Autonomous Robots

CSE 4360 – 001

12/05/2024

Due Date : Dec 11 , 2024

Project 3

Group Members :

Jason Cabrejos

Don Bui

Aryan Mainkar

**The Design** :

A close-up of a machine

Description automatically generated

**Design choices :**

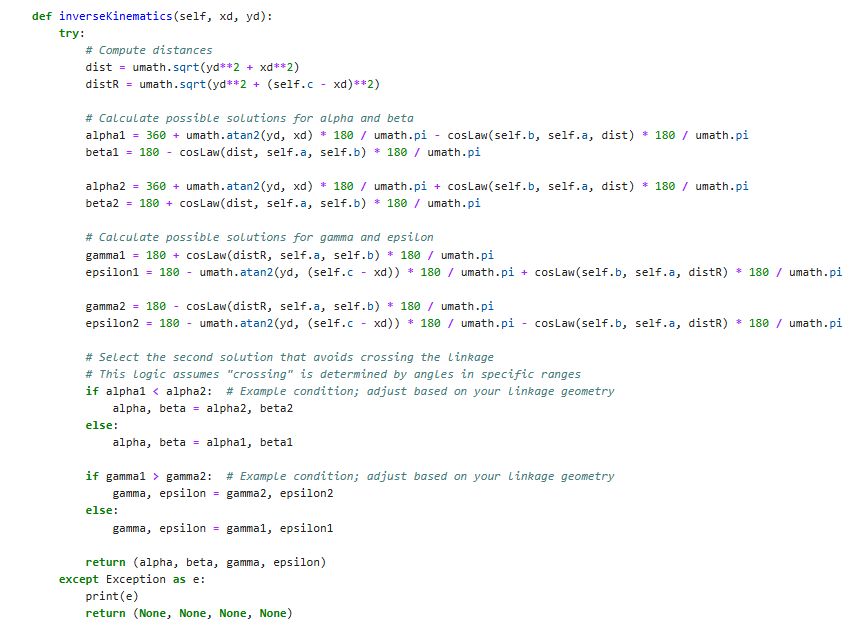
The design of the robot:

* Structure influenced by a pantograph: Like a pantograph, the robot's planar linking mechanism comprises a network of connected beams and joints. Motions at the end-effector (pen) can be precisely replicated or scaled thanks to this design.
* Mechanical Linkages: Five links are used in the arrangement, which are joined by pivot points to allow for controlled movement in a two-dimensional plane. For ease of building and customisation, the linkages seem to be constructed using modular components such as those found in LEGO. Pen holder An end-effector is a pen or other writing tool that is positioned vertically and intended to come into contact with a drawing surface. Rigid supports stabilise the holder, guaranteeing accurate writing or sketching. Gear Mechanism for Motion Control: The design incorporates gears, most likely to increase torque or precision and convey motion. Motors are probably attached to these gears in order to actuate the linkages.
* Motorised Control: To regulate the movement of the linkage system, the mechanism appears to be driven by motors, which are probably concealed or installed beneath the structure.
* Control device:The motors and linkage motions are managed by a central hub, which may be an intelligent programmable device.For jobs involving writing or drawing, it could be configured to adhere to particular inputs or patterns.
* Modular Construction: The design makes use of easily assembled, modular parts that allow for customisation and flexibility for various experiments or activities.
* Stable Base and Drawing Surface: The end-effector may travel across a level surface for reliable operation, and the base is rigid to reduce vibrations.

**Picasso.py :**

* Picasso code is made to operate a robotic arm with several motors and uses inverse kinematics to calculate joint angles for accurate placement.
* With the Picasso class encompassing parameters like arm link lengths, gear ratios, and the robot's starting location, the code employs a class-based structure.
* The fundamental functionality is supplied by the inverse kinematics approach, which determines potential configurations by computing joint angles using trigonometric equations such as the cosine rule. For seamless transitions between designated target positions, the InterpolateMoves function creates intermediate points.
* By interpolating pathways in tiny stages, it guarantees fine-grained control. By instructing motors to perform estimated movements and guaranteeing synchronisation through PID control loops written in helper functions (TurnMotorForAngle and TurnMotorsForAngle), the plot approach combines these aspects.
* For jobs requiring delicate control, such as drawing or object manipulation, articulated robotic arms can be used because of the design's emphasis on accuracy and versatility for various motion trajectories.

