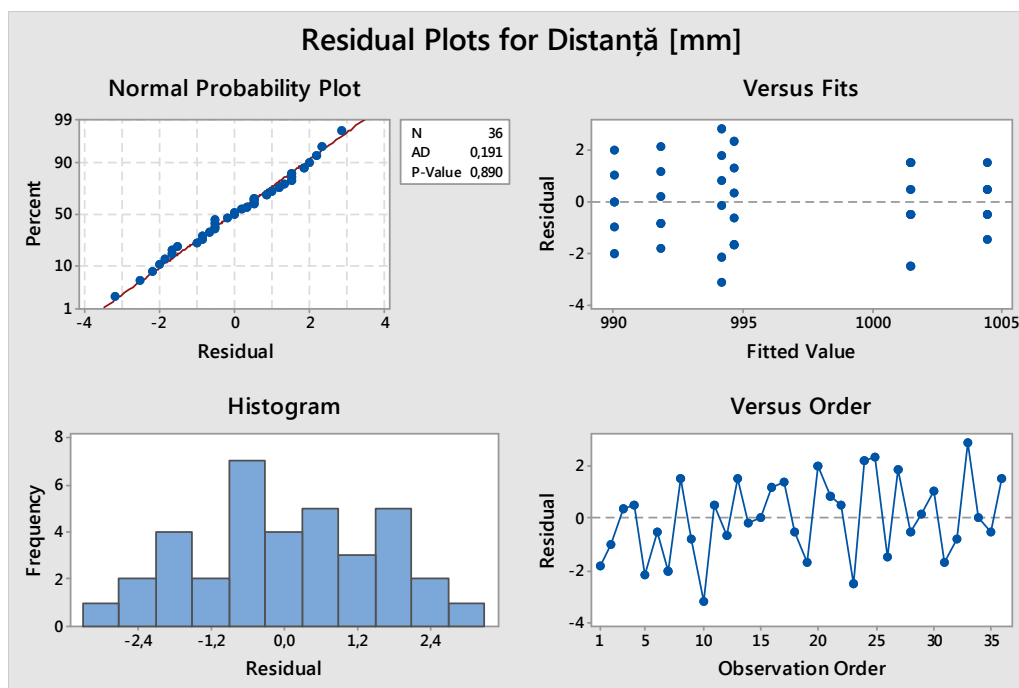


Fig. 5.8 Analiza valorilor reziduale pentru cele șase tipuri de camere

5.1.2. Analiza măsurării distanțelor la diferite fluxuri luminoase

În tabelul 5.2 se observă distanța de 1 m măsurată, utilizând camerele video, la diferite intensități luminoase. Se poate vedea că în principiu valorile nu diferă mult de valoarea nominală de 1 m, deoarece programul efectuează egalizare de histogramă de culori și auto calibrare de culori.

Tabel 5.2 Măsurarea distanței de 1 m folosind camere video la diferite fluxuri luminoase

Putere [W]	Flux Luminos [lm]	Distanță [mm]
25	375	996
40	600	996
60	900	1002
75	1125	1003
100	1500	997
150	2250	1004
200	3000	995

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. Histograma mai de grabă

constă din două bucăți, ceea ce denotă că există valori exacte de 1 m, dar există valori foarte apropiate de sub sau peste 1 m. Pentru așa puține valori nici nu ne putem aștepta la o histogramă ideală. Graficul de probabilitate normală denotă același lucru, că valorile sunt foarte apropiate de 1 m, toate fiind sub sau peste această valoare, nici una exact 1 m. Valoarea AD (Anderson-Darling) este 0,574 și valoarea p este 0,084, deci peste $\alpha = 0,05$, astfel ipoteza nulă nu este rejectată, astfel se poate afirma că nu sunt diferențe mari între valorile distanței măsurate la diferite fluxuri luminoase. Cele mai importante valori, sunt valorile de la concluziile graficelor, unde se observă valoarea indexului de capabilitate a procesului, $C_{pk} = 1,32$, care are o valoare foarte bună, deoarece la $C_{pk} = 1,33$, procesul este de 4σ , ceea ce este prezent în toate mediile industriale pentru un proces existent, care funcționează al toleranțe minime. Valoarea indexului de performanță a procesului este $P_{pk}=1,48$, care este mai mare ca și C_{pk} , cum era și de așteptat, la fel și valorile C_p și P_p sunt mai mari. 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.

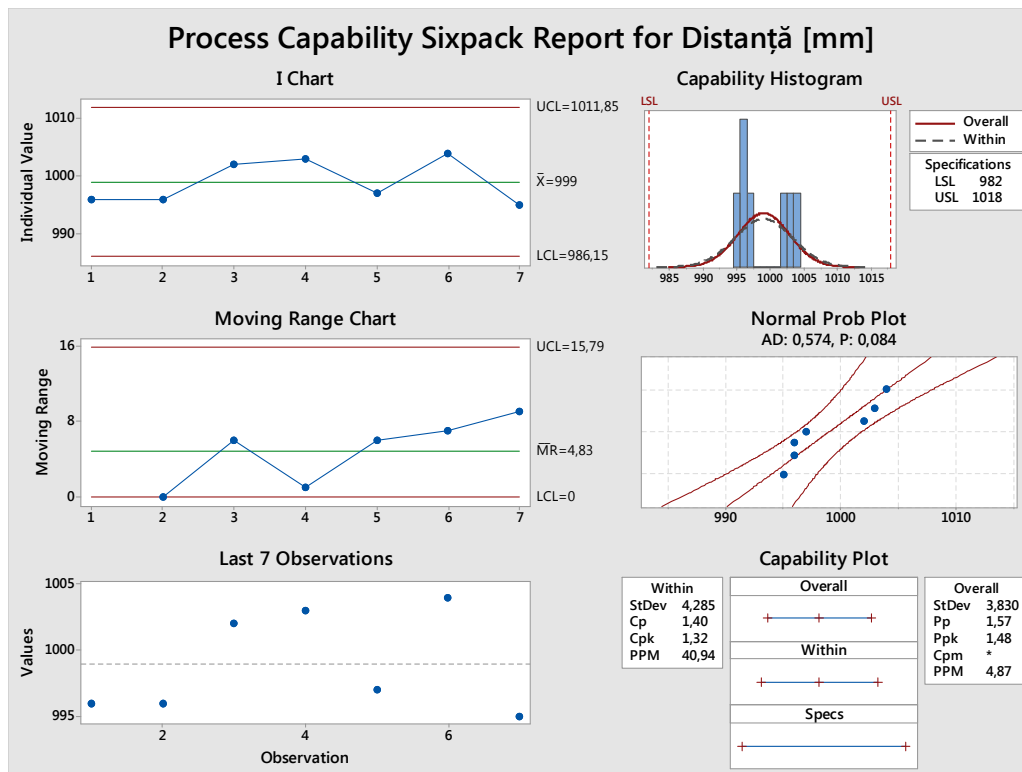


Fig. 5.9 Analiza de capabilitate pentru măsurarea distanței de 1 m folosind camere video la diferite fluxuri luminoase

Dacă este analizată histograma din fig. 5.10, acesta conține valorile care se pot interpreta. S-a făcut analiza pe doar 7 măsurători, adică la 7 tipuri de flux luminos. Media este foarte apropiată de 1 m, adică 999 mm. Deviația standard la toate măsurătorile este 3,8297 mm, iar la subgrup este 4,2849 mm. $C_{pk} = 1,32$,

adică cum se observă și din valoarea Z de 3,94, adică aproape 4σ și $P_{pk} = 1,48$, adică cum reiese și din valoarea Z de 4,42, adică mult peste 4σ . $P_p = 1,57$ și $C_p = 1,4$ sunt mai mari ca C_{pk} și P_{pk} . PPM-ul, adică eroarea pe milion trebuie rotunjită la un număr întreg, deci valoarea este 5 la toate măsurătorile și 41 la subgrup.

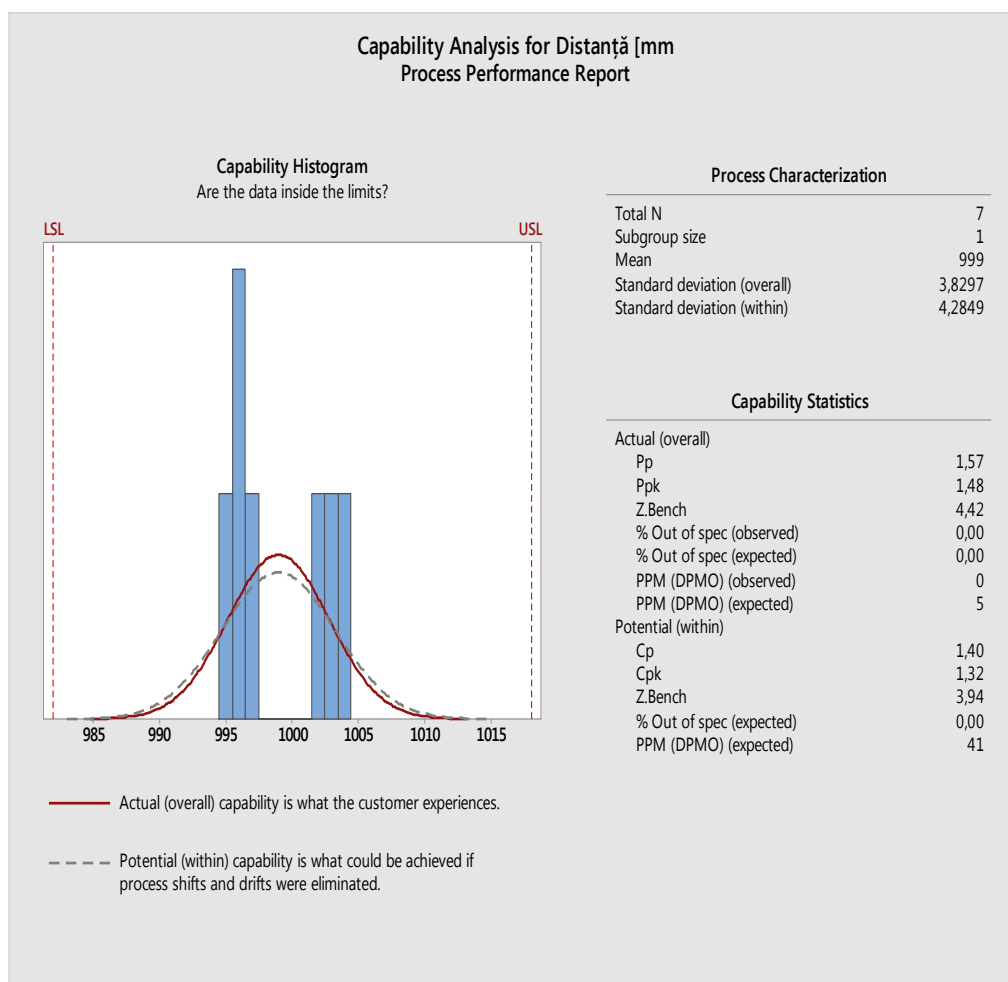


Fig. 5.10 Histograma de capabilitate pentru măsurarea distanței de 1 m folosind camere video la diferite fluxuri luminoase

Precum se observă din fig. 5.11 analiza I-MR (valoare individuală și limită mobilă), se poate spune că măsurătorile sunt în limite, iar testul de normalitate (Anderson-Darling) este trecut cu valoarea p de 0,084, care este peste valoarea limită de $\alpha = 0,05$, deci ipoteza nulă nu este rejectată, astfel se poate afirma că nu sunt diferențe mari între valorile măsurate.

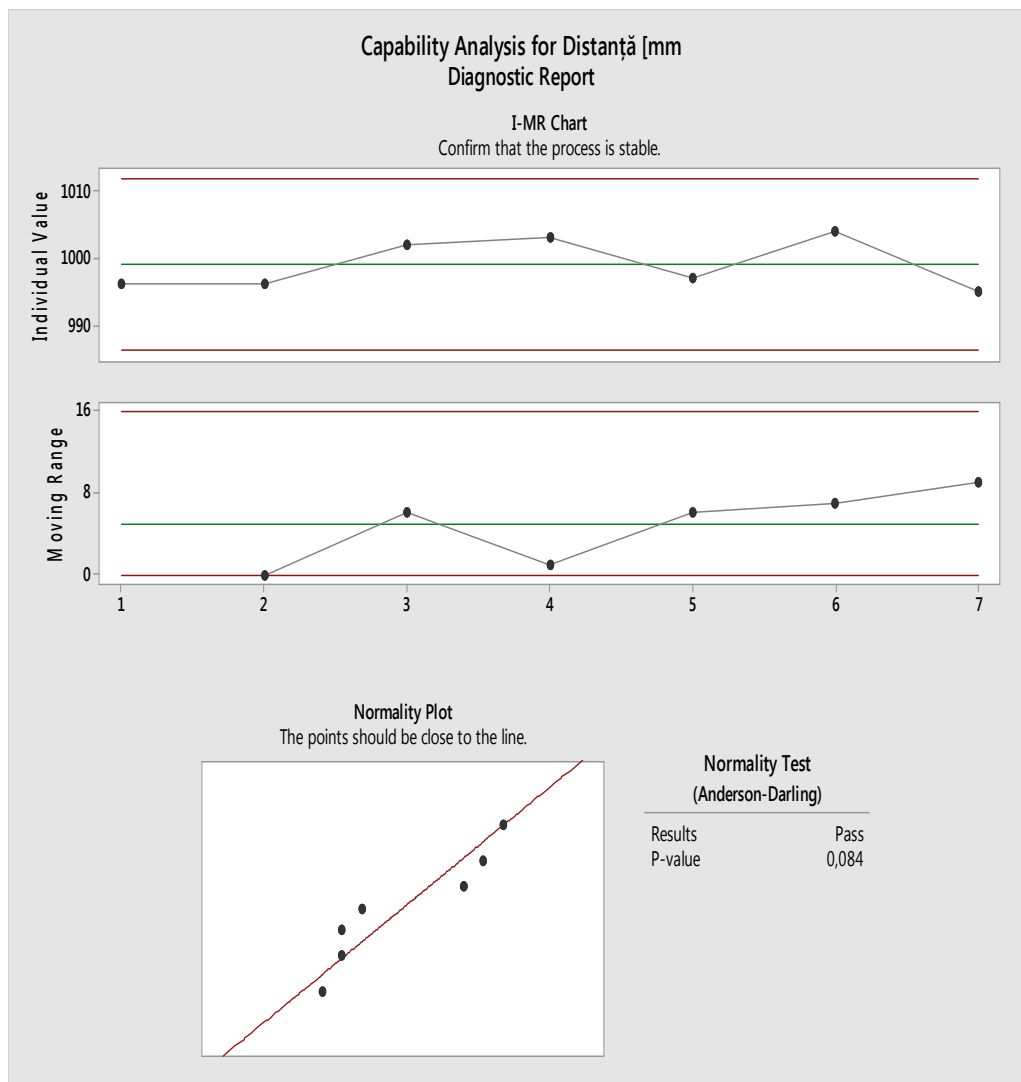


Fig. 5.11 Analiza I-MR (valoare individuală – limită mobilă) și testul de normalitate pentru măsurarea distanței de 1 m folosind camere video la diferite fluxuri luminoase

În fig. 5.12 se observă și raportul creat la testul de capacitate a sistemului de măsurare. Putem vedea aceleași valori mai sus amintite. Se observă o evidențiere a valorilor Z pentru C_{pk} și P_{pk} , care sunt în jur de 4σ , care sunt foarte bune pentru un proces actual. 6σ este valoarea pentru procesul perfect, inexistent în lumea reală. Se pot vedea și comentarii, deoarece sistemul are un PPM cu valoarea 5, adică 5 erori la milion, astfel la raport apare valoarea 0,00%. Erorile sunt estimate, dar nu sunt prezente în raportul actual. Interesul se concentrează pe valoarea tuturor măsurărilor, nu pe subgrup.

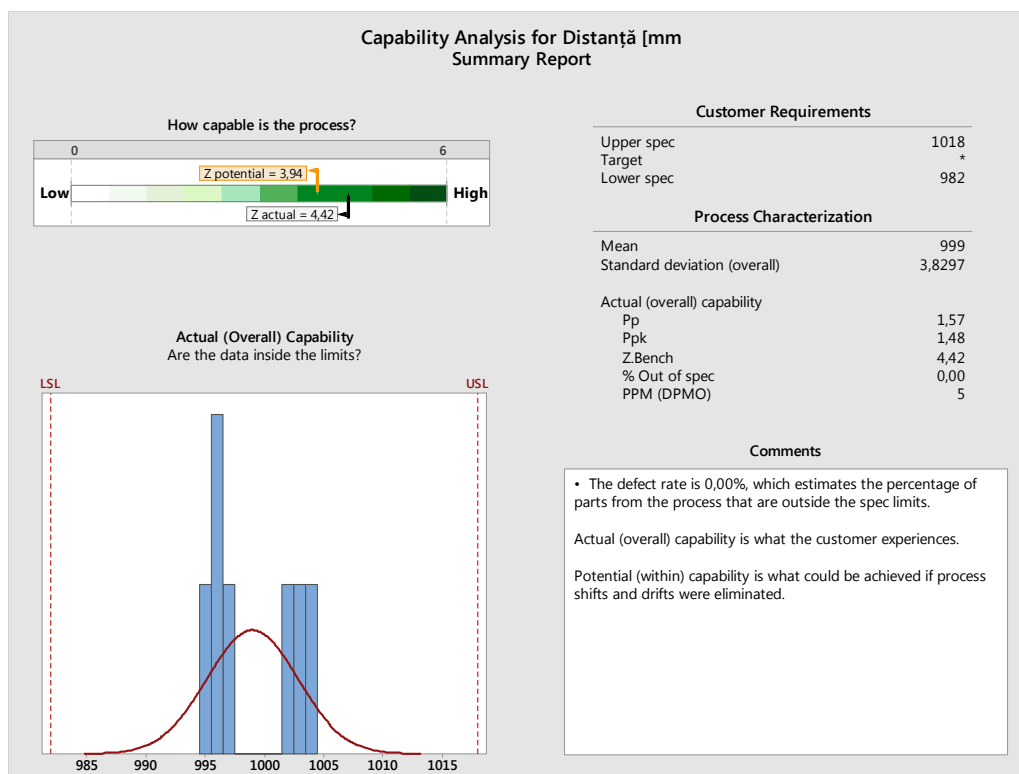


Fig. 5.12 Sumarul analizei de capabilitate pentru măsurarea distanței de 1 m folosind camere video la diferite fluxuri luminoase

5.1.3. Analiza măsurării distanțelor la diferite temperaturi de culoare

În Tabelul 5.3 se observă măsurarea distanței de 1 m folosind camerele video la diverse temperaturi de culoare. Se poate vedea că distanța măsurată nu diferă mult de la distanța reală de 1 m. Acest lucru este și dat faptul că sistemul face o egalizare de histogramă și astfel sunt eliminate influențele externe.

