

Міністерство освіти і науки України
Державний вищий навчальний заклад
"Національний гірничий університет"

Електротехнічний
(факультет)

Кафедра Електропривода
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
дипломної роботи
магістра

(назва освітньо-кваліфікаційного рівня)

галузь знань 14 Електрична інженерія
(шифр і назва галузі знань)

спеціальність 141 Електроенергетика, електротехніка та електромеханіка
(код і назва спеціальності)

(освітня програма «Електромеханічні системи автоматизації та електропривод»)

освітній рівень магістр
(назва освітнього рівня)

Кваліфікація 2151.2 (Int) Інженер-електромеханік
(код і назва кваліфікації)

на тему: «Система керування колаборативним роботом маніпулятором на основі машинного зору»

Виконавець:

Студент 6 курсу, групи 141м-16-4

_____ Лелько В.М.
(підпис) (прізвище та ініціали)

Керівники	Прізвище, ініціали	Оцінка	Підпис
проекту	Худолій С.С.		
розділів:			
Аналітичний розділ	Худолій С.С.		
Математичний опис	Худолій С.С.		
Моделювання	Худолій С.С.		
Адаптація до роботи з людиною	Худолій С.С.		

Рецензент			
-----------	--	--	--

Нормоконтроль	Казачковський М.М.		
---------------	--------------------	--	--

Міністерство освіти і науки України
Державний вищий навчальний заклад
"Національний гірничий університет"

ЗАТВЕРДЖЕНО:

завідувач кафедри
Електропривода
(повна назва)

_____ Казачковський М.М.
(підпис) (прізвище, ініціали)

« _____ » _____ 2017 року

ЗАВДАННЯ
на виконання кваліфікаційної роботи магістра
спеціальності 141 Електроенергетика, електротехніка та електромеханіка
(код і назва спеціальності)
(освітня програма «Електромеханічні системи автоматизації та електропривод»)
студенту 141м-16-4 Лелько В.М.
(група) (прізвище та ініціали)

Тема дипломної роботи «Система керування колаборативним роботом маніпулятором на основі машинного зору»

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ Ректора ДВНЗ "НГУ" від 31.10.2017 № 1806-л

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень - система машинного зору для формування траєкторії колаборативного робота маніпулятора.

Предмет досліджень - дослідити можливість використання системи машинного зору на основі неспеціалізованого апаратного забезпечення, для формування траєкторії колаборативного робота маніпулятора.

Мета НДР - створити систему здатну ефективно координувати рухи робота маніпулятора в залежності від даних отриманих через систему машинного зору.

Вихідні дані для проведення роботи: переддипломна практика та літературні джерела _____

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна полягає в:

- Можливості спростити конструкцію колаборативного робота маніпулятора та підвищити його доступність за рахунок перекладання частини задач з спеціалізованих систем на загальнодоступні, такі як ПК з веб камерою.
- Використанні загально доступних засобів (бібліотек машинного зору) для розробки системи керування роботом маніпулятором.

Практична цінність полягає в:

- Зниженні складності системи керування для завдання траєкторії, що повинно посприяти популяризації подібного роду систем.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Результати роботи повинні відповідати предметній області спеціальності «Електромеханічні системи автоматизації та електропривод», мати наукову новизну та практичну цінність, бути обґрунтованими та достовірними.

Результати досліджень мають бути поданими у вигляді, що дозволяє їх безпосереднє використання для удосконалення систем електропривода та автоматизації.

5. ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Підготовка обладнання для проведення дослідів та випробування системи керування. Аналіз літературних джерел.	01.09.2017-17.10.2017
Складання кінематичної схеми, аналіз методів розрахунку інверсної кінематики та її розрахунок	17.10.2017- 02.12.2017
Аналіз способів реалізації системи машинного зору. Розробка програмного забезпечення.	02.12.2017- 19.01.2018

6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект від реалізації результатів роботи очікується позитивним завдяки можливості спрощення конструкції колаборативного робота маніпулятора.

Соціальний ефект від реалізації результатів роботи очікується позитивним завдяки можливості популяризації подібного роду систем.

7. ДОДАТКОВІ ВИМОГИ

Відповідність оформлення ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення.

Завдання видав

(підпис)

Худолій С.С.

(прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Лелько В.М.

(прізвище, ініціали)

Дата видачі завдання: 01.09.2017

Термін подання дипломної роботи до ЕК: 17.01.2018

РЕФЕРАТ

Пояснювальна записка 90 с., 39 рис., 8 табл., 2 додатки, 14 джерел.

Об'єкт детальної розробки: робот маніпулятор DoArm з 6 ступенями свободи, керування яким здійснюється за допомогою плати Arduino під'єднаної до ПК.

Мета роботи: розробка системи керування роботом маніпулятором на основі машинного зору та адаптація робота до роботи в кооперації з людиною.

Виконано розрахунок інверсної кінематики. Розглянуто методи розпізнавання об'єктів та обрано оптимальний. Розроблена програма в пакеті MS Visual Studio на мові програмування C#, що дозволяє задавати траєкторію для робота за допомогою веб камери. Розроблена програма для плати Arduino, яка формує керуючі імпульси для сервоприводів, на основі отриманих через СОМ-порт даних.

Розроблені заходи з охорони праці під час розробки даного проекту.

КОЛАБОРАЦІЙНИЙ РОБОТ МАНІПУЛЯТОР, СИСТЕМА МАШИННОГО
ЗОРУ, ARDUINO.

					ЕП.МР.18.09.Р.ПЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

РЕФЕРАТ

Пояснительная записка содержит 90 с., 39 рис., 8 табл., 2 приложения, 14 источников.

Объект детальной разработки: робот манипулятор DoArm с 6 степенями свободы управляемый при помощи платы Arduino подсоединённой к ПК.

Цель работы: разработка системы управления роботом манипулятором на основе машинного зрения и адаптация работа к работе в кооперации с человеком.

Выполнен расчет инверсной кинематики. Рассмотрены методы распознавания объектов и выбран оптимальный. Разработанная программа в пакете MS Visual Studio на языке программирования C #, которая позволяет задавать траекторию для робота с помощью веб-камеры. Разработана программа для платы Arduino, которая формирует управляющие импульсы для сервоприводов, на основе полученных через СОМ-порт данных.

Разработаны мероприятия по охране труда при разработке данного проекта.

КОЛЛАБОРАТИВНЫЙ РОБОТ МАНИПУЛЯТОР, СИСТЕМА МАШИННОГО
ЗРЕНИЯ, ARDUINO.

					ЕП.МР.18.09.Р.ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

ABSTRACT

The explanatory note includes 90 p., 39 fig., 8 tables, 2 applications, 14 sources.

The object of detailed design: the robot manipulator DoArm with 6 degrees of freedom controlled by Arduino Uno board which is connected to the PC.

The purpose of the work: the development of a robot-arm control system based on the machine vision system and adaptation of the robot to work in cooperation with human.

The invers kinematic calculations are done. The methods of object recognition are examined and the best was selected. The programs on the programming language C# are developed in the package MS Visual Studio. It allows specifying the trajectory for the robot arm using a webcam. Moreover the program for Arduino board which generates and transmits control signals which are based on data received from PC through Com-port are developed.

The labor protection measures are developed.

COLLABORATIVE ROBOTIC ARM, MACHINE VISION SYSTEM, ARDUINO

					<i>ЕП.МР.18.09.Р.ПЗ</i>	<i>Арк.</i>
						5
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ЗМІСТ

ВСТУП.....	8
1. ОГЛЯД ТЕХНІЧНИХ РІШЕНЬ ТА АНАЛІЗ МЕТОДИК РОЗРАХУНКУ В ОБЛАСТІ МЕХАТРОННИХ ПРИСТРОЇВ ТА СИСТЕМ ВІЗУАЛІЗАЦІЇ, ФОРМУЛЮВАННЯ ЗАДАЧ ДОСЛІДЖЕНЬ.....	10
1.1. Загальна характеристика колаборативних мехатронних систем.....	11
1.2. Перспективи розвитку мехатронних пристроїв	15
1.3. Аналіз методики розрахунку кінематики	17
1.4. Огляд методів розпізнавання об'єктів в кадрі.....	20
1.4.1. Огляд інструментальних засобів використаних для розробки програмного забезпечення	23
1.5. Висновки та постановка задач досліджень	25
2. МАТЕМАТИЧНИЙ ОПИС РУХУ ЛАНОК МАНІПУЛЯТОРА ТА ОПИС АЛГОРИТМУ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ В КАДРІ.....	26
2.1. Опис робота-маніпулятора. Метод Денавіта-Хартенберга.....	27
2.2. Розрахунок інверсної кінематики.....	36
2.3. Алгоритм розпізнавання об'єктів.....	45
3. МОДЕЛЮВАННЯ РОБОТИ РОЗРОБЛЕНОЇ СИСТЕМИ.....	53
3.1. Моделювання роботи алгоритму інверсної кінематики.....	56
3.2. Програмна реалізація системи розпізнавання об'єктів.....	64
4. АДАПТАЦІЯ РОБОТА ДО РОБОТИ В КОЛАБОРАЦІЇ З ЛЮДИНОЮ...	70
4.1. Механічні небезпеки	71
4.1.1. Система машинного зору як основа безпеки при роботі з маніпулятором.	72
4.1.2. Використання датчиків струму для визначення зіткнення з перешкодою	77
4.2. Небезпека враження людини електричним струмом при роботі з маніпулятором та ПК.....	81
4.3. Висновки.....	82

					ЕП.МР.18.09.3.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВИСНОВКИ.....	83
ПЕРЕЛІК ПОСИЛАНЬ	84
Додаток А.....	86
Додаток Б	88

					<i>ЕП.МР.18.09.3.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Дипломна робота присвячена актуальному на сьогоднішній день питанню, пов'язаному з використанням колаборативних роботів для задач дрібної автоматизації. Представлені далі рішення покликані спростити взаємозв'язок людини та робота, а також на практичному прикладі показати доступність технологій автоматизації не тільки для великих промислових підприємств, але й для звичайних людей.

В основній частині дипломної роботи розглянуто сучасний ринок роботів маніпуляторів, виділені тенденції характерні для розвитку цього виду мехатронних систем. Розглянуто основні задачі та принципи при створення подібних систем. Проаналізована можливість використання розробленої в дипломному проекті системи для опрацювання прикладних задач. Сформовані вимоги до створюваної мехатронної системи, проведено вибір та обґрунтування методів розробки. Описано розрахунок кінематики, алгоритм створення системи машинного зору. Розроблено програмне забезпечення для керування роботом маніпулятором. Проведено моделювання роботи розробленої системи. Впроваджено ряд заходів направлених на запобігання отримання пошкоджень при роботі людини в кооперації з мехатронною системою. Досліджено можливість популяризації наведеної системи, та проаналізовано її переваги та недоліки.

Мета роботи: розробити алгоритми і програмні рішення, що дозволять керувати роботом маніпулятором за допомогою машинного зору та адаптувати робота до роботи в кооперації з людиною.

Предмет дослідження : параметри процесу та програмна реалізація системи формування траєкторії на основі машинного зору.

Об'єкт дослідження : мехатронна система на базі робота маніпулятора DoArm та система візуалізації.

Наукові задачі:

- Розробити кінематичну структуру та розглянути методи розрахунку інверсної кінематики;

					ЕП.МР.18.09.ВС.ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

- Розглянути можливі способи візуального контролю, та обґрунтувати вибір.
- Розробити алгоритм візуального контролю.
- Реалізувати систему керування на основі розглянутих алгоритмів.
- Проаналізувати можливість впровадження систем контролю безпеки людини при її роботі біля робота або в кооперації з ним та реалізувати вибрані варіанти.

Методи дослідження: в ході дипломного проекту було використано аналітичні, а саме математичні методи та експериментальні методи дослідження, серед яких моделювання та тестування.

Основні результати: досліджено концептуальні засади побудови мехатронного комплексу з системою машинного зору, сформовано вимоги та обґрунтовано технологію розробки системи, реалізовано розглянуті алгоритми в вигляді програмного комплексу з інтерфейсом користувача.

					<i>ЕП.МР.18.09.ВС.ПЗ</i>	<i>Арк.</i>
						9
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

**1. ОГЛЯД ТЕХНІЧНИХ РІШЕНЬ ТА АНАЛІЗ МЕТОДИК
РОЗРАХУНКУ В ОБЛАСТІ МЕХАТРОННИХ ПРИСТРОЇВ ТА
СИСТЕМ ВІЗУАЛІЗАЦІЇ, ФОРМУЛЮВАННЯ ЗАДАЧ
ДОСЛІДЖЕНЬ**

					<i>ЕП.МР.18.09.01.ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Лелько В.М.</i>			<i>ОГЛЯД ТЕХНІЧНИХ РІШЕНЬ ТА АНАЛІЗ МЕТОДИК РОЗРАХУНКУ В ОБЛАСТІ МЕХАТРОННИХ ПРИСТРОЇВ ТА СИСТЕМ ВІ- ЗУАЛІЗАЦІЇ, ФОРМУЛЮВАННЯ ЗАДАЧ ДОСЛІДЖЕНЬ</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>		<i>Худолій С.С.</i>					<i>10</i>	<i>94</i>
<i>Реценз.</i>						<i>НГУ</i>		
<i>Н. Контр.</i>		<i>Казачковський</i>						
<i>Затверд.</i>								

1.1. Загальна характеристика колаборативних мехатронних систем

Сучасний етап науково-технічної революції характеризується комплексною автоматизацією виробництва. За останні десятиліття автоматизація основних технологічних операцій досягла такого рівня, що допоміжні операції, пов'язані з транспортуванням та складуванням, розвантаженням, завантаженням і сортуванням, фарбуванням, зварюванням, виконуються за допомогою спеціалізованих роботів або ж роботів маніпуляторів.

Перші маніпулятори називалися телеоператорами. Такі пристрої дозволяли оператору проводити на безпечній відстані такі роботи, як переміщення радіоактивних речовин. Перший такий апарат був сконструйований в 1947 році. А вже в 1954 році Джордж Девольт і Джожеф Енгельберг запатентували програмований пристрій для перенесення предметів або, маніпулятор під назвою «Unimate» (Рис.1.1)

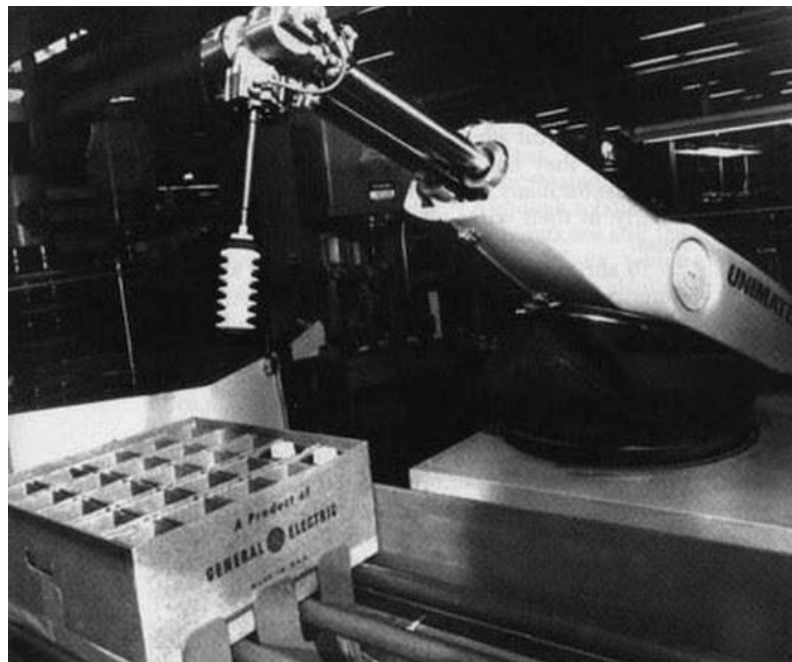


Рис.1.1. - маніпулятор під назвою «Unimate».

Далі за довгі сорок років з 1960 по 2000 роки було створено безліч промислових роботів, які працюють за захисним огороженням. Вони набули широкого визнання в автомобільній промисловості і галузях обробної промисловості. Хоч вони і мають ряд недоліків:

					ЕП.МР.18.09.01.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

- Ризик заподіяти шкоду життю і здоров'ю людини, при знаходженні останнього в робочій зоні робото-технічного комплексу (РТК).
- Вузька спеціалізація промислових РТК.
- Складність технологічної та організаційної інтеграції РТК в існуючу інфраструктуру підприємства.
- Трудомісткість процесу переорієнтації РТК на виконання нових завдань.
- Постійна потреба в послугах висококваліфікованого фахівця для обслуговування і програмування РТК.
- Потреба в ізоляції робочої зони РТК від персоналу.

Логічним розвитком в галузі робототехніки було б усунення перерахованих проблем. Що привело до заснування в 2005 році компанії Universal Robotics. Її засновники дослідники з університету Південної Данії після аналізу існуючих автоматизованих рішень виробили свій погляд на подальший розвиток робототехніки. Їх метою було розробити гнучкий, зручний і легкий робот. І вже в 2008 році був створений перший в світі колаборативний робот UR5 (Рис.1.2). Він міг безпечно працювати поруч з людьми без захисної огорожі. Після цих подій колаборативна робототехніка стала окремим класом, який включає в себе нове покоління індустриальних і сервісних роботів.

Ключові ознаки колаборативного РТК:

- Безпека, що забезпечується можливістю розпізнавання перешкод в робочій зоні РТК і попередження зіткнень або реалізації безпечного «м'якого» контакту з ними.
- Дружність інтерфейсів і інтуїтивні методи програмування РТК, орієнтовані на користувачів, що не володіють спеціальними знаннями в цій області
- Компактність і висока рухливість виконавчих пристроїв РТК.

Великі виробники роботів, такі як Kuka, Abb, Fanuc також почали розробку і випуск цих роботів.

					ЕП.МР.18.09.01.ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис.1.2. - колаборативний робот UR5.

На сьогоднішній день колаборативні роботи доволі прості в програмуванні. Їх навчання може виконуватись за допомогою ручного управління (переміщення), при цьому контролер робота автоматично реєструються задані траєкторії маніпулятора або за допомогою спеціального терміналу, як показано на рис. 1.3.



Рис. 1.3 – Способи формування траєкторії для колаборативного робота UR5

					ЕП.МР.18.09.01.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

В даному дипломному проекті пропонується альтернативний, не менше простий, варіант завдання траєкторії руху для роботів маніпуляторів заснований на машинному зорі. Що у разі успіху дозволить без проблем програмувати маніпулятори які не мають датчиків зворотного зв'язку по положенню.

					<i>ЕП.МР.18.09.01.ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		14

1.2. Перспективи розвитку мехатронних пристроїв

Колаборативні роботи складають новий сектор динамічного ринку промислової робототехніки. Технологічні досягнення і доступна ціна зробили ці пристрої реальністю для багатьох виробничих підприємств.

В останні роки помітний тренд на зростання різноманітності застосування промислових колаборативних маніпуляторів. Як приклад такого нетипового для початку століття використання роботів можна привести: 3D-друк, авіація (пілотування), археологія (допомога археологу в очищенні знахідок), прасування тканини, деревообробка, приготування їжі, медицина, наукові дослідження.

Згідно з міжнародною дослідницькою і консалтинговою компанією «IDC» до 2019 року до 30% розгорнутих роботів будуть відноситися до категорії "колаборативних" (кооперативних). До цього часу такі роботи навчаться працювати втричі швидше сучасних моделей, зберігши безпеку функціонування в безпосередній близькості від людей.

Згідно з даними для всієї галузі роботизованих систем в цілому (Рис1.4.) триватиме безсумнівне зростання попиту на такого роду системи автоматизації, причому не тільки для використання в промисловості але і для персонального використання.

\$ 11,5 млрд - прогноз обсягу ринку колаборативних роботів від аналітиків Barclays на 2025 рік у 2016 році[14].

					ЕП.МР.18.09.01.ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис.1.4. - Оцінка ринку робототехніки консалтинговою компанією Tractica

На сьогодні середня вартість колаборативного робота становить приблизно 24 тисячі доларів, і за прогнозами до 2020 вона може знизитися до 10 тисяч доларів.

Як висновок роботи, які можуть працювати поруч з людьми, перевернуть світ промислових роботів і автоматизації, ставши доступними багатьом малим і середнім компаніям.

1.3. Аналіз методики розрахунку кінематики

Предметом кінематики маніпулятора є аналітичний опис геометрії руху маніпулятора щодо деякої заданої абсолютної системи координат без урахування сил і моментів, що породжують цей рух. Таким чином, завданням кінематики є аналітичний опис просторового розташування маніпулятора залежно від часу, і, зокрема, установлення зв'язку між значеннями приєднаних координат маніпулятора й положенням і орієнтацією його захвату в декартовому просторі.

Механічний маніпулятор можна розглядати як розімкнутий ланцюг, що складається з кількох твердих тіл (ланок), послідовно з'єднаних обертальними або поступальними зчленуваннями, що приводяться в рух силовими приводами. Один кінець цього ланцюга з'єднаний з основою, а інший кінець вільний і з'єднаний з робочим інструментом, що дає змогу впливати на об'єкти маніпулювання або виконувати різні технологічні, наприклад збиральні, операції. Відносний рух зчленувань передається ланкам, у результаті чого захват маніпулятора займає в просторі задане положення.

Кінематика маніпулятора вивчає геометрію руху маніпулятора щодо заданої абсолютної системи координат, не розглядаючи сили й моменти, що породжують цей рух. Таким чином, її предметом є опис просторового положення маніпулятора як функції часу, і, зокрема, співвідношення між простором приєднаних змінних маніпулятора — узагальненими координатами, положенням і орієнтацією схвата. Існує два основні завдання кінематики маніпулятора.

1. Для конкретного маніпулятора за відомим вектором приєднаних кутів — узагальнених координат $q(t) = (q_1(t), \dots, q_n(t))^T$ та заданими геометричними параметрами ланок (n — кількість ступенів свободи) визначити положення й орієнтацію схвата маніпулятора щодо абсолютної системи координат.

2. За відомими геометричними параметрами ланок знайти всі можливі вектори приєднаних змінних маніпулятора, що забезпечують задані положення й орієнтацію захвату щодо абсолютної системи координат.

					<i>ЕП.МР.18.09.01.ПЗ</i>	Арк.
						17
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Перше із цих завдань прийнято називати прямою, а друге — зворотною задачею кінематики маніпулятора. Оскільки власними незалежними змінними маніпулятора є приєднані змінні, а завдання, як правило, формується в координатах абсолютної системи відліку, зворотнє завдання кінематики виникає частіше, ніж пряме.

