Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 171 (2020) 2459–2468

Procedia Computer Science

www.elsevier.com/locate/procedia

Third International Conference on Computing and Network Communications (CoCoNet'19)

Design and Development ofa Low-Cost CNC Alternative SCARA Robotic Arm

Ashwin Misra[1]*, Ayush Sharma[1], Ghanvir Singh, Ashish Kumar, Vikas Rastogi

Department of Mechanical Engineering, Delhi Technological University, New Delhi, India

Abstract

The cost of Computer Numerical Control Machines is very high which leads to their widely inaccessible nature by small to mid-range manufacturing companies. Though, the technology is aimed at increasing efficiency and accuracy by achieving optimization in the traditional manufacturing methods, its most major drawback is the high cost of maintenance and installation. The article discusses an affordable solution to this issue. As a viable alternative, a SCARA Robotic arm is used to execute the process. The initial computer aided design is analysed for feasibility in real conditions using rigorous finite element methods and experimental validation. The workspace of the SCARA is modelled on general usage approximations; a detailed kinematic and dynamic robotic analysis is also done to ensure better process conditions and the future possibility to achieve force feedback control. This prototype can be operated using user-based systems and can be designed following the methodology proposed in this article. A combination of metals and non-metals are used depending upon various factors such as strength requirements, design complexity and cost. Both traditional and additive manufacturing techniques are employed to manufacture the parts, An Arduino microcontroller is used to execute all control structures and operations. The control for a batch operation can be entered through the touchscreen attached in the robot, with a special mode for individual production. The technology proposed in this paper is aimed at assisting Micro to small manufacturing enterprises (MSMEs) to achieve higher productivity.

Peer-review under responsibility of the scientific committee of the Third International Conference on Computing and Network Communications (CoCoNet'19).

Keywords: SCARA robot;CNC; Modelling; Robotic Analysis; PRR; Control

* Corresponding author. Tel.: +919971158598
E-mail address: ashwin_bt2k16@dtu.ac.in
[1] Authors contributed equally

1877-0509
Peer-review under responsibility of the scientific committee of the Third International Conference on Computing and Network Communications (CoCoNet'19).
10.1016/j.procs.2020.04.266

1. Introduction

Robotics and automation are paving new paths for manufacturing optimization and increasing the efficiency and productivity of industries. This results in more profitability for the company as well as more capital to allocate to research and development. Robotic arms are used in production facilities for their accuracy and repeatability; this eliminates the need of human operators and uncertainty associated with human operation. Another technology, Computer Numerical Control for various mechanical operations such as drilling, milling and lathe etc. are also used for the same purpose. In machines employing this technology, computer codes are executed with the use of G-codes and M-codes to manufacture a mechanical component. These are very accurate in operation as they are void of human errors. However, both of these technologies are very expensive as they are manufactured by large robotic and CNC companies. This is the reason due to which many small to mid-scale manufacturing companies are not able to benefit from this advancement. This article proposes a technique to counter this drawback with a cost-effective amalgam of Robotics and CNC.This article proposes a CNC SCARA Robotic arm to execute automated drilling operations. The complete, step by step process of the design as well as fabrication phase is discussed in the further sections. Firstly, a similar workspace robot is studied to find out the ballpark dimensions of the robot. These approximations are then removed and a final Computer Aided Design is developed according to the needs of various manufacturers. The workspace and robotic dimensions are based on a manufacturer survey in the area of New Delhi, India. This robot model is then tested in ANSYS for operational stress and strain limits. Weight reduction is also achieved using topology optimization. This robot is then imported to RoboAnalyzer for kinematic and dynamic analysis using the D-H Parameters. Then the manufacturing phase is executed, with a combination of conventional and additive manufacturing depending upon the stress analysis and cost of the robot. The whole electronics system is controlled by an Arduino microcontroller with various sensors to obtain operational data. The control structure is developed and executed through the Arduino interface and real time operation codes are also developed through an optional TFT screen.