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. 5.13 se observă analiza de capabilitate. Graficele I-MR, adică graficul I (valoare individuală) și graficul MR (limită mobilă) sunt în limitele marcate cu linie roșie. Histograma nu prea seamănă cu o histogramă, deoarece sunt mult prea puține valori, există doar 3. Graficul de probabilitate normală arată că valorile toate

sunt foarte apropiate de valoarea reală de 1000 mm. Se poate vedea că valoarea AD (Anderson-Darling) este 0,212 și valoarea p este 0,536, deci mai mare ca $\alpha = 0,05$, deci ipoteza nulă nu este rejectată, astfel se poate afirma că nu sunt diferențe mari între valorile distanței măsurate la diferite temperaturi de culoare. Indexul de capacitate al procesului, $C_{pk}=1,57$, este mult mare ca 1,33, deci este un proces mult peste 4σ . Indexul de performanță al procesului, $P_{pk}=1,37$, este de asemenea peste 4σ , iar valorile C_p și P_p sunt mult mai mari. Deviația standard la întreg procesul este 4,041 mm, iar la subgrup este 3,546 mm, care este o valoare bună. PPM-ul, adică erorile la părți pe milion, la întreg procesul, este 19,48, iar la subgrup este 1,33. Aceste valori sunt, de asemenea, valori bune.

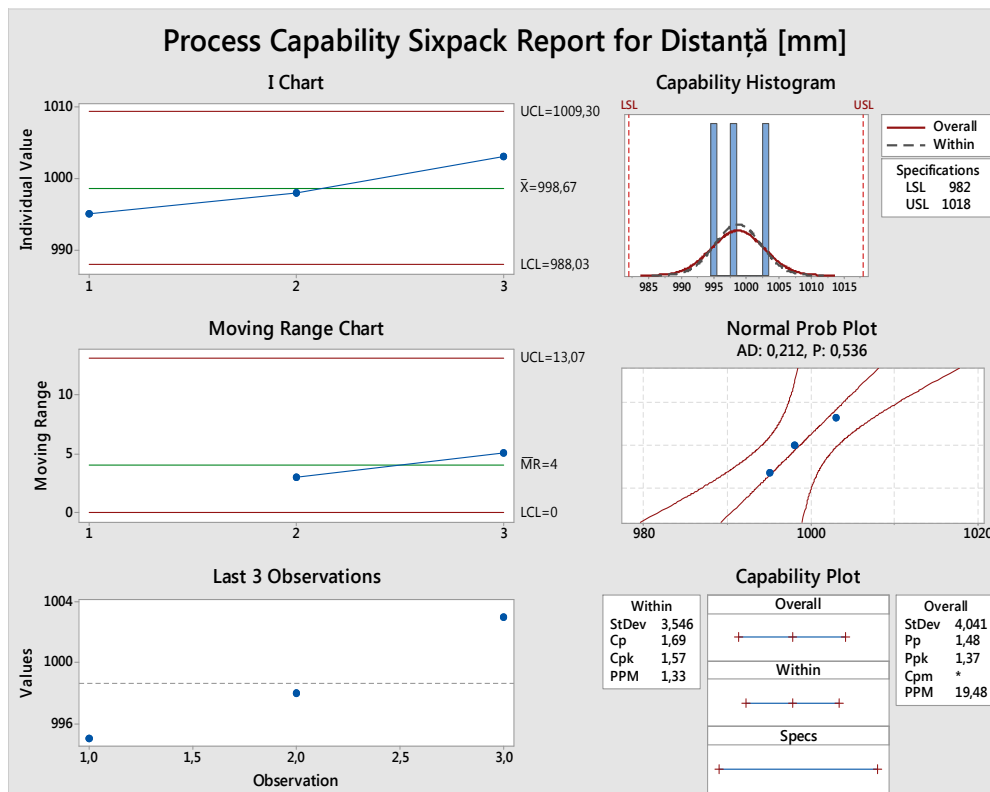


Fig. 5.13 Analiza de capacitate pentru măsurarea distanței de 1 m folosind camere video la diferite temperaturi de culoare

După ce este analizată fig. 5.14, media este de 998,67 mm, deviația standard este 4,0415 mm la întreg procesul și 3,5461 mm la subgrup. La întreg procesul, $P_p=1,48$, $P_{pk}=1,37$, astfel valoarea Z este 4,11, deci este un proces 4σ . Erorile la părți pe milion (PPM) sunt în număr de 19. La subgrup, $C_p=1,69$ și $C_{pk}=1,57$, deci valoarea Z este de 4,7, mult peste 4σ . PPM-ul este 1, deci este o singură eroare la părți pe milion. Acest lucru denotă că măsurarea este foarte bună.

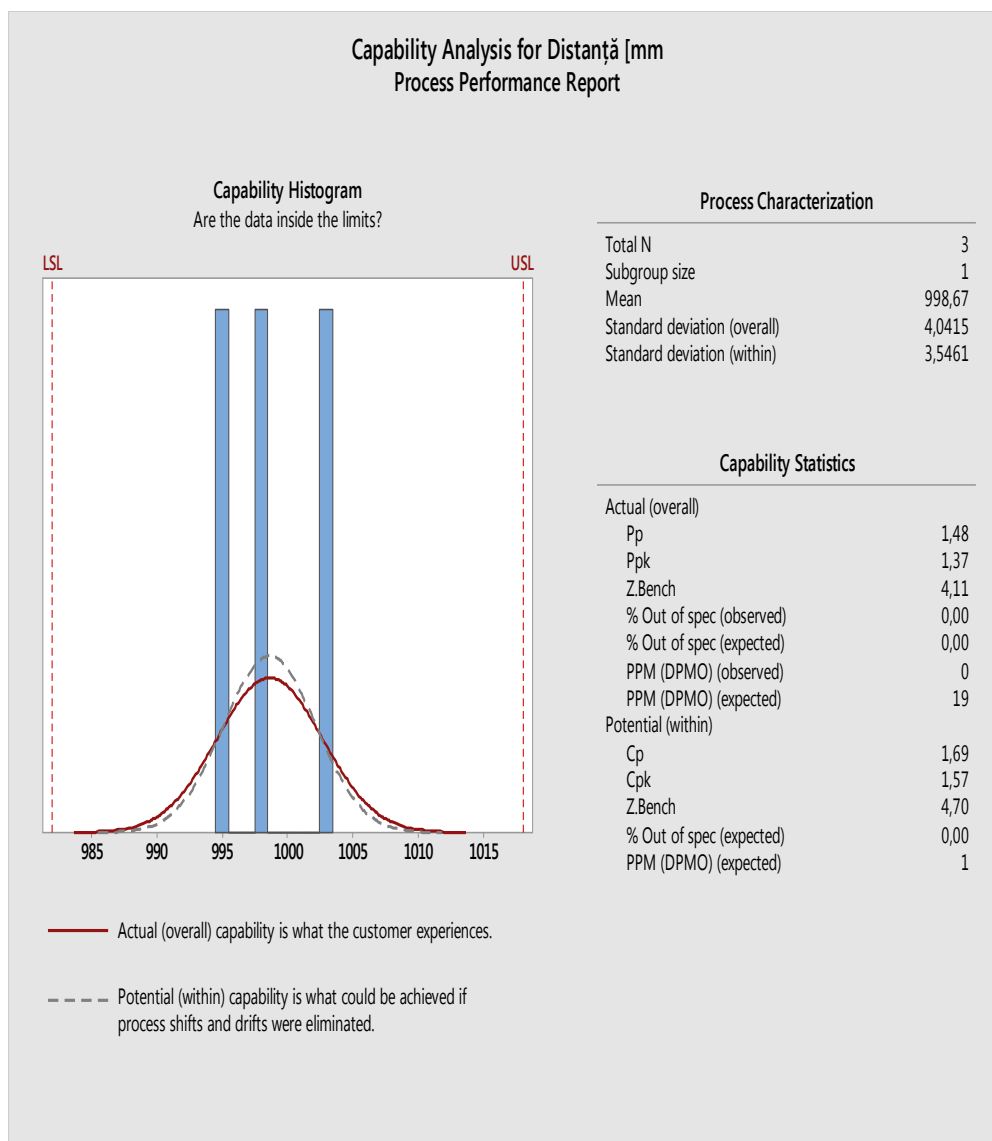


Fig. 5.14 Histograma de capabilitate pentru măsurarea distanței de 1 m folosind camere video la diferite temperaturi de culoare

Dacă se analizează fig. 5.15, analiza I-MR (valoare individuală și limită mobilă), se poate spune, că măsurătorile sunt în limitele trasate cu linie roșie, iar testul de normalitate (Anderson-Darling) este trecut cu valoarea p de 0,536, deci peste valoarea limită $\alpha = 0,05$, deci ipoteza nulă nu este rejectată, astfel se poate afirma că nu este prea mare diferență între valorile măsurate.

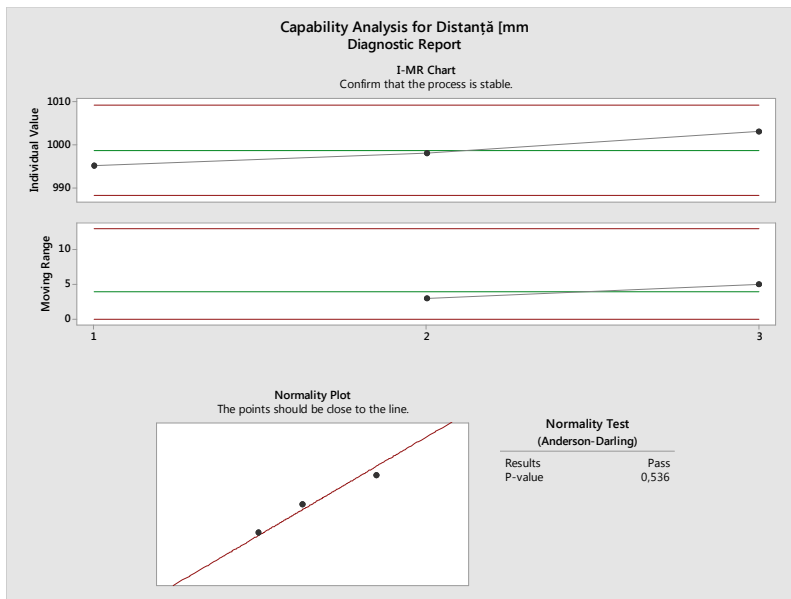


Fig. 5.15 Analiza I-MR (valoare individuală – limită mobilă) și testul de normalitate pentru măsurarea distanței de 1 m folosind camere video la diferite temperaturi de culoare

În fig. 5.16 se observă raportul la analiza de capabilitate al sistemului de măsurare. Sunt evidențiate valorile Z, care sunt 4,11, la întreg procesul și 4,7 la subgrup, deci toate cu mult peste 4σ . Rata de eroare este 0,00%, deoarece PPM-ul este 19. Aceste informații exprimă că sistemul de măsurătoare este foarte bun și nu este influențat nici de temperatura de culoare.

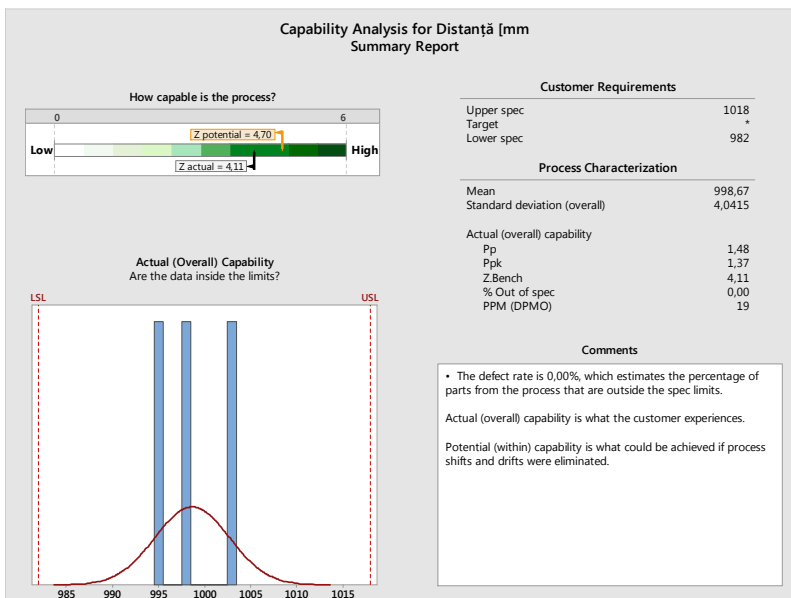


Fig. 5.16 Sumarul analizei de capabilitate pentru măsurarea distanței de 1 m folosind camere video la diferite temperaturi de culoare

5.1.4. Analiza măsurării distanțelor în diferite poziții

În tabelul 5.4, sistemul este testat făcând măsurări la diferite distanțe. Plaja de valori utilizate pentru măsurare este între 100 mm și 3000 mm. Se poate observa valoarea reală măsurată cu o metodă laser și distanța calculată de sistemul implementat, folosind camere video. S-a calculat și delta (eroarea), care este diferența dintre valoarea reală și valoarea măsurată cu sistemul de camere.

Tabel 5.4 Măsurarea distanței folosind camere video la diferite distanțe [mm] cu evidențierea erorii de măsurare [mm]

Distanță Reală [mm]	Distanță Calculată [mm]	Delta [mm]
100	99	1
200	202	-2
300	303	-3
400	395	5
500	502	-2
600	598	2
700	699	1
800	797	3
900	904	-4
1000	1005	-5
1100	1101	-1
1200	1204	-4
1300	1298	2
1400	1402	-2
1500	1494	6
1600	1599	1
1700	1701	-1
1800	1802	-2
1900	1905	-5
2000	2005	-5
2100	2105	-5
2200	2202	-2
2300	2302	-2
2400	2404	-4
2500	2496	4
2600	2598	2
2700	2697	3

2800	2796	4
2900	2902	-2
3000	3006	-6

În fig. 5.17 se observă histograma valorilor reale, suprapusă cu histograma valorilor măsurate. Se observă că, cele două histograme, aproape se suprapun. Pe cele 30 de valori măsurate, ambele histograme au media în jur de 1550 mm și deviația standard în jur de 880 mm.

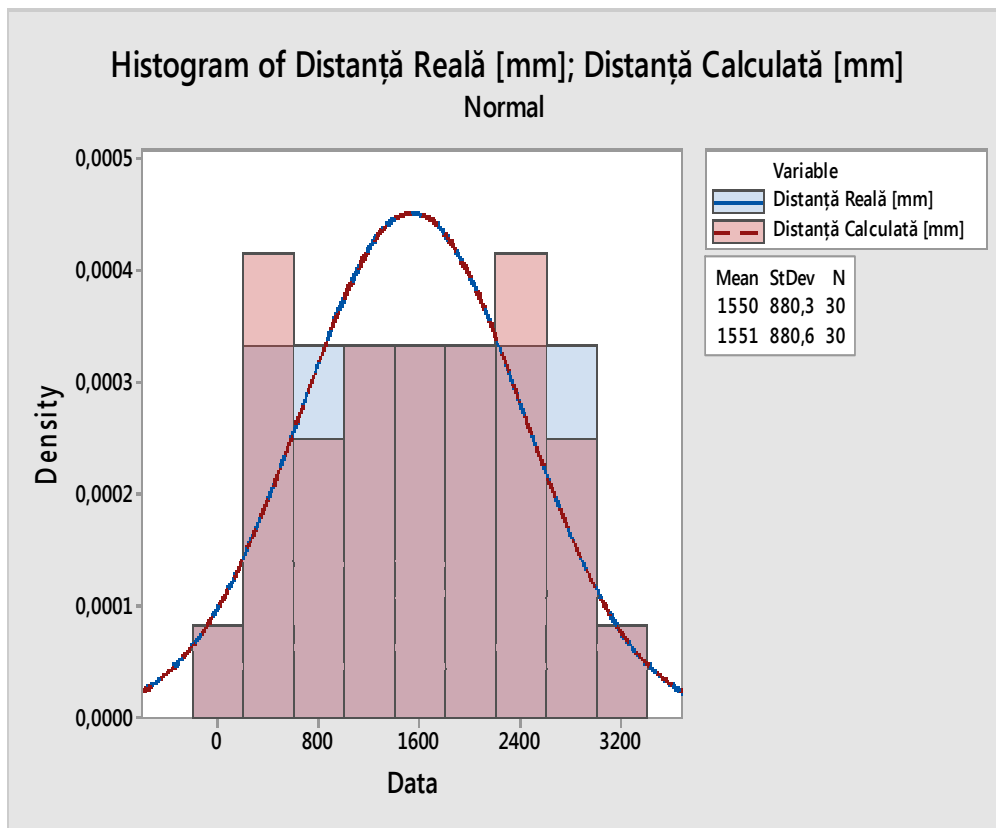


Fig. 5.17 Histograma distanței reale și distanței calculate de sistemul folosind camere video

În fig. 5.18 se observă graficul de distribuție normală, cu un interval de încredere de 95% a celor două măsurători, distanța reală și distanța măsurată cu sistemul de camere video. Se observă că și aceste grafice se suprapun aproape perfect, deci distanța măsurată, nu diferă foarte mult, de distanța reală. La 30 de valori valoarea AD (Anderson-Darling) este 0,321, și valoarea p este 0,514, deci mai mare ca $\alpha = 0,05$, deci ipoteza nulă nu este rejectată, se poate afirma că valorile de măsurare a distanței nu diferă prea mult între ele.

