

Universitatea  
Transilvania  
din Brașov

FACULTATEA DE INGINERIE ELECTRICĂ  
ȘI ȘTIINȚA CALCULATOARELOR

# PROIECT DE DIPLOMĂ

Conducător științific:

Conf. dr. ing. Cociăș Tiberiu Teodor

Absolvent:

Adrian Dumitru Oncioiu

BRAȘOV, 2023

Departamentul de Automatică și Tehnologia Informației  
Programul de studii: Robotică

*ONCIOIU Adrian Dumitru*

# Robot pentru desen caricatural







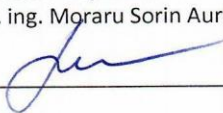
**Conducător științific:**

Conf. Dr. Ing. *COCIAȘ Tiberiu Teodor*

Brașov, 2023

# FIȘA PROIECTULUI DE DIPLOMĂ

Universitatea Transilvania din Brașov	Lucrare de licență/ proiect de diplomă/ disertație nr. ....
Facultatea de Inginerie Electrică și Știința Calculatoarelor	
Departamentul Automatică și Tehnologia Informației	Viza facultății
Programul de studii <b>ROBOTICĂ</b>	Anul universitar <b>2022-2023</b>
Candidat <b>Oncioiu GH. Adrian Dumitru</b>	Promoția <b>vară 2023</b>
Conducător științific <b>Conf.dr.ing. Cocias Tiberiu Teodor</b>	Grupa <b>4LF891</b>
PROIECT DE DIPLOMĂ	
Titlul lucrării: <b>Robot pentru desen caricatural</b>	
Problemele principale tratate: Problemele principale care vor fi tratate: 1. Studiul actual al cercetării 2. Descrierea sistemului 3. Implementarea software 4. Proiectarea și simularea sistemului Concluzii și cercetări viitoare	
Locul și durata practicii: Corp V, Laborator V III 15, 60 ore	
Bibliografie: [1] Macesanu G, Cocias T.T., Grigorescu S.M. , Sisteme de vedere artificială : Îndrumar de laborator, 2013 [2] Massimo Banzi, Getting Started with Arduino (Make: Projects), 2009 [3] Paul ViolaMichael J. Jones, Robust Real-Time Face Detection , 2004	
Aspecte particulare: - Nu sunt	
Primit tema la data de: 03.10.2022	
Data predării lucrării:	
Director departament, <i>Prof. dr. ing. Moraru Sorin Aurel</i>	Conducător științific, <i>Conf.dr.ing. Cocias Tiberiu Teodor</i>
Candidat, <i>Oncioiu GH. Adrian Dumitru</i>	

PROIECT DE DIPLOMĂ – VIZE –		
Data vizei	Capitole/ problemele analizate	Semnătura conducătorului științific
14.12.2022	Studiu asupra stadiului actual în domeniul și tema vizată; livrabil: raport	
15.03.2023	Proiectare / Arhitecturi, tehnologii, limbaje de programare, algoritmi de utilizat etc; livrabil: raport, document editat	
26.04.2023	Demonstrator / Proof of Concept / Aplicație; livrabil: prototip	
5.06.2023	Documentație; livrabil: document în formatul de proiect de pe situl ATI	
22.06.2023	Verificare procentaj similaritate [1%]	
APRECIEREA ȘI AVIZUL CADRULUI DIDACTIC ÎNDRUMĂTOR		
<p>Lucrarea îndeplinește criteriile de calitate pentru a putea fi susținută. Candidatul a prezentat progres remarcabil de-a lungul întregului semestru. Rezultatele experimentale obținute m-au convins să îmi dau acordul pentru susținerea lucrării de licență în cadrul sesiunii vară 2023</p>		
Data: 05.06.2023	ADMIS pentru susținere/ <u>RESPINS</u>	CONDUCĂTOR ȘTIINȚIFIC Conf.dr.ing. Cocias Tiberiu Teodor 
AVIZUL DIRECTORULUI DE DEPARTAMENT		
Data: <u>15.06.23</u>	ADMIS pentru susținere/ <u>RESPINS</u>	Director departament Prof. dr. ing. Moraru Sorin Aurel 
SUSȚINEREA PROIECTULUI DE DIPLOMĂ		
Sesiunea Vară 2023		
Rezultatul susținerii	PROMOVAT cu media:	
	RESPINS cu refacerea lucrării	
	RESPINS fără refacerea lucrării	
PREȘEDINTE COMISIE Prof. dr. ing. Grigorescu Sorin Mihai		

# Cuprins

Listă de figuri .....	5
1    Introducere .....	6
1.1   Stadiul actual .....	7
1.2   Motivație .....	9
1.3   Obiective și scopul proiectului .....	9
1.4   Structura lucrării .....	10
2    Aspecte Teoretice .....	11
2.1   Sisteme de vedere artificială .....	11
2.2   Modelare matematică .....	13
3    Sistem pentru desen caricatural .....	16
3.1   Achiziția de imagini .....	16
3.2   Filtrarea imaginilor .....	17
3.3   Extracție punctelor de interes .....	19
3.4   Controlul robotului pentru desenarea caricaturilor .....	21
3.5   Model geometric direct .....	22
3.6   Model cinematic invers .....	25
4    Proiectare robot caricatural .....	30
4.1   Modelarea 3D .....	30
4.2   Procesul de printare .....	31
4.3   Analiza prin element finit .....	33
5    Implementarea și evaluarea sistemului .....	35
5.1   Echipamentul utilizat .....	35
5.2   Procesul de realizare a sistemului robotic .....	37
5.3   Analiza și evaluarea rezultatelor .....	39
6    Concluzii și Perspective viitoare .....	42
6.1   Concluzii .....	42
6.2   Perspective viitoare .....	43
7    Bibliografie .....	45
Rezumat .....	47
Abstract .....	48
DECLARAȚIE PRIVIND ORIGINALITATEA .....	49

## LISTĂ DE FIGURI

---

- Figura 1.1 Robot Ai-Da.
- Figura 1.2 Robot E-David.
- Figura 3.1 Imagine erodată și dilatată.
- Figura 3.2 Imagine cu blur.
- Figura 3.3 Imagine la nivel monocrom.
- Figura 3.4 Imagine binară.
- Figura 3.5 Matrice imagine.
- Figura 3.6 Schemă geometrică.
- Figura 3.7 Traiectorie.
- Figura 3.8 Schemă cinematică inversă.
- Figura 3.9 Rezultate cupla 1.
- Figura 3.10 Rezultate cupla 2.
- Figura 3.11 Rezultate cupla 3.
- Figura 3.12 Poziție de pornire simulare.
- Figura 3.13 Poziție de oprire simulare.
- Figura 4.1 Ansamblu CAD.
- Figura 4.2 Prindere masa și pix CAD.
- Figura 4.3 Piese printate.
- Figura 4.4 Piese printate.
- Figura 4.5 Fișiere STL.
- Figura 4.6 Deplasare.
- Figura 4.7 Tensiune.
- Figura 5.1 Dynamixel AX12-A.
- Figura 5.2 74LS241N.
- Figura 5.3 Schemă conexiuni motor.
- Figura 5.4 Poziții motor.
- Figura 5.5 Conexiune Raspberry PI și Arduino Mega.
- Figura 5.6 Conexiuni Robot.
- Figura 5.7 Portrete inițiale.
- Figura 5.8 Portret caricatural.

## 1 INTRODUCERE

---

**Stadiul actual**

**Motivație**

**Obiectivele și scopul proiectului**

**Structura lucrării**

---

Proiectul constă în realizarea unui robot capabil să producă portrete caricaturale prin utilizarea unui Raspberry Pi 4 pentru procesarea imaginilor și a unui Arduino Mega 2560 cu rolul de a controla un braț articulat, împreună cu alte componente hardware.

Procesul începe cu captarea unei imagini, cu ajutorul unei camere web conectate la Raspberry Pi. Imaginea este procesată prin mai multe etape și punctele de interes sunt identificate prin utilizarea unui algoritm simplu. Coordonatele acestor puncte sunt trimise pe serială, pe rând, către Arduino Mega, care controlează mișcarea robotului pentru a lăsa câte un punct pe hârtie.

Pentru a permite transmiterea datelor între Arduino Mega și motoarele folosite pentru brațul robotului, s-a folosit o conexiune directă prin fire. Brațul articulat este confecționat din motoare și un șasiu special de la Robotis, iar suportul pentru pix și baza au fost printate din PLA, ceea ce ne-a permis să obținem o mișcare precisă și controlată.

În ceea ce privește avantajele, putem menționa faptul că robotul este capabil să producă portrete caricaturale, cu un nivel ridicat de precizie și control. De asemenea, utilizarea unui Raspberry Pi și a unui Arduino Mega a permis să avem o arhitectură modulară și ușor de modificat sau de actualizat. Deși există și unele dezavantaje, cum ar fi costul relativ mare al unor componentelor utilizate, necesitatea unor cunoștințe de programare și de electronică, acestea sunt depășite de beneficiile pe care le-am obținut.

Acest robot reprezintă un exemplu al utilizării tehnologiei moderne într-un mod creativ și util. Combinația dintre Raspberry Pi și Arduino Mega, împreună cu alte componente hardware și software, a permis să obținem un produs final, care poate fi utilizat în diverse scopuri și aplicații. În ziua de azi, portretele caricaturale devin tot mai populare, orice om își realizează o astfel de amintire în diverse locuri.

## 1.1 STADIUL ACTUAL

Roboții artiști reprezintă o ramură în plină dezvoltare și relativ nouă în robotică, având ca obiectiv crearea unor sisteme autonome capabile să producă opere de artă. Această abordare inovatoare implică combinarea cunoștințelor din domeniul științei cu cele din artă și design, reprezentând un pas important în evoluția tehnologiei contemporane. [1]

Într-o lume în continuă schimbare și avans tehnologic, roboții artiști deschid noi perspective și provocări în ceea ce privește creativitatea și expresivitatea artistică. Capacitatea lor de a inova în procesul de creație reprezintă o îmbinare interesantă între abilitățile tehnologice și potențialul artistic. De la pictură și muzică, până la sculptură și dans, roboții artiști explorează diverse forme de expresie artistică, aducând o perspectivă inedită în lumea artei. [1] [2]

Această intersecție între tehnologie și artă aduce beneficii și provocări interesante. Pe de o parte, roboții artiști pot elibera creativitatea umană de constrângerile fizice și tehnice, deschizând noi orizonturi pentru explorarea și materializarea ideilor artistice. Pe de altă parte, apare întrebarea asupra rolului și semnificației artistului uman în această lume în continuă evoluție tehnologică. Astfel, dezvoltarea roboților artiști ne provoacă să reflectăm asupra naturii artei și să explorăm limitele creativității umane. [1] [2]

Prin urmare, fascinația mea pentru tehnologie și artă m-a condus către acest proiect, unde îmi propun să contribui la dezvoltarea roboților artiști, încercând să deschid noi orizonturi în ceea ce privește interacțiunea dintre tehnologie și creativitate umană.

Printre cei mai cunoscuți roboți artiști se numără Ai-Da, o creație inovatoare a lui Aidan Meller. Ai-Da este capabilă să deseneze portrete cu creionul, într-un mod remarcabil de precis și detaliat. Aceasta combină abilitățile tehnologice avansate, cum ar fi recunoașterea facială și controlul motor, cu expresia artistică autentică. Prin intermediul algoritmilor sofisticăți și al învățării automate, Ai-Da poate capta esența și trăsăturile distincte ale subiecților săi, oferind o perspectivă captivantă asupra interacțiunii dintre tehnologie și artă. [3]





Figura 1.1 Robot Ai-Da [3]

Un alt robot artist notabil este "E-David", dezvoltat la Universitatea din Stuttgart. E-David se remarcă prin capacitatea sa de a crea picturi abstracte și de a aplica straturi de culoare cu o precizie uimitoare. Prin utilizarea unor algoritmi și a unui sistem avansat de viziune artificială, acest robot poate analiza și interpreta obiectele și scenele în timp real, generând compoziții artistice surprinzătoare. Fiecare pensulă și nuanță aplicată de E-David este rezultatul unei alegeri creative și a unei abordări autonome, evidențiind abilitățile sale artistice remarcabile. [4]



Figura 1.2 Robot E-David [4]

Acești roboți artiști demonstrează că tehnologia și creativitatea pot merge mână în mână, deschizând noi orizonturi în lumea artei contemporane. Prin capacitatea lor de a genera opere unice și de a exprima emoții și concepte artistice, acești roboți provoacă limitele tradiționale ale creației artistice și explorează noi posibilități de expresie. [3] [4]

Artiștii roboți precum Ai-Da, E-David și mulți alții, reprezintă exemple fascinante ale îmbinării tehnologiei și artei. Prin abordarea lor inovatoare și capacitățile lor artistice avansate, acești roboți deschid noi perspective și debateri cu privire la rolul și influența tehnologiei în domeniul artistic. Sunt cu siguranță niște pionieri în explorarea și extinderea creativității umane, deschizând drumul către un viitor în care interacțiunea dintre tehnologie și artă va continua să evolueze și să ne uimească. [3] [4]

## 1.2 MOTIVAȚIE

Manipulatorul, cunoscut și sub denumirea de robot, reprezintă un sistem mecatronic avansat, care imită caracteristicile unui braț uman și poate fi utilizat într-o varietate de sarcini. Acesta a fost creat de echipe de ingineri specializați cu scopul de a înlocui activitățile periculoase sau dăunătoare pentru oameni, pe lângă acest aspect, brațul robotic poate ușura efortul fizic și poate înlocui chiar și activitățile umane în anumite situații. Această inovație reprezintă una dintre cele mai remarcabile realizări ale tehnologiei moderne, fiind indispensabilă în era consumului. Într-un interval de timp scurt, roboții au devenit esențiali în toate domeniile industriale, contribuind la producerea de echipamente complexe folosind o precizie maximă, ce nu pot fi realizate de om.