Денавіт і Хартенберг розробили єдиний загальний підхід до опису просторової геометрії маніпулятора щодо заданої абсолютної системи координат, заснований на застосуванні матричної алгебри. Вони запропонували використовувати для опису взаємного просторового розташування ланок матриці однорідних перетворень, що мають розмірність 4×4 . Отже, пряме завдання кінематики звелось до визначення матриці однорідного перетворення, що задає зв'язок між системою координат схвата й абсолютною системою координат. Матрицями однорідних перетворень зручно користуватися й при виводі рівнянь динаміки руху маніпулятора. Зворотнє завдання кінематики може бути вирішено різними методами. До числа найбільш розповсюджених відносять методи матричної алгебри, метод ітерацій і геометричний підхід[3].

До переваг матричної алгебри та геометричного підходу, які разом відносяться до аналітичного методу розв'язання оберненої задачі кінематики відносять отримання довільної точності рішення. Однак, знаходження точного рішення, у вигляді аналітичної залежності для узагальнених координат від конструктивних параметрів і заданого вектору положення маніпулятора, представляється можливим не для всіх маніпуляторів. Аналітичне рішення, таким чином, існує тільки для роботів з певною конструкцією.

До недоліків аналітичного рішення відносяться складність отримання узагальнених координат в явному вигляді і невизначеність, пов'язану з кінематичною неоднозначністю і використовуваними тригонометричними функціями. Однак, основним недоліком, все ж, є неможливість отримання рішення для довільних конструкцій маніпуляторів.

У разі, якщо рішення оберненої задачі за допомогою матричної алгебри неможливе, використовуються ітераційні методи.

					<i>ЕП.МР.18.09.01.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

Ітераційний метод дозволяє вирішити завдання інверсної кінематики для тих конструкцій маніпуляторів, які не мають точного рішення в аналітичних виразах. Це основна перевага при використанні ітераційного методу.

Однак час збіжності того чи іншого методу заздалегідь невідомо і багато в чому залежить від початкового наближення.

					<i>ЕП.МР.18.09.01.ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		19

1.4. Огляд методів розпізнавання об'єктів в кадрі

На сьогоднішній день системи машинного зору користуються значним попитом в робототехніці. Вони надають прості способи реалізації навчання апаратів автономним діям в природному світі.

Технологія автоматизованого розпізнавання і подальшого відстеження образів на зображенні в реальному часі, як підрозділ області комп'ютерного зору має зараз великий потенціал. Це пов'язано з наступними факторами:

- Існує безліч наукових робіт, присвячених даній тематиці, які створюють непогану базу, для освоєння даної області інженерами і розробниками програмного забезпечення.
- За останні час були досягнуті значні результати в практичному використанні методів розпізнавання образів. На даний момент створені та активно підтримуються розробниками різні бібліотеки і набори інструментальних засобів, що полегшують застосування цих методів при вирішенні різних актуальних завдань.
- Розвиток апаратних засобів, а саме зростання обчислювальної потужності переносної електроніки і поширеність систем хмарних обчислень, робить можливим використання програмних продуктів, заснованих на алгоритмах комп'ютерного зору, причому не тільки в сфері наукової і інженерної діяльності, а й у побутовій споживчій сфері.

Внаслідок цих факторів різноманітність практичних завдань дозволяє відкривати нові горизонти для застосування такого роду технологій, а також вимагає комплексних програмних продуктів, що полегшують рішення цих задач на призначеному для користувача рівні.

В рамках дипломного проекту буде розглянута задача, мета якої - розробка програмного комплексу для виділення та захоплення в реальному часі об'єкта на зображенні і подальшого спостереження за переміщенням обраного об'єкта.

					ЕП.МР.18.09.01.ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

Основним завданням, що виникає при відстеженні об'єктів в відео потоці, є їх автоматичне знаходження на кожному окремо взятому кадрі. Для вирішення цієї задачі можна використовувати такі методи:

- Сегментація зображення - це процес поділу цифрового зображення на декілька множин пікселів, або присвоєння кожному пікселю таких міток, щоб пікселі з однаковими мітками мали спільні візуальні характеристики [7]. Мета сегментації полягає в спрощенні уявлення зображення, щоб його було легше аналізувати. В результаті сегментації зазвичай виділяють кордони та об'єкти на зображеннях. Пікселі, що належать до одних і тих же сегментів, схожі за деякою характеристикою (наприклад, за кольором, яскравості), яку можна вирахувати, а сусідні елементи істотно відрізняються по ній.
- Пошук за шаблоном - ідея пошуку за шаблоном полягає в порівнянні шаблону, на якому зображений шуканий об'єкт, з областями оброблюваного зображення.
- Детектування характерних ознак - процес пошуку заснований на порівнянні характерних ознак (до найпростіших ознак належать кути, грані) оброблюваного кадру і шаблону, на якому зображений шуканий об'єкт.
- Категоріальне розпізнавання - рішення основне на класифікації об'єктів (визначення набору ознак або дескрипторів і машинного навчання класифікатора) [8].

Далі представлений порівняльний огляд різних методів розпізнавання об'єктів в кадрі.

					<i>ЕП.МР.18.09.01.ПЗ</i>	Арк.
						21
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Таблиця 1.1. Порівняльні характеристики методів розпізнавання образів

Назва класу методів	Переваги	Недоліки
Сегментація зображень	- простота в використанні - швидке навчання - стійка робота в разі масштабування і зсуву	- висока чутливість до змін освітлення сцени
Пошук за шаблоном	- висока точність при статичних об'єктах однакової форми	- нестійка робота при масштабуванні, зсуві, повороту зображень
Детектування характерних ознак	- інваріантність до малих поворотів, і до зміни освітлення сцени	- складність розпізнавання об'єктів що динамічно видозмінюються
Категоріальне розпізнавання	- висока швидкість пошуку об'єктів, найкраща серед розглянутих підходів якість розпізнавання	- тривалий час навчання, потреба у великій кількості зображень шуканого об'єкта для формування навчальної вибірки - складність в налаштуванні і перенавчанні

Виходячи з вищезазначеної інформації, для реалізації поставленого завдання пропонується використовувати метод сегментації зображення як найбільш оптимальний в плані співвідношення простоти і функціональності.

- обробка стереозображень.

Варто додати невелике пояснення, так як в подальшому замість назви OpenCV буде використовуватися назва EmguCV.

EmguCV - це бібліотека, заснована на OpenCV і має той же функціонал, але розрахована на роботу з .NET сумісними мовами програмування, такими як C #, VB, VC ++, IronPython і т.д. [6].

					<i>ЕП.МР.18.09.01.ПЗ</i>	Арк.
						24
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

1.5. Висновки та постановка задач досліджень

Ринок колаборативної робототехніки ще тільки формується, та починає швидко рости. Переваги використання колаборативних роботів - підвищена гнучкість, легкість в програмуванні, підвищена ефективність та суттєве зниження операційних витрат. Їх мета - допомогти людям виконувати частину їх обов'язків і дати можливість компаніям залишатися конкурентоспроможними на світовому ринку. І застосування систем машинного зору в даній галузі робототехніки в даному контексті дипломної роботи тільки підвищить привабливість таких систем. Запропоноване в даному дипломному проекті рішення направлене на популяризацію технологій машинного зору для вирішення завдань формування траєкторії для роботів маніпуляторів.

Виходячи з вищевикладеного, в даній дипломній роботі нам потрібно:

1. Виконати розрахунок інверсної кінематики для заданої моделі робота.
2. Розробити алгоритм розпізнавання об'єктів в відео кадрі.
3. Реалізувати розроблений алгоритм та розрахунок кінематики в якості програмного забезпечення для формування траєкторії.
4. Реалізувати в розробленій програмі алгоритм захисту людини при роботі з роботом.

					<i>ЕП.МР.18.09.01.ПЗ</i>	Арк.
						25
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

2. МАТЕМАТИЧНИЙ ОПИС РУХУ ЛАНОК МАНІПУЛЯТОРА ТА ОПИС АЛГОРИТМУ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ В КАДРІ

					<i>ЕП.МР.18.09.02.ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Лелько В.М.</i>			МАТЕМАТИЧНИЙ ОПИС РУХУ ЛАНОК МАНІПУЛЯТОРА ТА ОПИС АЛГОРИТМУ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ В КАДРІ	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Худолій С.С.</i>				26	94	
<i>Реценз.</i>		<i>П.І.Б.</i>				<i>НГУ</i>		
<i>Н. Контр.</i>		<i>Казачковський</i>						
<i>Затверд.</i>		<i>П.І.Б.</i>						

2.1. Опис робота-маніпулятора. Метод Денавіта-Хартенберга

З точки зору кінематики, робот маніпулятор може розглядатися як незамкнений ланцюг твердих ланок, з'єднаних послідовно за допомогою призматичних (поступальних), або шарнірних (обертальних) зчленувань. Один кінець цього ланцюга з'єднаний з основою маніпулятора, інший її кінець, як правило, оснащений пристроєм захоплення. Здатність здійснювати будь-які рухи досягається завдяки тому, що маніпулятор володіє декількома ступенями свободи[5].

Число ступенів свободи - це число можливих переміщень системи. Розташування та орієнтація пристрою захоплення залежить від спільної дії обертання і/або перенесення кожного зчленування ланцюга ланок. У даному дипломному проекті розглядається робот-маніпулятор DoArm фірми "DOIT" з шістьма обертальними ступенями свободи. За реалізацію переміщення ланок робота вздовж своїх осей відповідальні 6 цифрових сервоприводів DS3218m і 1 відповідає за захоплення. Характеристики сервоприводів наведені в таблиці 2.1.

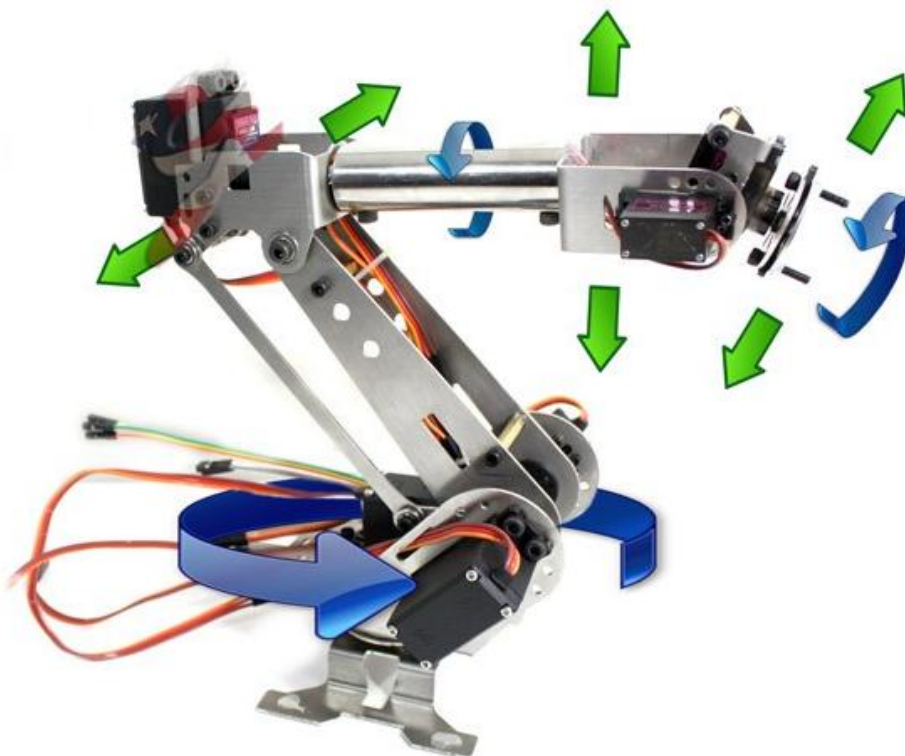


Рис. 2.1 - Робот-маніпулятор DoArm

										Арк.
										27
Змн.	Арк.	№ докум.	Підпис	Дата						

ЕП.МР.18.09.02.ПЗ

Таблиця 2.1. - Характеристика сервоприводів

Сервоприводів DS3218m	Основні характеристики	
	Діапазон робочих температур	-10°C ~ 50°C
	Робоча напруга	5V ~ 7.2 V
	Робоча швидкість (без навантаження)	0.14 сек/ 60°
	Робочій струм (без навантаження)	80mA
	Пусковий момент	15kg.cm
	Пусковий струм	1.3A
	Струм холостого ходу	4mA
	Параметри керування	
	Робоча частота	50-330 Гц
	Кут керування	180° (від 500 до 2500 мкс)
	Нейтральна позиція	1500 мкс
	Мертва зона з	3 мкс
	Напрямок обертання	проти годинникової стрілки (від 1500 до 2000 мкс)

Управління маніпулятором передбачає переміщення захоплення в певну точку з певною орієнтацією. Це переміщення здійснюється за рахунок внесення узагальнених координат. Узагальненими координатами є вимірювані кутові переміщення між двома сусідніми ланками робота. З кожною ланкою робота пов'язана локальна, декартова система координат (Рис 2.2).

Перша базова система координат, як правило, розміщується біля основи робота (біля стійки). Всі інші початкові точки систем координат поміщаються в рухливі суглоби, початок останньої шостої системи координат розміщується на перетині прямих, які зв'язують лапки робота з продовженням останньої ланки. Таким чином, початок знаходиться в середині між лапками захоплення робота.

					ЕП.МР.18.09.02.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

Всього виходить шість локальних систем координат і система координат стійки.

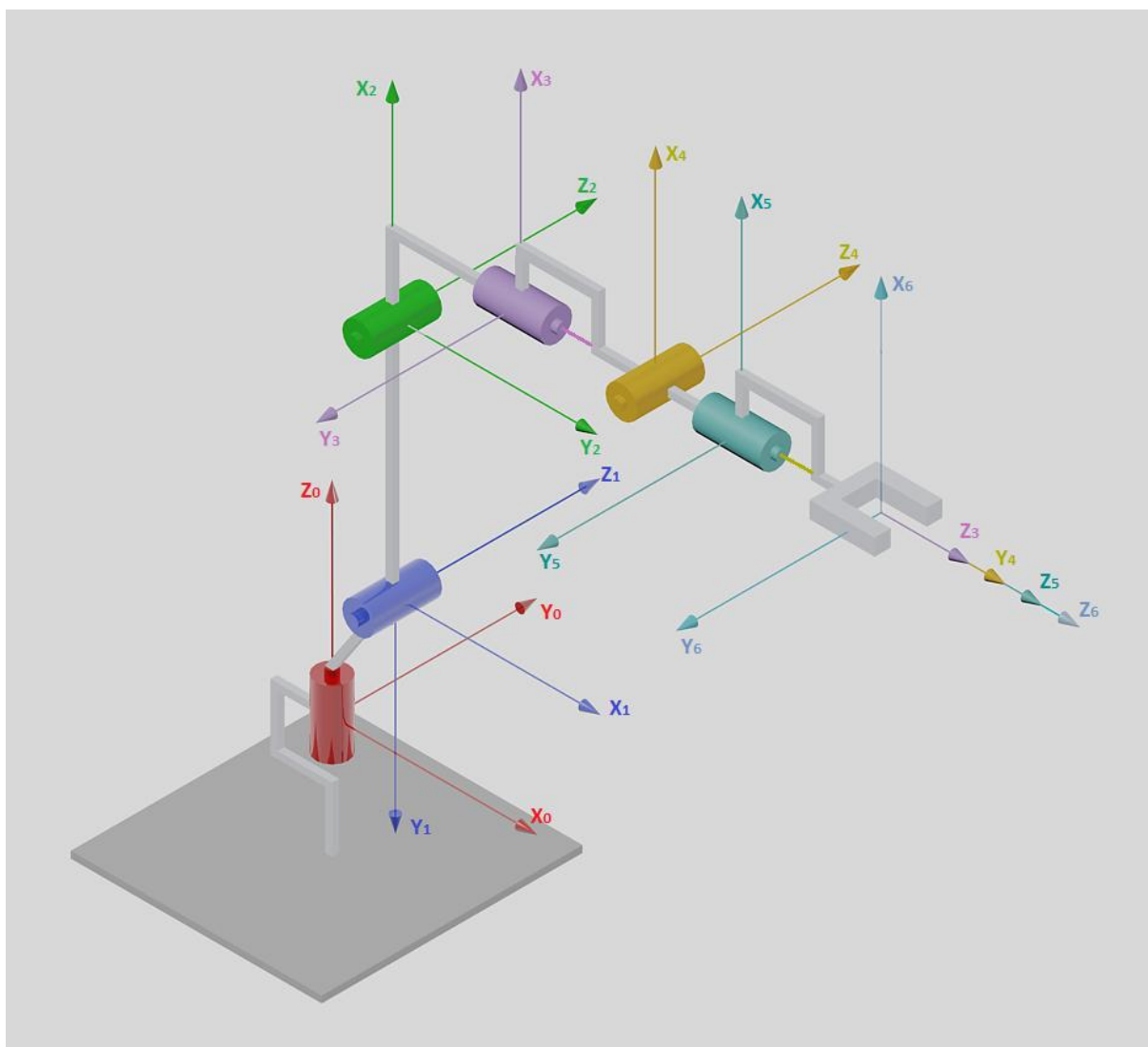


Рис 2.2. - Система локальних координат робота-маніпулятора

Кожна система координат формується на основі трьох правил:

1. Вісь z_{i-1} направлена уздовж вісі i -го з'єднання.
2. Вісь x_i перпендикулярна вісі z_{i-1} и направлена від неї.
3. Вісь y_i доповнює вісь x_i, z_i так щоб система координат була правобічна.

Сенс введення локальних систем координат, полягає в тому, що поворот ланки зручно виразити через поворот локальної системи координат щодо базової.

Після введення локальних систем координат, як способу опису повороту ланок відносно системи координат стійки, виникає три завдання:

										Арк.
										29
Змн.	Арк.	№ докум.	Підпис	Дата						

- 1) Отримати позицію кінцевого ланки знаючи лише узагальнені координати.
- 2) За розрахованим положенням схвата, отримати значення узагальнених координат, за допомогою яких можна перемістити його в потрібну позицію.
- 3) Визначити яким чином перехід від однієї системи локальних координат до іншої буде відповідати геометрії ланки.

Перша і друга задача є прямою та інверсною задачею кінематики.

Для вирішення цих завдань, з метою опису обертальних і поступальних зв'язків між сусідніми ланками Денавіт і Хартенберг запропонували матричний метод побудови систем координат, пов'язаний з кожною ланкою кінематичного ланцюга. Принцип уявлення Денавіта-Хартенберга полягає у формуванні однорідної матриці перетворення, що має розмірність 4x4 і описує стан системи координат кожної ланки щодо системи координат попередньої. Це дає можливість послідовно перетворити координати схвата маніпулятора з системи відліку, пов'язаної з останньою ланкою, в базову систему відліку [4].

Так як керування полягає у формуванні кута між двома сусідніми ланками - зміна цього кута призводить до зміни положення однієї локальної системи координат відносно іншої, що призводить до необхідності описувати кожну наступну локальну систему щодо попередньої. Для опису зміни положення і орієнтації однієї локальної системи координат щодо попередньої використовуються однорідні матриці перетворення (А-матриці, ці матриці також називають перетворенням Денавіта-Хартенберга, рис 2.3.).

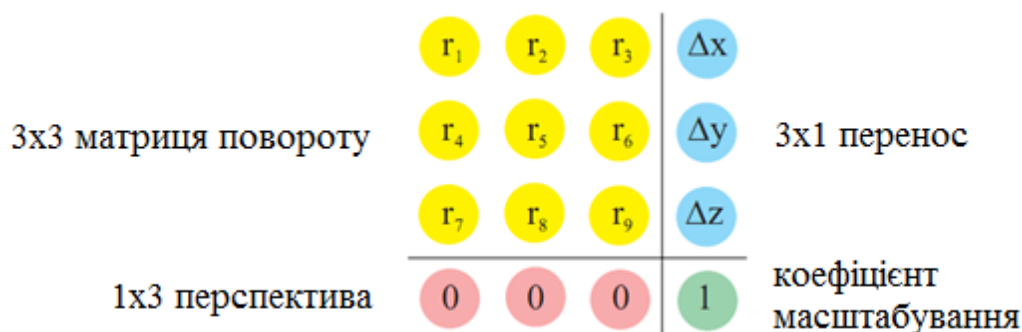


Рис.2.3. - Однорідна матриця перетворення.

Матриця перетворення ставиться у відповідність кожній локальній системі координат (крім найпершої, що біля стійки) і має два завдання.

- 1) Вона описує стан і орієнтацію локальної системи координат (i -ї) щодо попередньої локальної системи координат ($i-1$, якщо нумерувати, починаючи зі стійки).
- 2) Являє собою засіб перетворення i -ї координатної системи в $i-1$, тобто координати вектору в i -й системі координат можуть бути представлені в $i-1$ -й шляхом множення координат цього вектору на i -у матрицю перетворень (A матрицю).

Матрицю A , як показано на малюнку 2.3., можна розділити на 4 зони:

- 1) Матриця 3×3 , яка визначає, з одного боку орієнтацію відповідної локальної системи координат щодо попередньої за допомогою кутів Ейлера, і з іншого боку, є матрицею повороту.
- 2) Вектор 3×1 , з одного боку, є координатою i -ї локальної системи щодо $i-1$ -ї та, з іншого боку, є вектором переносу.
- 3) Рядок 1×3 визначає перспективу.
- 4) Останній елемент (в правому нижньому кутку), є коефіцієнтом масштабування.

Останні дві частини не використовуються. Вектор положення і коефіцієнт масштабування взагалі з точки зору лінійної алгебри можна сприймати як єдине ціле - вектор, виражений в однорідних координатах.

Матриці перетворення для локальної системи координат вираховуються згідно з геометричних особливостей ланок. Уявлення ланок згідно Денавіта-Хартенберга залежать від чотирьох геометричних параметрів, які відповідають кожній ланці. Ці чотири параметри повністю описують будь який обертальний або поступальний рух і їх набір достатній для опису кінематичної схеми кожної ланки (Рис 2.4)[2]. Вони представлені нижче:

- 1) θ_i – приєднаний кут (узагальнена координата) - кут, на який треба повернути вісь x_{i-1} навколо осі z_{i-1} , щоб вона стала спрямована в один бік з віссю x_i (знак визначається відповідно до правила правої руки).