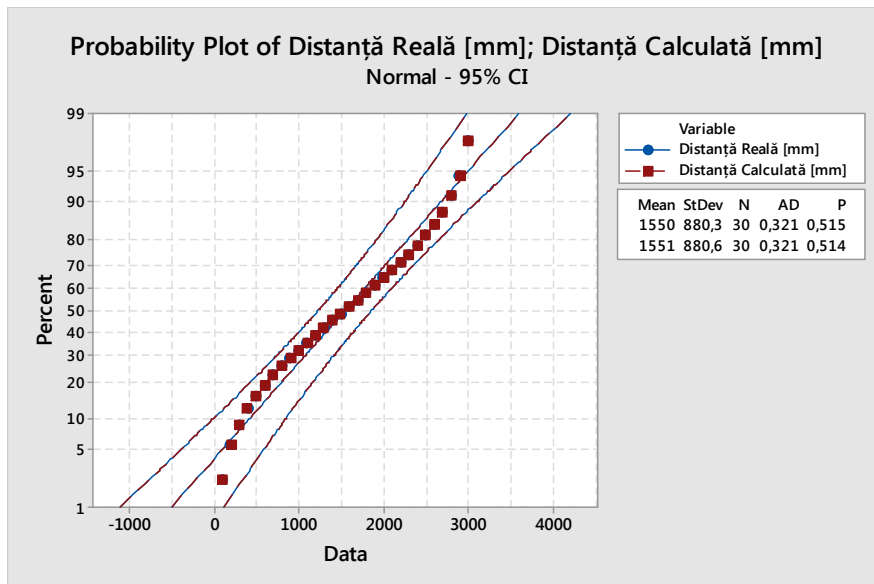


Fig. 5.18 Graficul de probabilitate normală pentru distanța reală și distanța calculată de sistemul folosind camere video

În fig. 5.19 se observă graficul de distribuție normală, cu un interval de încredere de 95%, a erorii delta, adică diferența dintre valoarea reală și valoarea măsurată. Media de $-0,7667$ mm este destul de bună. Deviația standard este de $3,38$ mm. Valoarea AD (Anderson-Darling) este $0,643$ și valoarea p de $0,084$, deci mai mare ca $\alpha = 0,05$, deci ipoteza nulă nu este rejectată, deci nici erorile delta nu diferă așa de mult între ele.

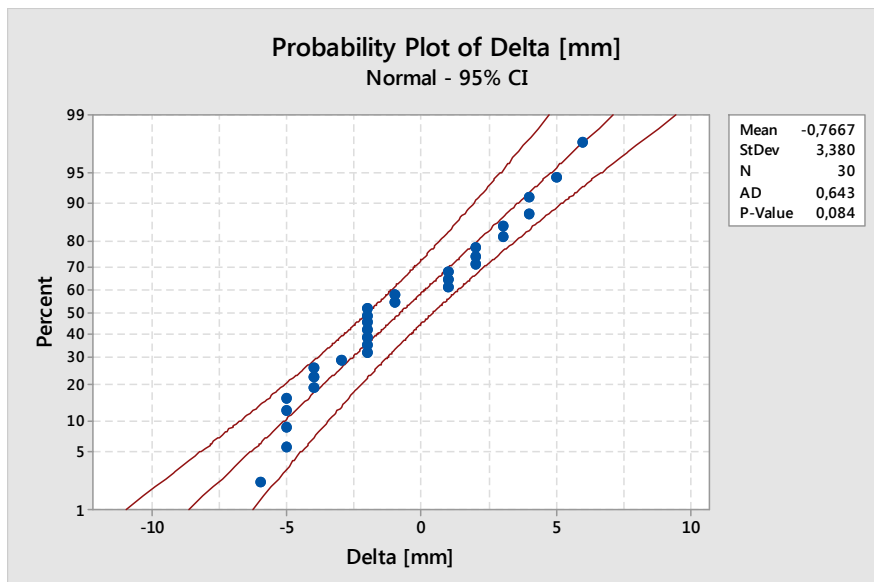


Fig. 5.19 Graficul de probabilitate normală pentru eroare (delta)

În fig. 5.20 se observă graficul de distribuție normală, cu un interval de încredere de 95%, a erorii delta absolut, delta este pus în modul pentru a vedea eroarea absolută. Media la eroarea absolută este de 3,033 mm. Deviația standard este de 1,586 mm. Valoarea AD (Anderson-Darling) este 1,261, iar valoarea $p < 0,005$, deci mai mică ca 0,05, deci ipoteza nulă este rejectată, valorile diferă mai mult între ele, ceea ce era și de așteptat, deoarece delta absolut este o valoare calculată, nu reiese din măsurători reale, s-a folosit pentru a vedea media maximă de eroare în modul, care este de valoare foarte bună, doar cu puțin peste 3 mm.

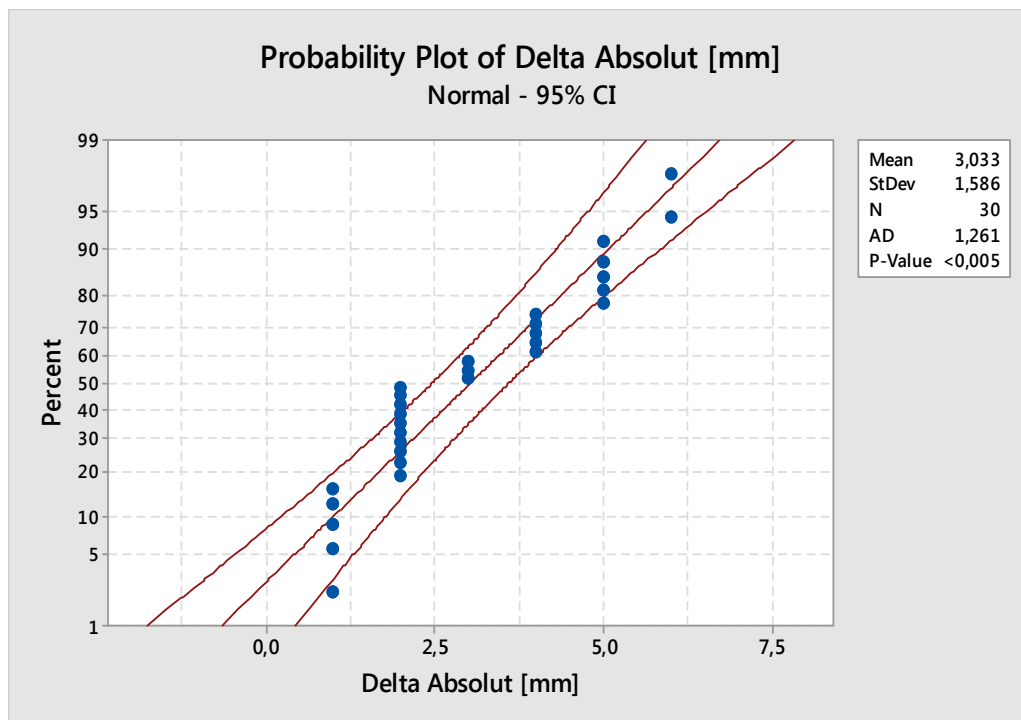


Fig. 5.20 Graficul de probabilitate normală pentru eroarea absolută (delta absolut)

În fig. 5.21 se observă graficul de potrivire pe linie, care este o metodă de analiză de regresie. Este reprezentată distanța calculată în funcție de distanța reală. Valorile obținute prima oară sunt cele de la distanța reală, iar apoi sunt obținute și valorile pentru distanța calculată. Deviația standard pentru aceste valori este 3,42693 mm. Valorile R pătrat (R^2) și R pătrat ajustat sunt 100%, ceea ce denotă că răspunsurile pot fi perfect prezise cunoscând valorile de intrare, adică dacă se cunoaște distanța măsurată, se poate obține distanța calculată de algoritmul de calcul al distanțelor, folosind camere video. Cu alte cuvinte, sistemul nu are multe erori, deoarece se poate obține distanța calculată cunoscând valorile de intrare, care sunt valorile de distanță măsurate. Pentru asta cu analiza de regresie s-a obținut și o formulă empirică prezentă și pe grafic.

$$dis\ tan\ ta\ _\ calculata[mm] = 0,246 + 1,000 \cdot dis\ tan\ ta\ _\ reala[mm]$$

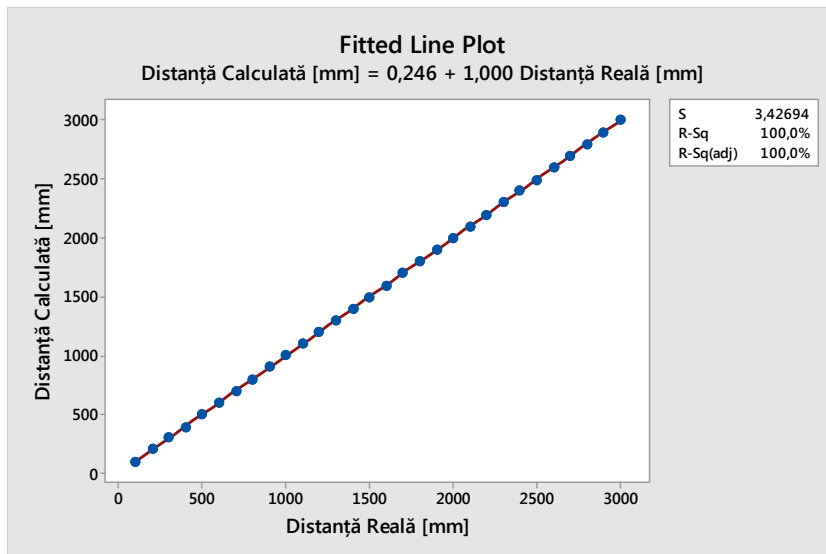


Fig. 5.21 Analiza de regresie pentru distanța reală și distanța calculată de sistemul folosind camere video

În fig. 5.22 se văd valorile reziduale obținute, după graficul de potrivire pe linie. Se poate vedea valoarea AD (Anderson-Darling) care este 0,549 și valoarea p care este 0,145, deci mai mare ca $\alpha = 0,05$, deci ipoteza nulă nu este rejectată, astfel se poate afirma că valorile reziduale nu diferă prea mult între ele. Histograma nu seamănă cu o distribuție gaussiană, dar la un așa număr mic de valori nici nu se poate aștepta la mai bine. Graficele versus arată o distribuție mare a valorilor, ceea ce este normal și era de așteptat.

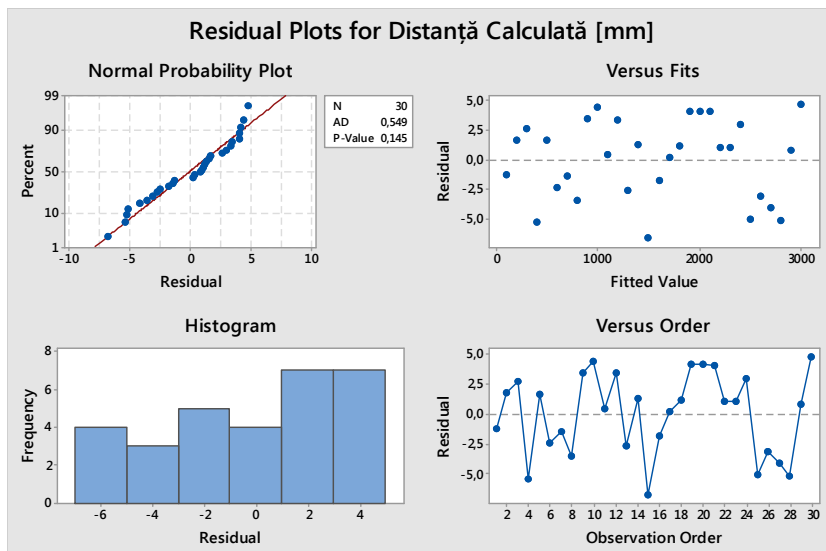


Fig. 5.22 Analiza valorilor reziduale pentru distanța reală și distanța calculată de sistemul folosind camere video

În fig. 5.23 este redată o analiză de regresie mai detaliată, unde se observă graficul versus, care indică că nu sunt probleme cu modelul de regresie, deoarece valorile sunt dispersate aleator peste și sub 0. Tot pe grafic este reprezentat cu punct roșu și o valoare reziduală mare la coordonata (-6, 1500), adică la măsurarea de 1500 mm este cea mai mare eroare delta de 6 mm.

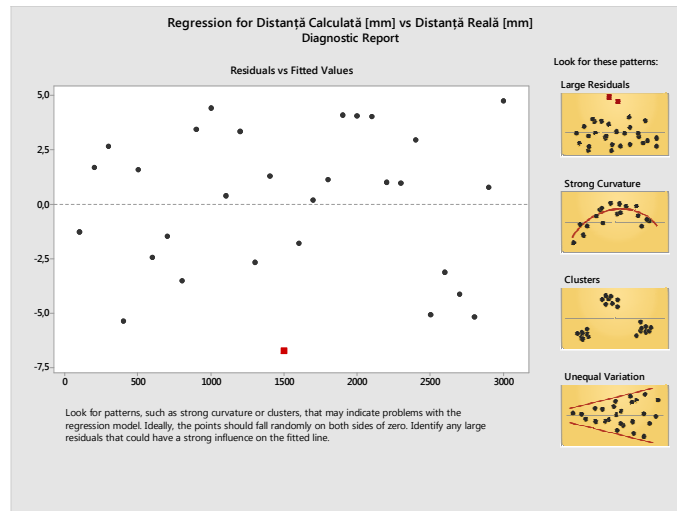


Fig. 5.23 Graficul de valori reziduale și valori potrivite pentru distanța reală și distanța calculată de sistemul folosind camere video

În fig. 5.24 se continuă analiza de regresie, între distanța reală și cea calculată de algoritmul de măsurare a distanței folosind camere video. Pentru fiecare valoare de intrare X (distanța reală) există o predicție Y (distanța calculată) folosind un interval de predicție de 95% pentru $\alpha = 0,05$.

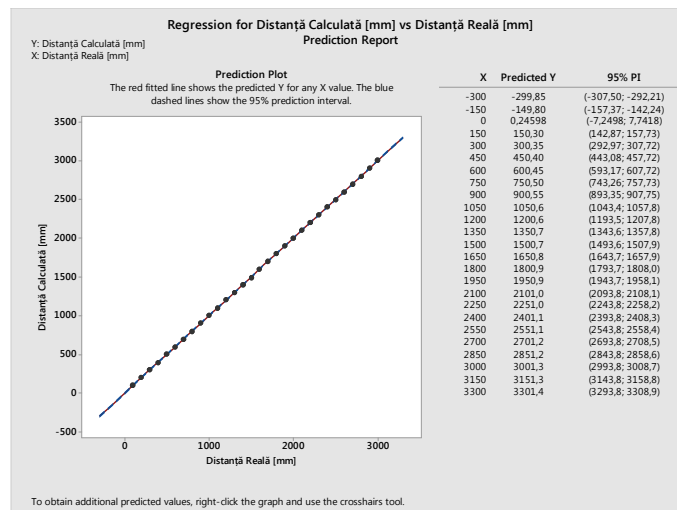


Fig. 5.24 Graficul de predicție pentru distanța reală și distanța calculată de sistemul folosind camere video

În fig. 5.25 se continuă analiza de regresie, cu evidențierea anumitor valori importante, pentru analiza de regresie. Vedem valoarea R pătrat (R^2) ajustat la 100%, ceea ce denotă că valorile de ieșire (distanța calculată) pot fi 100% prezise din valorile de intrare (distanța reală). Valoarea p în general este mai mică ca și valoarea $\alpha = 0,05$, deci ipoteza nulă este rejectată, astfel se poate afirma că sunt diferențe suficient de mari între valorile distanței calculate și măsurate, ceea ce este necesar pentru a putea face o relație matematică între acestea. Deviația standard reziduală la modelul liniar este 3,427 mm, iar la modelul cvadratic alternativ este 3,468 mm, care sunt valori foarte bune și erau și de așteptat. Pe grafic se observă și formula empirică obținută după analiza de regresie și cea mai mare valoare reziduală de 6 mm la măsurarea distanței de 1500 mm.

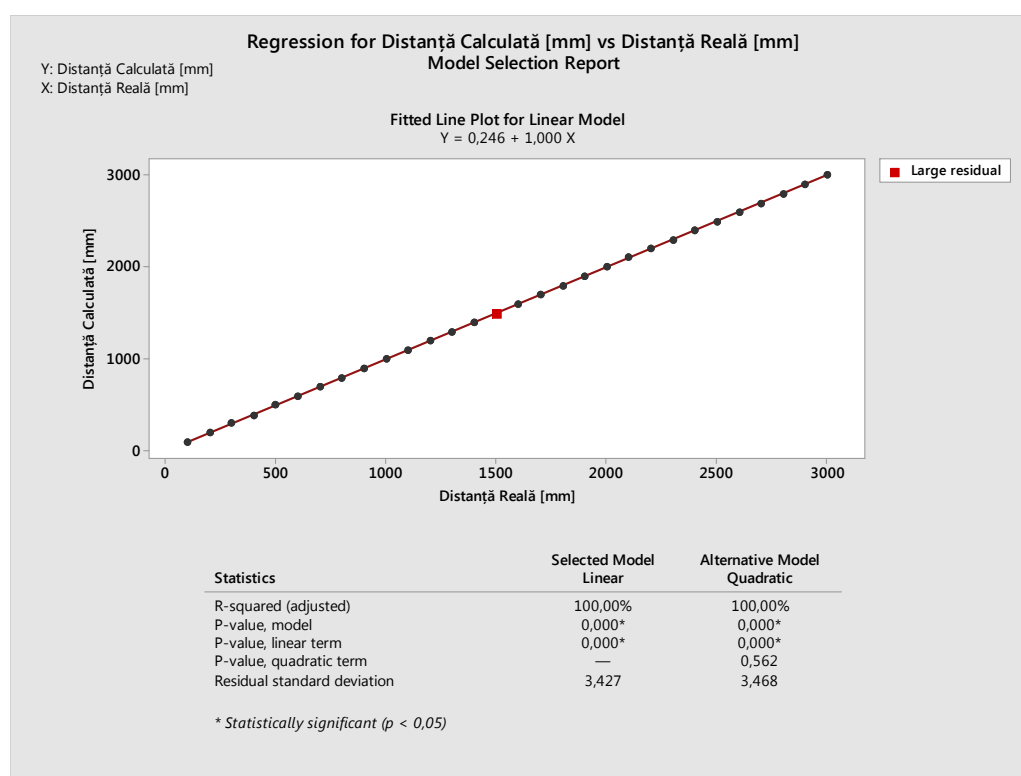


Fig. 5.25 Graficul de valori potrivite pe o dreaptă pentru distanța reală și distanța calculată de sistemul folosind camere video

În fig. 5.26 este prezentat și raportul final, pentru analiza de regresie. Se poate vedea că valoarea $p < 0,001$, care este mai mică ca $\alpha = 0,05$, deci ipoteza nulă este rejectată, astfel valorile măsurate diferă suficient de mult între ele, încât să se poate forma o relație matematică între Y (distanța calculată) și X (distanța reală). Variația modelului, adică valoarea R pătrat (R^2) este 100%, deci valorile de ieșire (distanța calculată) pot fi prezise 100% din valorile de intrare (distanța reală), cu alte cuvinte variația modelului este foarte mică sau inexistentă sau că valorile măsurate cu sistemul de camere video sunt foarte apropiate de valoarea reală măsurată cu laser. Există o corelare perfectă între Y (distanța calculată) și X (distanța reală) folosind ecuația empirică obținută de analiza de regresie.