Motivația mea pentru a mă implica în acest proiect derivă din pasiunea pentru tehnologie și pentru felul în care aceasta poate aduce beneficii oamenilor și societății. Sunt ferm convins că dezvoltarea și utilizarea roboților reprezintă un pas important către un viitor în care interacțiunea dintre om și tehnologie va deveni mai eficientă și mai productivă. Cu fiecare avans tehnologic în domeniul roboticii, deschidem ușa către noi oportunități și potențiale nelimitate. Prin implementarea acestor sisteme avansate, putem depăși limitele capacităților noastre umane și putem realiza lucruri incredibile. Este impresionabil să observăm cum roboții pot îndeplini sarcini complexe, permițându-ne să ne concentrăm mai mult asupra creativității și inovației. Încrederea mea în puterea tehnologiei m-a condus către această inițiativă.

Pe lângă pasiunea pentru tehnologie și interacțiunea dintre om și mașină, m-am simțit atras și de lumea artei și de modalitățile prin care aceasta poate fi creată și mai mult de cum poate fi interpretată. Descoperirea că există chiar și roboți artiști m-a captivat complet, aceștia fiind dezvoltați cu scopul de a crea opere artistice unice și fascinante.

## 1.3 OBIECTIVE ȘI SCOPUL PROIECTULUI

Robotul artist este un braț mecatronic care poate desena o caricatură după o imagine dată. Acesta este format dintr-un braț cu trei cuple de rotație, două pe axa Z și una pe axa Y. Cuplele de rotație permit brațului să se deplaseze în spațiu și să coboare pixul pe hârtie pentru a crea desenul.

Brațul articulat este format din motoare Dynamixel AX12-A și este controlat de un Raspberry Pi care prelucrează imaginile și trimite comenzi către un Arduino Mega pentru a pune în mișcare brațul. În prezent, pentru realizarea caricaturii, se folosește o imagine din care se

extrage o matrice binară, iar la fiecare valoare „1” din matricea binară, robotul pune un punct pe hârtie.

Deși proiectul este încă în stadiul său incipient, am obținut deja progrese semnificative. În viitor, intenționez să îmbunătățesc algoritmul de procesare a imaginilor și să adaug mai multe funcții pentru a permite robotului să creeze imagini mai complexe. De asemenea, doresc să optimizez mișcarea brațului pentru a obține o precizie mai mare și o viteză mai mare de desenare.

Scopul final al acestui proiect este de a crea un robot artist care să fie capabil să creeze imagini și desene complexe într-un mod autonom. În timp ce acest lucru poate părea îndepărtat, există convingeri că progresele tehnologice recente, în domeniul inteligenței artificiale și al roboților, vor face acest lucru posibil în viitorul apropiat.

Astfel, proiectul de robot artist reprezintă un pas important în evoluția inteligenței artificiale și a roboților. Cu ajutorul tehnologiei și al creativității, sper să se contribuie la dezvoltarea unui robot care poate crea opere de artă cu adevărat remarcabile și să deschidă noi orizonturi în domeniul inteligenței artificiale și al roboticii.

## 1.4 STRUCTURA LUCRĂRII

Pe parcursul lucrării, se va prezenta un proiect fizic care constă într-un braț articulat capabil să deseneze o caricatură după o imagine. În cadrul lucrării se va detalia echipamentul utilizat, inclusiv componentele hardware, procesul software și o descriere de ansamblu a brațului articulat, împreună cu specificațiile tehnice. De asemenea, se abordează subiecte precum procesarea imaginilor și algoritmi utilizați pentru extragerea punctelor de interes și segmentarea imaginilor. Lucrarea va include și o secțiune dedicată modelării matematice, inclusiv modelul geometric și cinematic, împreună cu simularea în Matlab. Se va prezenta, de asemenea, procesul de modelare 3D, inclusiv crearea modelelor și imprimarea acestora. În final, se analizează performanțele sistemului, se evaluează modelul matematic și modelul 3D. După care, o comparație între rezultatele obținute și obiectivele proiectului poate fi interesantă. Lucrarea se va încheia cu concluzii și perspective viitoare, împreună cu o bibliografie detaliată a surselor utilizate.

## 2 ASPECTE TEORETICE

---

Sisteme de vedere artificială

Modelare matematică

---

### 2.1 SISTEME DE VEDERE ARTIFICIALĂ

Procesarea imaginilor reprezintă un domeniu amplu și esențial în dezvoltarea sistemelor de viziune artificială și robotică. Acest domeniu se ocupă cu aplicarea unor algoritmi pentru modificarea și analizarea imaginilor cu scopul de a extrage informații relevante și de a realiza diverse operațiuni cu ele. Procesarea imaginilor este utilizată într-o gamă variată de aplicații, cum ar fi recunoașterea de obiecte, segmentarea imaginilor, detecția de caracteristici și multe altele. [5] [6]

Una dintre primele etape în procesarea imaginilor este pregătirea imaginii prin aplicarea unor operații de filtrare și ajustare pentru a îmbunătăți calitatea și relevanța informațiilor conținute în imagine. Aceste operații pot include redimensionarea, corecția contrastului, eliminarea zgomotelor, îmbunătățirea clarității imaginii și multe altele. După care, urmează partea în care se aplică algoritmi și tehnici specifici pentru a extrage caracteristici și informații relevante din imagine, cum ar fi detectarea de contururi, identificarea obiectelor sau extragerea punctelor de interes. [5] [6]

Procesarea imaginilor joacă un rol fundamental în transformarea imaginilor de referință în coordonate și instrucțiuni pentru mișcarea brațului robotic. Folosind biblioteca Open CV, s-au implementat algoritmi și tehnici de procesare a imaginilor pentru a realiza această transformare. Detaliind mai departe aceste procese în 3 părți, achiziția de imagini, unde se preia o imagine în formatul necesar, filtrare unde se pregătește imaginea și în final segmentarea imaginii, în așa fel încât să se obțină ceea ce avem nevoie de la ea. Algoritmi precum eroziunea,

dilatarea, blurarea utilizând un filtru gaussian și transformarea în imagine binară sunt aplicate în etapele de pregătire a imaginilor și extragerea de informații necesare. [5] [6] [7] [8]

Filtrul Gaussian este unul din filtrele ce au făcut o mare diferență în rezultatele obținute. Acest filtru prelucrează imaginile pentru a reduce zgomotul și pentru a estompa detaliile fine. Aplică o medie ponderată în jurul fiecărui pixel din imagine, utilizând o funcție Gaussiană ca bază pentru ponderare. Fiind o funcție matematică continuă, descrie o curbă în formă de clopot ce se utilizează în calcularea ponderilor pentru media aplicată, în jurul fiecărui pixel. Imaginile cărora li se aplică un astfel de filtru, vor suferi un fenomen de estompare sau difuzie a contrastului, datorită ponderilor mai mari acordate pixelilor din vecinătatea fiecărui pixel în timpul medierii. Majoritatea aplicațiilor ce utilizează vederea artificială se folosesc de această funcție într-un fel sau altul. [5] [6]

Ecuția matematică a filtrului Gaussian poate fi exprimată ca:

$$G(x,y) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.1)$$

*unde:  $G(x,y)$  este valoarea filtrului Gaussian la coordonatele  $(x,y)$ ;*

*$\sigma$  reprezintă deviația standard a distribuției Gaussiene, care controlează cât de mult se extinde filtrul*

Această ecuație reprezintă un filtru bidimensional Gaussian simetric, care este utilizat pentru a aplica un efect de eng.blur sau estompare asupra unei imagini sau a unui set de date bidimensional. Valorile negative ale lui „x” și „y” reprezintă poziții la stânga sau sus față de centrul filtrului, iar valorile pozitive reprezintă poziții la dreapta sau jos față de centrul filtrului.

La baza filtrului Gaussian stă o operație matematică fundamentală, convoluția ce se utilizează pentru aplicarea unui filtru asupra imaginii. Operația de convoluție între două funcții, în cazul nostru imaginea de intrare și kernelul Gaussian, produce o nouă funcție care reprezintă combinația ponderată a valorilor celor două funcții. Această combinație ponderată reprezintă media ponderată a valorilor pixelilor din jurul fiecărui pixel al imaginii. Filtrul Gaussian este caracterizat de distribuția Gaussiană a valorilor din kernel. Această distribuție Gaussiană determină cum sunt ponderate valorile pixelilor în timpul convoluției. Valorile mai apropiate de centrul kernelului primesc ponderi mai mari, în timp ce valorile mai îndepărtate de centrul kernelului primesc ponderi mai mici. Un aspect important al convoluției este că ea poate fi realizată într-un mod eficient utilizând transformata Fourier. Transformarea Fourier permite reprezentarea imaginii și a kernelului în domeniul frecvenței, ceea ce facilitează calculul convoluției prin înmulțirea spectrelor de frecvență și apoi prin transformarea inversă Fourier pentru a obține rezultatul în domeniul spațial. [5] [6]

Ecuția matematică a convoluției între două funcții „f” și „g” este reprezentată astfel:

$$(f \cdot g)(t) = \int_{-\infty}^{+\infty} f(\tau) \cdot g(t - \tau) d\tau \quad (2.2)$$

unde:  $\tau$  este variabila de integrare

$t$  este variabila de rezultat între funcțiile „f” și „g”.

Această ecuație arată că fiecare valoare a funcției rezultate „(f\*g)(t)” este obținută prin integrarea produsului valorilor funcției „f” și funcției „g” într-o anumită fereastră de timp (sau spațiu). Rezultatul este o nouă funcție care reprezintă combinația ponderată a valorilor celor două funcții. [5] [6]

## 2.2 MODELARE MATEMATICĂ

Modelarea matematică reprezintă o componentă fundamentală în dezvoltarea și integrarea sistemelor robotice. Prin intermediul ei, putem dezvolta și analiza comportamentul și performanțele acestor sisteme utilizând concepte și tehnici matematice specifice. În acest capitol, vom aborda în mod teoretic modelarea matematică a unu braț articulat. Aceasta implică descrierea relațiilor geometrice și cinematice ale componentelor sale, precum și definirea parametrilor și variabilelor relevante. Prin intermediul modelării matematice, putem obține o reprezentare abstractă și formală a robotului, care ne permite să analizăm și să prezicem comportamentul său în diverse scenarii și configurații. Astfel, vom obține o mai bună înțelegere a modului în care brațul robotic se mișcă și interacționează cu mediul înconjurător, contribuind astfel la optimizarea proiectării și controlului sistemelor robotice. Mai departe se vor analiza doua modele pentru proiectul acesta, modelul geometric si modelul cinematic.[9]

Modelul geometric care se va utiliza pornește de la următoarele ipoteze fundamentale. În primul rând, se presupune că fenomenele care vor fi descrise în acest model sunt incluse în relația de cauzalitate dintre poziția unghiulară a fiecărei cuple și orientarea împreună cu poziția efectorului, în cazul acesta, postul fiind ocupat de pixul artistului robot. În al doilea rând, se ignoră fenomenele cinematice, ceea ce înseamnă că nu se vor lua în considerare vitezele. În plus, se consideră că fenomenele sunt statice, adică nu există variații în timp și nu există interacțiuni cu timpul în model. Relația de cauzalitate se poate studia în două moduri diferite. În prima metodă, cauza este impusă prin variabilele unghiulare, iar cu ajutorul metodei geometriei directe se poate determina poziția și orientarea efectorului. Iar cea de a doua metodă, cauza este impusă de poziția și orientarea efectorului și se determină valorile variabilelor unghiulare prin metoda geometriei inverse. Constructiv, manipulatorul este format dintr-un lanț cinematic compus din (n+1) elemente interconectate prin articulații și cuple cinematice, care pot fi rotații sau translații, iar transformările omogene au rolul de a descrie

aceste mișcări. Fiecare element are propriul său sistem de referință, iar poziția și orientarea acestuia sunt determinate de o transformare omogenă față de un reper cunoscut. Astfel, postura obiectului manipulat este compusă din poziția originii sistemului de coordonate în care se definește obiectul și orientarea acestuia. Poziția unui punct este reprezentată de cele trei coordonate (X, Y, Z), în timp ce orientarea unui sistem de referință este definită prin intermediul unui ansamblu de trei unghiuri. Postura obiectului manipulat depinde de postura tuturor elementelor lanțului cinematic, iar pentru a descrie această postură, ne putem raporta la sistemul de coordonate asociat acestuia. [9] [10] [11]

Modelarea geometrică a lanțului cinematic în robotică este necesară pentru a descrie relațiile dintre mișcarea articulațiilor și mișcarea brațului robotic în ansamblu. Aceasta ne permite să calculăm poziția și orientarea brațului robotic în funcție de unghiurile articulațiilor și să rezolvăm două probleme: cinematica directă și cinematica inversă. [10] [11]

Cinematica directă se referă la determinarea poziției și orientării efectorului (punctul final, pixul) al brațului robotic, cunoscând valorile unghiurilor articulațiilor. Folosind cunoștințe de trigonometrie pentru a calcula transformările omogene, care descriu relațiile spațiale între sistemele de coordonate succesive din lanțul cinematic, putem obține poziția și orientarea efectorului în funcție de valorile unghiurilor articulațiilor. [10] [11]

Pe de altă parte, cinematica inversă se referă la determinarea valorilor unghiurilor articulațiilor, cunoscând poziția și orientarea dorită a efectorului. Aici, utilizăm matricea Jacobianului pentru a calcula vitezele articulațiilor în funcție de viteza efectorului și inversăm acest proces pentru a obține valorile unghiurilor articulațiilor. [10] [11]