											ЕП.МР.18.09.02.ПЗ	Арк.
												31
Змн.	Арк.	№ докум.	Підпис	Дата								

- 2) d_i – відстань між перетином осі z_{i-1} з віссю x_i і початком ($i-1$ -ї системи координат, що відраховується вздовж осі z_{i-1}).
- 3) a_i – лінійний зсув - відстань між перетином осі z_{i-1} з віссю x_i та початком i -ї системи координат, що відраховується вздовж осі x_i , тобто найкоротша відстань між осями z_{i-1} і z_i .
- 4) α_i – кутовий зсув - кут, на який треба повернути вісь z_{i-1} навколо осі x_i , щоб вона стала спрямованої в один бік з віссю z_i (знак визначається відповідно до правила правої руки).

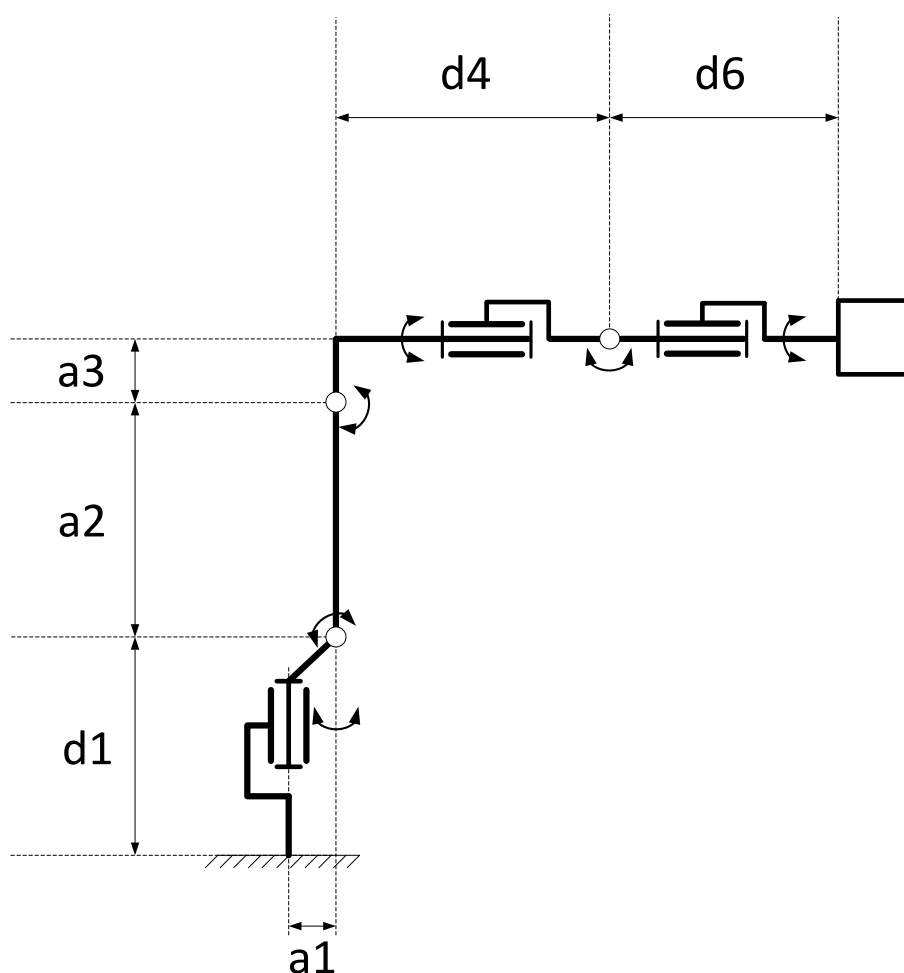


Рис 2.4. - Кінематична схема маніпулятора

Знаючи параметри кожної ланки, і маючи кінематичну схему, побудуємо таблицю з параметрами зчленувань для заданого робота (табл. 2.2).

Таблиця 2.2. - Параметри систем координат ланок маніпулятора в уявленні Денавіта - Хартенберга

З'єднання, i	θ	α	a (мм)	d (мм)
1	Θ_1	-90°	$a_1=45$	$d_1=75$
2	Θ_2-90°	0°	$a_2=115$	0
3	Θ_3	-90°	$a_3=20$	0
4	Θ_4	90°	0	$d_4=130$
5	Θ_5	-90°	0	0
6	Θ_6	0°	0	$d_6=50$

Як тільки системи координат сформовані для всіх ланок, можна побудувати однорідні матриці перетворення, що зв'язують i-у та i-1-у системи координат. Робиться це згідно з наступною формулою:

$$A_{i-1}^i = R(z_{i-1}, \theta_i)T(z_{i-1}, d_i)T(x_i, a_i)R(x_i, a_i) \quad (2.1)$$

Надалі індекс i-1 будемо нехтувати, і називати матрицю A_{i-1}^i просто матрицею A_i .

Тобто систему i-1 можна перетворити в систему i за допомогою повороту $R(z_{i-1}, \theta_i)$, двох переміщень (переносів) $T(z_{i-1}, d_i)T(x_i, a_i)$ та ще одного повороту $R(x_i, a_i)$, які виконуються в наступному порядку:

1. Поворот системи i-1 навколо вісі z_{i-1} на кут θ_i до тих пір, поки вісь x_{i-1} не стане паралельною вісі x_i .

$$R(z_{i-1}, \theta_i) = \begin{pmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.2)$$

2. Зміщення поверненої системи уздовж вісі z_{i-1} на величину d_i доти, поки вісі x_{i-1} і x_i не опиняться на одній прямій.

$$T(z_{i-1}, d_i) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.3)$$

3. Зміщення вздовж осі x_i на величину a_i до тих пір, поки початки координат не будуть співпадати.

$$T(x_i, a_i) = \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.4)$$

4. Поворот навколо вісі x_i на кут α_i до суміщення вісі z_{i-1} с віссю z_i .

$$R(x_i, \alpha_i) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_1) & -\sin(\theta_1) & 0 \\ 0 & \sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.5)$$

Кожному з цих елементарних рухів відповідає одна з матриць: або матриця обертання, або матриця зсуву. Результуюча матриця переходу A_i , що зв'язує системи $i-1$ та i , є добутком цих матриць:

Після перемноження отримуємо:

$$A_i = \begin{pmatrix} \cos(\theta_i) & -\cos(\alpha_i) \cdot \sin(\theta_i) & \sin(\alpha_i) \cdot \sin(\theta_i) & a_i \cdot \cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i) \cdot \cos(\theta_i) & -\sin(\alpha_i) \cdot \cos(\theta_i) & a_i \cdot \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.6)$$

Далі щоб отримати позицію кінцевого ланки в системі координат основи, потрібно виконати перемноження однорідних матриць перетворення.

$$T_0^6 = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6 \quad (2.7)$$

Для робота розглянутого в даному дипломному проекті ці матриці приймуть вид:

$$A_1 = \begin{pmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & 45 \cdot \cos(\theta_1) \\ \sin(\theta_1) & 0 & \cos(\theta_1) & 45 \cdot \sin(\theta_1) \\ 0 & -1 & 0 & 75 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.8)$$

$$A_2 = \begin{pmatrix} \cos(\theta_2 - \frac{\pi}{2}) & -\sin(\theta_2 - \frac{\pi}{2}) & 0 & 115 \cdot \cos(\theta_2 - \frac{\pi}{2}) \\ \sin(\theta_2 - \frac{\pi}{2}) & \cos(\theta_2 - \frac{\pi}{2}) & 0 & 115 \cdot \sin(\theta_2 - \frac{\pi}{2}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.9)$$

$$A_3 = \begin{pmatrix} \cos(\theta_3) & 0 & -\sin(\theta_3) & 20 \cdot \cos(\theta_3) \\ \sin(\theta_3) & 0 & \cos(\theta_3) & 20 \cdot \sin(\theta_3) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.10)$$

$$A_4 = \begin{pmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & 0 \\ \sin(\theta_4) & 0 & -\cos(\theta_4) & 0 \\ 0 & 1 & 0 & 130 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.11)$$

$$A_5 = \begin{pmatrix} \cos(\theta_5) & 0 & -\sin(\theta_5) & 0 \\ \sin(\theta_5) & 0 & \cos(\theta_5) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.12)$$

$$A_6 = \begin{pmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & 50 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.13)$$

Припустимо, що:

$$T_0^6 = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.14)$$

тоді,

p_x, p_y, p_z – координати вектору положення

a_x, a_y, a_z – координати вектору підходу

o_x, o_y, o_z – координати вектору орієнтації

n_x, n_y, n_z – координати вектору нормалі

Графічно перераховані вище вектори можна зобразити, так як це показано на рис 2.5.

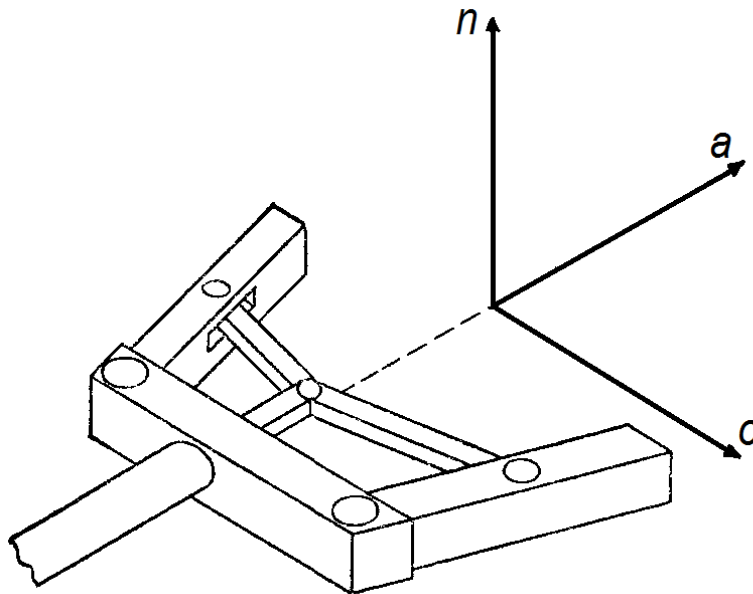


Рис 2.5. - Орієнтація схвата і його система координат

2.2. Розрахунок інверсної кінематики

Завдання, що розглядається в даному підрозділі, пов'язана зі знаходженням кута θ_i для кожного з'єднання, за умови, що інші параметри маніпулятора, а саме, α_i, d_i, a_i спочатку відомі.

Як тільки параметри ланок встановлені і ми задалися початковим станом, можуть бути легко побудовані матриці A_{i-1}^i і здійснено послідовне перемноження, що вже було розглянуто в попередньому розділі. Подальше рішення залежить від вибору конкретного методу розв'язання задачі інверсної кінематики. Існує два варіанти математичний (алгебраїчний) або геометричний. Не складно здогадатися, що велика кількість ступенів свободи сприяє ускладненню математичного рішення. Тому в подальшому розглянемо геометричне рішення задачі зворотної кінематики.

Знаходження θ_1

Починаючи вирішувати завдання, на вході ми маємо матрицю T_0^6 , в якій вказані кінцеві координати і положення схвата. Нам же, для знаходження першого кута, та й надалі, потрібно розглядати матрицю \bar{T}_0^4 (2.15).

										Арк.
										36
Змн.	Арк.	№ докум.	Підпис	Дата						

$$\bar{T}_0^4 = A1 \cdot A2 \cdot A3 \cdot A4 = \begin{pmatrix} n_x & o_x & a_x & T_{0x}^4 \\ n_y & o_y & a_y & T_{0y}^4 \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.15)$$

Щоб перейти від першої матриці до другої скористаємося наступним виразом:

$$P_m = T_0^6 - W \cdot d_6 = \begin{pmatrix} T_{0x}^6 \\ T_{0y}^6 \\ T_{0z}^6 \\ 1 \end{pmatrix} - \begin{pmatrix} a_{x6} \\ a_{y6} \\ a_{z6} \\ 0 \end{pmatrix} \cdot d_6 = \begin{pmatrix} T_{0x}^4 \\ T_{0y}^4 \\ p_z \\ 1 \end{pmatrix} \quad (2.16)$$

Таким чином вектор P_m має координати, які описують стан робота без урахування зап'ястя (схвата), що достатньо для визначення першого кута. Щоб знайти кут θ_1 спроектуємо вектор \bar{T}_0^4 на систему координат основи (Рис 2.6.). Далі, опираючись на одержані проекції T_{0x}^4 та T_{0y}^4 , як відношення протилежного катета до прилеглого катета, знайдемо потрібний кут.

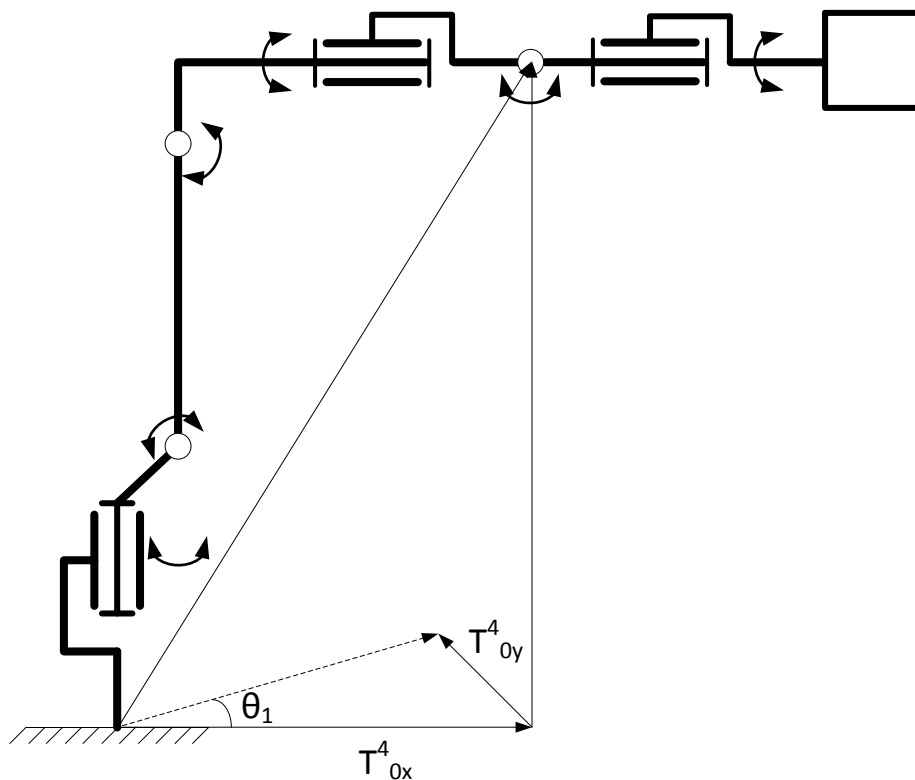


Рис. 2.6. - Геометричне зображення для розрахунку першого кута

Тоді,

$$\tan(\theta_1) = \frac{T_{0y}^4}{T_{0x}^4} \rightarrow \theta_1 = \arctan\left(\frac{T_{0y}^4}{T_{0x}^4}\right) \quad (2.17)$$

Для того що б рішення було в придатній формі (а саме кути розраховувалися з урахуванням знаків та визначені в інтервалі від $-\pi$ до $+\pi$) скористаємося наступним виразом:

$$\theta_1 = \arctan2(T_{0y}^4, T_{0x}^4) \quad (2.18)$$

Знаходження θ_2

Згідно рис.2.7. шуканий кут буде дорівнювати:

$$\theta_2 = \begin{cases} \frac{\pi}{2} - \beta - \gamma & \text{кут позитивний} \\ \frac{\pi}{2} - \beta + \gamma & \text{кут негативний} \end{cases} \quad (2.19)$$

де,

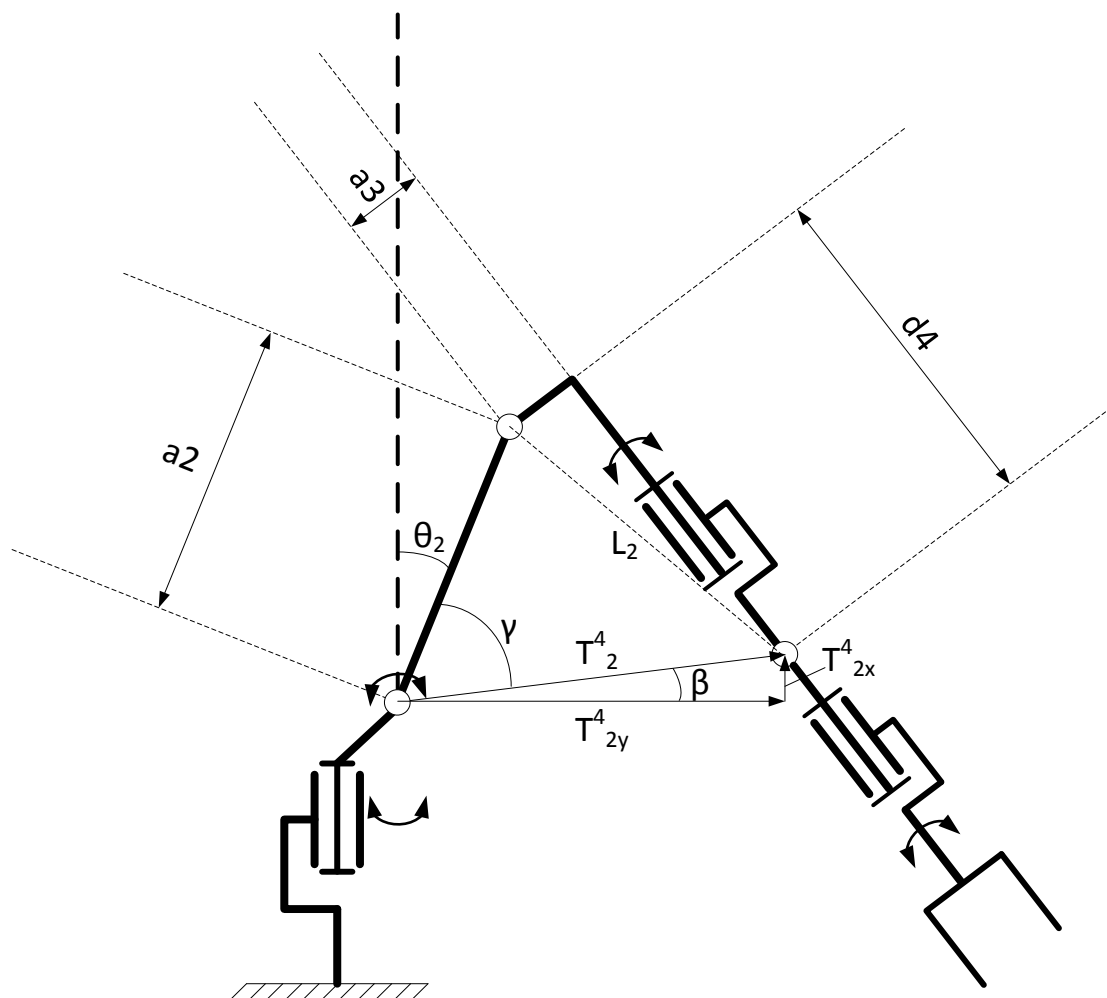


Рис.2.7. - Геометричне зображення для розрахунку другого кута

$$\beta = \arctan\left(\frac{T_{2x}^4}{T_{2y}^4}\right) \quad (2.20)$$

або

$$\beta = \arctan2(T_{2x}^4, T_{2y}^4) \quad (2.21)$$

де

$$\bar{T}_2^4 = A2 \cdot A3 \cdot A4 = \begin{pmatrix} n_x & o_x & a_x & T_{2x}^4 \\ n_y & o_y & a_y & T_{2y}^4 \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.22)$$

Варто зазначити що для того що б отримати координати T_{2x}^4 та T_{2y}^4 як показано в рівнянні 2.22, потрібно вектор P_m помножити на зворотну матрицю A_1^{-1} . Матрицю A_1 можна скласти на підставі раніше розрахованого кута. Тоді:

$$P_{11} = A_1^{-1} \cdot P_m = \begin{pmatrix} T_{2x}^4 \\ T_{2y}^4 \\ p_z \\ 1 \end{pmatrix} \quad (2.23)$$

Кут гамма знайдем через теорему косинусів, де

$$\cos(\gamma) = \frac{a_2^2 + |\bar{T}_2^4|^2 - L_2^2}{2 \cdot a_2 \cdot |\bar{T}_2^4|} \quad (2.24)$$

тоді

$$\gamma = \arccos\left(\frac{a_2^2 + |\bar{T}_2^4|^2 - L_2^2}{2 \cdot a_2 \cdot |\bar{T}_2^4|}\right) \quad (2.25)$$

де

$$|\bar{T}_2^4| = \sqrt{(T_{2x}^4)^2 + (T_{2y}^4)^2} \quad (2.26)$$

$$L_2 = \sqrt{(a_3)^2 + (d_4)^2} \quad (2.27)$$

Знаходження θ_3

Згідно рис.2.8. шуканий кут буде дорівнювати:

$$\theta_3 = \begin{cases} \pi - \varphi - \alpha & \text{кут позитивний} \\ \pi - \varphi + \alpha & \text{кут негативний} \end{cases} \quad (2.28)$$

Згідно вже згаданої теореми косинусів знайдемо φ та α .