The inspiration to achieve this objective through the help of SCARA robotic arm is due to its wide applications and control. A Selective Compliance Assembly Robotic Arm is widely used in the industry in pick and place operations, insertions and other similar operations. Its accuracy through the vertical axis is the major reason for it's application in these fields. Computer Numerical Control is a cutting-edge technology widely used by engineers worldwide. This article aims to merge these technologies and develop a low-cost solution. India, observes relatively lesser usage of these engineering resources due to their intrinsic expensive nature. This prototype aims to bridge the gap between technology and money.

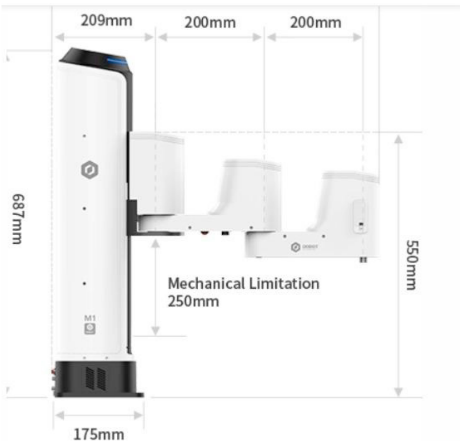

Fig. 1. Dobot M1 Industrial Robotic Arm

The Industrial robot used as a reference for the development of this article is shown in Fig.1. This industrial robot has wide applications in the industry ranging from drilling to high-tech operations employing the use of computer vision. The drawback being the price tag of $6,000 which is very expensive for small manufacturers in a developing country. The prototype developed in this article, is developed at a similar workspace and specifications as this robot. The advantage being the reduced price which is around $300.

2. Literature review

The design and development of an intuitive graphical user interface is discussed in the article [1]. A specific 4 degree of freedom Selective Compliance and Assembly Robotic Arm is focused to give a tangible perspective to the tool. The methodology is novel in developing a guide for forward and inverse kinematic analysis with proper data validation. It minimizes the time and effort required in the analysis process by developing a user-friendly application. However, no industrial integration is discussed in the paper and the main focus is on the firmware and not the practical usage of a robot.

In the article [2], the design and construction of a 3 degree-of-freedom SCARA robot is discussed. A comprehensive method to fabricate the whole robot by Computer Aided Design and Stress analysis is proposed. The interfacing is achieved by MATLAB with a proper electronic framework for accurate execution. A prototype is also developed as a practical output of this research. This paper also highlights the whole process of mechatronic and software design required to fabricate a robotic arm.

In the article [3], robot trajectories are discussed both in the joint space and the Cartesian space. A new control system is proposed which is tested and validated to confirm the results obtained from this research. With the integration of kinematic analysis, trajectory generation is developed for the simple pick and place operation. This acts as a guide to path planning for the end effector in achieving a required operation.

The article [5] details the Inverse dynamic study of a 3 degree-of-freedom PRR Planar robot. A recursive modeling technique is used in the Kinematic and Dynamic analysis of this robot. This technique is used to find position, orientation and angles of the robot. A control motion comparison of parallel robotics is the main motive of this study with various actuation systems.

The novelty of this article lies in the development of a SCARA CNC drilling machine. Existing research on SCARA Robots have been executed, but not in the CNC framework. A new robotic system is proposed which aims to bring the advantages of automation at a low budget to the small to midrange manufacturing companies. No literature exists on this specific amalgamation of technologies and system. Hence it proposes a novel Industrial automation concept to enhance the reach of robotic systems. The methods used to drastically reduce the cost of such technologies are also explained in the following sections.