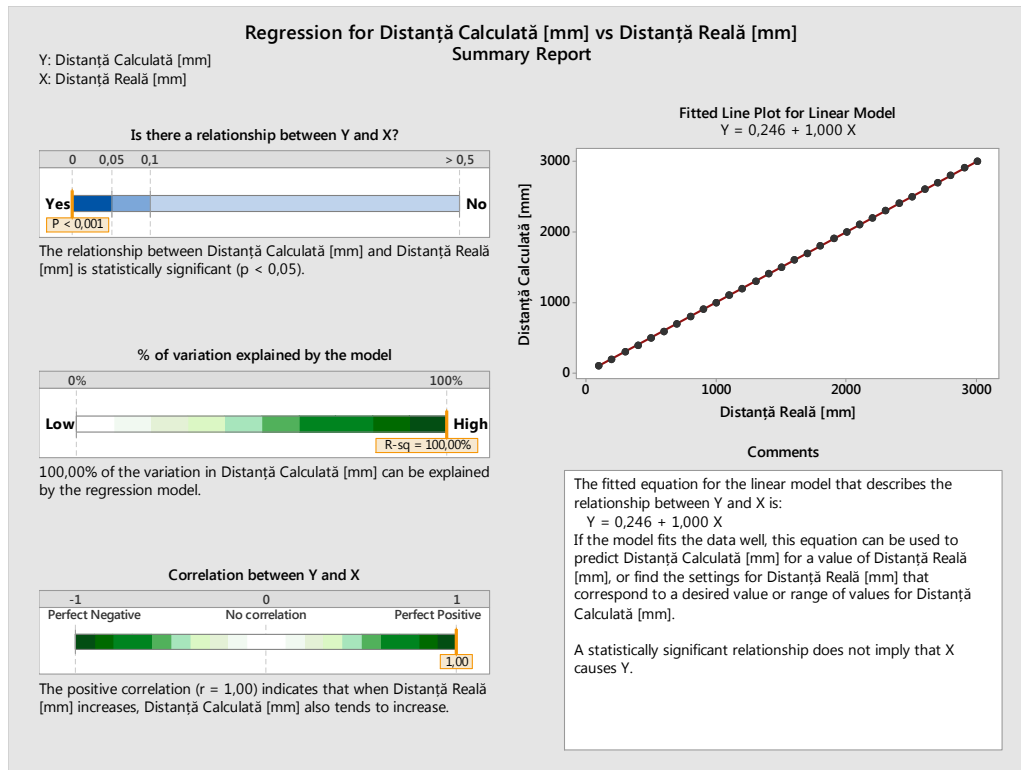


Fig. 5.26 Sumarul analizei de regresie pentru distanța reală și distanța calculată de sistemul folosind camere video

În fig. 5.27 este prezentată și o analiză de capabilitate, a celor 30 de măsurări de distanță. Graficele I-MR (valoare individuală și limită mobilă) sunt în limite. Histograma are o tendință gaussiană, însă pentru o histogramă, 30 de valori, sunt foarte puține, ceea ce se observă și pe histogramă, este că nu există valori măsurate la care eroarea delta să fie 0, adică nu există măsurătoare fără eroare, însă nu sunt probleme foarte mari, deoarece aceste erori sunt foarte mici. La graficul de probabilitate normală, valoarea AD (Anderson-Darling) este 0,643 și valoarea p este 0,084, deci peste $\alpha = 0,05$, astfel ipoteza nulă nu este rejectată, astfel se poate afirma că nici erorile delta nu diferă prea mult între ele. Cele mai importante sunt valorile din concluzii, aici deviația standard este 2,965 mm la subgrup și 3,38 mm la toate măsurătorile. Valoarea C_{pk} este 1,38, adică peste 4σ , ceea ce este un rezultat foarte bun la un sistem de acest gen. C_p este 1,46, P_p este 1,28, iar C_{pk} este 1,21, toate valorile sunt foarte aproape de 4σ , ceea ce este un rezultat foarte bun pentru un sistem care rulează în condiții normale. PPM-ul, adică eroarea raportată la un milion de măsurători, este 20,21 la subgrup și 171,11 la toate măsurările, această valoare este una foarte bună.

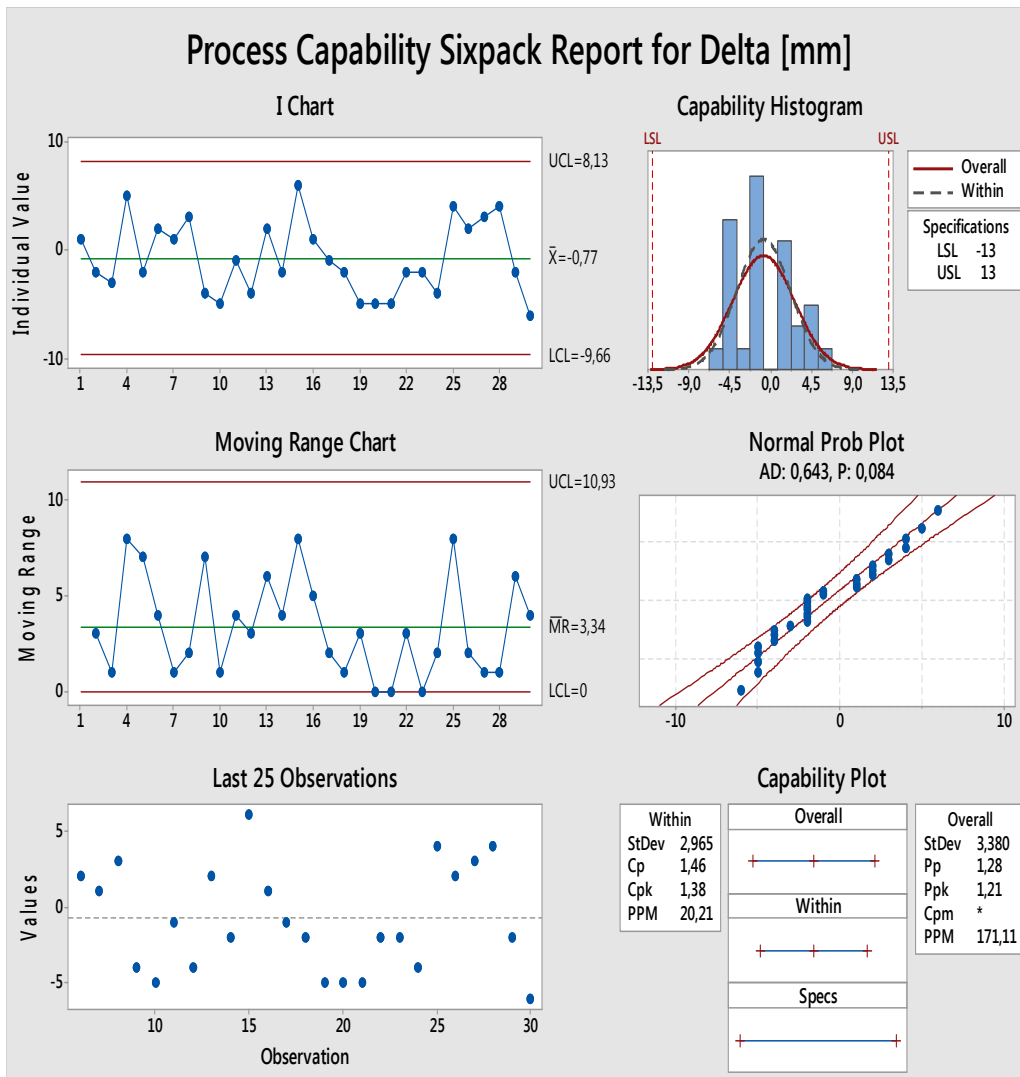


Fig. 5.27 Analiza de capabilitate pentru distanța reală și distanța calculată de sistem folosind camere video

În fig. 5.28 se observă histograma de la analiza de capabilitate, pentru valorile de eroare delta. Pentru cele 30 de valori delta măsurate, media de eroare este foarte bună, aceasta are valoarea $-0,76667$ mm. Celelalte rezultate se regăsesc și la concluziile graficului anterior. Se poate vedea evidențiat valorile Z calculate din C_{pk} și P_{pk} pentru toate măsurătorile, care indică $3,58\sigma$ pentru toate măsurătorile și $4,11\sigma$ pentru subgrup, ceea ce pentru o proces real este foarte bun. Se poate vedea și valoarea PPM rotunjită, pentru toate măsurătorile aceasta este 171 și pentru subgrup este 20.

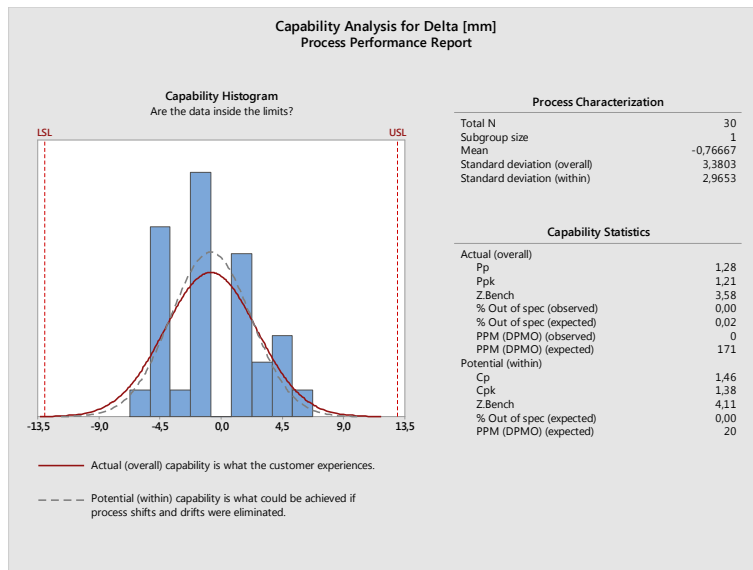


Fig. 5.28 Histograma de capabilitate pentru distanța reală și distanța calculată de sistemul folosind camere video

În fig. 5.29 se continuă analiza de capabilitate, al sistemului de măsurare a distanțelor, folosind camere video. Se pot vedea grafice I-MR (valoare individuală și limită mobilă), valorile pe aceste grafice sunt în limite. Testul de normalitate (Anderson-Darling) este trecut cu valoarea p de 0,084, care este mai mare ca $\alpha = 0,05$, deci ipoteza nulă nu este rejectată, deci se poate afirma că valorile delta nu diferă foarte mult între ele.

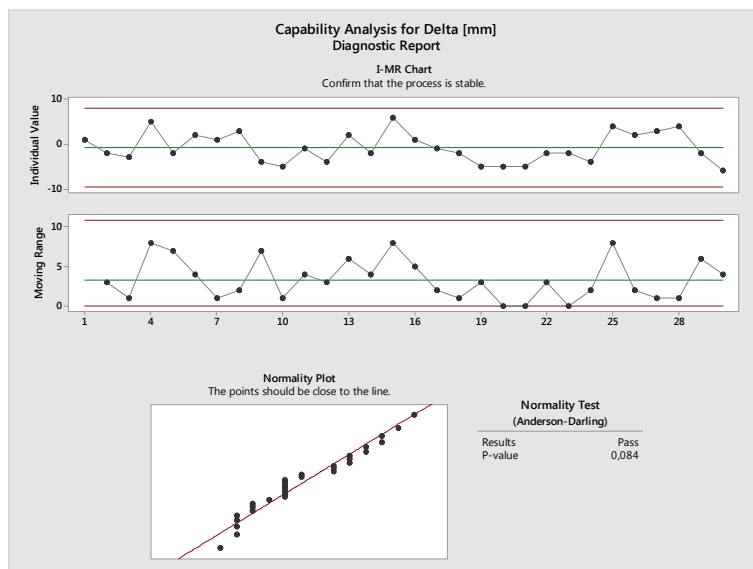


Fig. 5.29 Analiza I-MR (valoare individuală – limită mobilă) și testul de normalitate pentru distanța reală și distanța calculată de sistemul folosind camere video

În fig. 5.30 se observă raportul de la analiza de capacitate al sistemului de măsurare al distanțelor folosind camere video. Se poate vedea capacitatea procesului de măsurare, procesul este foarte apropiat de 4σ , ceea ce este un rezultat foarte bun pentru un proces real. Este reprezentată și media de $-0,76667$ mm și deviația standard de $3,3803$ mm. Procentul care nu se încadrează în limite este de doar $0,02\%$ pentru toate măsurătorile, iar valorile eronate măsurate la un milion de măsurători (PPM) este de 171 .

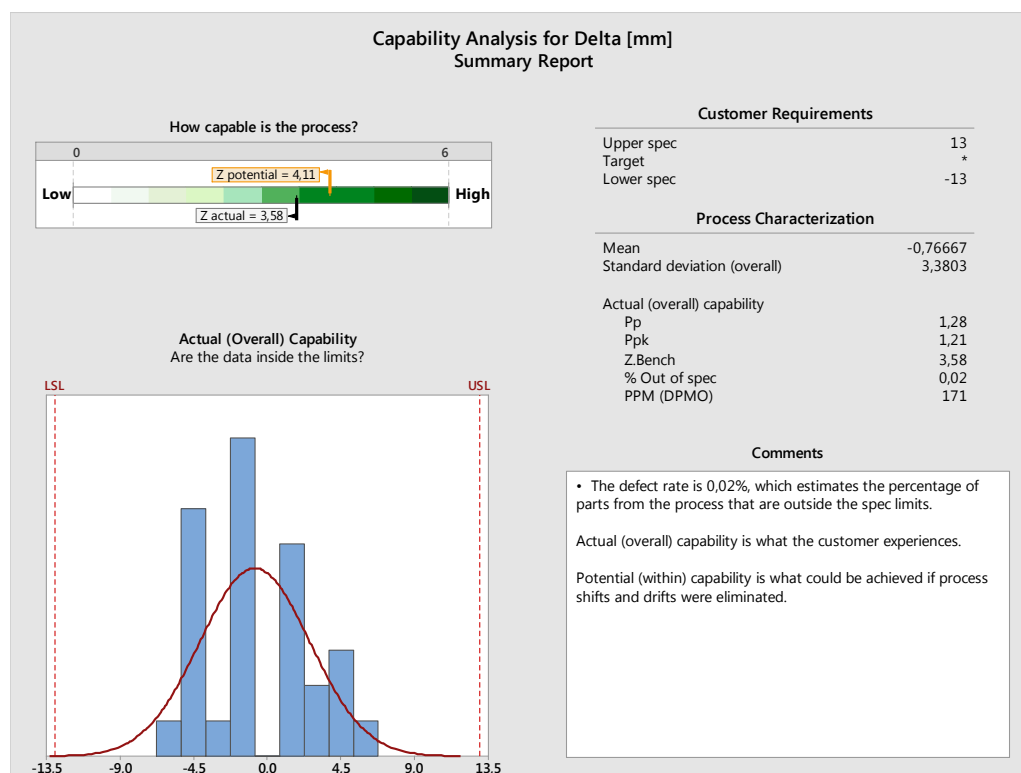


Fig. 5.30 Sumarul analizei de capacitate pentru distanța reală și distanța calculată de sistemul folosind camere video

În fig. 5.31 se observă fluctuația erorii delta, măsurată în mm, în funcție de distanța reală, măsurată cu 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. Acest lucru era și de așteptat, deoarece acest grafic reprezintă proporționalitatea, deci după acest grafic se poate afirma că erorile delta sunt în mare constante pe toată plaja de distanțe. Acest grafic este mai degrabă unul calitativ decât cantitativ, arată eroarea în funcție de creșterea distanței de măsurate.

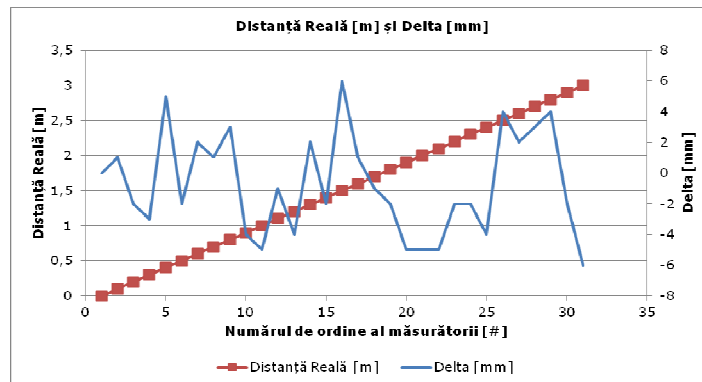


Fig. 5.31 Grafic de prezentare distanța reală [m] și eroarea de măsurare (delta) [mm]

5.2. Comparația metodei prezentate cu alte metode din literatură

În tabelul 5.5 este prezentat o comparație între sistemul care a fost prezentat în această teză și alte sisteme de control al brațelor robotice folosind camere video, realizate de alți cercetători și prezentate în reviste de specialitate.

Tabel 5.5 Comparația între metoda prezentată și alte metode similare

Caracteristici	Metoda din teză	M. Seelinger [38]	R. Kelly [39]	V. Lippiello [40]	M. Kazemi [41]
Număr articulații	3	6	2	multi deget	6
Cost	reduc	reduc	reduc	ridicat	reduc
Precizie	mare	mare	mare	mare	mare
Complexitate	reducă	medie	reducă	mare	mare
Utilizare memorie	reducă	medie	reducă	mare	mare
Caracteristici	R. T. Fomena [42]	L. Behera [43]	J. J. Heuring [44]	F. Chaumette [45]	In-Won Park [46]
Număr articulații	6	3	6	6	7 sau 4
Cost	reduc	reduc	reduc	reduc	reduc
Precizie	mare	mare	mare	mare	mare
Complexitate	mare	reducă	mare	mare	medie
Utilizare memorie	mare	reducă	mare	mare	medie

Se poate observa că sistemul prezentat este destul de bun în comparație cu alte sisteme similare, complexitatea și utilizarea memoriei este redusă. Singurul neajuns este că sistemul prezentat controlează momentan doar 3 articulații.

Restul de parametri sunt comparabili cu implementările din revistele de specialitate. Sistemul prezentat în cadrul acestei teze de doctorat se poate extinde la mai multe articulații prin folosirea repetată a algoritmului, astfel se pot controla și brațe robotice cu 7 articulații, care sunt prezente în industria aerospațială.

