

**Project Robot / manipulator design with 5 degrees of freedom**

**Hop Nguyen**

**404839**

**Group:03**

**Table of Contents**

[**1.State of art** 3](#_Toc93514740)

[**2.Mechanism** 5](#_Toc93514741)

[**3.FOWARD KINEMATICS** 6](#_Toc93514742)

[3.1 Mechanism 6](#_Toc93514743)

[3.2 Kinematic table 6](#_Toc93514744)

[3.3 Obtained Results 6](#_Toc93514745)

[**4.Inverse Kinematic** 9](#_Toc93514746)

[4.1 MAPLE CODE AND POSSIBLE SOLUTIONS FOR GIVEN POSITON 9](#_Toc93514747)

[4.2 THE SOLUTION FOR 10](#_Toc93514748)

[**5.Simple control** 11](#_Toc93514749)

[5.1 ROS and V-REP 11](#_Toc93514750)

[5.2 Arduino with Servos 12](#_Toc93514751)

[**6.Model** 13](#_Toc93514752)

[**7.APPENDIX** 14](#_Toc93514753)

[7.1 MATLAB CODE 14](#_Toc93514754)

[7.2 ROSNODE CODE 15](#_Toc93514755)

[7.3 ARDUINO CODE 18](#_Toc93514756)

# **1.State of art**

In this project I will construct robot arm (T050000)

One of the most common and typical 5DOF robotic arms in the factory today. This mechanism is flexible and can be combined with automation to quickly and easily complete work that requires 2-3 people to complete.

With the machines' ability to lift heavy objects, they are used in auto plants, manufacturing materials or freight.

The construction of the robot can be adapted to be placed on a driving system based on linear guides and a high precision toothed bar drive. Thanks to this, the robot gains an additional axis of movement along which it can move with the load at almost any distance

A picture containing snow, automaton

Description automatically generatedA picture containing indoor

Description automatically generatedI have tried to find some similar mechanism in the factories and laboratories:

**Diagram

Description automatically generated**

# **2.Mechanism**

A picture containing tool

Description automatically generated

**Diagram

Description automatically generated**Drawings of construction:

Degrees of freedom:

5 elements.

5 pair of class 1.

W= 6\*5 - 5\*5 = 5 degrees.

-Manipulators having three rotary joints are included in the group of anthropomorphic manipulators.

**A picture containing icon

Description automatically generated**

# **3.FOWARD KINEMATICS**

## 3.1 Mechanism

**Chart, line chart

Description automatically generated**

## 

## 3.2 Kinematic table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| System |  |  |  |  |
| 1 | 0 | 0 |  |  |
| 2 |  | 0 |  | 0 |
| 3 |  | 0 |  | 0 |
| 4 |  | 0 |  | 0 |
| 5 | 90 | 0 | 0 | 90 |
| 6 |  |  | 0 | 0 |

## 3.3 Obtained Results

A1 =

[1, 0, 0, a1]

[0, cos(alpha1), -sin(alpha1), 0]

[0, sin(alpha1), cos(alpha1), 0]

[0, 0, 0, 1]

A2 =

[cos(theta2), -sin(theta2), 0, a2\*cos(theta2)]

[sin(theta2), cos(theta2), 0, a2\*sin(theta2)]

[ 0, 0, 1, 0]

[ 0, 0, 0, 1]

A3 =

[cos(theta3), -sin(theta3), 0, a3\*cos(theta3)]

[sin(theta3), cos(theta3), 0, a3\*sin(theta3)]

[ 0, 0, 1, 0]

[ 0, 0, 0, 1]

A4 =

[cos(theta4), -sin(theta4), 0, a3\*cos(theta4)]

[sin(theta4), cos(theta4), 0, a3\*sin(theta4)]

[ 0, 0, 1, 0]

[ 0, 0, 0, 1]

A5 =

[0 0 1 0]

[1 0 0 0]

[0 1 0 0]

[0 0 0 1]

A6 =

[cos(theta6), -sin(theta6), 0, 0]

[sin(theta6), cos(theta6), 0, 0]

[ 0, 0, 1, d6]

[ 0, 0, 0, 1]

T06 =

[-sin(theta2+theta3+theta4)\*cos(theta6), sin(theta2+theta3+theta4)\*sin(theta6), cos(theta2+theta3+theta4), a1+a3\*cos(theta2+theta3)+a2\*cos(theta2)+a3\*cos(theta2+theta3+theta4)+ d6\*cos(theta2+theta3+theta4)]

