

UNIVERSITATEA POLITEHNICA TIMIȘOARA

SISTEM NECONVENȚIONAL DE CONDUCERE A ROBOȚILOR INDUSTRIALI

Teză destinată obținerii
titlului științific de doctor inginer
la
Universitatea Politehnica Timișoara
în domeniul inginerie mecanică
de către

Ing. Robert KRISTOF

Conducător științific:
prof.univ.dr.ing. Inocențiu MANIU

Seriile Teze de doctorat ale UPT sunt:

- | | |
|---|---|
| 1. Automatică | 11. Știința și Ingineria Materialelor |
| 2. Chimie | 12. Ingineria Sistemelor |
| 3. Energetică | 13. Inginerie Energetică |
| 4. Inginerie Chimică | 14. Calculatoare și Tehnologia Informației |
| 5. Inginerie Civilă | 15. Ingineria Materialelor |
| 6. Inginerie Electrică | 16. Inginerie și Management |
| 7. Inginerie Electronică și Telecomunicații | 17. Arhitectură |
| 8. Inginerie Industrială | 18. Inginerie Civilă și Instalații |
| 9. Inginerie mecanică | 19. Inginerie Electronică, Telecomunicații și Tehnologii Informaționale |
| 10. Știința Calculatoarelor | |

Universitatea Politehnică 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 Politehnică – Timișoara, 2023

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 Politehnică Timișoara. Toate încălcările acestor drepturi vor fi penalizate potrivit Legii române a drepturilor de autor.

România, 300223 Timișoara, Bd. Vasile Pârvan 2B
Tel./fax 0256 404677
e-mail: editura@upt.ro

Cuvânt înainte

"În cunoaștere stă puterea, iar căutarea cunoașterii este o călătorie fără sfârșit către iluminare intelectuală."

- Albert Einstein

Teza de doctorat a fost elaborată pe parcursul activității mele de cercetare în interiorul Departamentului de Mecatronică din cadrul Universității Politehnica Timișoara, pe durata stagiului de doctorat.

Mulțumiri deosebite se cuvin conducătorului de doctorat prof. dr. ing. Inocențiu MANIU, care a fost un mentor deosebit și care m-a îndrumat pe tot parcursul acestei etape, în special pentru discuțiile academice și informațiile valoroase pe care mi le-a oferit. Totodată, aș vrea să mulțumesc și comisiei de îndrumare, în frunte cu conf. univ. dr. ing. Valentin CIUPE, care m-a ghidat încă din primele zile de cercetare, ș.l. univ. dr. ing. Cristian MOLDOVAN și ș.l. univ. dr. ing. Cristian POP care mi-au oferit sfaturi și sugestii valoroase privind implementarea soluției propuse cât și pentru citirea și evaluarea tezei de doctorat. Feedbackul tuturor a jucat un rol esențial în dezvoltarea și îmbunătățirea acestei lucrări.

Totodată, aș vrea să mulțumesc domnului director al Departamentului de Mecatronică, prof. univ. dr. ing. Erwin-Christian LOVASZ, care mi-a permis utilizarea întregii baze materiale a departamentului, m-a sprijinit în activitatea de cercetare și pentru faptul că mi-a oferit posibilitatea să încep o carieră academică ca și cadru didactic universitar.

În același timp, aș vrea să mulțumesc președintelui comisiei, prof. univ. dr. ing. Liviu MARȘAVINA, cât și celor doi referenți, prof. univ. dr. ing. Doina Liana PÎSLĂ și prof. univ. dr. ing. Radu Cătălin ȚARCĂ, care și-au dedicat timp pentru evaluarea tezei, cât și pentru realizarea referatelor de specialitate.

Un aport semnificativ la realizarea tezei de doctorat l-a avut și compania DRÄXLMAIER care a răspuns pozitiv la solicitarea mea de a sponsoriza realizarea unui nou laborator de robotică, cu suma de aprox. 90.000 euro, prin care s-a renovat și dotat sala 411 din cadrul Departamentului de Mecatronică și prin care s-a achiziționat un robot modern, de la Universal Robots, modelul UR10e, pe care s-a desfășurat activitatea mea de cercetare.

Nu în ultimul rând, aș vrea să mulțumesc familiei mele, care a fost înțelegătoare cu mine, în special în perioada redactării tezei de doctorat și care m-a susținut permanent în toată această etapă din viața mea.

Timișoara, iulie 2023

dipl. ing. Robert KRISTOF

KRISTOF, Robert

Sistem neconvențional de conducere a roboților industriali

Teze de doctorat ale UPT, Seria X, Nr. YY, Editura Politehnica, 200Z, 129 pagini, 87 figuri, 18 tabele.

ISSN:

ISBN:

Cuvinte cheie

Teleoperare, roboți industriali, master-slave, Myo Armband, Kuka KR15, Universal Robots UR10e, recunoașterea gesturilor

Rezumat

Lucrarea prezintă o soluție de control a unui robot industrial cu ajutorul gesturilor și mișcărilor mâinii, practic a unei alternative cu funcții de bază, a consolei de învățare, care vine în dotarea oricărui robot industrial. Ideea de bază este deschiderea posibilităților de programare a un robot industrial către publicul larg, indiferent de noțiunile tehnice dobândite.

Activitatea de cercetare a vizat controlul roboților Kuka KR15 și Universal Robots UR10e cu ajutorul brățării Myo Armband, prin care se pot detecta gesturile și mișcările mâinii, realizate de utilizator.

Rezultatele cercetării constau în testarea soluției de către utilizatori, măsurând atât timpul și scorul de precizie, cât și aplicarea de chestionare de feedback. Totodată este prezentată și o analiză comparativă dintre conducerea unui robotului industrial cu ajutorul consolei mobile de învățare și cu ajutorul Myo Armband.

CUPRINS

1. INTRODUCERE	7
1.1. Introducere în robotică	7
1.1.1. Roboții și robotica.....	7
1.1.2. Evoluția sistemelor robotice.....	8
1.1.3. Clasificarea roboților	11
1.2. Roboții industriali	12
1.2.1. Considerații generale.....	12
1.2.2. Statistici	13
2. STADIUL ACTUAL ÎN DOMENIUL CONDUCERII ROBOȚILOR INDUSTRIALI	17
2.1. Consolă mobilă de învățare	20
2.2. Microsoft Kinect.....	22
2.3. Telefonul mobil.....	25
2.4. Joystick	26
2.5. Gamepad	27
2.6. Leap Motion.....	28
2.7. Myo Armband.....	29
3. SOLUȚIA PROPUȘĂ ȘI ETAPELE CERCETĂRII	30
3.1. Dispozitive utilizate.....	30
3.1.1. Robotul industrial KUKA KR15	30
3.1.2. Brățara Myo Armband.....	31
3.1.3. Universal Robots UR10e	34
3.2. Soluția propusă	34
3.3. Etapele cercetării	35
3.4. Obiectivele cercetării	35
4. EVALUAREA SOLUȚIEI OPTIME	36
4.1. Aplicație conexiune Myo Armband cu Arduino.....	36
4.1.1. Arhitectura sistemului	36
4.1.2. Schema de conectare	38
4.1.3. Încercări și rezultate experimentale	39
4.1.4. Concluzii.....	40
4.2. Aplicație conexiune Myo cu KUKA KR15	41
4.2.1. Arhitectura sistemului	41
4.2.2. Schema de conectare	42
4.2.3. Încercări și rezultate experimentale	42
4.2.4. Concluzii.....	58
4.3. Aplicație conexiune Dual Myo – KUKA KR15.....	58
4.3.1. Arhitectura sistemului	59
4.3.2. Schema de conectare	59
4.3.3. Încercări și rezultate experimentale	61
4.3.4. Concluzii.....	67
4.4. Aplicație Myo Armband – Universal Robots UR10e	68

4.4.1.	Arhitectura sistemului	68
4.4.2.	Schema de conectare	69
4.4.3.	Încercări și rezultate experimentale	69
4.4.4.	Concluzii.....	80
5.	METODOLOGIA DE CERCETARE ȘI TESTAREA SOLUȚIEI OPTIME	81
5.1.	Testul 1 - Preluarea și plasarea a trei obiecte (paralelipiped din plastic)	84
5.2.	Testul 2 - Atingerea celor patru repere (bețe).....	86
5.3.	Testul 3 – Preluarea și plasarea obiectului (sticlă PET) în cutie...	86
6.	REZULTATE EXPERIMENTALE	87
6.1.	Rezultatele chestionarelor	87
6.1.1.	Ușurința mișcării robotului	88
6.1.2.	Precizia de poziționare	89
6.1.3.	Timpul de programare.....	89
6.1.4.	Satisfacția generală de utilizare.....	90
6.1.5.	Concluzii.....	91
6.2.	Rezultatele măsurărilor	92
6.2.1.	Testul 1 - Preluarea și plasarea a trei obiecte (paralelipiped din plastic)	92
6.2.2.	Testul 2 - Atingerea celor patru repere (bețe).....	96
6.2.3.	Testul 3 - Preluarea și plasarea unui obiect (sticlă PET) în cutie.....	97
6.2.4.	Concluzii.....	98
7.	CONCLUZII. CONTRIBUȚII PERSONALE. DIRECȚII VIITOARE DE CERCETARE	99
7.1.	Concluzii finale.....	99
7.2.	Contribuții personale	100
7.3.	Direcții de cercetare viitoare	101
8.	BIBLIOGRAFIE.....	102
	LISTA FIGURILOR.....	110
	LISTA TABELELOR	112
	ANEXE	113

1. INTRODUCERE

1.1. Introducere în robotică

1.1.1. Roboții și robotica

Prima referință a cuvântului robot datează încă din anul 1920, când scriitorul ceh Karel Čapek a scris *Rossumovi Univerzální Roboti (RUR)* (Fig. 1) unde a menționat cuvântul „*rabota*” pentru a exemplifica un caracter ficțional de tip umanoid[1]. Conform [2] și [3] adevăratul inventator al cuvântului este de fapt Josef Čapek, fratele lui Karel. Cercetările spun că în timp ce Karel Čapek lucra la RUR, s-a întors către fratele acestuia pentru a-i da o sugestie de cuvânt care descria un caracter uman lucrător.

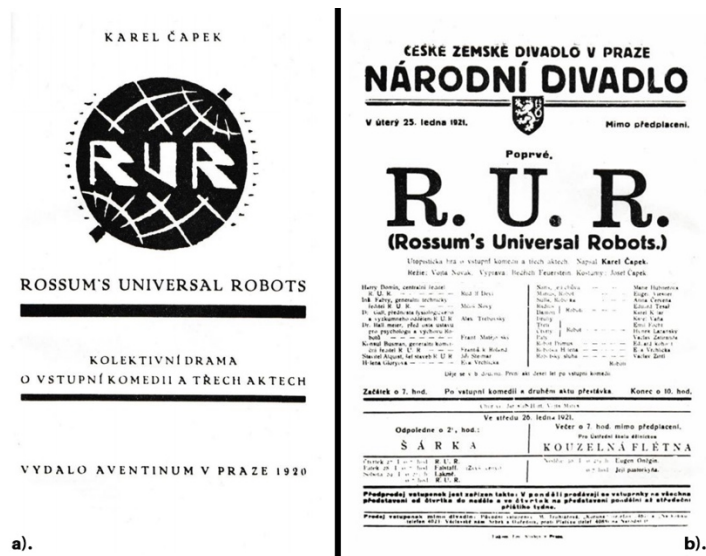


Fig. 1 a). Coperta RUR; b).Programul pentru premiera RUR în cadrul Teatrului Național din Praga. 21 Jan 1921 [3]

Institutul de Robotică din America definește un robot în felul următor: „Un robot este un manipulator multifuncțional reprogramabil conceput pentru a muta material, piese, unelte sau dispozitive specializate prin mișcări variabile programate pentru îndeplinirea unei varietăți de sarcini” [4]. Roboții sunt folosiți în domenii diverse, cel mai adesea în producție, datorită avantajelor pe care le aduc în creșterea productivității.

Robotica este definită de Saied B. Niku, în[5], ca fiind „arta, baza de cunoștințe și know-how-ul de proiectare, aplicare și utilizare a roboților în eforturile umane”. Robotica este un subiect care îmbină elemente de mecanică, electronică și informatică.

Vukobratovic, în [6], menționează că sistemele robotice reprezintă tehnologii noi și complexe de automatizare a procesului de producție astfel încât munca manuală

poate fi eliminată complet. Acestea pot fi schimbate cu ușurință pentru a satisface mai multe tipuri de operații. Astfel, roboții industriali reprezintă mașini polivalente care îndeplinesc cerințele actuale ale pieței în materie, realizând astfel un avantaj economic și productiv.

Termenul de robotică a fost însă cel mai bine mediatizat de către scriitorul de ficțiune, Isaac Asimov, care în 1942 în *Runaround* [7] și mai apoi în 1950 în *I, Robot* [8] a introdus cele trei legi ale roboticii care mai sunt numite și Legile lui Asimov:

„Legea 1: Un robot nu poate să rănească o ființă umană sau, prin inacțiune, să permită unei ființe umane să facă rău.

Legea 2: Un robot trebuie să se supună ordinelor date de ființele umane, cu excepția cazurilor în care astfel de ordine ar intra în conflict cu prima lege.

Legea 3: Un robot trebuie să-și protejeze propria existență, atâta timp cât o astfel de protecție nu contravine primei sau celei de-a doua legi.” [7]

Legile lui Asimov au avut un impact asupra eticii în ceea ce a urmat, dezvoltarea inteligenței artificiale, fiind un punct important de plecare pentru pionierii din domeniu.

1.1.2. Evoluția sistemelor robotice

Sistemele de conducere ale roboților industriali au evoluat treptat în timp, practic ușurând capacitatea utilizatorului de a controla robotul. Robotul industrial a fost întotdeauna ghidat și condus prin intermediul unui sistem de conducere, încă din primele zile de când a fost inventat. Schimbările cerințelor de producție, de la convenționala producție în masă, la diversificarea design-ului de produs cu cantitatea flexibilă de producție, au stimulat mult interes în domeniul industrial pentru robotică.[9]

În [10], Joseph F. Engelberger a făcut o comparație între abilitățile umane și cum acestea ar putea fi preluate de către un robot industrial. Acesta a analizat capacitățile umane, menționând că ar trebui să fie posibilă întocmirea unei liste de acțiuni pe care un robot industrial ar putea să le preia. Această idee a fost concretizată în Tabelul 1. În partea stângă a tabelului sunt enumerate activitățile și alți factori relevanți pentru orice operator care lucrează la o mașină unealtă. În centrul tabelului se află un comentariu asupra performanței unui operator uman. Modalitatea de realizare a acțiunii, de către un operator robot, este prezentată în coloana din dreapta. Proprietățile fizice umane care nu sunt esențiale în producție, nu au fost incluse în această listă. Totodată, robotul industrial nu are picioare pentru că nu are nevoie de această abilitate pentru a opera ca o mașină, iar din cauza limitărilor tehnologice robotul nu va putea să se bucure de simțurile umane precum gustul, mirosul, auzul, atingerea sau vederea.

Tabelul 1- Comparație între caracteristicile unui om și ale unui robot [10]

FUNCȚIE	OPERATOR UMAN	OPERATOR ROBOT
alegere piesă de prelucrat	H1. Utilizează simțurile vederii și atingerii.	R1. Metodele de detecție vizuală sunt destul de dificil de implementat. Acestea sunt limitate la detectarea contactului fizic și printr-o comandă pre-programată care direcționează robotul către locația piesei.
manipulare	H2. Utilizează combinația brațelor, mâinilor și corpului. Posibilă nevoie de asistență mecanică la ridicarea de obiecte grele.	R2. Robotul funcționează asemenea ca omul, având un „braț” și o „mână”. Acestea trebuie conectate pentru a permite mâinii să se miște în trei dimensiuni și să se rotească. Utilizarea puterii hidraulice poate produce o capacitate de manipulare mai mare decât cea a unui operator uman.
situare	H3. Similar cu H2, dar își folosește ochii pentru a se asigura că piesa de prelucrat este orientată corect și poate fi plasată.	R3. Similar cu R2, dar fără simțul vederii, orientarea devine dificilă. Un operator robot are nevoie de piese care îi sunt prezentate într-o poziție predefinită.
operare utilaj	H4. Folosește oricare sau toate cele cinci simțuri ale sale pentru a urmări funcționarea mașinii și pentru a activa comenzile după cum este necesar. Are o memorie, cu care poate învăța secvența și programul de rulare.	R4. Deoarece nu este capabil să „vadă”, „să audă” sau să asiste la funcționarea mașinii, un robot trebuie să fie pre-programat pentru a-și desfășura operațiunile conform unei secvențe predefinite. Programarea se face de către un om, iar programul și datele trebuie stocate în memoria internă a robotului.
descarcă utilaj	H5. Similar cu H2	R5. Similar cu R2
stivuire piesă de prelucrat	H6. Similar cu H3	R6. Similar cu R3, dar orientarea este mai facilă, deoarece mașina va fi, de obicei, una care menține fiecare piesă în aceeași poziție.

10 - INTRODUCERE

inspectare piesă de prelucrat	H7. Un om ar trebui să poată inspecta, după cum este necesar, folosindu-și simțurile vederii și atingerii sau folosind indicatoare și alte echipamente de măsurare. El este capabil să vorbească și să-i spună superiorului orice problemă.	R7. Un robot ar putea inspecta prin etalonare automată, sondare și prin telemetrie sau stocarea istorică a datelor. Deși nu este capabil să vorbească, robotul ar putea fi proiectat pentru a da un avertisment (semnal) sonor sau de altă natură cu privire la orice defect al piesei de prelucrat.
defecțiunea utilajului	H8. Capabil să folosească oricare din simțurile sale pentru a percepe posibile probleme. De exemplu, el poate recunoaște mirosul de la un motor fierbinte și poate opri mașina.	R8. Un robot ar putea să detecteze o defecțiune a mașinii numai în cazul în care aceasta produce defecte în piesele pe care robotul le-ar putea detecta sau în cazul în care au fost instalate puncte telemetrice în mașină. Totodată, inspecția se poate realiza și pe baza vibrațiilor utilajelor.
setarea utilajului	H9. Trebuie să solicite asistență calificată ori de câte ori mașina are nevoie de resetare sau de instalare de echipamente noi.	R9. Robotul ar putea fi programat să se oprească după o perioadă calculată pentru a preveni distrugerea sculelor. De asemenea, este posibilă comandarea robotului pentru sondarea piesei de prelucrat și/sau utilajul pentru defecțiuni la găuri de foraj, robinete, alezaje și alte scule. Un operator uman calificat este necesar pentru a efectua rectificarea și resetarea.
schimbul de ture	H10. Un om trebuie să plece acasă după un schimb care durează între opt și douăsprezece ore. El trebuie să fie mobil - capabil să meargă și să alerge.	R10. Nu este nevoie ca un robot să fie mobil. Poate fi fixat pe podea și comandat să lucreze în continuu în fiecare zi. Întrucât trei schimburi sunt operate în decurs de 24 de ore pe o singură mașină, un operator robot poate lua locul a trei oameni sau mai mulți.
pauze de lucru	H11. Un om trebuie să își ia timp liber de la muncă pentru a mânca și pentru alte nevoi fizice	R11. Roboții nu au nevoie de timp liber pentru a mânca sau pentru alte nevoi naturale. Aceștia vor lua pauză doar în timpul programului de întreținere sau când vor exista defecțiuni ocazionale.

greve, mersuri încetinite și interdicții de lucru	H12. Acțiunea colectivă perturbatoare din câmpul muncii este o problemă care poate lovi orice unitate de producție	R12. Un robot se va supune în totalitate comenzilor și nu există nicio organizație de muncă pentru roboți.
boală sau oboseală	H13. Oamenii obolesc și pot lipsi de la locul de muncă atunci când se îmbolnăvesc. Acele condiții pot fi aggravate sau cauzate de medii de lucru neplăcute (căldură, zgomot, vapori toxici, mirosuri, spațiu înghesuit, etc.).	R13. Este de așteptat ca roboții să se defecteze ocazional (vezi R 11), însă este posibil să se proiecteze un robot care să funcționeze în condiții extreme, condiții care ar putea deteriora sănătatea mentală sau fizică a unui om.

1.1.3. Clasificarea roboților

Există mai multe variante de clasificări ale roboților. În [6], Vukobratovic realizează o clasificare generală a sistemelor robotice astfel:

1. Sisteme robotice pentru manipulare;
2. Sisteme robotice mobile;
3. Sisteme robotice pentru informație și control.

O altă variantă de clasificare ar fi cea făcută de Asociația Franceză de Robotică

(AFR):

- A. dispozitive de manipulare cu control manual de telerobotică;
- B. dispozitive de manipulare automată cu cicluri predeterminate;
- C. roboți programabili, servo-controlați, cu traiectorii continue sau punct-cu-punct;
- D. similar cu C, dar capabili să obțină informații din mediul lor. [5]

Cea mai actuală și cea mai folosită clasificare este cea dată de Federația Internațională de Robotică (IFR), care a ajuns la o înțelegere cu producătorii de roboți, astfel începând din 2004 roboții se clasifică doar după structura mecanică, după cum urmează:

1. Roboți liniari (inclusiv roboții cartezieni sau fixați în poziție);
2. Roboți SCARA;
3. Roboți articulați;
4. Roboți paraleli (DELTA);
5. Roboți cilindrici;
6. Alții;
7. Neclasificați. [11]

1.2. Roboții industriali

1.2.1. Considerații generale

Inventatorul robotului industrial este americanul George C. Devol, care în 1954 a înregistrat brevetul pentru primul robot industrial. Robotul industrial reprezintă cea mai mare inovare în domeniul producției, element care a făcut ca cele mai importante companii să adopte linii de producție automatizate, încă din anul 1962, când General Motors a instalat primul robot industrial denumit Unimation, pe baza brevetului cumpărat de la Devol în 1961. [12]

Conform ISO 8373:2012, robotul industrial este definit ca „*un manipulator automat controlat, reprogramabil și multifuncțional, programabil în trei sau mai multe axe, care poate fi fixat în poziție sau mobil pentru utilizare în aplicații de automatizare industrială.*” [13]

Robotul industrial este probabil cea mai importantă invenție care a venit în sprijinul domeniului industrial și întregii activități de producție. Prin avantajele sale, acesta contribuie semnificativ la productivitate, în orice gamă de producție, tendința globală în domeniu fiind de automatizare prin roboți industriali.

Avantajele roboților industriali:

- a) Costuri scăzute de producție
Prin faptul că un robot poate lucra 24 ore/zi și poate realiza operațiuni mult mai rapid decât un operator uman, un robot industrial va fi mult mai productiv.
- b) Produse de calitate
Un robot realizează produse cu o precizie foarte mare. Astfel, se asigură faptul că la un volum mare de produse, în procesul tehnologic, produsele rezultate vor fi aproape identice și de calitate superioară.
- c) Eficiență în utilizarea spațiului de lucru
Un robot industrial nu trebuie să se deplaseze în interiorul fabricii. Acesta este poziționat într-un spațiu și realizează operațiunile din acel punct, lăsând astfel loc în fabrică pentru alte activități.
- d) Folosirea eficientă a materialelor
Întrucât un robot are o precizie foarte mare, în cadrul operațiilor tehnologice, materialele vor fi folosite la maxim, iar resturile vor fi reduse.
- e) Siguranță
Operațiunile periculoase care erau executate de oameni, pot fi realizate de roboți industriali, crescând astfel siguranța în muncă. Sistemele de siguranță implementate pe roboți asigură faptul că șansele ca un operator uman să fie rănit să fie minime.
- f) Amortizare rapidă a investiției
Crescând productivitatea, cresc și veniturile, astfel un robot industrial se amortizează într-un timp relativ scurt.
- g) Utilizare în mai multe domenii
Un robot industrial poate fi adaptat să realizeze mai multe operațiuni în domeniul producției, astfel el poate suda, manipula, împacheta, paletiza, tăia, etc.

Dezavantajele roboților industriali:

- a) Investiție inițială relativ mare
Un robot industrial deși se amortizează într-un timp relativ scurt, necesită o investiție inițială destul de mare pe care nu orice companie își poate permite să o asume.
- b) Personal calificat
Nu oricine poate programa un robot industrial sau chiar să realizeze operațiuni de mentenanță, astfel, într-o lume în care aproape toate companiile suferă de lipsă de forță de muncă calificată, va fi destul de greu să se găsească specialiști care să asigure programarea și mentenanța roboților industriali.
- c) Concedieri
Un robot poate înlocui munca a mai multor persoane, astfel, inevitabil se vor produce concedieri masive odată cu achiziția unui robot industrial, care pot duce la greve sau procese în instanță.

Domeniul roboților industriali a cunoscut o evoluție notabilă cel puțin în ultimii 40 de ani, atunci când a fost prezentat și lansat primul robot din familia KUKA. Începând din acel moment, roboții au devenit tot mai inteligenți și sunt folosiți în tot mai multe arii de activitate.[14]

Teleoperarea dispozitivelor robotice este un domeniu destul de bine studiat în diverse aplicații precum [15]–[22].

1.2.2. Statistici

Analizând datele furnizate de Federația Internațională de Robotică (IFR) din Tabelul 2, publicate în [23], putem vizualiza stadiul actual privind numărul de roboți instalați anual în funcție de regiunea geografică. Se poate observa cum regiunea Asia/Australia este lideră în ceea ce privește numărul de instalări anuale, iar regiunea America este codașă în acest clasament.

Tabelul 2 - Instalări anuale de roboți pe regiuni (mii unități) [23]

Anul	Asia/Australia	Europa	America	Total
2011	89	44	26	159
2012	85	41	28	154
2013	99	43	33	175
2014	134	46	33	213
2015	161	50	38	249
2016	200	56	41	297
2017	280	67	46	393
2018	284	76	55	415
2019	259	74	47	380
2020	277	68	39	384
2021	381	84	51	516

Tot pe baza acestui tabel putem observa în Fig. 2 evoluția instalărilor roboților industriali în perioada 2011-2021, inclusiv din perioada pandemiei COVID-19. Astfel, în 2019-2020 putem observa o scădere față de anul 2018, iar în 2021, anul doi de la declanșarea pandemiei, se observă o explozie a numărului de instalări în toate regiunile, în special în Asia/Australia, având China ca locomotivă de dezvoltare în această regiune. Companiile au realizat importanța și beneficiul roboților industriali care puteau funcționa și în pandemie, fiind imuni la boli sau dezastre naturale.

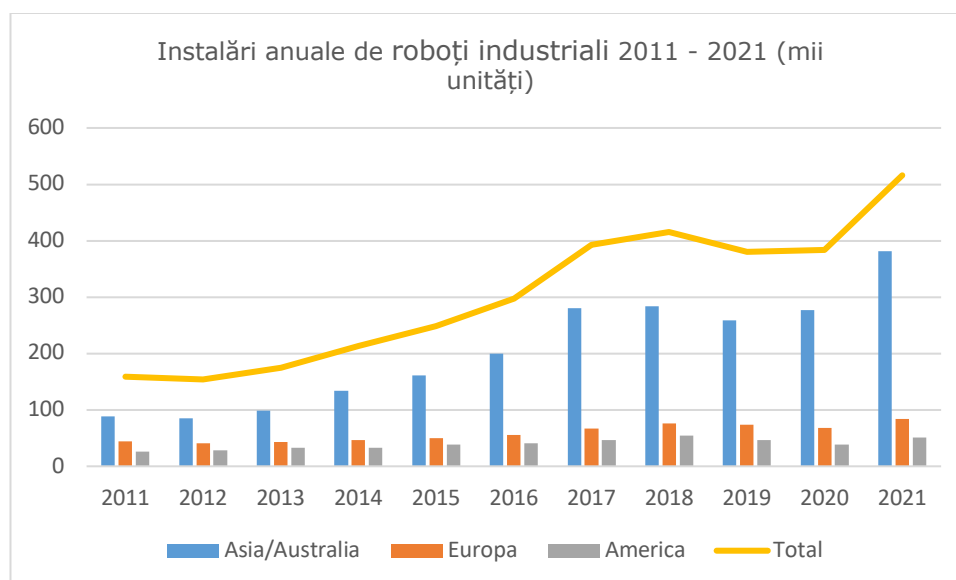


Fig. 2 – Instalări anuale de roboți industriali 2011 - 2021 (mii unități)

În Tabelul 3 putem observa numărul de instalări de roboți industriali, în funcție de domeniul de activitate al companiilor. Astfel, putem observa faptul că lider în acest clasament este domeniul electric/electronic, urmat de domeniul automotive.

Tabelul 3 - Instalări anuale de roboți industriali în funcție de domeniul de activitate 2019-2021 (mii unități)[23]

Domeniul	2019	2020	2021
Electric/electronic	89	110	137
Automotive	102	84	119
Metal și mașinării	52	44	64
Plastic și produse chimice	18	19	24
Alimentar	11	12	15
Altele	30	37	52
Nespecificat	87	87	107

În Fig. 3 putem observa evoluția instalărilor de roboți industriali în funcție de domeniul de activitate, în perioada 2019-2021. Din analiza graficului putem să distingem faptul că până în anul 2019 domeniul automotive a fost lider ca domeniu de implementare a roboților industriali. Din anul 2020, domeniul electric/electronic a preluat conducerea și este lider în acest clasament, iar domeniul alimentar este domeniul cu cei mai puțini roboți industriali instalați.

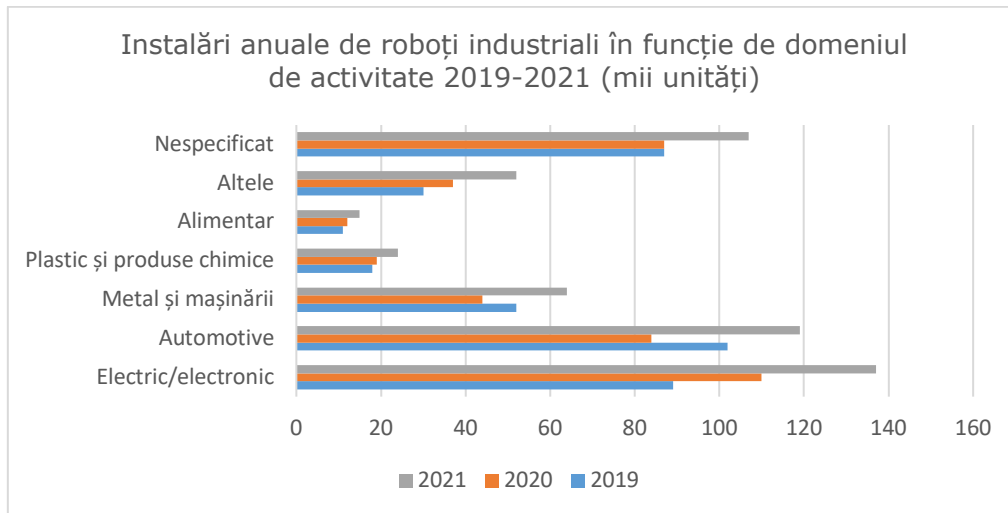


Fig. 3 - Instalări anuale de roboți industriali în funcție de domeniul de activitate 2019-2021 (mii unități)

În Tabelul 4 putem să observăm topul primelor 15 țări în funcție de numărul de instalări de roboți industriali în 2021. Se poate observa cum China domină clar acest clasament, având în 2021 aproximativ 268 200 roboți industriali instalați, cu o creștere de 51% față de anul precedent, astfel China atingând un record absolut.

Tabelul 4 - Top 15 țări după instalări roboți industriali în 2021[23]

Țara	Roboți Industriali (mii unități)	Evoluție
China	268.2	51%
Japonia	37.2	22%
SUA	35	14%
Republica Coreea	31.1	2%
Germania	23.8	6%
Italia	14.1	65%
Taipei	9.6	31%
Franța	5.9	11%
Mexic	5.4	61%

India	4.9	54%
Canada	4.3	66%
Thailanda	3.9	36%
Singapore	3.5	-35%
Spania	3.4	1%
Polonia	3.3	56%

În Fig. 4 putem observa cum pandemia COVID-19 a influențat topul celor 15 țări după numărul de instalări. Astfel, se poate observa că șase țări au accelerat semnificativ implementarea de roboți industriali, având creșteri de peste 50% față de anul precedent: China, Italia, Mexic, India, Canada, Polonia. O singură țară a avut o evoluție negativă: Singapore.

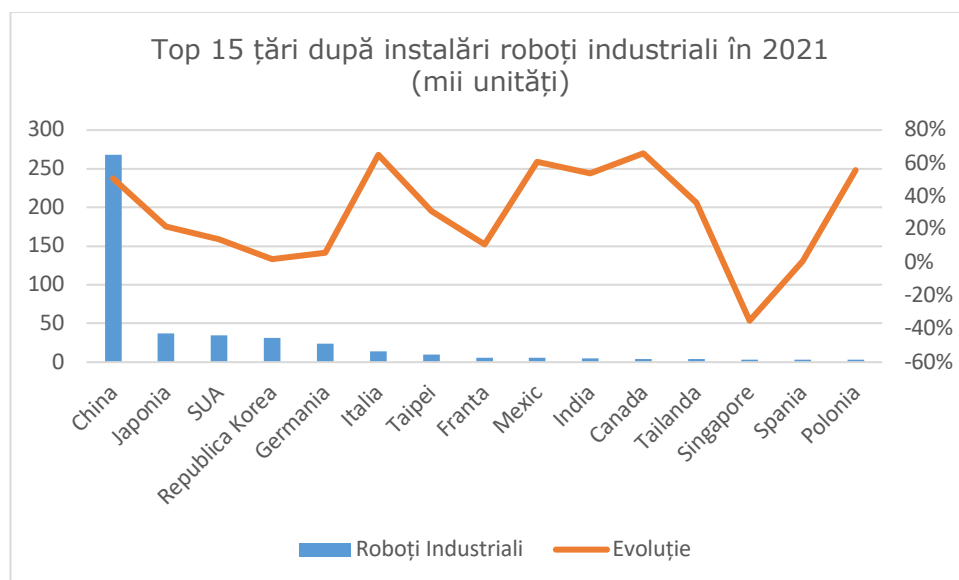


Fig. 4 - Top 15 țări după instalări roboți industriali în 2021

2. STADIUL ACTUAL ÎN DOMENIUL CONDUCERII ROBOȚILOR INDUSTRIALI

Toți roboții industriali au un sistem de tip controler prin care se primesc, se procesează și se transmit comenzi către robotul industrial propriu-zis. Acesta poate fi controlat și prin intermediul altor dispozitive atașate la controlerul robotului, practic realizând o operațiune de teleoperare. Teleoperarea este domeniul unde oamenii controlează de la distanță senzori și dispozitive de acționare [24]. Există, de asemenea, o formă avansată a acesteia, unde un operator uman exercită un control de supraveghere asupra unui sistem robot prin intermediul unui calculator sau al unui dispozitiv similar. Această metodă se numește telerobotică și permite operatorului să execute sarcini la distanță și să obțină informații despre cele executate anterior. [25]

Pe de altă parte, în conceptul de master-slave, cele două componente comunică unidirecțional, elementul master fiind cel care comandă unul sau mai multe elemente slave. [26]

În Fig. 5 se poate observa atât schema bloc a sistemului de comandă master-slave la manipuloarele sincrone MS-200, MS-500, MS-1000, unul dintre primele astfel de sisteme de la noi din țară, cât și gruparea elementelor pe subsisteme principale, și anume, subsistemul Master și subsistemul Slave care pot fi observate mai în detaliu în Fig. 6. Cele două componente sunt conectate printr-o rețea informațională multiplă, cu un traseu direct și mai multe trasee inverse, de reacție (de forță, de poziție, tactilă). [26]

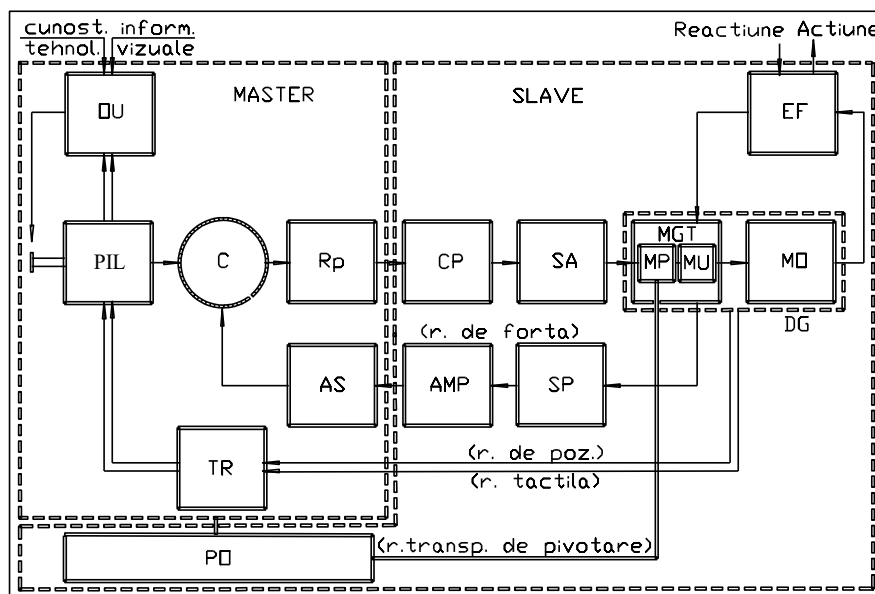


Fig. 5 – Schema bloc a sistemului de comandă master-slave la manipuloarele sincrone MS-200, MS-500, MS-1000[26]

Astfel, autorii din [26] au definit următoarele elemente componente:

- PIL: mecanism de pilotare, pe care se găsesc elemente de intrare;
- OU: operator uman;
- C: comparatorul de semnale;
- Rp: regulator PIC de joasă putere;
- CP: convertor de putere;
- DG: dispozitiv de ghidare;
- MGT: mecanism generator de traiectorie;
- MO: mecanism de orientare;
- MP: mecanism de pivotare;
- MU: mecanism de urmărire;
- EF: efector final;
- PO: platforma operatorului;
- TR: transmisii;
- SP: senzori piezoelectrice;
- AMP: bloc de amplificare;
- AS: adaptor de semnale.