În cadrul modelului cinematic, se utilizează și formalismul Reperului Translatat, care este un formalism simplificat utilizat în configurații speciale ale lanțului cinematic. Acesta se bazează pe definirea parametrilor cinematici, cum ar fi lungimile segmentelor brațului, unghiurile de rotație ale articulațiilor și pozițiile articulațiilor față de sistemul de coordonate global. [10] [11]

Un alt aspect important în robotică face referire la traiectoriile articulate. Acestea pot fi descrise prin diferite metode, cum ar fi metodele polinomiale sau metoda Bang-Bang, care se ocupă de definirea funcțiilor de poziție, viteză și accelerație în funcție de timp. Aceste traiectorii reprezintă mărimile dorite ale sistemului de control și sunt utilizate pentru a ghida mișcarea brațului robotic într-o anumită configurație. [10] [11] [12]

După cum se observă, calculul matricei Jacobianului este obligatoriu. Aceasta reprezintă un concept fundamental în domeniul roboticii. Rezultatul este obținut din funcțiile de poziție și orientare și are proprietatea de a descrie o relație matriceală  $2 \times N$  între vitezele variabilelor

generalizate ale cuplurilor și vitezele efectorului. Calculul Jacobianului poate fi realizat numeric, folosind algoritmul de propagare a vitezei, dar poate fi și reprezentat sub formă analitică, explicită, permițând analiza influenței fiecărei cuplă asupra vitezelor efectorului. Prin urmare, Jacobianul este necesar atât în modelul geometric, cât și în modelele cinematice. [9] [10] [11]



## 3 SISTEM PENTRU DESEN CARICATURAL

---

Achiziția de imagini

Filtrarea imaginilor

Extracție punctelor de interes

Controlul robotului pentru desenarea caricaturilor

Model geometric direct

Model cinematic invers

---

Mai departe se vor descrie pașii ce s-au urmat în realizarea și implementarea acestui proiect. Împărțirea acestor pași, în mai multe etape constructive, a realizat un progres organizat și a împărțit în mod eficient timpul pentru obținerea rezultatelor necesare. Logica prin care s-a organizat acest proiect, precum urmează, a stat la baza progresului treptat obținut. Începând cu partea de vedere artificială, s-a cunoscut ce se poate obține, după care s-a proiectat un braț articulat ce satisfacea nevoile și se completa cu datele obținute în partea anterioară.

### 3.1 ACHIZIȚIA DE IMAGINI

Algoritmul începe prin citirea unei imagini, acest lucru se poate face indirect cu o imagine salvată anterior sau direct realizându-se o captură cu o camera web conectată la sistem. Ambele variante sunt ușor de implementat prin librăria Open CV, ce ne oferă posibilitatea utilizării funcțiilor: „cv2.imread()” sau „cv2.VideoCapture()”. Prima funcție va primi în paranteză ca parametrii fișierul de tip „.jpg” sau un format asemănător iar cea de a doua va primi o valoare de la „0” la „n” ce reprezintă numărul camerei conectate. [8] [13]

Ulterior, redimensionarea imaginii reprezintă un pas prioritar în procesarea imaginilor, având scopul de a se adapta la spațiul de lucru al brațului robotic și de a optimiza performanța algoritmului utilizat. Inițial, dimensiunile imaginii pot fi mult mai mari decât spațiul disponibil,

dar algoritmul trebuie implementat pentru momentul ulterior când se va îmbina cu robotul. Acest lucru ne permite adaptarea algoritmului rapid către alți roboți articulați, putându-se doar adapta dimensiunile spațiului de lucru cu dimensiunea imaginii pe care o vom procesa. Altfel, conduce la ineficiență în mișcările brațului robotic și poate genera erori. Mai concret, pentru a remedia această situație, se aplică o redimensionare a imaginii folosind un factor de scalare „fx” și „fy” prin intermediul funcției „cv2.resize()”. Scopul este de a obține dimensiuni adecvate, de exemplu 120x170 de pixeli în cazul acestui proiect, care să fie compatibile cu spațiul de lucru. Prin urmare, această redimensionare nu doar optimizează spațiul de lucru, ci contribuie și la reducerea timpului de procesare și a resurselor necesare. [6] [13]

### 3.2 FILTRAREA IMAGINILOR

Pasul următor, la fel de important ca oricare altul, cuprinde aplicarea mai multor filtre peste imaginea de lucru obținută după redimensionare. Acești pași încearcă și chiar reușesc să simplifice imaginea și să excludă informațiile neesențiale.

În primă fază a acestei etape, s-a început prin aplicarea unor filtre de eroziune și dilatare ce reprezintă două operații morfologice fundamentale utilizate în prelucrarea imaginilor. În cazul acesta, se folosesc funcțiile „cv2.erode()” și „cv2.dilate()” pentru a aplica aceste operații asupra imaginii. Eroziunea constă în reducerea dimensiunii regiunilor de interes prin erodarea pixelilor din vecinătatea acestora. Pe de altă parte, dilatarea implică extinderea regiunilor de interes prin adăugarea de noi pixeli în jurul acestora. Ambele au ca scop final eliminarea zgomotelor din imagine, pentru a evidenția elementele de interes și ale separa de cele suplimentare. Prin controlul numărului de iterații și a dimensiunii elementului structural eng.kernel, se poate obține un nivel mai precis de ajustare în funcție de necesități. Prin aplicarea erodării, se îndepărtează micile detalii și contururi nedorite din imagine, oferind un aspect mai simplificat al obiectelor. Iar, dilatarea este utilizată pentru a umple golurile din interiorul obiectelor și pentru a conecta componentele separate ale acestora. Este important să se ajusteze parametrii operațiilor de erodare și dilatare în funcție de natura și caracteristicile specificului imaginii, pentru a obține rezultatele dorite. Alegerea numărului de iterații și a dimensiunii kernelului depinde de detaliile pe care dorim să le evidențiem sau să le ajustăm, un exemplu se poate vedea în figura 3.1. [8] [13] [14]



*Figura 3.1 Imagine erodată și dilatăată*

Operația de estompare sau mai bine cunoscută drept eng.blur, este utilizată pentru a reduce zgomotul imaginii. În acest algoritm, se utilizează funcția "cv2.GaussianBlur()" pentru a aplica estomparea imaginii dilatate. S-a folosit aceasta funcție în locul celei clasice „cv2.blur()” deoarece filtrul gaussian oferă o estompare mai naturală și mai plăcută, acesta bazându-se pe ponderi gaussiene. Aceste ponderi atribuie o intensitate mai mare pixelilor din vecinătatea centrală a filtrului și o intensitate mai mică pixelilor din vecinătățile mai îndepărtate. De altfel, această funcție permite controlul mai precis al estompării prin ajustarea deviației standard a distribuției gaussiene și prin modificarea valorii deviației standard, puteți regla cât de mult sunt estompate detaliile din imagine permițând obținerea de rezultate mai fine. Acest lucru conduce la o estompare mai uniformă și mai naturală a detaliilor din imagine. Prin acest proces, detaliile fine și zgomotul sunt reduse, iar obiectele de interes rămân mai clare și mai ușor de identificat. Alegerea dimensiunii filtrului poate afecta nivelul de estompare și detaliile reținute în imagine. Un filtru de estompare cu o dimensiune mai mare va produce o estompare mai pronunțată, eliminând mai mult zgomotul, dar poate duce și la pierderea unor detalii importante. Pe de altă parte, un filtru de estompare cu o dimensiune mai mică va conserva mai multe detalii, dar poate permite pătrunderea unui nivel mai mare de zgomot în imagine. Prin ajustarea dimensiunii filtrului de estompare în funcție de specificul imaginii și de obiective, am putut obține un rezultat optim care să îmbunătățească claritatea și să reducă zgomotul în mod corespunzător. Funcția necesită niște parametrii introduși ce reprezintă imaginea de intrare, dimensiunea kernelului și deviația standard. Dimensiunea kernelului reprezintă fereastra de pondere utilizată medierii și este construită dintr-o pereche de numere impare. În figura 3.2, se poate analiza cum arată un astfel de filtru peste imaginea pe care o procesăm. [14]



*Figura 3.2 Imagine cu blur*

Convertirea imaginii color în nivel de gri (monocrom) reprezintă următorul pas prin care a trecut imaginea, figura 3.3. Funcția „cv2.cvtColor()” are rolul de a realiza această conversie asupra imaginii deja estompate. Prin acest proces, informațiile referitoare la culoare sunt eliminate, iar informațiile despre intensitatea luminii sunt păstrate, culorile fiind o informație în plus față de ceea ce avem nevoie. Imaginea rezultată conține doar nuanțe de gri, ceea ce simplifică procesarea ulterioară și extragerea caracteristicilor necesare. Conversia la nivel de gri poate fi utilă în situații în care informațiile despre culoare nu sunt necesare, având astfel o analiză mai simplă și mai rapidă a imaginii. Prin urmare, prin convertirea imaginii color în nivel de gri, se facilitează procesul de analiză și extragere a caracteristicilor relevante, îmbunătățind astfel performanța și eficiența algoritmului utilizat. [8] [13]



*Figura 3.3 Imagine la nivel monocrom*

### **3.3 EXTRACȚIE PUNCTELOR DE INTERES**

Cel mai important pas a fost identificarea punctelor de interes și prelucrarea acestor date pentru a le putea utiliza. Această etapă de segmentare a avut obiectivul de a crea o matrice binară în care coordonatele „x” să reprezinte numărul coloanei, iar coordonatele „y” numărul liniei. Aceste date sunt transmise drept intrări către brațul robotic pentru a atinge coordonatele succesiv în scopul realizării portretului caricatural. [13]

Transformarea imaginii într-o imagine binară a reprezentat o etapa cruciala în acest proiect. Scopul a fost de a delimita clar obiectele de interes și de a le putea identifica și coordona în spațiul 2D. Folosind funcția „cv2.treshold()” am impus aplicarea unui prag denumit eng.treshold. Pragul reprezintă o valoare limită, iar fiecare pixel din imagine este comparat cu acest prag. În funcție de relația dintre valoarea pixelului și prag, acesta va fi considerat parte din obiectul de interes sau fundal. Dacă valoarea pixelului este mai mare ca pragul, acesta va fi considerat parte din obiectul de interes și va fi setat la o valoare specificată, „255”. În schimb, dacă valoarea pixelului este mai mică sau egală cu pragul, acesta va fi considerat fundal și va fi setat la o altă valoare specificată, „0”.

Rezultatul este o imagine binară în care obiectele de interes sunt reprezentate de pixeli negri, iar restul imaginii este reprezentat de pixeli albi, în funcție de cum se setează parametrii tresholdului. Această transformare simplifică ulterior procesarea și analiza imaginii, deoarece obiectele de interes devin evidente și ușor de segmentat. Este important să alegem pragul corespunzător în funcție de specificul imaginii și de obiective, fiecare imagine având caracteristica ei. Orice diferență de luminozitate sau contrast poate influența total rezultatul. [7] [13]



*Figura 3.4 Imagine binară*

După segmentarea imaginii și obținerea imaginii binare, următorul pas este convertirea acestei imagini într-o matrice binară. Scopul acestei etape este de a transforma informațiile pixelilor într-o reprezentare matriceală, astfel încât să putem accesa fiecare punct în mod individual. Am parcurs lista de pixeli și am înlocuit valorile sub pragul specific „50” cu „1” și valorile mai mari cu „0”. Astfel, am obținut o matrice cu elemente binare, unde „1” reprezintă puncte de interes și „0” reprezintă puncte de fond. Mai departe, folosind o iterație prin matrice, am parcurs fiecare element și am atribuit valorile corespunzătoare într-o matrice. Linia și coloana fiecărui element din matrice reprezintă coordonatele sale în imagine. De exemplu, valoarea „1” de la linia  $i$  și coloana  $j$  reprezintă punctul de interes cu coordonatele  $(i, j)$ , acesta

trebuie atins de către brațul robotic în timpul procesului de desenare. Setul de date va fi trimis către Arduino Mega prin portul serial. Pentru fiecare punct de interes cu valoarea „1” în matrice, am format un șir de caractere reprezentând coordonatele „x” și „y” ale punctului. Aceste coordonate au fost normalizate și transformate în șiruri de caractere pentru a fi trimise către Arduino Mega prin portul serial. [7] [13]

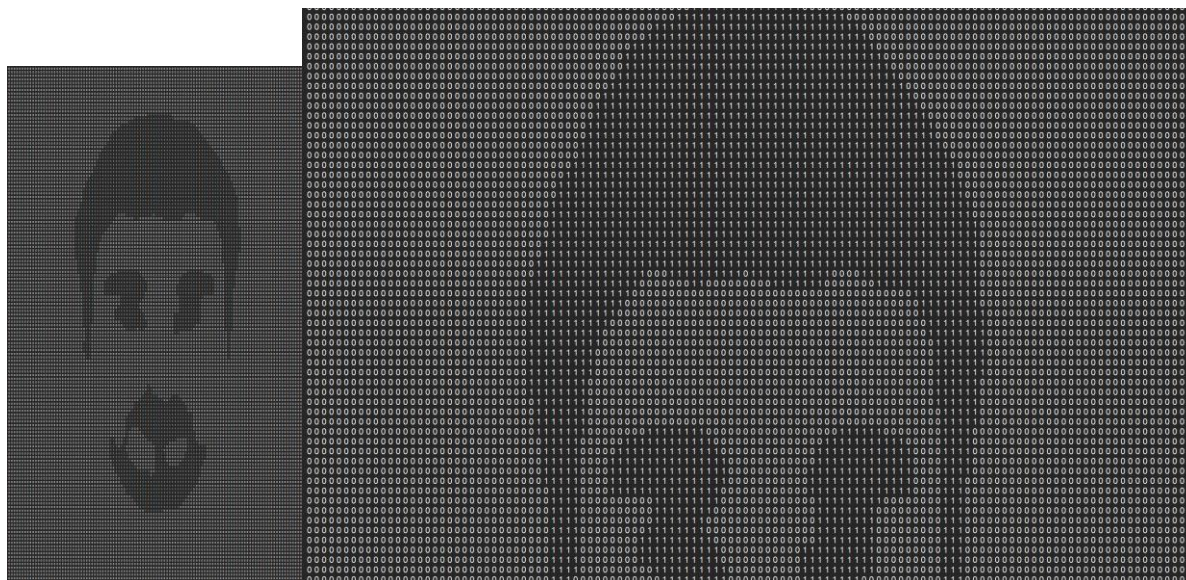


Figura 3.5 Matrice imagine

### 3.4 CONTROLUL ROBOTULUI PENTRU DESENAREA CARICATURILOR

În acest subcapitol, se examinează modul în care informațiile obținute anterior din procesarea imaginilor sunt utilizate pentru a controla mișcările robotului în vederea desenării caricaturilor. Vom explora legătura directă între procesarea imaginilor și modelele matematice în cadrul sistemului, iar mai departe analizăm modul în care informațiile obținute din procesarea imaginilor sunt utilizate pentru a controla mișcările robotului și a realiza desenele.