In-Won Park, et. al. [46] poate controla un braț robotic cu 7 articulații în programul de simulare și unul de 4 articulații în experimentul real, folosind metode clasice cu matrice Jacobiană, însă utilizează calibrare și ajustări cu un sistem de laseri montați pe efectorul final și o cameră video, care urmărește cele două puncte de laser proiectate pe un perete. Acest lucru s-a introdus, deoarece sistemul clasic cu matrice Jacobiană are neajunsuri, uneori poziția calculată de acest algoritm nu este corectă și în anumite puncte din spațiu brațul robotic nu poate ajunge [46]. Pentru aceasta este util un sistem video pentru supraveghere. Abordarea este puțin ieșită din comun, deoarece camera este în spate și urmărește punctul laser proiectat pe un perete de la efectorul final, astfel nu filmează brațul robotic în sine, sau obiectul manipulat.

În fig. 5.19 este un grafic de probabilitate normală, la care se observă eroarea medie la experimentul propus, în cadrul acestei teze de doctorat, acesta este de -0,7667 mm, care în modul este mai mic, decât ca cea mai mică medie, de pe Tabelul 5.7 din experimentul din articolul realizat de In-Won Park, et. al. [46], care este 0,85 mm. Deviația standard însă, în experimentul prezentat este 3,38 mm, iar în experimentul din articolul realizat de In-Won Park, et. al. [46], este 0,37 mm minim. Se poate spune că ambele metode sunt destul de bune, fiecare punctează la un anumit parametru. Pe lângă asta mai trebuie menționat că în experimentul din articolul realizat de In-Won Park, et. al. [46] sunt folosite distanțe foarte mici de ordinul milimetrilor, iar în această teză sunt folosite distanțe de ordinul metrilor, iar în algoritmul prezentat în cadrul acestei teze de doctorat precizia crește o dată cu reducerea distanței, astfel în anumite părți s-ar obține valori comparabile cu experimentul din articolul realizat de In-Won Park, et. al. [46].

Se poate vedea că și în articolul realizat de In-Won Park, et. al. [46] sunt minim 25 de pași de calibrare, care în sistemul prezentat în cadrul acestei teze, deoarece calibrarea se face în timp real, în timpul execuției. Cu alte cuvinte, în cel mai rău caz, poziția este recalculată și brațul robotic este mișcat mai precis în poziția țintă. Într-un caz defavorabil, brațul robotic la prima încercare ajunge foarte aproape de poziția țintă, dar nu exact, astfel va avea toleranțe de ordinul milimetrilor. În acest moment, camerele video sesizează acest lucru și vor da un impuls, de recalculare a poziției. Din acest moment poziția țintă al brațului robotic se va recalcula din noua poziție, la această distanță mai mică eroarea va fi corectată.. Din încercări s-a văzut că poziția se recalculează de maxim trei ori, însă de cele mai multe ori brațul robotic reușește să ajungă exact în poziția țintă din prima încercare.

Tabel 5.6 Erorile de distanță pentru cele două spoturi de laser de pe perete după procesul de calibrare (experiment), preluat după [46]

		Calibration Set (mm)							Testing Set (mm)						
		l_1				l_2			l_1				l_2		
		Poses	Mean	STD	Max	Mean	STD	Max	Poses	Mean	STD	Max	Mean	STD	Max
EKF	Case 1	25	0.85	0.37	1.52	0.87	0.64	2.42	25	2.62	2.96	12.04	6.19	4.60	16.26
	Case 2	25	1.44	0.81	3.42	2.85	2.27	8.95	25	1.23	0.45	2.22	2.11	1.92	8.67
	Case 3	50	1.10	0.61	2.89	1.95	1.89	9.57	-	-	-	-	-	-	-

În articolul realizat de M. Seelinger, et. al. [38] se poate vedea o abordare interesantă și destul de rar întâlnită, deoarece se folosesc trei camere video, care controlează un braț robotic cu 6 articulații. O cameră este plasată în față și două în lateral. Este interesantă abordarea, deoarece în articol se urmăresc doar mișcările brațului robotic, dar acestea nu sunt controlate efectiv de camere, controlul fiind făcut de cinematica inversă și directă, adică este făcut tot de metode clasice.

L. Behera, et. al. [43] au o abordare apropiată de ceea ce prezentată în această lucrare. Sunt folosite două camere video și este controlat un braț robotic cu 3 articulații. În această abordare, de asemenea, sunt folosite metode clasice ca și cinematica inversă și directă, însă camerele sunt folosite parțial și pentru controlul brațului robotic. În [43] camerele sunt plasate una în spate și una într-o parte a brațului robotic.

Sistemul de camere stereo nu este realizat, deoarece camerele nu sunt pe aceeași parte, însă totuși controlul este realizat folosind un alt algoritm, care completează metodele clasice ca și cinematica inversă și directă. Algoritmul nu este nici în această abordare de sine stătător, însă este un sistem care controlează brațul robotic folosind camere video, iar apoi mai înglobează și metodele clasice. În fig. 5.19 se poate vedea eroarea medie de la experimentul prezentat în teză, care este - 0,7667 mm, care este mai mică decât eroarea din articolul realizat de L. Behera, et. al. [43], (1,18 mm, în cel mai bun caz, tabelul 5.7). Tot aici se poate vedea că durata de calibrarea minimă este 10 minute, care este o perioadă destul de lungă.

Se mai poate observa că pentru a învăța brațul robotic în articolul realizat de L. Behera, et. al. [43], este necesar un număr minim de 750 de pași de învățare, destul de mare, să nu mai vorbim că această operație durează 5 ore.

Tabel 5.7 Performanțele la diferiți algoritmi, preluat după [43]

Table 1. Performance of Different Algorithms			
<i>Scheme</i>	<i>Average Error</i>	<i>Traning Steps</i>	<i>Time</i>
Model-based approach	1,67 mm	Not applicable	10 min
Model free scheme	1,18 mm	750	5 hours
Hybrid approach	1,21 mm	2550	30 min

În articolul realizat de R. Kelly, et. al. [39] se observă o abordare puțin diferită, deoarece se folosește doar o singură cameră, care este montată pe efectorul final al brațului robotic, iar brațul robotic are doar două articulații. Pentru această mișcare nici nu este nevoie de două camere, deoarece cu două articulații brațul robotic se poate mișca doar în plan, iar pentru acesta o cameră este suficientă. Abordarea este destul de diferită, deoarece camera filmează obiectul manipulat și nu brațul robotic în sine. Abordarea însă prezintă avantaje, deoarece este tratată dinamica brațului robotic, neluată în considerare în această teză. Până acum dinamica nu a influențat sistemul, deoarece pentru plasarea obiectelor relativ ușoare dintr-o poziție în alta, unde nici viteza de deplasare nu contează foarte mult, nu există probleme. Pentru plasare de obiecte grele cu o viteză ridicată, tratarea dinamicii unui braț robotic este un aspect de interes major.

5.3. Concluzii și contribuții

În acest capitol au fost prezentate graficele pentru evaluarea sistemului de măsură a distanței implementat.

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

Dacă se efectuează măsurări cu două camere, și pe axa OX, 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.

Contribuții

1. **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.
 - 1.1. Analiza șase sigma (6σ) a fost completată de grafice Minitab și Excel.
 - 1.2. A fost analizată influența condițiilor de iluminare, a temperaturii de culoare și a modulului distanței până la obiectul țintă.

6. CONCLUZII FINALE ȘI CONTRIBUȚII

6.1. 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. Dacă sunt mai multe puncte țintă de culoare verde, se va considera cea mai mare pată de culoare, restul va fi eliminat, dar se poate implementa ca brațul robotic să ajungă la două puncte țintă, ordinea fiind realizată în funcție de mărimea acestora.

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:

- a. 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.
- b. 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.
- c. 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.
- d. Cel mai bun compromis este Python pe sistemul de operare Linux, care este relativ simplu și nici din resurse nu ocupă prea mult.
- e. 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).

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

Măsurarea în sistemul 2D (axele 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.

6.2. 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 ajutătoare 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 capacitate, 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 modulului distanței până la obiectul țintă.

6.3. Direcții viitoare de cercetare

Una din direcțiile viitoare este îmbunătățirea preciziei (toleranțe mai mici de ± 1 mm). Această ajustare nu trebuie făcută din formule, ci necesită multe încercări și teste cu robotul real. Pentru a ajunge la o asemenea precizie, ar trebui folosit un robot de tip KUKA (O dorință viitoare personală este ca tot sistemul să fie portat pe un robot de tip KUKA).

Sistemul a fost implementat prima oară pe PC în Windows, apoi pe PC în Linux, apoi a fost portat pe un sistem dedicat cu microprocesor (Raspberry PI). Ultima implementare a fost portarea sistemului pe FPGA. S-a obținut un sistem portabil, reconfigurabil și flexibil. Una din direcțiile viitoare este de a realiza un ASIC cu acest sistem.

O altă direcție este utilizarea întregului spectru util, respectiv refacerea experimentelor cu camere NIR (Near Infrared), o abordare diferită față de cea prezentă.

6.4. Lista publicațiilor personale

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, "Programmable Interface for a Signal Generator and Logic Analyzer System," SIITME 2013 – 19th International Symposium for Design and Technology in Electronic Packaging, Galați, România, 24-27 octombrie, 2013, pp. 231-235.
- [8] **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.
- [9] **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.
- [10] **R. Szabó**, A. Gontean, Lie I., "Embedded Temperature Monitoring System with a Microcontroller Used in the Automotive Industry," TELFOR 2012 – 20th Telecommunications Forum, Belgrad, Serbia, 20-22 noiembrie, 2012, pp. 1012-1015.

Citat de:

- [10.1] A.F.A. El-Gawad, "Suggested Temperature Monitoring System for Distribution Transformers by Using Microcontroller Scheme," Journal of Electrical Engineering Technologies, 2015, pp. 709-714.

- [11] **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.
- [12] **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.
- Citat de:**
- [12.1] 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, **indexat BDI**.
- [12.2] 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.
- [12.3] 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.
- [12.4] 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.
- [12.5] V. Pereira, V. A. Fernandes, J. Sequeira, "Low cost object sorting robotic arm using Raspberry Pi," GHTC-SAS 2014 – Global Humanitarian Technology Conference – South Asia Satellite, Trivandrum, India, 26-27 septembrie, 2014, pp. 1-6, **indexat BDI**.
- [12.6] 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.
- [12.7] 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.
- [12.8] 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.
- [12.9] 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.
- [12.10] 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
- [12.11] 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.
- [12.12] 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.
- [13] **R. Szabó**, A. Gontean, Lie I., "Acquisition Software Design for Temperature Sensors," TSP 2012 – 35th International Conference on Telecommunications and Signal Processing, Praga, Cehia, 3-4 iulie, 2012, pp. 272-276.
- [14] **R. Szabó**, A. Gontean, Lie I., "Sound Based Coin Recognition and Clapper," MENDEL 2010 – 16th International Conference on Soft Computing, Brno, Cehia, 23-25 iunie, 2010, pp. 509-516.

Conferințe indexate BDI

- [15] **R. Szabó**, A. Gontean, "Full 3D Robotic Arm Control with Stereo Cameras Made in LabVIEW," FedCSIS 2013 – Federated Conference on Computer Science and Information Systems, Cracovia, Polonia, 8-11 septembrie, 2013, pp. 37-42.
Citat de:
[15.1] S. Kumar, S. R. Lee, "A Simple and Efficient Wireless Control Protocol for Small Scale Robotic Arm," International Journal of Future Generation Communication and Networking, vol. 7, nr. 5, 2014, pp. 151-162.
[15.2] M. G. Papoutsidakis, D. D. Piromalis, G. E. Chamilothoris, "Modern control interface for SCORBOT ER-III robot," ISFEE 2014 – International Symposium on Fundamentals of Electrical Engineering, București, România, 28-29 noiembrie 2014, pp. 1-5, **indexat BDI.**
- [16] **R. Szabó**, A. Gontean, I. Lie, M. Băbăiță, "Comparison between Agilent and National Instruments Functional Test Systems," SISO 2010 – 8th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, 10-11 septembrie, 2010, pp. 87-92.
Citat de:
[16.1] M. Serban, Y. Vagapov, Z. Chen, R. Holme, S. Lupin, "Universal platform for PCB functional testing," APEDE 2014 – International Conference on Actual Problems of Electron Devices Engineering, vol. 2, 25-26 septembrie 2014, pp. 402-409, **indexat BDI.**
- [17] A. Gontean, **R. Szabó**, I. Lie, "LabVIEW Powered Remote Lab," SIITME 2009 – 15th International Symposium for Design and Technology of Electronics Packages, Gyula, Ungaria, 17-20 septembrie, 2009, pp 335-340.
Citat de:
[17.1] R. Costa, "An IEEE1451.0-compliant FPGA-based reconfigurable weblab," teză de doctorat, Coimbra University.
[17.2] G. Sharafutdinova, M. Cvetanovski, D. Walton, B. Guest, J. Foster, G. Piszczuk, J. Furst, "Distance learning laboratory: A remote Atomic and Nuclear Physics experiment," Australian Physics, vol. 50, nr. 2, martie 2013, pp. 55-59.
[17.3] G. Ramachandran, A. Vijayasai, G. Ramirez, T. Dallas, "Development and Testing of a Remote Access MEMS Lab for Distance Education," Innovations, chapter 15, 2012, pp. 171-179.
[17.4] G. Ramachandran, "Remote experimentation testing and characterization of micro electro mechanical systems," august 2012, lucrare de disertație, Texas Tech University.

Carte

- [18] A. Gontean, **R. Szabó**, "LabVIEW Remote Lab," Labview - Modeling, Programming and Simulations, InTech, chapter 13, 2011, ISBN: 978-953-307-521-1, pp. 275-306.
Citat de:
[18.1] J. Pechousek, "Application of virtual instrumentation in nuclear physics experiments," Practical Applications and Solutions Using Labview™ Software, chapter 4, InTech 2011, ISBN: 978-953-307-650-8, pp. 57-80.

Revistă

- [19] **R. Szabó**, A. Gontean, I. Lie, M. Băbăiță, "Oscilloscope Control with PC," International Journal of Computers and Communications," vol. 3, nr. 3, 2009, pp. 33-40.
Citată de:
[19.1] B. D. Hatwar, A. C. Wani, "Wireless Oscilloscope Powered By Android," International Journal of Emerging Technology and Advanced Engineering, vol. 4, nr. 7, iulie 2014, pp. 573-577.

BIBLIOGRAFIE

Lista publicațiilor 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, Varşovia, 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," GHTC-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.

Conferințe indexate BDI

- [11] **R. Szabó**, A. Gontean, "Full 3D Robotic Arm Control with Stereo Cameras Made in LabVIEW," FedCSIS 2013 – Federated Conference on Computer Science and Information Systems, Cracovia, Polonia, 8-11 septembrie, 2013, pp. 37-42.

Citată de:

- [11.1] S. Kumar, S. R. Lee, "A Simple and Efficient Wireless Control Protocol for Small Scale Robotic Arm," International Journal of Future Generation Communication and Networking, vol. 7, nr. 5, 2014, pp. 151-162.
- [11.2] M. G. Papoutsidakis, D. D. Piromalis, G. E. Chamilothis, "Modern control interface for SCORBOT ER-III robot," ISFEE 2014 – International Symposium on Fundamentals of Electrical Engineering, București, România, 28-29 noiembrie 2014, pp. 1-5, **indexată BDI**.
- [12] **R. Szabó**, A. Gontean, I. Lie, M. Băbăiță, "Comparison between Agilent and National Instruments Functional Test Systems," SISY 2010 – 8th

International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, 10-11 septembrie, 2010, pp. 87-92.

Citată de:

- [12.1] M. Serban, Y. Vagapov, Z. Chen, R. Holme, S. Lupin, "Universal platform for PCB functional testing," APEDE 2014 – International Conference on Actual Problems of Electron Devices Engineering, vol. 2, 25-26 septembrie 2014, pp. 402-409, **indexată BDI**.
- [13] A. Gontean, **R. Szabó**, I. Lie, "LabVIEW Powered Remote Lab," SIITME 2009 – 15th International Symposium for Design and Technology of Electronics Packages, Gyula, Ungaria, 17-20 septembrie, 2009, pp 335-340.
- Citată de:**
- [13.1] R. Costa, "An IEEE1451.0-compliant FPGA-based reconfigurable weblab," ianuarie 2014, teză de doctorat, Coimbra University, Portugalia.
- [13.2] G. Sharafutdinova, M. Cvetanovski, D. Walton, B. Guest, J. Foster, G. Piszczuk, J. Furst, "Distance learning laboratory: A remote Atomic and Nuclear Physics experiment," Australian Physics, vol. 50, nr. 2, martie 2013, pp. 55-59.
- [13.3] G. Ramachandran, A. Vijayasai, G. Ramirez, T. Dallas, "Development and Testing of a Remote Access MEMS Lab for Distance Education," Innovations, capitolul 15, S.U.A., 2012, pp. 171-179.
- [13.4] G. Ramachandran, "Remote experimentation testing and characterization of micro electro mechanical systems," august 2012, lucrare de disertație, Texas Tech University, S.U.A.

Carte

- [14] A. Gontean, **R. Szabó**, "LabVIEW Remote Lab," Labview - Modeling, Programming and Simulations, InTech, capitolul 13, Croația, 2011, ISBN: 978-953-307-521-1, pp. 275-306.
- Citată de:**
- [14.1] J. Pechousek, "Application of virtual instrumentation in nuclear physics experiments," Practical Applications and Solutions Using Labview™ Software, InTech, capitolul 4, Croația, 2011, ISBN: 978-953-307-650-8, pp. 57-80.

Referințe bibliografice de la alți autori

- [15] Wong Guan Hao, Yap Yee Leck, Lim Chot Hun, "6-DOF PC-Based Robotic Arm (PC-ROBOARM) with efficient trajectory planning and speed control," ICOM 2011 – 4th International Conference On Mechatronics, Kuala Lumpur, Malaysia, 17-19 mai, 2011, pp. 1-7.
- [16] Woosung Yang, Ji-Hun Bae, Yonghwan Oh, Nak Young Chong, Bum-Jae You, Sang-Rok Oh, "CPG based self-adapting multi-DOF robotic arm control," IROS 2010 – IEEE/RSJ International Conference on International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18-22 octombrie, 2010, pp. 4236-4243.
- [17] E. Oyama, T. Maeda, J. Q. Gan, E. M. Rosales, K. F. MacDorman, S. Tachi, A. Agah, "Inverse kinematics learning for robotic arms with fewer degrees of freedom by modular neural network systems," IROS 2005 – IEEE/RSJ International Conference on International Conference on Intelligent Robots and Systems, 2-6 august, 2005, pp. 1791-1798.
- [18] N. Ahuja, U. S. Banerjee, V. A. Darbhe, T. N. Mapara, A. D. Matkar, R.K. Nirmal, S. Balagopalan, "Computer controlled robotic arm," 16th IEEE Symposium on Computer-Based Medical Systems, New York, NY, SUA, 26-27 iunie, 2003, pp. 361-366.