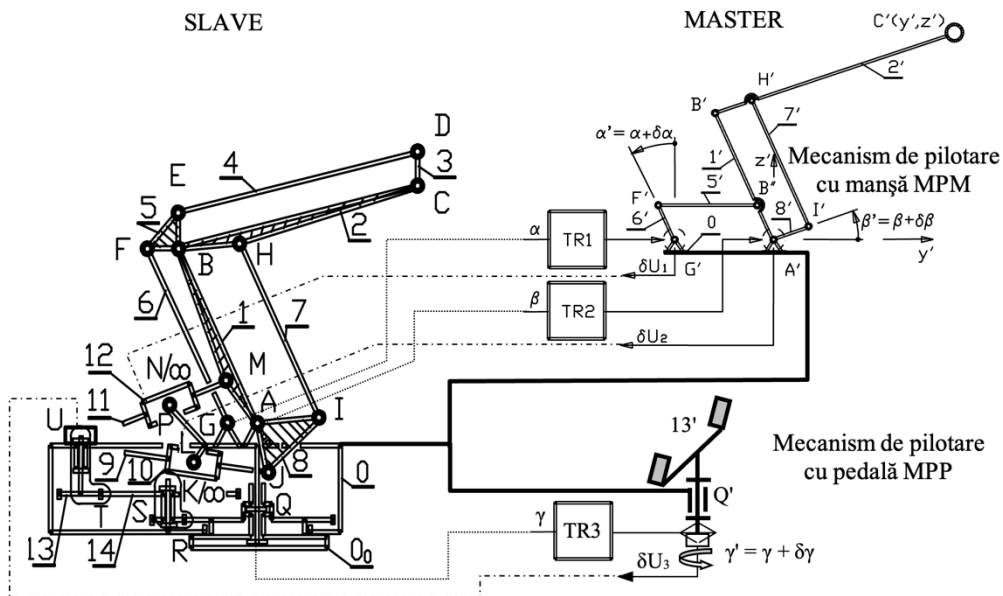


Fig. 6 - Schema integrării sistemului de comandă cu sistemul mecanic la manipuloarele sincrone MS-200, MS-500, MS-1000[26]

În Fig. 6, deplasarea manuală în plan a manetei de comandă C' aparținând mecanismului de pilotare cu manșă MPM, produce rotații cu unghiul $\delta\alpha$ la brațul de comandă $1'$ și cu unghiul $\delta\beta$ la antebrățul de comandă $2'$. Acționarea pedalei de comandă $13'$ din componenta mecanismului de pilotare cu pedală MPP produce pivotarea cu unghiul $\delta\gamma$. [26]

Comparatoarele din cuplele cinematice de comandă G' , A' și Q' , având ca intrări atât unghiurile $\alpha' = \alpha + \delta\alpha$, $\beta' = \beta + \delta\beta$, $\gamma' = \gamma + \delta\gamma$ de la mecanismele de

pilotare, cât și unghiurile de reacție α , β , γ culese din cuplele cinematice conduse G , A , și Q ale lanțului cinematic slave, convertesc diferențele acestor unghiuri prin intermediul unor punți potențiometrice, în tensiunile de comandă: δU_1 - pentru motorul din N/∞ în vederea acționării brațului 1, δU_2 - pentru motorul din K/∞ în vederea acționării antebrațului 2 și δU_3 - pentru motorul din U în vederea acționării elementului pivotant 0. [26]

Roboții industriali sunt comandați și programați prin intermediul unor console de învățare. Aceste console sunt conectate direct la controlerul robotului și sprijină operatorul uman în activitatea de a ghida și programa robotul.



Fig. 7 – Consolă mobilă de învățare KUKA KR15

Consolele de învățare au evoluat semnificativ în ultimii ani, întrucât producătorii de roboți industriali au făcut eforturi pentru a moderniza aceste dispozitive lucrând în special la interfață. Astfel, au fost introduse programarea bazată pe icoane [24], ecranul tactil color, joystick 3D (ABB Robotics), dispozitive 6D (KUKA) [27]. În Fig. 7 este prezentat un exemplu de astfel de consolă de învățare de la KUKA KR15.

În [28] se menționează noi tendințe în dezvoltarea consolelor de învățare de către departamentul de cercetare și inovare al KUKA, și anume proiectul MORPHA privind dezvoltarea de noi generații de console de învățare care să aibă ecran tactil și funcție de comandă vocală (Fig. 8). Ideea de bază în aceste proiecte este ca utilizatorul să poată programa roboții fără prea multe noțiuni de programare sau robotică. Astfel, două proiecte urmează două idei importante care sunt fundamentale pentru noile generații de console de învățare:

1. control multi-robot, prin care o consolă de învățare poate controla mai mulți roboți industriali;
2. conectivitate wireless între consola de învățare și controlerul robotului.



Fig. 8 - Consolă mobilă de învățare KUKA cu ecran tactil și recunoaștere vocală [28]

O altă perspectivă de viitor menționată în [28] o reprezintă realitatea augmentată iar primul pas către acest deziderat se poate face odată cu evoluția dispozitivelor de control a realității augmentate, montate pe capul utilizatorului.

Însă nu doar consolele mobile de învățare pot controla robotul industrial. Din analiza realizată, putem distinge mai multe tipuri de dispozitive care pot controla robotul industrial:

- a) Consolă mobilă de învățare;
- b) Microsoft Kinect;
- c) Telefonul mobil;
- d) Joystick;
- e) Gamepad;
- f) Leap Motion;
- g) Myo Armband.

2.1. Consolă mobilă de învățare

Această metodă implică conectarea unui dispozitiv mobil la controler, care are o tastatură și ecran. Dispozitivul, pe lângă faptul că poate controla de la distanță robotul industrial, poate juca și rolul de interfață de programare, fie prin învățarea treptată a pozițiilor, fie prin scrierea unor linii de cod în limbajul de programare specific fiecărui robot industrial.



Fig. 9 - Consolă mobilă de învățare pentru roboții industriali a) MOTOMAN NX100 b). ABB IRC5 c). KUKA Robotics d). COMAU Robotics e). FANUC [29]

În Fig. 9 putem distinge mai multe tipuri de design pentru consolele mobile de învățare, în funcție de producătorul robotului industrial.

Consola mobilă de învățare face parte dintr-un întreg sistem privind conducerea și controlul unui robot industrial. În Fig. 10 sunt prezentate toate componentele acestui sistem.

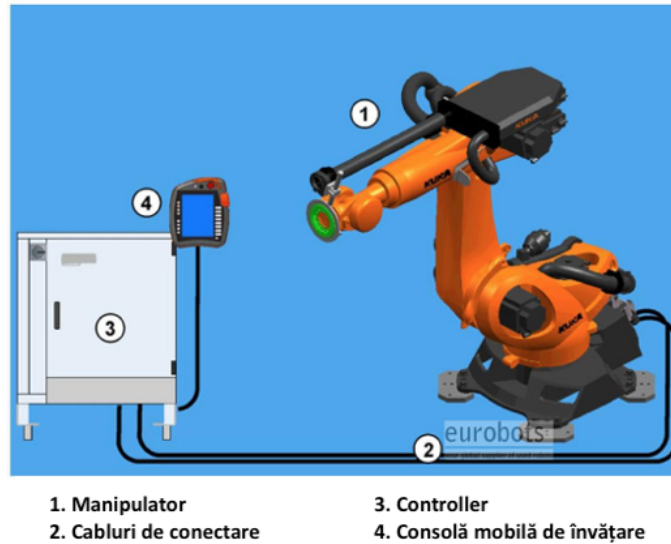


Fig. 10 - Componente ale unui sistem de conducere, control și învățare a unui robot industrial KUKA [25]

Din punctul de vedere al utilizatorului, controlerul este echipat cu un dispozitiv, care permite utilizatorului să miște sau să învețe robotul. Panoul de control oferă un comutator cu trei căi, precum și diferite moduri de operare permițând realizarea aplicațiilor și testarea acestora în siguranță. Panoul de control mai poate fi folosit și pentru afișaj, execuție și editare a programelor în limbajul de programare specific robotului industrial (Fig. 11). [30]

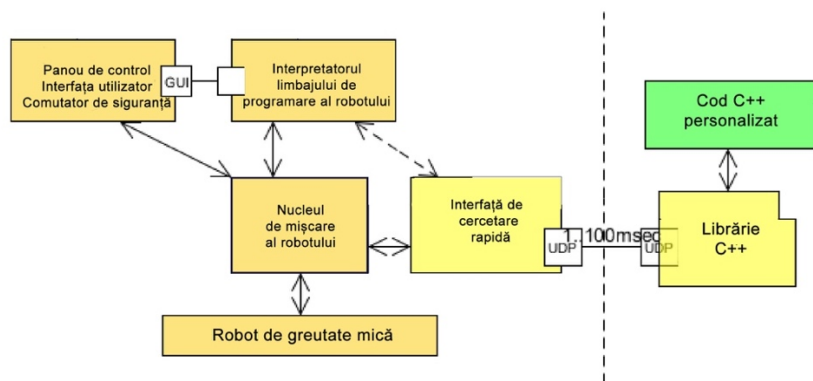


Fig. 11 - Exemplu de schemă de comunicare în cadrul unui robot KUKA [30]

Aceasta este cea mai răspândită metodă de a controla roboții și cea mai ușoară de implementat, având o precizie foarte bună.

2.2. Microsoft Kinect

Microsoft Kinect este un dispozitiv creat pentru industria jocurilor pe calculator. Dispozitivul a revoluționat felul în care oamenii se joacă oferind posibilitatea acestora să interacționeze cu jocul prin intermediul corpului lor, într-un mod natural. Analizând limbajul corpului, computerul trebuie în primul rând să înțeleagă mai întâi ce face un utilizator înainte ca el să poată reacționa. Aceasta a fost întotdeauna o cercetare activă, dar care s-a dovedit extrem de dificil de realizat prin intermediul camerei video. Senzorul Kinect permite calculatorului să „simtă” direct a treia dimensiune (adâncimea) jucătorilor și a mediului, făcând sarcina mult mai ușoară. Dispozitivul înțelege, de asemenea, când utilizatorii vorbesc, știe cine sunt atunci când aceștia se apropie de el, și poate interpreta mișcările lor transformându-le într-un format pe care dezvoltatorii îl pot folosi pentru a construi noi experiențe.[31]

Senzorul Kinect încorporează mai multe elemente avansate de hardware pentru detectare (Fig. 12). Acesta conține un senzor de adâncime, o cameră color și o matrice de patru microfoane care permite captarea mișcării 3D, recunoașterea facială și recunoașterea vocală.[31]

Progresele în domeniul camerelor cu adâncime 3D, cum ar fi senzorii Microsoft Kinect, au creat numeroase oportunități pentru computere multimedia.



Fig. 12 - Microsoft Kinect - a). Senzor Kinect de la Xbox 360 b). proiector infraroșu, cameră infraroșu și cameră RGB în interiorul senzorului Kinect [31]

Colectivul din care am făcut parte a realizat o integrare a unui dispozitiv Kinect v2 cu un manipulator robotic virtual în [32]. Astfel în Fig. 13 se poate observa cum s-au controlat doi roboți virtuali UR10, prin intermediul acestui dispozitiv.

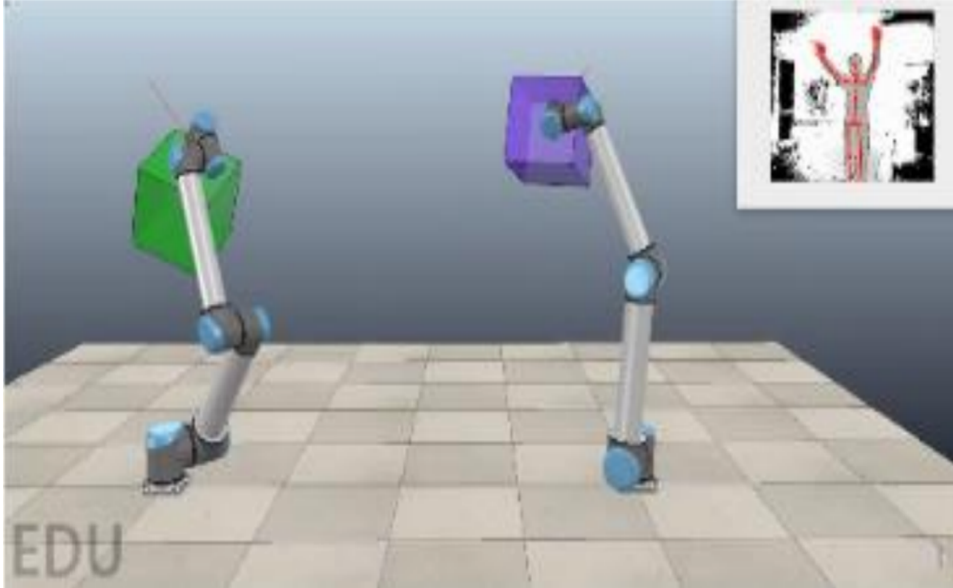


Fig. 13 - Manipulare cuburi cu ajutorul Microsoft Kinect v2

În [33] autorii au folosit un dispozitiv Microsoft Kinect cu ajutorul căruia au transmis comenzi către un robot industrial KUKA KR-125. Imaginile video au fost procesate prin intermediul unui calculator, iar cu ajutorul protocolului de comunicare ethernet, au transmis comenzi de poziționare și manipulare către robotul industrial. Cu ajutorul camerei video montate pe dispozitivul Microsoft Kinect, se pot detecta obiecte, numărul acestora precum și poziționarea lor (Fig. 14).

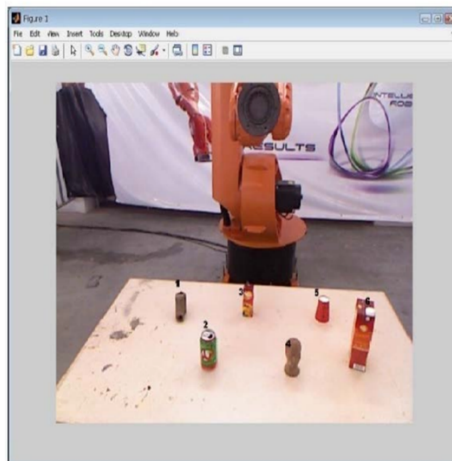


Fig. 14 - Secvență de numărare a obiectelor cu ajutorul Microsoft Kinect [33]

24 - STADIUL ACTUAL ÎN DOMENIUL CONDUCERII ROBOȚILOR INDUSTRIALI

Un alt exemplu de control al unui robot industrial cu ajutorul Microsoft Kinect este prezentat în [34] unde autorii au folosit un robot KUKA LBR iiwa 14 R820, dotat cu un efector final tip dispozitiv de prehensiune, cu 7 grade de libertate pentru a manipula obiecte cu ajutorul mișcărilor mâinii. (Fig. 15).



Fig. 15 - Control cu ajutorul Microsoft Kinect si KUKA LBR iiwa R820 [34]

Alte exemple de control al unui robot cu ajutorul Microsoft Kinect sunt prezentate în [35]–[48]. Unele folosesc recunoașterea gesturilor mâinii, altele sunt echipate cu dispozitive suplimentare, iar câteva implică factori de interferență, cum ar fi oameni și obiecte similare.

Schema principală de conectare, în majoritatea exemplelor, este prezentată în Fig. 16. Conectând dispozitivul la calculator, iar acesta la controler, se primesc informațiile, se procesează și se transmit ca mișcări către robotul propriu-zis.

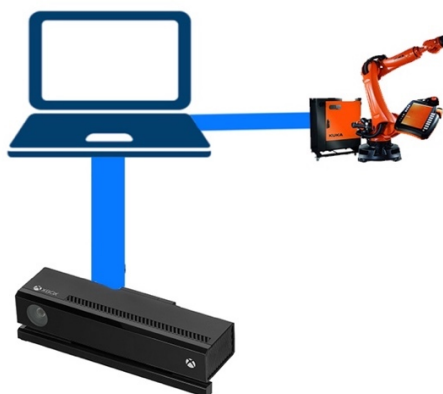


Fig. 16 - Schema de conectare Kinect - Robot Industrial

2.3. Telefonul mobil

Un alt sistem de control al roboților industriali este prin intermediul unui telefon mobil de tip smartphone.

În prezent, smartphone-urile și tabletele devin din ce în ce mai performante și cu caracteristici noi și utile care se pot integra cu roboții, pentru a dezvolta un sistem de control [9]. Utilizarea de smartphone-uri și tablete în dezvoltare și cercetare nu se întâmplă doar în cadrul sistemelor de control, ci în toate domeniile, deoarece acestea reprezintă o oportunitate de afaceri semnificativă pentru producători, care în mod constant dezvoltă sisteme de operare și hardware mai bune. [49]

Un exemplu foarte bun este prezentat în [50], unde cercetătorii de la Technische Universität Berlin, au realizat o aplicație pe un telefon mobil de tip iPhone care practic înlocuiește o consolă mobilă de învățare. Cu ajutorul telefonului mobil, orice utilizator poate modifica poziția robotului sau să realizeze operațiuni de programare.

Operatorul uman are practic două variante de a controla robotul:

a) Prin intermediul accelerometrului

Accelerometrul este un senzor care măsoară accelerația după trei direcții ortogonale. Primele dispozitive cu accelerometru au fost dezvoltate în anii 1980, când acestea erau tentante, fiind un exemplu al evoluției tehnologice însă la acea vreme nu erau folosite, dat fiind costul relativ mare pentru achiziția lor [51]. Astăzi accelerometrele sunt încorporate în toate telefoanele mobile de tip smartphone, cât și în aproape toate dispozitivele inteligente.

În [50], telefonul mobil folosește datele accelerometrului și le transformă în mișcări către robot. Acesta se mișcă de-a lungul axelor X, Y și Z separat (translație). Rotația pe toate cele trei axe se face separat, pe fiecare axă în parte. Utilizatorul trebuie să rotească telefonul în stânga sau dreapta pentru a mișca robotul pe axa X, în față sau spate pentru a mișca robotul pe axa Y, iar folosind opțiunile de pe aplicație poate mișca robotul pe axa Z. (Fig. 17)

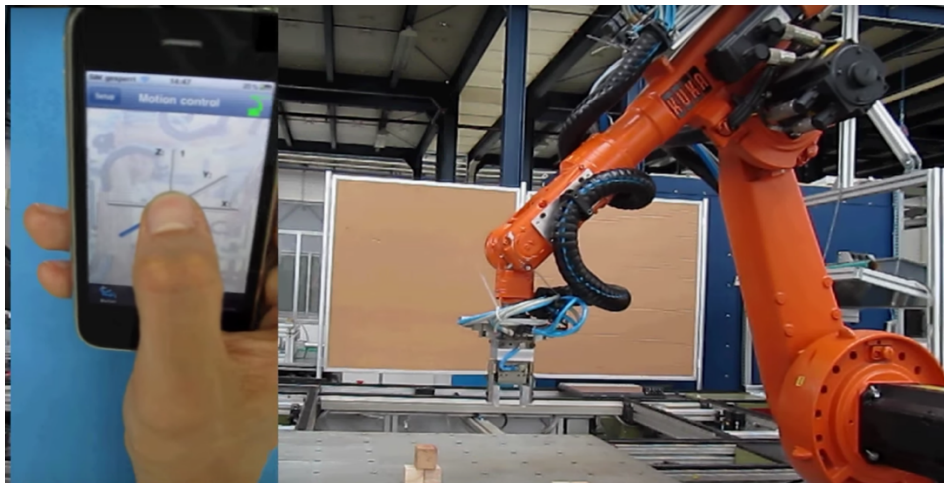


Fig. 17 - Robot industrial controlat cu accelerometrul telefonului mobil [50]

26 - STADIUL ACTUAL ÎN DOMENIUL CONDUCERII ROBOȚILOR INDUSTRIALI

Un alt exemplu de control al unui robot industrial cu ajutorul accelerometrului este prezentat în [52] însă în această situație nu este folosit un telefon mobil, ci direct un accelerometru montat pe mâna unui utilizator.

b) Prin intermediul ecranului tactil

În [50], în cadrul aplicației realizate pe telefonul mobil, utilizatorul are opțiunea de a mișca robotul pe fiecare axă în parte. Au fost create butoane pentru fiecare axă, iar utilizatorul poate incrementa sau decrementa poziția robotului, printr-o simplă apăsare a unui buton. Tot prin intermediul acestei aplicații utilizatorul poate mări viteza robotului sau poate închide sau deschide dispozitivul de prehensiune atașat robotului industrial. (Fig. 18)

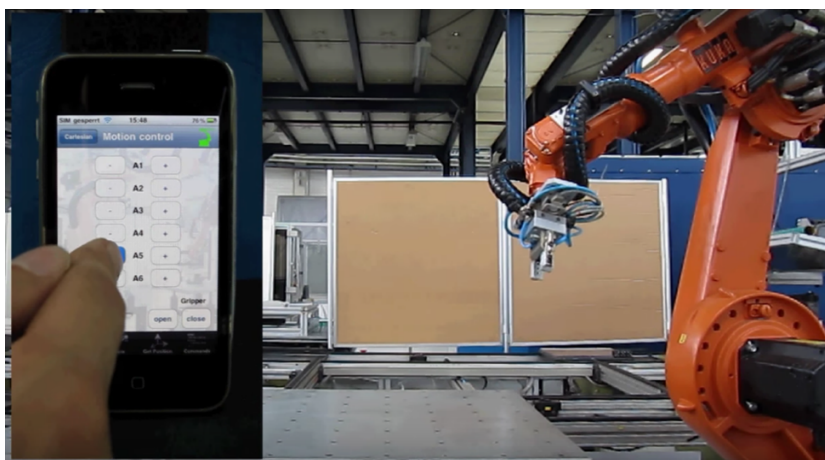


Fig. 18 - Robot industrial controlat prin intermediul ecranului tactil al telefonului mobil [50]

Alte aplicații de teleoperare a roboților industriali cu ajutorul telefonului mobil, prin intermediul unei aplicații mobile se pot observa în [53]–[58].

2.4. Joystick

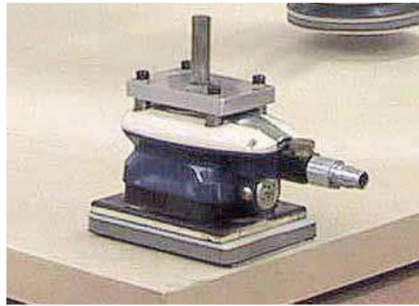
O altă metodă de a controla robotul industrial este prin folosirea unui dispozitiv de tip Joystick, care pivotează și trimite unghiul lui către calculator. Acesta este un dispozitiv periferic al calculatorului.

În general aceste dispozitive au anexate și mai multe butoane care pot trimite diverse informații către calculator. Joystick-ul se folosește uzual în jocurile pe calculator însă asemenea dispozitive se folosesc și în aviație pentru controlul aparatelor de zbor (avion, elicopter, etc.) având același principiu prin care mișcarea 3D se procesează și se transformă în comenzi de mișcare.

În [59] și [60] autorii au realizat un sistem prin care un robot folosit pentru a realiza acțiuni de poliș este controlat cu ajutorul unui joystick, astfel cum se poate observa în Fig. 19, implementând diverse modele matematice. Aceștia au concluzionat, în urma experimentelor, faptul că un operator poate realiza operații de polișare cu ajutorul joystick-ului.



Robot de polișare



Dispozitiv de polișare



Joystick

Fig. 19 - Sistem de folosire a robotului de polișare cu ajutorul unui joystick [59]

Alte aplicații de control a unor sisteme robotice cu ajutorul Joystick-ului se pot observa în [61]–[64].

2.5. Gamepad

Roboții industriali pot fi conduși și prin intermediul unui gamepad, o telecomandă folosită în general în cadrul jocurilor pe televizor. Interconectând gamepad-ul cu robotul industrial, acesta poate efectua mișcări de rotație sau translație, folosind butoanele și joystick-ul încorporat în gamepad.

În [65] autorii au creat un set de ustensile foarte utile în vederea interfațării cu KUKA Sunrise.OS care ajută la dezvoltarea și implementarea de aplicații robot. Folosind această bibliotecă de ustensile, autorii au realizat o aplicație prin care controlează un robot industrial cu ajutorul unui gamepad, după cum se poate observa în Fig. 20.

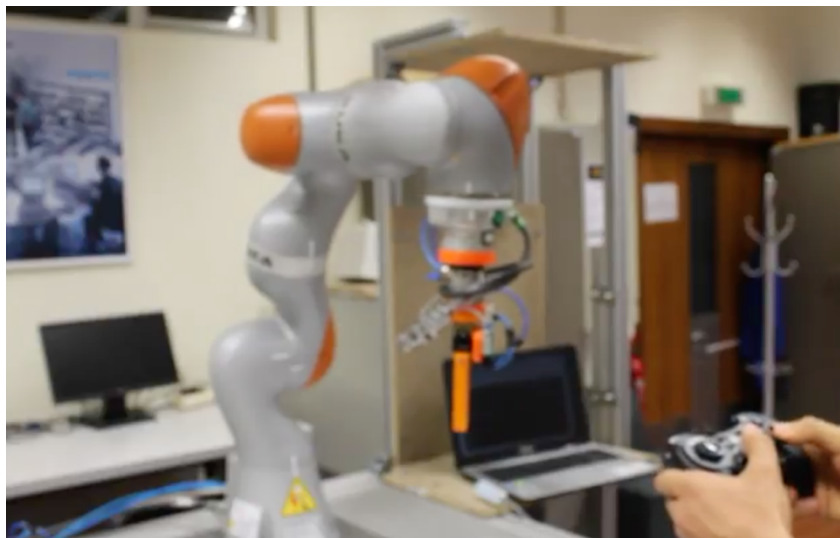


Fig. 20 - Robot industrial controlat prin intermediul unui gamepad [65], [66]

Alte aplicații de control ale roboților industriali cu un Gamepad sunt prezentate în [67]–[69].

2.6. Leap Motion

Leap Motion este un dispozitiv de recunoaștere a gesturilor, însă față de Kinect, acesta se concentrează pe gest și nu pe mișcarea corpului. Mișcările identificate sunt de dimensiuni reduse, dar au o precizie ridicată. Acest dispozitiv are două camere în interior care pot realiza imagini la 120Hz pentru a obține informații despre cum se modifică poziția mâinii în spațiul 3D, așa cum se poate observa în. Fig. 21. [70]

Utilizări ale Leap Motion pentru a teleopera roboți se pot observa în [71]–[84]

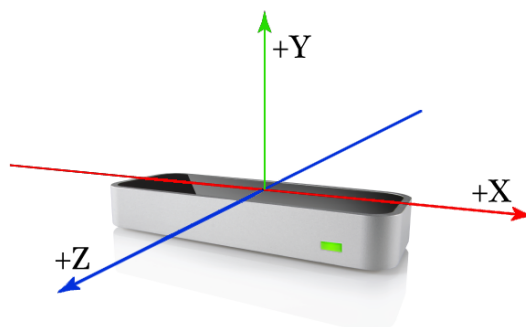


Fig. 21 – Leap Motion [70]

2.7. Myo Armband

Brățara Myo (Fig. 22) este un dispozitiv dezvoltat de compania Thalmic Labs, dispozitiv care se poate purta pe mână (poziționat pe antebraț) pentru a interacționa cu sistemele. Ea este echipată cu mai mulți senzori care recunosc gesturile palmei sau mișcarea mâinii. Aceasta este caracterizată de utilizarea unui proces numit electromiografie (EMG), identificând gesturile în funcție de activitatea musculară din mână. [85]

Aceasta a fost folosită pentru diverse aplicații de control ale roboților industriali precum:

1. În [86] autorii au reușit să controleze un robot KUKA iiwa cu ajutorul a două dispozitive Myo Armband;
2. [87] ne prezintă o soluție de control al unui KUKA youBot cu ajutorul Myo Armband;
3. Predicția forței de aplicare a unui robot KUKA iiwa a fost realizată cu ajutorul Myo Armband în [88];
4. Autorii din [89] ne prezintă cum au reușit controlul unui Universal Robots UR5e cu ajutorul Myo Armband și ochelari VR;
5. În [90] autorii au reușit să controleze o interfață robot cu ajutorul Myo Armband.

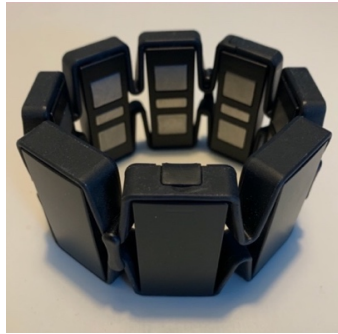


Fig. 22 - Myo Armband

3. SOLUȚIA PROPUȘĂ ȘI ETAPELE CERCETĂRII

3.1. Dispozitive utilizate

3.1.1. Robotul industrial KUKA KR15

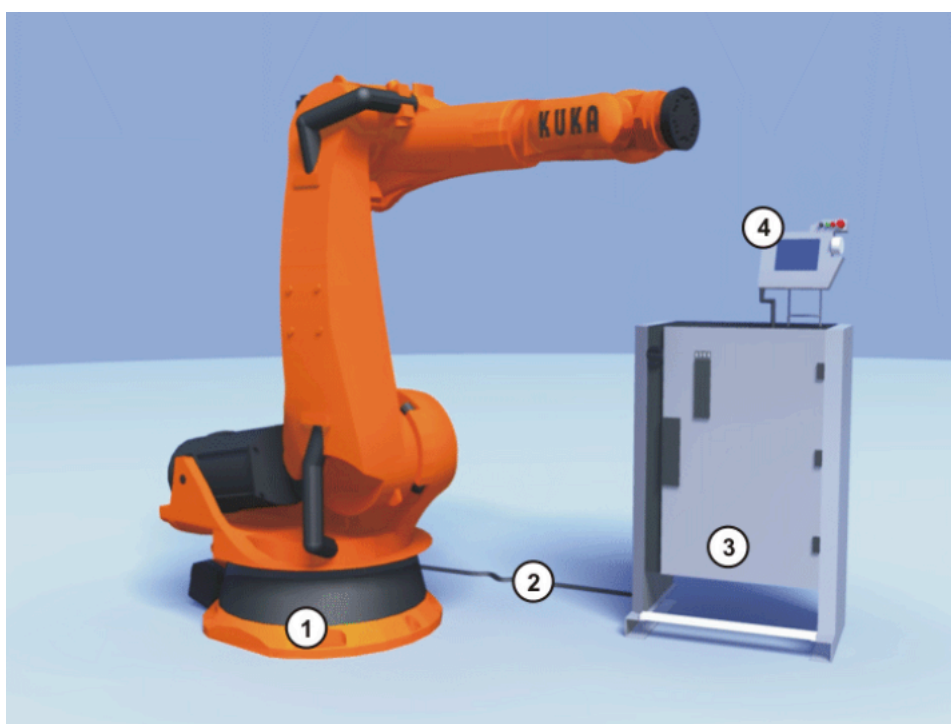


Fig. 23 - Exemplu de sistem robot KUKA [91]

În Fig. 23 este prezentat un sistem robot, care este format din următoarele componente:

- 1- Robot;
- 2- Cabluri de conectare;
- 3- Controler robot;
- 4- Consolă de învățare.

Robotul industrial KUKA KR15, existent în cadrul Departamentului de Mecatronică al UPT, este o versiune educațională cu funcții și posibilități limitate. Acesta are o structură din oțel inoxidabil și poate fi folosit în domenii diverse începând de la manipulare până la vopsire sau lipire.

Caracteristici Robot Industrial KUKA KR15: [91]

Axe: 6
 Sarcină utilă: 15.00kg
 Înălțime: 1570.00mm
 Repetabilitatea: ± 0.1000 mm
 Masa robotului: 315.00kg
 Structură: Articulat
 Montare: Podea, inversat

Axa	Viteză de deplasare	Spațiul de lucru
A1	152 °/s (2.65 rad/s)	$\pm 185^\circ$
A2	152 °/s (2.65 rad/s)	$+115^\circ - 55^\circ$
A3	152 °/s (2.65 rad/s)	$+70^\circ - 210^\circ$
A4	284 °/s (4.96 rad/s)	$\pm 350^\circ$
A5	293 °/s (5.11 rad/s)	$\pm 135^\circ$
A6	604 °/s (10.54 rad/s)	$\pm 350^\circ$

Roboții KUKA sunt folosiți în diverse aplicații precum industria aerospațială, un domeniu complex în care roboții vin să simplifice și să automatizeze procesul de producție al aeronavelor [92], domeniul automotive care s-a schimbat complet de când a început să fie automatizat [93], domeniul producției de componente electronice, care este într-un proces continuu de dezvoltare și optimizare [94], domeniul alimentar unde este necesară realizarea unor efectoare finale speciale pentru manipularea alimentelor [95], domeniul injecției de mase plastice [96] sau domeniul industriei de prelucrare a metalelor [97].

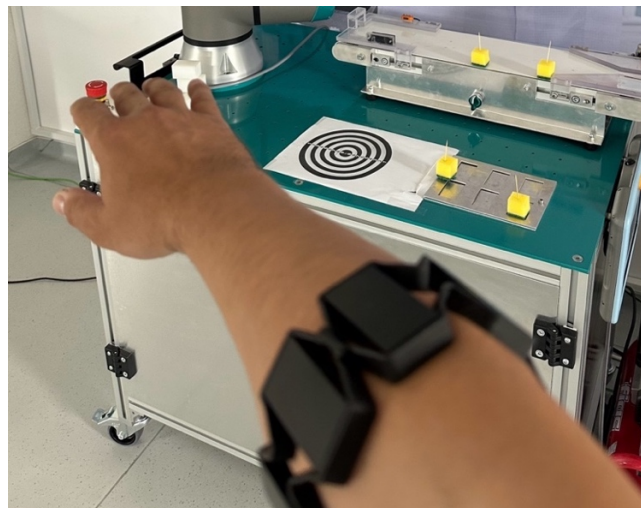
3.1.2. Brățara Myo Armband

Fig. 24 - Brățara Myo Armband

Brățara Myo (Fig. 24) realizată de compania Thalmic Labs, ajută la controlarea prin gesturi a diferitelor obiecte. Brățara se poziționează pe antebraț, asemenea unui tensiometru, și poate controla obiecte prin mișcarea degetelor sau a încheieturii.

Brățara este realizată dintr-o serie de bucăți dreptunghiulare de plastic, legate împreună printr-un suport de cauciuc. Sub fiecare bucată dreptunghiulară se găsesc trei senzori electromiografici din oțel inoxidabil care detectează impulsul electric din mușchii de la antebraț, prin identificarea diferenței de potențial. Brățara se poate ajusta în funcție de mână purtătorului prin intermediul unor elemente de strângere. Pe un element din brățară, există două indicatoare cu LED. Unul este logoul celor de la Thalmic Labs, iar celălalt este o fantă de lumină dreptunghiulară care atunci când devine verde, indică faptul că brățara se încarcă iar când este albastru indică faptul că este conectată, așa cum se poate observa în Fig. 25.