3. Robotic Analysis

3.1 DenavitHartenberg (DH) Parameters :

Denavit and Hartenberg (DH) proposed to assign a systematic notation for assigning the orthogonal frames of right hand to each link within an open cinematic chain of ties, using which transformations can be done by 4X4 homogenous transformation matrix between the adjacent coordination frames. Below are the DH parameters of the SCARA robot descriptions [4]:

Table 1: Description of various DH parameters

Arm Parameter	Symbol	Revolute Joint	Prismatic Joint
Joint Angle	ϴ	Variable	Fixed
Joint Distance	D	Fixed	Variable
Link Length	A	Fixed	Fixed
Link Twist Angle	α	Fixed	Fixed

Table 2 : DH parameters for a PRR-type SCARA

Link	α_i	a_i	d_i	Θ_i
1	0	l_1	q_1	0
2	π	l_2	0	q_2
3	π	l_3	0	q_3

3.2 Kinematic and Dynamic Analysis

3.2.1Forward kinematics analysis

Potential kinematics involves using the cinematic formulas of a robot to determine the end-effector location from the values defined for the joint parameters. For robots, computer games and film, the kinematical formulas of the robot are used. The reverse mechanism that calculates the parameters of the joint that achieve a certain end-effector condition is known as reverse kinematics. Here a_i is the length of i^{th} link and is constant whereas $\theta_i$is the joint angle made by i^{th} link and is a variable. $A_i$is the homogeneous transformation matrix that expresses the position and orientation of $o_i x_i y_i z_i$ with respect to $o_{i-1}x_{i-1}y_{i-1}z_{i-1}$.

$$A_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & s\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_1 = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & a_1 c\theta_1 \\ s\theta_1 & c\theta_1 & 0 & a_1 s\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & a_2 c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & a_2 s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_3 = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & a_3 c\theta_3 \\ s\theta_3 & c\theta_3 & 0 & a_3 s\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where c$\theta_i = cos\theta_i$, $c\alpha\alpha_i = cos\alpha_i$,$s\theta_i = sin\theta_i$ and $s\alpha\alpha_i = sin\alpha_i$.

The transformation matrix T_j^ihelps in transforming the position and orientation of$o_j x_j y_j z_j$ to $o_i x_i y_i z_i$ where

$$T_j^i = A_{i+1}A_{i+2} \ldots\ldots A_{j-1}A_j \text{ , if i} < j$$

The transformation matrices for various links are given by

$$T_1^0 = A_1$$

$$T_2^0 = A_1 A_2$$

$$T_3^0 = A_1 A_2 A_3$$

Here T_3^0represent the transformation matrix of end effector frame with respect to base frame

3.2.2 Inverse Kinematics

Inverse kinematics is the problem of calculation of joint angles for a given position of the end effector, unlike that of forward kinematics, which involves the determination of the position of the end effecter for a given set of joint angles.This sub-section discusses an extremely simplified approach used to solve an inverse kinematics problem [4].

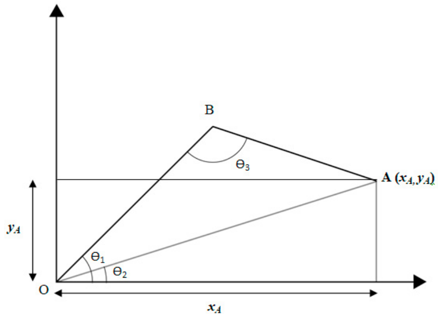

Fig. 2. Inverse kinematics problem.

Let's say one wishes to move to point A (fig. 1) having coordinates (x_A, y_A), then,

$$\theta_2 = \arctan\left(\frac{y_A}{x_A}\right) OA = \sqrt{x_A^2 + y_A^2}$$

Using the above expression, we can calculate $\theta_1$by applying law of cosines if the link lengths *OB* and *BA* are known, as follows:

$$\theta_1 = \arccos\left(\frac{OB^2 + OA^2 - BA^2}{2 * OB * OA}\right)$$

θ_3 can be evaluated similarly:

$$\theta_3 = \arccos\left(\frac{OB^2 + BA^2 - OA^2}{2 * OB * BA}\right)$$

3.2.3 Dynamic Analysis:

Manipulator dynamics is concerned with the equations of motion, the way in which the manipulator moves in response to torques applied by the actuators, or external forces. The equations of motion for an m-axis manipulator are given by,

$$\mathrm{Q} = \mathrm{M}\dot{\vec{q}}\ddot{\vec{q}} + \mathrm{C}(\dot{\vec{q}}\ddot{\vec{q}})\dot{\vec{q}} + \mathrm{F}(\dot{\vec{q}}) + \mathrm{G}(\mathrm{q})$$