- [19] M. H. Liyanage, N. Krouglicof, R. Gosine, "Design and control of a high performance SCARA type robotic arm with rotary hydraulic actuators," CCECE'09 – Canadian Conference on Electrical and Computer Engineering, St. John's, NL, SUA, 3-6 mai, 2009, pp. 827-832.
- [20] M. Mariappan, T. Ganesan, M. Iftikhar, V. Ramu, B. Khoo, "A design methodology of a flexible robotic arm vision system for OTOROB," ICMET 2010 – 2nd International Conference on Mechanical and Electrical Technology, Singapore, 10-12 septembrie, 2010, pp. 161-164.
- [21] Guo-Shing Huang, Xi-Sheng Chen, Chung-Liang Chang, "Development of dual robotic arm system based on binocular vision," CACS 2013 – International Automatic Control Conference, Nantou, Taiwan, 2-4 decembrie, 2013, pp. 97-102.
- [22] N. C. Orger, T. B. Karyot, "A symmetrical robotic arm design approach with stereo-vision ability for CubeSats," RAST 2013 – 6th International Conference on Recent Advances in Space Technologies, Istanbul, Turcia, 12-14 iunie, 2013, pp. 961-965.
- [23] F. Medina, B. Nono, H. Banda, A. Rosales, "Classification of Solid Objects with Defined Shapes Using Stereoscopic Vision and a Robotic Arm," ANDESCON 2012 – Andean Region International Conference, Cuenca, Spania, 2012, pp. 226.
- [24] M. Puheim, M. Bundzel, L. Madarasz, "Forward control of robotic arm using the information from stereo-vision tracking system," CINTI 2013 – 14th International Symposium on Computational Intelligence and Informatics, Budapesta, Ungaria, 19-21 noiembrie, 2013, pp. 57-62.
- [25] T. P. Cabre, M. T. Cairol, D. F. Calafell, M. T. Ribes, J. P. Roca, "Project-Based Learning Example: Controlling an Educational Robotic Arm With Computer Vision," IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, vol. 8, nr. 3, 2013, pp. 135-142.
- [26] G. S. Gupta, S. C. Mukhopadhyay, M. Finnie, "WiFi-based control of a robotic arm with remote vision," I2MTC'09 – IEEE Instrumentation and Measurement Technology Conference, Singapore, 5-7 mai, 2009, pp. 557-562.
- [27] Haoting Liu, Wei Wang, Feng Gao, Zhaoyang Liu, Yuan Sun, Zhenlin Liu, "Development of Space Photographic Robotic Arm based on binocular vision servo," ICACI 2013 – Sixth International Conference on Advanced Computational Intelligence, Hangzhou, China, 19-21 octombrie, 2013, pp. 345-349.
- [28] 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.
- [29] 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.
- [30] T. Kizaki, A. Namiki, "Two ball juggling with high-speed hand-arm and high-speed vision system," ICRA 2012 – IEEE International Conference on Robotics and Automation, Saint Paul, MN, SUA, 14-18 mai, 2012, pp. 1372-1377.

- [31] Manual de utilizare SCORBOT-ER III, <http://www.theoldrobots.com/book45/ER3-Manual.pdf>, vizitat ultima oară 20 octombrie, 2014.
- [32] Calcul distanță stereo, <http://www.alexandria.nu/ai/blog/entry.asp?E=32>, vizitat ultima oară 10 decembrie, 2014.
- [33] Stereo triangulare, <http://www.dis.uniroma1.it/~iocchi/stereo/triang.html>, vizitat ultima oară 9 decembrie, 2014.
- [34] 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.
- [35] Calcul PPI, <https://www.sven.de/dpi/>, vizitat ultima oară 20 februarie, 2015.
- [36] Model 3D braț robotic, <http://www.3dcadbrowser.com/download.aspx?3dmodel=17132>, vizitat ultima oară 30 septembrie, 2015.
- [37] Desen braț robotic, <https://www.cdli.ca/courses/ep/ds2/t06/02knowledge-skills/act-01a.htm>, vizitat ultima oară 30 septembrie, 2015.
- [38] 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.
- [39] 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.
- [40] 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.
- [41] 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.
- [42] 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.
- [43] L. Behera, N. Kirubanandan, "A hybrid neural control scheme for visual-motor coordination", *IEEE Control Systems*, vol. 19, nr. 4, 1999, pp. 34-41.
- [44] J. J. Huring, D. W. Murray, "Modeling and copying human head movements", *IEEE Transactions on Robotics and Automation*, vol. 15, nr. 6, 1999, pp. 1095-1108.
- [45] F. Chaumette, S. Hutchinson, "Visual servo control. II. Advanced approaches [Tutorial]", *IEEE Robotics & Automation Magazine*, vol. 14, nr. 1, 2007, pp. 109-118.
- [46] 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.
- [47] Zhang Hanqi, "Hand/eye calibration for electronic assembly robots," *IEEE Transactions on Robotics and Automation*, vol. 14, nr. 4, 1998, pp. 612-616.
- [48] 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.

- [49] R. Kelly, "Robust asymptotically stable visual servoing of planar robots," *IEEE Transactions on Robotics and Automation*, vol. 12, nr. 5, 1996, pp. 759-766.
- [50] 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.
- [51] 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.
- [52] 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.
- [53] G. Silveira, "On Intensity-Based Nonmetric Visual Servoing," *IEEE Transactions on Robotics*, vol. 30, nr. 4, 2014, pp. 1019-1026.
- [54] 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.
- [55] 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.
- [56] 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.
- [57] 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.
- [58] Z. Pezzementi, E. Plaku, C. Reyda, G. D. Hager, "Tactile-Object Recognition From Appearance Information," vol. 27, nr. 3, 2011, pp. 473-487.
- [59] Yiu Cheung Shiu, S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $AX=XB$," *IEEE Transactions on Robotics and Automation*, vol. 5, nr. 1, 1989, pp. 16-29.
- [60] H. Hadj-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.
- [61] R. F. Wolffenbuttel, K. M. Mahmoud, Paul P. L. Regtien, "Compliant capacitive wrist sensor for use in industrial robots," *IEEE Transactions on Instrumentation and Measurement*, vol. 39, nr. 6, 1990, pp. 991-997.
- [62] Xiao Shunli, Li Yangmin, "Visual Servo Feedback Control of a Novel Large Working Range Micro Manipulation System for Microassembly," *Journal of Microelectromechanical Systems*, vol. 23, nr. 1, 2014, pp. 181-190.
- [63] Vladimir J. Lumelsky, E. Cheung, "Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, nr. 1, 1993, pp. 194-203.
- [64] Usama Iqbal, Abdul Samad, Zainab Nissa, Jamshed Iqbal, "Embedded Control System for Autarep - A Novel Autonomous Articulated Robotic Educational Platform," *Tehnicki Vjesnik-Technical Gazette*, vol. 21, nr. 6, 2014, pp. 1255-1261.

- [65] G. Silveira, "On intensity-based 3D visual servoing," *Robotics and Autonomous Systems*, vol. 62, nr. 11, ediție specială, 2014, pp. 1636-1645.
- [66] Jason M. Godlove, Erin O. Whaite, Aaron P. Batista, "Comparing temporal aspects of visual, tactile, and microstimulation feedback for motor control," *Journal of Neural Engineering*, vol. 11, nr. 4, 2014, nr. articol 046025.
- [67] D. Font, T. Palleja, M. Tresanchez, D. Runcan, J. Moreno, D. Martinez, M. Teixido, J. Palacin, "A Proposal for Automatic Fruit Harvesting by Combining a Low Cost Stereovision Camera and a Robotic Arm," *Sensors*, vol. 14, nr. 7, 2014, pp. 11557-11579.
- [68] S. Manzoor, R. U. Islam, A. Khalid, A. Samad, J. Iqbal, "An open-source multi-DOF articulated robotic educational platform for autonomous object manipulation," *Robotics and Computer-Integrated Manufacturing*, vol. 30, nr. 3, 2014, pp. 351-362.
- [69] Luka Lukic, Jose Santos-Victor, Aude Billard, "Learning robotic eye-arm-hand coordination from human demonstration: a coupled dynamical systems approach," *Biological Cybernetics*, vol. 108, nr. 2, 2014, pp. 223-248.
- [70] Amira Y. Haikal, Mostafa A. El-Hosseini, "A Smart Robot Arm Design for Industrial Application," *Studies in Informatics and Control*, vol. 23, nr. 1, 2014, pp. 107-116.
- [71] M. Z. Abdullah, M. S. Awang, Y. C. Tan, J. M. Abdullah, "A Preliminary Study of Using Active Vision Guided Robotic Arm for Bone Drilling in a Developing Country," *Journal of Neurological Surgery Part A-Central European Neurosurgery*, vol. 75, nr. 2, 2014, pp. 155-157.
- [72] J. D. McNulty, T. Klann, J. Sha, M. Salick, G. T. Knight, L. S. Turng, R. S. Ashton, "High-precision robotic microcontact printing (R-mu CP) utilizing a vision guided selectively compliant articulated robotic arm," *Lab on a Chip*, vol. 14, nr. 11, 2014, pp. 1923-1930.
- [73] C. Y. Lin, Y. P. Chiu, C. Y. Lin, C. R. Tsai, "Development of a binocular vision-based catcher robot system using DSP platform," *Journal of the Chinese Institute of Engineers*, vol. 37, nr. 2, 2014, pp. 210-223.
- [74] F. Perez-Pena, A. Morgado-Estevez, A. Linares-Barranco, A. Jimenez-Fernandez, F. Gomez-Rodriguez, G. Jimenez-Moreno, J. Lopez-Coronado, "Neuro-Inspired Spike-Based Motion: From Dynamic Vision Sensor to Robot Motor Open-Loop Control through Spike-VITE," *Sensors*, vol. 13, nr. 11, 2013, pp. 15805-15832.
- [75] Kuei-Shu Hsu, Limei Peng, Chen Yu, "Design and application of the stereo vision manipulator with novel scheduling policies control," *Multimedia Tools and Applications*, vol. 67, nr. 1, 2013, pp. 249-268.
- [76] Andry Maykol Pinto, Luis F. Rocha, A. Paulo Moreira, "Object recognition using laser range finder and machine learning techniques," *Robotics and Computer-Integrated Manufacturing*, vol. 29, nr. 1, 2013, pp. 12-22.
- [77] Chia-Hung Chen, Han-Pang Huang, "Pose estimation for autonomous grasping with a robotic arm system," *Journal of the Chinese Institute of Engineers*, vol. 36, nr. 5, 2013, pp. 638-646.
- [78] Tobi Delbruck, Manuel Lang, "Robotic goalie with 3 ms reaction time at 4% CPU load using event-based dynamic vision sensor," *Frontiers in Neuroscience*, vol. 7, 2013, nr. articol 223.
- [79] Joseph Coombs, Rahul Prabhu, Greg Peake, "Overcoming the challenges of porting OpenCV to TI's embedded ARM plus DSP platforms," *International Journal of Electrical Engineering Education*, vol. 49, nr. 2, ediție specială. 2012, pp. 260-274.

- [80] L. Minati, A. Nigri, C. Rosazza, M. G. Bruzzone, "Thoughts turned into high-level commands: Proof-of-concept study of a vision-guided robot arm driven by functional MRI (fMRI) signals," *Medical Engineering & Physics*, vol. 34, nr. 5, 2012, pp. 650-658.
- [81] L. Cuvillon, E. Laroche, J. Gangloff, M. de Mathelin, "A Multivariable Methodology for Fast Visual Servoing of Flexible Manipulators Moving in a Restricted Workspace," *Advanced Robotics*, vol. 26, nr. 15, 2012, pp. 1771-1797.
- [82] Han-Pang Uang, Chia-Hung Chen, Kuang-Chih Liu, "Stereo-Based 3D Localization for Grasping known Objects with a Robotic Arm System," *Journal of the Chinese Society of Mechanical Engineers*, vol. 32, nr. 4, 2011, pp. 323-330.
- [83] 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.
- [84] K. M. Tsui, D. J. Kim, A. Behal, D. Kontak, H. A. Yanco, "«I want that:» Human-in-the-loop control of a wheelchair-mounted robotic arm," *Applied Bionics and Biomechanics*, vol. 8, nr. 1, 2011, pp. 127-147.
- [85] Ruhizan Liza Ahmad Shauri, Kenzo Nonami, "Assembly manipulation of small objects by dual-arm manipulator," *Assembly Automation*, vol. 31, nr. 3, 2011, pp. 263-274.
- [86] E. B. Brokaw, T. Murray, T. Nef, P. S. Lum, "Retraining of interjoint arm coordination after stroke using robot-assisted time-independent functional training," *Journal of Rehabilitation Research and Development*, vol. 48, nr. 4, 2011, pp. 299-316.
- [87] Jorge Pomares, Pablo Gil, Fernando Torres, "Visual Control of Robots Using Range Images," *Sensors*, vol. 10, nr. 8, 2010, pp. 7303-7322.
- [88] Sandra Nope, Humberto Loaiza, Eduardo Caicedo, "3D-2D Gesture reconstruction using monocular video information applied to a robotic arm," *Revista Facultad de Ingenieria-Universidad de Antioquia*, nr. 53, 2010, pp. 145-154.
- [89] P. Zacharia, N. Aspragathos, I. Mariolis, E. Dermatas, "A robotic system based on fuzzy visual servoing for handling flexible sheets lying on a table," *Industrial Robot-An International Journal*, vol. 36, nr. 5, 2009, pp. 489-496.
- [90] Chaiyapol Kulpate, Raman Paranjape, Mehran Mehrandezh, "Precise 3D Positioning of a Robotic Arm Using a Single Camera and a Flat Mirror," *International Journal of Optomechatronics*, vol. 2, nr. 3, 2008, pp. 205-232.
- [91] Anthony Remazeilles, Francois Chaumette, "Image-based robot navigation from an image memory," *Robotics and Autonomous Systems*, vol. 55, nr. 4, 2007, pp. 345-356.
- [92] Dadi Gudmundsson, Ken Goldberg, "Optimizing robotic part feeder throughput with queueing theory," *Assembly Automation*, vol. 27, nr. 2, 2007, pp. 134-140.
- [93] V. Lippiello, L. Villani, B. Siciliano, "An open architecture for sensory feedback control of a dual-arm industrial robotic cell," *Industrial Robot-An International Journal*, vol. 34, nr. 1, 2007, pp. 46-53.
- [94] Eduardo Bayro-Corrochano, Julio Zamora-Esquivel, "Differential and inverse kinematics of robot devices using conformal geometric algebra," *Robotica*, vol. 25, parte 1, 2007, pp. 43-61.

- [95] Andrew Perks, "Advanced vision guided robotics provide "future-proof" flexible automation," *Assembly Automation*, vol. 26, nr. 3, 2006, pp. 216-220.
- [96] E. Bayro-Corrochano, L. Reyes-Lozano, J. Zamora-Esquivel, "Conformal geometric algebra for robotic vision," *Journal of Mathematical Imaging and Vision*, vol. 24, nr. 1, 2006, pp. 55-81.
- [97] S. Okumura, N. Take, N. Okino, "Error prevention in robotic assembly tasks by a machine vision and statistical pattern recognition method," *International Journal of Production Research*, vol. 43, nr. 7, 2005, pp. 1397-1410.
- [98] E. Boivin, I. Sharf, "Optimum grasp planner and vision-guided grasping using a three-finger hand," *Industrial Robot-An International Journal*, vol. 32, nr. 1, 2005, pp. 35-42.
- [99] F. Boughorbel, Y. Zhang, S. K. Kang, U. Chidambaram, B. Abidi, A. Koschan, M. Abidi, "Laser ranging and video imaging for bin picking," *Assembly Automation*, vol. 23, nr. 1, 2003, pp. 53-59.
- [100] J. Blasco, N. Aleixos, J. M. Roger, G. Rabatel, E. Molto, "Robotic weed control using machine vision," *Biosystems Engineering*, vol. 83, nr. 2, 2002, pp. 149-157.
- [101] A. Nagchaudhuri, S. Kuruganty, A. Shakur, "Introduction of mechatronics concepts in a robotics course using an industrial SCARA robot equipped with a vision sensor," *Mechatronics*, vol. 12, nr. 2, 2002, pp. 183-193.
- [102] A. Chella, D. Guarino, I. Infantino, R. Pirrone, "A vision system for symbolic interpretation of dynamic scenes using ARSOM," *Applied Artificial Intelligence*, vol. 15, nr. 8, 2001, pp. 723-734.
- [103] Y. M. Chen, C. S. Hsueh, "Complementary data fusion in vision-guide and control of robotic tracking," *Robotica*, vol. 19, partea 1, 2001, pp. 53-58.
- [104] C. Jones, G. Ross, J. R. Hewit, A. P. Slade, "Robotic sorting of paper items from a random pile," *Mechatronics*, vol. 10, nr. 8, 2000, pp. 869-880.
- [105] D. K. Pai, J. C. Lang, J. Lloyd, R. J. Woodham, "ACME, a telerobotic Active Measurement Facility," *Experimental Robotics VI*, vol. 250, 2000, pp. 391-400.
- [106] W. K. Song, H. Lee, Z. Bien, "KARES: Intelligent wheelchair-mounted robotic arm system using vision and force sensor," *Robotics and Autonomous Systems*, vol. 28, nr. 1, 1999, pp. 83-94.
- [107] M. Seitz, "Towards autonomous robotic servicing: Using an integrated hand-arm-eye system for manipulating unknown objects," *Robotics and Autonomous Systems*, vol. 26, nr. 1, 1999, pp. 23-42.
- [108] J. L. Buessler, J. P. Urban, "Visually guided movements: learning with modular neural maps in robotics," *Neural Networks*, vol. 11, nr. 7-8, 1998, pp. 1395-1415.
- [109] J. Mulligan, "Fast calibrated stereo vision for manipulation," *Real-Time Imaging*, vol. 3, nr. 5, 1997, pp. 331-341.
- [110] S. Moon, L. E. Bernold, "Vision-based interactive path planning for robotic bridge paint removal," *Journal of Computing in Civil Engineering*, vol. 11, nr. 2, 1997, pp. 113-120.
- [111] R. Sargent, B. Bailey, C. Witty, A. Wright, "Dynamic object capture using fast vision tracking," *AI Magazine*, vol. 18, nr. 1, 1997, pp. 65-72.
- [112] R. Cipolla, N. Hollinghurst, "Visually guided grasping in unstructured environments," *Robotics and Autonomous Systems*, vol. 19, nr. 3-4, 1997, pp. 337-346.