Cum s-a punctat în capitolele anterioare, procesarea imaginilor joacă un rol fundamental în obținerea datelor necesare pentru controlul mișcărilor robotului. Utilizând biblioteca Open CV în limbajul de programare Python s-au extras și transmis coordonatele valorilor „1” din matricea imaginii. Mai departe aceste informații despre coordonatele obținute trebuie îmbinate cu partea matematică pentru a determina postura brațului articulat în atingerea acestor puncte.

Modelul geometric direct este utilizat pentru a calcula poziția și orientarea sistemului de coordonate al efectorului. Acesta implică transformări omogene și matricea Jacobianului pentru a determina mișcările precise ale robotului.

Modelarea matematică prin cinematica inversă este apoi aplicată pentru a determina postura robotului necesară pentru a atinge coordonatele „x” și „y” ale unui punct specific.

Aceasta permite controlul precis al poziției și orientării brațului robotic, astfel încât pixul să fie plasat cu exactitate pe hârtie în punctele determinate.

Astfel, procesarea imaginilor și modelele matematice se îmbină în mod sinergic pentru a asigura controlul robotului în procesul de desenare caricaturală. Procesarea imaginilor furnizează datele necesare, iar modelele matematice traduc aceste date în mișcări precise ale robotului, realizând desenele dorite.

### 3.5 MODEL GEOMETRIC DIRECT

Modelul geometric direct rezolvă problema calculului poziției și orientării sistemului de coordonate al efectorului, prin determinarea transformărilor omogene de la un sistem de referință la altul, cu scopul de a descrie modificarea posturii. Acest formalism se bazează pe conceptul Denavit-Hartenberg, care definește transformările omogene pentru elementele unui robot. Astfel, pentru fiecare element al robotului, se obține un sistem de coordonate propriu.

În acest model, cunoscute fiind valorile „q1”, „q2” și „q3”, putem determina posturile transformate pentru orice puncte și putem obține desenul configurației. Mai departe sunt prezentate matricele cuplelor de rotație pe axa z, y și pentru efector s-a considerat în calcul, o matrice de translație pe axa x, ecuațiile 3.1, 3.2, 3.3 și 3.4. Evident s-a determinat și matricea Jacobianului, ecuația 3.12, pentru a trece mai departe în următorii pași. [10] [12]

$${}^{i-1}_iT_1 = \begin{bmatrix} c_{qi} & -s_{qi} & 0 & t_x \\ s_{qi} & c_{qi} & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^{i-1}_iT_{R_z} \quad (3.1)$$

$${}^{i-1}_iT_2 = \begin{bmatrix} c_{qi} & -s_{qi} & 0 & t_x \\ s_{qi} & c_{qi} & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^{i-1}_iT_{R_z} \quad (3.2)$$

$${}^{i-1}_iT_3 = \begin{bmatrix} c_{qi} & 0 & s_{qi} & t_x \\ 0 & 1 & 0 & t_y \\ -s_{qi} & 0 & c_{qi} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^{i-1}_iT_{R_y} \quad (3.3)$$

$${}^{i-1}_iT_4 = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^{i-1}_iT_{D_x} \quad (3.4)$$

unde:  $q_i$  este variabila unghiulară din cuple;  
 $t_x$ ,  $t_y$  și  $t_z$  sunt parametrii geometrici ai elementelor.

$$T1 = \begin{bmatrix} c_{q2} & -s_{q2} & 0 & 0 \\ s_{q2} & c_{q2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = T1_{Rz} \quad (3.5)$$

$$T2 = \begin{bmatrix} c_{q2} & -s_{q2} & 0 & 0.092 \\ s_{q2} & c_{q2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = T2_{Rz} \quad (3.6)$$

$$T3 = \begin{bmatrix} c_{q1} & 0 & s_{q1} & 0.11 \\ 0 & 1 & 0 & 0 \\ -s_{q1} & 0 & c_{q1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = T3_{Ry} \quad (3.7)$$

$$T4 = \begin{bmatrix} 1 & 0 & 0 & 0.055 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = T4_{Dx} \quad (3.8)$$

unde:  $q1, q2$  sunt variabilele unghiulare din cuplele 1 și 2.

$${}^0_1W = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.9)$$

$${}^0_2W = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.10)$$

$${}^0_3W = \begin{bmatrix} -C_{q1}S_{q2} - C_{q2}S_{q1} \\ C_{q1}C_{q2} - S_{q1}S_{q2} \\ 0 \end{bmatrix} \quad (3.11)$$

$$J = \begin{bmatrix} \frac{\partial P_x}{\partial q_1} & \frac{\partial P_x}{\partial q_2} & \frac{\partial P_x}{\partial q_3} \\ \frac{\partial P_y}{\partial q_1} & \frac{\partial P_y}{\partial q_2} & \frac{\partial P_y}{\partial q_3} \\ \frac{\partial P_z}{\partial q_1} & \frac{\partial P_z}{\partial q_2} & \frac{\partial P_z}{\partial q_3} \end{bmatrix} \quad (3.12)$$

unde :  $q1$  și  $q2$  - variabilele unghiulare din cuplele 1 și 2

$W$  - versorul pe care se face mișcarea

$J$  - matricea Jacobian

Mai sus sunt prezentate etape din rezolvarea acestor ecuații. Fiind vorba de un braț articulat cu trei cuple de rotație, dimensiunea matricelor transformărilor omogene și Jacobianului sunt foarte mari, cu termeni lungi, astfel că s-a optat pentru a construi un script în matlab care să rezolve aceste înmulțiri matriceale și calcule cu derivate parțiale. După care s-au preluat aceste date și s-a realizat o analiză complexă cu ajutorul aceluiași software, matlab.

Definind vectorii „T1”, „T2”, „T3” și „T4”, pozițiile relative ale articulațiilor, s-a realizat calculul matricelor de rotație pentru fiecare cuplă și astfel s-a obținut matricea transformărilor omogene ce descrie pozițiile și orientările relative ale cuplelor față de baza fixă a robotului. Mai departe, au fost necesari vectorii de direcție ai cuplelor, denumiți „ ${}^0_1W$ ”, „ ${}^0_2W$ ” și „ ${}^0_3W$ ” care indică



orientarea articulațiilor. În final a rezultat un desen constructiv, figura 3.6, al robotului la care s-au adăugat axele de referință.

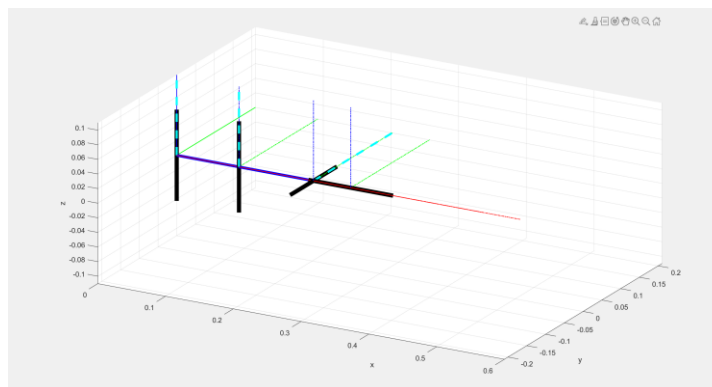


Figura 3.6 Schemă geometrică

Ulterior se generează și o animație, figura 3.7, a traiectoriei efectorului, cu aceleași date de la desen. Pe baza acestei traiectorii putem testa funcționalitatea modelului ales, eventualele erori sau alte probleme care pot apărea. Cu ajutorul acestui script putem foarte ușor testa mai multe variante și vizualiza diferite traiectorii unde analizăm cu precădere limitările configurației alese. Totodată aici am întâmpinat și ideea de volum de lucru a robotului la care trebuie să aliniez și dimensionarea imaginii.

Volumul de lucru al unui manipulator este definit ca zona din spațiu la care efectorul poate avea acces. Acest volum poate fi reprezentat ca un ansamblu de puncte ale căror poziție poate fi calculată iterativ prin discretizarea coordonatelor generalizate. Prin determinarea valorilor posibile ale coordonatelor generalizate, putem obține o aproximare a volumului de lucru al manipulatorului. Această informație este utilă în proiectarea și planificarea mișcărilor manipulatorului, oferind o înțelegere mai clară a spațiului în care acesta poate opera. Volumul de lucru reprezintă pierderea capacității de mișcare a brațului. Dacă „ $\det(J(q)) = 0$ ” atunci putem afla  $q$ -urile pentru frontiera volumului de lucru. [10]

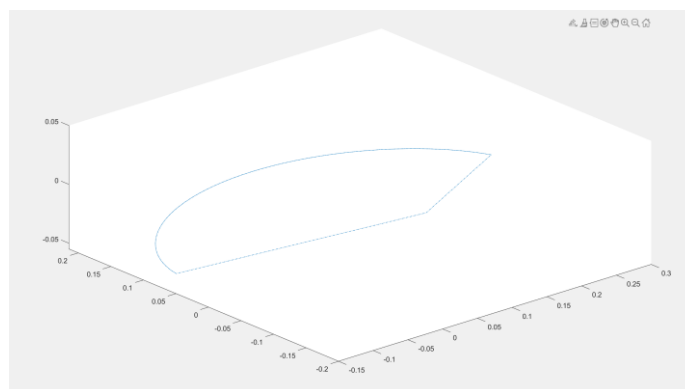


Figura 3.7 Traietorie

### 3.6 MODEL CINEMATIC INVERS

În modelul cinematic invers, cunoaștem poziția și orientarea dorită a efectorului în spațiul de lucru și trebuie să determinăm unghiurile articulațiilor corespunzătoare pentru a atinge acea poziție și orientare.

Obiectivul este de a găsi soluțiile adecvate pentru unghiurile articulațiilor astfel încât brațul robotic să ajungă la poziția și orientarea dorită, având în vedere constrângerile și limitele sistemului. Prin rezolvarea cinematicii inverse, putem planifica și controla mișcările precise ale brațului robotic, permițându-i să îndeplinească sarcinile sale într-un mod eficient și precis.

Implementând un program în limbajul C pe Arduino Mega, s-a realizat comandarea brațului robotic. Acest program citește coordonatele punctului dorit prin intermediul portului serial de la Raspberry Pi și utilizează o funcție specială pentru a controla mișcarea brațului. În interiorul acestei funcții, se realizează calculul lungimii „L” a vectorului de la origine la punctul dorit. Acest calcul este realizat folosind formula din ecuația „3.13”, care determină distanța reală dintre origine și punctul dorit în planul bidimensional. Coordonatele „xe” și „ye” reprezintă valorile punctului dorit în sistemul de coordonate al brațului robotic. [10] [12]

$$L = \sqrt{(xe^2 + ye^2)} \quad (3.13)$$

În continuare, se calculează valorile pentru articulația „q1”. În primul rând, în această etapă se utilizează arcotangenta raportului dintre „ye” și „xe”, pentru a determina înclinația prehensorului față de axa de referință. Apoi, se aplică teorema cosinusului pentru a obține unghiul dintre prima parte a brațului și a doua, prin scăderea rezultatului funcției „3.14”. Pentru a obține valorile în formatul corect, rezultatul este convertit din radiani în grade înmulțind cu  $\pi/180$  și apoi în unități digitale înmulțind din nou cu 200, la fel pentru ambele cuple. [10] [12]

$$q_1 = \text{atan}\left(\frac{ye}{xe}\right) - a \cos\left(\frac{L_2^2 - L^2 - L_1^2}{-2L_1L}\right) \quad (3.14)$$

În etapa următoare, se calculează valorile pentru articulația „q2” folosind formula „3.15”. Această formulă se bazează, de asemenea, pe teorema cosinusului și permite determinarea unghiului dintre a doua parte a brațului și cea de a treia. Similar cu calculele anterioare, rezultatul este convertit în grade și apoi în unități digitale. Pentru a asigura o poziționare și orientare corectă a prehensorului, sunt adăugate compensații de eng.offset la valorile articulațiilor „q1” și „q2”. Aceste offset-uri permit ajustarea fină a poziționării și orientării prehensorului pentru a atinge punctul dorit într-un mod precis. [10] [12]

$$q_2 = \text{acos}\left(\frac{L^2 - L_1^2 - L_2^2}{2L_1L_2}\right) \quad (3.15)$$

Ecuțiile au fost determinate pe baza teoriei cinematicii indirecte, pornind de la Figura 3.8, fiind o schema reprezentativă descrierii de mai sus.

