Logo, company name

Description automatically generated

ENG5220: Real Time Embedded Programming

Team – 11

|  |  |
| --- | --- |
| **NAME** | **ID** |
| Athul Prasad Prasad Geetha | 2789738P |
| Gaurav Madihalli Sreekantegowda | 2809092M |
| Aishwarya Palit | 2808985P |
| Rohan Kumawat | 2771441K |

|  |
| --- |
|  |

A picture containing text, clipart

Description automatically generated

Graphical user interface

Description automatically generated

Introduction

This project is about building a robotic arm which is equipped with a camera sensor and machine learning algorithms, our robot is designed to play table tennis in real-time by tracking the ball's trajectory and hitting the ball if its within its proximity.

Our project is divided into two parts – 1) Building of Robotic Arm and 2) Ball tracking and trajectory generation.

1. Building Robotic Arm

Below are the steps we follow to build our robotic arm:

1. **Determine the requirements and specifications of the robotic arm**:

To design any robotic arm, the basic criteria is the purpose. We choose this project as this is one of the real time embedding projects which can be useful in sports training, where athletes can practice against a reliable opponent. Next step is to determine the size, weight, payload capacity, and range of motion of the robotic arm. This information will help you choose the appropriate materials, components, and tools needed to build the arm.

1. **Design the robotic arm:**

The next step is to design of robotic arm. We designed the robotic arm in Fusion 360. We have decided about the number of joints, type of actuators, sensors, and control system in that phase. We have designed our robotic arm with 5 degrees of freedom.

Icon

Description automatically generated

**Fig.1 CAD Model**

1. **Choose the components:**

Select the appropriate components such as motors, sensors, controllers, and power supply based on the specifications and design of the robotic arm. For the actuation of the robot, we choose servo motors because servo motor is designed to provide precise control of its angular position, speed, and torque. They provide precise control of the arm's movements, allowing the arm to move and manipulate objects with precision and accuracy.

1. **Solving kinematics of robotic arm:**

The kinematics of a robotic arm describes the motion of its joints and the position and orientation of its end effector, which is the device or tool attached to the end of the arm. The kinematics of a robotic arm is important for understanding how the arm moves and how it can manipulate objects in space.

There are two types of kinematics that are commonly used in robotics: forward kinematics and inverse kinematics.

Forward kinematics is the process of determining the position and orientation of the end effector of a robotic arm given the joint angles. It involves calculating the transformation matrix that relates the position and orientation of the end effector to the joint angles. This is often represented by a mathematical model or algorithm that maps the joint angles to the end effector's position and orientation.

A piece of paper with writing on it

Description automatically generated with medium confidence

**Fig.2 D-H Table and Forward Kinematics**

**Text, letter

Description automatically generated**

**Fig.3 Inverse Kinematics Equation**

1. **Simulation of Robotic Arm:**

Once the CAD model is ready, we have imported the CAD model and simulate it in Webot in C++. Simulation allows robotics engineers to design and optimize the robot's mechanical and control systems without the need for physical prototypes. This can save time and reduce costs associated with building and testing physical prototypes.

A picture containing grass, green

Description automatically generated

**Fig.4 Simulation in Webot**

1. **3D printing the parts:**

You need to fabricate the parts of the robotic arm using tools such as 3D printers, CNC machines, or manual machining tools. It's important to ensure that the parts are precise and fit together correctly.

A picture containing text, indoor, white, arranged

Description automatically generated

**Fig.5 3D-Print Materials**

1. **Assemble the robotic arm:**

You need to assemble the parts of the robotic arm using fasteners such as bolts, screws, or adhesives. You should test the arm's functionality at each stage of the assembly process.

1. **Trajectory of Robotic Arm:**

The trajectory of a robotic arm refers to the path followed by its end effector as it moves through space to complete a task or reach a specific target. The trajectory is determined by the motion of the arm's joints and can be controlled by the robot's control system.

To determine the trajectory of a robotic arm, the kinematics of the arm are first calculated. This involves determining the joint angles and positions required to move the end effector from one position to another. The desired trajectory can be programmed into the control system of the robot or can be generated in real-time based on feedback from sensors or other inputs.

There are different types of trajectories that can be used to control the motion of a robotic arm, depending on the task at hand. We have used Joint Space Trajectory.

**Joint space trajectory** refers to the path followed by the joints of a robotic arm as it moves through space to complete a task or reach a specific target. It is also known as joint space motion or joint space planning.

In joint space trajectory planning, the path of the end effector is specified in terms of the desired positions and orientations of the end effector at various points in time. These positions and orientations are then converted into joint angles that correspond to the movement of the robotic arm's individual joints.

1. **Program the robotic arm:**

You need to write the code that controls the robotic arm's movements and behaviour. Once you get the trajectory generation, you can code the servo motors based on your trajectory generation.

1. **Test and calibrate the robotic arm:**

You need to test the robotic arm's performance and calibrate the sensors and actuators to ensure that they work properly. You should also test the arm's safety features and make any necessary adjustments.

**2)** Ball tracking and trajectory generation

In this part, we have placed two cameras, one at the right side of the robotic arm and another one at the top. The side camera will track the projectile motion of the ball and top camera will give the linear motion of the ball.

To calculate the projectile motion of the ball, we have used **Polynomial Regression** model of order 3 and for linear motion **Linear Regression Model.**

**Polynomial Regression** - In polynomial regression, a polynomial equation of a certain degree is fitted to the data points, with the aim of finding the best fit curve that passes through as many data points as possible. The degree of the polynomial equation is determined by the complexity of the data.

Polynomial regression is often used in machine learning and data analysis to model complex relationships between variables, particularly when the relationship is nonlinear. It can be used to predict future values of a dependent variable based on the values of one or more independent variables.

However, it is important to note that polynomial regression can be prone to overfitting, which is when the model is too closely fitted to the training data and does not generalize well to new data. To avoid overfitting, it is important to validate the model using techniques such as cross-validation and regularization.

In the case of the projectile estimation from the side view camera we are going to fit a polynomial curve of n=3 to estimate the co-responding next output values from the previous two values in the trajectory. This led to good estimation of the final trajectory estimation of the ball from side view.

**Linear Regression -** Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. The goal of linear regression is to find the best-fit line or plane that can explain the variance in the dependent variable based on the values of the independent variable(s).

This algorithm is used for the front view camera in which the estimation is given out from a linear regression model for final movement it is merged with side view projection co-ordinates together to give a approximate estimate of the next position of the ball in the projectile.