Din punct de vedere dimensional, brățara Myo are o circumferință de 19,5 cm care se poate extinde până la 33 cm. Brățara cântărește doar 93,55 grame, iar fiecare element dreptunghiular de pe brățară are 1,14 cm lățime. Brățara se poate achiziționa în culorile negru sau alb.[98]

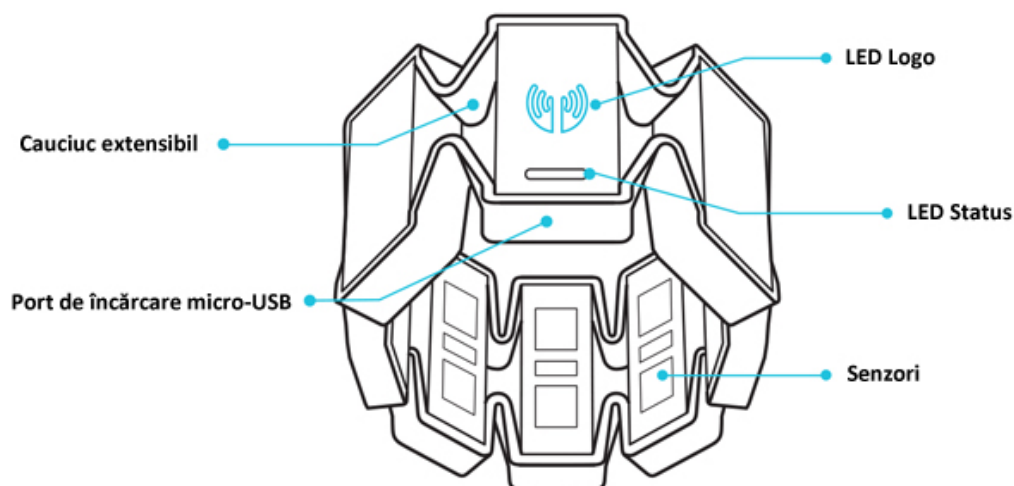


Fig. 25 - Componente Myo Armband[66]

Brățara este livrată cu un cablu micro USB pentru încărcarea bateriilor și un adaptor USB Bluetooth pentru conectarea la calculator.

Brățara folosește un accelerometru, un magnetometru și un giroscop, pe trei axe, pentru a detecta mișcarea. Aceasta este dotată cu un procesor ARM Cortex M4 care comunică cu dispozitivele conectate prin bluetooth. Myo este compatibilă cu PC-urile care rulează Windows 7, Windows 8, și Mac OS X 10.8 sau versiune superioară.

Conform celor de Thalmic Labs, bateria reîncărcabilă lithium-ion poate păstra Myo funcțional pentru o zi întreagă cu o singură încărcare. Cei de la Thalmic Labs au dezvoltat o platformă pentru aplicații Myo. Astfel, o incursiune prin Myo App Market ne arată că brățara este foarte utilă în multe domenii. Printre aplicații putem remarca cele compatibile cu iTunes, Windows Media Player, Arduino, și cu jocuri precum Counter-Strike, Civilization 5 sau Tetris. Brățara se poate conecta la orice controler

prin intermediul unui calculator și a soft-urilor speciale venite din partea firmei producătoare.[98]

După conectare la calculator, datele brățării Myo se pot vedea pe pagina de <http://diagnostics.myo.com/>, iar interfața se poate observa în Fig. 26.

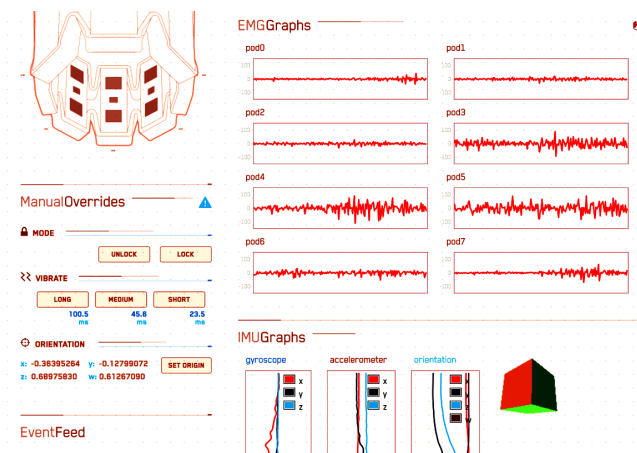


Fig. 26 - Pagina MyoDiagnostics

Gesturile posibile a fi realizate cu Myo Armband sunt prezentate în Fig. 27.

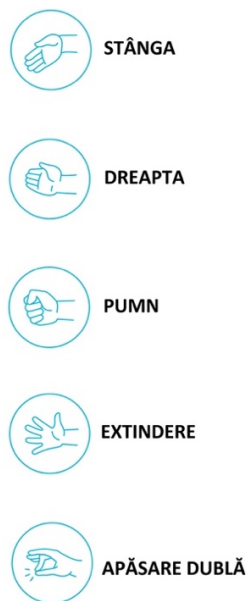


Fig. 27 - Gesturi posibile Myo Armband

3.1.3. Universal Robots UR10e

Universal Robots este o companie care realizează roboți colaborativi, numiți și coboți. Coboții sunt folosiți în special pentru a veni în sprijinul oamenilor implicați în diverse procese industriale.[99]

Unul dintre cele mai utilizate modele și cele mai bine vândute de către Universal Robots, este modelul UR10e (Fig. 28). Acest model ocupă doar un diametru de Ø190 mm la sol, poate atinge distanțe de până la 1300mm, cu o sarcină utilă de maxim 12,5 kg, cântărind doar 33,5kg.[100]



Fig. 28 - Universal Robots UR10e

3.2. Soluția propusă

Soluția propusă prezintă realizarea unui sistem de control a unui robot industrial, cu ajutorul Myo Armband însă, ca element de noutate față de stadiul actual ar fi faptul că, tot prin gesturi, utilizatorul are posibilitatea să salveze poziții și să genereze un program care mai apoi poate fi rulat direct de robot. Practic, cu ajutorul Myo Armband, orice om, fără niciun nivel de cunoștințe de programare, să aibă capacitatea să deplaseze un robot industrial și să creeze un traseu de deplasare al robotului.

Cu ajutorul acestui sistem s-ar deschide orizonturile de utilizare a roboților industriali dincolo de bariera tehnică necesară programării lor.

Soluția ar presupune o corelare a mișcărilor brațului omului, cu deplasarea unui robot industrial pe coordonate carteziane, iar cu ajutorul gestului pumn, un efector final de tip dispozitiv de prehensiune să se închidă, iar cu ajutorul gestului degete extinse, să deschidă dispozitivul de prehensiune.

3.3. Etapele cercetării

Prima etapă ar fi testarea integrării brății Myo Armband cu alte dispozitive și testarea facilităților acestora. După ce s-a reușit acest lucru cu succes și s-a înțeles funcționarea dispozitivului Myo Armband, următoarea etapă ar fi controlul unor roboți industriali. Activitatea de cercetare doctorală a început în laboratorul KUKA din cadrul Departamentului de Mecatronică, unde se află doi roboți industriali. În cadrul acestor activități de cercetare s-a testat conectarea brății Myo Armband cu robotul KUKA KR15 (Fig. 29) și controlul acestuia, în primă fază, doar pe coordonate carteziene, pe axele x , y , z , iar ulterior s-a testat și controlul orientării robotului R_x , R_y , R_z , cu o a doua brătară Myo.



Fig. 29 – Robot KUKA KR15



Fig. 30 – Laborator DRÄXLMAIER

Ulterior, prin inițiativă personală, am reușit obținerea unei finanțări din partea companiei DRÄXLMAIER, realizând un nou laborator de Robotică (Fig. 30), care are în dotare atât un robot UR10e de la Universal Robots, 16 sisteme desktop all-in-one conectate la robotul UR10e, cât și o tablă inteligentă Samsung Flip 3 pe care se poate viziona ecranul consolei de învățare atașată robotului. Activitatea de cercetare a vizat controlarea robotului UR10e cu brățara Myo. Ultima etapă a constat în analiza felului în care se comportă cei doi roboți, KUKA KR15 și Universal Robots UR10e la controlul acestora cu Myo Armband și demararea încercărilor experimentale.

3.4. Obiectivele cercetării

Obiectivele acestei teze sunt:

- O1. Realizarea unui sistem de control a unui robot industrial cu brățara Myo Armband, care să permită mișcări fără șocuri;
- O2. Realizarea de teste prin care se poate măsura timpul și precizia, pentru a compara utilitatea controlului robotului, atât cu Myo Armband, cât și cu consola de învățare;
- O3. Colectarea opiniilor utilizatorilor prin aplicarea de chestionare;
- O4. Analiza măsurătorilor realizate în cadrul testelor, precum și a răspunsurilor date în chestionare, pentru a determina dispozitivul mai bun.

4. EVALUAREA SOLUȚIEI OPTIME

4.1. Aplicație conexiune Myo Armband cu Arduino

Pentru a testa capacitățile brățării Myo Armband, am dezvoltat o primă aplicație, controlând un robot mobil pe șenile. Scopul aplicației a fost conducerea robotului mobil pe șenile exclusiv cu ajutorul brățării Myo.

4.1.1. Arhitectura sistemului

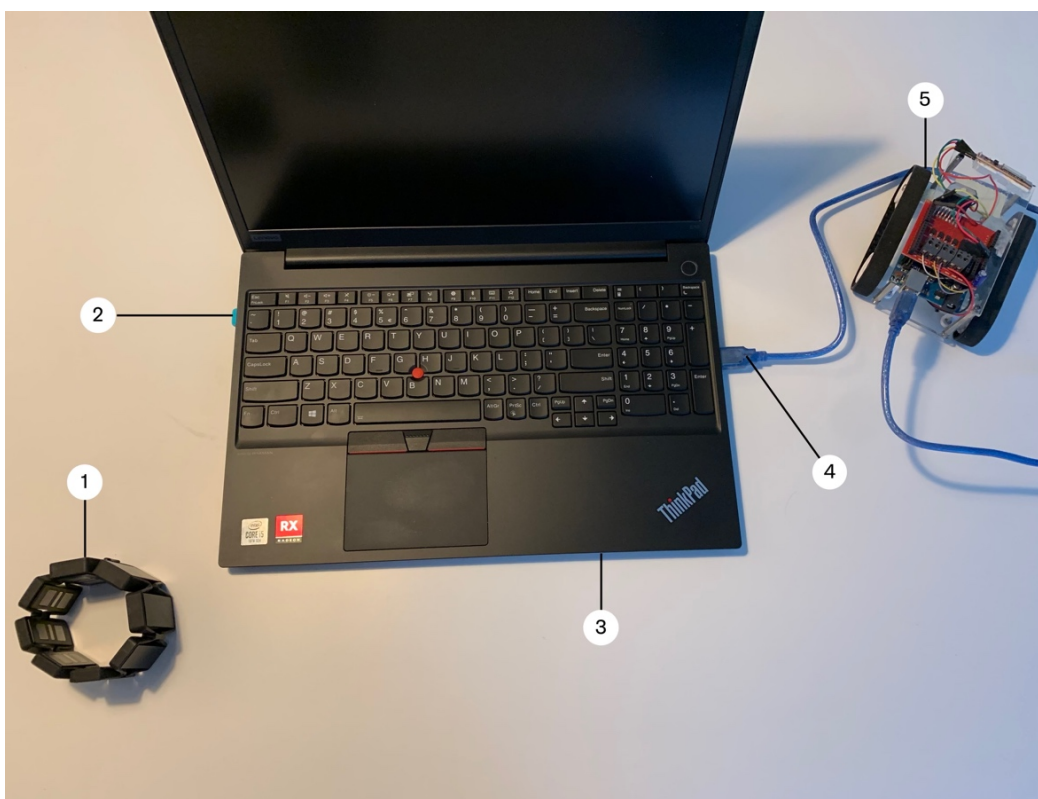


Fig. 31 - Arhitectură sistem de conectare Myo Armband cu Arduino

Arhitectura sistemului Myo Armband-Arduino (Fig. 31) este compusă din :

- 1- Dispozitiv comunicare bluetooth;
- 2- Computer portabil;
- 3- Cablu serial;
- 4- Robot mobil cu șenile.

- Robotul mobil cu șenile este format din:
- Structură cu șenile;
 - Arduino Uno;
 - Driver motor L298, tip shield.

- Structură cu șenile

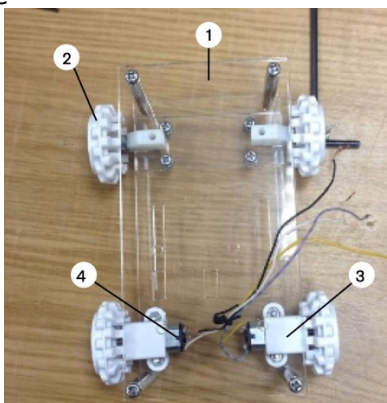


Fig. 32 - Structură cu șenile

În Fig. 32 este prezentată structura robotului care este compusă din:

- trei bucăți plexiglas transparent;
- set roți cu șenile Pololu 30T;
- suport de fixare și prindere micro motoare;
- două micro motoare DC 50:1.

- Arduino Uno

Arduino este o placă de dezvoltare care prin intermediul componentelor hardware și software poate trimite și citi semnale de la medii și dispozitive externe. Placa Arduino pune la dispoziție mai mulți pini pentru intrare sau ieșire prin intermediul cărora pot fi construite legături cu diverse componente (motoare, becuri, senzori, etc.).

Placa de dezvoltare Arduino are mai multe variante, în funcție de opțiuni și facilități: Uno, Mega, Diecimila, Duemilanove, Mini, Nano și chiar Bluetooth Arduino. Cele mai folosite produse sunt Arduino UNO (Fig. 33) și Arduino MEGA 2560.



Fig. 33 - Arduino Uno

c. Driverul motor L298, tip shield

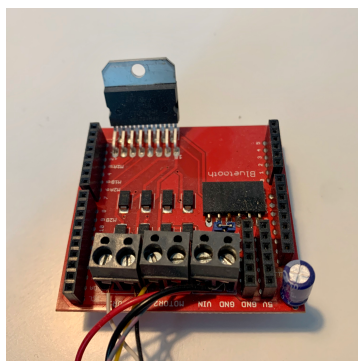


Fig. 34 - Driver motor L298, tip shield

Driver-ul de motoare L298, tip shield (Fig. 34) este un element utilizat de plăcile Arduino pentru a controla motoarele de curent continuu. Placa Arduino nu poate oferi la porturile sale o putere foarte mare, fiind insuficientă pentru a putea roti un motor. Acesta are în componență și o punte H care permite controlul direcției. În limbajul de specialitate, acest amplificator se numește „driver de motoare”.

4.1.2. Schema de conectare

Brățara Myo este conectată la computerul portabil prin dispozitivul bluetooth care facilitează transmiterea datelor. Computerul portabil procesează informațiile și le transmite către placa Arduino, conform Fig. 35.

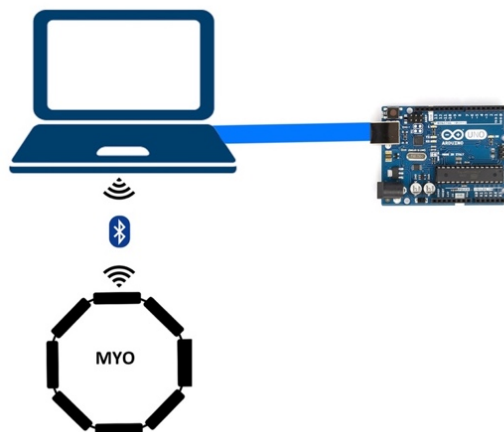


Fig. 35 - Schema de conectare Myo Armband cu Arduino

Placa Arduino citește informațiile primite de la calculatorul portabil, le interpretează și în funcție de informațiile primite, transmite comenzi către driverul de motoare care alimentează motoarele pentru a realiza deplasarea dorită.

Conectarea elementelor se realizează conform schemei din Fig. 36.

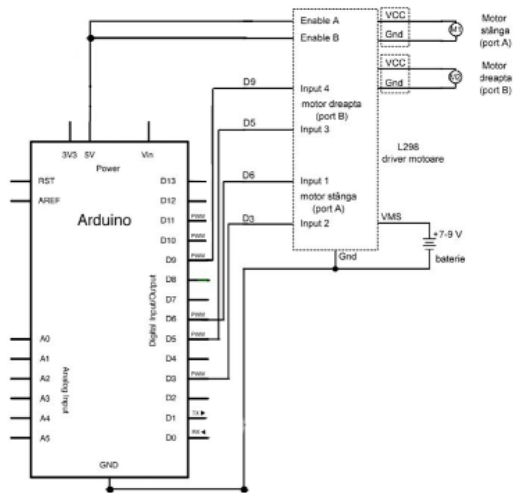


Fig. 36 - Schema de conectare Arduino - Driver motoare L298

4.1.3. Încercări și rezultate experimentale

Pentru a mișca robotul mobil se vor folosi gesturi pe care brățara Myo le va identifica și, în funcție de gestul realizat, va trimite date către robotul mobil care va realiza o mișcare specifică (Fig. 37):

1. Mișcare palmă spre stânga – Pivotare robot spre stânga;
2. Mișcare palmă spre dreapta – Pivotare robot spre dreapta;
3. Strângere pumn – Mers în față;
4. Extindere degete – Mers înapoi;
5. Apăsare dublă cu degetul mijlociu și cel mare - Robotul se oprește.

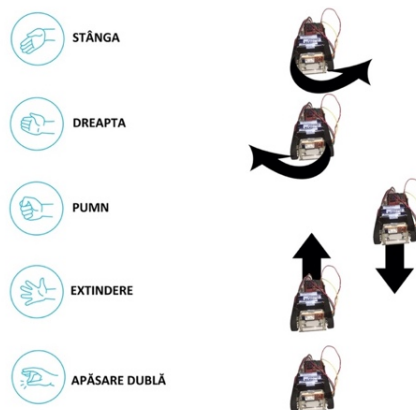


Fig. 37 - Corelare între gesturile și mișcările ale robotului

Schema bloc a programului Arduino este prezentată în Fig. 38.

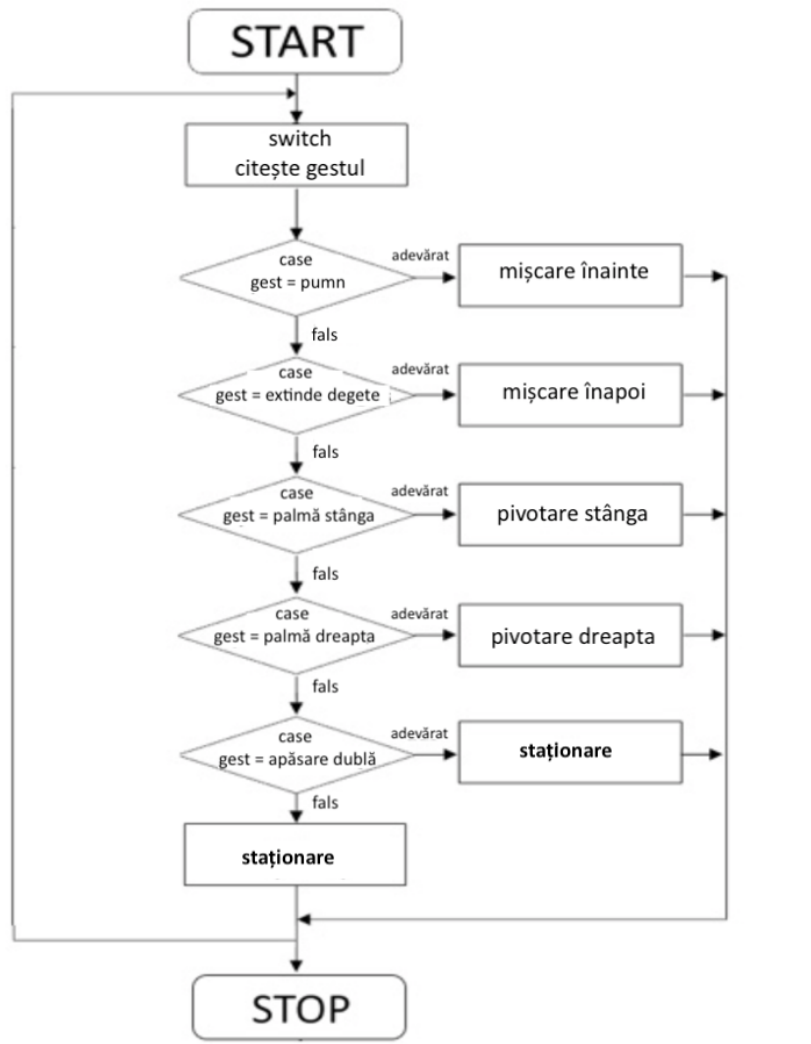


Fig. 38 - Schema bloc program Arduino

4.1.4. Concluzii

Acest test a arătat capacitățile brățării Myo precum și faptul că aceasta este un instrument foarte bun pentru controlul roboților mobili, întrucât are o comunicare foarte bună și rapidă. Brățara răspunde foarte bine la mișcări, însă este esențial ca aceasta să fie calibrată înainte de folosire. Activitatea de cercetare s-a materializat printr-o lucrare științifică publicată în [101].

4.2. Aplicație conexiune Myo cu KUKA KR15

Primul experiment de conectare a brățării Myo cu un robot industrial KUKA KR15 (Fig. 39) a început prin analiza robotului aflat în dotarea laboratorului. Astfel, acesta este o versiune educațională din care lipsesc anumite module. Problema majoră depistată a fost că singura variantă de comunicare a fost găsită prin intermediul unui port serial.

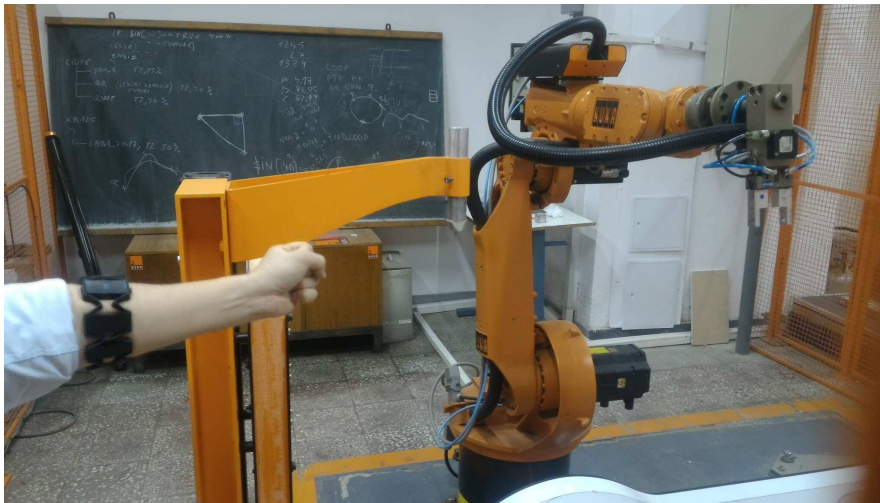


Fig. 39 - Aplicație conectare brățara Myo - KUKA KR15

4.2.1. Arhitectura sistemului

În Fig. 40 se pot observa principalele componente ale sistemului:

- 1- Brățara Myo;
- 2- Dispozitiv comunicare bluetooth
- 3- Cablu serial
- 4- Robot industrial KUKA KR15.

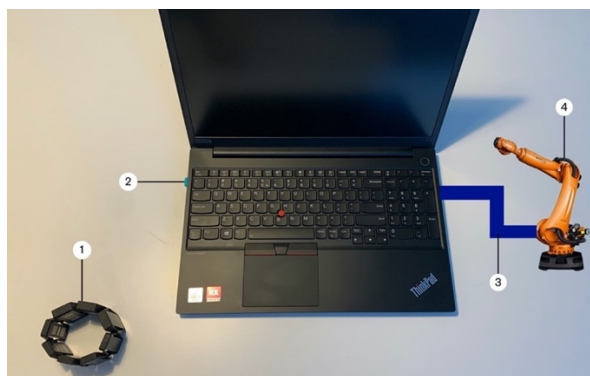


Fig. 40 - Componente conectare sistem Myo - KUKA KR15

4.2.2. Schema de conectare

Brățara Myo este conectată la un computer portabil, prin intermediul dispozitivului bluetooth, iar computerul este conectat la rândul său la robotul industrial KUKA KR15, prin intermediul unui cablu serial (Fig. 41)



Fig. 41 - Schema de conectare Myo – KUKA KR15

4.2.3. Încercări și rezultate experimentale

Brățara Myo trimite date wireless prin dispozitivului bluetooth către computerul portabil, care recepționează informația, o prelucrează folosind un program C#, iar apoi transmite datele de poziționare către controlerul KUKA KR15.

Controlerul KUKA KR15 rulează, de asemenea, un program care „ascultă” după mesaje transmise prin portul RS232. Când o informație validă este recepționată, controlerul comandă executarea unei mișcări specifice.

a. Program Myo

Pentru a citi, interpreta gesturi, cât și pentru a transmite informații, s-a realizat un program în limbajul de programare C# folosind Microsoft Visual Studio.

Visual Studio este un IDE (Integrated Development Environment) creat de Microsoft, prin care se pot realiza programe pe calculator, pagini web, aplicații web sau aplicații mobile. Programul are atât un editor de cod, cât și o interfață vizuală care permite crearea de interfețe grafice, unde utilizatorii pot interacționa. Visual Studio suportă 36 de limbaje de programare diferite, având și un modul de verificare și evidențiere a erorilor de cod.

Așadar, în vederea realizării programului care realizează conexiunea dintre brățara Myo și robotul industrial KUKA KR15, s-a creat un program în Visual Studio, în limbajul de programare C#.

Producătorii brățării Myo, Thalmic Labs, au pus la dispoziție tuturor utilizatorilor atât o bibliotecă de programe, cât și un soft special numit Myo Connect, care realizează conexiunea dintre brățara Myo și sistemul de operare al utilizatorului. Așadar s-a instalat Myo Connect pe sistemul de operare Windows, conform indicațiilor din Fig. 42.

Aplicație conexiune Myo cu KUKA KR15 - 43

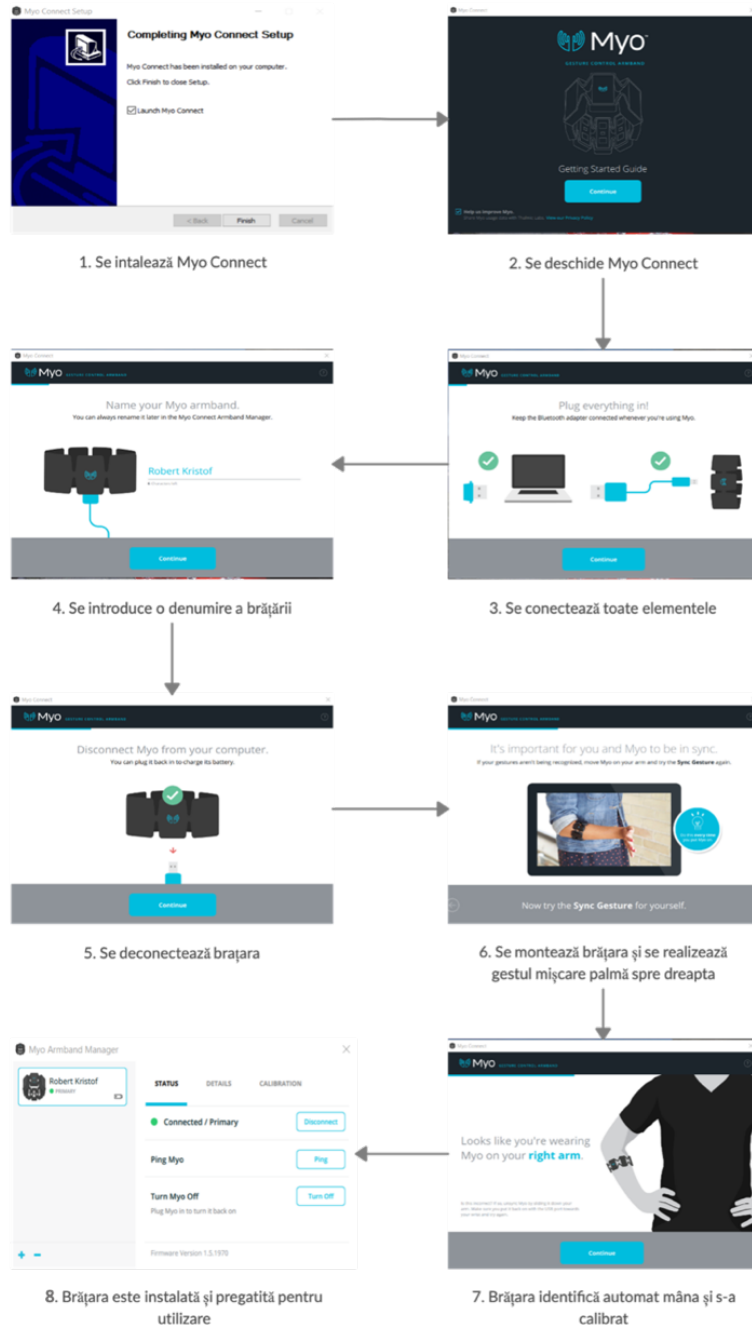


Fig. 42 - Pași pentru instalare și conectare brășara Myo

Schema bloc de funcționare a programului Myo în C# este prezentată în Fig. 43.

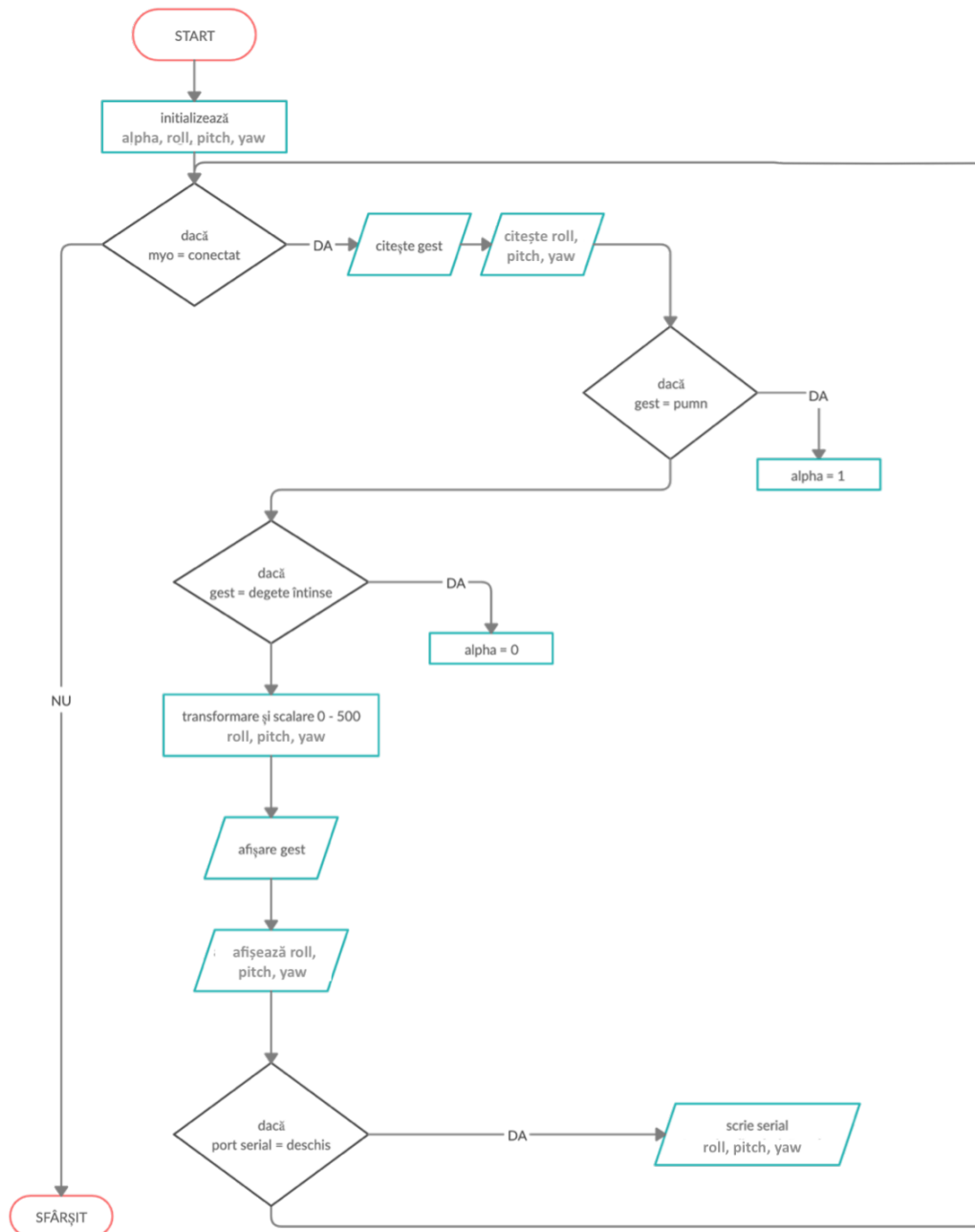


Fig. 43 - Schema bloc de funcționare a programului Myo în Visual Studio

Programul C# în Visual Studio are două componente:

1. Interfață vizuală
2. Cod program C#

1. Interfață vizuală

În interfață Design din cadrul Visual Studio s-a creat o fereastră prin care utilizatorul poate vizualiza datele, precum și deschide sau închide porturile seriale.

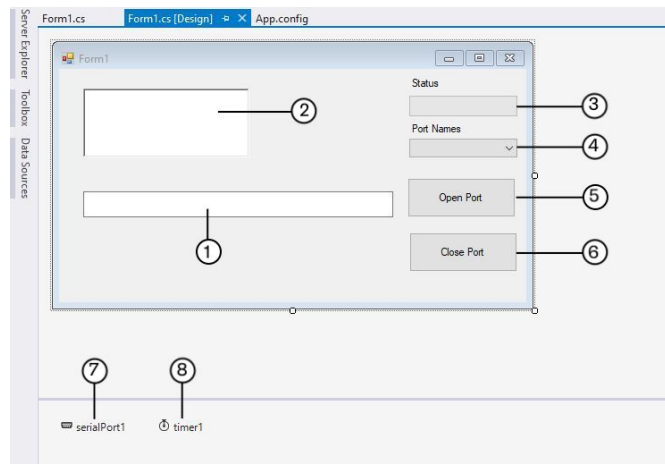


Fig. 44 - Elementele interfeței vizuale C# Visual Studio

Elementele din interfața vizuală (Fig. 44) sunt:

- 1- text;
- 2- text multiplu;
- 3- bară de progres;
- 4- listă;
- 5- buton 1;
- 6- buton 2;
- 7- port serial;
- 8- temporizator.

a. Cod program C#

Se includ toate bibliotecile care oferă toate funcțiile din Visual Studio. MyoSharp este un API dezvoltat de doi studenți de la MIT, care permite folosirea limbajului de programare C# în relația cu brățara Myo.[102]

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.IO.Ports;
using System.Drawing;
using System.Linq;
```

```

using System.Threading.Tasks;
using System.Windows.Forms;
using System.Threading;
using System.Text;
using MyoSharp.Device;
using MyoSharp.Communication;
using MyoSharp.Exceptions;
using MyoSharp.Poses;

```

Se inițializează variabilele care vor fi folosite în program. Variabila „data” de tip string este folosită pentru a integra toate datele care vor fi transmise prin portul serial. „roll”, „pitch”, „yaw” sunt variabilele folosite pentru calculul valorilor mișcărilor de orientare (pronație-supinație, flexie-extensie, aducție-abducție). Întrucât valorile trimise au fost foarte apropiate și multe dintre ele nu ar fi produs nicio mișcare, ci mai mult o eroare, s-au inițializat variabilele „roll_v”, „pitch_v”, „yaw_v”, „filtru”, „filtruv” care sunt folosite pentru aplicarea unui filtru. Variabila logică „prima” este folosită pentru verificarea momentului în care intră al doilea șir de valori pentru „yaw”, „pitch”, „roll” și se poate aplica filtrul. „offsetx”, „offsety” și „offsetz” sunt variabile care permit afișarea de valori ușor de înțeles pentru „roll”, „pitch”, „yaw”. „Alpha” este variabila folosită pentru a afla dacă s-a realizat gestul pumn sau degete extinse.

```

string data;
double yaw;
double pitch;
double roll;
double yaw_v=0;
double pitch_v=0;
double roll_v=0;
bool prima = false;
double offsetx;
double offsety;
double offsetz;
double filtru;
double filtruv;
double alpha;

```

Atunci când soft-ul este rulat, primul subprogram care se încarcă este *Form1_Load* care inițializează brățara Myo.

```

private void Form1_Load(object sender, EventArgs e)
{
    canalMyo =
    Channel.Create(ChannelDriver.Create(ChannelBridge.Create(),
    MyoErrorHandlerDriver.Create(MyoErrorHandlerBridge.Create())));
    HubulMyo = Hub.Create(canalMyo);
    HubulMyo.MyoConnected += HubulMyo_MyoConnected;
}

```

```

HubulMyo.MyoDisconnected += HubulMyo_MyoDisconnected;
canalMyo.StartListening();
}

```

Subprogramul *HubulMyo_MyoConnected* se activează atunci când este detectată brățara Myo. Imediat după conectare se va afișa mesajul „Myo conectat”, iar brățara va vibra. Variabila „pose” stochează tipul gestului realizat de către utilizator, intervalul de citire fiind 0,5 secunde. *Myo_OrientationDataAcquired* returnează valorile de orientare ale brățării Myo.

