ROBOTICS PROJECT REPORT

Submitted by:

Kanishk Raj (RA2011038010001)

Mohamed Hisham (RA2011038010022)

Varoon (RA2011038010031)

Under the guidance of:

Dr. A. Vimala Starbino

(Associate Professor, Department of Mechatronics Engineering)

Of

Fifth Semester (18MHE451T)

Mechatronics Engineering Specialization in Robotics

Of

SRM

Institute of Science and Technology

Faculty Of Engineering & Technology

S.R.M Nagar,

Kattankulathur,

October 2022

Project for CLA-4

Title: Prototype of RRR Manipulator

AIM:- To make a working prototype of Rotation- Rotation-Rotation (RRR) manipulator and getting the DH parameter for it.

COMPONENTS REQUIRED:

- Arduino UNO
- Ice-cream sticks
- Bread Board
- Connection Wires
- 9V Batteries
- Joystick
- Glue Gun
- Button Switch

Theory of RRR Manipulator:

The robot manipulator is set up as an R-R-R configuration (3 revolute joints). It is controlled by an Arduino Uno microcontroller which accepts input signals from a user by means of a joystick. The arm is made up of three rotary joints and an end effector, where rotary motion is provided by three servomotors.

DH Parameter:

Denavit-Hartenberg (DH) parameters are often used in robotics to describe the robot properties like axis orientations and arm lengths.

- coordination axes are used to describe where the robot actuators are placed
- the Z axis is the rotation axis for a rotational actuator and prismatic axis for a linear one
- the DH parameters describe 4 of the 6 degrees of freedom to transform one coordinate system to the next: Z translate and rotate, then X translate and rotate. Not all transformations are possible (Y translate and rotate).
- the DH parameters are defined in M669 A parameter.

DH is using 4 parameters:

- rotation by Z axis
- translate by Z axis
- rotation by X axis
- translate by X axis

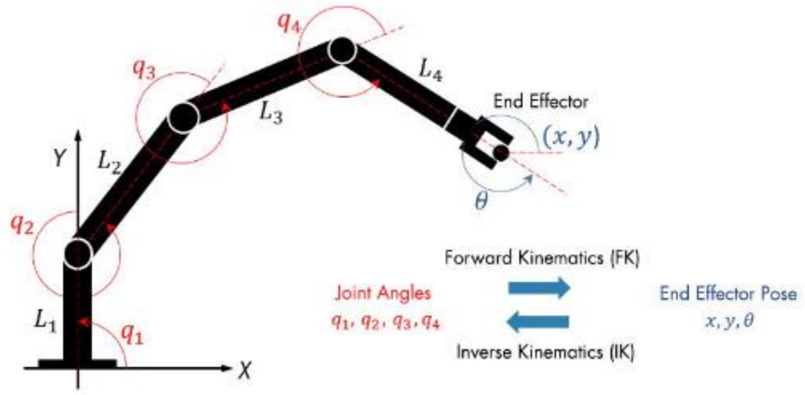
To give maximum flexibility, robot kinematics allows two additional parameters:

- rotation by Y axis
- translate by Y axis

The order of transformations is important, because matrix multiplications' orders are not commutative. DH transformations are in the order ZX, the extended version ZYX (which is the same order like RPY in aviation).
The transformations are 4x4 matrices as described above and are chained to calculate the complete transformation from base to endpoint. The result is a position and orientation of the endpoint (e. g. hotend nozzle, drill tip).

Inverse kinematics:

Kinematics is the study of motion without considering the cause of the motion, such as forces and torques. Inverse kinematics is the use of kinematic equations to determine the motion of a robot to reach a desired position. For example, to perform automated bin picking, a robotic arm used in a manufacturing line needs precise motion from an initial position to a desired position between bins and manufacturing machines. The grasping end of a robot arm is designated as the end-effector. The robot configuration is a list of joint positions that are within the position limits of the robot model and do not violate any constraints the robot has.



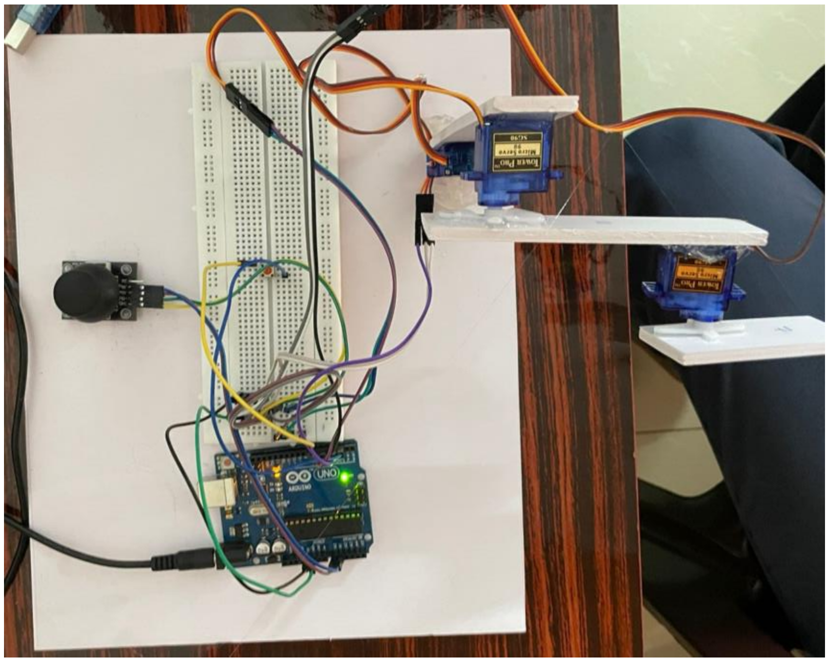
Once the robot's joint angles are calculated using the inverse kinematics, a motion profile can be generated using the Jacobian matrix to move the end-effector from the initial to the target pose. The Jacobian matrix helps define a relationship between the robot's joint parameters and the end-effector velocities.

