

# A RRP Configuration Robot Arm for Drawing Application

Pratik Baid\*, Manoj Kumar.V

Department of Mechanical Engineering, S.R.M University, Kattankulathur, Chennai – 603202

\*Corresponding author: E-Mail: pratikbaid@rocketmail.com

## ABSTRACT

This work presents the design and construction of a serial robotic manipulator system solely dedicated to the drawing activity. The arm's configuration used here is of a three-joint planar arm, with an assigned joint which allows the movement of the pen in upward and downward direction in order to make or break contact with the drawing surface. The modelling of the drawing robot was done using SOLID WORKS 2013 software with representation of 3-D geometries and various proximities related to modelling and motion were analyzed. A basic prototype of the drawing robot was fabricated using wood fibre and mild steel material. A combination of Arduino UNO board and servo motors is used to control the link motions. The input image file is processed using MATLAB 2013 software and is provided to the robotic hand system via ARDUINO IDE software. The MATLAB program is coded on the basis of canny edge detection algorithm. By using the canny operator the skeleton of the image can be obtained. This will be drawn by the robotic hand system.

**KEY WORDS:** Calibration, Canny edge, Canny Operator, Edge detection, Joint angle, Link length, Robot arm, RRP, SCARA.

## 1. INTRODUCTION

Previously there have been a lot of researches in recreational uses of robots. Recent research on robots has devoted significant effort on developing robots that can match human behavior on high level tasks that require integration of sensing, physical motion and intelligence. This makes the robot behavior more human-like. The drawing robot is one of them. There have been several demonstrations of drawing robots in past few years.

A drawing robot is a robot arm which is capable of drawing the image provided to it via a computer system making use of image processing techniques. It is solely dedicated to the drawing activity. It can be defined as a robot that reads the pixel values of an image provided to it and then draws the same. It consists of mechanical links driven by servomotors which may be controlled using microcontrollers. The drawing robot must be capable of obtaining a fine portrait of the input image with good pixel readings and high stroke length such that it is closely similar to the given image. A drawing robot requires human-specific skills which are challenging tasks in the field of robotics.

Drawing robots can possess serial manipulators or parallel manipulators with sufficient number of degrees of freedom in accordance to work space and the robot movement in it. Degrees of freedom refer to the number of possible free movements available for the robot. Practically, a SCARA manipulator with three degrees of freedom is preferred on basis of feasibility and operation cost.

The working of the drawing robot involves two major phases. The first phase involves extraction of the required features from the image to be drawn fed using a computer system and converting it into appropriate code using MATLAB software. In the second phase the MATLAB program is transferred to a robotic arm in terms of coordinates which are drawn or traced by the robot on the drawing surface. The drawing robot requires a pre calibration step before the robot can initialize the image drawing process. A fresh calibration is required in case the drawing surface is disturbed. Moreover, the drawing process can take place only on flat pre-calibrated surfaces.

**Back ground:** Robot Paul, a robotic installation that can produce observational face drawings of people. Paul is a naive drawer and it does not have high level knowledge of the structures constitutive of the human face such as the mouth, nose and eyes nor has the capability of learning expertise based on experience as a human can. However, Paul was able to draw up to a similar level based on a number of processes mimicking drawing skills and technique, which together formed a drawing cycle.

A demonstration of a robot equipped with force sensing capability that can draw portraits on a non-calibrated, arbitrarily shaped surface was done at BARC, Mumbai. This robot was able to draw on a non-calibrated surface by adjusting its drawing tool normal to the drawing surface, the tool's orientation was computed from the forces being sensed by a force sensor.

Robot Draw, combined recently developed internet based programming tools to generate three dimensional virtual values and models of robot arms from a DH parameter table. Robot-Draw combines hypertext markup language (HTML), practical extraction and report language (PERL), and virtual reality modeling language (VRML).