- [113] P. A. Couvignou, N. P. Papanikolopoulos, M. Sullivan, P. K. Khosla, "The use of active deformable models in model-based robotic visual servoing," *Journal of Intelligent & Robotic Systems*, vol. 17, nr. 2, 1996, pp. 195-221.
- [114] Control braț robotic cu recunoaștere de imagine, <http://www.robidouille.com/>, vizitat ultima oară 1 aprilie, 2015.
- [115] T. J. Tarn, P. K. De, A. K. Bejczy, Z. Li, "Visualization of Dual-Arm Robot Motion Space Under Kinematic Constraints," *International Journal of Robotics Research*, vol. 14, nr. 1, 1995, pp. 9-18.
- [116] K. Tarabanis, R. Y. Tsai, D. S. Goodman, "Calibration of a Computer-Controlled Robotic Vision Sensor with a Zoom Lens," vol. 59, nr. 2, 1994, pp. 226-241.
- [117] W. I. Clement, K. A. Knowles, "An Instructional Robotics and Machine Vision Laboratory," *IEEE Transactions on Education*, vol. 37, nr. 1, 1994, pp. 87-90.
- [118] E. W. Fant, S. L. Tran, "Robotic Arm Self-Taught Path Planning Using Transportation Model Heuristics," *Robotica*, vol. 12, partea 1, 1994, pp. 17-21.
- [119] S. A. Schneider, R. H. Cannon, "Experimental Object-Level Strategic Control with Cooperating Manipulators," *International Journal of Robotics Research*, vol. 12, nr. 4, 1993, pp. 338-350.
- [120] E. W. Fant, S. L. Tran, "Robotic Arm Visual Servoing Using Light-Intensity in Contrast," *International Journal of Production Research*, vol. 31, nr. 4, 1993, pp. 947-956.
- [121] N. P. Papanikolopoulos, P. K. Khosla, T. Kanade, "Visual Tracking of a Moving Target by a Camera Mounted on a Robot - A Combination of Control and Vision," *IEEE Transactions on Robotics and Automation*, vol. 9, nr. 1, 1993, pp. 14-35.
- [122] S. A. Stansfield, "Robotic Grasping of Unknown Objects - A Knowledge-Based Approach," *International Journal of Robotics Research*, vol. 10, nr. 4, 1991, pp. 314-326.
- [123] W. Simonton, "Automatic Geranium Stock Processing in a Robotic Workcell," *Transactions of the ASAE*, vol. 33, nr. 6, 1990, pp. 2074-2080.
- [124] U. Meshram, P. Bande, P. A. Dwaramwar, R. R. Harkare, "Robot arm controller using FPGA," *IMPACT '09 - International Multimedia, Signal Processing and Communication Technologies*, Aligarh, India, 14-16 martie, 2009, pp. 8-11.
- [125] Jung Uk Cho, Quy Ngoc Le, Jae Wook Jeon, "An FPGA-Based Multiple-Axis Motion Control Chip," *IEEE Transactions on Industrial Electronics*, vol. 56, nr. 3, 2009, pp. 856-870.
- [126] J. J. Martinez-Alvarez, A. Guerrero-Gonzalez, J. L. Pedreno-Molina, A. Villaescusa-Fernandez, J. M. Ferrandez, J. López-Coronado, "Hardware implementation of a controller based on neurobiological adaptive models of the human motor-control system," *IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, 6-9 octombrie, 2002, pp. 3.
- [127] F. Perez-Pena, A. Morgado-Estevez, T. Serrano-Gotarredona, F. Gomez-Rodriguez, V. Ferrer-Garcia, A. Jimenez-Fernandez, A. Linares-Barranco, "Spike-based VITE control with dynamic vision sensor applied to an arm robot," *ISCAS 2014 - IEEE International Symposium on Circuits and Systems*, Melbourne, Australia, 1-5 iunie, 2014, pp. 463-466.
- [128] D. Petrick, A. Geist, D. Albaijes, M. Davis, P. Sparacino, G. Crum, R. Ripley, J. Boblitt, T. Flatley, "SpaceCube v2.0 space flight hybrid reconfigurable data

- processing system," IEEE Aerospace Conference, Big Sky, MT, SUA, 1-8 martie, 2014, pp. 1-20.
- [129] R. Marin, G. Leon, R. Wirz, J. Sales, J. M. Claver, P. J. Sanz, "Remote control within the UJI Robotics Manufacturing Cell using FPGA-based vision," ECC 2007 – European Control Conference, Kos, Greece, 2-5 iulie, 2007, pp. 1378-1383.
- [130] G. V. Persiano, S. Rapuano, F. Zoino, A. Morganella, G. Chiusolo, "Distance Learning in Digital Electronics: Laboratory Practice on FPGA," IMTC 2007 – IEEE Instrumentation and Measurement Technology Conference Proceedings, Varşovia, Polonia, 1-3 mai, 2007, pp. 1-6.
- [131] J. Sales, R. Beltran, P. J. Sanz, R. Marin, R. Wirz, G. Leon, J. Claver, J. Alemany, "The UJI industrial robotics telelaboratory: Real-time vision and networking," IROS 2008 – IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, Franța, 22-26 septembrie, 2008, pp. 4136.
- [132] F. Perez-Pena, A. Morgado-Estevez, C. Rioja-Del-Rio, A. Linares-Barranco, A. Jimenez-Fernandez, J. Lopez-Coronado, J. L. Munoz-Lozano, "Towards AER VITE: Building spike gate signal," ICECS 2012 – 19th IEEE International Conference on Electronics, Circuits and Systems, Seville, Spania, 9-12 decembrie, 2012, pp. 881-884.
- [133] P. Caldwell, R. Al-Bayat, C. Kellar, I.-S. Shin, "Biomechanics: Surface electromyography prosthesis control," BMEI 2012 – 5th International Conference on Biomedical Engineering and Informatics, Chongqing, China, 16-18 octombrie 2012, pp. 768-789.
- [134] M. H. Liyanage, N. Krouglicof, "A single time scale visual servoing system for a high speed SCARA type robotic arm," ICRA 2014 – IEEE International Conference on Robotics and Automation, Hong Kong, 31 mai-7 iunie, 2014, pp. 4153-4160.
- [135] K. Schmid, H. Hirschmuller, "Stereo vision and IMU based real-time ego-motion and depth image computation on a handheld device," ICRA 2013 – IEEE International Conference on Robotics and Automation, Karlsruhe, Germania, 6-10 mai, 2013, pp. 4671-4678.
- [136] R. Legrand, K. Machulis, D. P. Miller, R. Sargent, A. Wright, "The XBC: a modern low-cost mobile robot controller," IROS 2005 – IEEE/RSJ International Conference on Intelligent Robots and Systems, 2-6 august 2005, pp. 3896-3900.
- [137] D. Liu, L. Cong, Jingmeng Liu, D. Xu, "The research on the control system of the minimally invasive surgical robot," IEEE Conference on Robotics, Automation and Mechatronics, Chengdu, China, 21-24 septembrie, 2008, pp. 592-597.
- [138] Wei Wang, Chuncheng Wang, Yanling Hao, "A hardware design of navigation receiver signal processing platform," ROBIO 2008 – IEEE International Conference on Robotics and Biomimetics, Bangkok, Thailanda, 22-25 februarie 2008, pp. 2139-2143.
- [139] Guyue Zhou, Ang Liu, Kang Yang, Tao Wang, Zexiang Li, "An Embedded Solution to Visual Mapping for Consumer Drones," CVPRW 2014 – IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, SUA, 23-28 iunie 2014, pp. 670-675.
- [140] Jeong Seob Kim, Seul Jung, "Joint control of ROBOKER arm using a neural chip embedded on FPGA," ISIE 2009 – IEEE International Symposium on Industrial Electronics, Seoul, Coreea de Sud, 5-8 iulie, 2009, pp. 1007-1012.

- [141] Shih-An Li, Chung-Wei Weng, Yi-Hong Chen, Chia-Hung Lo, Min-Hao Yang, Yi-Chun Lin, Ming-Hua Hsieh, Ching-Chang Wong, "Servo motor controller design for robotic manipulator," *ISPACS 2012 – International Symposium on Intelligent Signal Processing and Communications Systems*, New Taipei, Taiwan, 4-7 noiembrie, 2012, pp. 254-257.
- [142] J. R. Guzman-Sepulveda, R. De Jesus Romero-Troncoso, "Digital System Control for Three-Degrees of Freedom Mechanical Arm with FPGA," *CERMA '08 – Electronics, Robotics and Automotive Mechanics Conference*, Morelos, Mexic, 30 septembrie-3 octombrie, 2008, pp. 496-501.
- [143] Jin Dang, Fenglei Ni, Yikun Gu, Minghe Jin, Hong Liu, "A highly integrated and flexible joint test system based on DSP/FPGA-FPGA," *ROBIO 2009 – IEEE International Conference on Robotics and Biomimetics*, Guilin, China, 19-23 decembrie, 2009, pp. 1877-1882.
- [144] Zheng Yili, Sun Hanxu, Jia Qingxuan, Shi Guozhen, "Kinematics control for a 6-DOF space manipulator based on ARM processor and FPGA Co-processor," *INDIN 2008 – 6th IEEE International Conference on Industrial Informatics*, Daejeon, Coreea de Sud, 13-16 iulie, 2008, pp. 129-134.
- [145] Min Xu, Wenzhang Zhu, Ying Zou, "Design of a Reconfigurable Robot Controller Based on FPGA," *SEC '08 – Fifth IEEE International Symposium on Embedded Computing*, Beijing, China, 6-8 octombrie, 2008, pp. 216-222.
- [146] V. Ramakrishnan, N. S. Gopal, R. Ashok, S. Moorthi, "FPGA based DC servo motor control for remote replication of movements of a surgical arm," *TENCON 2011 – IEEE Region 10 Conference*, Bali, Indonezia, 21-24 noiembrie, 2011, pp. 671-675.
- [147] R. O. Reynolds, P. H. Smith, L. S. Bell, H. U. Keller, "The design of Mars lander cameras for Mars Pathfinder, Mars Surveyor '98 and Mars Surveyor '01," *IEEE Transactions on Instrumentation and Measurement*, vol. 50, nr. 1, 2001, pp. 63-71.
- [148] Moksoon Jang, F. Dawson, G. Bailak, "Control system for multiple joint robotic arm powered by ultrasonic motor," *APEC '04 – Nineteenth Annual IEEE Applied Power Electronics Conference and Exposition*, vol. 3, 2004, pp. 1844-1848.
- [149] Woon Kyu Lee, Seul Jung, "FPGA Design for Controlling Humanoid Robot Arms by Exoskeleton Motion Capture System," *ROBIO '06 – IEEE International Conference on Robotics and Biomimetics*, Kunming, China, 17-20 decembrie, 2006, pp. 1378-1383.
- [150] F. L. Ni, M. H. Jin, H. L. Wang, H. Liu, G. Hirzinger, "Joint Fault-Tolerant Design of the Chinese Space Robotic Arm," *IEEE International Conference on Information Acquisition*, Weihai, China, 20-23 august, 2006, pp. 528-533.
- [151] Yiwei Liu, Zongwu Xie, Qi Zhang, Bin Wang, "HIT-ARM I high speed dexterous robot arm," *ROBIO 2012 – IEEE International Conference on Robotics and Biomimetics*, Guangzhou, China, 11-14 decembrie, 2012, pp. 26-29.
- [152] K. Chuengsatiansup, K. Sajjapongse, P. Kruapraditsiri, C. Chanma, N. Termthanasombat, Y. Suttasupa, S. Sattaratnamai, E. Pongkaew, P. Udsatid, B. Hattha, P. Wibulpolprasert, P. Usaphapanus, N. Tulyanon, M. Wongsaisuwan, W. Wannasuphprasit, P. Chongstitvatana, "Plasma-RX: Autonomous Rescue robots," *ROBIO 2008 – IEEE International Conference on Robotics and Biomimetics*, Bangkok, Thailanda, 22-25 februarie, 2009, pp. 1986-1990.

-
- [153] Chuangqiang Guo, Fenglei Ni, Jingting Sun, Yechao Liu, Hong Liu, "Joint driver and control design for large torque, long arm space remote manipulator," ICIA 2010 – IEEE International Conference on Information and Automation, Harbin, China, 20-23 iunie, 2010, pp. 1426-1431.
 - [154] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale, R. Siegwart, "A synchronized visual-inertial sensor system with FPGA pre-processing for accurate real-time SLAM," ICRA 2014 – IEEE International Conference on Robotics and Automation, Hong Kong, 31 mai-7 iunie, 2014, pp. 431-437.
 - [155] M. Petko, T. Uhl, "Embedded controller design-mechatronic approach," Proceedings of the Second International Workshop on Robot Motion and Control, Bukowy Dworek, Polonia, 18-20 octombrie, 2001, pp. 195-200.
 - [156] L. Zouari, M. Ben Ayed, M. Abid, "Embedded control of robot arm driven by Brushless DC motor on FPGA," WCCS 2014 – Second World Conference on Complex Systems, Agadir, Maroc, 10-12 noimebrie, 2014, pp. 722-727.

ANEXA A

În această secțiune sunt prezentate panourile frontale, diagramele bloc și subprogramele (SubVI) create în programul LabVIEW pentru programul de control al brațului robotic folosind camere video.