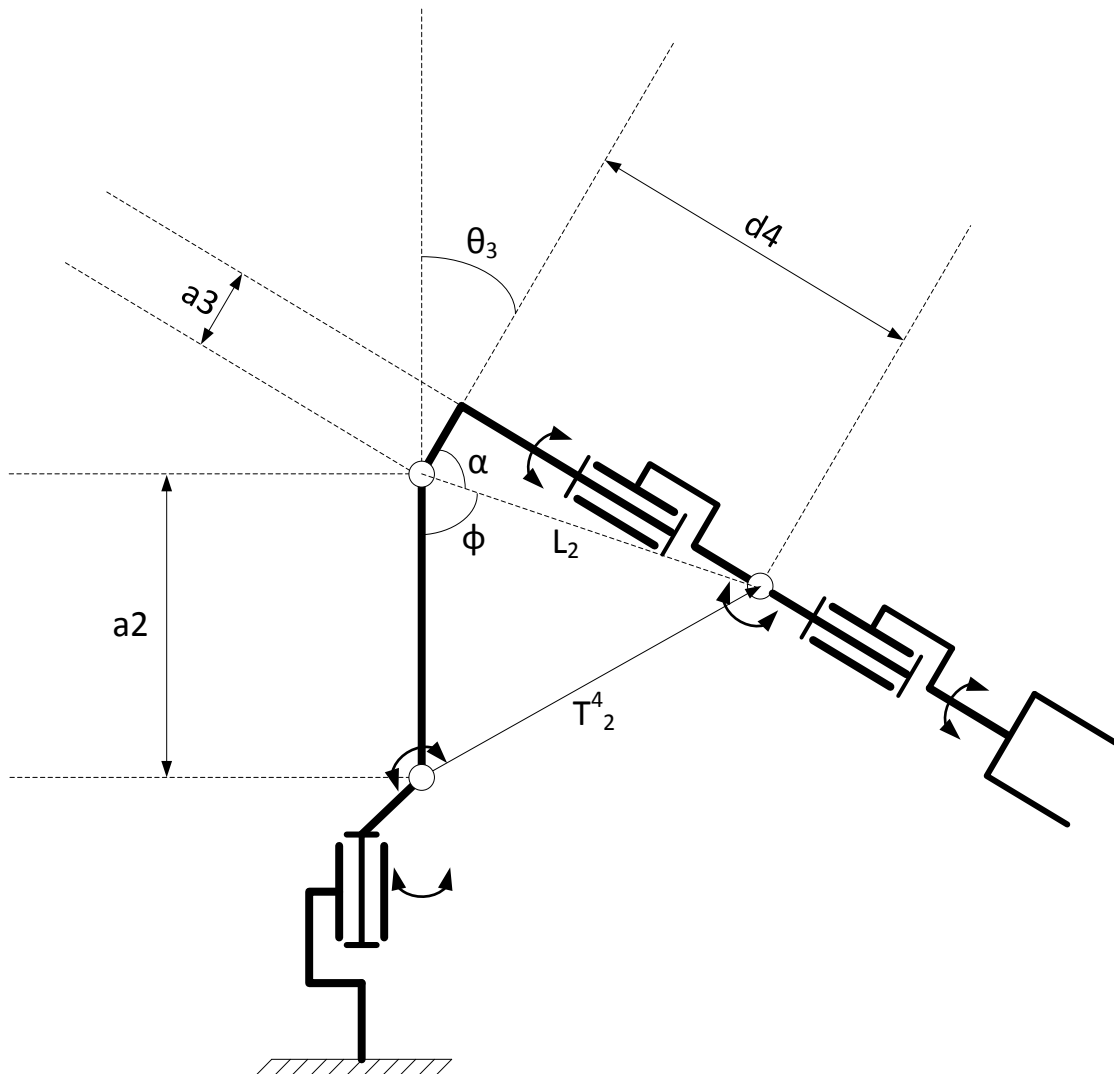


Рис.2.8. - Геометричне зображення для розрахунку третього кута де,

$$\alpha = \arccos\left(\frac{a_3^2 + L_2^2 - d_4^2}{2 \cdot a_2 \cdot L_2}\right) \quad (2.29)$$

$$\varphi = \arccos\left(\frac{a_2^2 + L_2^2 - |T_2^4|^2}{2 \cdot a_2 \cdot L_2}\right) \quad (2.30)$$

Знаходження θ_5

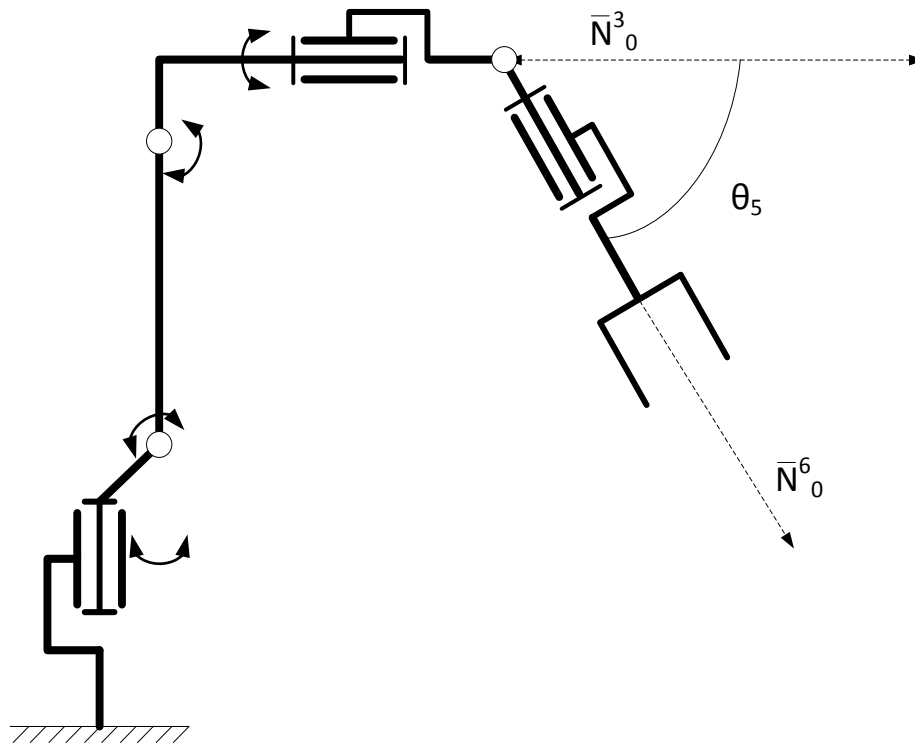


Рис. 2.9. - Геометричне зображення для розрахунку п'ятого кута

Вектор \bar{N}_0^6 , який заданий задалегідь, описується щодо системи координат основи наступним рівнянням:

$$\bar{T}_0^6 = A1 \cdot A2 \cdot A3 \cdot A4 \cdot A5 \cdot A6 = \begin{pmatrix} n_{x6} & o_{x6} & a_{x6} & p_{x6} \\ n_{y6} & o_{y6} & a_{y6} & p_{y6} \\ n_{z6} & o_{z6} & a_{z6} & p_{z6} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.31)$$

де,

$$\bar{N}_0^6 = \begin{pmatrix} a_{x6} \\ a_{y6} \\ a_{z6} \end{pmatrix} \quad (2.32)$$

В свою чергу вектор \bar{N}_0^3 відносно системи координат основи можна виразити таким рівнянням ($A1, A2, A3$ – отримуємо на підставі раніше розрахованих кутів):

$$\bar{T}_0^3 = A1 \cdot A2 \cdot A3 = \begin{pmatrix} n_{x3} & o_{x3} & a_{x3} & p_{x3} \\ n_{y3} & o_{y3} & a_{y3} & p_{y3} \\ n_{z3} & o_{z3} & a_{z3} & p_{z3} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.33)$$

де,

										ЕП.МР.18.09.02.ПЗ	Арк.
											41
Змн.	Арк.	№ докум.	Підпис	Дата							

$$\bar{N}_0^3 = \begin{pmatrix} a_{x3} \\ a_{y3} \\ a_{z3} \end{pmatrix} \quad (2.34)$$

З формули скалярного добутку векторів знаходимо шуканий кут:

$$\bar{N}_0^6 \cdot \bar{N}_0^3 = |\bar{N}_0^6| \cdot |\bar{N}_0^3| \cdot \cos(\theta_5) \quad (2.35)$$

$$\theta_5 = \arccos\left(\frac{\bar{N}_0^6 \cdot \bar{N}_0^3}{|\bar{N}_0^6| \cdot |\bar{N}_0^3|}\right) \quad (2.36)$$

Знаходження θ_4

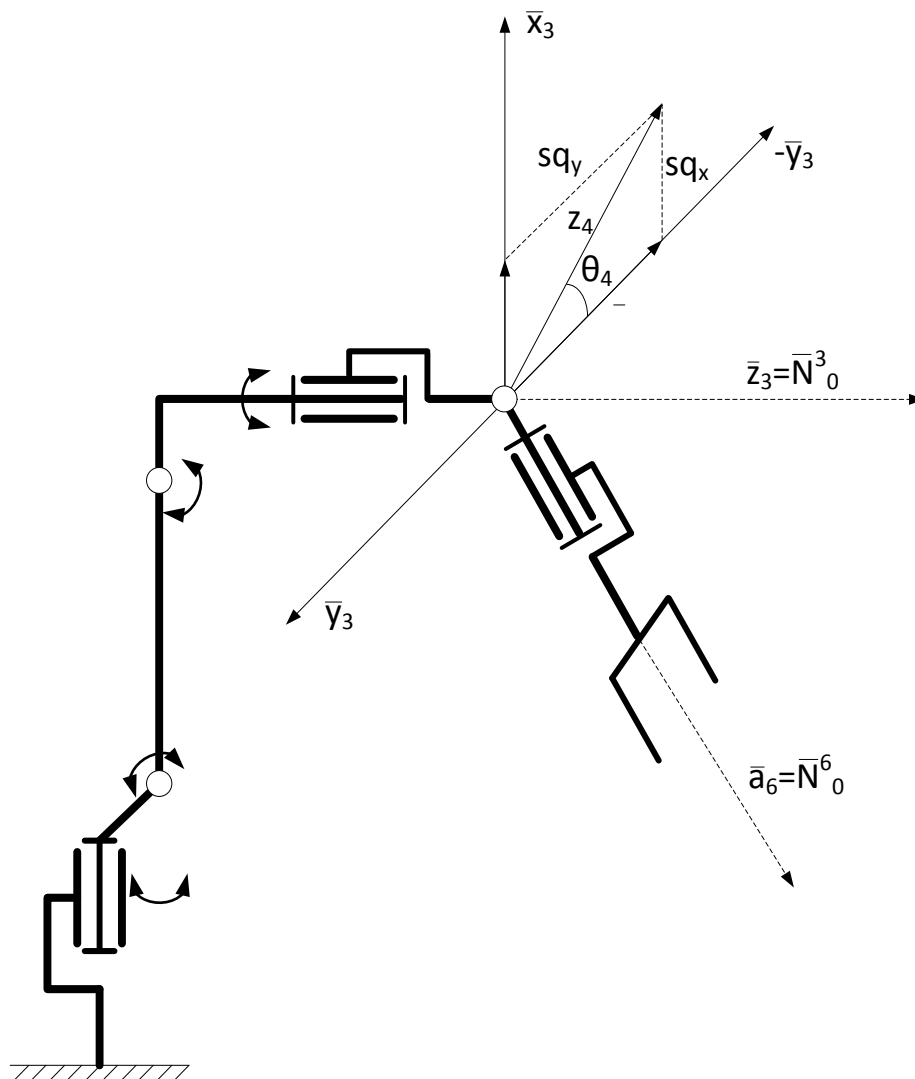


Рис. 2.10. Геометричне зображення для розрахунку четвертого кута

Як бачимо з рис.2.10. кут θ_4 можна знайти, знаючи проєкції вектору \bar{z}_4 на вектор \bar{x}_3 і \bar{y}_3 . В такому випадку θ_4 буде дорівнювати:

$$\theta_4 = \arctan\left(\frac{sq_y}{sq_x}\right) \text{ или } \theta_4 = \arctan2(sq_y, sq_x) \quad (2.37)$$

					ЕП.МР.18.09.02.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

де,

$$sq_y = \bar{z}_4 \cdot (-\bar{y}_3) \quad (2.38)$$

$$sq_x = \bar{z}_4 \cdot \bar{x}_3 \quad (2.39)$$

в свою чергу \bar{z}_4 являється векторних добутком \bar{z}_3 и \bar{a}_6 ,

$$\bar{z}_4 = \bar{z}_3 \times \bar{a}_6 \quad (2.40)$$

Звернемося до рівнянь 2.31 та 2.33 і виразимо інші невідомі змінні:

$$\bar{z}_3 = \bar{N}_0^3 = \begin{pmatrix} a_{x3} \\ a_{y3} \\ a_{z3} \end{pmatrix} \quad (2.41)$$

$$\bar{x}_3 = \begin{pmatrix} n_{x3} \\ n_{y3} \\ n_{z3} \end{pmatrix} \quad (2.42)$$

$$\bar{y}_3 = \begin{pmatrix} o_{x3} \\ o_{y3} \\ o_{z3} \end{pmatrix} \quad (2.43)$$

$$\bar{a}_6 = \bar{N}_0^6 = \begin{pmatrix} a_{x6} \\ a_{y6} \\ a_{z6} \end{pmatrix} \quad (2.44)$$

Знаходження θ_6

Як бачимо з рис.2.11. кут θ_6 можна знайти, знаючи проєкції вектору \bar{x}_6 на вектор \bar{x}_5 та \bar{y}_5 . В такому випадку θ_6 буде дорівнювати:

$$\theta_6 = \arctan\left(\frac{cq_x}{cq_y}\right) \text{ или } \theta_6 = \arctan2(cq_x, cq_y) \quad (2.45)$$

де,

$$cq_y = \bar{x}_6 \cdot \bar{y}_5 \quad (2.46)$$

$$cq_x = \bar{x}_6 \cdot \bar{x}_5 \quad (2.47)$$

Визначимо інші невідомі.

					ЕП.МР.18.09.02.ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

$$\bar{x}_6 = \bar{N}_0^6 = \begin{pmatrix} a_{x6} \\ a_{y6} \\ a_{z6} \end{pmatrix} \quad (2.48)$$

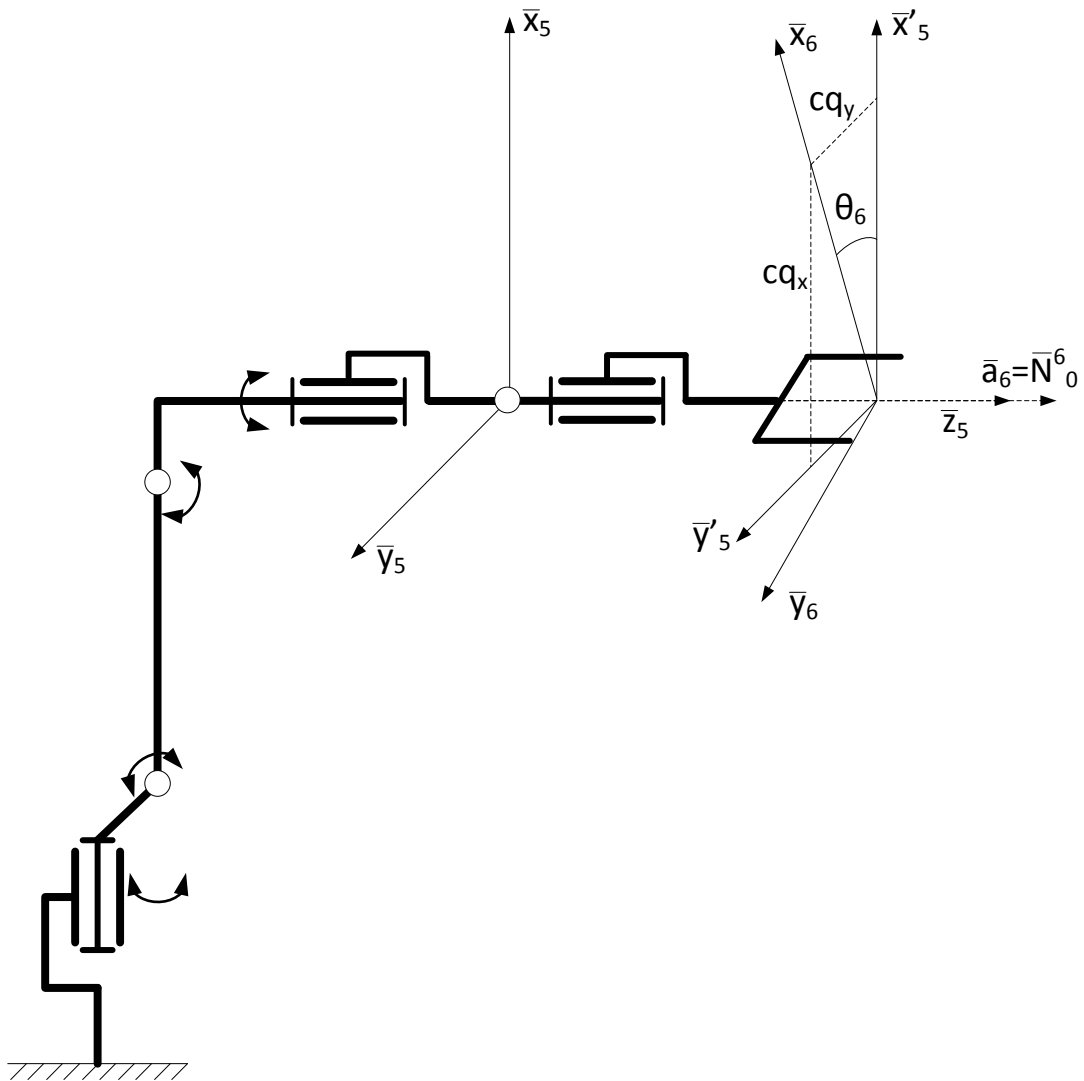


Рис.2.11. - Геометричне зображення для розрахунку шостого кута

$$\bar{T}_0^5 = A1 \cdot A2 \cdot A3 \cdot A4 \cdot A5 = \begin{pmatrix} n_{x5} & o_{x5} & a_{x5} & p_{x5} \\ n_{y5} & o_{y5} & a_{y5} & p_{y5} \\ n_{z5} & o_{z5} & a_{z5} & p_{z5} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.49)$$

$$\bar{y}_5 = \begin{pmatrix} o_{x5} \\ o_{y5} \\ o_{z5} \end{pmatrix} \quad (2.50)$$

$$\bar{x}_5 = \begin{pmatrix} n_{x5} \\ n_{y5} \\ n_{z5} \end{pmatrix} \quad (2.51)$$

2.3. Алгоритм розпізнавання об'єктів

Алгоритм розпізнавання (Рис. 2.12.) заснований на розглянутому раніше методі сегментації. У такому випадку нам потрібно виділити об'єкт певного кольору, з одержуваного в реальному часі зображення. Цю процедуру можна описати таким чином.

Зображення, отримане з веб камери, має RGB модель кодування кольору, побудовану на трьох основних кольорах. Кожен з кольорів R-Червоний, G-Зелений і B-Синій мають один з 256 рівнів інтенсивності. Найчастіше дану модель представляють у вигляді куба (Рис. 2.13.), де кольори розподілятися уздовж зазначених векторів.

Але при створенні програми використання даної моделі не є оптимальним, це пов'язано з особливостями реалізації бібліотеки EmguCV і з тим фактом, що пристрій колірної моделі HSV ближче до людського сприйняття кольорів, компоненти кольору в HSV відображають інформацію про колір в більш звичній формі для людського сприйняття: Що це за колір? Наскільки він насичений? Наскільки він світлий або темний? [4]

					ЕП.МР.18.09.02.ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 2.12. – Блок схема алгоритму розпізнавання

Далі детально розглянемо алгоритм розпізнавання і функції які він викликає.

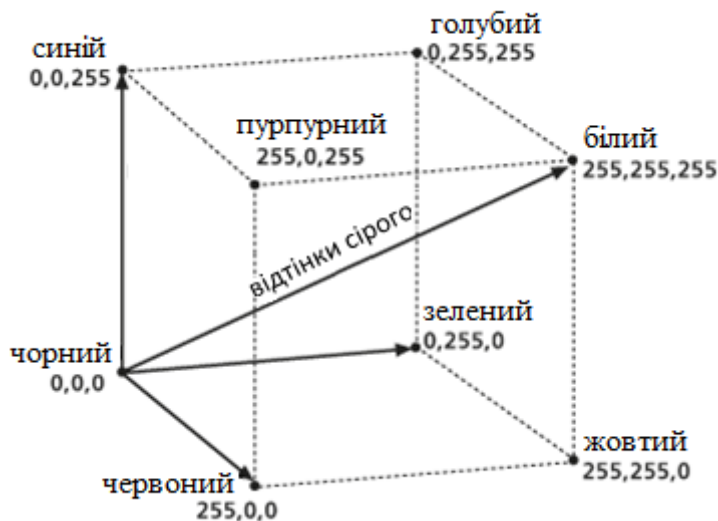


Рис. 2.13. – Просторове відображення моделі RGB

У моделі HSV (Рис. 2.14.) колір розкладається на 3 складові Hue (колірний тон) - частота колірної хвилі, яка відбивається від об'єкта, змінюється від 0 ° до 360 °; Saturation (Насиченість) - є чистотою кольору, змінюється від 0 до 100%; Value (Brightness) - загальна яскравість кольору, змінюється від 0 до 100%.

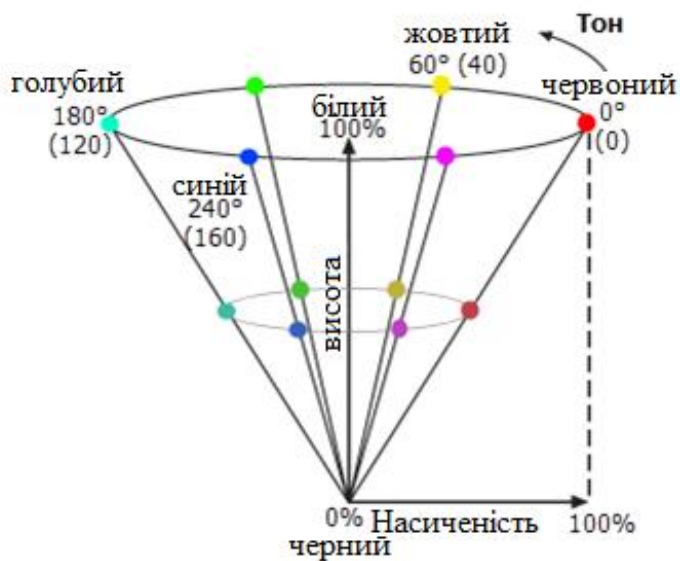


Рис. 2.14. – Просторове відображення моделі HSV

Для переходу між колірними моделями існують такі вирази [9]:

$$S = \begin{cases} 0, \text{ если } MAX = 0 \\ 1 - \frac{MIN}{MAX}, \text{ інакше} \end{cases} \quad (2.52)$$

$$H = \begin{cases} 60 \times \frac{G - B}{MAX - MIN} + 0, \text{ якщо } MAX = R \text{ и } G \geq B \\ 60 \times \frac{G - B}{MAX - MIN} + 360, \text{ якщо } MAX = R \text{ и } G < B \\ 60 \times \frac{G - B}{MAX - MIN} + 120, \text{ якщо } MAX = G \\ 60 \times \frac{G - B}{MAX - MIN} + 240, \text{ якщо } MAX = B \end{cases} \quad (2.53)$$

$$V = MAX \quad (2.54)$$

де,

MAX – максимальне значення з R, G и B, а MIN — мінімальне з них.