A computer screen shot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

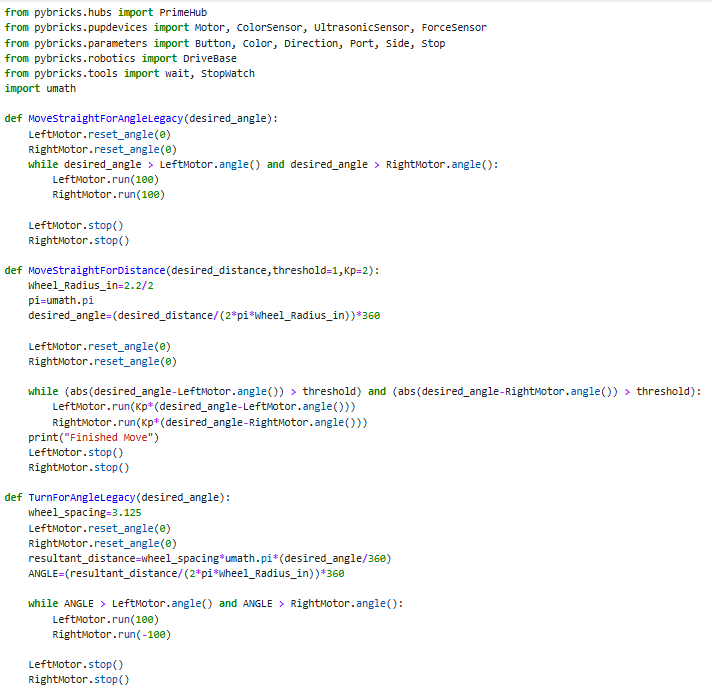
Description automatically generated

A computer screen shot of a program

Description automatically generated

**CalibrationTest.py :**

* Code B focuses on controlling a two-wheeled robot with motors for simple motions like turning and straight drive.
* It offers several locomotion techniques, such as MoveStraightForDistance, which uses proportional control to guarantee precision and determines the motor angles needed to travel a given distance based on the wheel radius.
* Similar to this, TurnForAngle uses a closed-loop control system to execute turning capability by precisely adjusting the heading using the onboard IMU (inertial measuring unit).
* For less complex control, simpler legacy techniques like MoveStraightForAngleLegacy and TurnForAngleLegacy are offered; these methods just use motor encoders and do not receive feedback from the IMU.
* The code is appropriate for wheeled robots doing navigation tasks like exploration or maze-solving since the entire design strives for robust and efficient movement with proportional control for error correction.

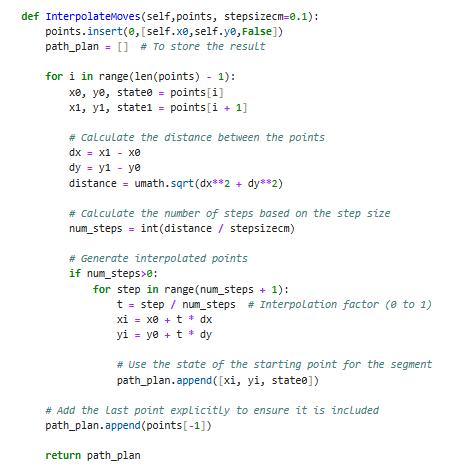


A screen shot of a computer code

Description automatically generated

**Navigation Strategy** :

* By creating intermediate waypoints, the InterpolateMoves function executes a navigation technique that guarantees accurate and seamless movement between designated points.
* The method determines the interpolated positions along the straight-line trajectory between each pair of subsequent points by segmenting the path into smaller steps according to a specified step size (stepsizecm).
* As a result, the robot can follow a consistent and continuous course, keeping each segment's starting point state (such as tool activation).
* The method works especially well for fine-grained control in applications that need precise transitions, like path-following mobile robots or robotic arms.



**Calibration Strategy** :

* The provided code demonstrates a calibration strategy aimed at ensuring precise and consistent movements for a robot using its motors and sensors.
* Motor angle sensors are reset at the beginning of movement functions (e.g., MoveStraightForAngleLegacy, TurnForAngle) to establish a reference point for measuring rotations or distances.
* Proportional control (Kp) is used in movement functions like MoveStraightForDistance and TurnForAngle to minimize errors by dynamically adjusting motor speeds based on the difference between desired and actual positions or angles.
* Additionally, the IMU (Inertial Measurement Unit) is utilized for precise angular calibration in turning, with its heading reset to zero before initiating a turn. These steps ensure reliable navigation and minimize drift during operation.

A computer screen shot of a code

Description automatically generated

A computer screen shot of a computer code

Description automatically generated