```

private void HubulMyo_MyoConnected(object sender, MyoEventArgs
e)
{
    MessageBox.Show("Myo conectat");
    e.Myo.Vibrate(VibrationType.Long);
    e.Myo.Unlock(UnlockType.Hold);
    var pose = HeldPose.Create(e.Myo, Pose.FingersSpread,
Pose.DoubleTap, Pose.Fist, Pose.WaveIn, Pose.WaveOut);
    //pauza între gesturi
    pose.Interval = TimeSpan.FromSeconds(0.5);
    pose.Start();
    pose.Triggered += Pose_Triggered;

    e.Myo.OrientationDataAcquired +=
Myo_OrientationDataAcquired;
}

```

Subprogramul *Myo_OrientationDataAcquired* preia valorile de orientare și calculează valorile „roll”, „pitch”, „yaw”. Folosind variabila logică „prima” se poate determina prima iterație, se calculează valorile inițiale ale lui „roll”, „pitch”, „yaw”, care mai apoi se stochează în variabilele „roll_v”, „pitch_v”, „yaw_v”, astfel tot timpul se vor cunoaște valorile curente precum și cele precedente. Pentru primul set de valori se aplică doar o variabilă care calibrează datele ce vor fi afișate și totodată determină spațiul de lucru pentru robotul KUKA. Al doilea șir de valori se va calibra prin intermediul unor variabile de filtrare care au menirea să reducă pe cât posibil valorile foarte apropiate, care nu determină mișcări vizibile ale robotului. La finalul subprogramului, variabilele numerice sunt convertite într-un șir de caractere în variabila „data”.

```

void Myo_OrientationDataAcquired(object sender,
OrientationDataEventArgs e)
{
    const float PI = (float)System.Math.PI;

    offset = 300;
    offsetz = 350;
    filtru = 0.05;
    filtruv = 0.95;

    if (prima == false)
    {
        roll = ((e.Roll + PI) / (PI * 2.0f) * -offset) + offset/2;
        pitch = ((e.Pitch + PI) / (PI * 2.0f+ 0.00000001) * offsetz) -
offsetz/2;
        yaw = (e.Yaw + PI) / (PI * 2.0f) * -offset + offset/2 + offsety;

        roll_v = roll;
        pitch_v = pitch;
        yaw_v = yaw;

        prima = true;
    }
    else
    {
        roll = (((e.Roll + PI) / (PI * 2.0f) * -offset) + offset/2) * filtru
+ roll_v * filtruv;
        pitch = (((e.Pitch + PI) / (PI * 2.0f+ 0.00000001) * offsetz)
- offsetz/2) * filtru + pitch_v * filtruv;
        yaw = (((e.Yaw + PI) / (PI * 2.0f) * -offset) + offset/2 +
offsety) * filtru + yaw_v * filtruv;

        roll_v = roll;
        pitch_v = pitch;
        yaw_v = yaw;
    }

    data = yaw.ToString("#.#") + " " +
        roll.ToString("#.#") + " " +
        pitch.ToString("#.##") + " ";
    InvokeData(data);
}

```

Prin subprogramul *GestRecunoscut*, pe de-o parte, se afișează în fereastra programului gestul curent, iar pe de altă parte se analizează dacă gestul curent este

de tip pumn, caz în care „alpha” devine 1.0, sau dacă este de tip degete deschise, caz în care „alpha” va fi 0.0.

```
private void GestRecunoscut (string gest)
{
    if (InvokeRequired)
    {
        this.Invoke(new Action<string>(GestRecunoscut), new
object[] { gest });
        return;
    }

    richTextBox1.AppendText(gest + Environment.NewLine);

    if (gest == "Fist")
    {
        alpha = 1.0;
    }
    if (gest == "FingersSpread")
    {
        alpha = 0.0;
    }
}
```

Subprogramul *getAvailablePorts* aduce în lista din fereastra de design (elementul 4 din Fig. 44), lista de porturi disponibile la care se poate conecta programul.

```
void getAvailablePorts()
{
    String[] ports = SerialPort.GetPortNames();
    comboBox1.Items.AddRange(ports);
}
```

În momentul în care se apasă butonul Open se execută subprogramul de mai jos, prin care, în primul rând, se verifică să fie selectat un port serial, iar mai apoi se deschide canalul de comunicare serial cu robotul industrial KUKA KR15, iar în momentul în care e deschis, bara de progres se colorează complet.

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        if (comboBox1.Text=="")
        {
            MessageBox.Show("Introduceți Serial Port");
        }
        else
        {
            serialPort1.PortName = comboBox1.Text;
            serialPort1.BaudRate = 9600;
            serialPort1.Open();
            progressBar1.Value = 100;
            button1.Enabled = false;
            button2.Enabled = true;
        }
    }
}
```

Subprogramul de mai jos se acționează atunci când se apasă butonul *Close*, iar prin aceste instrucțiuni se închide portul serial, bara de progres devine 0, iar programul se oprește din rulare.

```
private void button2_Click(object sender, EventArgs e)
{
    timer1.Enabled = false;
    serialPort1.Close();
    progressBar1.Value = 0;
    button1.Enabled = true;
    button2.Enabled = false;
}
```

Programul de mai jos rulează repetitiv, la o frecvență presetată, toate instrucțiunile programului. Așadar, în fereastră se vor afișa valorile „*roll*”, „*pitch*”, „*yaw*”, precum și valoarea lui „*alpha*”. În cazul în care portul serial este deschis, acesta va transmite către robotul industrial aceste date.

```
private void timer1_Tick(object sender, EventArgs e)
{
    textBox1.Text = roll.ToString("0.0") + " " + yaw.ToString("0.0")
+ " " + pitch.ToString("0.0") + " " + alpha.ToString("0");
    if (serialPort1.IsOpen)
    {
        serialPort1.Write(textBox1.Text);
    }
}
```


b. Program KUKA

Diagrama de funcționare a programului care rulează pe robotul industrial KUKA este prezentată în Fig. 45.

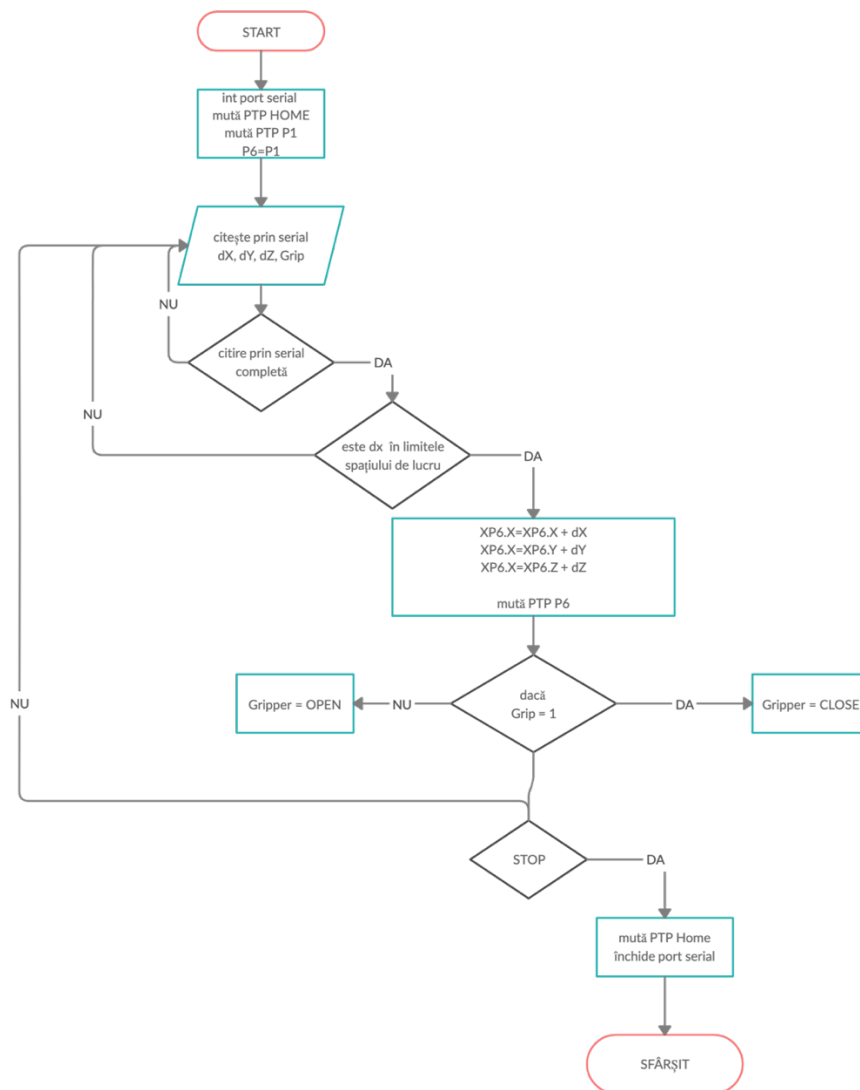


Fig. 45 - Diagrama de funcționare a programului KUKA

52 - EVALUAREA SOLUȚIEI OPTIME

Programul robotului industrial KUKA KR15 a fost realizat în KRL (KUKA Robot Language), un limbaj de programare specific roboților KUKA, care este foarte similar cu Pascal.

În prima fază s-au inițializat toate variabilele care vor fi folosite în program. „MW_T”, „MR_T”, „TIMEOUT” și „OFFSET” sunt variabile folosite atunci când se citesc date prin intermediul portului serial. COPEN este o instrucțiune care definește portul serial care va fi folosit. „base” și „tool” sunt variabile care definesc valorile bazei și a efectorului final folosit. „dX”, „dY”, „dZ” sunt variabilele în care se vor stoca valorile „roll”, „pitch”, „yaw” trimise de către programul Visual Studio prin intermediul portului serial. „Grip” este o variabilă care va prelua valoarea lui „alpha” din programul Visual Studio și va indica dacă se va închide dispozitivul de prehensiune sau nu.

```
INI
MW_T=#SYNC
MR_T=#COND
TIMEOUT=3.0
OFFSET=0
COPEN(:SER_3,HANDLE)
$base=base_data[1]
$tool=tool_data[1]
dX=0
dY=0
dZ=0
Grip=0
```

Instrucțiunea de mai jos va duce efectorul final în poziția inițială, iar apoi acesta se va deplasa în punctul P1, care va fi practic deasupra obiectului care se dorește a fi preluat. Mai apoi se va crea punctul P6 care va fi o clonă a punctului P1.

```
PTP HOME Vel= 100 % DEFAULT

PTP P1 Vel= 100 % PDATA9 Tool[2]:Pen Base[1]:Table

XP6=XP1
```

Secvența de mai jos se repetă continuu până când un operator uman oprește programul. CREAD este o instrucțiune care citește valorile primite prin intermediul portului serial și le stochează în variabilele „dX”, „dY”, „dZ” și „Grip”. Prin urmare „roll”, „pitch”, „yaw” și „alpha”, transmise de către programul C#, se vor regăsi în aceste variabile. În cazul în care „dX”, „dY”, „dZ” sunt în interiorul spațiului de lucru pe care l-am definit, atunci punctul XP6 incrementează poziția pe axele x, y și z cu valorile „dX”, „dY”, „dZ”. PTP XP6 instruieste robotul industrial să execute mișcarea în noul punct XP6. Astfel, de fiecare dată când avem valori noi, acestea se vor adăuga, iar robotul se va deplasa în noua poziție. În ultima parte, programul verifică dacă variabila „Grip” este 1, astfel dacă s-a realizat gestul pumn, iar dacă răspunsul este afirmativ, atunci se închide dispozitivul de prehensiune, în caz contrar, dispozitivul de prehensiune se deschide.

```
LOOP

  REPEAT
    OFFSET=0
    CREAD(HANDLE,SR_T,MR_T,TIMEOUT,OFFSET,"%f %f %f %d", dX,
dY, dZ, Grip)

  UNTIL (SR_T.RET1==#CMD_OK)
  IF (dX>=-250) AND (dX<=250) THEN
    XP6.X=XP6.X + dX
    XP6.Y=XP6.Y + dY
    XP6.Z=XP6.Z + dZ

    PTP XP6

  ENDIF

  IF (Grip==1) THEN

    Gripper = CLOSE

  ELSE

    Gripper = OPEN

  ENDIF

  dX=0
  dY=0
  dZ=0
ENDLOOP
```

În ultima parte a programului, se închide portul serial prin instrucțiunea CCLOSE.

```
CCLOSE(HANDLE,SC_T)
```

c. Funcționarea programului.

În momentul în care începe rularea programului, robotul industrial KUKA KR15 se deplasează în poziția HOME, iar apoi în poziția P1, așa cum este prezentată și în Fig. 46, acesta fiind punctul de plecare. Din acest punct se începe controlul cu ajutorul brățării Myo. Ca măsură de siguranță, s-a introdus o cutie de carton sub obiectul care

se dorește a fi preluat, pentru a preveni o coliziune cu masa de lucru, în cazul în care apar erori în program.



Fig. 46 - Poziția inițială P1

Mișcând brațul operatorului, robotul se deplasează în consecință. În Fig. 47 se observă coordonatele interne ale operatorului uman și ale robotului industrial.

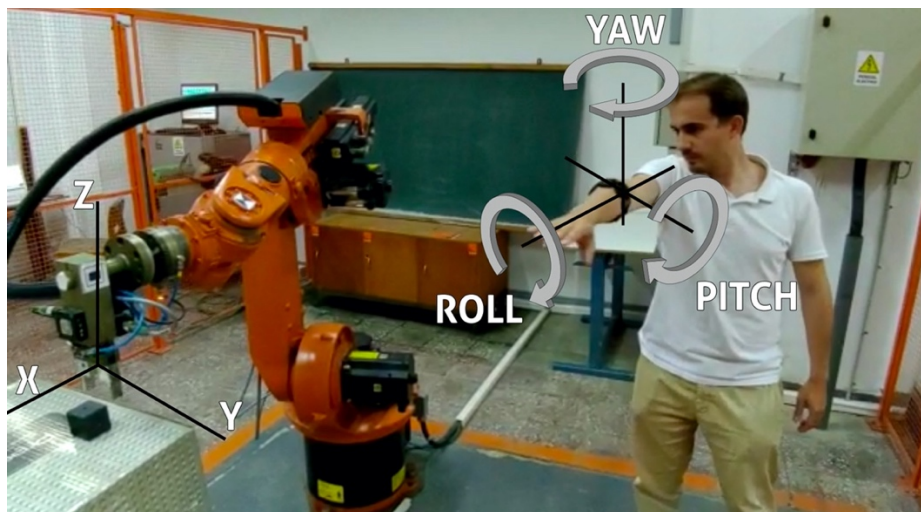


Fig. 47 - Corelare coordonate carteziene KUKA și orientare Myo Armband

Corelarea dintre mișcarea operatorului uman și mișcarea robotului industrial se poate observa în Fig. 48.

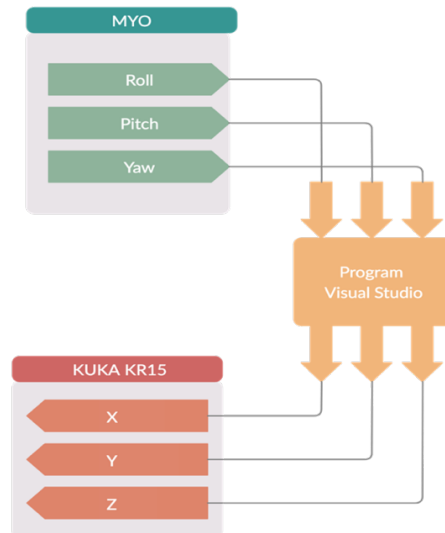


Fig. 48 – Corelare valori Myo Armband cu KUKA KR15

În momentul în care operatorul uman ajunge cu dispozitivul de prehensiune deasupra obiectului, se realizează gestul pumn, iar astfel dispozitivul de prehensiune se închide. Fig. 49 prezintă această operațiune.



Fig. 49 - Gest pumn - închidere dispozitiv de prehensiune

56 - EVALUAREA SOLUȚIEI OPTIME

După ce dispozitivul de prehensiune s-a închis, mișcând brațul se poate manipula și muta obiectul (Fig. 50).

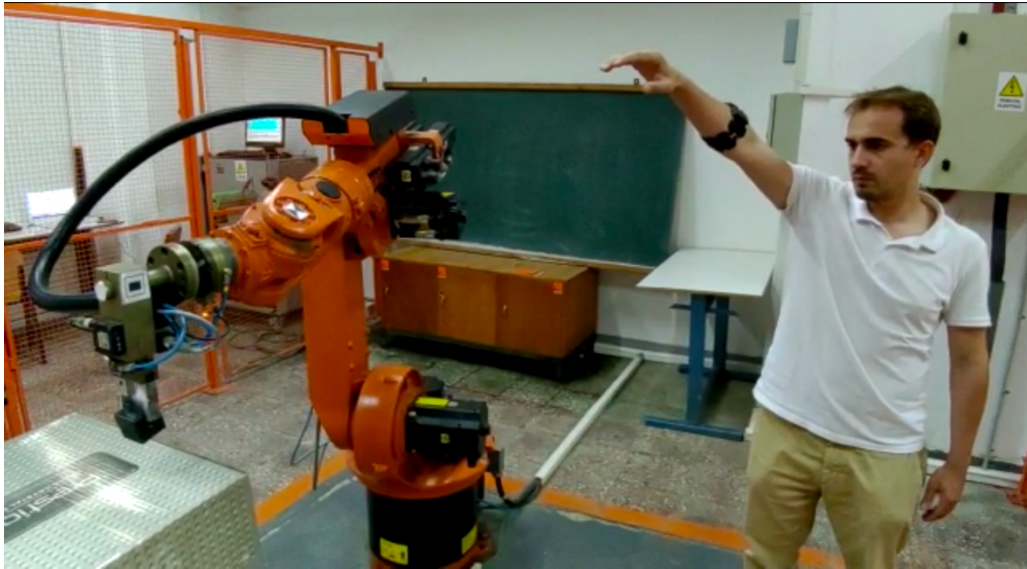


Fig. 50 - Manipulare obiect

În Fig. 51 este prezentată locația unde se mută obiectul.



Fig. 51 - Obiect mutat

În Fig. 52 este prezentată modalitatea de a deschide dispozitivul de prehensiune prin intermediul gestului degete deschise.

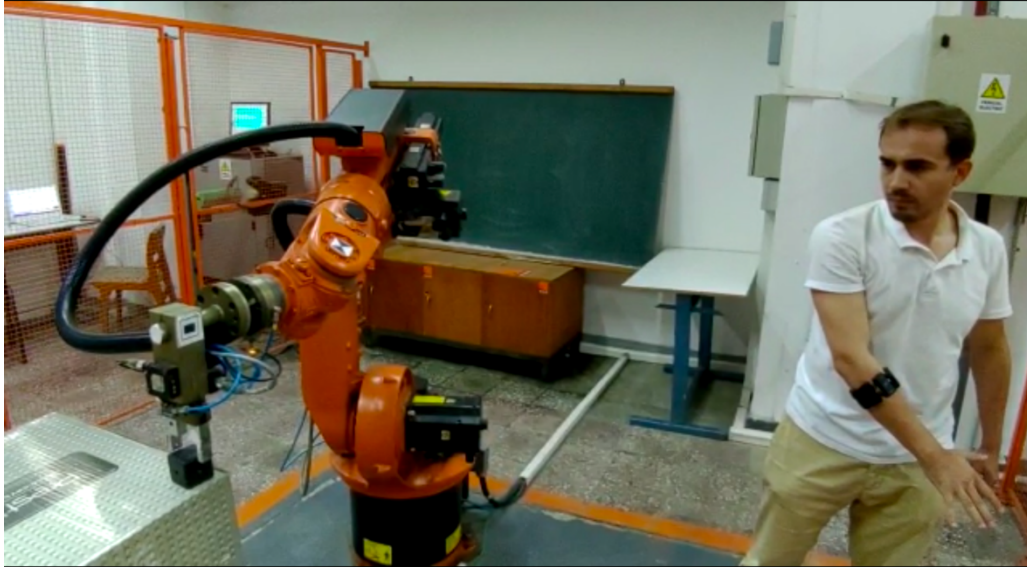


Fig. 52 - Deschidere dispozitiv de prehensiune

Având dispozitivul de prehensiune deschis, se poate aduce efectorul final al robotului în altă locație (Fig. 53)



Fig. 53 - Mutare efector final în altă locație

4.2.4 Concluzii

Experimentul a fost de mare ajutor și a demonstrat într-o mare măsură faptul că robotul industrial KUKA KR15 se poate controla cu ajutorul brățării Myo. Activitatea de cercetare a fost publicată în [103].

Problemele identificate au fost următoarele:

1. Întrucât din brățara Myo se extrag doar trei valori ale orientării, s-a putut muta robotul doar pe axele x , y și z , fără a permite modificarea valorilor de orientare ale KUKA KR15. Astfel, dacă obiectul este poziționat oblic, nu ar putea fi preluat;
2. Comunicarea serială este funcțională, dar nu extrem de practică. De multe ori volumul de date trimis a fost mai mare decât capacitatea de recepție a robotului industrial KUKA KR15, prin comunicare serială, și a dus la blocarea programului;
3. Brățara trebuie calibrată înainte de rularea programului, pentru a recunoaște gesturile din prima încercare, altfel apar erori în identificarea gesturilor;
4. Mișcarea robotului este în șocuri, din cauză că se folosește comanda PTP care frânează robotul la finalul fiecărei poziții.

4.3. Aplicație conexiune Dual Myo – KUKA KR15

Pentru a rezolva problema controlului orientării efectorului final al robotului industrial KUKA KR15, am decis conectarea a două brățări Myo (Fig. 54), una care să controleze poziționarea robotului pe axele x , y , z și a doua care să controleze orientarea $alpha$, $beta$, $gamma$.



Fig. 54 - Aplicație conexiune Dual Myo – KUKA KR15

4.3.1. Arhitectura sistemului

În Fig. 55 este prezentată arhitectura sistemului Dual Myo – KUKA KR15 care are ca principale componente:

- 1- Brățara Myo nr. 1;
- 2- Dispozitiv comunicare bluetooth nr.1;
- 3- Brățara Myo nr. 2;
- 4- Dispozitiv comunicare bluetooth nr.2;
- 5- Cablu serial;
- 6- Robot industrial KUKA KR15.



Fig. 55 - Arhitectura sistemului de conectare Dual Myo - KUKA KR15

4.3.2. Schema de conectare

Sistemul are în compoziție două brățări Myo. Prima brățară Myo este conectată la un sistem de operare virtual creat în interiorul sistemului de operare de pe computerul portabil, prin intermediul dispozitivului bluetooth aferent. Pe același principiu, a doua brățară este conectată direct la sistemul de operare al computerului portabil, prin intermediul dispozitivului bluetooth aferent. Robotul KUKA KR15 este conectat la computerul portabil prin intermediul unui cablu serial, așa cum este prezentat și în Fig. 56.

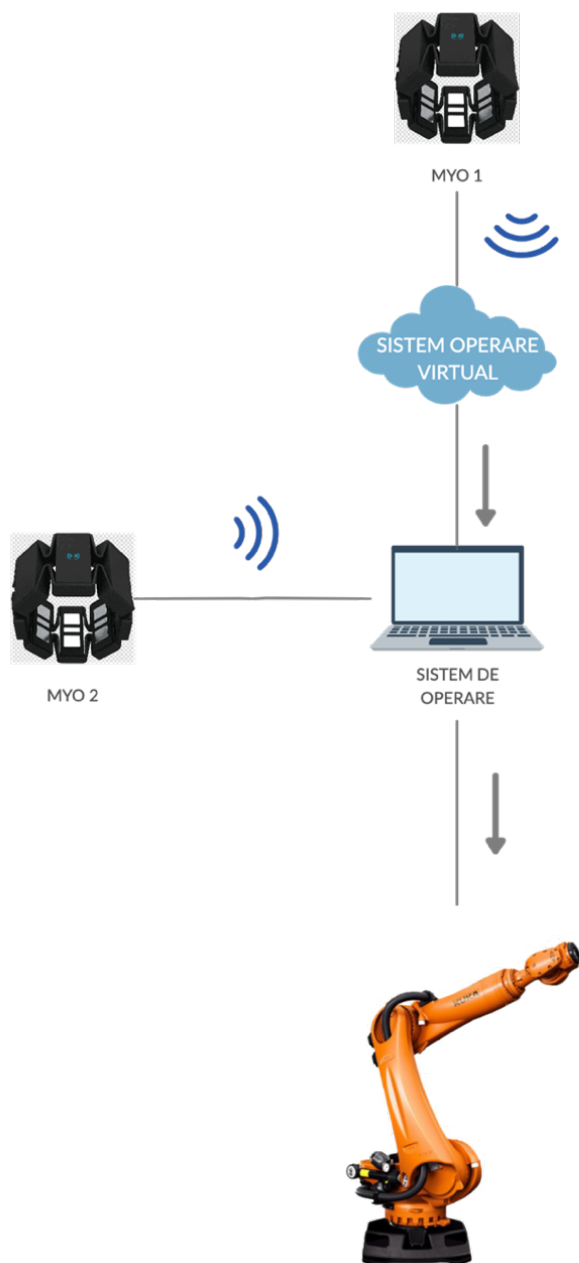


Fig. 56 - Schema de conectare a sistemului

4.3.3. Încercări și rezultate experimentale

Pentru a pune în practică sistemul, în primul rând, pe computerul portabil, trebuie instalat un sistem de operare virtual. În situația prezentă, s-a instalat Oracle VM VirtualBox, un soft care a permis instalarea sistemului de operare virtual Windows 10, așa cum este prezentat în Fig. 57.

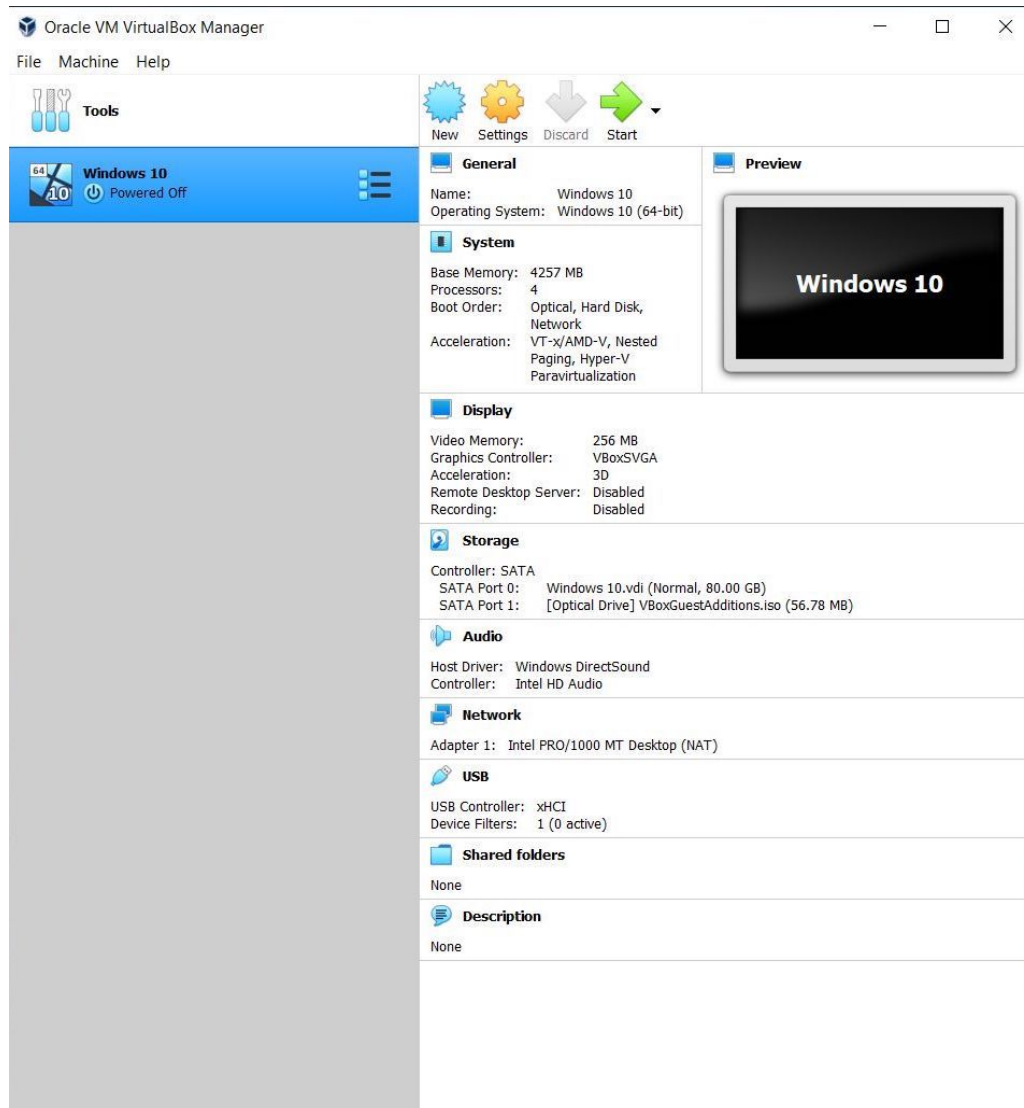


Fig. 57 - Oracle VM VirtualBox

Urmează apoi crearea a două porturi seriale virtuale, prin intermediul programului HDD Virtual Serial Port, conform fluxului din Fig. 58, Fig. 59, Fig. 60, Fig. 61.

62 - EVALUAREA SOLUȚIEI OPTIME

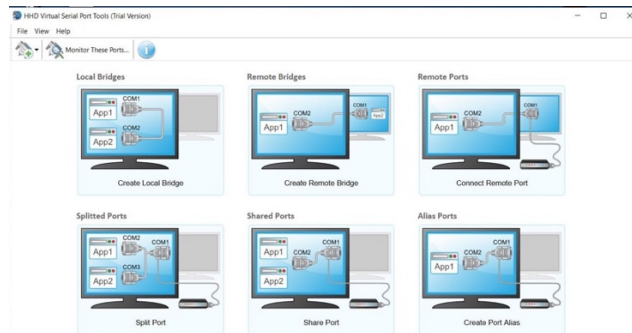


Fig. 58 - Pasul 1: Se deschide programul HDD Virtual Serial Port Tools

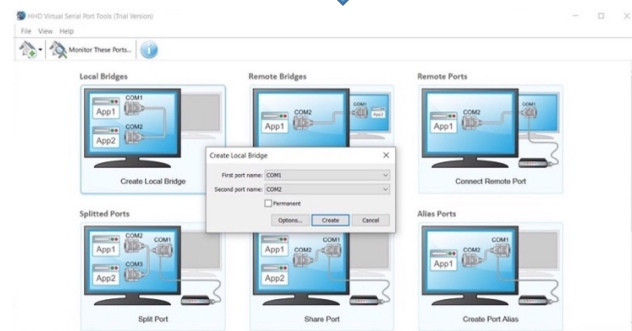


Fig. 59 - Pasul 2: Se selectează „Create Local Bridge” și se alege primul port COM1 iar portul al doilea COM2

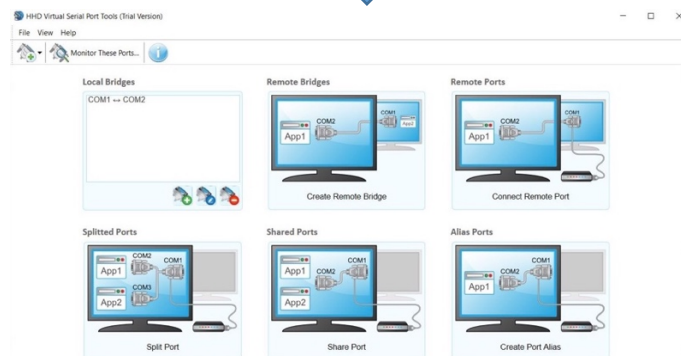


Fig. 60 - Pasul 3: Se realizează conectarea porturilor virtuale

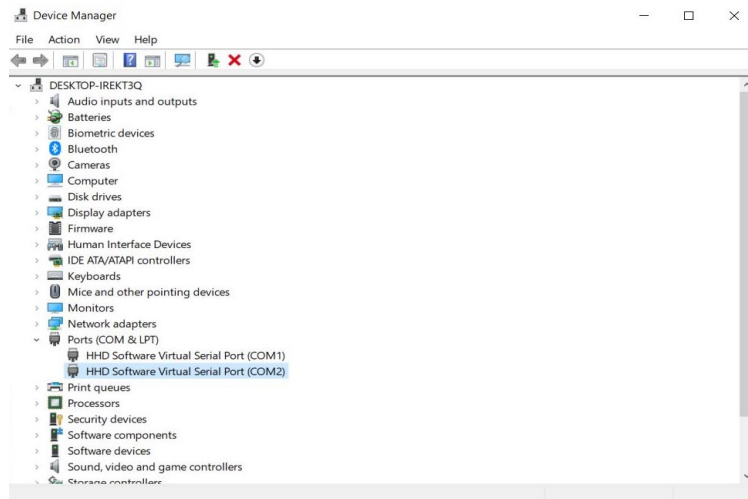


Fig. 61 - Pasul 4: Se verifică dacă s-au creat cele două porturi virtuale

După realizarea porturilor seriale virtuale, ultima etapă de pregătire este alocarea portului USB al dispozitivului de comunicare bluetooth aferent brățării Myo, în sistemul de operare virtual, conform Fig. 62. Controlerul KUKA KR15 rulează de asemenea un program care „ascultă” după mesaje transmise prin portul serial RS232. Când o informație validă este recepționată, controlerul execută o mișcare specifică.

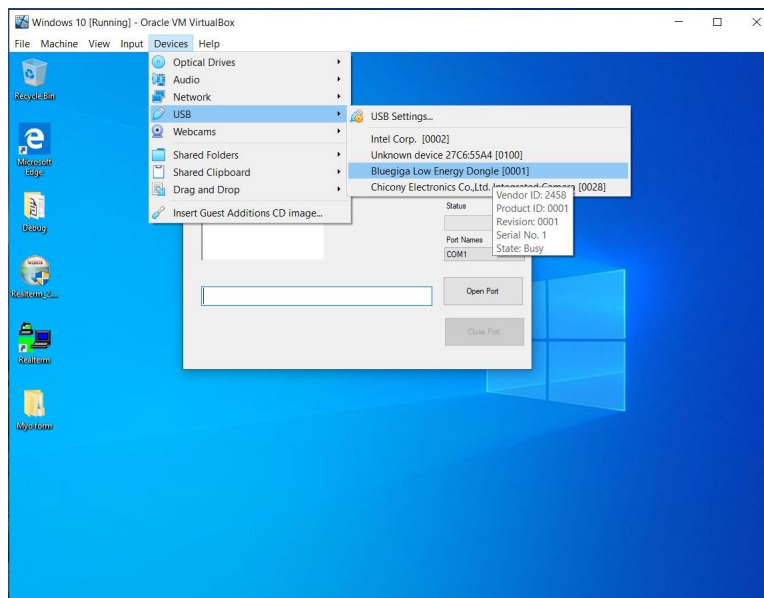


Fig. 62 - Alocare port USB al dispozitivului de comunicare bluetooth aferent brățării Myo 1

a. Program Myo

În sistemul de operare virtual, va rula programul C# Visual Studio prezentat în capitolul 4.2. În plus față de experimentul inițial cu o singură brățară, s-a realizat un program C# în Visual Studio care preia valorile de la brățara Myo nr. 1 prin intermediul porturilor virtuale și a sistemului de operare virtual, iar pe de altă parte, programul citește valorile „roll”, „pitch”, „yaw” primite de la brățara Myo nr.2. Aceste două șiruri de valori sunt introduse într-o singură variabilă, care va fi trimisă către robotul industrial KUKA KR15.

Schema bloc de funcționare a programului Myo în Visual Studio este prezentată în Fig. 63.