$H \in [0, 360]$

$S, V, R, G, B \in [0, 1]$

Після конвертації потрібно встановити межі для кожного каналу, які дозволять виділити об'єкт потрібного кольору.

					ЕП.МР.18.09.02.ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

пікселі, які є у всіх трьох порогових зображень) за це відповідає спеціально створена функція `cvAndHsvImage ()`, її блок схема приведена на рисунку 2.17.



Рис. 2.17. – Коротка блок схема розробленої функції «`cvAndHsvImage`»

Далі визначаємо контури виділеної ділянки і знаходимо координати, центра обмежувального прямокутника за допомогою функції `RecDetection ()`, блок схема приведена на Рис. 2.18.

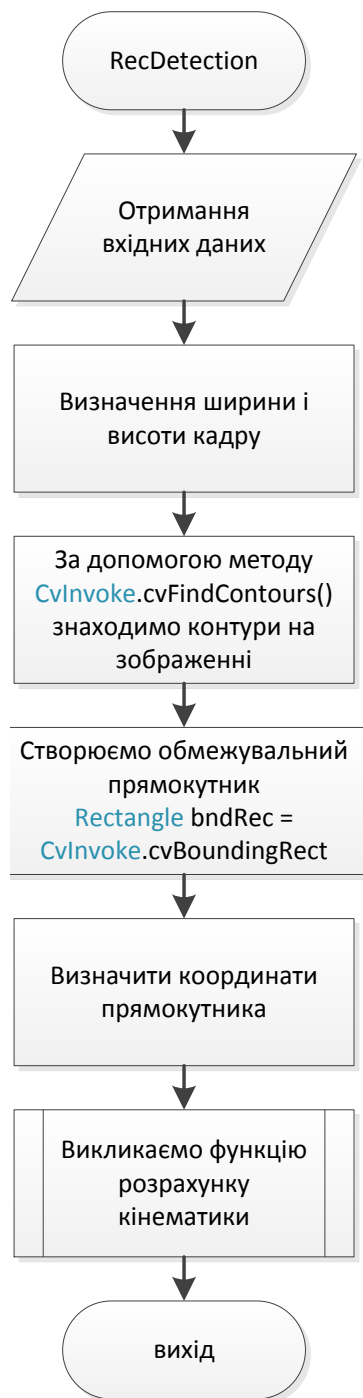


Рис. 2.18. – Коротка блок схема розробленої функції «RecDetection»
 Як можемо бачити в результаті виконання вищезгаданої функції (Рис. 2.19), об'єкт виділеного кольору був поміщений в прямокутну область.

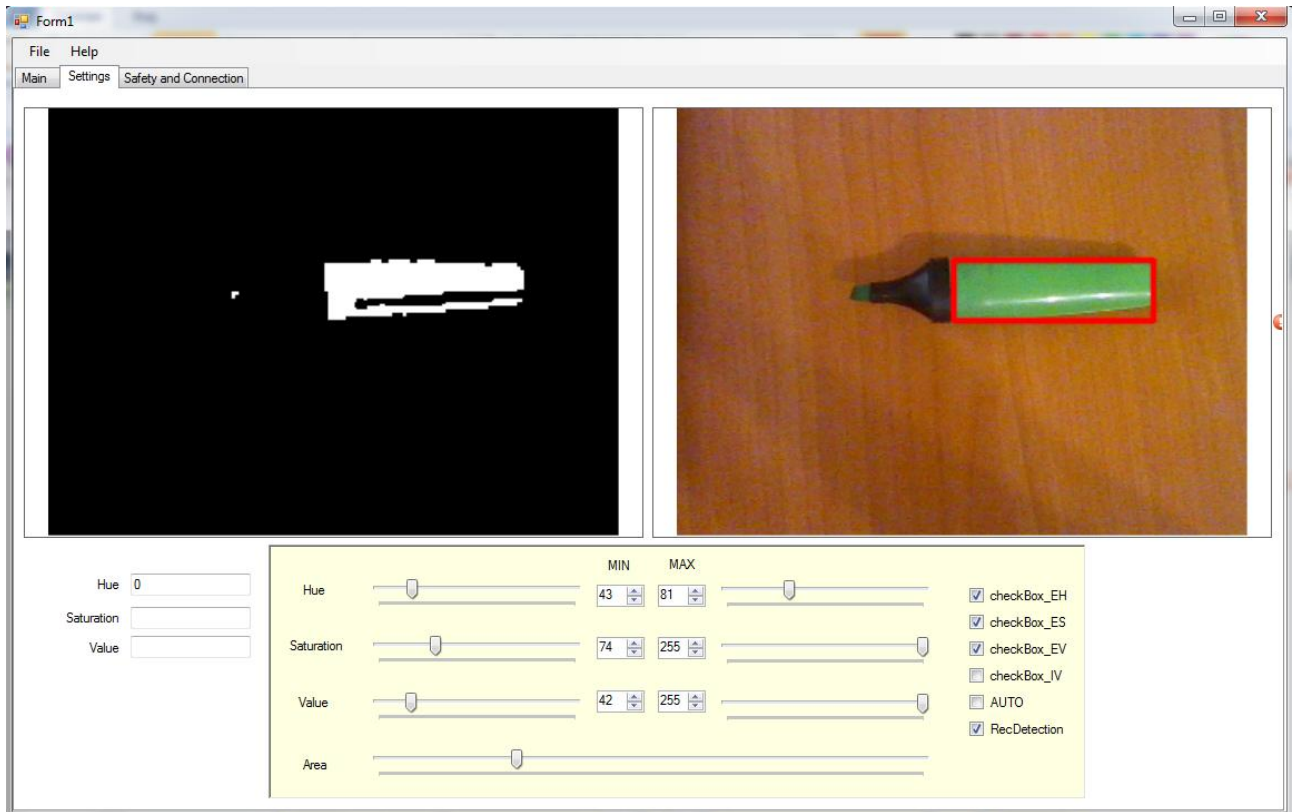


Рис. 2.19. – Інтерфейс розробленої програми

3. МОДЕЛЮВАННЯ РОБОТИ РОЗРОБЛЕНОЇ СИСТЕМИ

					<i>ЕП.МР.18.09.03.ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Лелько В.М.</i>			<i>МОДЕЛЮВАННЯ РОБОТИ РОЗРОБЛЕНОЇ СИСТЕМИ</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>		<i>Худолій С.С.</i>					<i>53</i>	<i>94</i>
<i>Реценз.</i>						<i>НГУ</i>		
<i>Н. Контр.</i>		<i>Казачковський</i>						
<i>Затверд.</i>								

В даному розділі, виконується моделювання роботи розробленого програмного забезпечення. Що дозволяє реалізувати вищезгадані алгоритми та розрахунки.

Для написання програмного забезпечення для робота маніпулятора була обрана об'єктно-орієнтована мова програмування C# призначена для програмування на платформі .NET. Вона оптимально використовує апаратні ресурси, і має дружній до розробника інтерфейс програмування.

Щоб створити додаток на вищезгаданій мові використовувалось середовище розробки Microsoft Visual Studio 2015, яке має набір інструментів для створення програмного забезпечення: від планування та розробки інтерфейсу користувача до написання коду, тестування, налагодження і аналізу продуктивності.

Для моделювання роботи системи використано пакет прикладних програм для числового аналізу MATLAB. В даному пакеті реалізовано розрахунок інверсної кінематики аналогічний тому що реалізований в Visual Studio. Алгоритм розрахунку інверсної кінематики представлений на рис. 3.2. Для більшої наочності роботи системи використано графічну трьох вимірну модель маніпулятора ABB IRB 4400, модель взято с сайту виробника[12]. В плані кінематики, ABB IRB 4400 є аналогом робота маніпулятора розглянутого в даному дипломному проекті.



3.1. Робот маніпулятор ABB IRB 4400

					ЕП.МР.18.09.03.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

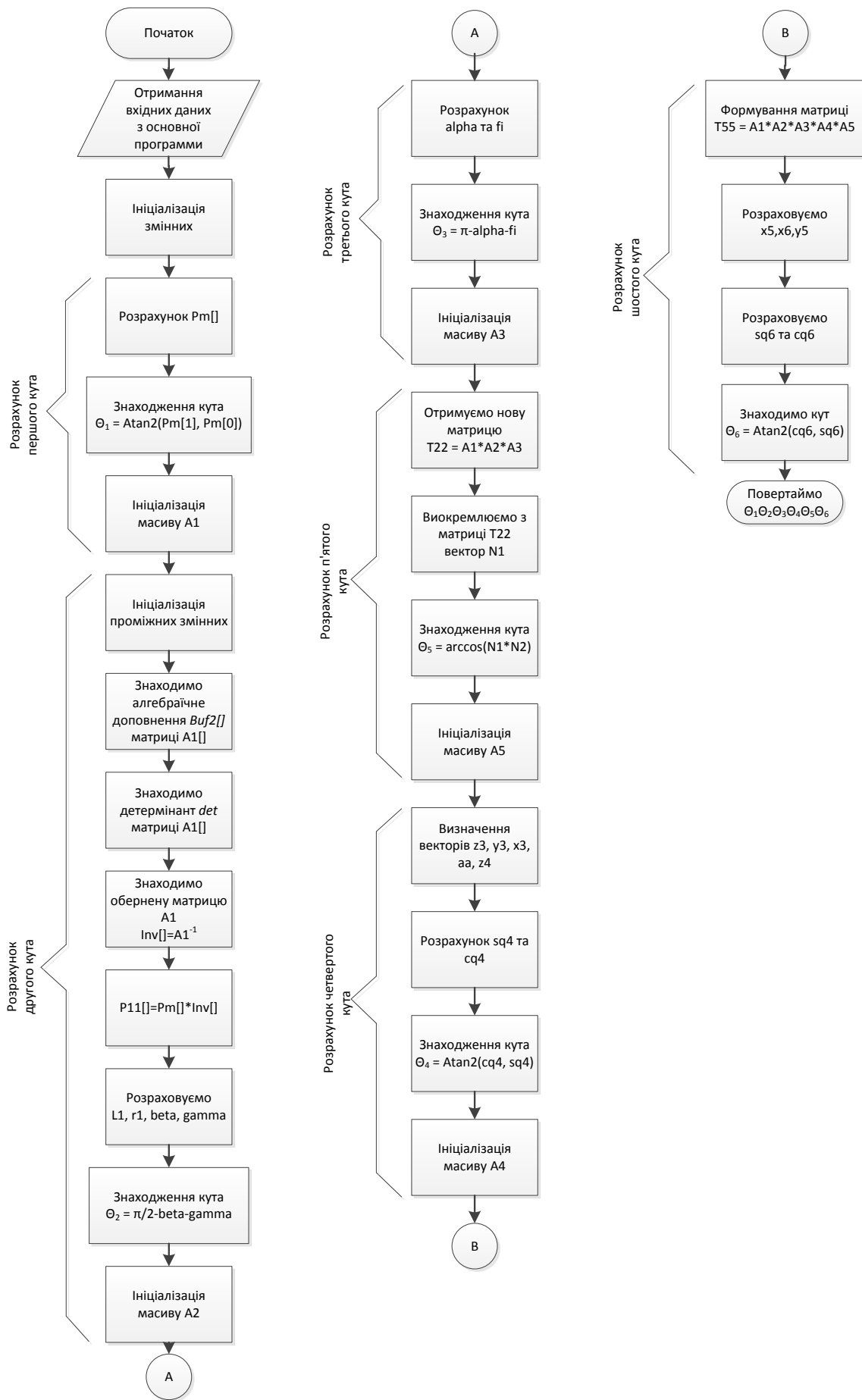


Рис. 3.2. Блок схема розробленої програми

Змн.	Арк.	№ докум.	Підпис	Дата

Процес розпізнавання об'єктів на зображенні, а саме алгоритми роботи основних виконавчих функцій були розглянуті в попередньому розділі. Тому в даному розділі приведемо лише приклади їх програмної реалізації в пакеті Visual Studio.

Для демонстрації роботи всієї системи в цілому збережемо координати контрольних точок що формують траєкторію руху кінцевою ланки маніпулятора в окремий файл, та за допомогою пакету MATLAB зчитуємо збереженні данні та інтерпретуємо їх у рух підключеної моделі маніпулятора.

3.1. Моделювання роботи алгоритму інверсної кінематики

Нижче представлена функція, яка є складовою частиною основної програми і реалізує розрахунок кутів і передачу розрахованих кутів в основну програму на підставі вхідних даних, а саме координат схвата робота.

```
public double[] Kinematic(double x, double y, double z, double w1, double
                        w2, double w3)
{
    // Initial data
    double[] a = new double[6] { 45, 115, 20, 0, 0, 0 };
    double[] d = new double[6] { 75, 0, 0, 130, 0, 50 };
    double[] a1 = new double[6] { -Math.PI / 2, 0, -Math.PI / 2,
    Math.PI / 2, -Math.PI / 2, 0 };
    double[] n = new double[6] { 0, 0, 0, 0, 0, 0 };
    double[,] T1 = new double[4, 4] { { 0, 0, w1, x },
                                      { 0, 1, w2, y },
                                      { -1, 0, w3, z },
                                      { 0, 0, 0, 1 } };
}
```

Далі виконується розрахунок кутів згідно формул представлених в 2 розділі.

Розрахунок першого кута:

```
double[] Pm = new double[4] { (T1[0, 3] - T1[0, 2] * d[5]),
                              (T1[1, 3] - T1[1, 2] * d[5]),
                              (T1[2, 3] - T1[2, 2] * d[5]),
                              (T1[3, 3] - T1[3, 2] * d[5]), };
n[0] = Math.Atan2(Pm[1], Pm[0]);
double[,] A1 = new double[4, 4] { { Math.Cos(n[0]), -
    Math.Cos(a1[0])*Math.Sin(n[0]), Math.Sin(a1[0])*Math.Sin(n[0]),
    a[0]*Math.Cos(n[0]) }, { Math.Sin(n[0]), Math.Cos(a1[0])*Math.Cos(n[0]), -
    Math.Sin(a1[0])*Math.Cos(n[0]), a[0]*Math.Sin(n[0]) },
    { 0, Math.Sin(a1[0]), Math.Cos(a1[0]), d[0] }, { 0, 0, 0, 1 } };
```

Розрахунок другого кута включає в себе знаходження оберненої матриці

A1.

```
double[] P1 = new double[4];
//-Find the minor
double[,] buf = new double[3, 3];
double[,] buf1 = new double[4, 4];
double[,] buf2 = new double[4, 4];
double[,] Inv = new double[4, 4];
double[] P11 = new double[4];
double det = 0;
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        int ki, kj, di, dj;
        di = 0;
        for (ki = 0; ki < 3; ki++)
        { // checking the line index
            if (ki == i) di = 1; dj = 0;
            for (kj = 0; kj < 3; kj++)
            { // column index check
                if (kj == j) dj = 1;
                buf[ki,kj] = A1[(ki + di),(kj + dj)];
            }
        }
        buf1[i, j] = (buf[0, 0] * buf[1, 1] * buf[2, 2]) +
(buf[2, 0] * buf[0, 1] * buf[1, 2]) + (buf[1, 0] * buf[0, 2] * buf[2, 1])
- (buf[2, 0] * buf[1, 1] * buf[0, 2]) -
(buf[1, 0] * buf[0, 1] * buf[2, 2]) - (buf[0, 0] * buf[1, 2] * buf[2,
1]);
        buf2[j, i] = buf1[i, j]; // transportation
    }
}
//-Find the determinant
for (int j = 0; j < 4; j++)
{
    det += Math.Pow(-1, 0 + j) * A1[0, j] * buf1[0, j];
}
/--Obtain inverse matrix A1
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        Inv[i, j] = Math.Pow(-1, i + j) / det * buf2[i, j];
    }
}
/--Pm[]*Inv[]
for (int i = 0; i < 4; i++)
{
```

									Арк.
									57
Змн.	Арк.	№ докум.	Підпис	Дата					

ЕП.МР.18.09.03.ПЗ

```

        P11[i] = Inv[i, 0] * Pm[0] + Inv[i, 1] * Pm[1] + Inv[i, 2] *
Pm[2] + Inv[i, 3] * Pm[3];
    }
    double L1 = Math.Sqrt(Math.Pow(a[2], 2) + Math.Pow(d[3], 2));
    double r1 = Math.Sqrt(Math.Pow(P11[0], 2) + Math.Pow(P11[1],
2));
    double beta = Math.Atan2(-P11[1], P11[0]);
    double gamma = Math.Abs(Math.Acos((Math.Pow(a[1], 2) +
Math.Pow(r1, 2) - Math.Pow(L1, 2)) / (2 * a[1] * r1)));
    n[1] = Math.PI / 2 - beta - gamma;
    double[,] A2 = new double[4, 4] { {Math.Cos(n[1] - Math.PI/2),
-Math.Cos(al[1]) * Math.Sin(n[1] - Math.PI/2),
Math.Sin(al[1]) * Math.Sin(n[1] - Math.PI/2), a[1] * Math.Cos(n[1] - Math.PI/2)
},
{Math.Sin(n[1] - Math.PI/2),
Math.Cos(al[1]) * Math.Cos(n[1] - Math.PI/2), -Math.Sin(al[1]) * Math.Cos(n[1] -
Math.PI/2), a[1] * Math.Sin(n[1] - Math.PI/2) },
{0,
Math.Sin(al[1]),
Math.Cos(al[1]),
0,
1 } };

```

Розрахунок третього кута:

```

    double alpha = Math.Abs(Math.Acos((Math.Pow(a[1], 2) -
Math.Pow(r1, 2) + Math.Pow(L1, 2)) / (2 * a[1] * L1)));
    double fi = Math.Abs(Math.Acos((Math.Pow(a[2], 2) -
Math.Pow(d[3], 2) + Math.Pow(L1, 2)) / (2 * a[2] * L1)));
    n[2] = Math.PI - fi - alpha;
    double[,] A3 = new double[4, 4] { {Math.Cos(n[2]), -
Math.Cos(al[2]) * Math.Sin(n[2]), Math.Sin(al[2]) * Math.Sin(n[2]),
a[2] * Math.Cos(n[2]) },
{Math.Sin(n[2]),
Math.Cos(al[2]) * Math.Cos(n[2]), -Math.Sin(al[2]) * Math.Cos(n[2]),
a[2] * Math.Sin(n[2]) },
{0,
Math.Sin(al[2]),
Math.Cos(al[2]),
0,
1 } };

```

Розрахунок п'ятого кута:

```

    double[,] T22b = new double[4, 4];
    double[,] T22 = new double[4, 4];
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            T22b[i, j] = A1[i, 0] * A2[0, j] + A1[i, 1] * A2[1, j]
+ A1[i, 2] * A2[2, j] + A1[i, 3] * A2[3, j];

```

					ЕП.МР.18.09.03.ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
}
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        T22[i, j] = T22b[i, 0] * A3[0, j] + T22b[i, 1] *
A3[1, j] + T22b[i, 2] * A3[2, j] + T22b[i, 3] * A3[3, j];
    }
}
double[] N1 = new double[3] { T22[0, 2], T22[1, 2], T22[2, 2]
};
double[] N2 = new double[3] { w1, w2, w3 };
n[4] = Math.Acos(N1[0] * N2[0] + N1[1] * N2[1] + N1[2] *
N2[2]);
double[,] A5 = new double[4, 4] { {Math.Cos(n[4]), -
Math.Cos(a1[4])*Math.Sin(n[4]), Math.Sin(a1[4])*Math.Sin(n[4]),
a[4]*Math.Cos(n[4]) },
{Math.Sin(n[4]),
Math.Cos(a1[4])*Math.Cos(n[4]), -Math.Sin(a1[4])*Math.Cos(n[4]),
a[4]*Math.Sin(n[4]) },
{0,
Math.Sin(a1[4]), Math.Cos(a1[4]), d[4] },
{0,
0, 1 } };

```

Розрахунок четвертого кута:

```

double[] z3 = new double[3] { T22[0, 2], T22[1, 2], T22[2, 2]
};
double[] y3 = new double[3] { T22[0, 1], T22[1, 1], T22[2, 1]
};
double[] x3 = new double[3] { T22[0, 0], T22[1, 0], T22[2, 0]
};
double[] aa = new double[3] { w1, w2, w3 };
double[] z4 = new double[3] {(z3[2]*aa[1]-z3[1]*aa[2]),
(z3[0]*aa[2]- z3[2] * aa[0]),
(z3[1]*aa[0]- z3[0] * aa[1]) };
double sq4 = z4[0] * y3[0] + z4[1] * y3[1] + z4[2] * y3[2];
double cq4 = z4[0] * -x3[0] + z4[1] * -x3[1] + z4[2] * -
x3[2];
n[3] = Math.Atan2(cq4,sq4);
double[,] A4 = new double[4, 4] { {Math.Cos(n[3]), -
Math.Cos(a1[3])*Math.Sin(n[3]), Math.Sin(a1[3])*Math.Sin(n[3]),
a[3]*Math.Cos(n[3]) },
{Math.Sin(n[3]),
Math.Cos(a1[3])*Math.Cos(n[3]), -Math.Sin(a1[3])*Math.Cos(n[3]),
a[3]*Math.Sin(n[3]) },
{0,
Math.Sin(a1[3]), Math.Cos(a1[3]), d[3] },
{0,
0, 1 } };

```

					ЕП.МР.18.09.03.ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

Розрахунок шостого кута:

```

double[,] T55 = new double[4,4];
double[,] T55b = new double[4, 4];
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        T55b[i, j] = T22[i, 0] * A4[0, j] + T22[i, 1] * A4[1,
j] + T22[i, 2] * A4[2, j] + T22[i, 3] * A4[3, j];
    }
}
for (int i = 0; i < 4; i++)
{
    for (int j = 0; j < 4; j++)
    {
        T55[i, j] = T55b[i, 0] * A5[0, j] + T55b[i, 1] *
A5[1, j] + T55b[i, 2] * A5[2, j] + T55b[i, 3] * A5[3, j];
    }
}
double[] x5 = new double[3] { T55[0, 0], T55[1, 0], T55[2, 0]
} ;
double[] y5 = new double[3] { T55[0, 1], T55[1, 1], T55[2, 1]
}; ;
double sq6 = Math.Round((x5[0] * y5[0] + x5[1] * y5[1] +
x5[2] * y5[2]), 3);
double cq6 = x5[0] * x5[0] + x5[1] * x5[1] + x5[2] * x5[2];
n[5] = Math.Atan2(sq6, cq6);

```

Повернення розрахованих кутів в основну програму.