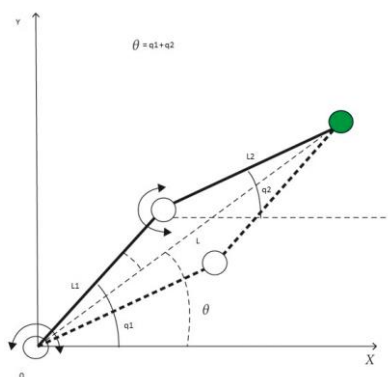


Figura 3.8 Schemă cinematică inversă

Înainte de realizarea fizică a acestui robot, s-a decis că o simulare oferă o modalitate controlată și sigură de a evalua și optimiza comportamentul brațului articulat. Aceasta ne permite să identificăm și să corectăm eventuale erori sau deficiențe în designul sau algoritmul de control al brațului, asigurându-ne că acesta funcționează corect și eficient. Astfel, s-a realizat o simulare utilizând mediul Simulink. În simulare, impunem o traiectorie limită în spațiul articular, definită de un set de unghiuri inițiale și finale pentru fiecare articulație. Durata traiectoriei este setată la „T” secunde, iar intervalul este discretizat în „n” puncte. Au fost calculate traiectoriile dorite pentru fiecare articulație. Aceste traiectorii sunt compuse din valori pentru poziție „Q”, viteză „dQ” și accelerație „ddQ”, pentru fiecare punct în timp. Am utilizat interpolarea polinomială pentru a obține valorile dorite pentru fiecare traiectorie. În simulare, am inclus și afișarea grafică a traiectoriilor, vitezelor și accelerațiilor pentru fiecare articulație. Acest lucru ne permite să vizualizăm evoluția acestor parametri în timp și să observăm cum se mișcă brațul articulat. De asemenea, am salvat traiectoriile în fișiere separate pentru fiecare articulație, sub formă de matrice timp-valoare. Aceste fișiere pot fi utilizate ulterior pentru analize suplimentare sau pentru a fi încorporate în alte aplicații.

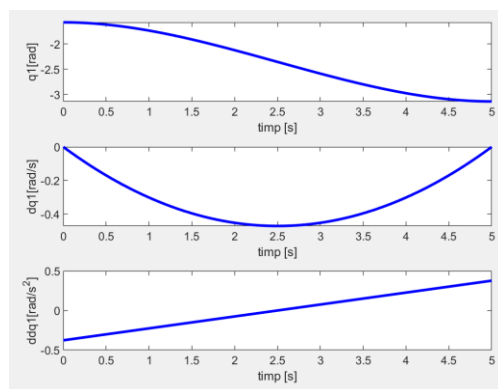


Figura 3.9 Rezultate cupla 1

Imaginea reprezentată în figura 3.9 de mai sus, prezintă cupla unu, împreună cu graficele corespunzătoare ale poziției „ $q_1$ ”, vitezei „ $dq_1$ ” și accelerației „ $ddq_1$ ”. Acest grafic ilustrează variația în timp a acestor trei mărimi. Poziția reprezintă locația sau amplasamentul cuplei în funcție de timp, viteza reprezintă rata de schimbare a poziției în funcție de timp, iar accelerația reprezintă rata de schimbare a vitezei în funcție de timp. Analizând acest grafic, putem observa cum se modifică poziția, viteza și accelerația în timp pentru această cuplă.

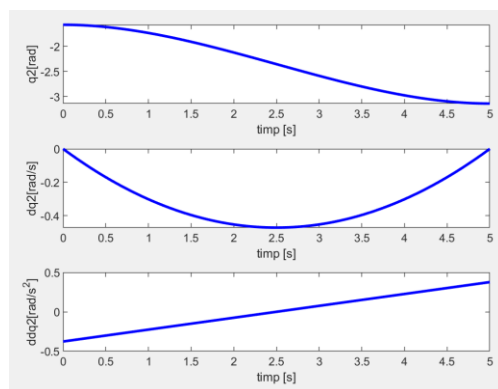


Figura 3.10 Rezultate cupla 2

Următoarea imagine, figura 3.10, prezintă cupla doi, împreună cu graficele sale pentru poziție, viteză și accelerație. Deși valorile mărimilor sunt identice cu cele ale cuplei unu, este important să examinăm graficul pentru a observa cum se comportă cupla doi în comparație cu cupla unu. Graficul poziției, vitezei și accelerației ne permite să analizăm orice diferențe sau similitudini între cele două cuple și să tragem concluzii relevante, iar rezultatul nu este deloc surprinzător folosind exact același motor și aceeași mișcare ca a primei cuple.

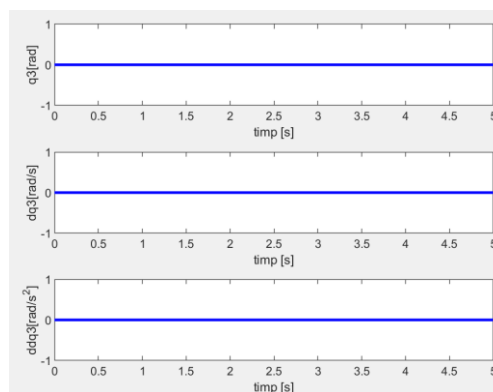


Figura 3.11 Rezultate cupla 3

În ultima imagine, reprezentată în figura 3.11 de mai sus, observăm cupla trei, unde toate mărimile (poziția, viteză și accelerația) au valoarea zero, rezultând faptul că în experiment, cupla trei nu suferă nicio mișcare sau schimbare în timp. Prin examinarea acestui grafic, putem constata că această cuplă trei, rămâne într-o poziție fixă și nu înregistrează nicio schimbare în viteza sau accelerația sa. Această cuplă de rotație reprezintă mișcarea pixului în sus sau în jos, momentul în care robotul atinge foaia de hârtie sau nu. În experimentul simulat în Matlab, s-a ales să nu se ia în considerare acest moment, deoarece nu influențează în vreun fel postura brațului articulat.

Mai departe sunt prezentate imaginile simulării realizate în Matlab Simulink. Putem observa în următoarea imagine brațul robotic în poziția de plecare. Această imagine reprezintă starea inițială a brațului. În această poziție de pornire, cupla unu și cupla doi sunt poziționate într-un mod specific, pregătite pentru o mișcare viitoare. Această poziție de plecare stabilește referința inițială pentru mișcarea ulterioară a brațului. [15]

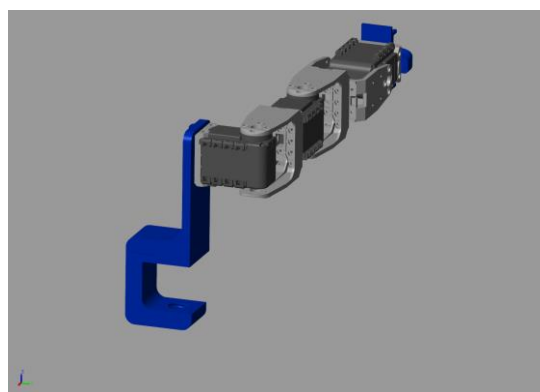
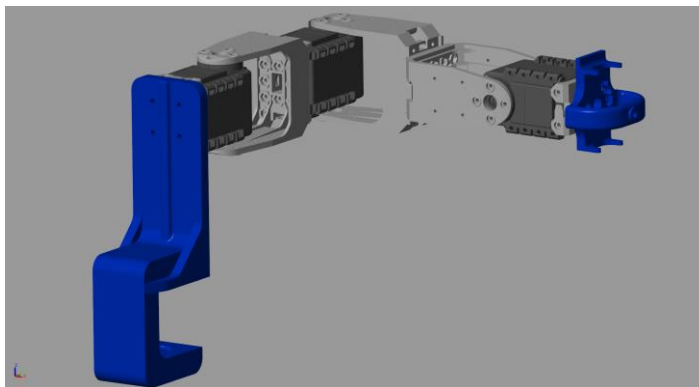


Figura 3.12 Poziție de pornire simulare

În cea de-a doua imagine, putem observa brațul în poziția finală. Această imagine reprezintă rezultatul mișcării brațului, care constă într-o rotație de 90 de grade pentru cupla unu și aceeași rotație și pentru cupla doi. Această simulare a rotației celor două cuple a fost realizată

cu ajutorul funcțiilor din Simulink, unde s-au introdus viteza și accelerația determinată anterior, iar prin mai multe experimente cu diverse poziții s-a analizat eficiența construcției robotului.



*Figura 3.13 Poziție de oprire simulare*

## 4 PROIECTARE ROBOT CARICATURAL

---

Modelarea 3D

Procesul de printare

Analiza prin elementul finit

---

### 4.1 MODELAREA 3D

Pentru a începe proiectarea brațului robotic, s-au furnizat CAD-urile producătorilor, în cazul motoarelor și al șasiului. Aceste CAD-uri au furnizat informații detaliate despre geometria, dimensiunile și specificațiile tehnice ale brațului. Cu aceste informații, realizarea unui model 3D al brațului robotic în Catia, ne-a permis să vizualizăm și să evaluăm designul înainte de construcție. Astfel s-a putut realiza o construcție anterioară a întregului sistem, de la masă, la fiecare șurub din acest proiect, precum se poate analiza în figura următoare 4.1.

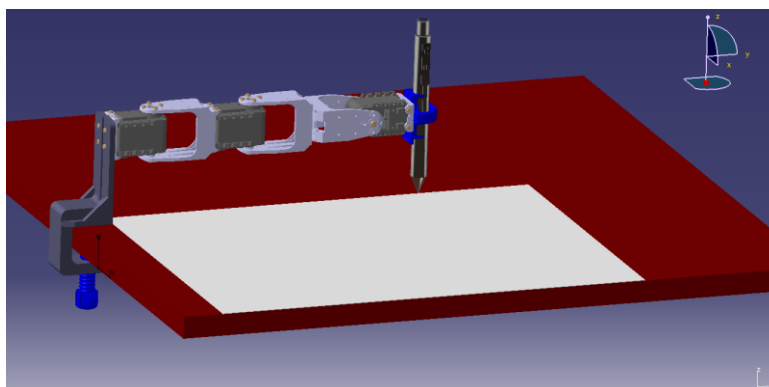


Figura 4.1 Ansamblu CAD

S-a efectuat o analiză a geometriei brațului, a articulațiilor și a limitărilor de mișcare, cu scopul de a obține o înțelegere clară a capacităților și restricțiilor brațului robotic. Această analiză a permis identificarea celei mai eficiente configurații și poziționări a cuplelor, în funcție de necesitatea proiectului.

Odată realizată această asamblare în 3D, a urmat adăugarea unui sistem de prindere pentru robotul de masă și a unui suport de pix atașabil șasiului brațului articulat. Suportul de

masă și suportul de pix au fost integrate în modelul 3D, având grijă ca acestea să se încadreze corespunzător în configurarea brațului. În plus, au fost importate mai multe dimensiuni de instrumente de scris pentru a verifica compatibilitatea acestora cu suportul.

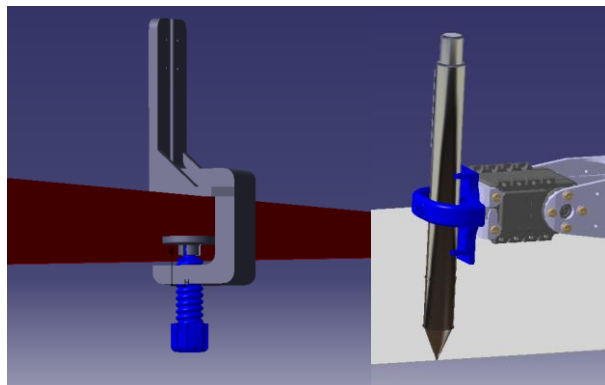


Figura 4.2 Prindere masă și pix CAD

## 4.2 PROCESUL DE PRINTARE

Pentru procesul de imprimare 3D a suportului de fixare, care leagă brațul robotic de masa de lucru și suportul de pix a fost necesară realizarea unui astfel de sistem dedicat, neputând să se găsească ceva în piața de domeniu. Modele au fost realizate în Catia, de unde s-au generat și fișierele „.stl” pentru a le putea pregăti programele de printare.

Pentru piesa de prindere dintre masă și robot, denumită menghină, imaginea din stânga a figurii 4.3, s-a realizat o deschidere maximă de 40 mm, stabilind anterior că media grosimilor maxime ale meselor este de această valoare. În imaginea următoare, sunt prezentate menghina și piesa atașată șurubului, care asigură o priză fixă cu partea inferioară a mesei. Ambele imagini sunt reprezentate tridimensional dintr-un unghi deschis, pentru a putea observa toate detaliile importante.