Where,

q: vector of generalized joint coordinates describing the pose of the manipulator.
$\dot{\vec{q}}$: vector of joint velocities.
$\ddot{\vec{q}}$: vector of joint acceleration.
M: manipulator tensor for inertia.
C: describes Coriolis and centripetal effect centripetal moments are proportional to q_2, while the Coriolis moments are proportional to q.
F: describes vison's and coulomb friction.
G: the gravity loading Q is the vector of generalized forces associated with the generalized coordinates q.

The Kinematics and dynamics of the manipulator are usually described by a *dyn* matrix in the robotics toolbox, which is the primary feedback justification for dynamic functionality.The analysis process is very straight forward which further strengthens the reason to use this software in addition to MATLAB. The time steps and time for which the analysis is needed can be entered and the Inverse dynamic analysis button does the needful.

RoboAnalyzer™ was used to perform these analyses. The resulting graphs are depicted as follows:

Figure 3. Variation of the x-coordinate of the end of links.

Fig. 3 shows the variation of the x-coordinate of the end of links 2 and 3, while Fig. 4 shows the variation of the y-coordinate of the end of links 2 and 3. Fig. 5 and Fig. 6, a part of the dynamic analyses, depict the variation of the joint accelerations and joint torques.

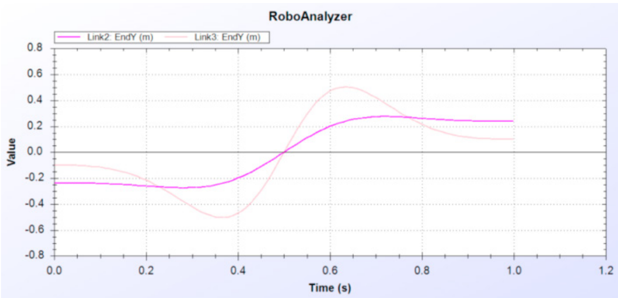

Fig. 4. Variation of y-coordinate of the end of links.

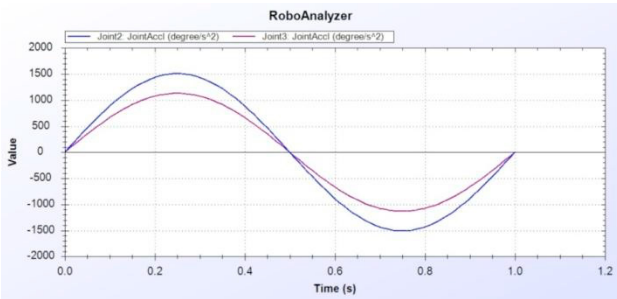

Fig. 5. Variation of joint accelerations.

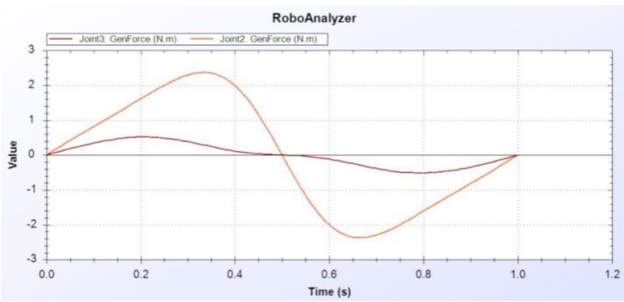

Fig. 6. Variation of joint torques.

With the help of joint torques evaluated from RoboAnalyzer™, the motor selection process was executed [5]. The comprehensive kinematic and dynamic analysis of the robot is important for the development and accurate dimensioning. The approach, as discussed in article [6], represents the same procedure but for a six degree-of-freedom robotic arm.