```

return new double[] { n[0], n[1], n[2], n[3], n[4], n[5] };
}

```

Результати розрахунку інверсної кінематики представлені на рис. 3.3.

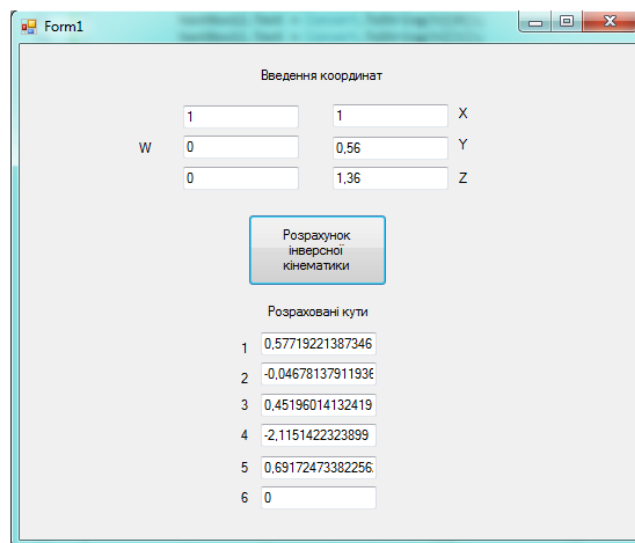
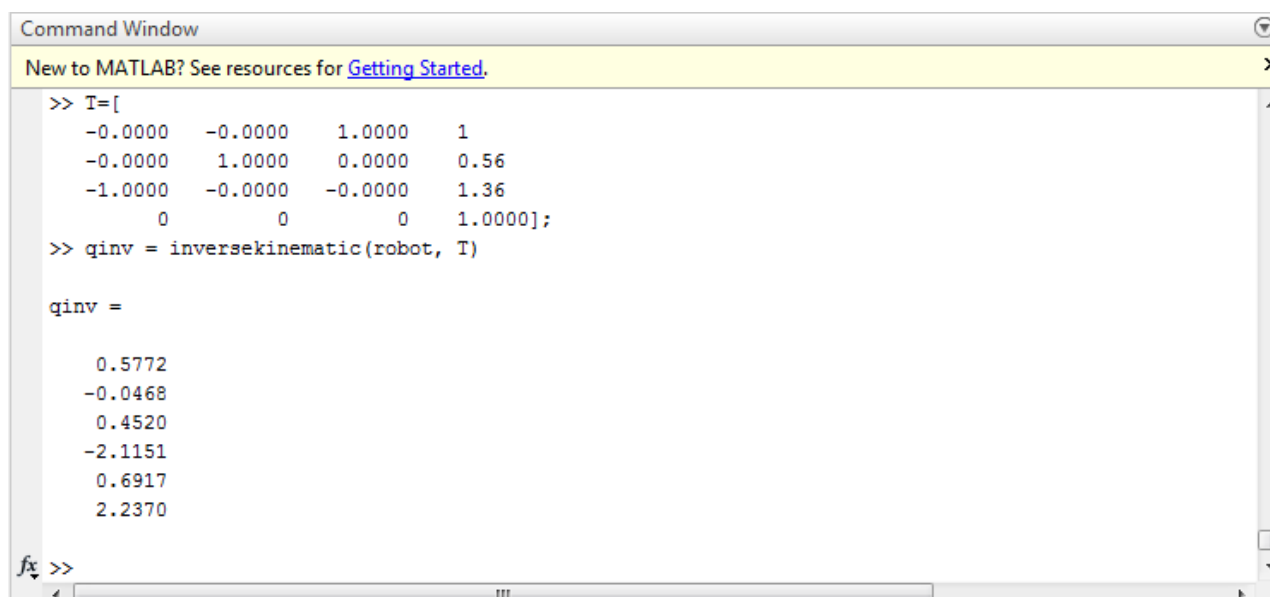


Рис. 3.3. Результат роботи описаної функції в пакеті MS Visual Studio

Для моделювання роботи системи в пакеті MATLAB був виконаний точно такий же розрахунок. Результати представлені на рисунку 3.4.



```
Command Window
New to MATLAB? See resources for Getting Started.
>> T=[
    -0.0000    -0.0000     1.0000     1
    -0.0000     1.0000     0.0000     0.56
    -1.0000    -0.0000    -0.0000     1.36
         0         0         0     1.0000];
>> qinv = inversekinematic(robot, T)

qinv =

    0.5772
   -0.0468
    0.4520
   -2.1151
    0.6917
    2.2370

fx >>
```

Рис. 3.4. Результат роботи описаної функції в пакеті MATLAB

Як можна побачити при введенні координат, а саме X: 1, Y: 0.56, Z: 1.36, програма вираховує кути для кожної частини маніпулятора. Для візуалізації проведених розрахунків скористаємося готовою моделлю робота ABB IRB 4400. Трансляція 3D-моделі (в форматі STL) в середовище MATLAB здійснюється за допомогою спеціального доповнення Arte[13].

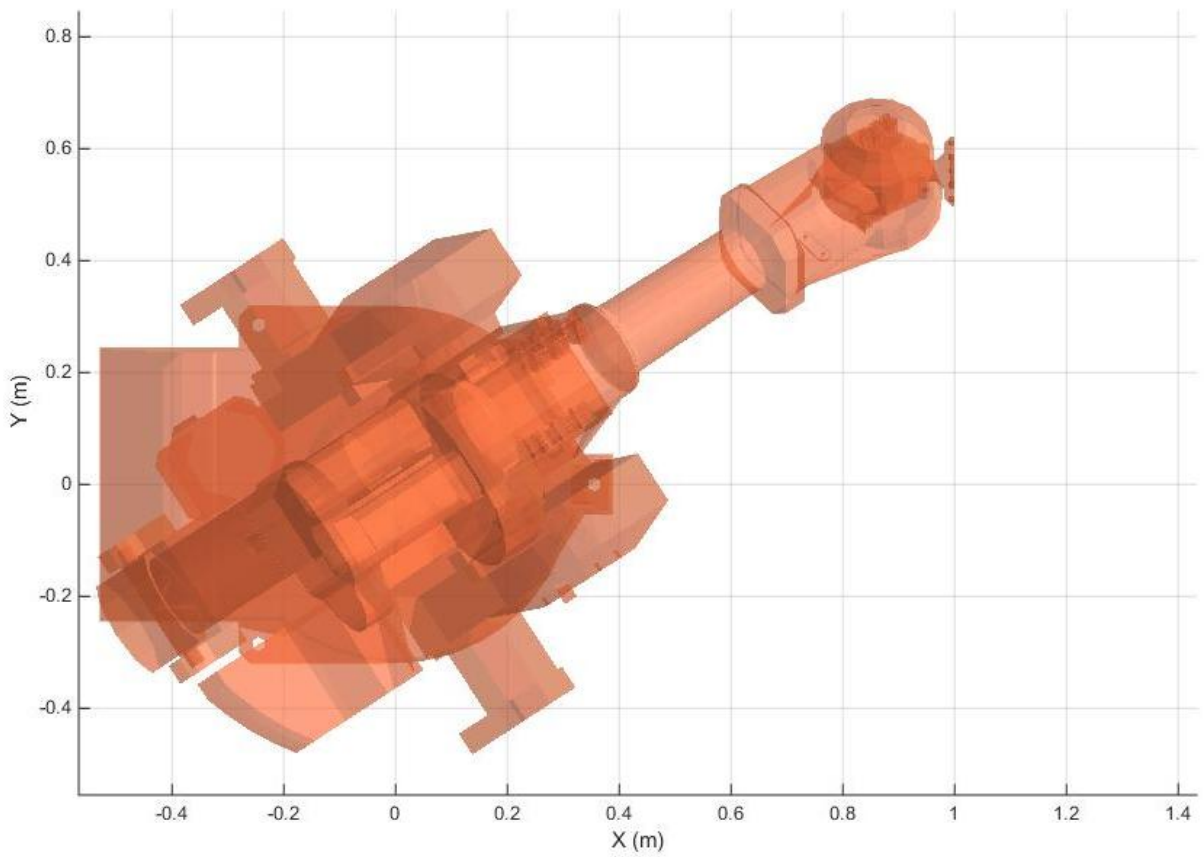


Рис. 3.5. Результат візуалізації розрахованої інверсної кінематики в пакеті MATLAB. Координата Y-X.

Як можна побачити за рис 3.5 та 3.6 робот виконав завдання та перемістився в задані координати, що підтверджує правильність розрахунку.

					<i>ЕП.МР.18.09.03.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

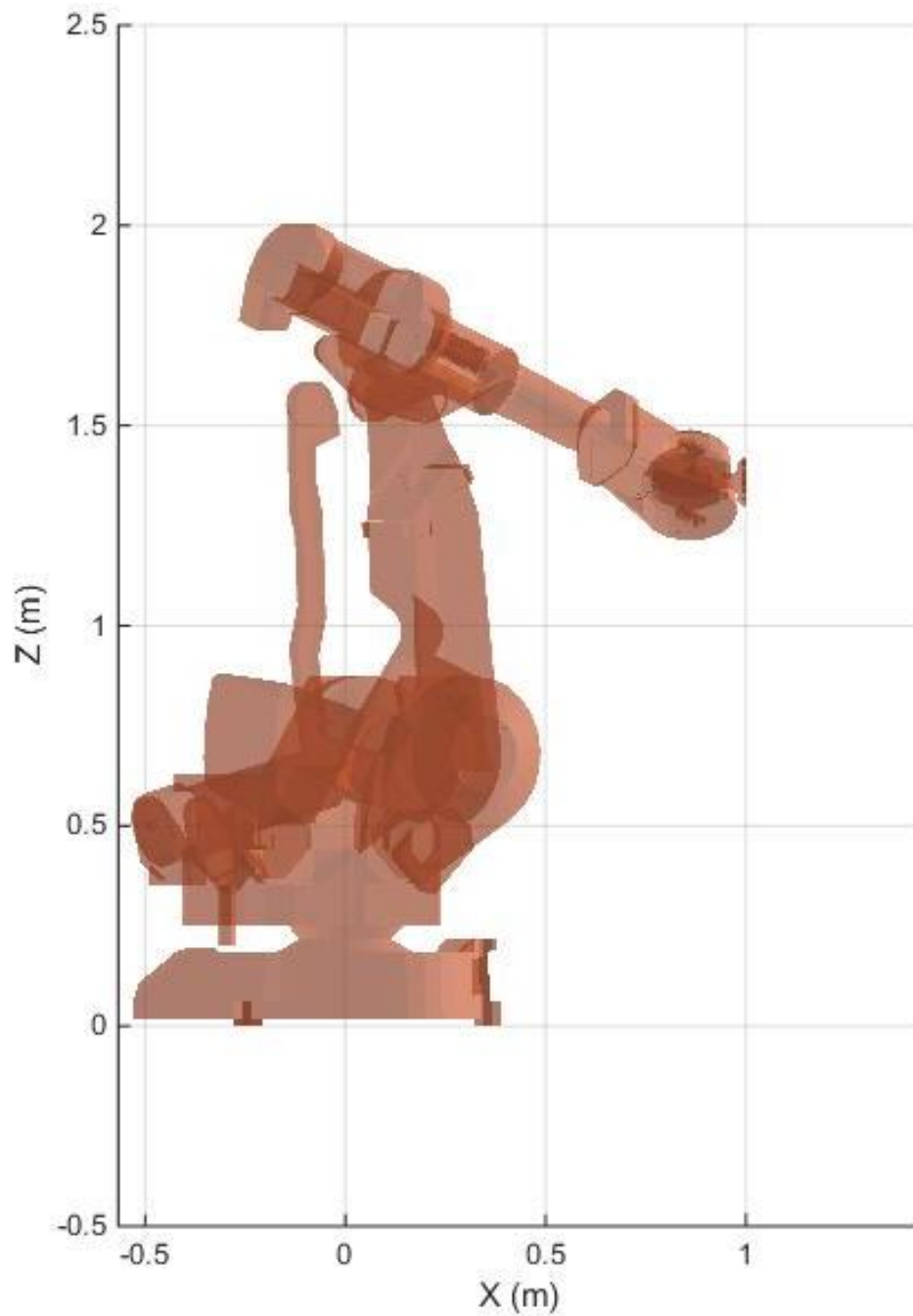


Рис. 3.5. Результат візуалізації розрахованої інверсної кінематики в пакеті
MATLAB. Координата Z-X.

3.2. Програмна реалізація системи розпізнавання об'єктів

Нижче наведена функція `inRangeImage` для формування бінарного зображення.

```
private Image<Gray, Byte> inRangeImage(Image<Hsv, Byte> hsvImage, int Lo,
int Hi, int con)
{
    Image<Gray, Byte> ResultImage = new Image<Gray,
Byte>(hsvImage.Width, hsvImage.Height); //конечный результат проверки
    Image<Gray, Byte> lLowCh = new Image<Gray,
Byte>(hsvImage.Width, hsvImage.Height, new Gray(Lo)); //нижний
предел(изобр где все пиксели = нижн пределу)
    Image<Gray, Byte> lHiCh = new Image<Gray,
Byte>(hsvImage.Width, hsvImage.Height, new Gray(Hi)); //верхний предел
    CvInvoke.cvInRange(hsvImage[con], lLowCh, lHiCh,
ResultImage); //Выполняет проверку диапазона для каждого элемента входного
массива
    return ResultImage;
}
```

Код функції «`cvAndHsvImage`» представлено нижче.

```
private Image<Gray, Byte> cvAndHsvImage(Image<Bgr, Byte> imgFame,
int L1, int H1, int L2, int H2, int L3, int H3, bool H, bool S, bool V,
bool I)
{
    Image<Hsv, Byte> hsvImage = imgFame.Convert<Hsv, Byte>();
    Image<Gray, Byte> ResultImage = new Image<Gray,
Byte>(hsvImage.Width, hsvImage.Height); // результат наложения
    Image<Gray, Byte> ResultImageH = new Image<Gray,
Byte>(hsvImage.Width, hsvImage.Height);
    Image<Gray, Byte> ResultImageS = new Image<Gray,
Byte>(hsvImage.Width, hsvImage.Height);
    Image<Gray, Byte> ResultImageV = new Image<Gray,
Byte>(hsvImage.Width, hsvImage.Height);
    Image<Gray, Byte> img1 = inRangeImage(hsvImage, L1, H1, 0); //
установка диапазона для H
    Image<Gray, Byte> img2 = inRangeImage(hsvImage, L2, H2, 1); //
установка диапазона для S
    Image<Gray, Byte> img3 = inRangeImage(hsvImage, L3, H3, 2); //
установка диапазона для V
    Image<Gray, Byte> img4 = inRangeImage(hsvImage, 0, L1, 0);
    Image<Gray, Byte> img5 = inRangeImage(hsvImage, H1, 180, 0);
    #region checkBox Color Mode
    // Формирование изображения в зависимости от выбранных каналов
для отображения
    if (H) // если выбран канал H
    {
        if (I) // если выбрана инверсия
        {
            CvInvoke.cvOr(img4, img5, img4, System.IntPtr.Zero);
        }
    }
}
```

					ЕП.МР.18.09.03.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

```

// выполняет побитовую дизъюнкцию двух массивов 4-го и 5-го
ResultImageH = img4;
    }
    else { ResultImageH = img1; }
}
if (S) ResultImageS = img2;// если выбран канал S
if (V) ResultImageV = img3;// если выбран канал V
// далее идут различные комбинации
if (H && !S && !V) ResultImage = ResultImageH;
if (!H && S && !V) ResultImage = ResultImageS;
if (!H && !S && V) ResultImage = ResultImageV;
if (H && S && !V)
{
    CvInvoke.cvAnd(ResultImageH, ResultImageS, ResultImageH,
System.IntPtr.Zero);
    ResultImage = ResultImageH;
}
if (H && !S && V)
{
    CvInvoke.cvAnd(ResultImageH, ResultImageV, ResultImageH,
System.IntPtr.Zero);
    ResultImage = ResultImageH;
}
if (!H && S && V)
{
    CvInvoke.cvAnd(ResultImageS, ResultImageV, ResultImageS,
System.IntPtr.Zero);
    ResultImage = ResultImageS;
}
if (H && S && V)
{
    CvInvoke.cvAnd(ResultImageH, ResultImageS, ResultImageH,
System.IntPtr.Zero);
    CvInvoke.cvAnd(ResultImageH, ResultImageV, ResultImageH,
System.IntPtr.Zero);
    ResultImage = ResultImageH;
}
}
#endregion
CvInvoke.cvErode(ResultImage, ResultImage, (IntPtr)null,
1);// размывает изображение
return ResultImage;
}

```

Нижче представлений код функції «RecDetection».

```

private void RecDetection(Image<Gray, Byte> img, Image<Bgr, Byte>
showRecOnImg, int areaV)
{
    Image<Gray, Byte> imgForContour = new Image<Gray,
byte>(img.Width, img.Height);
    CvInvoke.cvCopy(img, imgForContour, System.IntPtr.Zero);
    IntPtr storage = CvInvoke.cvCreateMemStorage(0);

```

					ЕП.МР.18.09.03.ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