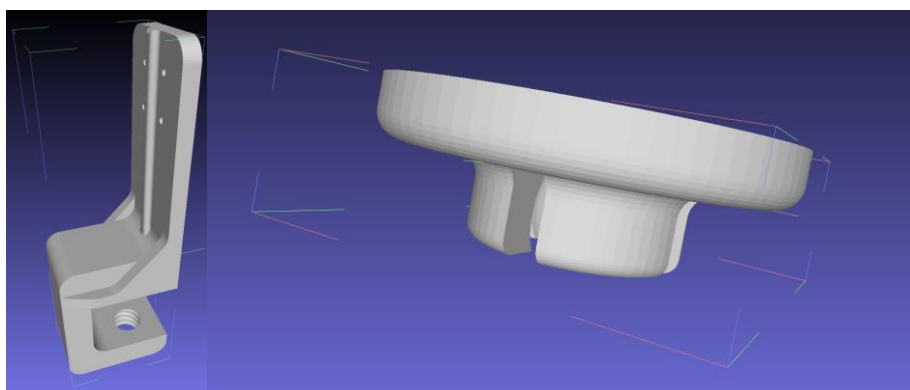
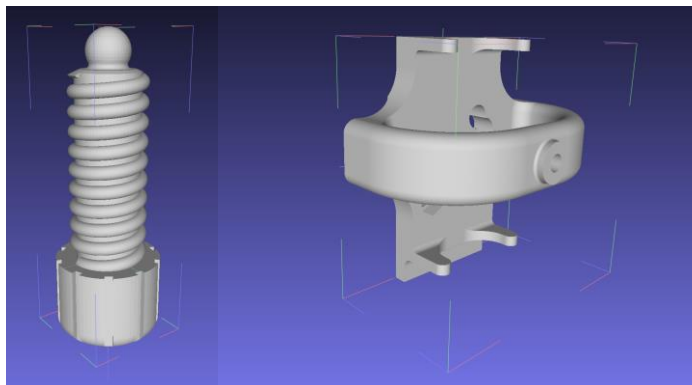


Figura 4.3 Piese printate

Mai departe a fost necesară și realizarea celorlalte piese din ansamblu, la fel de importante ca cele anterioare, una reprezentând șurubul de strângere dedicat menghinei și



cealaltă un suport de pix improvizat, simplu, ce are un filet pentru un șurub de strângere a pixului sau a instrumentului de scris ce i se va atașa. Șurubul de strângere, imaginea din stânga a figurii 4.4, a fost realizat cu funcția „helix” din Catia în concordanță cu filetul interior de la menghină.



*Figura 4.4 Piese printate*

Aceste fișiere „.stl”, Standard Tessellation Language, sunt o reprezentare tridimensională a obiectelor și sunt utilizate în procesul de imprimare 3D. Studiind felul în care se puteau printa aceste piese, s-a ales să se folosească PLA, acid polilactic pentru a se putea obține un rezultat de calitate superioară, astfel s-a căutat să se seteze cat mai optim parametrii pentru imprimare. S-a utilizat o temperatură de 200°C, o viteză redusă de printare de 35 mm/s și o grosime a stratului de 0.12 mm. Un aspect important a fost procentul de eng.infill, adică gradul de umplere al obiectului, care a fost setat la 100% pentru a asigura o duritate ridicată a pieselor. Suportul de fixare a fost realizat folosind tehnologia „FDM” (Fused Deposition Modeling). În acest proces, filamentul de „PLA” a fost încălzit și depus în straturi succesive pentru a crea obiectul dorit. Cum se poate observa și în imaginea următoare, figura 4.5, înainte de printarea piesei, se generează automat o bază utilizând programul în care se execută codul pentru imprimantă. Baza îndeplinește mai multe roluri în asigurarea calității piesei: furnizează un suport stabil, menține un nivel corect pentru un print precis, previne deformările sau desprinderile și facilitează desprinderea finală a piesei fără a o afecta.

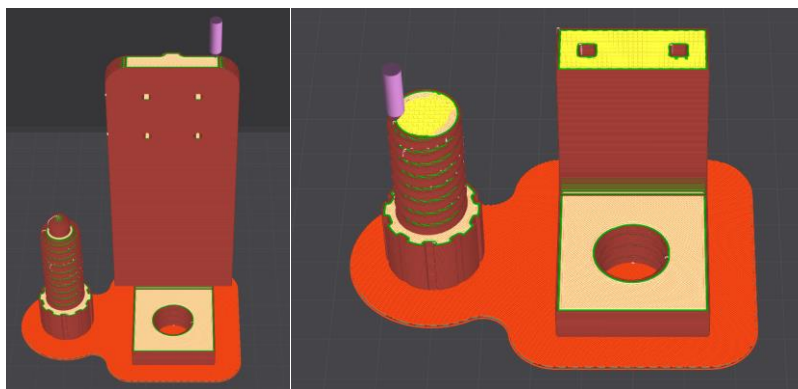


Figura 4.5 Fișiere STL

### 4.3 ANALIZA PRIN ELEMENT FINIT

Analiza elementului finit reprezintă o metodă utilizată pentru evaluarea comportamentului structural al componentelor și asamblărilor prin aplicarea principiilor matematice și a calculului numeric. În cadrul acestui proiect, s-a efectuat o analiză a elementului finit pentru suportul de fixare care conectează brațul robotic la masa de lucru, asemănat cu o menghină. Scopul principal al acestei analize a fost de a evalua comportamentul suportului de fixare sub încărcăturile exercitate de brațul robotic și de a identifica potențialele zone de deformare sau tensiune. Prin utilizarea metodei elementului finit, s-au realizat simulări computerizate care au permis obținerea unor rezultate detaliate privind răspunsul structural al suportului de fixare în diverse condiții de încărcare. Această analiză a oferit informații importante referitoare la rezistența și stabilitatea suportului de fixare, permițând astfel identificarea eventualelor zone vulnerabile sau necesități de îmbunătățire. Rezultatele obținute au contribuit la optimizarea designului suportului de fixare și la asigurarea unei performanțe adecvate în cadrul aplicației robotice. [16] [17]

Pentru realizarea analizei elementului finit, s-a utilizat modelul 3D al suportului de fixare creat în software-ul CATIA. Analiza elementului finit a fost efectuată în același software. Înainte de a iniția analiza, s-au definit condițiile de încărcare și restricțiile corespunzătoare. În cadrul simulării, s-au luat în considerare forțele exercitate de brațul robotic asupra suportului de fixare. Pentru simularea fixării suportului de masă, s-a creat un element rigid în acea zonă, fără a lua în considerare forța cu care se strânge șurubul de masă. Pentru brațul robotic, s-a simulat o forță aplicată în patru puncte de pe partea superioară a suportului. Deoarece rezistența piesei este suficientă pentru a suporta apăsarea robotului, s-a ridicat fictiv greutatea robotului de cinci ori, în scopul de a observa comportamentul suportului în cazul unei încărcări ridicate. De asemenea, s-au setat materialele și proprietățile lor în conformitate cu specificațiile tehnice ale

componentelor. Aceasta a inclus selectarea adecvată a materialelor utilizate în suportul de fixare și brațul robotic, precum și specificarea parametrilor materialului, cum ar fi modulul de elasticitate.

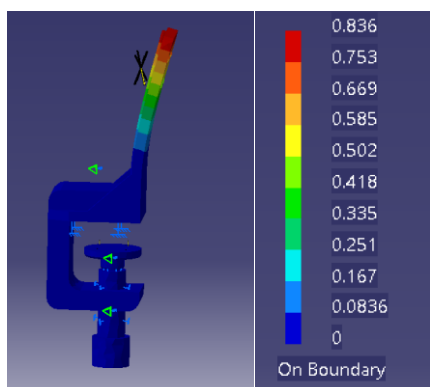


Figura 4.6 Deplasare

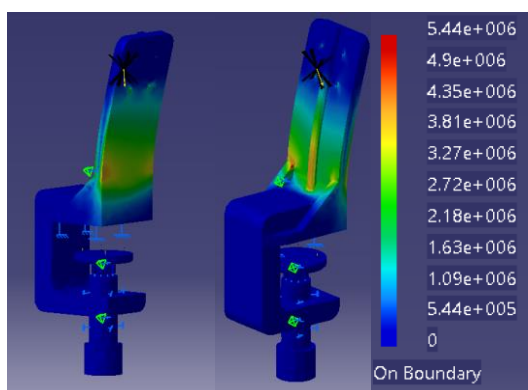


Figura 4.7 Tensiune

În cadrul analizei elementului finit, s-au examinat mai multe aspecte ale comportamentului suportului de fixare, inclusiv deformarea, tensiunea și deplasarea. Aceste măsurători au furnizat informații despre comportamentul structural și rezistența suportului de fixare în condițiile de încărcare impuse. Deformarea, figura 4.6, a fost evaluată pentru a identifica zonele de deformare excesivă, care ar putea afecta funcționalitatea și stabilitatea sistemului. Tensiunea a fost analizată pentru a determina zonele în care suportul de fixare ar putea fi supus stresului maxim. Deplasarea a fost măsurată pentru a evalua stabilitatea și poziționarea suportului de fixare sub încărcăturile specifice. Rezultatele analizei elementului finit ne-au furnizat o înțelegere detaliată a comportamentului suportului, permițând identificarea zonelor critice, figura 4.7, care necesită o atenție specială în cazul modificării structurii brațului articulată. Astfel, pot fi luate măsuri corespunzătoare pentru a îmbunătăți rezistența și performanța suportului de fixare în astfel de situații.

## 5 IMPLEMENTAREA ȘI EVALUAREA SISTEMULUI

---

Echipament utilizat

Procesul de realizare a sistemului robotic

Analiza și evaluarea rezultatelor

---

### 5.1 ECHIPAMENTUL UTILIZAT

În cadrul proiectului, au fost utilizate mai multe componente hardware, printre care se numără motoarele Dynamixel AX12-A, un Raspberry Pi și un Arduino Mega. Motoarele Dynamixel sunt considerate ideale pentru aplicațiile robotice datorită preciziei ridicate și puterii pe care o oferă. Raspberry Pi este o placă de dezvoltare cu o capacitate de procesare suficient de mare pentru a manipula imagini și pentru a transmite comenzi către Arduino Mega. Arduino Mega, la rândul său, este responsabil de mișcarea efectivă a brațului robotic.

Brațul robotului este format din trei cuple de rotație, două pe axa Z și una pe axa Y. Cea de-a treia cuplă de rotație de pe axa Y are rolul de a coborî pixul pentru a desena caricaturile. Acest design permite robotului să se deplaseze cu ușurință și să deseneze punctele sau liniile necesare.

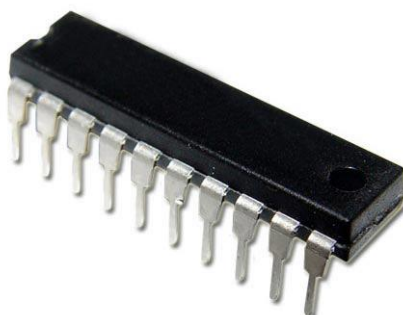
Dynamixel AX-12A este un servo-motor inteligent cu feedback, produs de compania coreeană Robotis. A fost ales pentru acest proiect datorită caracteristicilor sale tehnice și a flexibilității de utilizare. AX-12A poate fi controlat foarte ușor prin intermediul unei plăcuțe de dezvoltare Arduino, în cazul acesta am utilizat un Arduino Mega. Motorul poate fi programat pentru a atinge poziții precise, având o rotație continuă de 360 de grade, ceea ce îl face ideal în multe aplicații, chiar și în acest caz. Un alt avantaj al acestui motor este că include un senzor de poziție și un senzor de temperatură, care permit monitorizarea stării motorului și a mediului înconjurător, senzorul de poziție luând un rol de baza în acționarea unui braț artistic precis. De asemenea, are o gamă largă de viteze, cu o viteză maximă de 59 RPM. Ca orice produs, are și

dezavantajele lui, costul său relativ ridicat, comparativ cu alte servo-motoare și de asemenea, are o limită de sarcină de 1,5 kg/cm, ceea ce poate fi o limitare pentru unele aplicații. [18] [19] [20]



*Figura 5.1 Dynamixel AX12-A*

Mai departe pentru comunicarea dintre Arduino Mega și AX12-A a fost necesar un 74LS245N, care este un circuit integrat octal bidirecțional de tip tranziție de nivel. Acesta este utilizat pentru a realiza conversia semnalelor digitale între dispozitivele care funcționează la tensiuni diferite. Acesta este compus din 2 porturi de 8 biți care permit transferul de date în ambele direcții. Circuitul integrează multiple caracteristici de protecție, cum ar fi diode Schottky la intrările porturilor, care protejează împotriva descărcărilor electrostatice și a supratensiunilor, lucru ce a fost de mare folos în timpul testelor inițiale. S-a ales acest circuit integrat deoarece permite transferul rapid și sigur al datelor între microcontrolerul Arduino Mega și brațele robotice, care funcționează la tensiuni diferite. În plus, 74LS245N este disponibil la un preț accesibil și este larg utilizat în aplicații similare. 74LS245N are 8 canale de intrare cât și de ieșire ce completează transmiterea feedback-ului de la motoare fiind capabil să transmită semnale în ambele direcții. Tehnologic vorbind, putem spune că este depășit dar încă se utilizează în industrii de baza precum automatizări, aeronautica, echipamente medicale etc. [21]



*Figura 5.2 74LS241N*

## 5.2 PROCESUL DE REALIZARE A SISTEMULUI ROBOTIC

Parcursul proiectului s-a bazat pe rezultate etapizate, astfel încât să se asigure un progres constant și să se creeze un mediu organizat. Acest proces de dezvoltare a proiectului a ajutat să se obțină o înțelegere mai clară a funcționării sistemului și să se identifice eventuale probleme și aspecte de îmbunătățit.

În faza inițială a proiectului, accentul s-a pus pe utilizarea plăcuței Arduino și a motoarelor Dynamixel AX12-A. Documentația acestora a fost consultată, iar bibliotecile existente au fost studiate pentru a înțelege funcționalitățile și metodele de programare. S-au efectuat diverse teste de control pentru a învăța cum să poziționăm motoarele la unghiurile necesare, utilizând potențiometre pentru a furniza valori precise pentru fiecare cuplă. Un aspect la fel de important a fost capacitatea de a primi feedback real de la motoare pentru a putea compara cu datele de intrare și a ne asigura că funcționează corect. Această etapă ne-a permis să ne familiarizăm cu motoarele și să obținem controlul asupra lor.