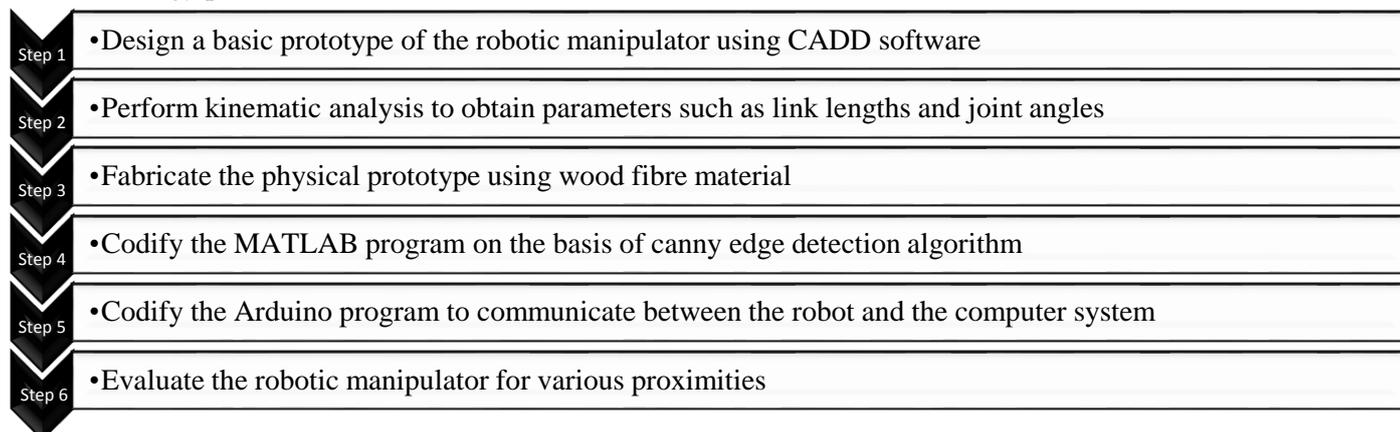
Betty, the portrait drawing humanoid robot solved line drawing problems presented via a modified Theta graph, called Furthest Neighbour Theta graph, which is computable in time. The results showed that the number of edges in the resulting drawing is significantly decreased without changing the details of the final output image.

A theory that it is possible to draw the portrait by the facial textures automatically was published. In order to analyze the detailed textures located at the neighborhood of the eyes, the texture analyzing method must be studied.

The principle of selecting the feature points and the portrait drawing method was done by B-spline curves using the self-reference parameter.

## 2. PROJECT METHODOLOGY

An apt methodology is followed to select suitable mechanism, kinematic concept and design for the drawing robot. First initialize the drawing robot design, select suitable link lengths and calculate the joint angle according to the workspace limitations. Model the drawing robot using SOLIDWORKS software. The control unit is formed using an arduino UNO board and a computer system which makes use of softwares like processing talk, MATLAB and arduino IDE. The drawing robot is programmed using the mentioned softwares and various experiments are performed to test the robot for various proximities and to evaluate the drawing robot's performance. Fig.1 depicts the methodology process.



**Fig. 1. Methodology flow chart**

**Kinematic Calculations:** The d-H table for the drawing robot of RRP configuration is-

d-H Parameters	a	$\alpha$	d	$\theta$
	$X_1$	+90	$X_2$	$\Theta_1$
	0	-90	0	$\Theta_2$
	0	0	$d_3$	0

Where, a = link length,  $\alpha$  = link twist, d = link offset,  $\theta$  = joint angle

$$T = {}^0T_1 X {}^1T_2 X {}^2T_3$$

$$T = \begin{bmatrix} C_1C_2 & -S_1 & -C_1S_2 & -C_1S_2/d_3 + x_1C_1 \\ S_1C_2 & C_1 & -S_1S_2 & -S_1S_2/d_3 + x_1S_1 \\ S_2 & 0 & C_2 & C_2/d_3 - x_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Let,  $\Theta_1 = \Theta_2 = 0$  and  $d_3 = 0.05$

$$T = \begin{bmatrix} 1 & 0 & 0 & 250 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -130 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D = \begin{pmatrix} -C_1S_2d_3 + x_1C_1 \\ -S_1S_2d_3 + x_1S_1 \\ d_3C_2 - x_2 \end{pmatrix}$$