[cos(theta2+theta3+theta4)\*cos(alpha1)\*cos(theta6)-sin(alpha1)\*sin(theta6), -sin(alpha1)\*cos(theta6)-cos(theta2+theta3 +theta4)\*cos(alpha1)\*sin(theta6), sin(theta2+theta3+theta4)\*cos(alpha1), cos(alpha1)\*(a3\*sin(theta2+theta3)+a2\*sin(theta2)+a3\*sin(theta2+theta3+theta )+d6\*sin(theta2+theta3+theta4))]

[cos(alpha1)\*sin(theta6)+cos(theta2+theta3+theta4)\*sin(alpha1)\*cos(theta6), cos(alpha1)\*cos(theta6)-cos(theta2+theta3+theta4)\*sin(alpha1)\*sin(theta6), sin(theta2+theta3+theta4)\*sin(alpha1), sin(alpha1)\*(a3\*sin(theta2+theta3)+a2\*sin(theta2)+a3\*sin(theta2+theta3+theta4)+d6\*sin(theta2+theta3+theta4))]

[0, 0, 0, 1]

# **4.Inverse Kinematic**

For calculation of forward kinematic problem, I used a MAPLE software.

For my calculation I took equations from the matrix to determine and because this matrix reduce calculations and the matrix determines only the orientation of the end point (and moves the axis in z-axis but this change can be also given before rotation in or just after the matrix without any consequences).

## 4.1 MAPLE CODE AND POSSIBLE SOLUTIONS FOR GIVEN POSITON

Text

Description automatically generated

Text, letter

Description automatically generated

Software solves equations and shows all solutions (a lot of). I show some solutions:

*{ = 0., = 0., = 0., = 2.21}, { = 0., = 1.57, = 1.57, = -2.50},*

*{= 3.14., = 2.65, = 0., = -2.21}, {= 0., = 3.14, = 2.83, = 2.06},*

*{= 3.14, = 3.14, = 0.614, = 2.05}, {= 0., = -1.57, = 1.71, = = 1.84}*

To make sure all solutions and code are right. I try calculating again with MATLAB:

clear all

alpha1 = 0., theta2 = 0., theta3 = 0., theta4 = 2.21;

a1=1; a2=1;a3=3;a4=2;d6=2;

x=a1+a3\*cos(theta2+theta3)+a2\*cos(theta2)+a3\*cos(theta2+theta3+theta4)+d6\*cos(theta2+theta3+theta4)

y=cos(alpha1)\*(a3\*sin(theta2+theta3)+a2\*sin(theta2)+a3\*sin(theta2+theta3+theta4)+d6\*sin(theta2+theta3+theta4))

z=sin(alpha1)\*(a3\*sin(theta2+theta3)+a2\*sin(theta2)+a3\*sin(theta2+theta3+theta4)+d6\*sin(theta2+theta3+theta4))

*Text

Description automatically generated with medium confidence A picture containing text

Description automatically generated Text

Description automatically generated with low confidence*

As we can see that the results are good.

## 4.2 THE SOLUTION FOR

ω = +

= ω -

ω – desired orientation of wheel in z-axis

From results we should remove ones with imaginary parts. As we can see the arm sometimes is able to obtain more than one result.