4. Development

4.1 Computer aided design (CAD) and Analysis

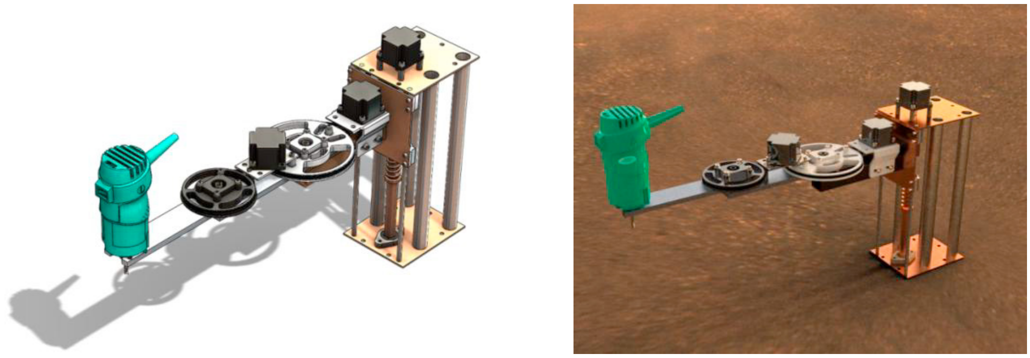

Fig.7. (a) CAD Model of the robot (b) Fabricated prototype of the robot

A CAD model for the components of the robot was developed (Fig. 7). The CAD model development and assembly was done using SolidWorks™2017.The CAD model for the pulleys was developed considering the torque amplification and speed reduction required.A finite element stress analysis was performed on these components (Fig. 8), after selecting a suitable material for them, using ANSYS™ R18.1.

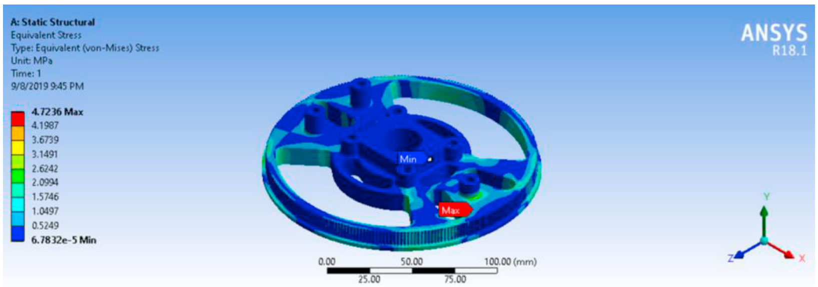

Fig. 8. FEA stress analysis of one of the pulleys

The NEMA motor mounts minimize their weight acting on the arm. Only one NEMA motor is supported by the link and rest of the two motors are supported by the column itself.

4.2 Manufacturing

Several mild steel plates, pipes and rodswere utilized in the robot, which serve different purposes, due to their ease of availability and low cost. The pipes and rods were machined on a Lathe and the plates were machined using Laser Jet Machining (LJM). The leadscrew of a Lathe machine, available as a spare part, was utilized in the robot. The pitch of leadscrew was one inch, and therefore, for one step of the stepper motor there would be an advancement of:

$$\frac{1\,inch * 25.4 * 1.8°}{360°} = 0.127\,mm$$

This gave us an excellent resolution along the z-axis, which can be further improved by half-stepping, quarterstepping and so on. Several commercially available bearingshave been utilized in the robot. The material for the links of the manipulator arm was chosen to be Aluminium, because of its light weight. The Aluminium was machined using a milling machine and a radial drilling machine. The pulleys were manufactured using an additive manufacturing process called Fused Deposition Modelling (FDM). This process was selected because the complex geometry of the pulley. The material used for the pulleys was Polylactic Acid (PLA), by virtue of its higher tensile strength and ductility.The use of commonly available items and materials in the robotic arm is what gives it its low cost.

4.3 Hardware Integration

The circuit diagram (Fig. 9) shows the integration of the hardware elements.

The Arduino Mega 2560 serves as the brain of the entire robotic arm. In the Arduino IDE, there is an inbuilt serial monitor which is used to send and receive information.