În figura 5.3 se prezintă o schemă de conexiune realizată în Thinkercad, o platformă gratuită care facilitează proiectarea de astfel de circuite. Schema arată modul în care am conectat motoarele împreună cu plăcuța de dezvoltare. Potențiometrul a fost utilizat pentru a roti la o poziție precisă. Plăcuța Arduino Uno folosește un convertor analog-digital cu o rezoluție de 10 biți, ceea ce înseamnă că poate reprezenta  $2^{10} = 1024$  de valori diferite, numerotate de la 0 la 1023. După primul test, am conectat trei astfel de motoare care să imite cele trei cuple de rotație din brațul nostru articulat, fiecare motor fiind asociat cu un potențiomtru corespunzător. În urma acestui experiment, am observat că motoarele au o zonă în care nu pot ajunge. Astfel, am descoperit că fiecare valoare transmisă prin Arduino corespunde unei valori în grade, unde un unghi de 0 grade reprezintă valoarea 0 furnizată de potențiomtru, iar valoarea maximă de 1023 reprezintă un unghi de 300 de grade. Această observație este mai clar ilustrată în figura 5.4.

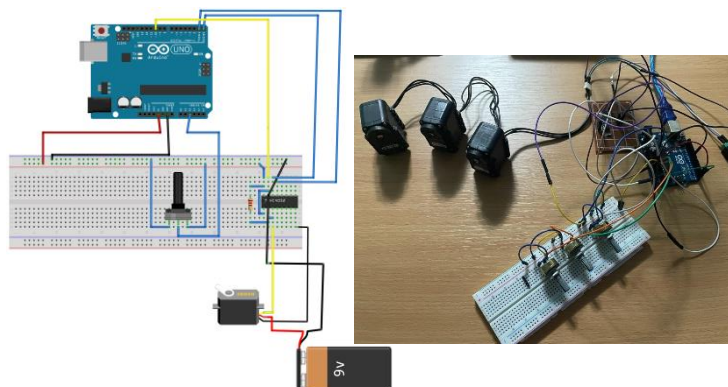


Figura 5.3 Schema conexiuni motor

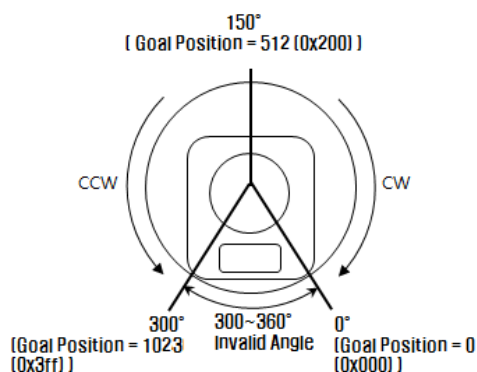


Figura 5.4 Poziții motor [7]

Următorul pas a fost controlul direct al motoarelor utilizând Raspberry Pi și limbajul de programare Python. Cu toate acestea, s-au întâmpinat dificultăți în comunicarea între Raspberry Pi și motoarele Dynamixel AX12-A din cauza lipsei unui cablu dedicat și a unor librării actualizate. Pentru a rezolva această problemă, s-a decis să se utilizeze o metodă de comunicare între Raspberry Pi și Arduino. Această comunicare a fost testată prin transmiterea rezultatelor de pe Raspberry Pi către Arduino și afișarea acestora pe un LCD. Această soluție a permis menținerea etapei de prelucrare a imaginilor pe Raspberry Pi, care oferă suficientă putere de calcul pentru această sarcină, iar coordonatele punctelor au fost transmise către Arduino, care se ocupă de poziționarea brațului la punctul respectiv. În timpul testelor, s-au întâmpinat erori în comunicarea sistemului. Arduino UNO, care a fost utilizat inițial, dispunea de un singur port serial, în timp ce pentru comunicarea între Raspberry Pi, Arduino și motoarele Dynamixel AX12-A era necesară utilizarea a două porturi seriale. Prin urmare, s-a decis să se facă un upgrade la Arduino Mega, care oferă mai multe porturi seriale. Acest upgrade a permis gestionarea cu succes a comunicării între toate componentele sistemului. [22]

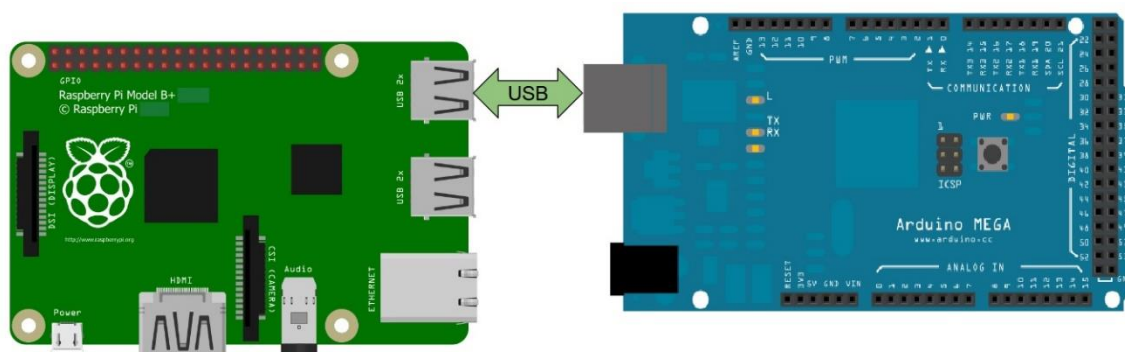


Figura 5.5 Conexiune Raspberry Pi și Arduino Mega

### 5.3 ANALIZA ȘI EVALUAREA REZULTATELOR

Proiectul realizează desene prin intermediul unui braț robotic controlat de un Raspberry Pi și o placă Arduino Mega. În cadrul analizei performanțelor, vom evidenția procesul prin care imaginea este prelucrată și transmisă către brațul robotic pentru a realiza desenele, precum și rezultatele obținute în urma acestui proces.

Imaginile utilizate în proiect sunt prelucrate pe Raspberry Pi în mai multe etape. În primul rând, imaginea este redimensionată pentru a se potrivi cu dimensiunile spațiului de lucru al brațului robotic. Apoi, se aplică operații de eroziune și dilatare pentru a elimina zgomote. Pentru a obține o imagine cu mai puține detalii și a elimina din imperfecțiuni, se aplică filtrul Gaussian, un filtru blur asupra imaginii. În final, imaginea este convertită în nuanțe de gri și se aplică un threshold pentru a obține o imagine binară. Punctele de interes sunt reprezentate de valori de „1”, în timp ce restul punctelor sunt reprezentate de valoarea „0”. Pe baza acestei imagini binare, se extrage o matrice cu dimensiunile corespunzătoare spațiului de lucru al brațului robotic, având nevoie de un dreptunghi egal în care să reprezentăm desenul, dimensiunea maximă rezultată este de 120 pe 170 mm. Fiecare valoare de „1” din matrice reprezintă un punct ce trebuie pus pe hârtie. Se calculează coordonatele fiecărui astfel de punct și se transmite o pereche de coordonate (x, y) prin intermediul serial către Arduino Mega. Pe Arduino Mega, folosind o funcție de calculare a poziției prin cinematică inversă, brațul articulat se deplasează la coordonatele primite pentru a atinge punctul respectiv. La atingerea acelei coordonate, cea de a treia cuplă, cu rotație pe axa y coboară pixul pentru a marca coordonata. Coordonatele fiecărui punct atins de brațul robotic sunt salvate într-un fișier text, care poate fi comparat cu afișajul LCD. Afișajul LCD arată fiecare punct atins de brațul robotic, permițând o verificare vizuală a rezultatelor obținute. Rezultatele obținute în urma procesului de desenare au fost în concordanță cu imaginile de referință și am putut observa o corelație între coordonatele transmise și punctele desenate de brațul robotic.

Astfel, performanțele obținute în proiectul final au demonstrat că sistemul nostru robotic este capabil să realizeze desene pe baza informațiilor extrase din imagini și să atingă punctele dorite în spațiul de lucru.

Mai departe, figura 5.6 prezintă o schemă bloc a conexiunilor din sistemului, ilustrând modul în care componentele sunt interconectate. Aceasta include plăcuțele de dezvoltare, motorul care reprezintă brațul articulat, deoarece celelalte motoare doar comunică între ele, LCD-ul ce afișează coordonatele la care trebuie să lase un punct pe hârtie, circuitul integrat ce



face posibilă comunicarea între Arduino și Dynamixel ax12-a și desigur se mai observă și un potențiometrul cu care doar reglez intensitatea afișajului de pe LCD. Schema bloc este utilă pentru a vizualiza structura și felul în care a fost realizată conexiunea elementelor.

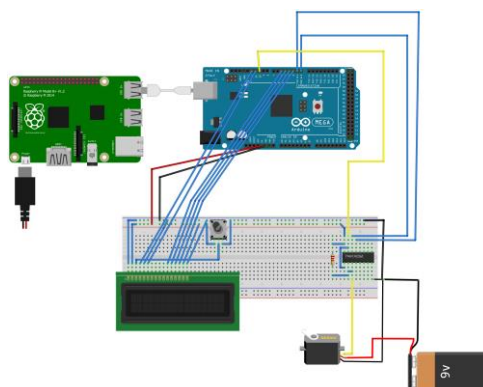


Figura 5.6 Conexiuni robot

În figura 5.7 se pot observa două portrete caricaturale realizate cu aceeași imagine inițială dar cu parametri diferiți. Prima imagine prezintă doar puncte lăsate de pix, care au fost generate în urma procesării inițiale a imaginii de intrare. În schimb, în partea dreaptă a imaginii, în apropierea acestor puncte, se observă prezența unor mici linii formate din 3-4 puncte. Această diferență subtilă în rezultatul final a fost generată prin modificarea anumitor parametri în etapa de procesare a imaginilor. Rezultatele acestui studiu evidențiază importanța alegerii și ajustării corespunzătoare a parametrilor de procesare a imaginilor în cadrul procesului de creare a portretelor caricaturale de către un robot. Chiar și mici modificări ale acestor parametri pot conduce la rezultate vizuale diferite, cum se poate observa în imaginea prezentată. Aceste descoperiri pot fi utile în dezvoltarea și optimizarea viitoarelor sisteme de generare automată a portretelor caricaturale. Pentru aprofundarea acestui subiect, se poate considera extinderea experimentului prin testarea și ajustarea altor parametri de procesare a imaginilor. Astfel de cercetări pot contribui la înțelegerea mai profundă a interacțiunii dintre roboți și arte vizuale, precum și la dezvoltarea sistemelor de inteligență artificială cu abilități artistice.

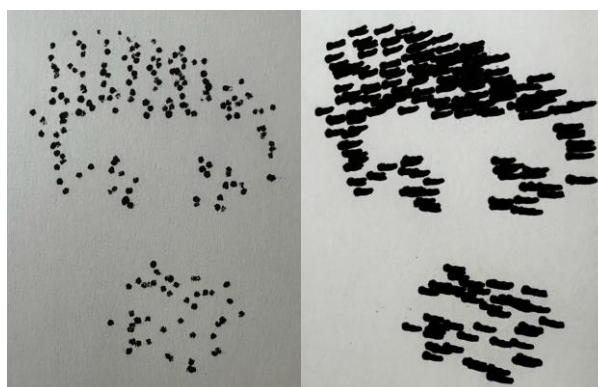
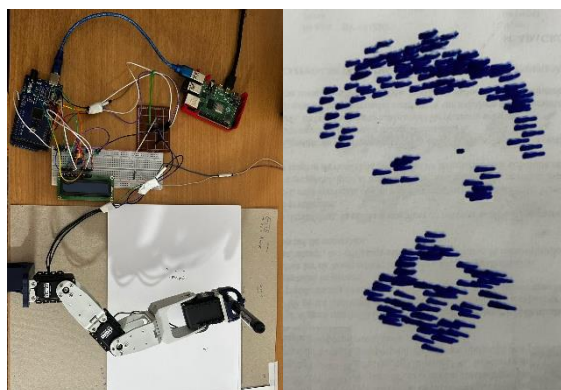


Figura 5.7 Portrete inițiale

În finalul lucrării, în figura 5.8, se poate observa o imagine ce cuprinde sistemul robotic în acțiune, iar în partea dreaptă unul dintre primele rezultate al unui portret caricatural realizat în procent complet de un braț robotic. Observăm că sunt mici diferențe între imaginea rezultată în urma procesării artificiale, dar asta face ca portretul să fie unul caricatural. Diferențele apar din cauza neconcordanței între viața reală, unde perturbațiile apar din toate direcțiile și cazurile analizate într-un mediu bine stabilit virtual și analitic.



*Figura 5.7 Portret caricatural*

## 6 CONCLUZII ȘI PERSPECTIVE VIITOARE

---

### Concluzii

### Perspective viitoare

---

### 6.1 CONCLUZII

În cadrul acestui proiect, s-a realizat un sistem robotic capabil să realizeze desene pe baza imaginilor primite. S-au utilizat o combinație de tehnologii, incluzând Arduino Mega, Raspberry Pi și motoare Dynamixel AX12-A, pentru a controla mișcarea brațului robotic și pentru a prelucra imaginile.