$$D = \begin{pmatrix} 250 \\ 0 \\ -149.95 \end{pmatrix}$$

www.jchps.com

For inverse kinematics,

$$T = \begin{bmatrix} C_1C_2 & -S_1 & -C_1S_2 & -C_1S_2/d_3 + x_1C_1 \\ S_1C_2 & C_1 & -S_1S_2 & -S_1S_2/d_3 + x_1S_1 \\ S_2 & 0 & C_2 & C_2/d_3 - x_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{bmatrix}$$

- $\theta_1 = \text{Atan2}(-r_{24}, -r_{14})$
- $\theta_2 = \text{Atan2}(\pm \sqrt{(r_{14}^2 + r_{24}^2)}, -r_{34})$
- $D_3 = \sqrt{(r_{14}^2 + r_{24}^2 + r_{34}^2)}$

Let,  $-100 < \Theta_1 < +100$  $-30 < \Theta_2 < +30$  $0.05\text{m} < d_3 < 0.5\text{m}$ 

$$T = \begin{bmatrix} 0.354 & 0.866 & 0.354 & 0.106 \\ -0.612 & 0.500 & -0.612 & -0.184 \\ 0.307 & 0 & 0.308 & 0.212 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

On substituting the appropriate values we get,

$$\theta_1 = 60^\circ \quad \theta_2 = 45^\circ \quad d_3 = 0.30$$

Where, T = Over-all transformation matrix

 ${}^0T_1, {}^1T_2, {}^2T_3$  = Transformation matrix of each link

D = Translation vector

 $X_1$  = Length of link 1 $X_2$  = Length of link 2 $X_3$  = Length of link 3 $C_1 = \cos \Theta_1$  $S_1 = \sin \Theta_1$  $C_{12} = \cos(\Theta_1 + \Theta_2)$  $S_{12} = \sin(\Theta_1 + \Theta_2)$  $d_1$  = joint distance from link1 to frame origin $d_2$  = joint distance from link2 to frame origin $d_3$  = joint distance from link3 to frame origin

Atan2 = Inverse tan function

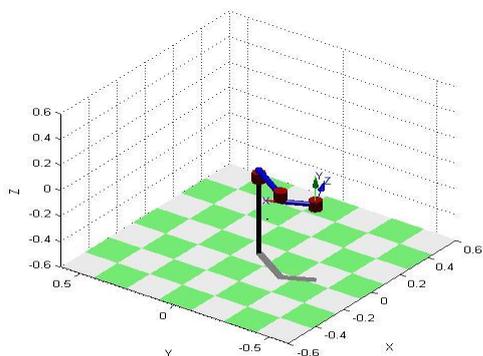


Fig.2. Kinematic Model

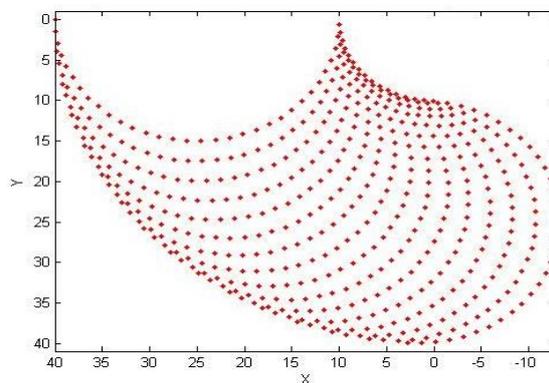
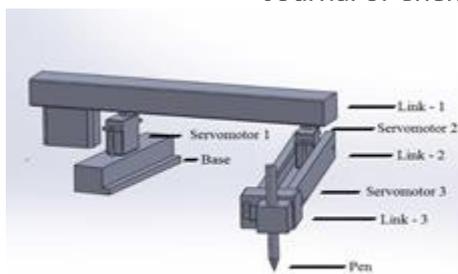


Fig.3. Workspace Model

**Design Considerations:** The drawing robot setup is composed of a left-handed robotic arm with a tool holder as an end-effector all attached to a wooden table as shown in fig. 4 and a laptop computer. Always present at the installation is a human assistant; their role is to change the paper and give the signal to the drawing robot that an image need to be drawn. The human operator also controls the reset button and is capable of initiating the pre-calibration process if required.

