Optimizing Robotic Arm Precision With Real-Time Machine Vision Feedback

Dr.Anusuya K V[1], Suberus Heartrisha G[2], Praveenbabu R S[3]

{kva.ece@psgtech.ac.in[1], sheartrisha2003@gmail.com[2], praveenbabu4542@gmail.com}

Associate Professor, ECE, PSG College of Technology[1], Scholar, ECE, PSG College of Technology [2] Scholar, ECE, PSG College of Technology[3]

Abstract. Industries play a vital role in the evolution of various innovations, and hardware and robotic arms are essential aids for automation. The main objective of a robotic arm is to perform infinite tasks in a variety of environments and can automate the execution of repetitive tasks. It is essential to update the design of robotic arms to achieve the desired accuracy. Most of the existing robotic arms use coordinate-based working space that results in offset errors in industrial applications where assembly lines are used. In this paper, machine vision is employed to cater to precision needs. On identification of the target site, the arm movements are generated accordingly.

Further processing and movement of the arm synchronize, providing a solution to the offset-based problems. The arm moves to the desired target site with no error or misalignment based on the actions. The two major sections of the arms include Image Segmentation (IS) and Inverse Kinematics (IK) for machine vision and arm movement. These two parts are at the server end, and arm movement is attained through a separate microcontroller. However, the more extensive convergence time towards the desired target site is the limitation. Moreover, the prototype can be improved by using industry-standard tools and fine-tuning the design of processing parts. This prototype can be extended to various applications of high-level assembly line robotic manipulators in industries.

Keywords: Color segmentation, Contouring, Image segmentation, Inverse Kinematics, Machine Vision, Manipulator

1 Introduction

Industrial robotic arms help industries boost their production efficiency and quality by enabling the automation of critical processes that enhance safety for workers, accelerate production, and improve productivity. Many existing robotic arms use Coordinate based working space, designed to repeat the exact routine multiple times. However, this may lead to imprecise manufacturing if there is a

ICSETPSD 2023, November 17-18, Coimbatore, India

DOI 10.4108/eai.17-11-2023.2342833

misalignment in the initial setup of the target. In industries where assembly lines are used, if an offset exists, a coordinate-based system results in errors. To overcome this defect, the current generation of robotic arms uses relative coordinate systems, where the robot takes some measurements to understand its environment. However, this model leads to a drastic increase in the convergence time, which slows down the entire production lane. An Image processing-based system identifies the active site the arm has to reach to overcome this defect.

The objective for proposing a machine vision-based robotic arm includes performing work based on the data collected from the environment as an image. The main aim is to make a simple working robotic arm model for specific industrial applications like soldering, adhesive sealing, etc. The arm is designed with five degrees of freedom and control of the arm, based on the video feed provided by the camera to work on the required work site accurately. The workplace and environment of the job are fed to the robotic arm. The arm is made intelligent by knowing the environment of its work by giving processed data from images captured. This process takes some time to calibrate the arm and start the actual work, increases production time, and reduces efficiency. Machine Vision is critical for site recognition, data analysis, and understanding. The arm performs the key processes even on dislocation of the workpiece. Thus, by doing so, the accuracy of functionalities of the robotic arm is improved to a greater extent at the expense of the convergence time.

The paper is organized as follows. Chapter 2 compares the various works that exist in the field of robotics and kinematics, image processing, and augmented reality. Chapter 3 dwells on the proposed model that will be used to demonstrate this idea. This chapter also covers the workflow and algorithms being used. Chapter 4 deals with the various stages of implementation. Chapter 5 discusses the summary of the results and limitations that were found while analyzing the results and offered recommendations and future scopes for this work.

2 Technical Review

Catering to the misalignment of the coordinate-based system is the paper's target. The two main processes to achieve this target are image segmentation and inverse kinematics. In [1], various types of segmentation of the opaque Image by using filters are discussed. The idea of color-based segmentation is also briefed. The segmentation of an image to understand its environment and features is discussed by considering the pixel level through various methods. Of all the methods discussed, threshold-based segmentation is the main key idea for the arm.

In [2], the implementation of inverse kinematics to an arm is discussed along with the control module of the arm. The main key features discussed are the Robot arm kinematics and the inverse kinematics analysis. Under that, the control module of the robotic arm is the vital factor considered. The arm structure expression through the DH parameter table and the formation of the table by trigonometric and geometric methods are studied.

Paper [3] discusses the integration of augmented reality in arms control.

Computer vision is employed for high-level control of the arm. The main focus is combining augmented reality for the computer and high-level control of arm integration. The description of the arm manipulator, the subjects under consideration, the location of the image-capturing element for the implementation of computer vision, the type of image processing, the integration of the processing, and the movement due to it are all the main factors covered.

Image processing for the arm's working [1,4,5,10] is the first step of the implementation. The segmentation of an image to understand its environment and feature at the pixel level is the core idea of processing the Image. Various types of segmentation of the opaque Image by using different filters [1] is the theory behind image segmentation.

The entire control module of the robotic arm utilizing both image processing and inverse kinematics is concurrently [3]. The integration of the individual processes that include the description of the arm manipulator, the subjects under consideration, the location of the Image capturing element for the implementation of computer vision, the type of image processing, the integration of the processing, and the movement due to it and tackling the intermediate delays provide the overall working of the proposed work.

The application consists of many unique hardware and software modules. Reviewing previous works in this area is necessary to choose reliable hardware and trustful software algorithms. The occurrence of positional error arises from inaccuracies within the entire robotic system. The reasons for error occurrence [6] considered are kinematic errors and dynamic errors, which are not accounted for in the dynamics. Thus, Catering to the misalignment of the coordinate-based system [6] is the target of this work. Regular maintenance and calibration are necessary for a coordinate-based robotic system employing forward or inverse kinematics. If the robotic arm undergoes a shift, it will deviate from its predetermined movement trajectory. The primary requirement for accomplishing this work is analyzing the end effector position concerning the target. The study on the position of the arm is done based on computational methods [7].