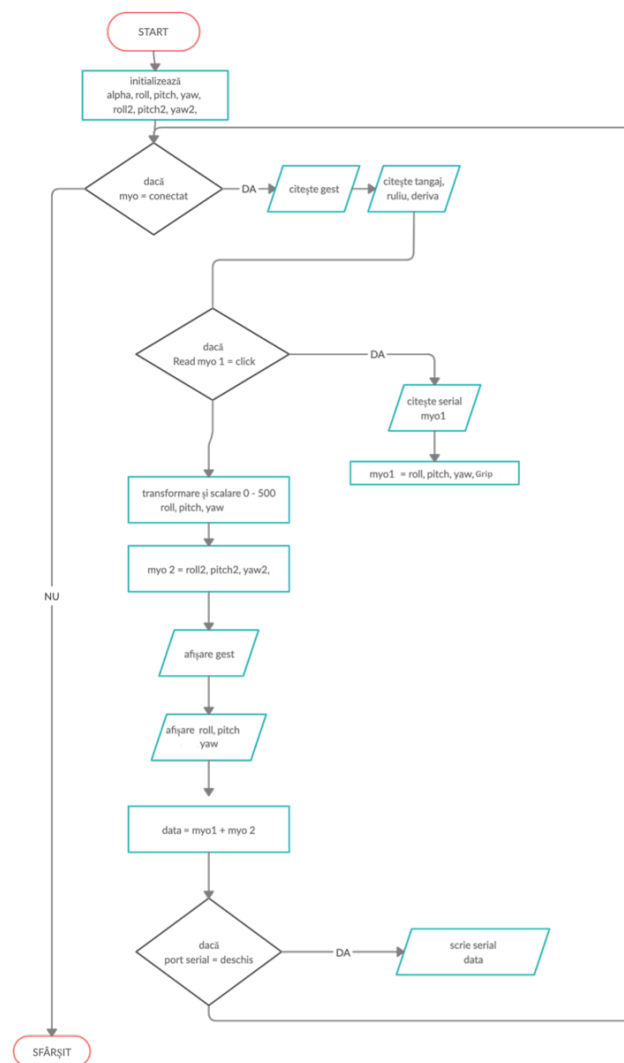


Fig. 63 - Diagrama de funcționare program C# Dual Myo

Programul C# în Visual Studio are două componente:

1. Interfață vizuală

În interfața Design din cadrul Visual Studio s-a creat o fereastră prin care utilizatorul poate vizualiza date, precum și deschide sau închide porturile seriale.

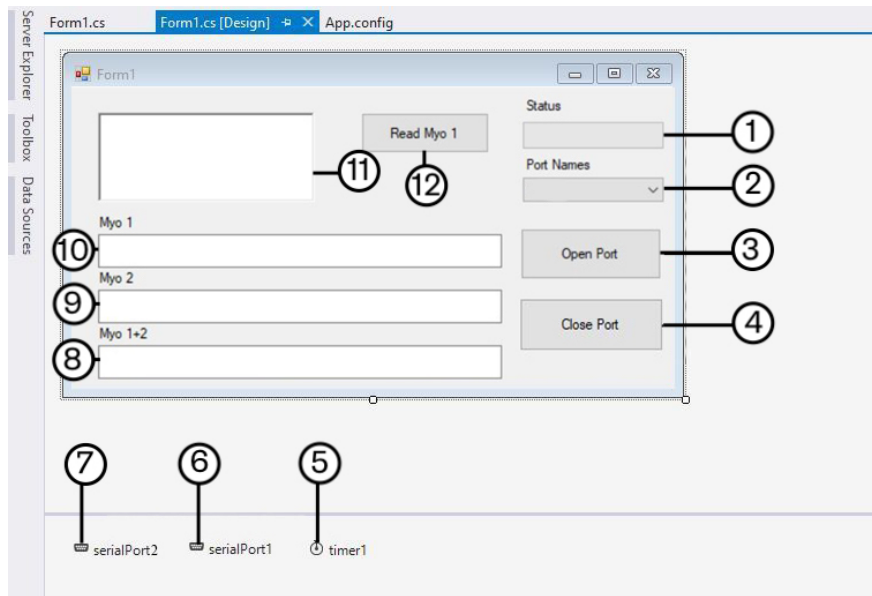


Fig. 64 - Elementele interfeței vizuale ale programului Visual Studio

Elementele din interfața vizuală (Fig. 64) sunt:

- 1- bară de progres;
- 2- port serial;
- 3- buton deschidere port serial;
- 4- buton închidere port serial;
- 5- temporizator;
- 6- port serial 1;
- 7- port serial 2;
- 8- afișaj date Myo 1+2;
- 9- afișaj date Myo 2;
- 10- afișaj date Myo 1;
- 11- text multiplu;
- 12- port serial;
- 13- contor.

2. Cod program C#

Față de programul de la capitolul 4.2, unde a funcționat doar o brățară, elementul de noutate este faptul că s-a introdus un buton nou „Read Myo 1”, prin care se citesc datele trimise de la brățara Myo nr.1. În acest caz s-a realizat achiziția de date și de la o a doua brățară numită Myo nr.2. Astfel, codul era preluat cu următoarea instrucțiune:

```
private void timer1_Tick(object sender, EventArgs e)
{
    textBox1.Text = roll.ToString("0.0") + " " +
yaw.ToString("0.0") + " " + pitch.ToString("0.0") + " " +
Grip.ToString("0");
    textBox2.Text = roll2.ToString("0.0") + " " +
yaw2.ToString("0.0") + " " + pitch2.ToString("0.0")

    textBox3.Text = textBox1.Text + " " + textBox2.Text
    if (serialPort1.IsOpen)
    {
        serialPort1.Write(textBox3.Text);
    }
}
```

b. Program KUKA KR15

Diagrama de funcționare a programului, care rulează pe robotul industrial KUKA este prezentată în Fig. 65.

Față de codul menționat la 4.2, în acest caz, având trei variabile în plus de situație, citirea lor în programul KUKA se va realiza utilizând instrucțiunea următoare:

```
CREAD(HANDLE,SR_T,MR_T,TIMEOUT,OFFSET,"%f %f %f %d %f %f %f ", dX, dY,
dZ, Grip, alpha, beta, gamma)
```

Astfel, vom avea cele trei valori de orientare stocate în variabilele „alpha”, „beta”, „gamma”. Fiecare nouă poziție se va actualiza utilizând următoarea secvență de cod:

```
XP6.X =XP6.X+dX
XP6.Y =XP6.Y+dY
XP6.Z =XP6.Z+dZ
XP6.a =XP6.a+alpha
XP6.b =XP6.b+beta
XP6.c =XP6.c+gamma
```

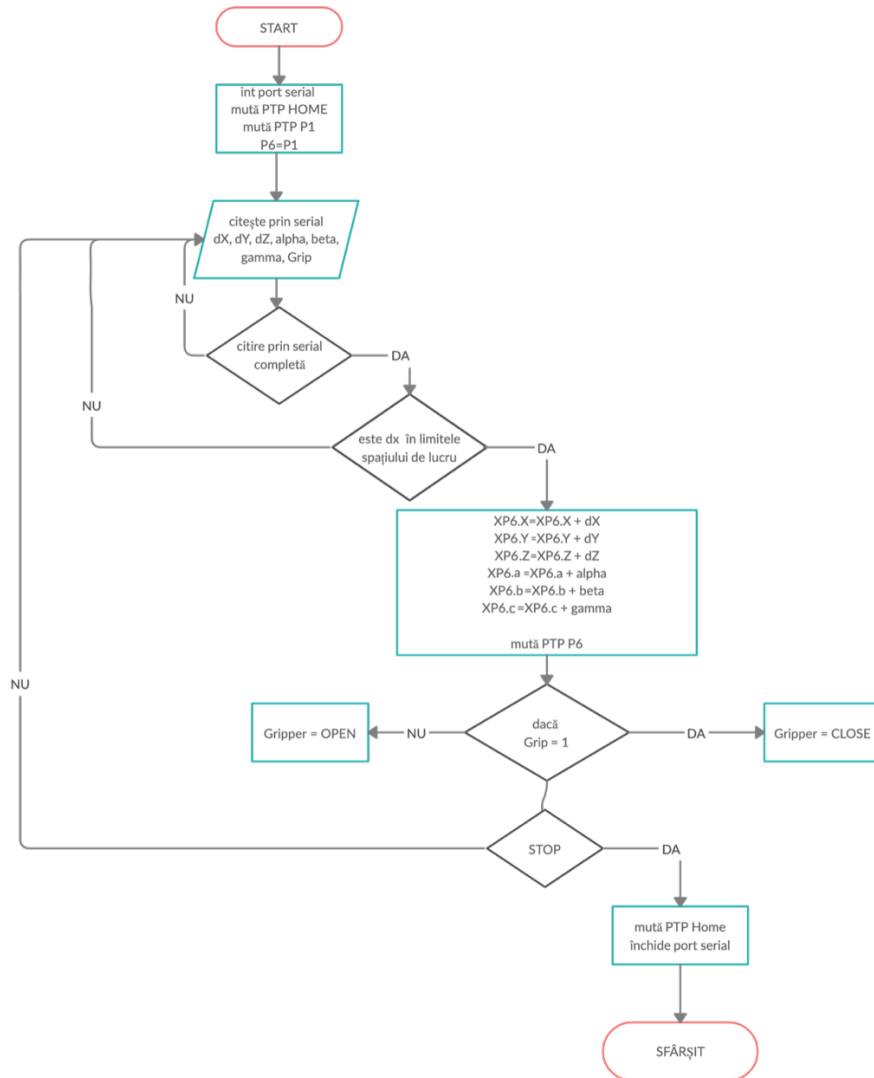



Fig. 65 - Diagrama de funcționare a programului KUKA

4.3.4. Concluzii

În cadrul acestui experiment s-a testat modalitatea de achiziție a datelor de la două brățări, una care să transmită coordonatele de poziționare x, y, z și starea dispozitivului de prehensiune, iar cealaltă brățară să transmită coordonatele de orientare R_x, R_y, R_z .

1. Mișcarea robotului industrial cu două brățări a fost extrem de dificilă întrucât este foarte greu de controlat șase coordonate în același timp, iar fiecare

- mișcare mai sensibilă a brățării, care transmitea datele de orientare, conducea spre o poziție de singularitate;
2. Nu este fezabilă soluția întrucât este extrem de dificilă realizarea mișcării unei singure mâini, ținând cealaltă mână în poziție fixă, din cauza legăturilor corporale dintre membre. Deși valorile s-au scalat și s-au filtrat, tot existau mișcări de orientare care îngreunau prinderea obiectelor;
 3. Experimentul va continua analizând doar coordonatele x , y , z , renunțând la orientare;
 4. La fel ca în primul experiment, mișcarea este sacadată din cauza comunicării prin portul serial și a mișcării PTP.

4.4. Aplicație Myo Armband – Universal Robots UR10e

4.4.1. Arhitectura sistemului

În acest caz sistemul este compus dintr-o robot colaborativ Universal Robots UR10e, o brățară Myo Armband, un sistem desktop și elementele de comunicare bluetooth cât și ethernet.

Principalele componente ale sistemului (Fig. 66):

- a) Sistem desktop PC;
- b) Dispozitiv comunicare bluetooth;
- c) Brățara Myo;
- d) Universal Robots UR10e;
- e) Cablu ethernet.

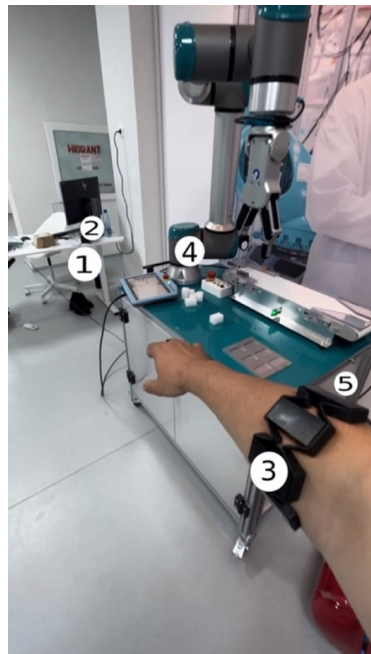


Fig. 66 - Principalele elemente componente ale arhitecturii Myo-UR10e

4.4.2. Schema de conectare

Brățara Myo Armband comunică cu sistemul desktop PC prin intermediul protocolului de comunicare bluetooth. Astfel informațiile sunt transmise de la brățară la sistemul desktop PC pe care rulează două programe: unul în C# care preia informațiile de la Myo, interpretează gesturile, le prelucrează și le transmite apoi prin rețeaua internă (localhost) către un program Python care interpretează datele și transmite coordonate de poziționare către robotul colaborativ UR10e. (Fig. 67)

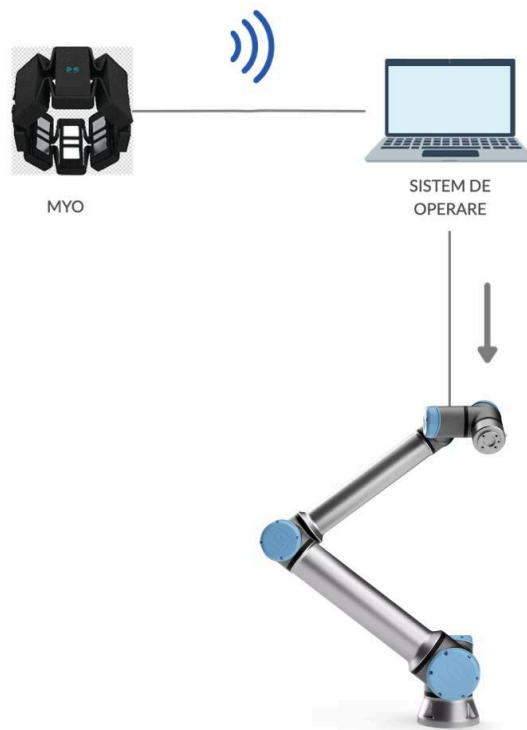


Fig. 67 - Schema de conectare Myo - UR10e

4.4.3. Încercări și rezultate experimentale

Pentru a efectua experimentele s-au realizat practic trei programe astfel:

1. Program C# pentru a putea prelua datele de la Myo Armband prin intermediul protocolului de comunicare bluetooth;

2. Program Python pentru a prelua datele de la programul C# din rețeaua locală (localhost) și pentru a le transmite către robotul Universal Robots UR10e prin intermediul unui cablu ethernet;
3. Program UR10e, pentru a prelua datele de la programul Python și pentru a mișca robotul propriu-zis.

Diagrama de comunicare se poate observa în Fig. 68.

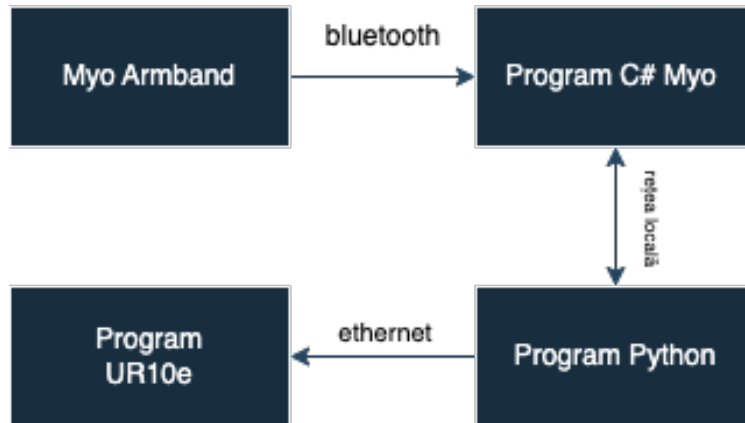


Fig. 68 - Diagrama de comunicare între programele Myo-UR10e

4.4.3.1. Programul C# Myo

În cadrul acestui program, principalul obiectiv este să transmitem valorile captate de la brățara Myo Armband prin intermediul unui modul bluetooth, către un program Python. Schema de funcționare a programului C# Myo este prezentată în Fig. 69. Astfel, se poate observa că prima etapă este deschiderea unui protocol de comunicare cu programul Python, mai apoi achiziția de date de la Myo Armband, care sunt prelucrate în valorile „yaw”, „pitch”, „roll” și în funcție de gestul realizat se transmite către programul Python una din valorile 1, 2, 3 sau 4. Acest program rulează continuu, până când utilizatorul îl oprește.

Față de programul scris la 4.2, în acest caz, datele nu se transmit pe protocolul de comunicare serial, ci intern, deschizând o comunicare în calculator pe rețeaua locală (127.0.0.1 sau localhost). Acest lucru s-a realizat prin intermediul următoarelor linii de cod:

```

using (TcpClient client = TcpClient())
{ try
  { client.Connect("127.0.0.1", 8888);
    NetworkStream stream = client.GetStream();
    byte[] bytes = Encoding.ASCII.GetBytes(data);
    stream.Write(bytes, 0, bytes.Length);
    client.Close(); }
  catch (Exception ex)
  { Console.WriteLine("Eroare: " + ex.Message);
  } }

```

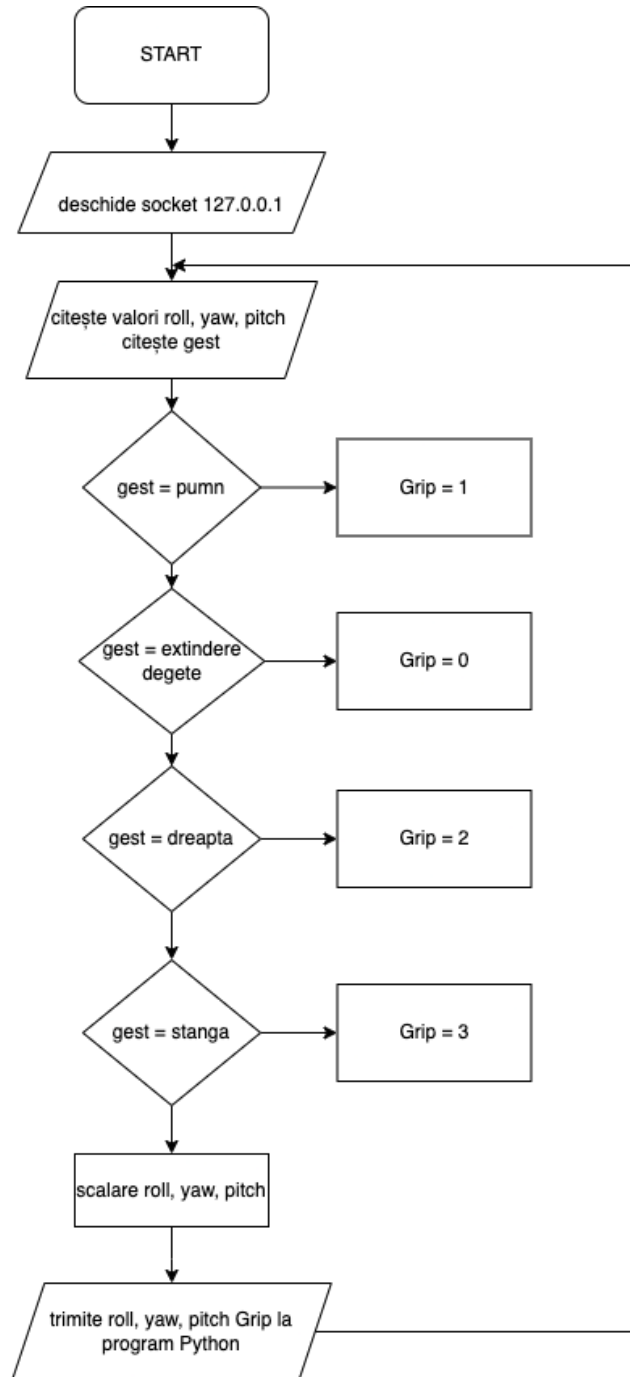


Fig. 69 - Schema de funcționare a programului C# - Myo

Așadar, se transmite o variabilă *data* care conține:

```
string data = roll.ToString("0.0") + " " + yaw.ToString("0.0") + " " +  
pitch.ToString("0.0") + " " + gestul.ToString("0") + " ";
```

În vederea transmiterii de la programul C# către programul Python, a valorii de gest corecte, s-a programat astfel încât, în funcție de gestul identificat, să se transmită o valoare de la 1 la 4, așa cum este prezentat mai jos:

```
if (gest == "FingersSpread") // gest EXTINDERE  
{  
    gestul = 0.0;  
}  
if (gest == "Fist") // gest PUMN  
{  
    gestul = 1.0;  
}  
if (gest == "WaveIn") // gest STÂNGA  
{  
    gestul = 2.0;  
}  
if (gest == "WaveOut") // gest DREAPTA  
{  
    gestul = 3.0;  
}
```

Gesturile identificate sunt prezentate în Fig. 70.



Fig. 70 – Gesturi posibile

4.4.3.2. Programul Python

Acest program are rolul de a prelua valorile „yaw”, „pitch”, „roll” de la programul C# și de a le transmite, printr-o comunicare ethernet, către robotul Universal UR10e, la o frecvență de 500Hz.

Schema bloc de funcționare a programului este prezentată în Fig. 71.

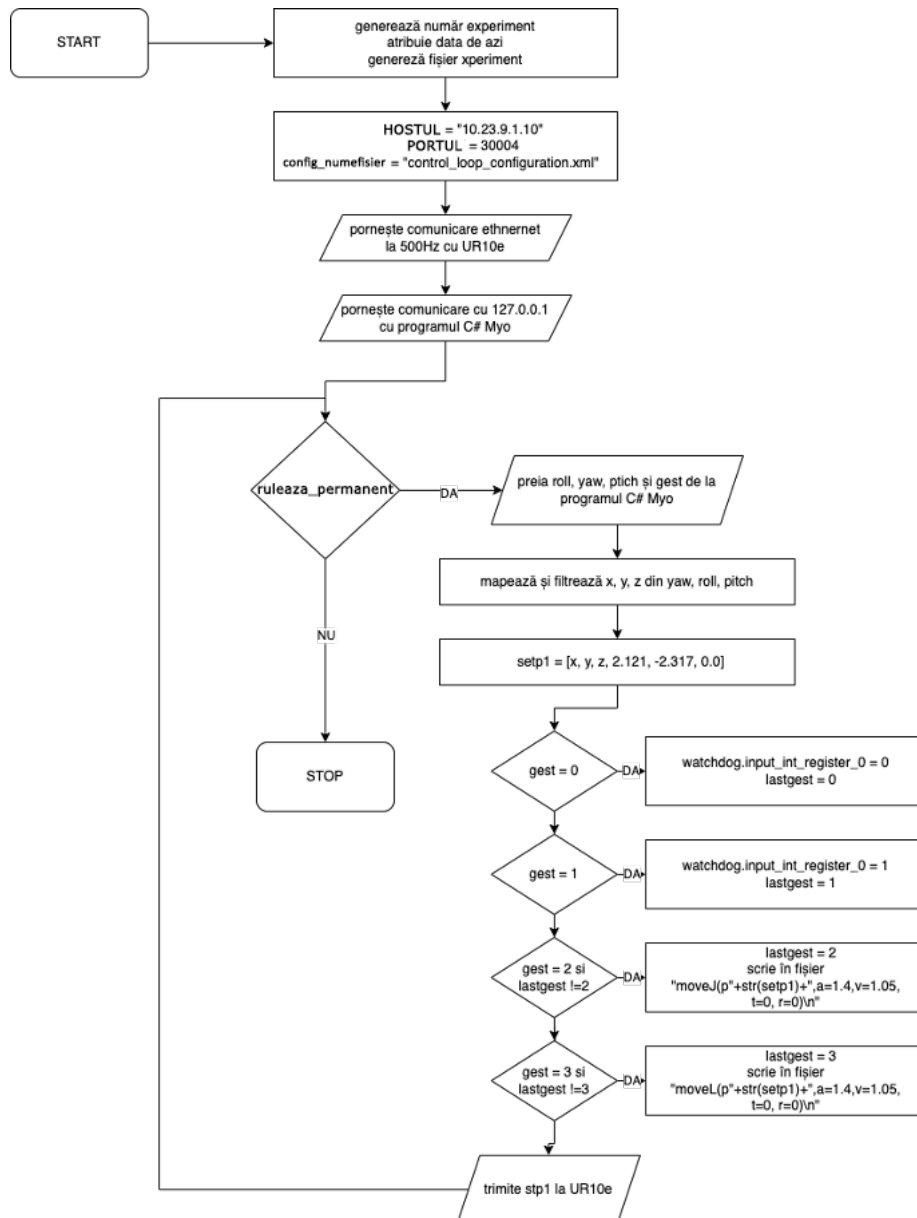


Fig. 71 – Schema bloc de funcționare a programului Python

Pentru a citi valorile de pe rețeaua internă locală (localhost), și pentru a prelua informațiile în variabilele „yaw”, „pitch”, „roll”, „grip”, s-a folosit următorul cod:

```
#creare conexiune pe localhost cu program Myo C#
conexiune_server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conexiune_server.bind(('127.0.0.1', 8888))
conexiune_server.listen(1)

# accepta conexiunea
client_socket, address = conexiune_server.accept()

# citeste datele de pe localhost
data = client_socket.recv(1024).decode('utf-8')

# procesare valori si atribuire in variabile
data_values = data.strip().split(" ")
roll, yaw, pitch, gest = map(float, data_values)
```

Pentru a transmite valorile interpretate de la programul C#, s-a folosit biblioteca RTDE, pusă la dispoziție de către Universal Robots, care permite accesarea robotului în timp real. [104]

Astfel, în primă fază este necesară conexiunea cu robotul UR10e, cu ajutorul următorului cod:[104]

```
# definire IP robot + log-uri
HOSTUL = "10.23.9.110"
PORTUL = 30004
config_numefisier = "control_loop_configuration.xml"
ruleaza_permanent = True
logging.getLogger().setLevel(logging.INFO)
configurare = rtde_config.ConfigFile(config_numefisier)
state_names, state_types = configurare.get_recipe("state")
setp_names, setp_types = configurare.get_recipe("setp")
watchdog_names, watchdog_types = configurare.get_recipe("watchdog")

# conectare la UR10e
conectare = rtde.RTDE(HOSTUL, PORTUL)
conectare.connect()
conectare.get_controler_version()

FRECVENTA = 500 # trimite la 500Hz in loc de 125hz
conectare.send_output_setup(state_names, state_types, FRECVENTA)
setp = conectare.send_input_setup(setp_names, setp_types)
watchdog = conectare.send_input_setup(watchdog_names, watchdog_types)
```

Fișierul *control_loop_configuration.xml* este extrem de important întrucât aici se definesc regiștrii interni care se transmit către robot.

În bucla infinită, *ruleaza_permanent=true*, se transmite către robotul UR10 prin 7 regiștrii interni, atât valorile *x, y, z, Rx, Ry, Rz*, cât și valoarea gestului, cu ajutorul următorului cod pe care l-am creat:

```
# definire punct unde va merge UR10e
setp1 = [x, y, z, 2.121, -2.317, 0.0]
state = conectare.receive()
#daca gest e pumn
if gest==1.0:
    watchdog.input_int_register_0=1
    lastgest=1

#daca gest e degete extinse
if gest==0.0:
    watchdog.input_int_register_0=0
    lastgest=0
#daca gest e mana spre dreapta
if gest == 2.0 and lastgest != 2:
    lastgest=2
    with open(fisier, "a") as file:
        file.write("moveJ(p"+str(setp1)+"",a=1.4,v=1.05,t=0,r=0)\n")
    time.sleep(0.2)
    print(lastgest)
    print(gest)

#daca gest e mana spre stanga
if gest == 3.0 and lastgest != 3:
    with open(fisier, "a") as file:
        file.write("moveL(p"+str(setp1)+"",a=1.4,v=1.05,t=0,r=0)\n")
    lastgest=3

#Trimite date la UR10e
new_setp = setp1
list_to_setp(setp, new_setp)
conectare.send(setp)

# trimite watchdog
conectare.send(watchdog)
# inchie socket
client_socket.close()
```

Astfel, atunci când utilizatorul va face gestul mișcare palmă spre stânga, atunci acesta va salva o instrucțiune MoveL (mișcare liniară) în locația unde se află efectorul final, iar când acesta va efectua gestul mișcare palmă spre dreapta, atunci se va salva poziția efectorului final cu instrucțiunea MoveJ (mișcare articulată). În situația în care utilizatorul închide pumnul, atunci se va închide dispozitivul de prehensiune, iar când extinde degetele, se va deschide efectorul final. Codul sursă complet al tuturor programelor este prezentat în Anexa 1.

4.4.3.3. Programul Universal Robots (UR)

În cadrul robotului UR10e va rula un program permanent care va citi valorile trimise de către programul Python, stocate în cei șase regiștrii de tip număr zecimal și gestul trimis în registrul intern de tip întreg. Astfel, după preluarea informațiilor, robotul va proceda la generarea unei variabile de poziție setp1. Pentru prima rulare acesta va poziționa efectorul final în prima poziție citită cu instrucțiunea MoveJ. Ulterior, robotul va folosi instrucțiunea ServoJ pentru a muta robotul în pozițiile viitoare. Totodată, acesta va citi valoarea de pe registrul intern de tip întreg în variabila „grip”. Dacă „grip” este 1, va închide dispozitivul de prehensiune, iar dacă este 0, îl va deschide. Programul a fost realizat tot pe baza a bibliotecii RTDE, puse la dispoziție de către Universal Robots. [104]

Instrucțiunea ServoJ permite controlul robotului în timp real, în funcție de pozițiile axelor, prin reglarea poziției curente către poziția dorită, utilizând un sistem de feedback. [105]

Parametrii ServoJ sunt: *servoJ* (*q*, *a*, *v*, *t*, *lookahead_time*, *gain*), unde:

q- pozițiile cuplelor prin care va deplasa robotul, sub formă de vector unghiular;

a- accelerația maximă;

v- viteza maximă;

t- timpul în care se actualizează controlul mișcării;

lookahead_time- parametru care ajută la anticiparea unei poziții viitoare;

gain- parametru ce ajustează poziția curentă spre poziția viitoare.

lookahead_time, *gain* și *t* sunt similari cu parametrii conceptului PID (proporțional, integrativ și derivativ).[105]

Pentru a determina pozițiile cuplelor se folosește funcția *get_inverse_kin()*, pusă la dispoziția utilizatorilor de către producător, prin care se fac calculele de cinematică inversă pentru fiecare poziție în care se află robotul.

Cinematica directă și cinematică inversă reprezintă concepte fundamentale în robotică și descriu mișcarea robotului, însă în două moduri diferite. În cazul cinematicii directe, se cunosc valorile unghiulare ale articulațiilor fiecărei axe a robotului și se determină coordonatele carteziane în spațiu ale efectorului final. Pe de altă parte, în conceptul de cinematică inversă se cunosc valorile carteziane și se dorește aflarea coordonatele unghiulare ale articulațiilor robotului [106]. Corelația dintre cinematica directă și cinematica inversă se poate observa în Fig. 72.

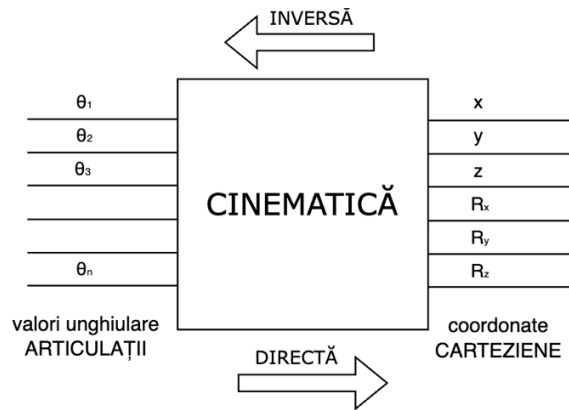


Fig. 72 - Schema de corelație între cinematica directă și inversă

În cazul cinematicii inverse se cunosc coordonatele de situare, x, y, z, R_x, R_y, R_z și se dorește aflarea valorilor unghiulare ale articulațiilor $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$.

În [107], Hawkins prezintă determinarea valorilor unghiulare (coordonate generalizate) astfel:

Se consideră ecuația de definire a cinematicii inverse:

$${}^0B_6(\theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i, \theta_5^i, \theta_6^i) = ({}^0B_6^d) = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

și ecuația de definire a cinematicii directe,

$${}^0B_6(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) = {}^0B_1(\theta_1) {}_1B_2(\theta_2) {}_2B_3(\theta_3) {}_3B_4(\theta_4) {}_4B_5(\theta_5) {}_5B_6(\theta_6) = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

Pornind de la ecuațiile (4.1) și (4.2), Hawking definește următoarele coordonate generalizate:

Din Fig. 73 se poate deduce ecuația (4.3) pentru coordonata θ_1 .

$$\theta_1 = \alpha_1 + \alpha_2 + \pi/2 = \text{atan} 2 \left(({}^o p_{05})_y, ({}^o p_{05})_x \right) \pm \cos^{-1} \frac{d_4}{R} + \pi/2 \quad (4.3)$$

Celelalte trei ecuații pentru $\theta_2, \theta_3, \theta_4$, se obțin la fel ca și în cazul unui braț plan cu trei axe de rotație. [107]

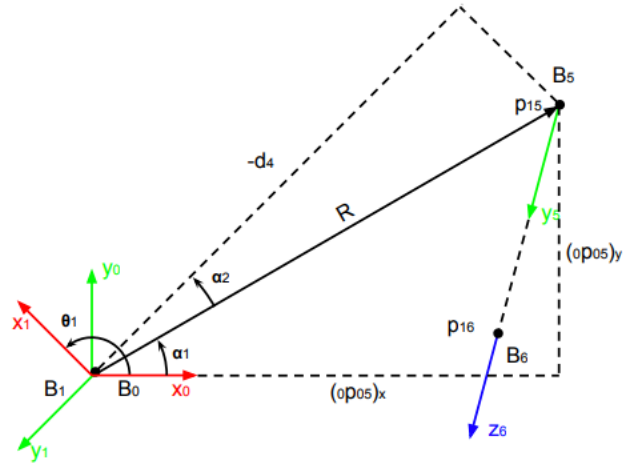


Fig. 73 - Geometria pentru găsirea valorii unghiulare θ_1 [107]

Din Fig. 74. se pot deduce ecuațiile (4.4) și (4.5) pentru determinarea coordonatei θ_5, θ_6 .

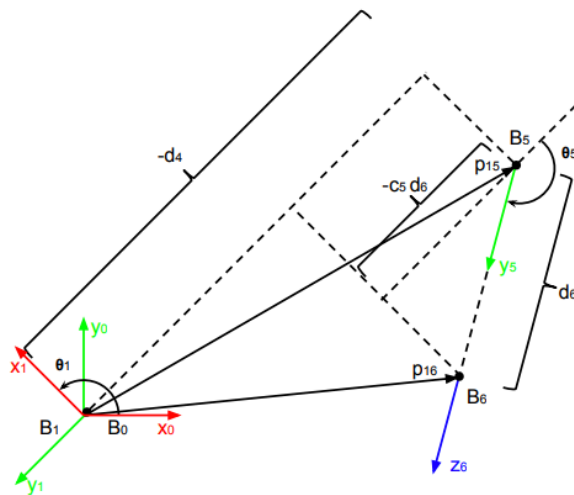


Fig. 74 - Geometria pentru găsirea valorilor unghiulare θ_5 și θ_6 [107]

$$\theta_5 = \pm \cos^{-1} \frac{p_x s_1 - p_y c_1 - d_4}{d_6} \quad (4.4)$$

$$\theta_6 = \operatorname{atan2} \left(\frac{-y_x s_1 + y_y c_1}{s_5}, \frac{-(-x_x s_1 + x_y c_1)}{s_5} \right) \quad (4.5)$$

Diagrama de funcționare a întregului program UR10e se poate observa în Fig. 75.

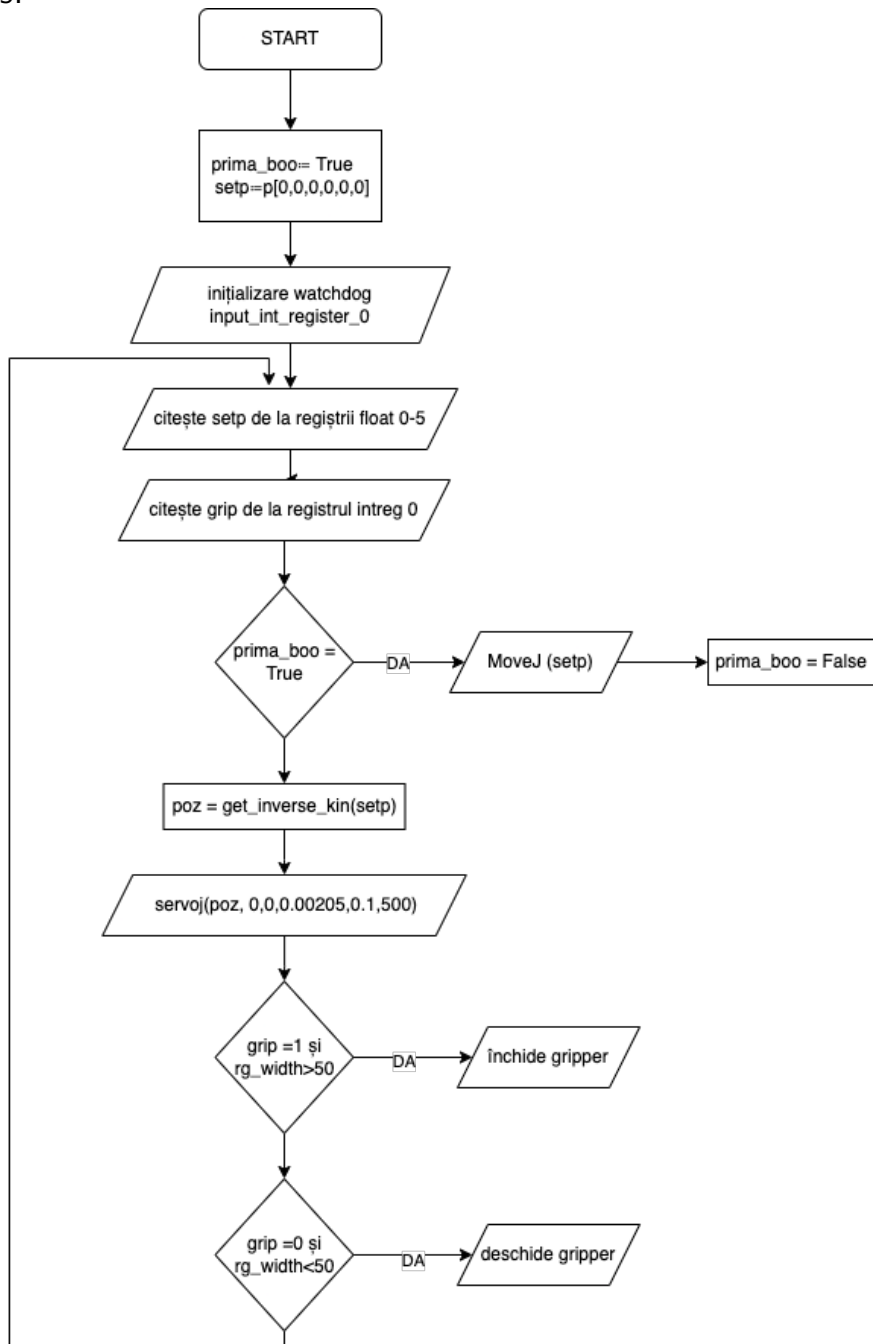


Fig. 75 - Diagrama de funcționare a programului UR10e

4.4.4. Concluzii

- experimentul a permis realizarea de mișcări continue, fără șocuri, așa cum s-a anticipat, datorită faptului că robotul UR10e permite comunicare prin ethernet, la 500Hz, iar funcția specială ServoJ permite ajustarea pozițiilor viitoare;
- utilizatorii pot salva foarte ușor pozițiile în care se deplasează cu brățara, doar prin gesturi;
- brățara trebuie testată pe utilizatori reali care să ofere feedback privind experiența de utilizare.