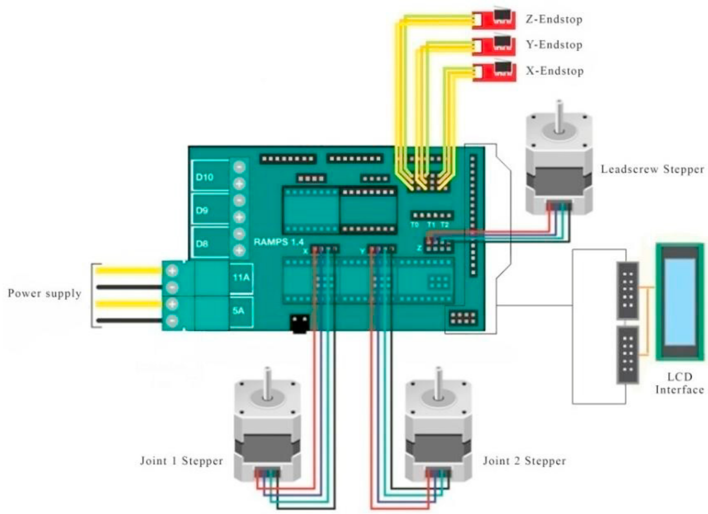

Fig. 9. Circuit diagram.

RAMPS 1.4 that serves as the interface between the Arduino other electronic devices on the robot (such as motors, end stops etc.). It organizes and amplifies the signals coming from the Arduino and directs them to the correct channels. The DRV8825 stepper motor drivers were coupled to the RAMPS 1.4 board in order to achieve stepper control. The NEMA 23, a hybrid stepper motor,was selected after performing a dynamic analysis to calculate joint torques required for the actuation of the manipulator arm. Three motors were utilized in the arm. Two motors control the arm movement in the *x-y* plane, while the remaining motor actuates the arm the *z*-axis. These motors are connected to the DRV8825 motor drivers. Some other components have also been used, for instance, a potentiometer for positional feedback, end stops, and a 128×64 smart LCD controller. Power is supplied to the electronics with the help of a Power Supply Unit (PSU).

Upon entering the G-code into the serial monitor, say G00 for rapid positioning, the G-code is recognized by the Arduino script, which then performs the necessary inverse kinematic calculations to calculate the motor angles required to position the end effector at the fed coordinates. Once these angles are determined, the Arduino actuates the stepper motors through RAMPS 1.4 and DRV8825 motor drivers, and the required angles are achieved.

4.4 Firmware Integration

An Arduino sketch has been created to execute the desired drilling cycle using G-codes and M-codes. The G-Code and M-code is used rather than developing a new system to facilitate its use by manufacturers. As they are habitual in using this coding vernacular, more utility will be gained this way. As of now, the G-Codes given in Table

3 are recognized by the microcontroller.

Table 3: List of G-Codes recognized by the microcontroller

G-Code	Function
G00	Rapid positioning to *x, y, z* coordinate
G01	Execute linear movement along the *z*-axis at a constant feed rate
G20	Dimensioning in ‘inch’ units
G21	Dimensioning in ‘metric’ units
G90	Specifies absolute input dimensions
G91	Specifies incremental input dimensions
G92	Set machine zero point
G94	Feed rate (along *z*-axis) in mm/s
G97	Drill RPM

When the host is a computer, the sketch first looks for a G-Code input to the serial monitor. It contains a *switch* statement to differentiate between different G-codes.

After the required motor angles have been calculated, they are then converted into steps, using the following formula :

$$Number\ of\ steps\ required = \frac{Angle\ (in\ degrees)}{1.8°}$$

For movement along the z-axis, the rotary motion of the stepper motor is converted into a linear motion with the help of a leadscrew. The number of steps required by the motor to achieve the desired z position is calculated as follows:

$$\frac{Steps}{inch} = Leadscrew\left(\frac{Revolution}{inch}\right) * \frac{1}{1.8°} * 200 = K(say)$$

Here, 200 is the number of steps per revolution for a *NEMA 23* stepper motor. Therefore, number of steps required for z mm of movement:

$$Number\ of steps = K * \left(\frac{z}{25.4}\right)$$

These steps are then passed on to the stepper motor through the motor driver and the desired z coordinate is achieved.