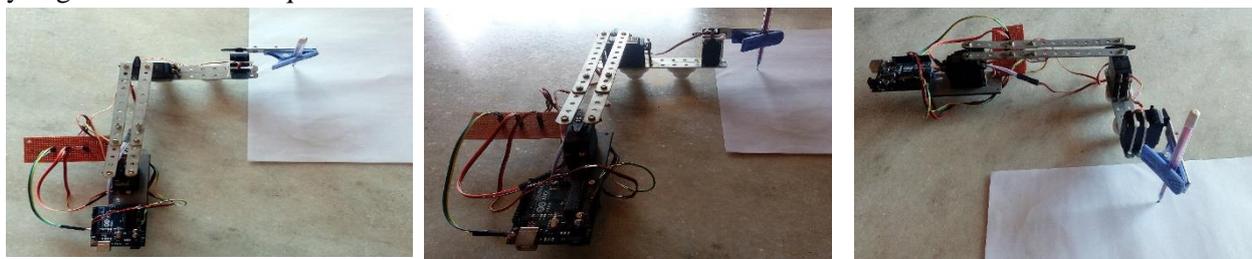


**Fig.4. CAD model of the prototype**

#### System Potrayal:

**Hardware:** The hardware of the drawing robot consists of three links in RRP configuration fabricated using wood fibre material. This material has less weight and optimal stress bearing capacity. The robot links are attached to servomotors. The servomotors are connected to an arduino UNO board via which the motion of the links is controlled. The CAD model of the prototype is done in Solid works 2013 which is shown in Fig. 4. The length of the three links are 250mm, 150mm and 40mm respectively. A uniform breadth of 40mm and thickness of 15mm is maintained for all the links. The physical model of the robot is shown in Fig. 5.

**Robotic control and software architecture:** An arduino UNO board and a laptop computer system which makes use of softwares like MATLAB and arduino IDE forms the control unit. Arduino IDE is used to control the servomotors via which the motion of the robotic links is controlled. MATLAB is used to process the image using canny edge detection technique.

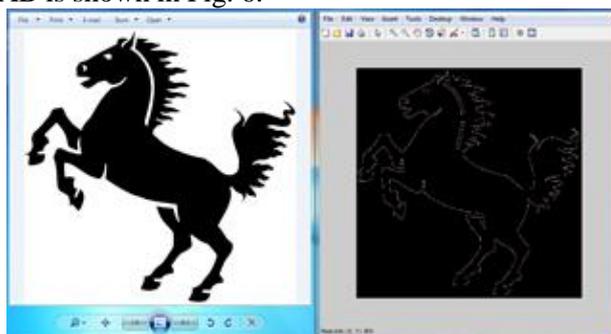


**Fig.5. Top, Front and Side view (from left) of the physical model of the prototype**

**Image Processing:** The input image is provided using a laptop computer and is processed using MATLAB software using canny edge detection algorithm. The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny's edge detection algorithm is a classical and robust method for edge detection in gray-scale images. The two significant features of this method are introduction of NMS (Non-Maximum Suppression) and double thresholding of the gradient image.

**Pre-Processing:** The canny edge detection algorithm is applicable only to gray scale images. Hence a control block to convert images of other formats to gray scale must be implied. The MATLAB 2013 software has inbuilt commands to convert images from one format to another. Using these commands a control block for image conversion is formed.

**Edge detection:** Whenever we draw an image, we try to find out the important edges or features that need to be drawn. The edge detection is a famous image processing technique and there are several edge detection algorithms available. We tried with various algorithms and found that canny edge detection algorithm was best suited for our task. The Canny edge detector finds out real and localized edges without being affected by noise. The output of edge detection algorithm via MATLAB is shown in Fig. 6.