5. METODOLOGIA DE CERCETARE ȘI TESTAREA SOLUȚIEI OPTIME

În cadrul acestei etape, s-a testat soluția optimă pe un eșantion de 13 utilizatori care au realizat trei experimente. Metodologia de cercetare a vizat realizarea acestor experimente identice atât cu Myo Armband, cât și cu consola de învățare și compararea rezultatelor măsurate și rezultatelor chestionarelor completate de utilizatori.

În cadrul fiecărui experiment, utilizatorii sunt monitorizați din punct de vedere al vitezei, dar și al preciziei de poziționare. Astfel, fiecare test este cronometrat pentru determinarea vitezei și analizat, din punct de vedere al preciziei, printr-un scor calculat.

Toți utilizatorii vor realiza următoarele trei teste:

1. Preluarea și plasarea a trei obiecte (paralelipiped din plastic);
2. Atingerea celor patru repere (bețe);
3. Preluarea și plasarea unui obiect (sticlă PET) în cutie.

Înainte de a începe testele, utilizatorilor li se acordă trei minute pentru a se familiariza cu fiecare dispozitiv, iar mai apoi trebuie să conducă robotul în trei puncte aleatorii și să salveze punctele respective într-un program.

Fiecare experiment are același punct de plecare și are toate obiectele plasate în locuri identice, pentru toți utilizatorii, atât când vor folosi brățara Myo Armband, cât și atunci când vor folosi consola de învățare.

La finalul fiecărui experiment, toți utilizatorii trebuie să completeze un chestionar de feedback prin care se va analiza comparativ atât experiența de utilizare a brățării Myo Armband, cât și aceea cu consola mobilă de învățare. Rezultatele experimentale sunt transcrise apoi într-un tabel și interpretate pentru a vedea corelarea dintre chestionare și realitate.

Pentru testele 1 și 3 se vor analiza individual atât viteza și precizia de poziționare, cât și analiza multicriterială echivalentă pentru aceeași parametrii cu ponderi 50%-50%. Pentru testul 2 se va face doar analiza vitezei.

a) Analiza timpilor

Pentru a compara timpii de realizare a experimentelor se vor folosi următoarele formule de calcul:

$$T_{mM} = \sum_n^1 \frac{t_{iM}}{n} \quad (5.1.)$$

unde:

T_{mM} = media timpilor testelor realizate de utilizatori cu Myo Armband;

t_{iM} = timpul utilizatorului i de finalizare a experimentului cu Myo Armband;

n = numărul total de măsurători.

și

$$T_{mc} = \sum_n \frac{t_{ic}}{n} \quad (5.2.)$$

unde:

T_{mc} = media timpilor testelor realizate de utilizatori cu consola de învățare
 t_{ic} = timpul utilizatorului i de finalizare a experimentului cu consola de învățare
 n = numărul total de măsurători.

Se compară T_{mc} cu T_{mM} , timpul cu valoarea cea mai mic, fiind timpul cel mai bun timp.

b) Analiza preciziei

Se consideră formulele de calcul:

$$S_{mM} = \sum_n \frac{p_{iM}}{n} \quad (5.3.)$$

unde:

S_{mM} = scorul mediu pentru precizia de îndeplinire a testului cu Myo Armband;
 p_{iM} = scorul de precizie al utilizatorului i de finalizare a experimentului cu Myo Armband;
 n = numărul total de măsurători.

și

$$S_{mc} = \sum_n \frac{p_{ic}}{n} \quad (5.4.)$$

unde:

S_{mc} = scorul mediu pentru precizia de îndeplinire a testului, cu consola de învățare;
 p_{ic} = scorul de precizie al utilizatorului i de finalizare a experimentului cu consola de învățare;
 n = numărul total de măsurători.

Se compara S_{mc} cu S_{mM} , scorul cel mai mare fiind scorul cel mai bun.

c) Analiza multicriterială echivalentă

Analiza multicriterială echivalentă este o metodă de comparație având mai mulți parametri de care se ține cont. În cazul nostru, parametri vor fi scorul de precizie, cu ponderea 50% și timpul, cu ponderea 50%.

Având în vedere că timpul se consideră mai bun, cel cu valoarea mai mică, iar scorul de precizie se consideră mai bun, cel cu valoarea mai mare, este necesară normalizarea valorilor după formula:

$$N_{t_i} = \frac{\min_{i=1}^n t_i}{t_i} \quad (5.5.)$$

unde:

N_{t_i} = valoarea normalizată a timpului i ;

t_i = timpul i pentru care se face normalizarea valorii, dintre toți timpii măsurați atât cu Myo Armband cât și cu consola de învățare;

n = numărul total de măsurători.

Ținând cont de faptul că scorul de precizie cel mai bun, este cel cu valoarea cea mare, este necesară normalizarea valorilor după formula:

$$N_{p_i} = \frac{p_i}{\max_{i=1}^n p_i} \quad (5.6.)$$

unde:

N_{p_i} = valoarea normalizată a scorului de precizie i

p_i = scorul de precizie i pentru care se face normalizarea valorii, dintre toate scorurile de precizie i , măsurate atât cu Myo Armband cât și cu consola de învățare;

n = numărul total de măsurători.

Astfel, pentru a face analiza comparativă se va face media aritmetică a valorilor normalizate și se vor pondera 50% timp, 50% precizie, folosind formulele:

$$S_M = \sum_n^1 \frac{N_{tM_i}}{n} \times 0.5 + \sum_n^1 \frac{N_{pM_i}}{n} \times 0.5 \quad (5.7.)$$

unde:

S_M = scorul analizei multicriteriale echivalente pentru Myo Armband;

N_{tM_i} = valoarea normalizată a timpului i , măsurat cu Myo Armband;

N_{pM_i} = valoarea normalizată a scorului de precizie i , măsurat cu Myo Armband;

n = numărul total de măsurători.

$$S_C = \sum_n^1 \frac{N_{tC_i}}{n} \times 0.5 + \sum_n^1 \frac{N_{pC_i}}{n} \times 0.5 \quad (5.8.)$$

unde:

S_C = scorul analizei multicriteriale echivalente pentru consola de învățare;

N_{tC_i} = valoarea normalizată a timpului i , măsurat cu consola de învățare;

N_{pC_i} = valoarea normalizată a scorului de precizie i , măsurat cu consola de învățare;

n = numărul total de măsurători.

Se vor compara S_M cu S_C iar scorul cel mai mare va determina dispozitivul cel mai bun.

Pentru a deplasa efectul final, utilizatorul poate realiza trei gesturi, pe baza parametrilor pronăție-supinație, flexie-extensie, aducție-abducție (roll, pitch, yaw), așa cum se poate observa în Fig. 76. Pentru a prehensa un obiect, utilizatorul va trebui să strângă pumnul, iar pentru a da drumul la obiect, acesta trebuie să realizeze gestul degete extinse.



Fig. 76 - Corelare mână - robot UR10e

5.1. Testul 1 - Preluarea și plasarea a trei obiecte (paralelipiped din plastic)

În cadrul acestui experiment, utilizatorul va fi instruit să preia, pe rând, cele trei obiecte amplasate pe masa de lucru iar mai apoi să le poziționeze cât mai aproape de punctul central al țintei aflate pe masa (Fig. 77). Acesta trebuie să obțină raportul timp-precizie cel mai bun.

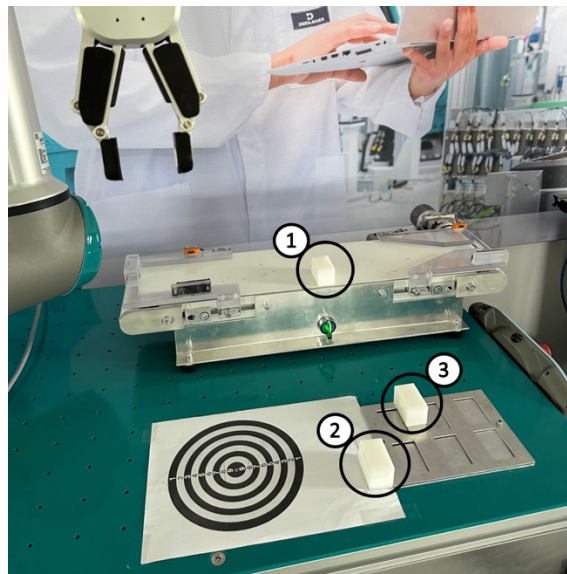


Fig. 77 – Testul 1 – Preluarea și plasarea obiectelor nr. 1, nr. 2 și nr. 3

Testul 1 - Preluarea și plasarea a trei obiecte (paralelipiped din plastic) - 85

Prin acest test se dorește observarea corelării timp-precizie folosind atât brățara Myo Armband, cât și cu ajutorul consolei de învățare.

Pentru a determina precizia după fiecare poziționare se va realiza un scor, pe baza punctelor adunate de către utilizator, după plasarea obiectului pe ținta gradată.

Astfel, scorul de precizie va fi egal cu suma punctelor pe care este poziționat obiectul, împărțită la numărul lor:

Exemplu 1: În Fig. 78 se poate observa cum obiectul atinge punctele 9,8,7,6 deci scorul de precizie = $(9+8+7+6)/4 = 7.5$

Exemplu 2: În Fig. 79 se poate observa cum obiectul atinge punctele 7,6,5,4 deci scorul de precizie = $(7+6+5+4)/4 = 5,5$



Fig. 78 - Exemplul 1 - Calcul puncte



Fig. 79 - Exemplul 2 - Calcul puncte

5.2. Testul 2 - Atingerea celor patru repere (bețe)

În cadrul acestui test se va urmări doar viteza de atingere a celor patru repere (bețe), întrucât bețele fiind foarte subțiri, precizia de atingere este obligatorie, astfel utilizatorii vor fi nevoiți să fie atenți. Aceștia vor fi instruiți să atingă cele patru bețe (Fig. 80) într-un timp cât mai scurt, atât cu brățara Myo Armband, cât și cu consola de învățare UR10e. Fiecare test va fi cronometrat și consemnat într-un tabel. La începutul fiecărui test toate obiectele vor fi în același loc și toți utilizatorii vor pleca din aceeași poziție a efortului final.



Fig. 80 - Testul 2 - Atingerea celor patru repere (bețe)

5.3. Testul 3 – Preluarea și plasarea obiectului (sticlă PET) în cutie

Acest test vizează punerea utilizatorului într-o situație mai dificilă, și anume, preluarea unei sticle PET și poziționarea acesteia într-o cutie unde există o țintă (Fig. 81). Utilizatorii sunt instruiți să obțină un raport timp-precizie cât mai bun. Toate experimentele sunt cronometrate, iar pentru a măsura precizia de poziționare, utilizatorul va primi un punct dacă a poziționat sticla în cutie, în interiorul țintei și 0 dacă sticla atinge suprafața albă din exteriorul țintei. La începutul fiecărui experiment, toate obiectele vor fi în același loc și toți utilizatorii vor pleca din aceeași poziție a efortului final.

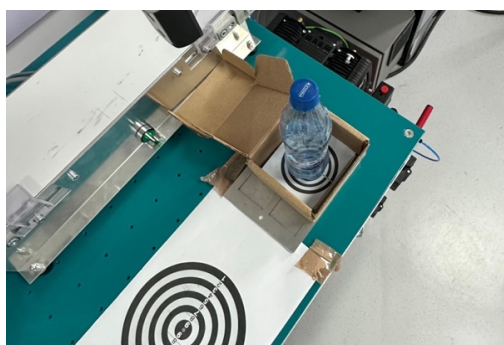


Fig. 81 - Testul 3 – Preluare și plasare obiect în cutie

6. REZULTATE EXPERIMENTALE

6.1. Rezultatele chestionarelor

Chestionarele au fost aplicate imediat după finalizarea celor trei teste pe un eșantion de 13 persoane cu vârstă medie de 26.5 ani. Un element esențial este faptul că utilizatorilor nu li s-au comunicat măsurătorile, iar completarea chestionarului s-a făcut prin estimarea utilizatorului, prin propriile simțuri.

Astfel, chestionarul aplicat este prezentat în Tabelul 5, acesta fiind structurat în trei părți: primele două analizând individual brățara Myo Armband și consola de învățare, iar a treia, fiind poate cea mai relevantă, utilizatorii trebuie să aleagă clar între cele două dispozitive, în funcție de ușurința mișcării robotului, precizia de poziționare, timpul de programare și satisfacția generală de utilizare.

Tabelul 5 - Chestionar feedback rezultate experimentale

Nume					
Vârsta					
Secțiune 1. Evaluarea Myo Armband					
1.1. Ușurința mișcării robotului	1	2	3	4	5
1.2. Precizia de poziționare	1	2	3	4	5
1.3. Timpul de programare	1	2	3	4	5
1.4. Satisfacția generală de utilizare	1	2	3	4	5
Secțiune 2. Evaluarea consolei de învățare					
2.1. Ușurința mișcării robotului	1	2	3	4	5
2.2. Precizia de poziționare	1	2	3	4	5
2.3. Timpul de programare	1	2	3	4	5
2.4. Satisfacția generală de utilizare	1	2	3	4	5
Secțiune 3. Alegerea dispozitivului preferat					
3.1. Ușurința mișcării robotului	Myo Armband			Consolă de învățare	
3.2. Precizia de poziționare	Myo Armband			Consolă de învățare	

3.3. Timpul de programare	Myo Armband	Consolă de învățare
3.4. Satisfacția generală de utilizare	Myo Armband	Consolă de învățare

*Nota: 1 = foarte rău, 2 = rău, 3= mediocru 4 = bine, 5 = foarte bine

6.1.1. Ușurința mișcării robotului

Rezultate chestionarului în privința ușurinței mișcării robotului se pot observa în Fig. 82. Analizând graficele, se pot trage următoarele concluzii:

- într-o proporție covârșitoare, utilizatorii au considerat că, dacă ar avea de ales între Myo Armband și consola de învățare, 92% ar alege Myo Armband atunci când vine vorba de ușurința mișcării robotului UR10e;
- la nivel individual, 85% au considerat Myo Armband foarte bun și bun, și doar 15% mediocru, iar aspectul cel mai interesant este că nimeni nu a ales varianta rău sau foarte rău; Pe de altă parte, în privința consolei de învățare, 8% au considerat dispozitivul rău;
- în câmpul de observații s-a menționat faptul că există o dificultate în a folosi ecranul tactil al consolei de învățare.

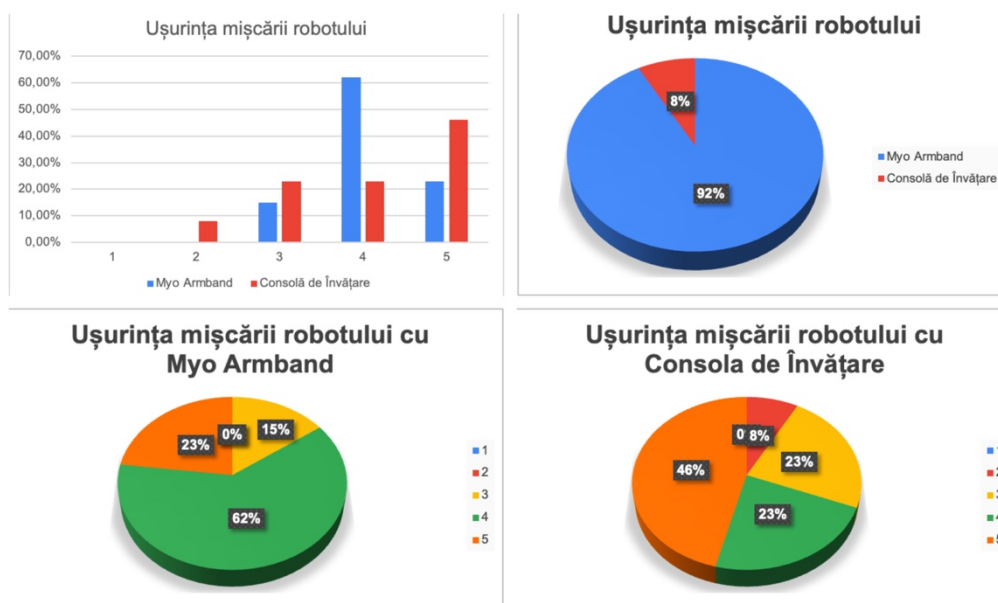


Fig. 82 - Rezultate chestionare - ușurința deplasării

6.1.2. Precizia de poziționare

Rezultatele chestionarului în privința preciziei de poziționare se pot observa în Fig. 83. Analizând graficele, se pot trage următoarele concluzii:

- 85% dintre utilizatori au considerat că din punct de vedere al preciziei de poziționare, este mai utilă consola de învățare decât Myo Armband, fiind aproape exact în antiteză cu rezultatele privind ușurința mișcării robotului;
- La nivel individual, 92% dintre utilizatori au considerat consola de învățare ca fiind foarte bună din punct de vedere al preciziei de poziționare și 8% doar bună; Pe de altă parte, 46% dintre utilizatori au considerat Myo Armband ca fiind mediocru din punct de vedere al preciziei și 15% au ales că este rău;
- în cadrul observațiilor s-a menționat că mâna este foarte sensibilă și este destul de greu să fie menținută fixă în cazul Myo Armband.

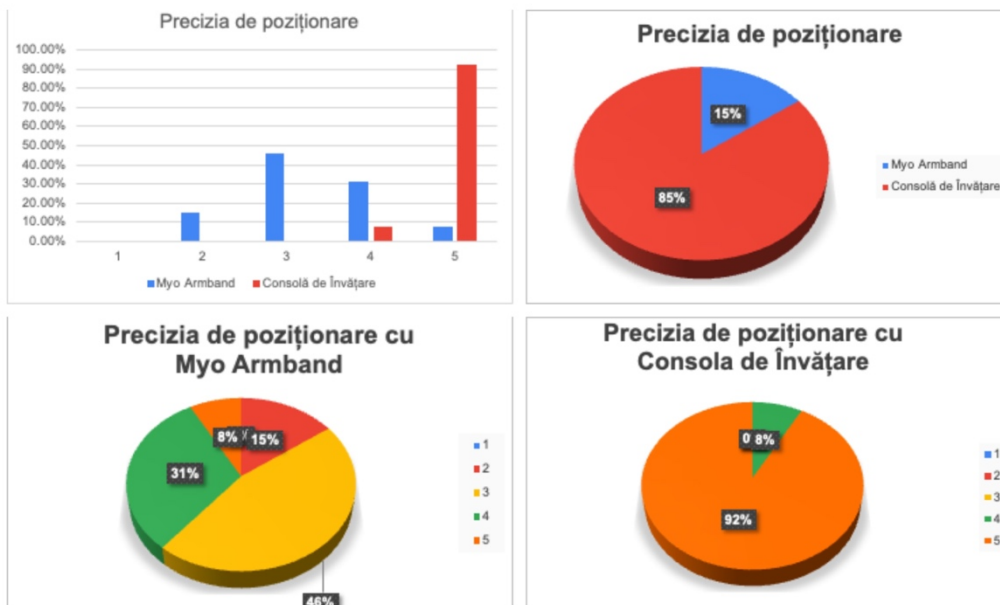


Fig. 83 - Rezultate chestionare - precizia de poziționare

6.1.3. Timpul de programare

În această situație utilizatorii au dat feedback privind modul în care au scris comenzile pe consola de învățare pentru a face mișcările MoveJ și MoveL și gesturile stânga și dreapta realizate cu mâna, folosind Myo Armband.

Rezultatele chestionarului în privința timpului de programare se pot observa în Fig. 84. Analizând graficele, se pot trage următoarele concluzii:

- într-o proporție covârșitoare, 92% dintre cei chestionați preferă să realizeze gesturi pentru a salva poziții în care să se deplaseze robotul, decât să scrie instrucțiuni în consola de învățare;
- la nivel individual, 77% dintre utilizatori consideră Myo Armband bună și foarte bună de a salva poziții în care să se deplaseze robotul, cu ajutorul

90 - REZULTATE EXPERIMENTALE

- gesturilor și 0% au considerat rău sau foarte rău acest dispozitiv, din acest punct de vedere;
- tot la nivel individual, în cazul consolei de învățare, părerile au fost împărțite, cel mai mare scor fiind 3-Mediocru, cu 31% dintre utilizatori; în acest caz, există și 23% care au considerat rău și foarte rău acest dispozitiv, pentru programare;
 - în cadrul rubricii de observații au fost utilizatori care au blamat ecranul tactil care nu era foarte ușor de utilizat, motiv pentru care au existat 8% dintre respondenți care au spus că este foarte rău.

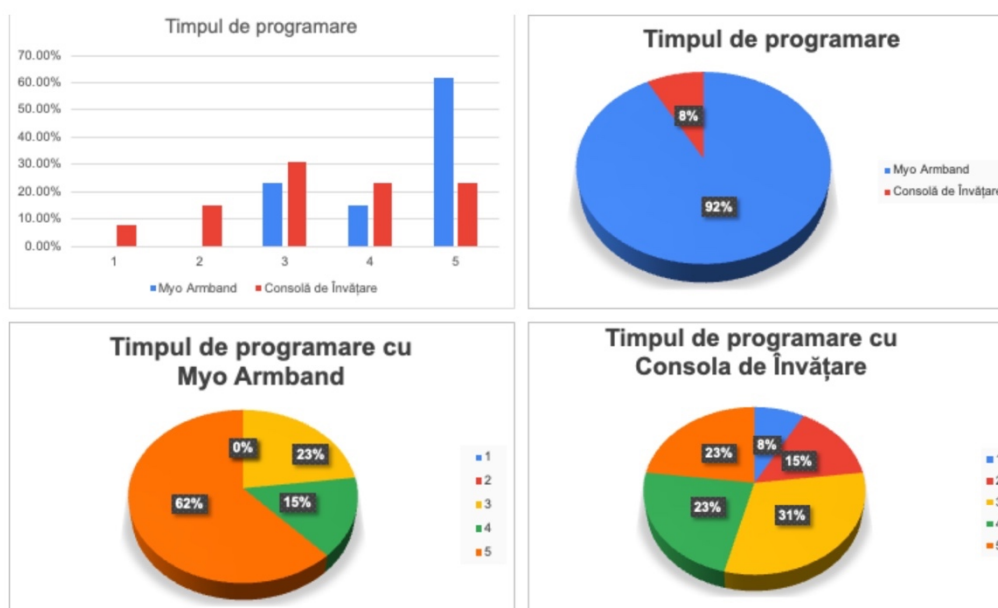


Fig. 84 – Rezultate chestionare - timpul de programare

6.1.4. Satisfacția generală de utilizare

Această rubrică a vizat alegerea unui dispozitiv per ansamblu, cu toate avantajele și dezavantajele sale. Rezultatele chestionarului în privința satisfacției generale de utilizare se pot observa în Fig. 85. Analizând graficele, se pot trage următoarele concluzii:

- Cu toate că cel mai mare dezavantaj al brățării Myo Armband este precizia de poziționare, iar chestionarele au validat acest aspect, 85% au considerat brățara Myo Armband mai bună în utilizare și doar 15% au ales consola de învățare. Acest aspect validează faptul că experiența de utilizare este cea mai importantă pentru utilizatori;
- La nivel individual, 85% dintre utilizatori au răspuns că satisfacția generală de utilizare este foarte bună, în timp ce 15% au răspuns că este bună; pe de altă parte, dacă ne uităm la satisfacția generală de utilizare a consolei de învățare, putem să observăm cum părerile sunt uniform distribuite, 46%

consideră consola de învățare foarte bună și bună, 46% o consideră mediocră și rea, iar 8% o consideră foarte rea.

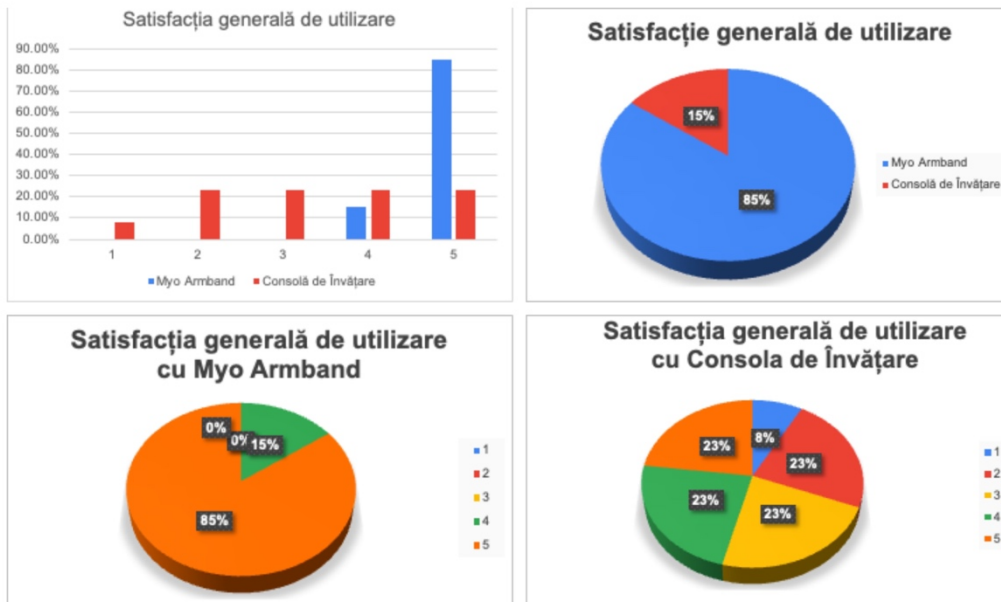


Fig. 85 - Rezultate chestionare - satisfacția generală de utilizare

6.1.5. Concluzii

Se poate observa că la trei din patru secțiuni (ușurința mișcării robotului, viteza de programare și satisfacția generală de utilizare) Myo Armband a fost ales ca dispozitiv mai bun, cu o majoritate evidentă și clară (peste 85%), singura secțiune care nu a ajuns la nivelul așteptărilor fiind precizia de poziționare, unde consola de învățare se poziționează foarte bine.

Din rezultate se observă că experiența de utilizare și satisfacția de a lucra cu brățara au învins dezavantajul preciziei, utilizatorii preferând poate să depună efort mai mare pentru o poziționare precisă, dar să aibă un dispozitiv ușor de utilizat.

Unele rezultate în privința consolei de învățare au fost influențate de defectul ecranului tactil, care a început să nu mai răspundă optim, deși robotul UR10e are doar șase luni de utilizare.

6.2. Rezultatele măsurătorilor

Pentru cele trei teste se vor analiza atât timpii și precizia individual, cât și varianta optimă timp-precizie, folosind analiza multicriterială echivalentă, unde se pretează această analiză. În cadrul acestui tip de analiză multicriterială se vor folosi ponderi echivalente pentru timp și precizie (50% timp, 50% precizie).

6.2.1. Testul 1 - Preluarea și plasarea a trei obiecte (paralelipiped din plastic)

a. Analiza timpilor

În Fig. 86 se poate observa timpul mediu aferent fiecărui obiect în parte, atât cu Myo Armband, cât și cu consola de învățare. Analizând datele, distingem faptul că în cadrul primului experiment de preluare a obiectului 1 de pe banda transportoare, realizarea acestei operațiuni s-au efectuat, în medie, cel mai rapid cu Myo Armband, la o diferență de 13,23 secunde, iar restul obiectelor au fost preluate, în medie, mai rapid cu consola de învățare.

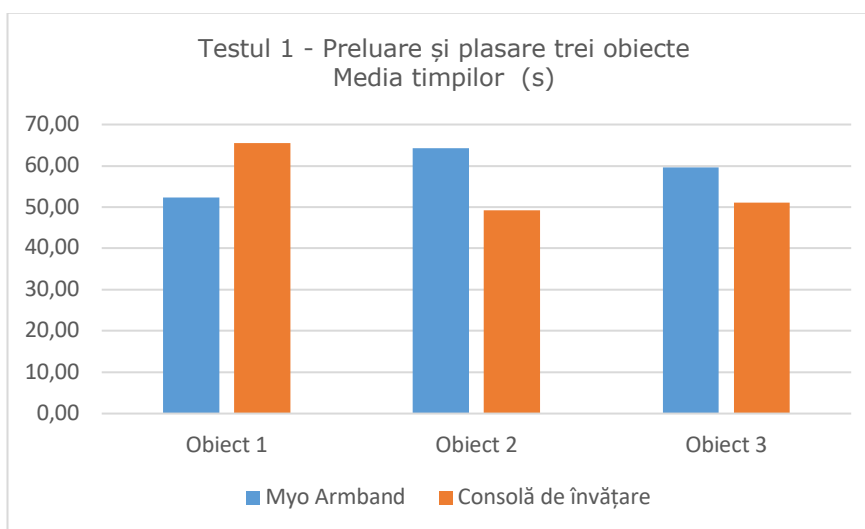


Fig. 86 - Analiza timpilor - Testul 1 - Preluare și plasare trei obiecte

Un alt element analizat a fost cel mai mic timp realizat cu oricare dintre cele două dispozitive și cel mai mare timp, pentru preluarea și plasarea celor trei obiecte. Analizând Tabelul 6, putem observa că timpul cel mai bun de preluare și plasare a obiectului 1 (cel plasat pe banda transportoare) a fost realizat cu Myo Armband, la o diferență de 10 secunde de cel mai bun timp obținut cu consola de învățare. Totodată, cel mai mare timp a fost realizat tot cu Myo Armband, la o diferență de 2 secunde de cel mai mare timp realizat cu consola de învățare.

Tabelul 6 - Analiza timpilor de preluare și plasare a obiectului 1

Indicator	Timp preluare-plasare obiect 1 (s)	
	Myo Armband	Consolă de învățare
Medie	52,31	65,54
Min	25	35
Max	100	98

Analizând Tabelul 7, putem observa că timpul cel mai bun de preluare și plasare a obiectului 2 (cel de lângă suportul de metal), a fost realizat cu consola de învățare, la o diferență de 8 secunde de cel mai bun timp obținut cu Myo Armband. Totodată, cel mai mare timp a fost realizat tot cu consola de învățare, la o distanță de 2 secunde de cel mai mare timp realizat cu Myo Armband. În cadrul acestui tip de test, utilizatorii au întâmpinat dificultăți la preluarea acestui obiect cu Myo Armband întrucât era lipit de suportul metalic, iar la orice mișcare greșită, robotul UR10e intra în coliziune cu acesta.

Tabelul 7 - Analiza timpilor de preluare și plasare a obiectului 2

Indicator	Timp preluare-plasare obiect 2 (s)	
	Myo Armband	Consolă de învățare
Medie	64,23	49,23
Min	37	29
Max	96	98

Analizând Tabelul 8, putem observa că timpul cel mai bun de preluare și plasare a obiectului 3 (cel plasat peste suportul metalic), a fost realizat cu Myo Armband, la o diferență de 2 secunde de cel mai bun timp obținut cu consola de învățare. Totodată, cel mai mare timp a fost realizat tot cu Myo Armband, la o diferență de 7 secunde de cel mai mare timp realizat cu consola de învățare.

Tabelul 8 - Analiza timpilor de preluare și plasare a obiectului 3

Indicator	Timp preluare-plasare obiect 3 (s)	
	Myo Armband	Consolă de învățare
Medie	59,62	51,08
Min	29	31
Max	91	84

b. Analiza preciziei

În Fig. 87 se prezintă graficul mediei preciziei de poziționare, cel mai mare scor al preciziei putând fi 8. Astfel, se observă cum, în toate cele trei cazuri, media preciziei de poziționare a fost mai bună în cazul consolei de învățare.

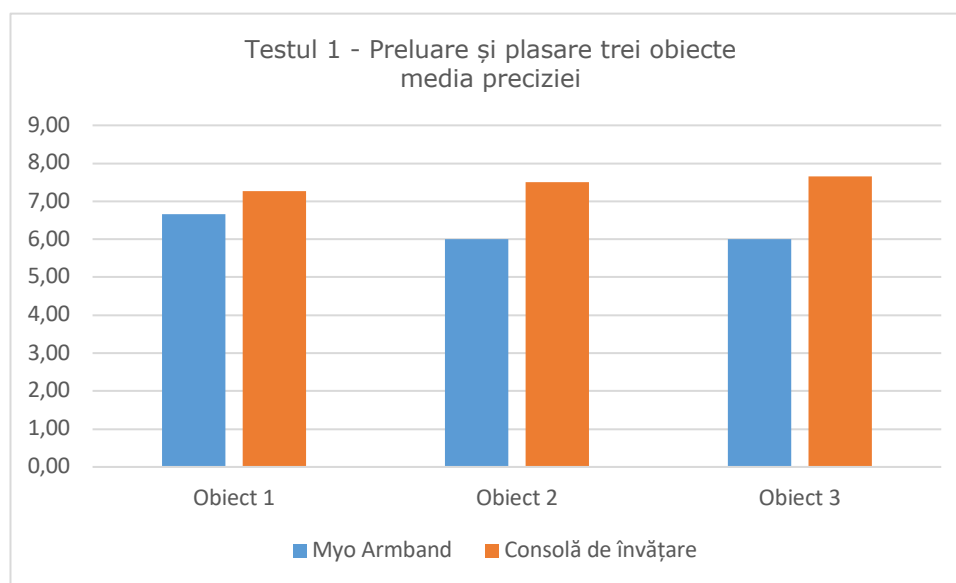


Fig. 87 - Media preciziei - Testul 1 - Preluare și plasare trei obiecte

Un alt element analizat a fost cea mai bună precizie realizată cu oricare dintre cele două dispozitive și cea mai redusă precizie, pentru preluare și plasarea celor trei obiecte. Analizând Tabelul 9, putem să observa că cea mai bună precizie de preluare și plasare a obiectului 1 (cel plasat pe banda transportoare) a fost realizată cu consola de învățare la un scor aproape de scorul maxim. În privința celei mai reduse precizii, ambele dispozitive au obținut un scor egal, însă în cazul consolei de învățare a fost doar o excepție, restul scorurilor fiind foarte bune. Cel mai bun scor a fost scorul maxim posibil, obținut cu ambele dispozitive.

Tabelul 9 - Analiza preciziei de preluare și plasare a obiectului 1

Indicator	Precizie preluare-plasare obiect 1	
	Myo Armband	Consolă de învățare
Medie	6,65	7,35
Min	3,5	3,5
Max	8	8

Analizând Tabelul 10, putem observa că cea mai bună precizie de preluare și plasare a obiectului 2 (cel de lângă suportul metalic), a fost realizată cu consola de învățare, fiind obținut scorul maxim. Cea mai redusă precizie a fost obținută de Myo Armband cu un scor de 3.

Tabelul 10 - Analiza preciziei de preluare și plasare a obiectului 2

Indicator	Precizie preluare-plasare obiect 2	
	Myo Armband	Consolă de învățare
Medie	6,00	7,50
Min	3	6,5
Max	7,5	8

Analizând Tabelul 11, putem observa că cea mai bună precizie de preluare și plasare a obiectului 3 (cel plasat pe suportul metalic), a fost realizată cu consola de învățare, fiind obținut scorul maxim. Cea mai redusă precizie a fost obținută de Myo Armband cu un scor de 1.