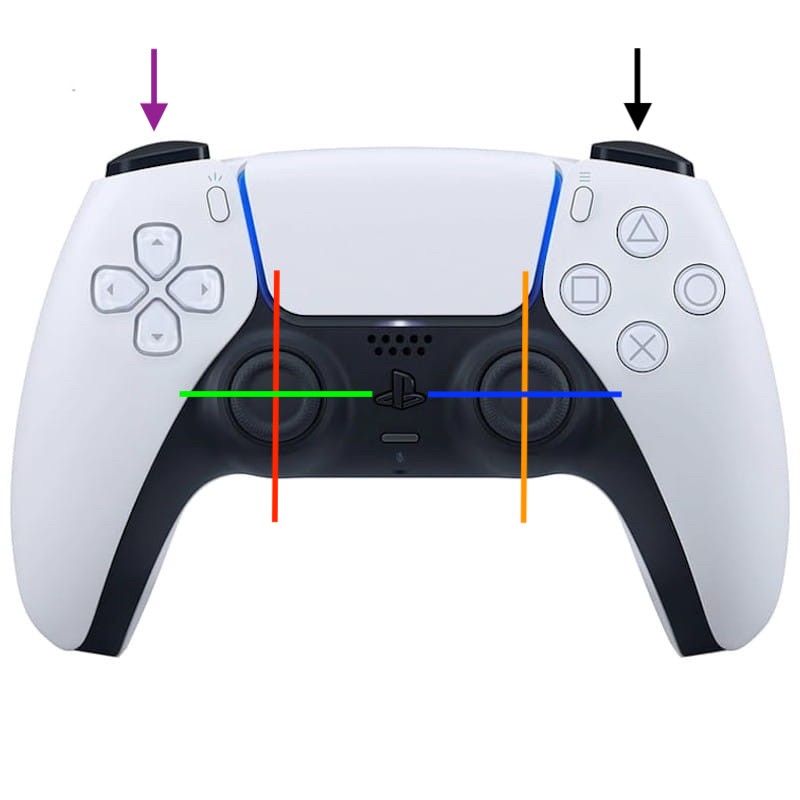
# **5.Simple control**

## 5.1 ROS and V-REP

In laboratory classes I have also made a simple control in ROS and V-REP program.

I control the arm via Sony PS5 Pad Controller.

**![Shape, arrow

Description automatically generated](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAANkAAAAsCAMAAADfP+3uAAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAABdUExURf////zc3fm/wfrKzPJcYu85QO0cJIlBQ4BER5RITPJiaFdXV21tbY8+QmJQUHBKTPemqWhoaGdnZ/39/WNPT1ZUVHFJTPenqpg7P7guNPJnbfaXm/zh4v73+AAAAEe6aOMAAAAfdFJOU////////////////////////////////////////wDNGXYQAAAACXBIWXMAACHVAAAh1QEEnLSdAAAAbElEQVRoQ+3PywqAIBBG4bGbJV1AEo3q/V+zzTyA7VTOx2wHzi8AAAAAAABok+kq0WtwrmGcamBnu3SanMcFv276XbJd5IjanMd5kRqWne0uC+HnsphSuvS7bO7W5FyPeU0Vp70AAAAAgFKJfGJAHaAcglMcAAAAAElFTkSuQmCC)**1st joint control axis

Icon

Description automatically generated 2nd joint control axis

 3rd joint control axis

 4th joint control axis

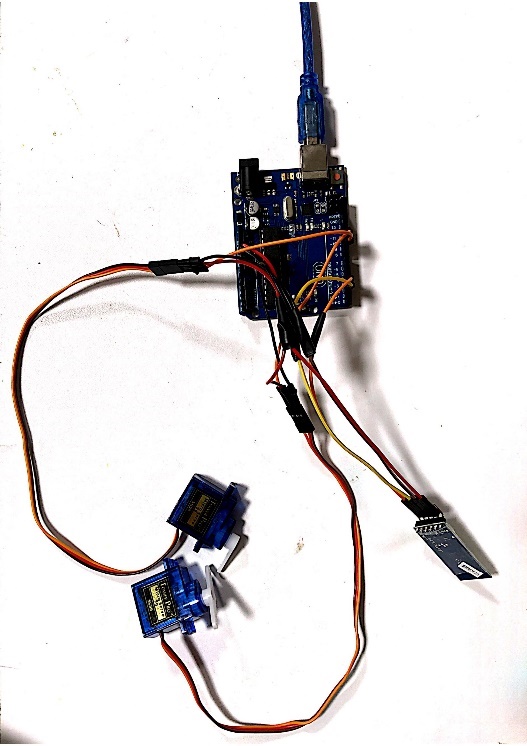
 5th joint control axis

To control the simulation in V-REP I had to create a node thanks which it would be possible. The code of the node is included in Appendix 2.

A picture containing chart

Description automatically generated

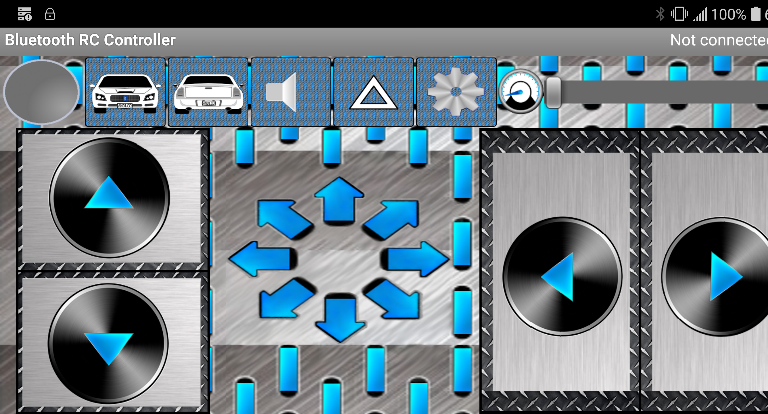
## 5.2 Arduino with Servos

To control Robot. I use Arduino UNO R3, Module Bluetooth HC-05 with Servos.