**Fig.6. Output of canny detection algorithm**

**Post-Processing:** After the important features of an image are detected in terms of edges, some processing needs to be performed before these edges can be transferred to the robot for drawing. This post processing is explained in the following subsections.

- **Branch Removal:** There should be no branches in an edge so that the robot can draw that edge in a single go from one end of the edge to the other. For this, the branch-points were first detected. The pixel at such a point was then removed i.e. it was made 0 from 1 so that the branches were simply separated from each other.
- **Removal of Small Objects:** The edges which had too small number of pixels or the area covered was very small, were removed from the final output so that the noisy and unnecessary features are not drawn.
- **Differentiation of Edges:** The edges extracted by canny edge detection algorithm are in the form of binary pixels. What we have is a binary image in which all the pixels are set to 1 where an edge exists and the remaining are set to 0. From this image, pixels belonging to a single edge need to be grouped together so that we have a collection of edges rather than a collection of pixels. For this purpose we used 8 connectivity of the pixels and 8 connected neighbours were grouped together into a single edge. After this step, we have a set of edges that need to be drawn by the robot.
- **End-Point detection:** The end points of each of the edges were detected so that robot can draw that edge from start point to end point. Circular or cyclic edges in an image have no end points. For such edges, we arbitrarily removed one pixel to break the cycle, and then detected end-points of the broken cycle.
- **Storage of Edges:** The edges obtained were finally stored in a sorted manner from the longest edge to the shortest one so that the robot draws the most important features first moving towards the smaller ones.

### 3. CONCLUSION & FUTURE WORK

A brief study over various drawing robots has been done. The drawing robot is capable of drawing various shapes, alphabets and numbers. The drawing robot is also capable of processing human faces with a good output. Design trails were made and rectified using SOLIDWORKS software. This robot can draw on any even surface based on tool mounted. Fabrication of the design was done and experiments were carried out on the drawing robot to analyze its efficiency. Further a force measuring sensor can be equipped to the end effector of the drawing a robot which will indicate if the tool is in contact with surface where the image is to be drawn. As a result of this the drawing robot will be capable of drawing on irregular surfaces too. The laptop computer can be connected to a real time camera and images captured by it can be processed. Also it is possible to change the drawing robot tool such that the robot is capable of carving images on hard surfaces.

### 4. ACKNOWLEDGMENT

We wish to acknowledge our project coordinator Dr.Stalin John.M.R and the head of our department Mr.Prabhu.S and thank them for their deep interest they have shown in this project and for motivating us to do our best.

### REFERENCES

- Haibin Di, and Dengliang Gao, Gray-level transformation and canny edge detection for 3D seismic discontinuity enhancement, *Proceedings of Computers & Geosciences*, 72, 2014, 192 – 200.
- Melinda F. Robinette, Robot - Draw an Internet - Based Visualization Tool for Robotics Education, *Transactions on education*, 44 (1), 2011, 448 – 459.
- Meng, Cheng, Lau, Jacky, Baltés, John, Anderson, and Stephane, Durocher, A Portrait Drawing Robot Using a Geometric Graph Approach: Furthest Neighbour Theta Graphs, *IEEE Transactions of robots*, 123, 2013, 930 – 935.
- Patrick, Tresset, and Frederic Fol Leymarie, Portrait drawing by Paul the robot, *Proceedings of Computers & Graphics*, 37, 2013, 348 – 363.
- Ranita, Biswas, and Jaya Sil, An Improved Canny Edge Detection Algorithm Based on Type-2 Fuzzy Sets, *Proceedings of Procedia Technology*, 4, 2012, 820 – 824.
- Shubham Jain, Prashant Gupta, and Vikash Kumar, A Force-Controlled Portrait Drawing Robot, *IEEE Transactions of robot*, 15, 2015, 3160 – 3165.
- Zhuang Fu, Fei Ni, Qixin Cao, and Yanzheng Zhao, The facial texture analysis for the portrait drawing, *Proceedings of pattern recognition*, 43, 2010, 962 – 972.