Tabelul 11 - Analiza preciziei de preluare și plasare a obiectului 3

Indicator	Precizie preluare-plasare obiect 3	
	Myo Armband	Consolă de învățare
Medie	6,00	7,65
Min	1	6,5
Max	7,5	8

c. Analiza multicriterială echivalentă

Analizând Tabelul 12, putem observa, în cazul obiectului 1, faptul că după normalizarea valorilor de precizie și timp, și aplicarea ponderilor de 50%, scorul total al Myo Armband este mai bun decât cel al consolei de învățare, însă cu o diferență mică. În aceste condiții, cel mai bun dispozitiv este Myo Armband.

Tabelul 12 - Analiza multicriterială echivalentă de preluare și plasare a obiectului 1

Indicator	Normalizare valori				Analiza multicriterială echivalentă	
	Timp		Precizie		Scor total	
	Myo Armband	Consola învățare	Myo Armband	Consola învățare	Myo Armband	Consola învățare
Medie	0,53	0,42	0,83	0,92	0,68	0,67

În Tabelul 13 sunt prezentate valorile normalizate ale timpului și ale preciziei, în cazul obiectului 2, iar după aplicarea ponderilor de 50% timp, 50% precizie, scorul total al consolei de învățare este cel mai bun. Astfel, în acest caz, consola de învățare este cel mai bun dispozitiv.

Tabelul 13 - Analiza multicriterială echivalentă de preluare și plasare a obiectului 2

Indicator	Normalizare valori				Analiza multicriterială echivalentă	
	Timp		Precizie		Scor total	
	Myo Armband	Consola învățare	Myo Armband	Consola învățare	Myo Armband	Consola învățare
Medie	0,49	0,66	0,75	0,94	0,62	0,80

Tabelul 14 ne prezintă analiza multicriterială echivalentă (ponderi 50% pentru ambele componente de timp și precizie) pentru obiectul 3. Astfel, după normalizarea valorilor, scorul cel mai bun este cel al consolei de învățare

Tabelul 14 - Analiza multicriterială echivalentă de preluare și plasare a obiectului 3

Indicator	Normalizare valori				Analiza multicriterială echivalentă	
	Timp		Precizie		Scor total	
	Myo Armband	Consola învățare	Myo Armband	Consola învățare	Myo Armband	Consola învățare
Medie	0,55	0,62	0,75	0,96	0,65	0,79

6.2.2. Testul 2 - Atingerea celor patru repere (bețe)

În cazul acestui test, precizia nu a mai fost măsurată întrucât era condiție obligatorie pentru îndeplinirea testului, singurul indicator măsurat fiind timpul de îndeplinire al testului.

Analizând Tabelul 15 putem observa cum, în medie, utilizatorii au îndeplinit sarcina mai rapid cu consola de învățare, însă timpul cel mai bun a fost realizat cu Myo Armband. Timpul cel mai mare a fost realizat tot cu Myo Armband, motiv care duce spre concluzia că rezultatele date de acest dispozitiv depind de îndemânarea utilizatorului.

Tabelul 15 - Analiza timpilor de atingere a celor patru bețe (s)

Indicator	Timp atingerea celor patru bețe (s)	
	Myo Armband	Consolă de învățare
Medie	67,69	58,85
Min	39,00	43,00
Max	120,00	86,00

6.2.3. Testul 3 - Preluarea și plasarea unui obiect (sticlă PET) în cutie

a. Analiza timpilor

După cum se poate observa în Tabelul 16, în medie, cel mai bun timp de preluare și plasare a unei sticle PET în cutie a fost obținut cu Myo Armband. Cel mai rapid timp dintre toate dispozitivele a fost obținut cu Myo Armband (15 secunde), iar timpul cel mai mare a fost obținut cu consola de învățare (124 secunde).

Tabelul 16 - Analiza timpilor de preluare și plasare a unei sticle în cutie

Indicator	Timp preluare și plasare sticlă în cutie (s)	
	Myo Armband	Consolă de învățare
Medie	54,23	60,08
Min	15,00	33,00
Max	104,00	124,00

b. Analiza preciziei

După cum se poate observa în Tabelul 17, în medie, cel mai bun scor de precizie pentru preluarea și plasarea unei sticle în cutie a fost obținut cu consola de învățare. Întrucât în cadrul acestui tip de test nu a existat o gradație foarte detaliată, ci doar s-a verificat dacă s-a plasat complet în interiorul țintei sau nu, în cazul ambelor dispozitive am avut situații în care s-a plasat sticla atât perfect, cât și imperfect.

În cadrul acestui test s-a constatat faptul că utilizatorii care își doreau să obțină un timp foarte bun, greșeau din punct de vedere al preciziei, și invers.

Tabelul 17 - Analiza preciziei de preluare și plasare a unei sticle în cutie

Indicator	Precizie preluare și plasare sticlă în cutie	
	Myo Armband	Consolă de învățare
Medie	0,85	0,92
Min	0,00	0,00
Max	1,00	1,00

c. Analiza multicriterială echivalentă

Analizând Tabelul 18 putem observa faptul că după normalizarea valorilor de precizie și timp, și aplicarea ponderilor de 50%, scorul total al Myo Armband este mai bun decât cel al consolei de învățare, însă cu o diferență extrem de mică.

Tabelul 18 - Analiza multicriterială echivalentă de preluare și plasare a unei sticle în cutie

Indicator	Normalizare valori				Analiza multicriterială echivalentă	
	Timp		Precizie		Scor total	
Dispozitiv	Myo Armband	Consolă Învățare	Myo Armband	Consolă Învățare	Myo Armband	Consolă Învățare
Medie	0,3629359	0,282535	0,846154	0,923077	0,60454488	0,60280577

Datele neprelucrate ale tuturor chestionarelor și ale măsurătorilor testelor se pot consulta în Anexa 2.

6.2.4. Concluzii

Aceste teste au fost un real ajutor pentru analiza comparativă a celor două dispozitive de control a roboților industriali întrucât au oferit cadrul de a putea analiza toate aspectele pozitive și negative ale fiecărui dispozitiv.

La nivel individual, atunci când analizăm timpul de operare, aproape în cadrul tuturor testelor, cel mai rapid timp a fost realizat cu Myo Armband, însă atunci când s-a realizat media timpilor, atât la preluarea și plasarea obiectelor 2 și 3, cât și la atingerea reperelor (bețelor), consola de învățare a fost cel mai rapid dispozitiv. Pe de altă parte, în privința preciziei, consola de învățare a fost cea mai bună, în cadrul tuturor testelor.

Dacă ne uităm la analiza multicriterială realizată cu ponderile timp-50% și precizie-50%, observăm că, pentru preluarea și plasarea în situații mai complexe și trasee mai lungi, cum este cazul preluării și plasării sticlei în cutie sau a obiectului 1 (cel de pe banda transportoare), Myo Armband este cel mai bun dispozitiv, iar pentru trasee simple și mai scurte, consola de învățare a fost cea mai bună.

7. CONCLUZII. CONTRIBUȚII PERSONALE. DIRECȚII VIITOARE DE CERCETARE

7.1. Concluzii finale

Lucrarea prezintă o modalitate alternativă și neconvențională de a conduce și programa un robot industrial cu ajutorul mâinii, folosind brățara Myo Armband. În acest demers s-au realizat activități de cercetare pe un robot KUKA KR15 și Universal Robots UR10e, fiind testat controlul acestora cu o brățară Myo Armband din care se preiau atât valorile de orientare ale brățării, cât și activitatea musculară din antebraț, care determină identificarea de gesturi. Rezultatele cercetării arată că UR10e este mai potrivit pentru astfel de activități dat fiind faptul că oferă o funcție specifică, care ține cont de mișcările viitoare și permite realizarea de mișcări fără șocuri, în comparație cu KUKA KR15 unde mișcările erau frânate, îndeplinind astfel obiectivul O1 al cercetării.

După stabilirea soluției optime de control, s-a procedat la realizarea a trei teste, atât cu Myo Armband, cât și cu consola de învățare, prin care s-au extras date privind timpul de efectuare al testelor și precizia, îndeplinind astfel obiectivul O2 al lucrării. După finalizarea tuturor testelor, utilizatorii au completat un chestionar prin care și-au exprimat părerea privind performanțele acestor dispozitive, fără a li se comunica rezultatele măsurărilor pentru a nu influența percepția proprie asupra experienței de utilizare. Prin acest demers s-a îndeplinit și obiectivul O3.

O vedere de ansamblu se poate contura atunci când comparăm rezultatele chestionarelor cu cele ale măsurărilor. Putem observa că, deși la toate testele de precizie consola de învățare a fost mai precisă sau că la unele teste scorul multicriterial precizie-timp a fost mai bun în cadrul consolei de învățare, la final, după aplicarea chestionarelor utilizatorii au ales Myo Armband ca dispozitiv preferat. Acest lucru demonstrează faptul că experiența de utilizare este mai importantă pentru om, decât timpul de realizare sau precizia.

Un alt aspect interesant este că, la nivel individual, în mare parte, răspunsul dat în chestionar a reflectat rezultatul măsurat, în sensul că utilizatorii care au spus că Myo Armband este mai ușor de folosit, au obținut și un timp mai bun la teste, iar cei care au spus că precizia consolei de învățare este mai bună, au obținut un scor bun de precizie, dar există și un număr de utilizatori care au ales Myo Armband ca dispozitiv mai bun de precizie, cu toate că au obținut un scor de precizie mai bun cu consola de învățare sau utilizatori care au spus că Myo Armband este mai ușor de utilizat pentru deplasarea robotului, dar în cadrul rezultatelor măsurate, au obținut timp net mai mare decât atunci când au folosit consola de învățare. Toate aceste contradicții demonstrează aceeași ipoteză, că pentru utilizator experiența de utilizare este mai importantă, iar Myo Armband s-a ridicat la nivelul așteptărilor în privința acestui aspect. Această analiză a determinat îndeplinirea obiectivului cercetării O4.

Totodată, pe parcursul activității de cercetare am întâmpinat și probleme, cum ar fi lipsa unui protocol de comunicare mai eficient cu robotul KUKA KR15, dat fiind faptul că robotul este unul de generație veche. Aceasta a fost cea mai mare provocare, și anume, lipsa unui robot de generație nouă care să permită testarea

soluției propuse. În cadrul stagiului am reușit rezolvarea acestei probleme cu ajutorul finanțării companiei DRĂXLMAIER, iar astăzi laboratorul este dat în exploatare și folosit de către studenți pentru activități didactice, dar și de către cadre didactice, pentru activități de cercetare, fiind cel mai modern laborator de robotică din cadrul Universității Politehnica Timișoara.

O altă problemă întâmpinată a fost controlarea roboților atât din punct de vedere al poziționării pe axele x , y , z , cât și al orientării efectorului final. Soluția cu două brățări, pe cele două mâini, nu s-a ridicat la nivelul așteptărilor întrucât atunci când se făceau mișcări mai ample ale unei mâini, involuntar se mișca și a doua mână din cauza legăturii corporale ale celor două membre.

7.2. Contribuții personale

Contribuțiile personale în cadrul acestei activități de cercetare pot fi enumerate, din punct de vedere al activității științifice:

- Crearea unui program C#, pe baza bibliotecii Myo Sharp[102], unde, printr-o interfață vizuală, se preiau date de la o brățara Myo Armband și se transmit prin protocolul de comunicare serial către robotul KUKA KR15;
- Realizarea programului KUKA de preluare a informațiilor de la brățara Myo și mișcarea robotului în funcție de valorile primite, cât și închiderea și deschiderea dispozitivului de prehensiune, în funcție de gestul trimis de brățara Myo Armband;
- Realizarea de încercări de control a robotului KUKA KR15 cu o brățară Myo Armband;
- Crearea unui sistem de operare virtual și conectarea concomitentă a două brățări Myo Armband, realizarea unui program C#, pe baza bibliotecii Myo Sharp[102], unde printr-o interfață vizuală se preiau date de la două brățări Myo Armband care se transmit prin protocolul de comunicare serial, către robotul KUKA KR15;
- Realizarea programului KUKA de preluare a informațiilor de la două brățări Myo și mișcarea robotului, în funcție de valorile primite, atât pe axele x , y , z , cât și orientările R_x , R_y , R_z , dar și închiderea și deschiderea dispozitivului de prehensiune, în funcție de gestul trimis de la una din cele două brățări Myo Armband;
- Realizarea de încercări de control a robotului KUKA KR15 cu două brățări Myo Armband;
- Realizarea unui program C#, pe baza bibliotecii Myo Sharp[102], care preia informațiile de la Myo Armband, deschide o comunicare în rețeaua locală (localhost) și transmite valorile către un program Python;
- Realizarea unui program Python, pe baza bibliotecii RTDE [104], [108], care citește pe rețeaua locală (localhost) valorile de la Myo Armband, prin intermediul programului C#, le interpretează, procesează și apoi transmite valori de poziționare și valori privind deschiderea sau închiderea dispozitivului de prehensiune, către UR10e, conectat în rețea cu sistemul desktop pe care rulează;
- Realizarea unui program pentru robotul UR10e, pe baza bibliotecii RTDE [104], [108] de citirea a valorilor transmise prin regiștrii interni și poziționarea robotului în consecință, precum și acționarea dispozitivului de prehensiune, dacă e cazul;

- Conceperea metodologiei de testare și implicit a celor trei teste prin care se măsoară parametrii de timp și scor de precizie, precum și a ecuațiilor matematice privind analiza comparativă a celor doi parametri;
- Colectarea, procesarea și interpretarea rezultatelor experimentale, atât prin realizarea de grafice, cât și prin tabele;
- Conceperea chestionarului de feedback pentru utilizatorii celor două dispozitive: Myo Armband și consola mobilă de învățare;
- Interpretarea rezultatelor chestionarelor și corelarea răspunsurilor cu rezultatele măsurătorilor;
- Poate una dintre contribuțiile personale esențiale este realizarea unei modalități de a programa un robot industrial doar cu ajutorul unei mâini, dat fiind faptul că prin realizarea de gesturi se pot salva poziții ale robotului.

Pe de altă parte, contribuția proprie, în altă categorie decât activitatea științifică, se cuantifică prin inițierea demersurilor și obținerea unei finanțări de aproximativ 90 000 euro de la compania DRÄXLMAIER prin care s-a achiziționat robotul UR10e și s-a dotat complet un nou laborator de robotică.

7.3. Direcții de cercetare viitoare

Un obiectiv important pentru activitățile viitoare de cercetare este realizarea unei modalități de control al orientării efectorului final al robotului.

Pe de altă parte, activitățile viitoare de cercetare vor viza controlul robotului UR10e cu mai multe dispozitive externe care, în funcție de modalitatea de control, pot fi:

- Kinect v2 – control prin intermediul gesturilor;
- Leap Motion – control prin intermediul gesturilor;
- Gamepad Xbox – control prin butoane;
- BitBrain – control prin intermediul activității neuronale ale creierului (prin gânduri).

Probabil cea mai provocatoare și interesantă abordare ar fi controlul robotului UR10e prin intermediul activității neuronale (prin gânduri), care poate fi realizat cu ajutorul unuia dintre dispozitivele puse la dispoziție de către cei de la BitBrain, însă pentru acest demers de cercetare este necesară o finanțare externă, întrucât un astfel de dispozitiv costă aproximativ 10 000 euro.

Ca și cercetător în acest domeniu, doresc să mă dedic acestui segment de teleoperare și control al roboților industriali, cu diverse dispozitive externe și analiza comparativă a acestor dispozitive, pentru a determina utilitatea lor în diverse aplicații.

8. BIBLIOGRAFIE

- [1] K. Capek, *Rossum's Universal Robots*, translated by Paul Selver. 1923.
- [2] T. R. Kurfess, *Robotics and automation handbook*, vol. 414. CRC press Boca Raton, FL, 2005.
- [3] I. Margolius, „The robot of Prague”, *The Friends of Czech Heritage*, 2017. <https://czechfriends.net/images/RobotsMargoliusJul2017.pdf>. (data accesării 9 iunie 2019).
- [4] K. Vijay, „K. Vijay, Introduction to robotics”. University of Pennsylvania, Philadelphia, USA.
- [5] S. B. Niku, *Introduction to robotics: analysis, control, applications*. John Wiley & Sons, 2020.
- [6] M. Vukobratovic, *Introduction to robotics*. Springer Science & Business Media, 2012.
- [7] I. Asimov, *Runaround*. New York: Street & Smith, 1942.
- [8] I. Asimov, *I, Robot*. Gnom Press, 1950.
- [9] R. V Aroca, A. Péricles, B. S. de Oliveira, L. Marcos, și G. Gonçalves, „Towards smarter robots with smartphones”, în *5th workshop in applied robotics and automation, Robocontrol, Citeseer*, 2012, pp. 1–6.
- [10] J. F. Engelberger, *Robotics in practice: management and applications of industrial robots*. Springer Science & Business Media, 2012.
- [11] I. F. O. R. (IFR), „Introduction”. https://ifr.org/img/office/Industrial_Robots_2016_Chapter_1_2.pdf. (data accesării 13 iunie 2023).
- [12] E. Mandfield, „The diffusion of industrial robots in Japan and the United States”, *Res Policy*, vol. 18, nr. 4, pp. 183–192, 1989.
- [13] I. O. f. Standardization, „ISO 8373:2012”, 2012. <https://www.iso.org/standard/55890.html> (data accesării 13 iunie 2019).
- [14] S. Shepherd și A. Buchstab, „KUKA Robots On-Site”, *Robotic Fabrication in Architecture, Art and Design 2014*, pp. 373–380, 2014, doi: 10.1007/978-3-319-04663-1_26.
- [15] P. F. Hokayem și M. W. Spong, „Bilateral teleoperation: An historical survey”, *Automatica*, vol. 42, nr. 12, pp. 2035–2057, dec. 2006, doi: 10.1016/J.AUTOMATICA.2006.06.027.
- [16] H. Jin, Q. Chen, Z. Chen, Y. Hu, și J. Zhang, „Multi-LeapMotion sensor based demonstration for robotic refine tabletop object manipulation task”, *CAAI Trans Intell Technol*, vol. 1, nr. 1, pp. 104–113, ian. 2016, doi: 10.1016/J.TRIT.2016.03.010.
- [17] D. Krupke, L. Einig, E. Langbehn, J. Zhang, și F. Steinicke, „Immersive remote grasping: Realtime gripper control by a heterogenous robot control system”, *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*, vol. 02-04-November-2016, pp. 337–338, nov. 2016, doi: 10.1145/2993369.2996345.
- [18] K. Kruusamae și M. Pryor, „High-precision telerobot with human-centered variable perspective and scalable gestural interface”, *Proceedings - 2016 9th International Conference on Human System Interactions, HSI 2016*, pp. 190–196, aug. 2016, doi: 10.1109/HSI.2016.7529630.
- [19] C. Li, C. Yang, J. Wan, A. S. K. Annamalai, și A. Cangelosi, „Teleoperation control of Baxter robot using Kalman filter-based sensor fusion”.

- <http://mc.manuscriptcentral.com/tssc>, vol. 5, nr. 1, pp. 156–167, ian. 2017, doi: 10.1080/21642583.2017.1300109.
- [20] J. I. Lipton, A. J. Fay, și D. Rus, „Baxter’s Homunculus: Virtual Reality Spaces for Teleoperation in Manufacturing”, *IEEE Robot Autom Lett*, vol. 3, nr. 1, pp. 179–186, ian. 2018, doi: 10.1109/LRA.2017.2737046.
- [21] S. Makris *et al.*, „Dual arm robot in cooperation with humans for flexible assembly”, *CIRP Annals*, vol. 66, nr. 1, pp. 13–16, ian. 2017, doi: 10.1016/J.CIRP.2017.04.097.
- [22] E. Nuño, L. Basañez, și R. Ortega, „Passivity-based control for bilateral teleoperation: A tutorial”, *Automatica*, vol. 47, nr. 3, pp. 485–495, mar. 2011, doi: 10.1016/J.AUTOMATICA.2011.01.004.
- [23] IFR, „Executive Summary World Robotics 2-2022”, 2022. https://ifr.org/img/worldrobotics/Executive_Summary_WR_Industrial_Robots_2022.pdf (data accesării 13 iunie 2023).
- [24] R. Bischoff, A. Kazi, și M. Seyfarth, „The MORPHA style guide for icon-based programming”, în *Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication*, IEEE, 2002, pp. 482–487.
- [25] Eurobots.net, „Eurobots”. https://www.eurobots.net/media/diagram/det_244_kuka_quantec_krc4_euro_bots.jpg (data accesării 13 iunie 2019).
- [26] I. Maniu, C. Rădulescu, I. Bogdanov, Ștefan Varga, V. Dolga, și V. Ciupe, *Robotică. Aplicații Robotizate*. EDITURA POLITEHNICA TIMIȘOARA, 2009.
- [27] G. Hirzinger, J. Bals, M. Otter, și J. Stelter, „The DLR-KUKA success story: robotics research improves industrial robots”, *IEEE Robot Autom Mag*, vol. 12, nr. 3, pp. 16–23, 2005.
- [28] A. Kazi, J. Bunsendal, D. Haag, R. Baum, și R. Bischoff, „Next generation teach pendants for industrial robots”, *Advances in Human-Robot Interaction*, pp. 47–66, 2005.
- [29] P. Neto, J. N. Pires, și A. P. Moreira, „3D CAD-based robot programming for the SME shop-floor”, în *20th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM*, 2010.
- [30] G. Schreiber, A. Stemmer, și R. Bischoff, „The fast research interface for the kuka lightweight robot”, în *IEEE workshop on innovative robot control architectures for demanding (Research) applications how to modify and enhance commercial controllers (ICRA 2010)*, Citeseer, 2010, pp. 15–21.
- [31] Z. Zhang, „Microsoft kinect sensor and its effect”, *IEEE multimedia*, vol. 19, nr. 2, pp. 4–10, 2012.
- [32] M. Banda, V. Ciupe, C. Moldovan, I. Maniu, și R. Kristof, „Investigations on collaborative remote control of virtual robotic manipulators by using a Kinect v2 sensor”, în *ITM Web of Conferences*, EDP Sciences, 2019, p. 03004.
- [33] H. Belhadj, S. Ben Hassen, K. Kaániche, și H. Mekki, „Kuka robot control based kinect image analysis”, în *2013 International conference on individual and collective behaviors in robotics (ICBR)*, IEEE, 2013, pp. 21–26.
- [34] Y. Yang, H. Yan, M. Dehghan, și M. H. Ang, „Real-time human-robot interaction in complex environment using kinect v2 image recognition”, în *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, IEEE, 2015, pp. 112–117.
- [35] G. Du, P. Zhang, J. Mai, și Z. Li, „Markerless kinect-based hand tracking for robot teleoperation”, *Int J Adv Robot Syst*, vol. 9, nr. 2, p. 36, 2012.

- [36] S. Shirwalkar, A. Singh, K. Sharma, și N. Singh, „Telemanipulation of an industrial robotic arm using gesture recognition with Kinect”, în *2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE)*, IEEE, 2013, pp. 1–6.
- [37] S. Moe și I. Schjølberg, „Real-time hand guiding of industrial manipulator in 5 DOF using Microsoft Kinect and accelerometer”, în *2013 IEEE RO-MAN*, IEEE, 2013, pp. 644–649.
- [38] J. L. Raheja, R. Shyam, U. Kumar, și P. B. Prasad, „Real-time robotic hand control using hand gestures”, în *2010 Second International Conference on Machine Learning and Computing*, IEEE, 2010, pp. 12–16.
- [39] R. A. El-Laithy, J. Huang, și M. Yeh, „Study on the use of Microsoft Kinect for robotics applications”, *Record - IEEE PLANS, Position Location and Navigation Symposium*, pp. 1280–1288, 2012, doi: 10.1109/PLANS.2012.6236985.
- [40] L. Cheng, Q. Sun, H. Su, Y. Cong, și S. Zhao, „Design and implementation of human-robot interactive demonstration system based on Kinect”, *Proceedings of the 2012 24th Chinese Control and Decision Conference, CCDC 2012*, pp. 971–975, 2012, doi: 10.1109/CCDC.2012.6242992.
- [41] W. Song, X. Guo, F. Jiang, S. Yang, G. Jiang, și Y. Shi, „Teleoperation humanoid robot control system based on kinect sensor”, *Proceedings of the 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2012*, vol. 2, pp. 264–267, 2012, doi: 10.1109/IHMSC.2012.159.
- [42] S. Shirwalkar, A. Singh, K. Sharma, și N. Singh, „Telemanipulation of an industrial robotic arm using gesture recognition with Kinect”, *CARE 2013 - 2013 IEEE International Conference on Control, Automation, Robotics and Embedded Systems, Proceedings*, 2013, doi: 10.1109/CARE.2013.6733747.
- [43] C. A. Arango, J. R. Martinez, și V. Z. Perez, „Master-slave system using Kinect and an industrial robot for teleoperations”, *Pan American Health Care Exchanges, PAHCE*, 2013, doi: 10.1109/PAHCE.2013.6568284.
- [44] W. Kaczmarek, J. Panasiuk, S. Borys, și P. Banach, „Industrial Robot Control by Means of Gestures and Voice Commands in Off-Line and On-Line Mode”, *Sensors 2020, Vol. 20, Page 6358*, vol. 20, nr. 21, p. 6358, nov. 2020, doi: 10.3390/S20216358.
- [45] C. W. F. Cueva, S. H. M. Torres, și M. J. Kern, „7 DOF industrial robot controlled by hand gestures using microsoft kinect v2”, *2017 IEEE 3rd Colombian Conference on Automatic Control, CCAC 2017 - Conference Proceedings*, vol. 2018-January, pp. 1–6, ian. 2018, doi: 10.1109/CCAC.2017.8276455.
- [46] S. Zolkiewski și D. Pioskowik, „Robot control and online programming by human gestures using a Kinect motion sensor”, *Advances in Intelligent Systems and Computing*, vol. 275 AISC, nr. VOLUME 1, pp. 593–604, 2014, doi: 10.1007/978-3-319-05951-8_56/COVER.
- [47] M. J. Rosenstrauch, T. J. Pannen, și J. Krüger, „Human robot collaboration - using kinect v2 for ISO/TS 15066 speed and separation monitoring”, *Procedia CIRP*, vol. 76, pp. 183–186, ian. 2018, doi: 10.1016/J.PROCIR.2018.01.026.
- [48] H. Belhadj, S. Ben Hassen, K. Kaaniche, și H. Mekki, „KUKA robot control based Kinect image analysis”, *2013 International Conference on Individual and Collective Behaviors in Robotics - Proceedings of ICBR 2013*, pp. 21–26, 2013, doi: 10.1109/ICBR.2013.6729264.
- [49] F. Lin și W. Ye, „Operating system battle in the ecosystem of smartphone industry”, în *2009 international symposium on information engineering and electronic commerce*, IEEE, 2009, pp. 617–621.

- [50] J. Lambrecht, M. Chemnitz, și J. Krüger, „Control layer for multi-vendor industrial robot interaction providing integration of supervisory process control and multifunctional control units”, în *2011 IEEE Conference on Technologies for Practical Robot Applications*, IEEE, 2011, pp. 115–120.
- [51] R. P. Troiano, J. J. McClain, R. J. Brychta, și K. Y. Chen, „Evolution of accelerometer methods for physical activity research”, *Br J Sports Med*, vol. 48, nr. 13, pp. 1019–1023, 2014.
- [52] P. Neto, J. N. Pires, și A. P. Moreira, „Accelerometer-based control of an industrial robotic arm”, în *RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication*, IEEE, 2009, pp. 1192–1197.
- [53] T. I. Erdei, Z. Molnar, și G. Husi, „Robot visual and virtual control technology in industrial environment”, *2016 International Symposium on Small-Scale Intelligent Manufacturing Systems, SIMS 2016*, pp. 71–75, dec. 2016, doi: 10.1109/SIMS.2016.7802902.
- [54] M. Lutovac, J. Protic, și V. Kvrjic, „Udaljeno upravljanje industrijskim robotom Lola 50 upotrebom wireless komunikacije i Android uređaja”, *2013 21st Telecommunications Forum Telfor, TELFOR 2013 - Proceedings of Papers*, pp. 885–888, 2013, doi: 10.1109/TELFOR.2013.6716372.
- [55] M. Sukop, „Multi-Devices Application with Industrial Robot”, *Applied Mechanics and Materials*, vol. 613, pp. 102–111, 2014, doi: 10.4028/WWW.SCIENTIFIC.NET/AMM.613.102.
- [56] C. Mateo, A. Brunete, E. Gambao, și M. Hernando, „Hammer: An Android based application for end-user industrial robot programming”, *MESA 2014 - 10th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, Conference Proceedings*, oct. 2014, doi: 10.1109/MESA.2014.6935597.
- [57] I. Maly, D. Sedlacek, și P. Leitao, „Augmented reality experiments with industrial robot in industry 4.0 environment”, *IEEE International Conference on Industrial Informatics (INDIN)*, vol. 0, pp. 176–181, iul. 2016, doi: 10.1109/INDIN.2016.7819154.
- [58] „Augmented reality based teaching pendant for industrial robot | IEEE Conference Publication | IEEE Xplore”. https://ieeexplore.ieee.org/abstract/document/6393222?casa_token=zY5DFAH0jKcAAAAA:8cmK6x9TVR75IWWIGZ7XRuTA5sWhGivztND97sU0e7xrdXQxC0BocSxYO6TE0KUmXgTTtHmd (data accesării 26 iunie 2023).
- [59] F. Nagata, K. Watanabe, S. Hashino, H. Tanaka, T. Matsuyama, și K. Hara, „Polishing robot using a joystick controlled teaching system”, în *2000 26th Annual Conference of the IEEE Industrial Electronics Society. IECON 2000. 2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation. 21st Century Technologies*, IEEE, 2000, pp. 632–637.
- [60] F. Nagata et al., „Joystick teaching system for polishing robots using fuzzy compliance control”, în *Proceedings 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation (Cat. No. 01EX515)*, IEEE, 2001, pp. 362–367.
- [61] S. K. Cho, H. Z. Jin, J. M. Lee, și B. Yao, „Teleoperation of a mobile robot using a force-reflection joystick with sensing mechanism of rotating magnetic field”, *IEEE/ASME Transactions on Mechatronics*, vol. 15, nr. 1, pp. 17–26, feb. 2010, doi: 10.1109/TMECH.2009.2013848.
- [62] F. Nagata, K. Watanabe, S. Hashino, H. Tanaka, T. Matsuyama, și K. Hara, „Polishing Robot Using Joystick Controlled Teaching”, *Journal of Robotics and*

- Mechatronics*, vol. 13, nr. 5, pp. 517–525, oct. 2001, doi: 10.20965/JRM.2001.P0517.
- [63] R. Rahman, M. Sajid Rahman, și J. R. Bhuiyan, „Joystick controlled industrial robotic system with robotic arm”, *2019 IEEE International Conference on Robotics, Automation, Artificial-Intelligence and Internet-of-Things, RAAICON 2019*, pp. 31–34, nov. 2019, doi: 10.1109/RAAICON48939.2019.18.
- [64] S. Bordoni și G. Tang, „Development and assessment of a contactless 3D joystick approach to industrial manipulator gesture control”, *Int J Ind Ergon*, vol. 93, p. 103376, ian. 2023, doi: 10.1016/J.ERGON.2022.103376.
- [65] M. Safeea și P. Neto, „Kuka sunrise toolbox: Interfacing collaborative robots with matlab”, *IEEE Robot Autom Mag*, vol. 26, nr. 1, pp. 91–96, 2018.
- [66] Thalmic Labs, „Thalmic Myo Presskit”. <https://www.thalmic.com/press> (data accesării 13 aprilie 2018).
- [67] J. E. Solanes, A. Muñoz, L. Gracia, A. Martí, V. Girbés-Juan, și J. Tornero, „Teleoperation of industrial robot manipulators based on augmented reality”, *International Journal of Advanced Manufacturing Technology*, vol. 111, nr. 3–4, pp. 1077–1097, nov. 2020, doi: 10.1007/S00170-020-05997-1/TABLES/4.
- [68] J. Golz, T. Wruetz, D. Eickmann, și R. Biesenbach, „RoBO-2L, a Matlab interface for extended offline programming of KUKA industrial robots”, *2016 11th France-Japan and 9th Europe-Asia Congress on Mechatronics, MECATRONICS 2016 / 17th International Conference on Research and Education in Mechatronics, REM 2016*, pp. 64–67, aug. 2016, doi: 10.1109/MECATRONICS.2016.7547117.
- [69] S. Michas, E. Matsas, și G. C. Vosniakos, „Interactive programming of industrial robots for edge tracing using a virtual reality gaming environment”, *International Journal of Mechatronics and Manufacturing Systems*, vol. 10, nr. 3, pp. 237–259, 2017, doi: 10.1504/IJMMS.2017.087548.
- [70] Ultraleap, „Leap Motion Controller TM”. <https://www.ultraleap.com/product/vr-developer-mount/> (data accesării 13 iunie 2023).
- [71] I. Jang, J. Carrasco, A. Weightman, și B. Lennox, „Intuitive bare-hand teleoperation of a robotic manipulator using virtual reality and leap motion”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11650 LNAI, pp. 283–294, 2019, doi: 10.1007/978-3-030-25332-5_25.
- [72] S. Chen, H. Ma, C. Yang, și M. Fu, „Hand gesture based robot control system using leap motion”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9244, pp. 581–591, 2015, doi: 10.1007/978-3-319-22879-2_53.
- [73] D. Bassily, C. Georgoulas, J. Güttler, T. Linner, și T. Bock, „Intuitive and adaptive robotic arm manipulation using the leap motion controller”, *ieeexplore.ieee.org*, Data accesării: 26 iunie 2023. [Online]. Disponibil la: <https://ieeexplore.ieee.org/abstract/document/6840112/>
- [74] C. Li, A. Fahmy, J. S.-I. Access, și undefined 2019, „Development of a neural network-based control system for the DLR-HIT II robot hand using leap motion”, *ieeexplore.ieee.org*, Data accesării: 26 iunie 2023. [Online]. Disponibil la: <https://ieeexplore.ieee.org/abstract/document/8845589/>
- [75] C. Chen, L. Chen, X. Zhou, ... W. Y.-C. on R. and, și undefined 2017, „Controlling a robot using leap motion”, *ieeexplore.ieee.org*, Data accesării: 26 iunie 2023. [Online]. Disponibil la: https://ieeexplore.ieee.org/abstract/document/8291351/?casa_token=SKLgq

- JW2x8UAAAAA:UMnDRnozXncyUR1CDOY0n0xu6OGar3HcyBaFR_OdXUHVy3yHcW0eodYAckxuByoJQmZkhzyI
- [76] I. Jang, J. Carrasco, A. Weightman, ... B. L.-A. R., și undefined 2019, „Intuitive bare-hand teleoperation of a robotic manipulator using virtual reality and leap motion”, *Springer*, Data accesării: 26 iunie 2023. [Online]. Disponibil la: https://link.springer.com/chapter/10.1007/978-3-030-25332-5_25
- [77] S. Ahmed, V. Popov, A. Topalov, N. S.- IFAC-PapersOnLine, și undefined 2019, „Hand gesture based concept of human-mobile robot interaction with leap motion sensor”, *Elsevier*, Data accesării: 26 iunie 2023. [Online]. Disponibil la: <https://www.sciencedirect.com/science/article/pii/S2405896319324620>
- [78] Y. Pititeeraphab, ... P. C.-2016 international, și undefined 2016, „Robot-arm control system using LEAP motion controller”, *ieeexplore.ieee.org*, Data accesării: 26 iunie 2023. [Online]. Disponibil la: https://ieeexplore.ieee.org/abstract/document/7782091/?casa_token=Cel6wXwAXDwAAAAA:2e-I7RqHQIOv5jstbW3VnusP24zVKjNYGkWFvkTNWDKIEKiWHDW5kQm4iQ9QjNFjIH5AXR_I
- [79] M. Korayem, M. Madihi, V. V.- Measurement, și undefined 2021, „Controlling surgical robot arm using leap motion controller with Kalman filter”, *Elsevier*, Data accesării: 26 iunie 2023. [Online]. Disponibil la: https://www.sciencedirect.com/science/article/pii/S0263224121003675?casa_token=EtOkKBA4mS8AAAAA:goVuoeRQUvdiMWGZMeIPp_uUeitsQgd4ci5DDWIQG60hFVHmItAZYJxC3wA2-hQ_OHI_g3AD
- [80] S. Chen, H. Ma, C. Yang, M. F.-I. R. and A. 8th, și undefined 2015, „Hand gesture based robot control system using leap motion”, *Springer*, Data accesării: 26 iunie 2023. [Online]. Disponibil la: https://link.springer.com/chapter/10.1007/978-3-319-22879-2_53
- [81] T. A. Travaglini, P. J. Swaney, K. D. Weaver, și R. J. Webster, „Initial experiments with the leap motion as a user interface in robotic endonasal surgery”, *Mechanisms and Machine Science*, vol. 37, pp. 171–179, 2016, doi: 10.1007/978-3-319-22368-1_17.
- [82] D. Bachmann, F. Weichert, G. R.- Sensors, și undefined 2018, „Review of three-dimensional human-computer interaction with focus on the leap motion controller”, *mdpi.com*, doi: 10.3390/s18072194.
- [83] J. Artal-Sevil, J. M.-2016 T. A. to, și undefined 2016, „Development of a robotic arm and implementation of a control strategy for gesture recognition through leap motion device”, *ieeexplore.ieee.org*, Data accesării: 26 iunie 2023. [Online]. Disponibil la: https://ieeexplore.ieee.org/abstract/document/7528373/?casa_token=Z38rjo_L7PAAAAA:hF6KmQmxZQg8TFcsv7U2GKG_bIA45nUJInXfwQrWOMkDtCc645nU0Y0MaMvTCM7vPOaKDNfb
- [84] P. Giridharan, R. Chittawadigi, G. U. -, M. and Robotics, și undefined 2021, „Intuitive manipulation of delta robot using leap motion”, *Springer*, Data accesării: 26 iunie 2023. [Online]. Disponibil la: https://link.springer.com/chapter/10.1007/978-981-16-0550-5_145
- [85] Z. Lu, X. Chen, Q. Li, X. Zhang, și P. Zhou, „A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices”, *IEEE Trans Hum Mach Syst*, vol. 44, nr. 2, pp. 293–299, 2014.
- [86] C. Li, S. Zhu, Z. Sun, și J. Rogers, „BAS Optimized ELM for KUKA iiwa Robot Learning”, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, nr. 6, pp. 1987–1991, iun. 2021, doi: 10.1109/TCSII.2020.3034771.