Proiectarea și implementarea acestui sistemului a implicat mai multe etape cheie. În primul rând, efectuarea unei analize a cerințelor și alegerea componentelor hardware potrivite. Astfel, s-au integrat motoarele Dynamixel AX12-A în structura brațului articulat și s-a realizat conexiunea cu Arduino Mega pentru a controla mișcarea acestuia. Mai departe a trebuit să alegem un sistem de comunicare între Raspberry Pi și Arduino Mega, astfel încât să se poată transmite coordonatele punctelor de interes din imagine către brațul robotic. Au fost utilizați algoritmi de prelucrare a imaginilor pentru a identifica și extrage punctele relevante dintr-o imagine de referință. Apoi, s-au efectuat calculele necesare pentru a determina poziția și orientarea brațului robotic în funcție de aceste coordonate. Algoritmii de prelucrare a imaginilor au fost implementați pentru a detecta și recunoaște punctele de interes, precum marcaje sau obiecte specifice. Ulterior, coordonatele acestor puncte au fost utilizate în calcule matematice și algoritmi de cinematică pentru a determina poziția brațului și unghiurile sale de rotație corespunzătoare. Aceste informații au fost folosite pentru a controla mișcarea brațului robotic și pentru a-l poziționa în mod precis în funcție de cerințele și indicațiile date.

Concluziile proiectului nu ar fi complete fără menționarea rolului important pe care modelarea matematică și simulările în Matlab și Simulink l-au jucat în dezvoltarea și optimizarea acestui sistem robotic.

În prima etapă, s-a realizat un model geometric utilizând Matlab. Acest model a inclus dimensiunile și geometria componentelor, precum și relațiile și constrângerile lor în spațiu. Informațiile obținute au fost utilizate pentru a defini cinematica sistemului, stabilind relația dintre poziția și orientarea brațului robotic și valorile unghiurilor motoarelor.

Modelul cinematic ne-a permis să calculăm precis unghiurile necesare pentru a poziționa brațul robotic în funcție de coordonatele primite de la Raspberry Pi. Acest lucru a facilitat deplasarea precisă a brațului robotic la punctele de interes în timpul procesului de desenare. Pentru a evalua și valida performanța proiectului, am realizat simulări. Modelând comportamentul brațului robotic, mișcările motoarelor și interacțiunea cu mediul de lucru și analizând parametrii cum ar fi viteza și accelerația brațului, precum și forțele și momentele exercitate asupra componentelor, s-au identificat potențialele probleme și constrângeri ale sistemului pentru a putea fi corectate în stadiile incipiente ale proiectului.

Prin integrarea modelului geometric și cinematic în Matlab și Simulink, s-a obținut o mai bună înțelegere al acestui sistem robotic și să optimizăm performanțele. S-a demonstrat că utilizarea acestor unelte matematice și de simulare sunt esențiale în dezvoltarea sistemelor robotice avansate și a permis să atingerea rezultatelor dorite în ceea ce privește precizia și performanța brațului robotic în procesul de desenare.

Prin analiza elementului finit, s-a evaluat comportamentul piesei de fixare a brațului robotic sub solicitările generate de mișcările acestuia. S-au identificat zonele de deformare și s-au obținut informații despre stresul și deplasarea produsă în structură.

În concluzie, proiectul a demonstrat viabilitatea unui sistem robotic eficient pentru desenarea caricaturilor, integrând tehnologii hardware și software. În urma planificării și testării riguroase a etapelor cheie, am obținut rezultatele dorite. Această experiență ne-a adus multe învățăminte și oferă o bază solidă pentru dezvoltarea și aplicarea ulterioară a tehnologiilor de robotică și prelucrare a imaginilor.

## **6.2 PERSPECTIVE VIITOARE**

Perspectivile viitoare ale proiectului sunt promițătoare și ne oferă oportunități interesante de dezvoltare ulterioară. În continuare voi prezenta ce îmbunătățiri propun să implementez.

În prima fază, propun un algoritm de identificare și delimitare a zonelor unde exista mai multe valori de „1” în matricea binară. Acest lucru ne va permite să recunoaștem și să separăm obiectele individuale prezente pe suprafața de lucru. Pasul acesta se poate realiza prin utilizarea unor algoritmi de segmentare și analiză a contururilor. Scopul fiind de a putea trece în cea de a doua fază unde vreau să folosesc o funcție de creare a liniilor. Liniile să se formeze aleatoriu între punctele cu valoarea „1” dar în interiorul obiectelor pentru a nu se trage o linie între păr și barbă de exemplu. Această etapă este posibil să se realizeze tot cu o funcție din Open CV, denumită „houghline”. Această abordare va recrea o mișcare similară cu a unui creion de artist, adăugând un element de spontaneitate și expresivitate în desene.

Iar ulterior, într-un viitor mai îndepărtat, se pot explora și implementa alte tehnici artistice în procesul de desenare. Acest lucru ar putea include utilizarea de pensule virtuale cu diferite forme și texturi, aplicarea de umbre și lumini pentru a obține efecte de volum și profunzime, precum și experimentarea cu diferite stiluri artistice, cum ar fi stilul impresionist sau abstract. Din acest punct deja necesitățile proiectului cresc și avem nevoie de mult mai multe îmbunătățiri, ca de exemplu pentru a permite utilizatorului să monitorizeze progresul desenului în timp real și să ofere feedback vizual, putem să integrăm un sistem de afișare pe un ecran în timp real cu rezultatul sistemului robotic. Acest lucru ne va permite să vizualizăm desenul în timp ce este în curs de realizare și să facem ajustări sau corecții dacă este necesar. Folosind o camera deasupra spațiului de lucru care înregistrează actul artistic al robotului.

Aceste perspective viitoare vor permite îmbunătățirea și inovarea sistemul robotic de desen, adăugând noi elemente artistice și funcționalități avansate. Îmi propun să explorez aceste direcții pentru a crea desene captivante și expresive într-un mod unic și inovator.

## 7 BIBLIOGRAFIE

- [1] T. W., W. Q. Toh, H. Zhang, X. S., S. L, Y. Liu și W. Jing, „RoboCoDraw: Robotic Avatar Drawing with,” în *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, Singapore, 2020.
- [2] M. Adamik, J. Goga, J. Pavlovicova, A. Babinec și I. Sekaj, „Fast robotic pencil drawing based on image evolution by means of,” *Robotics and Autonomous Systems*, nr. 148, p. 11, 2022.
- [3] R. Ai-Da, „About Ai-Da Robot,” 2019. [Interactiv]. Available: <https://www.ai-darobot.com/about>. [Accesat Noiembrie 2022].
- [4] J. M. Gülzow, P. Paetzold și O. Deussen, „Recent Developments Regarding Painting Robots for Research în Automatic Painting, Artificial Creativity, and Machine Learning,” Konstanz, 2020.
- [5] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2010.
- [6] R. Gonzalez și R. Woods, *Digital Image Processing, USA: Prentice Hall*, 2008.
- [7] OpenCV, „OpenCV-Open Computer Vision,” 2022. [Interactiv]. Available: <https://opencv.org/>. [Accesat Aprilie 2023].
- [8] G. Bradski și A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly Media, 2008.
- [9] B. Siciliano, L. Sciavicco, L. Villani și G. Oriolo, *Robotics: Modelling, Planning and Control*, Springer, 2010.
- [10] C. Pozna, *Curs Sisteme de Conducere în Robotica*, Universitatea Transilvania, 2021.
- [11] C. Pozna, *Curs Bazele Cinematicii Roboților Industriali*, Universitatea Transilvania, 2021.
- [12] C. Pozna, *Modelarea Roboților cu Post Fix, Matrix Rom*, 2015.
- [13] Python, „Data Structures,” 2021. [Interactiv]. Available: <https://docs.python.org/3/tutorial/datastructures.html#more-on-lists>. [Accesat Martie 2023].
- [14] G. Măceșanu, T. Cociaș și S. Grigorescu, *Sisteme de vedere artificială*, 2013.

[15] MathWorks, „Simulink Documentation,” 2022. [Interactiv]. Available: <https://www.mathworks.com/help/simulink/index.html>. [Accesat Martie 2023].

[16] G. Mogan și S. Butnariu, *Analiza cu elemente finite. Aplicații în CATIA*, Universitatea Transilvania, 2007.

[17] S. Butnariu, *Analysys of mechanical structures using finite element method*, Universitatea Transilvania, 2013.

[18] Robotis, „AX-12A,” 2022. [Interactiv]. Available: <https://emanual.robotis.com/docs/en/dxl/ax/ax-12a/>. [Accesat Decembrie 2022].

[19] R. Pi, „Raspberry Pi 4 Model B specifications,” 2022. [Interactiv]. Available: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>. [Accesat Ianuarie 2023].

[20] Arduino.CC, „Mega 2560 Rev3,” 2022. [Interactiv]. Available: <https://docs.arduino.cc/hardware/mega-2560>. [Accesat Martie 2023].

[21] T. Instruments, „SNx4LS245 Octal Bus,” Septembrie 2016. [Interactiv]. Available: <https://www.ti.com/lit/ds/symlink/sn74ls245.pdf>. [Accesat Decembrie 2022].

[22] Arduino.CC, „Serial-Arduino Reference,” 2021. [Interactiv]. Available: <https://www.arduino.cc/reference/en/language/functions/communication/serial/>. [Accesat Februarie 2023].

## REZUMAT

---

Proiectul descrie construcția unui sistem mecatronic, folosind componentele Raspberry Pi și Arduino Mega, cu scopul de a dezvolta un robot capabil să realizeze portrete caricaturale prin intermediul vederii artificiale și controlului cinematic invers al brațului articulat. Sistemul implementat prelucrează o imagine folosind limbajul Python și funcții din biblioteca Open CV, apoi transmite informațiile necesare brațului robotic programat în limbajul C.

În cadrul procesului de prelucrare a imaginii, se obține o matrice binară care reprezintă imaginea de intrare. Coordonatele elementelor de valoare „1” din această matrice sunt extrase și transmise prin portul serial către Arduino Mega. Cu ajutorul principiilor de cinematică inversă, Arduino Mega determină postura brațului articulat necesară pentru a atinge aceste coordonate. Cu ajutorul unui pix montat la capătul brațului, este creat un punct în locul fiecărui element cu valoarea „1” din matrice, urmând indicațiile primite de la Raspberry Pi. În cadrul acestui proiect, s-au abordat subiecte de vedere artificială, modelarea matematică, proiectare și analiza a componentelor.

Această lucrare cuprinde toate informațiile esențiale pentru înțelegerea bazelor teoretice ale proiectului, precum și detalii amănunțite despre procesul de realizare și testare etapizată a robotului. Fiecare etapă a fost documentată în detaliu, oferind o înțelegere clară a modului în care a fost dezvoltat acest robot. Testele efectuate la fiecare etapă au confirmat funcționalitatea și eficiența sistemului implementat.

În concluzie, această lucrare aduce o contribuție semnificativă în domeniul mecatronicii și roboților manipulatori, deschizând perspective pentru viitorii roboți artiști. Proiectul oferă o soluție completă și eficientă pentru controlul unui braț robotic prin integrarea vederii artificiale și a controlului cinematic invers. Implementarea acestui proiect are un potențial promițător în diverse domenii de cercetare și de artă, subliniind impactul semnificativ al tehnologiei roboților în progresul societății.



## ABSTRACT

---

The project discusses the development of a mechatronic system combining Raspberry Pi and Arduino Mega components to create a robot capable of making caricature portraits using computer vision and inverse kinematic control of the articulated arm. The implemented system processes an image with the Python language and OpenCV library functions before transmitting the essential information to the C-programmed robot arm.

A binary matrix representing the input image is obtained throughout the image processing process. The coordinates of the "1" values in this matrix are retrieved and sent to the Arduino Mega through the serial port. The Arduino Mega estimates the required posture of the articulated arm to achieve these coordinates using inverse kinematics concepts. Following the Raspberry Pi's instructions, a point is formed in place of each element with the value "1" in the matrix using a pen attached to the end of the arm. Computer vision, mathematical modeling, component design and analysis are all covered in this project.

This document contains all the necessary material for understanding the project's theoretical basis, as well as thorough information about the process of creating and testing the robot in stages. Each stage has been meticulously recorded, allowing for a clear understanding of how this robot was created. The testing carried out at each level confirmed the system's functioning and efficiency.

Finally, this work makes an important addition to the field of mechatronics and manipulator robotics, paving the way for future artist robots. By combining computer vision and inverse kinematic control, the research provides a comprehensive and efficient approach for directing a robotic arm. This project's implementation offers remarkable potential in a variety of research and artistic disciplines, demonstrating the huge effect of robotic technology on societal growth.

**DECLARAȚIE PRIVIND ORIGINALITATEA  
LUCRĂRII DE LICENȚĂ / PROIECTULUI DE DIPLOMĂ / DISERTAȚIEI**

UNIVERSITATEA TRANSILVANIA DIN BRAȘOV  
FACULTATEA INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR  
PROGRAMUL DE STUDII ROBOTICĂ

NUMELE ȘI PRENUMELE ONCIOIU ADRIAN DUMITRU

PROMOȚIA 2019-2023

SESIUNEA VARĂ 2023

TEMA LUCRĂRII / PROIECTULUI/ DISERTAȚIEI

ROBOT PENTRU DESEN CARICATURAL

.....  
CONDUCĂTOR ȘTIINȚIFIC CONF. DR. ING. COCIAȘ TIBERIU TEODOR

Declar pe propria răspundere că lucrarea de față este rezultatul muncii proprii, pe baza cercetărilor proprii și pe baza informațiilor obținute din surse care au fost citate și indicate conform normelor etice, în textul lucrării/proiectului, în note și în bibliografie.

Declar că nu s-a folosit în mod tacit sau ilegal munca altora și că nici o parte din teză/proiect nu încalcă drepturile de proprietate intelectuală ale altcuiva, persoană fizică sau juridică.

Declar că lucrarea/ proiectul nu a mai fost prezentat(ă) sub această formă vreunei instituții de învățământ superior în vederea obținerii unui grad sau titlu științific ori didactic.

În cazul constatării ulterioare a unor declarații false, voi suporta rigurile legii.

Data: 22.06.2023

Absolvent

ONCIOIU ADRIAN DUMITRU

