Description

We plan to design and develop a manipulator arm path planner for Acme robotics. With the advent of industrial robotics, a manipulator is used to direct material without direct physical contact with the operator. It was developed to handle hazardous materials or access places that are not easily accessible by humans. At Acme, the manipulator will be used for picking heavy containers from the conveyer belt of the production line and placing it on the packaging line. The manipulator will have four degrees of freedom and will be placed on a support structure, between the production and packaging line conveyor belts.

The programming language used will be C++ and build system Make.

Design and development process.

The software design includes five blocks namely kinematics, path planning, control system, simulation, and testing. Incremental Rotary encoders will be used at each motor link to calculate the angle moved by a particular arm.

- **Kinematics**
  This block consists of two sections, **Inverse Kinematics** and **Forward Kinematics.**
  The description of the blocks are as follows
  - **Inverse kinematics**
    This block will use inverse kinematics to deduce the desired angle to be moved by the links for the end effector to achieve a particular target. The target is set by the **Path planning block**.
  - **Forward Kinematics**
    This block will use forward kinematics to verify the position of the end effector during unit testing.

- **Path planning**
  The path planning block will deduce an optimal path based on the start and target point coordinates, mechanical constraints such as link lengths, and angular constraints. The algorithm used for path planning is **RRT***. A brief introduction to RRT* is given below.
  - **RRT***
    The algorithm records the distance each vertex has traveled relative to its parent vertex. This is referred to as the vertex. After the closest node is found in the graph, a neighborhood of vertices in a fixed radius from the new node is examined. If a node with a cheaper cost than the proximal node is found, the cheaper node replaces the proximal node. After a vertex has been connected to the cheapest neighbor, the neighbors are again examined. Neighbors are checked if being rewired to the newly added vertex will make their cost decrease. If the cost does indeed decrease, the neighbor is rewired to the newly added vertex.

- **Control system**
  Based on the feedback from the **Kinematics block** of the current position of the end effector and the path defined by the Path planning block, the **Control system** block

ensures that the path is followed by the links and the end-effector reaches the desired target point within a pre-defined error margin.

- **Simulation**
  Matlab will be used for simulation of the algorithm.

- **Testing**
  Software testing will be done to check the repeatability, effectiveness of the output. Unit tests will be performed to check the values of encoders and the working of specific functions.

Risks and Mitigation

Risk: The system involves a risk of encoder malfunction, motor or link breakage.

Mitigation: Unit tests will be performed at the startup of the manipulator. These tests will check the encoder values as well as the functioning of the motors. Also, dedicated functions will be present in the software to check these values in real-time. If any malfunction happens, the software will auto-stop the process.

Final Deliverables

The final deliverable will include a Software program able to control the four degrees of freedom manipulator.

Team members

Ameya Konkar
Rahul Karanam

References

https://theclassytim.medium.com/robotic-path-planning-rrt-and-rrt-212319121378