Conclusion

Hence, the complete procedure of the development of the CNC Robotic arm is discussed in the paper. The various disciplines involved in the development process from design to firmware integration are elaborated. The aim of this article was to achieve low-cost operation than the existing technologies. The cost reduction is substantial as a mix of additive and conventional manufacturing is used to compensate for cost and stress analysis. Price reduction is observed up to 95% with achievable accuracy rates in the work piece. The firmware is designed to be running on the conventional CNC manufacturing code so as to be easily adapted by manufacturers worldwide. The only limitation experienced is the accuracy of manufacturing, as compared to CNC Machines. This drawback can be corrected by precision manufacturing and higher resolution motors. A further step in the development is the accommodation of a Neural Network to predict the parameters of the drilling operation involved in the process. This robot can be used for milling as well as other purposes, for simplicity purposes, drilling process is concentrated on in this article. The tool used can be varied depending upon work piece materials and the firmware can be tweaked for different manufacturing operations. For further research purposes, the neural network as discussed can be used under uncertainty, as in real life there are imprecise nature in various aspects such as operator accuracy, sensor values, etc. [7]. An improved model can be developed after gaining an operational data of a long duration of time, so as to gain enough data for analysis.

Appendix A. Sample Arduino code to actuate stepper motor for a desired angle

```
#define DIR_PIN 2 // used to select motor drive direction through a motor driver
#define STEP_PIN 3
#define ENABLE_PIN 4
void setup()
{
pinMode(STEP_PIN, OUTPUT);
pinMode(DIR_PIN, OUTPUT);
pinMode(ENABLE_PIN, OUTPUT);
}
void simpleMove(int steps)
{
  int interval = 100;
  for (int i = 0; i< steps; i++)
{
digitalWrite(STEP_PIN, HIGH);
digitalWrite(STEP_PIN, LOW);
delayMicroseconds(interval);
  }
}
void loop() {
digitalWrite(DIR_PIN, LOW); //to make motor turn clockwise
simpleMove(200); //executes 1 revolution clockwise
digitalWrite(DIR_PIN, HIGH); //to make motor turn counterclockwise
simpleMove(200); //executes 1 revolution counterclockwise
  while(true);
}
```

References

[1] Vallinayagam, Sundara. (2018). A GUI Based Kinematic Model Development OF 6 DOF Manipulator Using Matlab.
[2] Urrea, C., Cortés, J., & Pascal, J. (2016). Design, construction and control of a SCARA manipulator with 6 degrees of freedom. *Journal of Applied Research and Technology*, *14*(6).
[3] Chen G., Yang Y., Zhai L., Zou K., Yang Y. (2012) SCARA Robot Control System Design and Trajectory Planning: A Case Study. In: Xie A., Huang X. (eds) Advances in Electrical Engineering and Automation. Advances in Intelligent and Soft Computing, vol 139. Springer, Berlin, Heidelberg.
[4] Soyaslan, Mücahit&Uk, Muhammed & Shah, Faris &Eldogan, Osman. (2018). Modelling, Control and Simulation of a SCARA PRR-Type Robot Manipulator. Scientia Iranica. 10.24200/sci.2018.51214.2065.
[5] Staicu, Stefan. (2009). Inverse dynamics of the 3-PRR planar parallel robot. Robotics and Autonomous Systems. 57. 556-563. 10.1016/j.robot.2008.09.005.
[6] Misra A., Singh G., Kinematic and Dynamic analysis of an Industrial six axis robotic manipulator, International Conference on Robotics, Automation and Non-Destructive evaluation, 2019.
[7] K. Shariatmadar, A. Misra, F. Debrouwere and M. Versteyhe, "Optimal Modelling of Process Variations in Industry 4.0 Facility under Advanced P-Box uncertainty," 2019 IEEE Student Conference on Research and Development (SCOReD), Bandar Seri Iskandar, Malaysia, 2019, pp. 180-185.doi: 10.1109/SCORED.2019.8896326