Connect Module Bluetooth with phone via app Bluetooth RC Controller.

Diagram, schematic

Description automatically generated**Scheme**



With this base we can connect to many servos to control. Due to the limited port of the Arduino, the Arduino can control up to 6 servos at the same time.

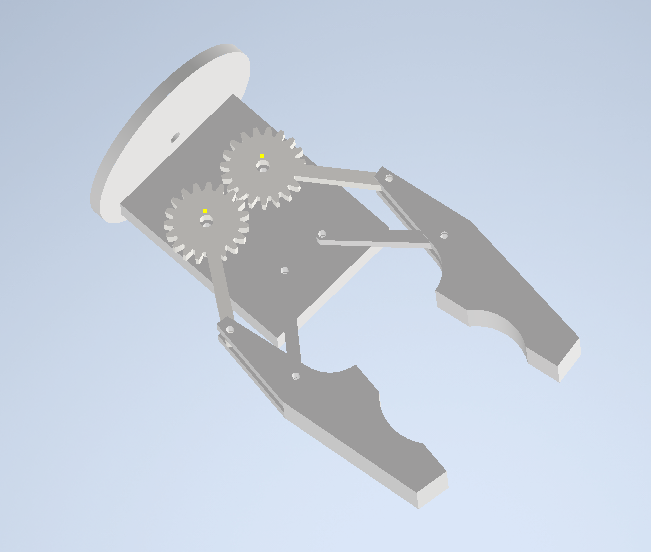
This is very convenient because it is remote control via Bluetooth

The code for Arduino in Appendix 3.

# **6.Model**

**A close-up of a sword

Description automatically generated with low confidence**

****

# **7.APPENDIX**

## 7.1 MATLAB CODE

clear all

close all

clc

syms a1 a2 a3 a4 ;

syms d6;

syms alpha1 theta2 theta3 theta4 theta6;

%system 1

Tranx\_a1=[1 0 0 a1;0 1 0 0;0 0 1 0;0 0 0 1];

Rotx\_alpha1=[1 0 0 0;0 cos(alpha1) -sin(alpha1) 0; 0 sin(alpha1) cos(alpha1) 0;0 0 0 1];

%system 2

Rotz\_theta2=[cos(theta2) -sin(theta2) 0 0;sin(theta2) cos(theta2) 0 0;0 0 1 0; 0 0 0 1];

Tranx\_a2=[1 0 0 a2;0 1 0 0;0 0 1 0;0 0 0 1];

%system 3

Rotz\_theta3=[cos(theta3) -sin(theta3) 0 0;sin(theta3) cos(theta3) 0 0;0 0 1 0; 0 0 0 1];

Tranx\_a3=[1 0 0 a3;0 1 0 0;0 0 1 0;0 0 0 1];

%system 4

Rotz\_theta4=[cos(theta4) -sin(theta4) 0 0;sin(theta4) cos(theta4) 0 0;0 0 1 0; 0 0 0 1];

Tranx\_a4=[1 0 0 a3;0 1 0 0;0 0 1 0;0 0 0 1];

%system 5

Rotz\_90=[0 -1 0 0;1 0 0 0;0 0 1 0;0 0 0 1];

Rotx\_90=[1 0 0 0;0 0 -1 0;0 1 0 0;0 0 0 1];

%system 6

Rotz\_theta6=[cos(theta6) -sin(theta6) 0 0;sin(theta6) cos(theta6) 0 0;0 0 1 0; 0 0 0 1];

Tranz\_d6=[1 0 0 0;0 1 0 0;0 0 1 d6;0 0 0 1];

%Danavit-Hartenberg Notations

A1=Tranx\_a1\*Rotx\_alpha1; A1=simplify(A1)

A2=Rotz\_theta2\*Tranx\_a2; A2=simplify(A2)

A3=Rotz\_theta3\*Tranx\_a3; A3=simplify(A3)

A4=Rotz\_theta4\*Tranx\_a4; A4=simplify(A4)

A5=Rotz\_90\*Rotx\_90 %A5=simplify(A5)

A6=Rotz\_theta6\*Tranz\_d6; A6=simplify(A6)

%Forward kinematic for each joint

T01=A1;

T02=T01\*A2;

T03=T02\*A3;

T04=T03\*A4;

T05=T04\*A5;

T06=T05\*A6;

