ROBOTIC DRAWING MACHINE

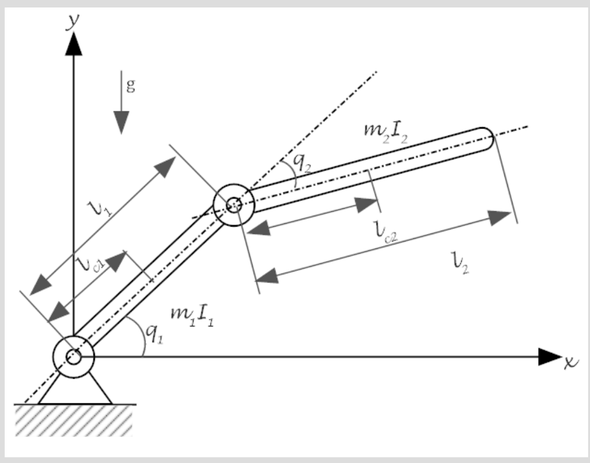
Rishabh Sharma,Shishir Mishra,Harsh Govil, Ahemad Khan Pathan,Jeet Patel
Kulbhushan Goswami, Ripudaman Singh, Vivek Kumar, Vishal Kumar
ROBOTICS CLUB

Abstract/Introduction

- Perhaps you have never been gifted at drawing, or you don't have time to do it yourself, so why not let a simple 2-D Robotic arm draw for you. This simple two axis device can accurately move a pen to draw out anything.
- This Robotic Arm has a two axis control and a special mechanism to raise and lower the pen.
- Each axis is controlled using a single Servo motor. Pen-control is also achieved using a servo. All electronics are mastered by an Arduino and powered by a wall adaptor.
- This arm is capable of drawing on A4 size sheet.
- Image reading is based on python OpenCV.

Results

- Finally, while applying code on an image results were great, the code was able to process the nodes in the image.
- When Hardware and Codesystem was implemented the results were satisfactory and minute error were detected in the drawing .
- The speed of machine depends on the complexity of the Image. More complex the image more the number of nodes more time will be required to process them.



Methodology

- Image is read and processed through Python Computer Vision(OpenCV library).
- Image is first converted from color to greyscale and then to black and white.
- Each of the resulting pixels is converted into location array.
- We applied "Max pooling" to extract efficient number of nodes from the image.
- A distance matrix is created with respect to the nodes.
- We apply DFS(Depth First Search) and we use a greedy approach for finding an optimal order.
- Simultaneously, a matrix is created for obtaining movement of pen perpendicular to plane of paper.
- Then we eliminated unnecessary nodes using method of slope matching.
- Finally the sequence of coordinates is fed into the Arduino after application of inverse kinematics.
- The Arduino controls the servo motors, which in turn move the arms accordingly.
- For the arm body we have used Acrylic Sheets drilled with holes of proper sizes for the two servos.

Conclusion

- The Image Processing and its Implementation on a Robotic arm was achieved precisely with minimal error in the drawn image.
- The error is mainly due to external factors such as weight of servos and arms, which are responsible for an unnecessary torque.

References

- Depth First Search (DFS) algorithm https://en.wikipedia.org/wiki/Depth-first_search
- Python Image Processing Via Open CV.
- Inverse Kinematics Angles
- Max Pooling
- 2 Degree of Freedom Planar Manipulator
- Arduino IDE (www.arduino.cc)
- PYSerial API