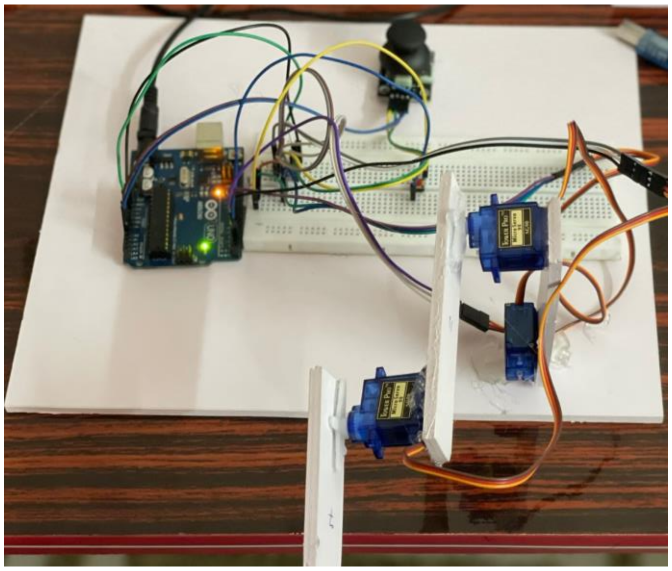
In contrast to forward kinematics (FK), robots with multiple revolute joints generally have multiple solutions to inverse kinematics, and various methods have been proposed according to the purpose. In general, they are classified into two methods, one that is analytically obtained (i.e., analytic solution) and the other that uses numerical calculation.

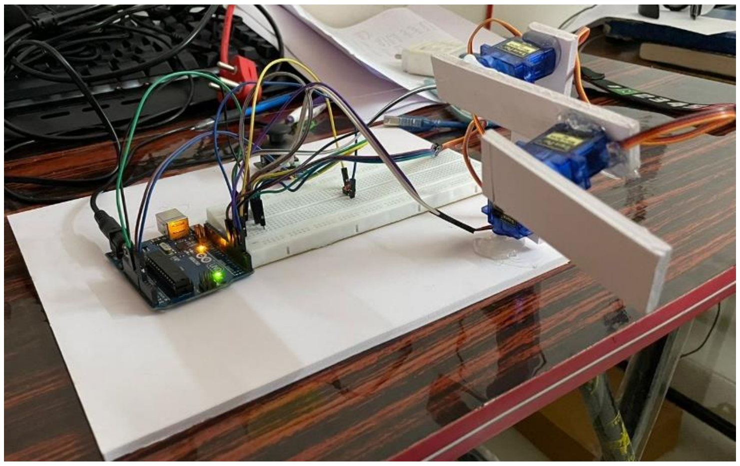
Applications:

- Spot welding
- Painting, Polishing, Grinding
- Retinal Microsurgery
- Machine Tending
- Machine Handling
- Arc Welding
- Assembly
- Picking, Packing, Palletizing

Pictures of the manipulator:







Program:

```
#include <Servo.h>


Servo Servo_X;
Servo Servo_Y;
Servo Servo_Grip;


#define Joy_X A0
#define Joy_Y A1
#define BUTTON 2


int Joy_Val = 0;
boolean state = false;
void setup()
{
  Servo_X.attach(3);
  Servo_Y.attach(5);
  Servo_Grip.attach(6);
  pinMode(BUTTON, INPUT_PULLUP);
}
void loop()
{
  Joy_Val = analogRead(Joy_X);
  Joy_Val = map(Joy_Val, 0, 1023, 180, 0);
  Servo_X.write(Joy_Val);

  Joy_Val = analogRead(Joy_Y);
  Joy_Val = map(Joy_Val, 0, 1023, 0, 180);
  Servo_Y.write(Joy_Val);
  delay(15);
 if (digitalRead(BUTTON) == LOW)
{
    if (state == false) {
    state = true;
    Servo_Grip.write(0);
    delay(500);
    } else {
      state = false;
      Servo_Grip.write(90);
      delay(500);
    }
  }
}
```

Inference:

- This RRR manipulator has 3 joints and 2 links. The manipulator was built with servo motors and the links are made with white board.
- The control of each motor is done with the help of joystick control, 3 motors for the revolute motion and a separate motor for the gripper’s motion.
- The program is uploaded to the Arduino UNO and the motors are driven by the motor driver which has pre-installed program in it.
- The Arduino is the central unit of the whole manipulator’s function as it holds the program and the algorithmic function of each actuator.