```

IntPtr contour = new IntPtr();
    CvInvoke.cvFindContours(
        imgForContour,
        storage,
        ref contour,
System.Runtime.InteropServices.Marshal.SizeOf(typeof(MCvContour)),
        Emgu.CV.CvEnum.RETR_TYPE.CV_RETR_EXTERNAL,
        Emgu.CV.CvEnum.CHAIN_APPROX_METHOD.CV_CHAIN_APPROX_NONE,
        new Point(0, 0));
    Seq<Point> seq = new Seq<Point>(contour, null);
    for (; seq != null && seq.Ptr.ToInt64() != 0; seq =
seq.HNext)
    {
        Rectangle bndRec = CvInvoke.cvBoundingRect(seq, 2);//
        Создает ограничительный прямоугольник для набора 2Д точек
        double areaC = CvInvoke.cvContourArea(seq,
MCvSlice.WholeSeq, 1) * -1;
        if (areaC > areaV)
        {
            CvInvoke.cvRectangle(showRecOnImg, new
Point(bndRec.X, bndRec.Y),
                new Point(bndRec.X + bndRec.Width, bndRec.Y +
bndRec.Height),
                new MCvScalar(0, 0, 255), 2, LINE_TYPE.CV_AA, 0);
        }
    }
}

```

Для тестування роботи системи розпізнавання об'єктів використаємо розроблену програму та модель робота в пакеті MATLAB. Задамо траєкторію переміщення для робота маніпулятора за допомогою веб камери та прослідкуємо чи відповідає переміщення моделі робота в MATLAB переміщенню об'єкта керування в відео кадрі.

Налаштуємо систему на розпізнавання синього кольору, як показано на рис. 3.6

					ЕП.МР.18.09.03.ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

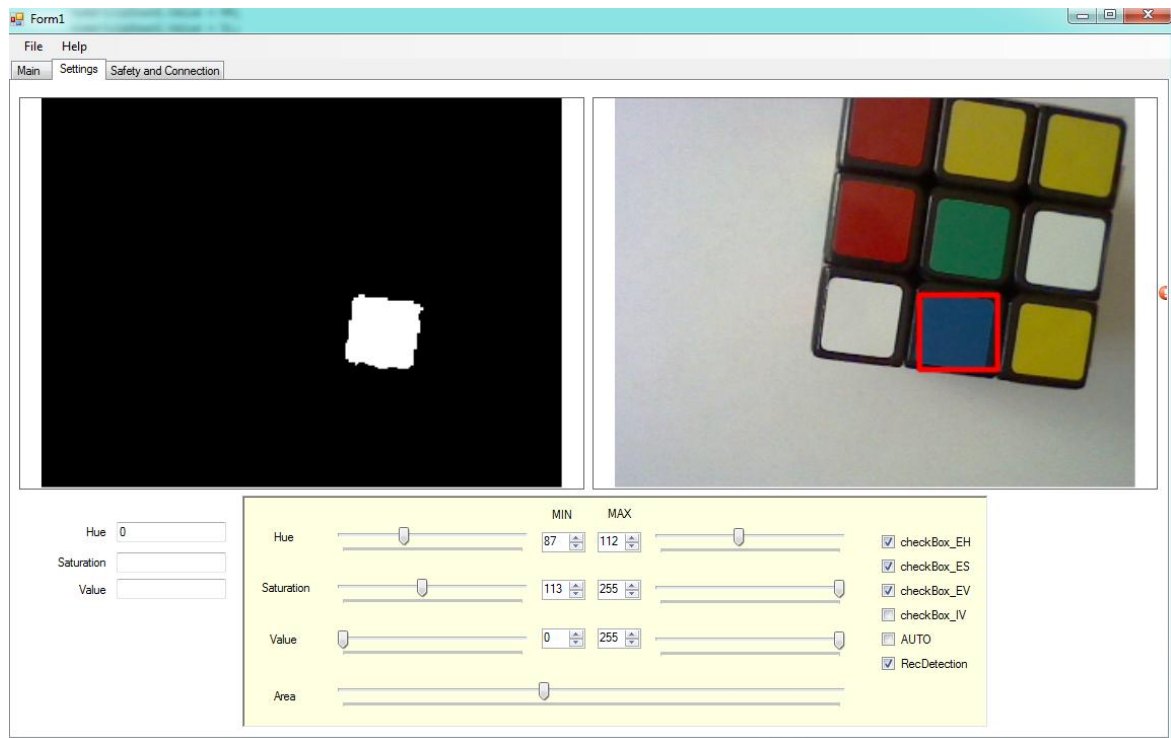


Рис. 3.6 – Інтерфейс розробленої програми

Після налаштування, задамо траєкторію переміщення рис. 3.7(малюнок зліва).

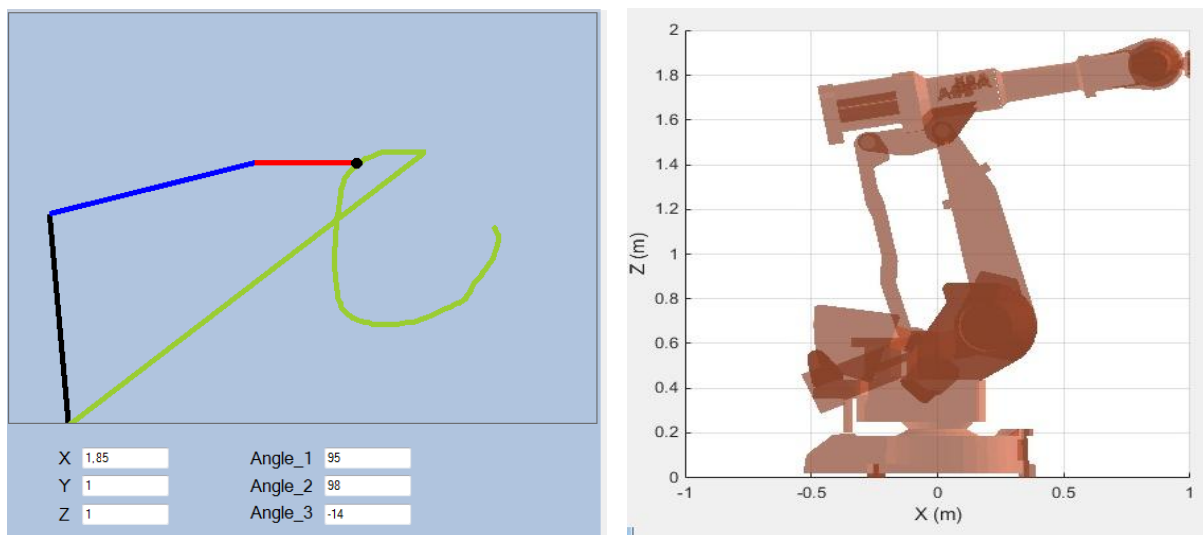


Рис 3.7 – Співставлення зображення двох моделей в точці 1

Для передачі отриманих координат на модель робота в MATLAB відмасштабуємо їх, щоб ті відповідали масштабу моделі робота. На рис 3.7. показані розміри ABB IRB 4400. Розміри відеокcadру становлять 557x387.

4. АДАПТАЦІЯ РОБОТА ДО РОБОТИ В КОЛАБОРАЦІЇ З ЛЮДИНОЮ

					<i>ЕП.МР.18.09.04.ПЗ</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Лелько В.М.</i>			<i>АДАПТАЦІЯ РОБОТА ДО РОБОТИ В КОЛАБОРАЦІЇ З ЛЮДИНОЮ</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>		<i>Худолій С.С.</i>					<i>70</i>	<i>94</i>
<i>Реценз.</i>						<i>НГУ</i>		
<i>Н. Контр.</i>		<i>Казачковський</i>						
<i>Затверд.</i>								

Розширення сфери застосування промислових роботів, їх зростаючий штучний інтелект, не зменшує актуальність проблем забезпечення безпеки праці. Тому далі буде розглянута система технічного зору, виділені основні причини виникнення аварійних ситуацій і вимоги, що пред'являються до безпеки праці. Також буде проведена оцінка шкідливих та небезпечних факторів та розробка заходів направлених на покращення умов людини при використанні запроєктованого маніпулятора.

Основними причинами, що формують небезпечні, критичні і аварійні ситуації при експлуатації робота маніпулятора, є:

- непередбачувані рухи виконавчих пристроїв промислових колаборативних роботів при налагодженні, ремонті, під час навчання і виконання керуючої програми;
- раптова відмова складових механізмів колаборативного робота або технологічного обладнання, спільно з яким він працює;

На проблему безпеки впливають такі аспекти: змінне число ступенів рухливості, автономний вибір напрямку та швидкостей.

Варто зазначити основні види небезпеки для здоров'я і життя обслуговуючого персоналу: механічні пошкодження, електротравми, шум та вібрації[10].

4.1. Механічні небезпеки

Механічні небезпеки при роботі маніпулятора є основними, які можуть завдати шкоди людині, оскільки в конструкції є рухомі частини. Для їх запобігання розроблені спеціальні заходи представлені в таблиці 4.1.

					ЕП.МР.18.09.04.ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.1 - Заходи та засоби попередження механічних небезпек

<i>Заходи</i>	<i>Засоби попередження небезпеки</i>
Технічні	Необхідно покрити усі рухомі частини маніпулятора м'якою обшивкою, яка могла б пом'якшити зіткнення виступивши демпфером. Треба використати різноманітні сенсорні системи в тому числі систем машинного зору.
Програмні	В систему програмування і керування необхідно інтегрувати інструменти діагностики.

В даному дипломному проекті розглянута система зниження ризику отримання травми людиною при зіткненні з маніпулятором, основана на машинному зорі. Основною задачею системи є розпізнавання обличчя людини, яка знаходиться в робочій області маніпулятора, та зниження швидкості переміщення маніпулятора. Також реалізована система розпізнавання зіткнення основана на використанні датчиків струму.

В подальшому представлено стислий огляд принципу функціонування реалізованих систем захисту.

4.1.1. Система машинного зору як основа безпеки при роботі з маніпулятором.

Основна ідея системи полягає в тому, щоб маніпулятор зрозумів, що в його робочій області знаходиться людина, та зреагував на це, а саме знизив швидкість руху. В даному випадку розпізнавання людини буде основане на розпізнаванні обличчя. Для реалізації пошуку обличчя в реальному часі будуть використанні функції, що інтегровані до бібліотеки EmguCV, а саме алгоритм пошуку оснований на методі Віоли-Джонса[11].

В загальному випадку алгоритм пошуку обличчя на цифровому зображенні виглядає наступним чином:

1. Отримуємо зображення з веб камери.
2. Конвертуємо кольорове зображення в відтінки сірого, що дозволить в подальшому збільшити втричі швидкість обробки зображення.

Скористаємось формулою:

$$I(x, y) = 0.2125R + 0.7154G + 0.0721B \quad (4.1)$$

3. Виконуємо приведення зображення до інтегральної форми. Інтегральне представлення зображення - це матриця, яка співпадає за розмірами з вихідним зображенням. В кожному її елементі зберігається сума інтенсивностей всіх пікселів, що знаходяться ліворуч і вище даного елемента (Рис4.1). Елементи матриці розраховуються за формулою:

$$S(x, y) = I(x, y) + S(x - 1, y) + S(x, y - 1) - S(x - 1, y - 1) \quad (4.1.2)$$

де,

S - результат попередніх ітерацій для даної позиції пікселя,

I – яскравість пікселя вхідного зображення.

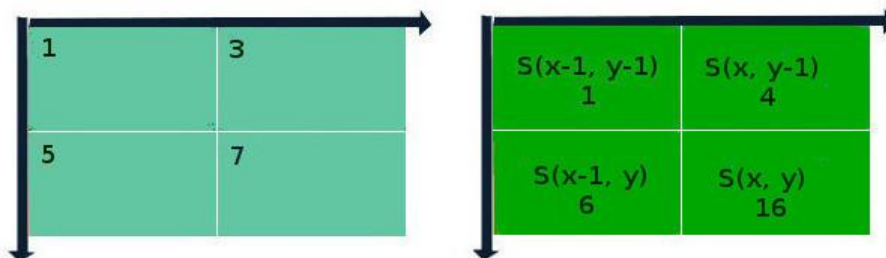


Рис.4.1 –Демонстрація формування інтегральної матриці зображення

Суть такого перетворення зображення полягає в тому, що створивши один раз інтегральну матрицю зображення, можна швидко вирахувати суму значень пікселів будь-якої прямокутної області в межах цього зображення. Що стане в нагоді в подальшому.

4. Виконуємо сканування зображення для знаходження елементів, що будуть свідчити про наявність обличчя людини. Перед описом алгоритму треба зауважити, що для верифікації зображення програма використовує заздалегідь навчений класифікатор, який був заздалегідь натренований на

розпізнавання обличь. Мається на увазі, що при аналізі зображення, ми маємо інтегральний опис вхідного зображення і еталонні ознаки, за якими воно ідентифікується. Ці еталонні ознаки (заздалегідь визначені критерії, за якими виконується пошук) були отримані завдяки виділенню спільних рис обличь на певній кількості фото(наприклад 500 фото) та згруповані в один XML документ(можна вважати базою даних).

В даному дипломному проекті не розглядаються принципи формування еталонних ознак для розпізнавання обличь.

Перейдемо до етапу порівняння класифікатором вхідного інтегрального зображення і певного еталону.

Вхідне зображення сканується вікном пошуку (при подальшому скануванні вікно пошуку збільшується), тобто сканується не одразу все, а розбивається на квадратні ділянки, які скануються по черзі. Розглянемо, що собою представляє сканування:

- Алгоритм сканування виконує пошук ознак Хаара. У стандартному методі Віюлі - Джонса використовуються прямокутні ознаки, зображені на малюнку 4.2, вони називаються примітивами Хаара.

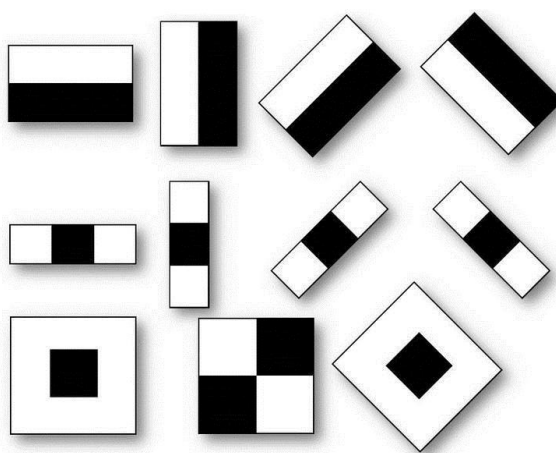


Рис. 4.2 - Примітиви Хаара.

- Якщо не вдаватися в деталі, то подальший алгоритм можна описати наступним чином. Примітиви Хаара накладаються на реальне зображення та виконується розрахунок ознак Хаара, підсумовуючи яскравість пікселів під світлою частиною примітиву і віднімаючи

суму тих, що під темною. Це являться мірою схожості тої частини зображення, що під примітивом, та самого примітива. В одному вікні пошуку може бути порівняно 200 000 тисяч примітивів. Далі результат порівняння формує вектор ознак. Щоб дізнатися чи є обличчя людини в вікні пошуку, треба відправити отриманий вектор ознак пошуку на класифікатор.

В підсумку можна зауважити, що така модель забезпечує високу якість вихідного результату, так як обличчя людей мають певні характерні риси що корелюються з класифікаторами Хаара. Наприклад, очі будуть темніші ніж область між ними, так само, як область рота буде темніше ніж лоб. Чим більше використовується різних примітивів, тим точніше можна потім класифікувати об'єкт.

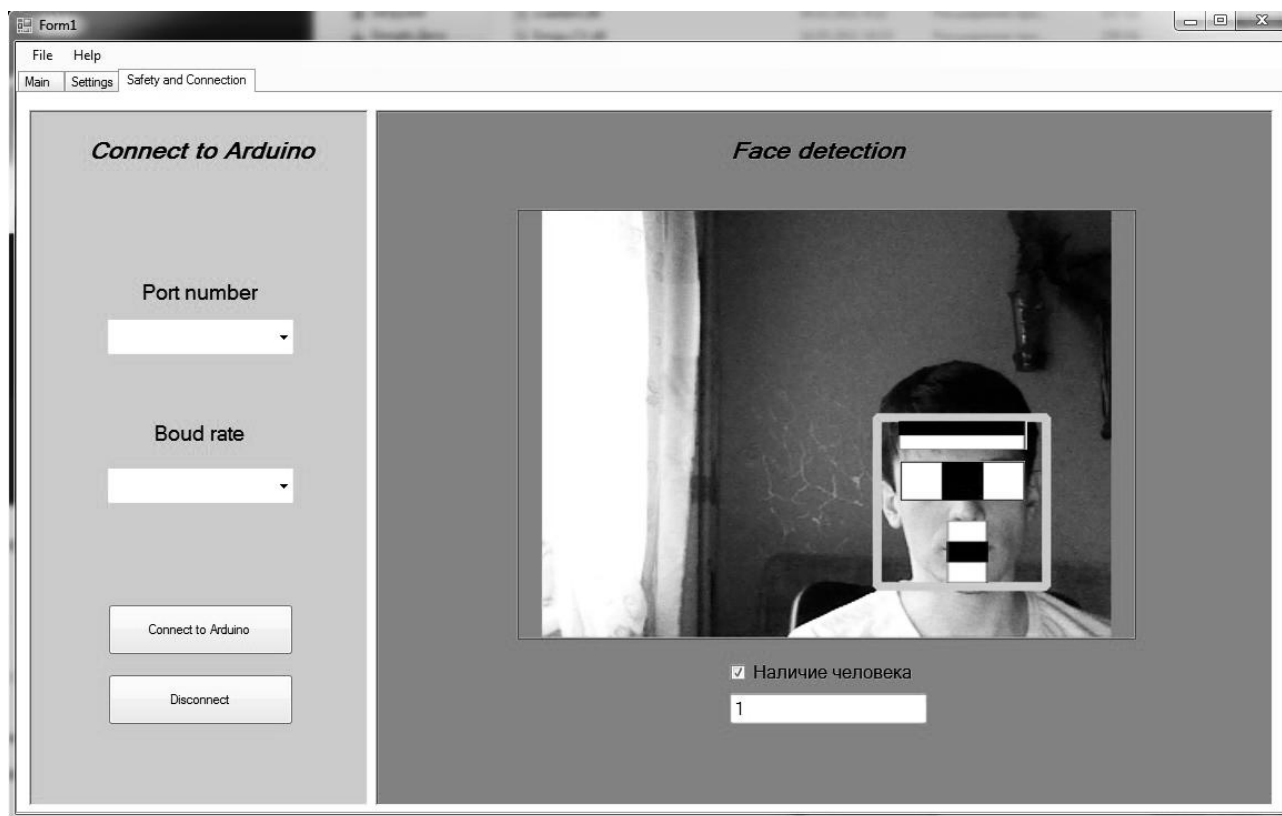


Рис. 4.3 – Знімок інтерфейсу розробленої програми, з демонстрацією застосування ознак Хаара

Програмний код:

```
private void timer1_Tick(object sender, EventArgs e)
{
    using (Image<Bgr, byte> nextFrame = cap.QueryFrame())
    {
        if (nextFrame != null)
        {
            // there's only one channel (greyscale), hence the zero index
            //var faces = nextFrame.DetectHaarCascade(haar)[0];
            Image<Gray, byte> grayframe = nextFrame.Convert<Gray, byte>();
            Image<Gray, Byte> gray = nextFrame.Convert<Gray, Byte>();
            var faces =
                grayframe.DetectHaarCascade(
                    haar, 1.2, -1,
                    HAAR_DETECTION_TYPE.DO_CANNY_PRUNING,
                    new Size(nextFrame.Width / 8, nextFrame.Height / 8)
                )[0];
            foreach (var face in faces)
            {
                nextFrame.Draw(face.rect, new Bgr(Color.LightGreen), 3);
            }
            pictureBox1.Image = nextFrame.ToBitmap();
        }
    }
}
private void Form1_Load(object sender, EventArgs e)
{
    cap = new Capture(0);
    // adjust path to find your xml
    haar = new HaarCascade("haarcascade_frontalface_alt2.xml");
    //haar = new HaarCascade("haarcascade_frontalface_alt_tree.xml");
}
```

Головну задачу виконує функція `grayframe.DetectHaarCascade()`, де

- `new Size(nextFrame.Width / 8, nextFrame.Height / 8)` – установка мінімального розміру вікна пошуку в відносних одиницях.
- `HAAR_DETECTION_TYPE.DO_CANNY_PRUNING` – активація функція оптимізації розрахунків .
- `haar, 1.2` – коефіцієнт масштабування вікна пошуку.

Далі функція `nextFrame.Draw(face.rect, new Bgr(Color.LightGreen), 3);`

Виділяє ділянку з обличчям на фінальному кадрі.

									Арк.
									76
Змн.	Арк.	№ докум.	Підпис	Дата					

4.1.2. Використання датчиків струму для визначення зіткнення з перешкодою

При зіткненні маніпулятора з перешкодою виникає різке підвищення струму. Якщо воно більше заданого значення, то потрібно зупинити маніпулятор.

Контроль струму буде виконуватись за допомогою датчиків(табл.4.2.) виконаних на базі мікросхеми ACS712ELC, яка в своєму складі містить дуже точний лінійний датчик Холла, інтегрований на кристал мікросхеми, і мідного провідника, розміщеного близько до кристалу. Електричний струм, протікаючи через провідник, створює магнітне поле, яке фіксується датчиком Холла і перетворюється в напругу, пропорційно значенню вхідного струму. Показання з датчиків знімаються за допомогою аналогових входів A0, A1, A2, A3 на платі Arduino(схема підключення показана на рис. 4.4 та на рис. 4.5).

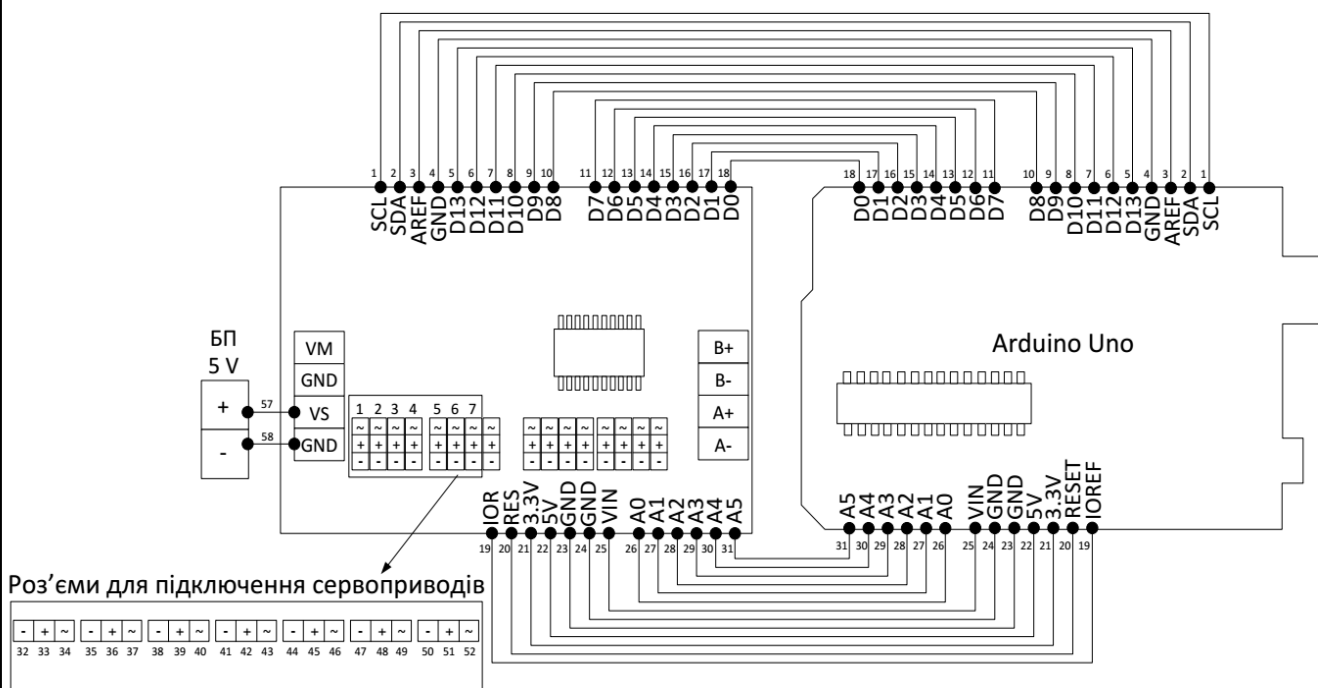


Рис.4.4.- Концептуальна схема підключення сервоприводів

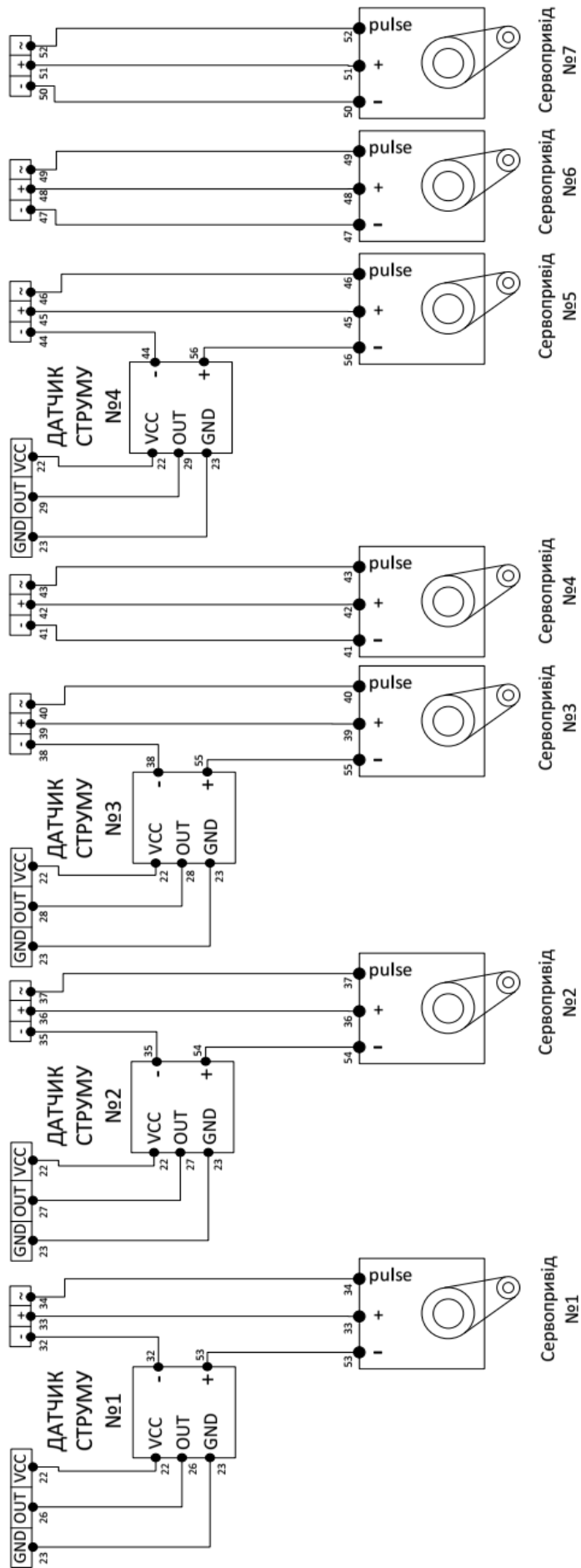


Рис.4.5.- Концептуальна схема підключення сервоприводів

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

Таблиця 4.2 - Характеристики датчиків струму

Датчик струму 20А	Напруга живлення	4.5V~5.5V DC
	Діапазон вимірюваного струму	-20 to +20 Amps
	Чутливість	100mV/A
Датчик струму 5А	Напруга живлення	4.5V~5.5V DC
	Діапазон вимірюваного струму	-5 to +5 Amps
	Чутливість	185mV/A

За допомогою датчика струму було експериментально встановлено граничне значення струму (табл. 4.3.).

Таблиця 4.3. - Характеристики сервоприводів

Рівень струм пускового	до 1.3 А
Рівень струму робочого	до 1 А
Рівень струму при зіткненні	1.3 А

Враховуючі показники рівня струму та динаміку його наростання та спадання, була розроблена програма для плати ARDUINO. Сама програма наведена в Додатку 2. Нижче представлений алгоритм роботи програми.

					ЕП.МР.18.09.04.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		79

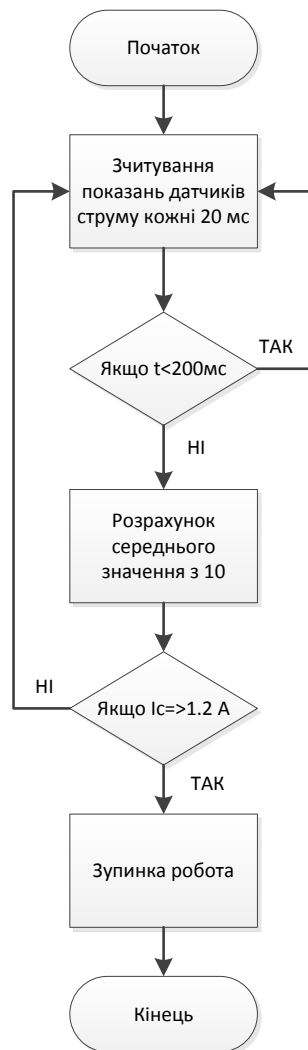


Рис.4.6. – Блок схема програми контролю струму сервоприводів

4.2. Небезпека враження людини електричним струмом при роботі з маніпулятором та ПК

В приміщення для живлення обладнання передбачається використання мережі змінного струму з напругою 220 В та частотою 50 Гц. В такому випадку основні джерела електробезпеки приведені у таблиці 4.4.

Таблиця 4.4. - Джерела та наслідки електробезпеки

<i>Джерело небезпеки</i>	<i>Можливі наслідки</i>
Пошкоджені кабелі	Опіки Смерть
Відсутність ізоляції	
Деталі, що знаходяться під напругою	

Для запобігання виникненню небезпечних ситуацій та ураження струмом пропонуються прийняти міри описані в таблиці 4.5.

Таблиця 4.5.- Заходи та засоби попередження електробезпеки

<i>Заходи</i>	<i>Засоби попередження небезпеки</i>
Технічні	<ul style="list-style-type: none"> • Перевірити, щоб усі системи та блоки живлення були оснащені запобіжниками; • Унеможливити контакт людини з деталями що знаходяться під напругою, розмістивши їх в недоступному, огороженому місці. • Електрокабелі повинні мати покриття стійке до впливу навколишнього середовища, роз'єми повинні бути герметичні. • Також необхідно, щоб приміщення, в якому буде розміщено обладнання, мало приховану проводку, також плюсом буде наявність диференційних пристроїв захисного відключення
Організаційні	<ul style="list-style-type: none"> • Ознайомити працівників з правилами технічної безпеки

Заходи індивідуальног о захисту	Не передбачено
---------------------------------------	----------------

4.3. Висновки

Заходи, розглянуті в розділі охорони праці, в першу чергу направлені на адаптацію робота до роботи в безпосередній близькості з людиною та враховують ті вимоги, які встановлені міжнародними стандартами ISO/TS 15066 (містить рекомендації про те, як забезпечити безпеку людей в умовах використання колаборативних роботів).

До колаборативного робота, як правило, висувають такі вимоги:

- Вони повинні бути оснащені датчиками, що обмежують зусилля і / або швидкість ланок, і, в залежності від застосування, можуть працювати в безпосередній близькості до людини без установки захисної огорожі.
- Матеріали, з яких виготовлений корпус робота, повинні бути досить м'якими для запобігання нанесення шкоди людині при зіткненні.

В нашому випадку була реалізована система, яка програмно та за рахунок датчиків, забезпечує безпеку людини при використанні колаборативного робота. Робота системи була перевірена в розділі моделювання і оцінюється як задовільна.

Треба зазначити, що рекомендації відносно пом'якшення корпусу робота не реалізовувалися.

					ЕП.МР.18.09.04.ПЗ	Арк.
						82
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В розглянутому дипломному проекті вирішена задача формування траєкторії колабораційним роботом маніпулятором за рахунок використання системи машинного зору.

Розглянуто існуючі робото-технічні рішення та проаналізовано тенденції подальшого розвитку галузі робототехніки. Сформульовано вимоги до колаборативних роботів. Проведено огляд систем машинного зору та методів розрахунку кінематики.

Виконано опис кінематичної схеми наявного робота маніпулятора. Згідно з кінематичною схемою проведено розрахунок інверсної кінематики. Розглянуто алгоритм розпізнавання зображення в кадрі за допомогою методу сегментації, що дозволило виділити об'єкт певного кольору, з одержуваного в реальному часі зображення.

Згідно з отриманими результатами в другому розділі та поставленими цілями було розроблене програмне забезпечення, яке дозволяє керувати роботом маніпулятором за допомогою веб камери. Було виконано моделювання роботи системи як на реальному роботі так і на віртуальному. В розділі 3 розглянуто моделювання роботи віртуального робота в симуляторі V-REP.

В роботі також проаналізовано найважливіші аспекти з охорони праці. Для уникнення можливості травмування при роботі в кооперації з роботом були реалізовані декілька програмних рішень, а саме зниження швидкості при появі в робочій зоні людини та використання датчиків струму для детектування зіткнення. Також розглянуті заходи направлені на нівелювання небезпеки враження людини електричним струмом.

Впровадження системи машинного зору та адаптація робота маніпулятора до роботи в кооперації з людиною відповідає всім сучасним тенденціям в галузі робототехніки та покликане спростити конструкцію колаборативного робота маніпулятора та підвищити його доступність за рахунок перекладання частини задач зі спеціалізованих систем на загальнодоступні, такі як ПК з веб камерою.

					<i>ЕП.МР.18.09.В.ПЗ</i>	Арк.
						83
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ПЕРЕЛІК ПОСИЛАНЬ

1. Arduino Uno [Електронний ресурс] – Режим доступу:
<http://arduino.ua/ru/hardware/Uno>, (дата звернення: 10.09.17).
2. Зенкевич С. Л., Ющенко А. С. Управление роботами. Основы управления манипуляционными роботами: Учебник для вузов – М.: Изд-во МГТУ им. Н. Э. Баумана, 2000. – 400 с.
3. Шахинпур М. Курс робототехники: Перевод с англ. – М.: Мир, 1990. – 527 с.
4. Фу К., Гонсалес Р., Ли К. Робототехника: Перевод с англ. – М.: Мир, 1989. – 624 с.
5. Цвіркун Л. І., Грулер Г. Робототехніка та мехатроніка: Навч. посібник. – Д.: Національний гірничий університет, 2007. – 216 с.2
6. EmguCV Tutorial [Електронний ресурс] – Режим доступу:
<http://www.emgu.com/wiki/index.php/Tutorial>, (дата звернення: 16.09.17).
7. Arbelaez P., Maire M., Fowlkes C., Malik J. Contour Detection and Hierarchical Image Segmentation // IEEE TPAMI. 2011. Vol. 33(5). Pp. 898–916.
8. Волосатова Т. М., Яблоков В. Е. Слежение за выделенными объектами на протяжении видеоряда: электронный научно технический журнал «Инженерный вестник»; ФГБОУ ВПО «МГТУ им. Н.Э. Баумана». Эл № ФС77-51036, 2015 г.,14 с.
9. HSV (цветовая модель) [Електронний ресурс] – Режим доступу:
[https://ru.wikipedia.org/wiki/HSV_\(цветовая_модель\)](https://ru.wikipedia.org/wiki/HSV_(цветовая_модель)), (дата звернення: 18.09.17).
10. Федотов А. Г., Поезжаева Е. В., Заглядов П. В., Безопасность труда при взаимодействии с промышленными роботами//Экология и научно-технический прогресс. Урбанистика,2014 г.
11. P. Viola and M. Jones. Robust real-time face detection. IJCV 57(2), 2004
12. ABB [Електронний ресурс] – Режим доступу:
<http://new.abb.com/products/robotics/industrial-robots/irb-4400/irb-4400-cad> ,

(дата звернення: 5.11.17).

ЕП.МР.18.09.П.ПЗ

Арк.

84

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

13. ARTE [Электронный ресурс] – Режим доступа:

http://arvc.umh.es/arte/index_en.html#introduction (дата звернення: 7.11.17).

14. Аналитическое исследование: Мировой рынок робототехники.

Национальная Ассоциация Участников Рынка Робототехники 2016 г.

					<i>ЕП.МР.18.09.П.ПЗ</i>	Арк.
						85
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ДОДАТКИ

Додаток А

Код програми розробленої для плати Arduino

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();

#define SERVOMIN 150 // this is the 'minimum' pulse length count (out of 4096)
#define SERVOMAX 610 // this is the 'maximum' pulse length count (out of 4096)

int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin
int potpin1 = 1; // analog pin used to connect the potentiometer
int val1; // variable to read the value from the analog pin

void setup() {
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
  pwm.begin();
  pwm.setPWMPFreq(60); // Analog servos run at ~60 Hz updates
  yield();
  pwm.setPWM(0, 0, map(90, 0, 180, SERVOMIN, SERVOMAX));
  pwm.setPWM(1, 0, map(90, 0, 180, SERVOMIN, SERVOMAX));
  pwm.setPWM(2, 0, map(90, 0, 180, SERVOMIN, SERVOMAX));
  pwm.setPWM(3, 0, map(90, 0, 180, SERVOMIN, SERVOMAX));
  pwm.setPWM(4, 0, map(90, 0, 180, SERVOMIN, SERVOMAX));
  pwm.setPWM(5, 0, map(90, 0, 180, SERVOMIN, SERVOMAX));
  pwm.setPWM(6, 0, map(90, 0, 180, SERVOMIN, SERVOMAX));
}

int Str[] = {90, 90, 90, 90, 90, 90, 90};
bool ERR = false;
/*****/
void loop()
{
  val = analogRead(potpin)/5.8;
  val1 = analogRead(potpin1)/5.8;
  val2 = analogRead(potpin1)/5.8;
  val3 = analogRead(potpin1)/5.8;
  if (val1>90 || val>90)
  {
    ERR = true;
  }if (val1<90 and val<90 and val2<90 and val3<90)
  {
    ERR = false;
  }
  if(ERR == false)
  {
    if (Serial.available() > 7)
```

									Арк.
									86
Змн.	Арк.	№ докум.	Підпис	Дата					

ЕП.МР.18.09.Д.ПЗ

```

{
  for (int i = 0; i < 3; i++)
  {
    Str[i] = Serial.parseInt();
    Serial.print("I received: ");
    Serial.println(Str[i]);
  }
}

/*****/
pwm.setPWM(1, 0, map(Str[0], 0, 180, SERVOMIN, SERVOMAX));
pwm.setPWM(2, 0, map(Str[1], 0, 180, SERVOMIN, SERVOMAX));
pwm.setPWM(3, 0, map(Str[2], 0, 180, SERVOMIN, SERVOMAX));
pwm.setPWM(4, 0, map(Str[3], 0, 180, SERVOMIN, SERVOMAX));
pwm.setPWM(5, 0, map(Str[4], 0, 180, SERVOMIN, SERVOMAX));
pwm.setPWM(6, 0, map(Str[5], 0, 180, SERVOMIN, SERVOMAX));
pwm.setPWM(7, 0, map(Str[6], 0, 180, SERVOMIN, SERVOMAX));
}
}

```

					<i>ЕП.МР.18.09.Д.ПЗ</i>	Арк.
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		87

Код програми розробленої в Visual Studio за виключенням функцій
використаних в основній частині дипломного проекту

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.IO;
using Microsoft.Speech.Recognition;
using Emgu.CV;
using Emgu.Util;
using Emgu.CV.Structure;
using Emgu.CV.CvEnum;

namespace Face_Detection_Concept
{
    public partial class Form1 : Form
    {
        //Для розпознавання
        bool ib3C;
        int ib3C_HV, ib3C_SV, ib3C_VV, ib3C_Lable_X, ib3C_Lable_Y, HH, HL, SH, SL, VH, VL;
        private Capture cap;
        private HaarCascade haar;
        //Для задання пути
        int play = 0;
        int tms = 0;
        const int size = 100;
        int X22 = 0;
        int Y22 = 0;
        //Начальные значения
        int X = 0, X1 = 0, X2 = 0, Xa = 0, Xb = 0, Xc = 0;
        int Y = 0, Y1 = 0, Y2 = 0, Ya = 0, Yb = 0, Yc = 0;
        int Z = 0;
        //Длина звеньев
        double L1 = 200;
        double L2 = 200;
        double L3 = 100;

        public Form1()
        {
            InitializeComponent();
            Arduibo_connect();
        }
    }
}

```

										Арк.
										88
Змн.	Арк.	№ докум.	Підпис	Дата	ЕП.МР.18.09.Д.ПЗ					

```

private void pictureBox2_MouseClick(object sender, MouseEventArgs e)// функция для
автоматического определения цвета
{
using (Image<Bgr, byte> nextFrame = cap.QueryFrame())
{
Image<Hsv, Byte> imgHsv = nextFrame.Convert<Hsv, Byte>();
Hsv hsv = imgHsv[e.Y, e.X];
if (!ib3C)
{
ib3C_HV = Convert.ToInt32(hsv.Hue);
ib3C_SV = Convert.ToInt32(hsv.Satuation);
ib3C_VV = Convert.ToInt32(hsv.Value);
ib3C_Lable_X = pictureBox2.Location.X + e.X;
ib3C_Lable_Y = pictureBox2.Location.Y + e.Y;
label7.Location = new Point(ib3C_Lable_X, ib3C_Lable_Y);
label7.Show();
}
}
textBox1.Text = Convert.ToString(ib3C_HV);
textBox5.Text = Convert.ToString(ib3C_SV);
textBox6.Text = Convert.ToString(ib3C_VV);
}
private void timer1_Tick(object sender, EventArgs e)// основная функция
private void Form1_Load(object sender, EventArgs e)
{
cap = new Capture(0);//Определение источника изображения
haar = new HaarCascade("haarcascade_frontalface_alt2.xml");
}
private Image<Gray, Byte> cvAndHsvImage(Image<Bgr, Byte> imgFame, int L1, int H1, int
L2, int H2, int L3, int H3, bool H, bool S, bool V, bool I)

private Image<Gray, Byte> inRangeImage(Image<Hsv, Byte> hsvImage, int Lo, int Hi, int
con)

private void RecDetection(Image<Gray, Byte> img, Image<Bgr, Byte> showRecOnImg, int
areaV)

public void Main_Calculation(int X, int Y, int X1, int Z)
{
double[] XY = new double[4];
XY = Three_DOF(L1, L2, L3, X, Y, 0);
try
{
pictureBox3.Invalidate();
// Пересчет координат по полученным углам
Xa = Convert.ToInt16(L1 * Math.Cos(XY[1]));
Ya = Convert.ToInt16(L1 * Math.Sin(XY[1]));
//L1*cos(Q1) + L2*cos(Q1+Q2)
Xb = Convert.ToInt16(L1 * Math.Cos(XY[1]) + L2 * Math.Cos(XY[1] + XY[2]));
Yb = Convert.ToInt16(L1 * Math.Sin(XY[1]) + L2 * Math.Sin(XY[1] + XY[2]));
//L1*cos(Q1) + L2*cos(Q1+Q2) + L3*cos(Q1+Q2+Q3)
Xc = Convert.ToInt16(L1 * Math.Cos(XY[1]) + L2 * Math.Cos(XY[1] + XY[2]) + L3 *
Math.Cos(XY[1] + XY[2] + XY[3]));
}
}
}

```

						<i>ЕП.МР.18.09.Д.ПЗ</i>	Арк.
							89
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			

```
Yc = Convert.ToInt16(L1 * Math.Sin(XY[1]) + L2 * Math.Sin(XY[1] + XY[2]) + L3 *  
Math.Sin(XY[1] + XY[2] + XY[3]));
```

```
textBox2.Text = Convert.ToString(360 + XY[1] * 180 / Math.PI);
```

```
textBox7.Text = Convert.ToString(XY[2] * 180 / Math.PI - 180);
```

```
textBox8.Text = Convert.ToString(XY[3] * 180 / Math.PI);
```

```
}
```

```
catch
```

```
{
```

```
    errorProvider1.SetError(pictureBox2, "Выход за пределы рабочей области");
```

```
}
```

```
try
```

```
{
```

```
    serialPort1.Write(" " + textBox2.Text);
```

```
    errorProvider1.Clear();
```

```
}
```

```
catch
```

```
{
```

```
    errorProvider1.SetError(pictureBox2, "Нет подключения к ардуино");
```

```
}
```

```
double alpha1 = Math.Atan(Z * 1.0 / X1 * 1.0);
```

```
X2 = Convert.ToInt32(Math.Cos(alpha1) * Xc);
```

```
Y2 = Convert.ToInt32(Math.Sin(alpha1) * Xc);
```

```
int ard = Convert.ToInt16(alpha1 * 180 / Math.PI);
```

```
textBox4.Text = Convert.ToString(Z); // значение по Z координате
```

```
}
```

```
private void pictureBox3_Paint(object sender, PaintEventArgs e) // X Y
```

```
{
```

```
    var h = e.Graphics;
```

```
    h.TranslateTransform((float)pictureBox3.Width/10, (float)pictureBox3.Height);
```

```
    Pen pen = new Pen(Color.Black, 5);
```

```
    Pen pen1 = new Pen(Color.Blue, 5);
```

```
    Pen pen2 = new Pen(Color.Red, 5);
```

```
    e.Graphics.DrawLine(pen, new Point(0, 0), new Point(Xa, -Ya));
```

```
    e.Graphics.DrawLine(pen1, new Point(Xa, -Ya), new Point(Xb, -Yb));
```

```
    e.Graphics.DrawLine(pen2, new Point(Xb, -Yb), new Point(Xc, -Yc));
```

```
    Pen pen5 = new Pen(Color.YellowGreen, 5);
```

```
    if (checkBox2.Checked == true)
```

```
    {
```

```
        List<Point> pointList = new List<Point>();
```

```
        for (int index = 0; index < size; index++)
```

```
        {
```

```
            pointList.Add(new Point(arrX[index], -arrY[index]));
```

```
        }
```

```
        Point[] pointArray = pointList.ToArray();
```

```
        e.Graphics.DrawLines(pen5, pointArray.ToArray()); //or DrawCurve(pen,
```

```
pointArray.ToArray(), 1F);
```

```
    }
```

```
}
```

```
private void pictureBox4_Paint(object sender, PaintEventArgs e) // Z
```

```
{
```

									Арк.
									90
Эмн.	Арк.	№ докум.	Підпис	Дата					

ЕП.МР.18.09.Д.ПЗ

```

var g = e.Graphics;
g.TranslateTransform((float)0, (float)pictureBox4.Height/2);
Pen pen3 = new Pen(Color.YellowGreen, 5);
e.Graphics.DrawLine(pen3, new Point(0, 0), new Point(X1, -Z));
}
#region Arduino connection

public void Arduibo_connect()
{
    //получаем список всех доступных ком портов в системе
    string[] ports = SerialPort.GetPortNames();
    //Выводим список в комбобокс
    comboBox1.Items.AddRange(ports);
}
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    // При закрытии программы, закрываем порт
    if (serialPort1.IsOpen) serialPort1.Close();
}
private void button1_Click(object sender, EventArgs e)//Arduibo_connect_button
{
    //задаем скорость и имя
    serialPort1.PortName = comboBox1.Text;
    serialPort1.BaudRate = Convert.ToInt32(comboBox2.Text);
    //открываем порт
    serialPort1.Open();
}
private void button2_Click(object sender, EventArgs e)//Arduibo_disconnect_button
{
    serialPort1.Close();
}
#endregion

#region Path Record
int i = 0, k = 0, v = 0;
int[] arrX = new int[1000];
int[] arrY = new int[1000];

private void timer2_Tick(object sender, EventArgs e)
{
    if (tms == 1 && play == 0)
    {
        i++;
        v = i;
        textBox1.Text = Convert.ToString(i);

        Point[] points = Enumerable.Range(0, 1000).Select(x => new Point { X = X, Y = Y
    }).ToArray();
        arrX[i] = points[i].X;
        arrY[i] = points[i].Y;
    }
}

```

										Арк.
										91
Эмн.	Арк.	№ докум.	Підпис	Дата						

ЕП.МР.18.09.Д.ПЗ

```

        richTextBox1.Text = Convert.ToString(points[i].X) + " " + Convert.ToString(points[i].Y) +
"\n" + richTextBox1.Text;
    }
    if (tms == 0 && play == 1)
    {
        if (v == k)
        {
            k = 0;
        }
        k++;
        textBox1.Text = Convert.ToString(k);
        X22 = arrX[k];
        Y22 = arrY[k];
        richTextBox2.Text = Convert.ToString(X2) + " " + Convert.ToString(Y2) + "\n" +
richTextBox2.Text;
        Main_Calculation(X22, Y22, 1, 1);
        pictureBox3.Invalidate();
        pictureBox4.Invalidate();
    }
}
private void button5_Reset_Click(object sender, EventArgs e)
{
}
private void button3_Rec_Click(object sender, EventArgs e)
{
    tms = 1;
    play = 0;
    timer2.Enabled = true;
}
private void button4_LaunRob_Click(object sender, EventArgs e)
{
    play = 1;
    tms = 0;
    i = 0;
    timer2.Enabled = true;
}
private void button6_Stop_Click(object sender, EventArgs e)
{
    timer2.Enabled = false;
    pictureBox3.Invalidate();
}
#endregion
}

```

					<i>ЕП.МР.18.09.Д.ПЗ</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		92

ВІДГУКИ ТА РЕЦЕНЗІЇ