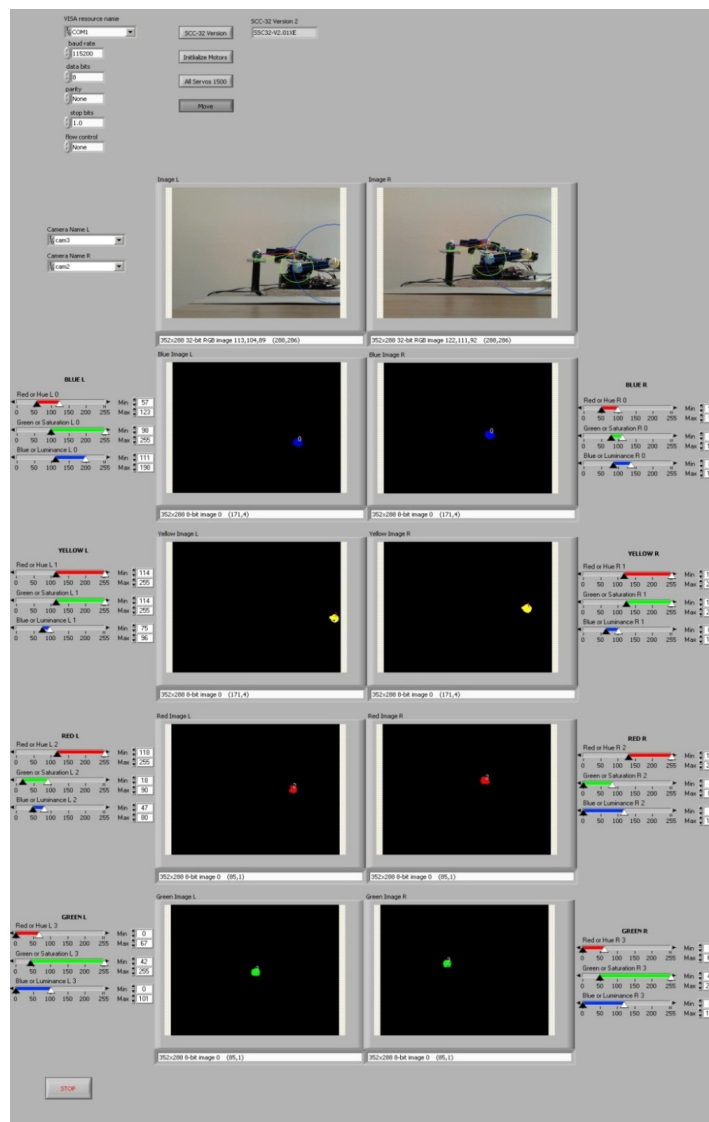


Fig. 4.28 Panoul frontal în LabVIEW pentru controlul brațului robotic în 3D

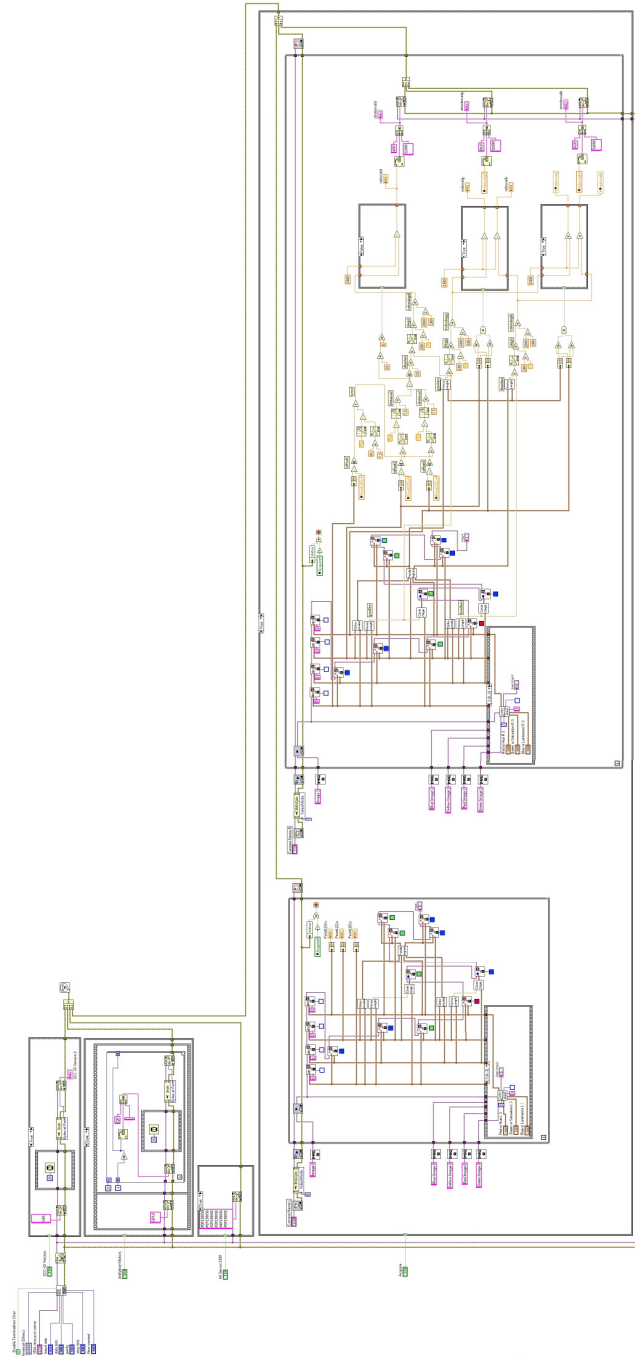


Fig. A.1 Diagrama bloc în LabVIEW pentru controlul brațului robotic în 3D

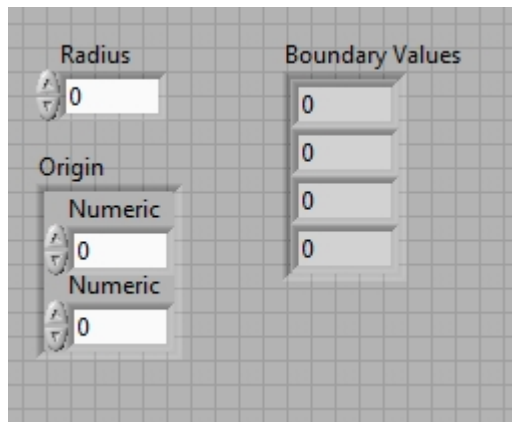


Fig. A.2 Panoul frontal pentru calculul originii cercului

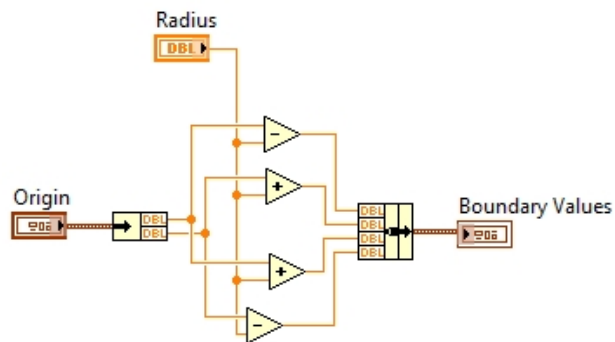


Fig. A.3 Diagrama bloc pentru calculul originii cercului

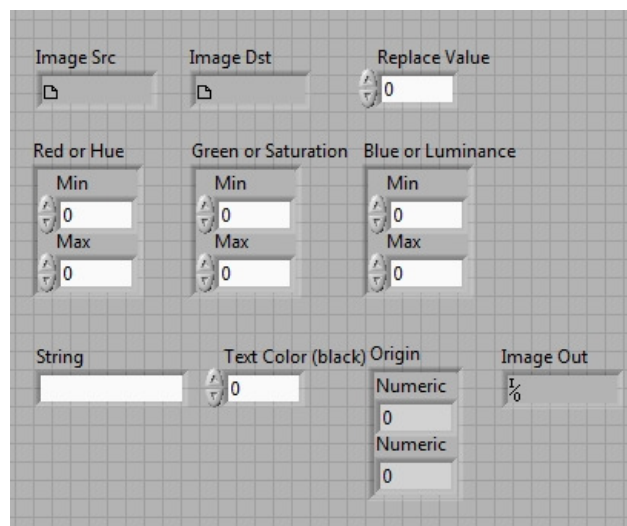


Fig. A.4 Panoul frontal pentru subprogramul de recunoaștere de culoare

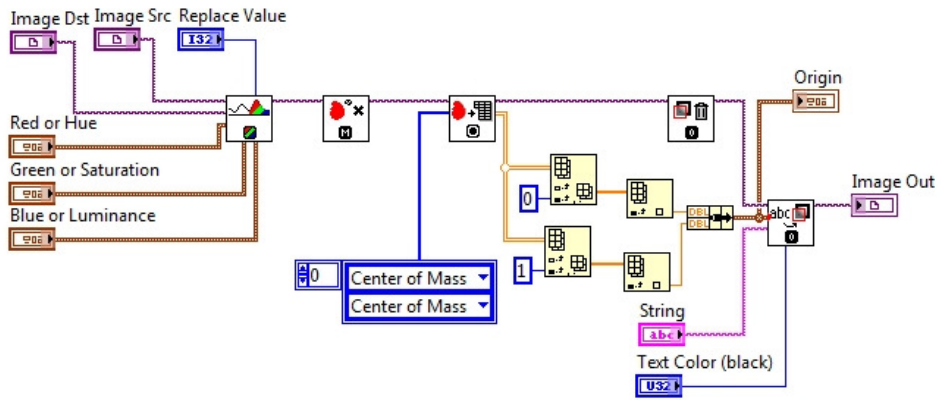


Fig. A.5 Diagrama bloc pentru subprogramul de recunoaștere de culoare

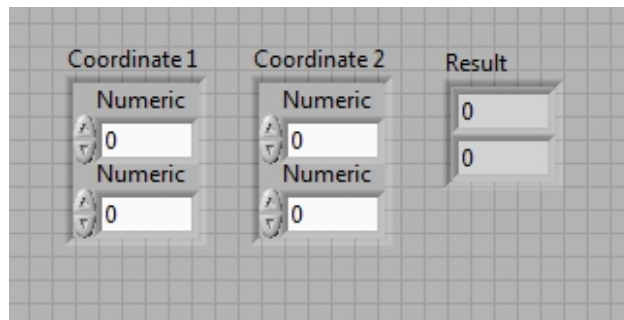


Fig. A.6 Panoul frontal pentru ortogonalizare

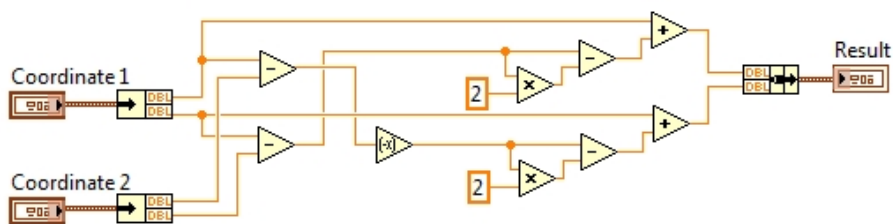


Fig. A.7 Diagrama bloc pentru ortogonalizare

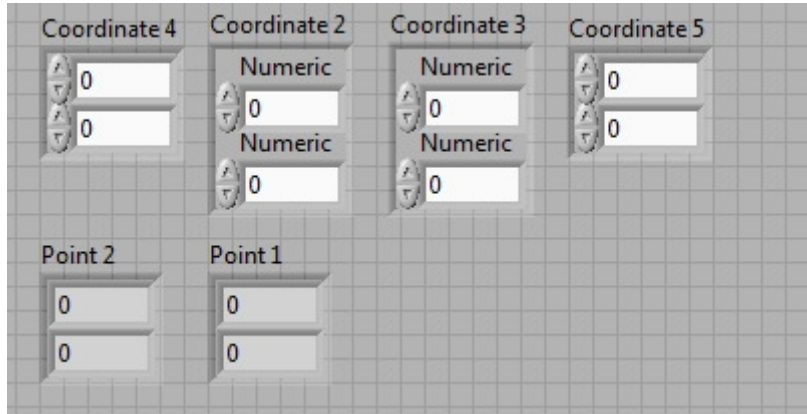


Fig. A.8 Panoul frontal pentru calculul de paralelogram

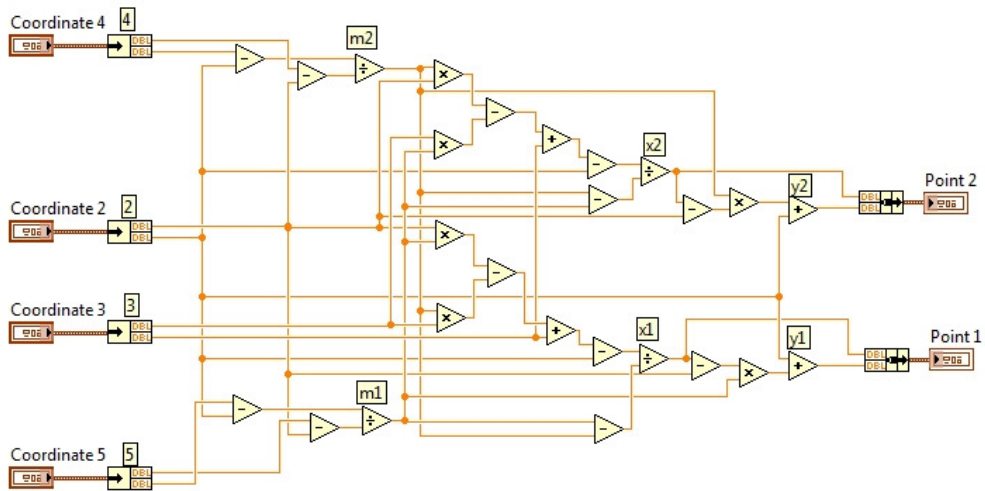


Fig. A.9 Diagrama bloc pentru calculul de paralelogram

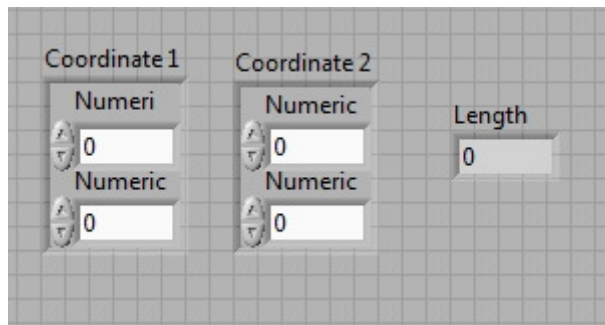


Fig. A.10 Panoul frontal pentru calculul lungimii unui vector

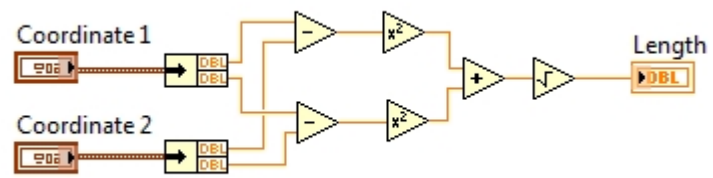


Fig. A.11 Diagrama bloc pentru calculul lungimii unui vector

ANEXA B

În această parte se va prezenta o aplicație realizată în Labwindows/CVI. Această aplicație este o metodă alternativă utilizată pentru controlul brațului robotic folosind camere video. Sistemul constă în învățarea porțiunilor din imagine, care apoi sunt detectate pe imaginea inițială, apoi sunt numerotate. Această metodă nu necesită ca porțiunile de interes să aibă diferite culori, pot fi diferite forme (triunghi, cerc sau pătrat), pe care acest sistem le va detecta.

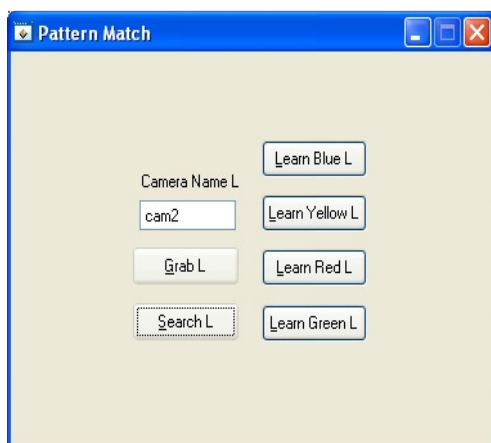


Fig. B.1 Interfața grafică în LabWindows/CVI pentru recunoașterea articulațiilor brațului robotic prin învățarea porțiunilor din imagine

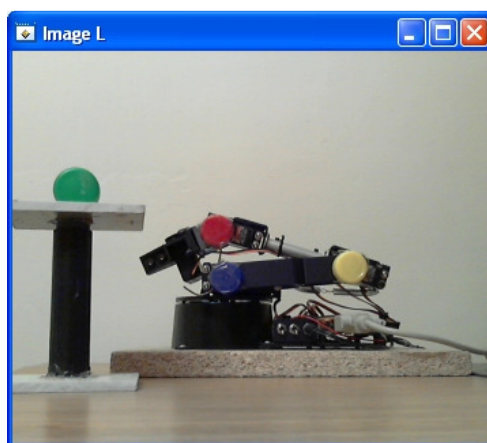


Fig. B.2 Imaginea inițială a brațului robotic Lynxmotion AL5B



Fig. B.3 Porțiuni din imagine învățate de program

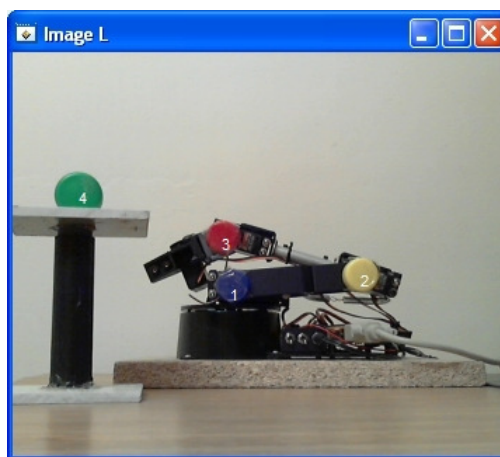


Fig. B.4 Articulațiile detectate din brațul robotic după ce au fost învățate de program

ANEXA C

Lucrarea prezentată are atașat un CD, care conține codurile sursă și proiectele programelor realizate pentru această teză.

Conținutul CD-ului este prezentat în cele ce urmează:

Pentru placa de dezvoltare ATLYS cu Linux

1. Cod în C și executabil, compilat în Linux în linie de comandă pe arhitectura OpenRISC-32 pe 32 biți (microprocesor OpenRISC-32 creat pe FPGA-ul Spartan-6 de pe placa ATLYS) pentru achiziție imagine cu o cameră IP și control braț robotic Lynxmotion AL5B pe interfața serială RS-232, totul afișat pe un monitor conectat la placa ATLYS prin interfața HDMI – fig. 4.9.

Pe PC în Windows cu RoboRealm

2. Proiect în LabWindows/CVI cu executabil în Windows pentru recunoașterea punctelor cheie ale unui braț robotic cu o cameră web utilizând RoboRealm – aplicația necesită RoboRealm instalat și fișierul *.robo încărcat – fig. 4.6, fig. 4.7.
3. Proiect în LabWindows/CVI cu executabil în Windows pentru recunoașterea și controlul cu o cameră web al brațului robotic Lynxmotion AL5B în 2D utilizând RoboRealm – aplicația necesită RoboRealm instalat și fișierul *.robo încărcat – fig. 4.12, fig. 4.13.
4. Proiect în LabWindows/CVI cu executabil în Windows pentru recunoașterea și controlul cu două camere web al brațului robotic Lynxmotion AL5B în 3D folosind algoritmul combinat și utilizând RoboRealm – aplicația necesită RoboRealm instalat și fișierul *.robo încărcat – fig. 4.25.
5. Proiect în LabWindows/CVI cu executabil în Windows pentru recunoașterea și controlul cu două camere web al brațului robotic Lynxmotion AL5B în 3D folosind un singur algoritm și utilizând RoboRealm – aplicația necesită RoboRealm instalat și fișierul *.robo încărcat – fig. 4.26.

Pe PC în Windows cu NI Vision Development Module

6. Proiect în LabWindows/CVI cu executabil în Windows pentru recunoașterea punctelor cheie ale unui braț robotic cu o cameră web utilizând NI Vision Development Module – fig. B1-B4.
7. Proiect în LabWindows/CVI cu executabil în Windows pentru recunoașterea și controlul cu două camere web al brațului robotic Lynxmotion AL5B în 3D utilizând NI Vision Development Module – fig. 4.31, fig. 4.32.
8. Diagramă în LabVIEW pe Windows pentru recunoașterea și controlul cu două camere web al brațului robotic Lynxmotion AL5B în 3D utilizând NI Vision Development Module – fig. 4.28, fig. 4.29, fig. A1-A11.

Pe PC în Ubuntu Linux cu OpenCV

9. Cod în C și executabil în Ubuntu Linux pentru recunoașterea punctelor cheie ale unui braț robotic cu o cameră web utilizând OpenCV – fig. 4.33, fig. 4.34.

10. Cod în C și executabil în Ubuntu Linux pentru recunoașterea și controlul cu două camere web al brațului robotic Lynxmotion AL5B în 3D utilizând OpenCV – fig. 4.37, fig. 4.38.
11. Cod în C și executabil în Ubuntu Linux pentru recunoașterea și controlul cu două camere IP al brațului robotic Lynxmotion AL5B (extensibil și pe SCORBOR-ER III) în 3D utilizând OpenCV – fig. 4.39, fig. 4.40.
12. Cod în C și executabil în Ubuntu Linux pentru recunoașterea și controlul cu două camere web al brațului robotic SCORBOR-ER III în 3D utilizând OpenCV – fig. 4.39, fig. 4.40.

Pentru placa de dezvoltare Raspberry PI în Raspbian Linux cu OpenCV

13. Cod în Python pe Raspbian Linux pe Raspberry PI pentru recunoașterea și controlul cu o cameră web al brațului robotic Lynxmotion AL5B (extensibil și pe SCORBOR-ER III, funcționând și pe ZYBO, ZedBoard) în 2D utilizând OpenCV – fig. 4.43, fig. 4.44.
14. Cod în Python în Raspbian Linux pe Raspberry PI pentru recunoașterea și controlul cu o cameră IP al brațului robotic Lynxmotion AL5B (extensibil și pe SCORBOR-ER III, funcționând și pe ZYBO, ZedBoard) în 2D utilizând OpenCV – fig. 4.46, fig. 4.47.

Pentru placa de dezvoltare ZYBO și ZedBoard în Ubuntu Linux cu OpenCV

15. Cod în C și executabil în Ubuntu Linux pe ZYBO și ZedBoard (Ubuntu Linux instalat pe SoC Zynq, care constă din microprocesor ARM Cortex – A9 Dual-core și FPGA de tip Atrix-7) pentru recunoașterea punctelor cheie ale unui braț robotic cu o cameră web utilizând OpenCV – fig. 4.33, fig. 4.34.
16. Cod în C și executabil în Ubuntu Linux pe ZYBO și ZedBoard (Ubuntu Linux instalat pe SoC Zynq, care constă din microprocesor ARM Cortex – A9 Dual-core și FPGA de tip Atrix-7) pentru recunoașterea și controlul cu două camere al brațului robotic Lynxmotion AL5B în 3D utilizând OpenCV – fig. 4.52, fig. 4.53.
17. Cod în C și executabil în Ubuntu Linux pe ZYBO și ZedBoard (Ubuntu Linux instalat pe SoC Zynq, care constă din microprocesor ARM Cortex – A9 Dual-core și FPGA de tip Atrix-7) pentru recunoașterea și controlul cu două camere IP al brațului robotic Lynxmotion AL5B (extensibil și pe SCORBOR-ER III) în 3D utilizând OpenCV – fig. 4.54, fig. 4.55.
18. Cod în C și executabil în Ubuntu Linux pe ZYBO și ZedBoard (Ubuntu Linux instalat pe SoC Zynq, care constă din microprocesor ARM Cortex – A9 Dual-core și FPGA de tip Atrix-7) pentru recunoașterea și controlul cu două camere web al brațului robotic SCORBOR-ER III în 3D utilizând OpenCV – fig. 4.54, fig. 4.55.