T06=simplify(T06)

## 7.2 ROSNODE CODE

#include <ros/ros.h>

#include <std\_msgs/Float64.h>

#include <sensor\_msgs/Joy.h>

ros::Publisher servo1\_pub;

ros::Publisher servo2\_pub;

ros::Publisher servo3\_pub;

ros::Publisher servo4\_pub;

ros::Publisher servo5\_pub;

void joyCallback(const sensor\_msgs::Joy::ConstPtr & joy) {

double first\_joint =joy->axes[0]

double second\_joint = joy->axes[1];

double third\_joint = joy->axes[2];

double fourth\_joint = joy->axes[3];

double fifth\_joint = joy->axes[4];

std\_msgs::Float64 servo1;

std\_msgs::Float64 servo2;

std\_msgs::Float64 servo3;

std\_msgs::Float64 servo4;

std\_msgs::Float64 servo5;

servo1.data=first\_joint;

servo2.data=second\_joint;

servo3.data=third\_joint;

servo4.data=fourth\_joint;

servo5.data=fifth\_joint;

servo1\_pub.publish(servo1);

servo2\_pub.publish(servo2);

servo3\_pub.publish(servo3);

servo4\_pub.publish(servo4);

servo5\_pub.publish(servo5);

}

int main (int argc, char \*\*argv) {

ros::init(argc,argv,"joy\_controller\_node");

ros::NodeHandle nh\_("~");

std::string servo1\_name;

std::string servo2\_name;

std::string servo3\_name;

std::string servo4\_name;

std::string servo5\_name;

std::string servo1\_string;

std::string servo2\_string;

std::string servo3\_string;

std::string servo4\_string;

std::string servo5\_string;

nh\_.param<std::string>("first\_motor\_name",servo1\_name,"RobotArm\_servo1");

nh\_.param<std::string>("second\_motor\_name",servo2\_name,"RobotArm\_servo2");

nh\_.param<std::string>("third\_motor\_name",servo3\_name,"RobotArm\_servo3");

nh\_.param<std::string>("fourth\_motor\_name",servo4\_name,"RobotArm\_servo4");

nh\_.param<std::string>("fifth\_motor\_name",servo5\_name,"RobotArm\_servo5");

servo1\_string.append("/vrep/")

servo1\_string.append(servo1\_name);

servo2\_string.append("/vrep/")

servo2\_string.append(servo2\_name);

servo3\_string.append("/vrep/")

servo3\_string.append(servo3\_name);

servo4\_string.append("/vrep/")

servo4\_string.append(servo4\_name);

servo5\_string.append("/vrep/")

servo5\_string.append(servo5\_name);

ros::Subscriber joy\_sub;

joy\_sub = nh\_.subscribe<sensor\_msgs::Joy>("joy",1,joyCallback);

servo1\_pub = nh\_.advertise<std\_msgs::Float64>(servo1\_string,1);

servo2\_pub = nh\_.advertise<std\_msgs::Float64>(servo2\_string,1);

servo3\_pub = nh\_.advertise<std\_msgs::Float64>(servo3\_string,1);

servo4\_pub = nh\_.advertise<std\_msgs::Float64>(servo4\_string,1);

servo5\_pub = nh\_.advertise<std\_msgs::Float64>(servo5\_string,1);

ros::Rate loop\_rate(5);

while (ros::ok()){

ros::spinOnce();

loop\_rate.sleep();

}

}

## 7.3 ARDUINO CODE

#include <SoftwareSerial.h>  
#include <Servo.h>  
SoftwareSerial HC05(2,3);  
char tem;  
Servo myservo1;  
int servoPin1=9;  
int angle1=0;  
Servo myservo2;  
int servoPin2=10;  
int angle2=0;  
void setup (){  
  Serial.begin (115200) ;  
  HC05.begin (9600) ;  
  myservo1.attach(servoPin1);  
  myservo2.attach(servoPin2);  
}  
void loop() {  
  if (HC05.available ()) {  
    tem= HC05.read () ;  
  }  
switch (tem) {  
  case 'F':  
  {  
  angle1=angle1+1;  
  break;  
  }  
  case 'B':  
  {  
  angle1=angle1-1;  
  break;  
  }  
  case 'L':  
  {  
  angle2=angle2+1;  
  break;  
  }  
  case 'R':  
  {  
  angle2=angle2-1;  
  break;  
  }  
  }  
   Serial.println (tem) ;  
   myservo1.write(angle1);  
   delay(5);  
   myservo2.write(angle2);  
   delay(5);  
}