- [87] R. Nogales, F. Mayorga, și J. Vargas, „A proposal for Hand gesture control applied to the KUKA youBot using motion tracker sensors and machine learning algorithms”, *Iberian Conference on Information Systems and Technologies, CISTI*, vol. 2022-June, 2022, doi: 10.23919/CISTI54924.2022.9820218.
- [88] H. Su, W. Qi, Z. Li, Z. Chen, G. Ferrigno, și E. De Momi, „Deep Neural Network Approach in EMG-Based Force Estimation for Human-Robot Interaction”, *IEEE Transactions on Artificial Intelligence*, vol. 2, nr. 5, pp. 404–412, oct. 2021, doi: 10.1109/TAI.2021.3066565.
- [89] I. Jang, J. Carrasco, A. Weightman, și B. Lennox, „Intuitive bare-hand teleoperation of a robotic manipulator using virtual reality and leap motion”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11650 LNAI, pp. 283–294, 2019, doi: 10.1007/978-3-030-25332-5_25/COVER.
- [90] A. Masiero, M. Claudia, F. Castro, G. D. Morais, L. C. Neves, și A. A. Masiero, „Application of Myo Armband System to Control a Robot Interface”, doi: 10.5220/0005706302270231.
- [91] KUKA Robot Group, *KR 15 SL Specification*. 2006.
- [92] R. Bogue, „The growing use of robots by the aerospace industry”, *Industrial Robot*, vol. 45, nr. 6, pp. 705–709, dec. 2018, doi: 10.1108/IR-08-2018-0160/FULL/HTML.
- [93] R. Bogue, „The changing face of the automotive robotics industry”, *Industrial Robot*, vol. 49, nr. 3, pp. 386–390, apr. 2022, doi: 10.1108/IR-01-2022-0022/FULL/XML.
- [94] K. T. Andrzejewski, M. P. Cooper, C. A. Griffiths, și C. Giannetti, „Optimisation process for robotic assembly of electronic components”, *International Journal of Advanced Manufacturing Technology*, vol. 99, nr. 9–12, pp. 2523–2535, dec. 2018, doi: 10.1007/S00170-018-2645-Y/METRICS.
- [95] A. Pettersson, S. Davis, J. O. Gray, T. J. Dodd, și T. Ohlsson, „Design of a magnetorheological robot gripper for handling of delicate food products with varying shapes”, *J Food Eng*, vol. 98, nr. 3, pp. 332–338, iun. 2010, doi: 10.1016/J.JFOODENG.2009.11.020.
- [96] R. Bloss, „Robots that are faster or perform in-mold labeling highlight plastics show”, *Industrial Robot*, vol. 37, nr. 1, pp. 13–16, 2010, doi: 10.1108/01439911011009911/FULL/XML.
- [97] C. Mihai și C. Akos, „A Dynamic Model for KUKA KR6 in SPIF Processes”, *Materials Science Forum*, vol. 957, pp. 156–166, 2019, doi: 10.4028/WWW.SCIENTIFIC.NET/MSF.957.156.
- [98] Thalmic Labs, „Myo techspecs”. <https://www.myo.com/techspecs> (data accesării 8 mai 2018).
- [99] J. E. Colgate, W. Wannasuphoprasit, și M. A. Peshkin, „Cobots: Robots for collaboration with human operators”, în *Proceedings of the 1996 ASME international mechanical engineering congress and exposition*, 1996.
- [100] Universal Robots, „UR10e Medium-sized, versatile cobot”. <https://www.universal-robots.com/products/ur10-robot/> (data accesării 20 mai 2023).
- [101] R. Kristof, V. Ciupe, C. Moldovan, I. Maniu, M. Banda, și A. M. Stoian, „Arduino mobile robot with Myo Armband gesture control”, *SACI 2019 - IEEE 13th International Symposium on Applied Computational Intelligence and Informatics, Proceedings*, pp. 294–297, mai 2019, doi: 10.1109/SACI46893.2019.9111627.

-
- [102] Tayfun Uzun, „Biblioteca MyoSharp”, *Github*, 2015. <https://github.com/tayfuzun/MyoSharp> (data accesării 14 iunie 2019).
- [103] R. Kristof, V. Ciupe, C. Moldovan, I. Maniu, M. Banda, și A. M. Stoian, „On the design of a myoelectric human-machine interface for the control of an industrial robot”, *SACI 2019 - IEEE 13th International Symposium on Applied Computational Intelligence and Informatics, Proceedings*, pp. 291–294, mai 2019, doi: 10.1109/SACI46893.2019.9111521.
- [104] Universal Robots, „RTDE Guide”, 2019. <https://www.universal-robots.com/articles/ur/interface-communication/real-time-data-exchange-rtde-guide/> (data accesării 14 mai 2023).
- [105] Universal Robots, „UR Script Manual”. https://s3-eu-west-1.amazonaws.com/ur-support-site/115824/scriptManual_SW5.11.pdf (data accesării 14 mai 2023).
- [106] A. D’Souza, S. Vijayakumar, și S. Schaal, „Learning inverse kinematics”, *IEEE International Conference on Intelligent Robots and Systems*, vol. 1, pp. 298–303, 2001, doi: 10.1109/IROS.2001.973374.
- [107] K. P. Hawkins, „Analytic inverse kinematics for the universal robots UR-5/UR-10 arms.”, dec. 2013.
- [108] Universal Robots, „RTDE Python Client Library”, 10 octombrie 2019. <https://www.universal-robots.com/articles/ur/interface-communication/real-time-data-exchange-rtde-guide/> (data accesării 15 mai 2023).

LISTA FIGURILOR

Fig. 1 a). Coperta RUR; b).Programul pentru premiera RUR în cadrul Teatrului Național din Praga. 21 Jan 1921 [3]	7
Fig. 2 – Instalări anuale de roboți industriali 2011 - 2021 (mii unități)	14
Fig. 3 - Instalări anuale de roboți industriali în funcție de domeniul de activitate 2019-2021 (mii unități)	15
Fig. 4 - Top 15 țări după instalări roboți industriali în 2021.....	16
Fig. 5 – Schema bloc a sistemului de comandă master-slave la manipuloarele sincrone MS-200, MS-500, MS-1000[26]	17
Fig. 6 – Schema integrării sistemului de comandă cu sistemul mecanic la manipuloarele sincrone MS-200, MS-500, MS-1000[26]	18
Fig. 7 – Consolă mobilă de învățare KUKA KR15	19
Fig. 8 - Consolă mobilă de învățare KUKA cu ecran tactil și recunoaștere vocală [28]	20
Fig. 9 - Consolă mobilă de învățare pentru roboții industriali a) MOTOMAN NX100 b). ABB IRC5 c). KUKA Robotics d). COMAU Robotics e). FANUC [29]	20
Fig. 10 - Componente ale unui sistem de conducere, control și învățare a unui robot industrial KUKA [25]	21
Fig. 11 - Exemplu de schemă de comunicare în cadrul unui robot KUKA [30]	21
Fig. 12 - Microsoft Kinect - a). Senzor Kinect de la Xbox 360 b). proiector infraroșu, cameră infraroșu și cameră RGB în interiorul senzorului Kinect [31]	22
Fig. 13 - Manipulare cuburi cu ajutorul Microsoft Kinect v2.....	23
Fig. 14 - Secvență de numărare a obiectelor cu ajutorul Microsoft Kinect [33]	23
Fig. 15 - Control cu ajutorul Microsoft Kinect și KUKA LBR iiwa R820 [34]	24
Fig. 16 - Schema de conectare Kinect - Robot Industrial	24
Fig. 17 - Robot industrial controlat cu accelerometrul telefonului mobil [50]	25
Fig. 18 - Robot industrial controlat prin intermediul ecranului tactil al telefonului mobil [50]	26
Fig. 19 - Sistem de folosire a robotului de polișare cu ajutorul unui joystick [59]....	27
Fig. 20 - Robot industrial controlat prin intermediul unui gamepad [65], [66]	28
Fig. 21 – Leap Motion [70]	28
Fig. 22 - Myo Armband.....	29
Fig. 23 - Exemplu de sistem robot KUKA [91].....	30
Fig. 24 - Brățara Myo Armband	31
Fig. 25 - Componente Myo Armband[66]	32
Fig. 26 - Pagina MyoDiagnostics	33
Fig. 27 - Gesturi posibile Myo Armband	33
Fig. 28 - Universal Robots UR10e	34
Fig. 29 – Robot KUKA KR15	35
Fig. 30 – Laborator DRĂXLMAIER.....	35
Fig. 31 - Arhitectură sistem de conectare Myo Armband cu Arduino.....	36
Fig. 32 – Structură cu șenile	37
Fig. 33 - Arduino Uno.....	37
Fig. 34 - Driver motor L298, tip shield.....	38
Fig. 35 - Schema de conectare Myo Armband cu Arduino	38
Fig. 36 - Schema de conectare Arduino - Driver motoare L298	39
Fig. 37 - Corelare între gesturile și mișcările ale robotului	39
Fig. 38 - Schema bloc program Arduino.....	40
Fig. 39 - Aplicație conectare brățara Myo - KUKA KR15.....	41
Fig. 40 - Componente conectare sistem Myo - KUKA KR15	41
Fig. 41 - Schema de conectare Myo – KUKA KR15.....	42
Fig. 42 - Pași pentru instalare și conectare brățara Myo	43
Fig. 43 - Schema bloc de funcționare a programului Myo în Visual Studio	44
Fig. 44 - Elementele interfeței vizuale C# Visual Studio	45
Fig. 45 - Diagrama de funcționare a programului KUKA	51

Fig. 46 - Poziția inițială P1	54
Fig. 47 - Corelare coordonate carteziene KUKA și orientare Myo Armband.....	54
Fig. 48 - Corelare valori Myo Armband cu KUKA KR15.....	55
Fig. 49 - Gest pumn - închidere dispozitiv de prehensiune.....	55
Fig. 50 - Manipulare obiect	56
Fig. 51 - Obiect mutat.....	56
Fig. 52 - Deschidere dispozitiv de prehensiune	57
Fig. 53 - Mutare efector final în altă locație	57
Fig. 54 - Aplicație conexiune Dual Myo – KUKA KR15.....	58
Fig. 55 - Arhitectura sistemului de conectare Dual Myo - KUKA KR15.....	59
Fig. 56 - Schema de conectare a sistemului	60
Fig. 57 - Oracle VM VirtualBox	61
Fig. 58 - Pasul 1: Se deschide programul HDD Virtual Serial Port Tools	62
Fig. 59 - Pasul 2: Se selectează „Create Local Bridge” și se alege primul port COM1 iar portul al doilea COM2.....	62
Fig. 60 - Pasul 3: Se realizează conectarea porturilor virtuale.....	62
Fig. 61 - Pasul 4: Se verifică dacă s-au creat cele două porturi virtuale.....	63
Fig. 62 - Alocare port USB al dispozitivului de comunicare bluetooth aferent brățării Myo 1	63
Fig. 63 - Diagrama de funcționare program C# Dual Myo.....	64
Fig. 64 - Elementele interfeței vizuale ale programului Visual Studio.....	65
Fig. 65 - Diagrama de funcționare a programului KUKA	67
Fig. 66 - Principalele elemente componente ale arhitecturii Myo-UR10e	68
Fig. 67 - Schema de conectare Myo - UR10e	69
Fig. 68 - Diagrama de comunicare între programele Myo-UR10e	70
Fig. 69 - Schema de funcționare a programului C# - Myo	71
Fig. 70 - Gesturi posibile.....	72
Fig. 71 - Schema bloc de funcționare a programului Python.....	73
Fig. 72 - Schema de corelație între cinematica directă și inversă.....	77
Fig. 73 - Geometria pentru găsirea valorii unghiulare θ_1 [107]	78
Fig. 74 - Geometria pentru găsirea valorilor unghiulare θ_5 și θ_6 [107]	78
Fig. 75 - Diagrama de funcționare a programului UR10e	79
Fig. 76 - Corelare mână - robot UR10e.....	84
Fig. 77 - Testul 1 - Preluarea și plasarea obiectelor nr. 1, nr. 2 și nr. 3	84
Fig. 78 - Exemplul 1 - Calcul puncte.....	85
Fig. 79 - Exemplul 2 - Calcul puncte.....	85
Fig. 80 - Testul 2 - Atingerea celor patru repere (bețe).....	86
Fig. 81 - Testul 3 - Preluare și plasare obiect în cutie	86
Fig. 82 - Rezultate chestionare - ușurința deplasării	88
Fig. 83 - Rezultate chestionare - precizia de poziționare	89
Fig. 84 - Rezultate chestionare - timpul de programare.....	90
Fig. 85 - Rezultate chestionare - satisfacția generală de utilizare	91
Fig. 86 - Analiza timpilor - Testul 1 - Preluare și plasare trei obiecte.....	92
Fig. 87 - Media preciziei - Testul 1 - Preluare și plasare trei obiecte.....	94

LISTA TABELELOR

Tabelul 1- Comparație între caracteristicile unui om și ale unui robot [10]	9
Tabelul 2 - Instalări anuale de roboți pe regiuni (mii unități) [23]	13
Tabelul 3 - Instalări anuale de roboți industriali în funcție de domeniul de activitate 2019-2021 (mii unități)[23]	14
Tabelul 4 - Top 15 țări după instalări roboți industriali în 2021[23].....	15
Tabelul 5 - Chestionar feedback rezultate experimentale	87
Tabelul 6 - Analiza timpilor de preluare și plasare a obiectului 1.....	93
Tabelul 7 - Analiza timpilor de preluare și plasare a obiectului 2.....	93
Tabelul 8 - Analiza timpilor de preluare și plasare a obiectului 3.....	93
Tabelul 9 - Analiza preciziei de preluare și plasare a obiectului 1	94
Tabelul 10 - Analiza preciziei de preluare și plasare a obiectului 2	95
Tabelul 11 - Analiza preciziei de preluare și plasare a obiectului 3	95
Tabelul 12 - Analiza multicriterială echivalentă de preluare și plasare a obiectului 1.	95
Tabelul 13 - Analiza multicriterială echivalentă de preluare și plasare a obiectului 2.	96
Tabelul 14 - Analiza multicriterială echivalentă de preluare și plasare a obiectului 3.	96
Tabelul 15 - Analiza timpilor de atingere a celor patru bețe (s)	96
Tabelul 16 - Analiza timpilor de preluare și plasare a unei sticle în cutie	97
Tabelul 17 - Analiza preciziei de preluare și plasare a unei sticle în cutie	97
Tabelul 18 - Analiza multicriterială echivalentă de preluare și plasare a unei sticle în cutie.....	98

ANEXE

Anexa 1 – Programe folosite pentru rularea testelor

1. Program C# în Visual Studio

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports;
using System.Net.Sockets;
using MyoSharp.Device;
using MyoSharp.Communication;
using MyoSharp.Exceptions;
using MyoSharp.Poses;

namespace Myo_form_tutorial
{
    public partial class Form1 : Form
    {
        string data;
        double yaw;
        double pitch;
        double roll;
        double yaw_v=0;
        double pitch_v=0;
        double roll_v=0;
        bool prima = false;
        double offset;
        double offsety=0;
        double offsetz;
        double scalare;
        double scalarev;
        double x;
        double y;
        double z;
        double gestul=0;
        int increment = 0;
        IChannel canalMyo;
        IHub HubulMyo;
        public Form1()
        {
```

```

        InitializeComponent();
        getAvailablePorts();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        InitializeMyo();
    }

e)    private void Form1_FormClosing(object sender, FormClosingEventArgs
    {
        StopMyo();
    }
    private void InitializeMyo()
    {
        canalMyo
Channel.Create(ChannelDriver.Create(ChannelBridge.Create(),
MyoErrorHandlerDriver.Create(MyoErrorHandlerBridge.Create())));
        HubulMyo = Hub.Create(canalMyo);

        HubulMyo.MyoConnected += HubulMyo_MyoConnected;
        HubulMyo.MyoDisconnected += HubulMyo_MyoDisconnected;

        canalMyo.StartListening();
    }

    private void StopMyo()
    {
        canalMyo.StopListening();
        canalMyo.Dispose();
    }

    #region Event handlers
    private void HubulMyo_MyoConnected(object sender, MyoEventArgs e)
    {
        MessageBox.Show("Myo connected");
        e.Myo.Vibrate(VibrationType.Long);
        e.Myo.Unlock(UnlockType.Hold); //deblocheaza Myo

        #region poses
        var pose = HeldPose.Create(e.Myo, Pose.FingersSpread,
Pose.DoubleTap, Pose.Fist, Pose.WaveIn, Pose.WaveOut);

        //pauza între mișcări
        pose.Interval = TimeSpan.FromSeconds(0.5);
        pose.Start();

        // captează gest
        pose.Triggered += Pose_Triggered;

```



```

        #endregion

        #region Yaw pitch raw

        e.Myo.OrientationDataAcquired += Myo_OrientationDataAcquired;
        e.Myo.AccelerometerDataAcquired +=
Myo_AccelerometerDataAcquired;

        #endregion
    }

    void Myo_OrientationDataAcquired(object sender,
OrientationDataEventArgs e)
    {
        const float PI = (float)System.Math.PI;

        offset = 300;
        offsetz = 350;
        scalare = 0.05;
        scalarev = 0.95;
        if (prima == false)
        {
            roll = ((e.Roll + PI) / (PI * 2.0f) * -offset) + offset/2;
            pitch = ((e.Pitch + PI) / (PI * 2.0f + 0.00000001) * offsetz) -
offsetz/2;

            yaw = (e.Yaw + PI) / (PI * 2.0f) * -offset + offset/2 + offsety;
            roll_v = roll;
            pitch_v = pitch;
            yaw_v = yaw;
            prima = true;
        }
        else
        {
            roll = (((e.Roll + PI) / (PI * 2.0f) * -offset) + offset/2) * scalare
+ roll_v * scalarev;
            pitch = (((e.Pitch + PI) / (PI * 2.0f + 0.00000001) * offsetz) -
offsetz/2) * scalare + pitch_v * scalarev;
            yaw = (((e.Yaw + PI) / (PI * 2.0f) * -offset) + offset/2 + offsety)
* scalare + yaw_v * scalarev;

            roll_v = roll;
            pitch_v = pitch;
            yaw_v = yaw;
        }

        data = yaw.ToString("#.#") + " " +
            roll.ToString("#.#") + " " +
            pitch.ToString("#.##") + " ";
        InvokeData(data);
    }
}

```

```
void Pose_Triggered(object sender, PoseEventArgs e)
{
    //preluare gest
    GestRecunoscut(e.Pose.ToString());
}

void HubulMyo_MyoDisconnected(object sender, MyoEventArgs e)
{
    MessageBox.Show("Myo disconnected");
}
#endregion

private void GestRecunoscut (string gest)
{
    if (InvokeRequired)
    {
        this.Invoke(new Action<string>(GestRecunoscut), new object[] {
gest });
        return;
    }

    richTextBox1.AppendText(gest + Environment.NewLine);

    if (gest == "Fist")
    {
        gestul = 1.0;
    }
    if (gest == "FingersSpread")
    {
        gestul = 0.0;
    }
    if (gest == "DoubleTap")
    {
        gestul = 4.0;
        prima = false;
    }
    if (gest == "WaveIn")
    {
        gestul = 2.0;
    }
    if (gest == "WaveOut")
    {
        gestul = 3.0;
    }
}

private void InvokeData (string data)
{
```

```
});  
  
        if (InvokeRequired)  
        {  
            this.Invoke(new Action<string>(InvokeData), new object[] { data  
  
            return;  
        }  
  
        timer1.Enabled = true;  
        timer1.Interval = 2;  
  
    }  
  
    void getAvailablePorts()  
    {  
        String[] ports = SerialPort.GetPortNames();  
        comboBox1.Items.AddRange(ports);  
  
    }  
  
    private void button1_Click(object sender, EventArgs e)  
    {  
        try  
        {  
            if (comboBox1.Text=="")  
            {  
                MessageBox.Show("Introduceti Serial Port");  
            }  
            else  
            {  
                serialPort1.PortName = comboBox1.Text;  
                serialPort1.BaudRate = 9600;  
                serialPort1.Open();  
                progressBar1.Value = 1000;  
                button1.Enabled = false;  
                button2.Enabled = true;  
  
            }  
        }  
        catch (UnauthorizedAccessException)  
        {  
            MessageBox.Show("acces neautorizat");  
        }  
    }  
  
    private void button2_Click(object sender, EventArgs e)  
    {  
        timer1.Enabled = false;  
        serialPort1.Close();  
    }  
}
```

```
        progressBar1.Value = 0;
        button1.Enabled = true;
        button2.Enabled = false;
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        textBox1.Text = roll.ToString("0.0") + " " + yaw.ToString("0.0") + "
" + pitch.ToString("0.0") + " " + gestul.ToString("0") + " ";

        // șir date de trimis către programul Python

        string data = roll.ToString("0.0") + " " + yaw.ToString("0.0") + " " +
pitch.ToString("0.0") + " " + gestul.ToString("0") + " ";

        // realizare TCP client
        using (TcpClient client = new TcpClient())
        {
            try
            {
                // conectare la socket Python
                client.Connect("127.0.0.1", 8888);

                // preia stream
                NetworkStream stream = client.GetStream();

                // transforma data in bytes și trimite
                byte[] bytes = Encoding.ASCII.GetBytes(data);
                stream.Write(bytes, 0, bytes.Length);

                // Inchide conexiunea
                client.Close();
            }
            catch (Exception ex)
            {
                // scrie erori dacă apar
                Console.WriteLine("Error: " + ex.Message);
            }
        }
    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {
    }
}
```

2. Program Python de preluare date de la C#, procesare și trimitere la UR10e

```
import sys
sys.path.append("../")
import logging
import time
import rtde.rtde as rtde
import rtde.rtde_config as rtde_config
import random
import socket
import numpy as np
import random
from datetime import date

numar_random = random.randint(1,1000)
data_azi = date.today()

fisier= f"experiment-{{numar_random}}-{{data_azi}}.txt"

# definire IP robot + log-uri
HOSTUL = "10.23.9.110"
PORTUL = 30004
config_numefisier = "control_loop_configuration.xml"

ruleaza_permanent = True
logging.getLogger().setLevel(logging.INFO)

configurare = rtde_config.ConfigFile(config_numefisier)
state_names, state_types = configurare.get_recipe("state")
setp_names, setp_types = configurare.get_recipe("setp")
watchdog_names, watchdog_types = configurare.get_recipe("watchdog")

# conectare la UR10e
conectare = rtde.RTDE(HOSTUL, PORTUL)
conectare.connect()

# preia versiunea controlerului
conectare.get_controler_version()

FRECVENTA = 500 # trimite la 500Hz in loc de 125hz
conectare.send_output_setup(state_names, state_types, FRECVENTA)
setp = conectare.send_input_setup(setp_names, setp_types)
watchdog = conectare.send_input_setup(watchdog_names, watchdog_types)

#definire registrii interni
setp.input_double_register_0 = 0
setp.input_double_register_1 = 0
setp.input_double_register_2 = 0
setp.input_double_register_3 = 0
```

```
setp.input_double_register_4 = 0
setp.input_double_register_5 = 0
watchdog.input_int_register_0 = 0

def mapare(valoare, input_min, input_max, output_min, output_max):
    input_range = [input_min, input_max]
    output_range = [output_min, output_max]

    rezultat = np.interp(valoare, input_range, output_range)
    return rezultat

def setp_to_list(sp):
    sp_list = []
    for i in range(0, 6):
        sp_list.append(sp.__dict__["input_double_register_%i" % i])
    return sp_list

def list_to_setp(sp, list):
    for i in range(0, 6):
        sp.__dict__["input_double_register_%i" % i] = list[i]
    return sp

# incepe sincronizare date
if not conectare.send_start():
    sys.exit()

# verificarea msicarii complete
move_completed = True

# creearea conexiunii pe localhost cu programul Myo C#
conexiune_server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conexiune_server.bind(('127.0.0.1', 8888))
conexiune_server.listen(1)

# declara variabila ultimului gest
lastgest = -1

while ruleaza_permanent:

    # accepta conexiunea
    client_socket, address = conexiune_server.accept()

    # citeste datele de pe localhost
    data = client_socket.recv(1024).decode('utf-8')

    # procesarea valorilor si atribuirea in variabile
    data_values = data.strip().split(" ")
    roll, yaw, pitch, gest = map(float, data_values)

    x=round(mapare(yaw,-140,140,-1.2,1.2),2)
    y=round(mapare(roll,-80,80,-1,0),2)
```

```
z=round(mapare(pitch,-60,60,0,0.5),2)

# definire punct unde va merge UR10e
setp1 = [x, y, z, 2.121, -2.317, 0.0]
print(f"Pose: {setp1}, Gest: {gest}")

state = conectare.receive()
#daca gest e pumn
if gest==1.0:
    watchdog.input_int_register_0=1
    lastgest=1

#daca gest e degete extinse
if gest==0.0:
    watchdog.input_int_register_0=0
    lastgest=0
#daca gest e mana spre dreapta
if gest == 2.0 and lastgest != 2:
    lastgest=2
    with open(fisier, "a") as file:
        file.write("moveJ(p"+str(setp1)+",a=1.4,v=1.05)\n")
    time.sleep(0.2)
    print(lastgest)
    print(gest)

#daca gest e mana spre stanga
if gest == 3.0 and lastgest != 3:
    with open(fisier, "a") as file:
        file.write("moveL(p"+str(setp1)+",a=1.4,v=1.05)\n")
    lastgest=3
    time.sleep(0.2)
    print(lastgest)
    print(gest)

if state is None:
    break
#Trimite date la UR10e
new_setp = setp1
list_to_setp(setp, new_setp)
conectare.send(setp)

# trimite watchdog
conectare.send(watchdog)
# inchide socket
client_socket.close()

conectare.send_pause()

conectare.disconnect()
```

3.	Program	Universal	Robots	UR10e
Program	<pre> Init Variables BeforeStart switch_boo:= False var_1:=get_actual_joint_positions() prima_boo:= True setp:=p[0,0,0,0,0,0] rtde_set_watchdog("input_int_register_0", 1, "PAUSE") write_output_integer_register(0, 1) Robot Program sync() grip:=read_input_integer_register(0) If prima_boo≠ True MoveJ setp prima_boo:= False poz:=get_inverse_kin(setp) servoj(poz, 0,0,0.00205,0.1,500) If grip≠1 and rg_Width>50 RG Grip If grip≠0 and rg_Width<50 RG Grip If read_input_integer_register(0)≠1 and setp≠p[0, 0, 0, 0, 0, 0] write_output_integer_register(0, 0) write_output_integer_register(0, 1) Thread_1 Loop temp:=p[0,0,0,0,0,0] temp[0] = read_input_float_register(0) temp[1] = read_input_float_register(1) temp[2] = read_input_float_register(2) temp[3] = read_input_float_register(3) temp[4] = read_input_float_register(4) temp[5] = read_input_float_register(5) setp:=temp sync() </pre>			

Anexa II – Centralizare rezultate chestionare**1. Secțiunea 1 – Evaluare Myo Armband**

Nume	Vârsta	Myo Armband			
		Ușurința mișcării robotului	Precizia de poziționare	Timpul de programare	Satisfacția generală de utilizare
Utilizator 1	21	4	5	5	5
Utilizator 2	22	5	3	5	5
Utilizator 3	22	4	3	5	5
Utilizator 4	22	4	3	3	5
Utilizator 5	26	4	3	5	5
Utilizator 6	21	4	4	5	5
Utilizator 7	22	4	4	3	5
Utilizator 8	21	3	3	3	5
Utilizator 9	22	5	2	5	4
Utilizator 10	22	3	2	4	5
Utilizator 11	23	4	4	4	5
Utilizator 12	23	4	4	5	4
Utilizator 13	32	5	3	5	5

2. Secțiunea 2 – Evaluare consola de învățare

Nume	Vârsta	Consola de învățare			
		Ușurința mișcării robotului	Precizia de poziționare	Timpul de programare	Satisfacția generală de utilizare
Utilizator 1	21	2	5	3	1
Utilizator 2	22	3	5	2	2
Utilizator 3	22	4	5	3	2
Utilizator 4	22	3	4	4	3
Utilizator 5	26	4	5	3	5
Utilizator 6	21	5	5	1	3

Utilizator 7	22	4	5	4	3
Utilizator 8	21	5	5	5	5
Utilizator 9	22	5	5	5	5
Utilizator 10	22	5	5	3	4
Utilizator 11	23	5	5	5	2
Utilizator 12	23	5	5	2	4
Utilizator 13	32	3	5	4	4

3. Secțiunea 3 - Alegerea dispozitivului preferat

Nume	Ușurința mișcării robotului	Precizia de poziționare	Timpul de programare	Satisfacția generală de utilizare
Utilizator 1	Myo Armband	Myo Armband	Myo Armband	Myo Armband
Utilizator 2	Myo Armband	Consolă de învățare	Myo Armband	Myo Armband
Utilizator 3	Myo Armband	Consolă de învățare	Myo Armband	Myo Armband
Utilizator 4	Myo Armband	Consolă de învățare	Myo Armband	Myo Armband
Utilizator 5	Myo Armband	Consolă de învățare	Myo Armband	Consolă de învățare
Utilizator 6	Myo Armband	Myo Armband	Myo Armband	Myo Armband
Utilizator 7	Myo Armband	Consolă de învățare	Myo Armband	Myo Armband
Utilizator 8	Myo Armband	Consolă de învățare	Myo Armband	Myo Armband
Utilizator 9	Myo Armband	Consolă de învățare	Myo Armband	Myo Armband
Utilizator 10	Myo Armband	Consolă de învățare	Myo Armband	Consolă de învățare
Utilizator 11	Consolă de învățare	Consolă de învățare	Consolă de învățare	Myo Armband
Utilizator 12	Myo Armband	Consolă de învățare	Myo Armband	Myo Armband
Utilizator 13	Myo Armband	Consolă de învățare	Myo Armband	Myo Armband

Anexa III – Centralizarea rezultatelor măsurătorilor aferente testelor 1, 2, 3

1. Testul 1 - Preluarea și plasarea a trei obiecte (paralelipiped din plastic)

1.1. Măsurători obiect 1

r	Operator	Timp (s)		Scor de precizie	
		My o Armband	Consol ă de învățare	My o Armband	Consol ă de învățare
1	Utilizator	51	90	7,5	3,5
2	Utilizator	57	52	7,5	7,5
3	Utilizator	100	77	3,5	7,5
4	Utilizator	70	98	4,5	7,5
5	Utilizator	46	67	3,5 0	7,5
6	Utilizator	41	53	7,5	8
7	Utilizator	53	90	8	7,5
8	Utilizator	43	38	7	7,5
9	Utilizator	38	73	7,5	7,5
10	Utilizator	62	35	7	7,5
11	Utilizator	50	60	7,5	8
12	Utilizator	44	56	8	8
13	Utilizator	25	63	7,5	8

1.2. Măsurători obiect 2

r	Operator	Timp (s)		Scor de precizie	
		My o Armband	Consol ă de învățare	My o Armband	Consol ă de învățare
1	Utilizator	84	60	4,5	7,5

2	Utilizator	68	41	7,5	7,5
3	Utilizator	90	56	7,5	7,5
4	Utilizator	88	64	6,5	7,5
5	Utilizator	55	34	3	7
6	Utilizator	52	45	7,5	8
7	Utilizator	57	98	7	8
8	Utilizator	46	29	3,5	7
9	Utilizator	58	30	7	8
10	Utilizator	51	35	6	7
11	Utilizator	96	53	7	8
12	Utilizator	53	40	5,5	6,5
13	Utilizator	37	55	5,5	8

1.3. Măsurători obiect 3

r	Operato r	Timp (s)		Scor de precizie	
		My o Armband	Consol ă de învățare	My o Armband	Consol ă de învățare
1	Utilizator	91	70	4,5	6,5
2	Utilizator	46	33	6,5	7,5
3	Utilizator	40	52	7,5	7,5
4	Utilizator	84	84	6,5	7,5
5	Utilizator	53	51	7	7,5
6	Utilizator	79	42	7	8
7	Utilizator	67	68	7	8
8	Utilizator	64	31	5	8
9	Utilizator	50	39	6	8

10	Utilizator	35	40	1	7,5
11	Utilizator	78	53	5,5	8
12	Utilizator	59	34	7	8
13	Utilizator	29	67	7,5	7,5

2. Testul 2 - Atingerea celor patru repere (bețe)

Operator	Timp (s)	
	Myo Armband	Consolă de învățare
Utilizator 1	120	86
Utilizator 2	67	46
Utilizator 3	46	60
Utilizator 4	67	82
Utilizator 5	43	51
Utilizator 6	115	70
Utilizator 7	45	58
Utilizator 8	55	44
Utilizator 9	46	56
Utilizator 10	39	43
Utilizator 11	77	58
Utilizator 12	120	51
Utilizator 13	40	60

3. Testul 3 – Preluare și plasare obiect (sticlă PET) în cutie

r	Operato	Timp (s)		Scor de precizie	
		Myo Armband	Consolă de învățare	Myo Armband	Consolă de învățare
1	Utilizator	80	98	1	1
2	Utilizator	56	71	1	1
3	Utilizator	44	36	1	1

128 - ANEXE

4	Utilizator	63	124	1	1
5	Utilizator	15	60	0	1
6	Utilizator	45	51	0	1
7	Utilizator	68	58	1	1
8	Utilizator	92	46	1	1
9	Utilizator	45	33	1	0
10	Utilizator	25	49	1	1
11	Utilizator	104	56	1	1
12	Utilizator	31	55	1	1
13	Utilizator	37	44	1	1

ACTIVITATE ȘTIINȚIFICĂ

1. Lucrări indexate ISI

Kristof, R., Ciupe, V., Moldovan, C., Maniu, I., Banda, M., & Stoian, A. M. (2019, May). Arduino mobile robot with Myo Armband gesture control. In *2019 IEEE 13th International Symposium on Applied Computational Intelligence and Informatics (SACI)* (pp. 000294-000297). IEEE.

Kristof, R., Ciupe, V., Moldovan, C., Maniu, I., Banda, M., & Stoian, A. M. (2019, May). On the design of a myoelectric human-machine interface for the control of an industrial robot. In *2019 IEEE 13th International Symposium on Applied Computational Intelligence and Informatics (SACI)* (pp. 291-294). IEEE.

2. Lucrări indexate BDI

Kristof, R., Moldovan, C., Ciupe, V., Maniu, I., & Banda, M. (2019). Applications based on electromyography sensors. In *ITM Web of Conferences* (Vol. 29, p. 02007). EDP Sciences.

Moldovan, C., Ciupe, V., Filipescu, H., Kristof, R., & Dolga, V. (2021). Model-Free Continuous to Discrete Workspace Transformation and Path Planning of a 2DOF Serial Arm for Visual Obstacle Avoidance. In *New Advances in Mechanisms, Mechanical Transmissions and Robotics: MTM & Robotics 2020 2* (pp. 262-271). Springer International Publishing.

Banda, M., Ciupe, V., Moldovan, C., Maniu, I., & Kristof, R. (2019). Investigations on collaborative remote control of virtual robotic manipulators by using a Kinect v2 sensor. In *ITM Web of Conferences* (Vol. 29, p. 03004). EDP Sciences.