The design and implementation of the arm based on various technical parameters [8], such as the DH table, Jacobian transformation matrices, and validation equations, are utilized for this work. The implementation of inverse kinematics along with the control module of the arm [9] is the second part of this work. From the positional coordinates obtained through image processing, the angles for all the rotational elements are computed by inverse kinematics. The working of the arm realized by image processing and identifying the target accordingly [9] is the crucial flow of this work. The overall flow can be accomplished by the two main processes, image segmentation, and inverse kinematics. Threshold-based segmentation [10] in multiple color spaces is essential for this work, as the target site is identified based on a marked color, segmenting the desired color to identify the target's position. From the global scope, identification of the local site [10] is the crucial reason for image segmentation at the pixel level.

3 Proposed model

The proposed work uses a scaled version of a 5-DoF articulated robotic arm as a testing platform. The robotic arm with five degrees of freedom was designed to cover all the joints that resemble the base, shoulder, elbow, tool pitch, and tool roll and consist of only rotary joints—controlling the arm based on the video feed provided by the cameras to work accurately on the required worksite and checking the quality of work through machine vision, for video capturing, analyzing, and finding the work site. A camera is used to capture the work environment.

The microcontroller is responsible for the arm's movement and is precise working. A separate server is considered for processing the video feed and obtaining the required angular coordinates for the arm based on the feedback from video processing. The server can also act as a hub, which can pave the way for multiple arms to be connected and operated. The processed data from the server is transmitted to the arm using communication protocols, here, Serial Communication with the microcontroller.

3.1 Design Parameters

The design considerations of the arm can be viewed as the Denavit - Hartenberg Parameter table for robotic arms. This method is an alternative for finding homogeneous transformation matrices. The homogeneous transformation matrices can express the position and orientation of the end effector frame (solder head) in terms of the base frame. To calculate the homogeneous transformation matrix from the base frame to the end effector frame, the only values considered are the length of each link and the angle of each servo motor.

Number of Rows of the table = Number of Frames of the end effector – 1 (1)

Number of Columns of the table = 2 columns for rotation and 2 columns for displacement (2)

Lo = 48 mm, L1 = 42.72mm, L2 = 120mm, L3 = 95mm, L4 = 28mm, L5 = 155mm

The DH table for the Five degrees of freedom is Table 1. Figure 1 and Table 1 provide the significant metrics of the used robotic arm. They are the essential parameters for the arm's operation and navigation to its desired end-effector position.

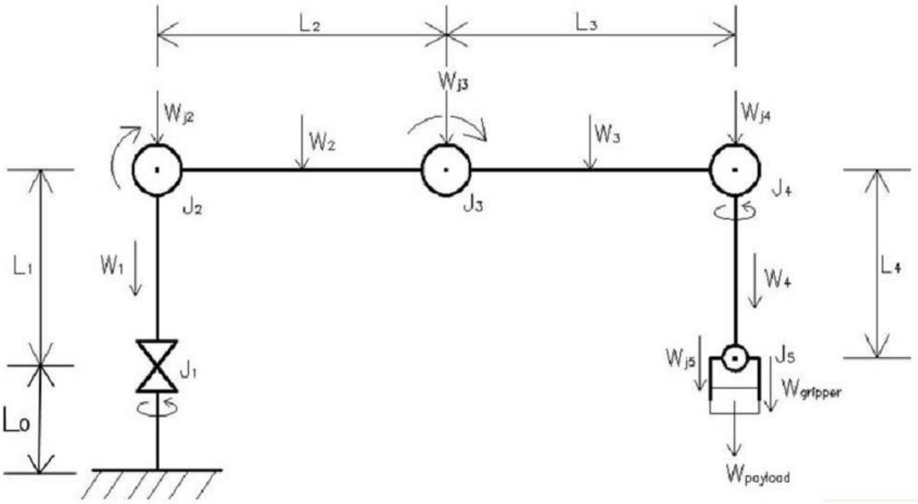

Fig. 1. Free body diagram of 5DoF robot arm

Table 1. DH value table of the arm

Frame(i)	a_i	α_i	d_i	θ_i
1	1.374	90	0	[0,180]
2	12	0	0	[0,170]
3	0	90	0	[0,135]
4	0	90	11.965	[0,180]
5	15.5	0	0	[0,180]

3.2 Proposed Algorithms

From the mechanical perspective, the robotic arm is designed with 5 DOF with the required dimensions, as shown in Table 1. Initially, the basic working of the arm is implemented using forward kinematics, and the working is tested. To test the image processing part, the USB Camera is fixed at the end effect of the arm at a specific angle to focus the end effector's edge location. From the video feed obtained, the Image is processed. Here, color segmentation is based on the frame size and desired color limit. The segmentation output is the pixels of only the desired color. Secondly, the contouring of the segmented Image is done. Many contours are obtained for the segmented Image. Then, the center of the contours is obtained. The desired work area is obtained from all the obtained contours by considering the area with the maximum contour. The maximum contour defines the job site. Thirdly, the difference between the coordinates of the center of the frame and the center of the maximum contour is obtained since the end effector is aligned with the center of the captured frame. Fourthly, the inverse kinematics of the robotic arm is implemented using the Tinyik package of Python. The difference in coordinates obtained is given to the Tinyik module, and the respective angles for the movement are obtained. The data is transmitted to the microcontroller via a serial port and fed to the robotic arm.

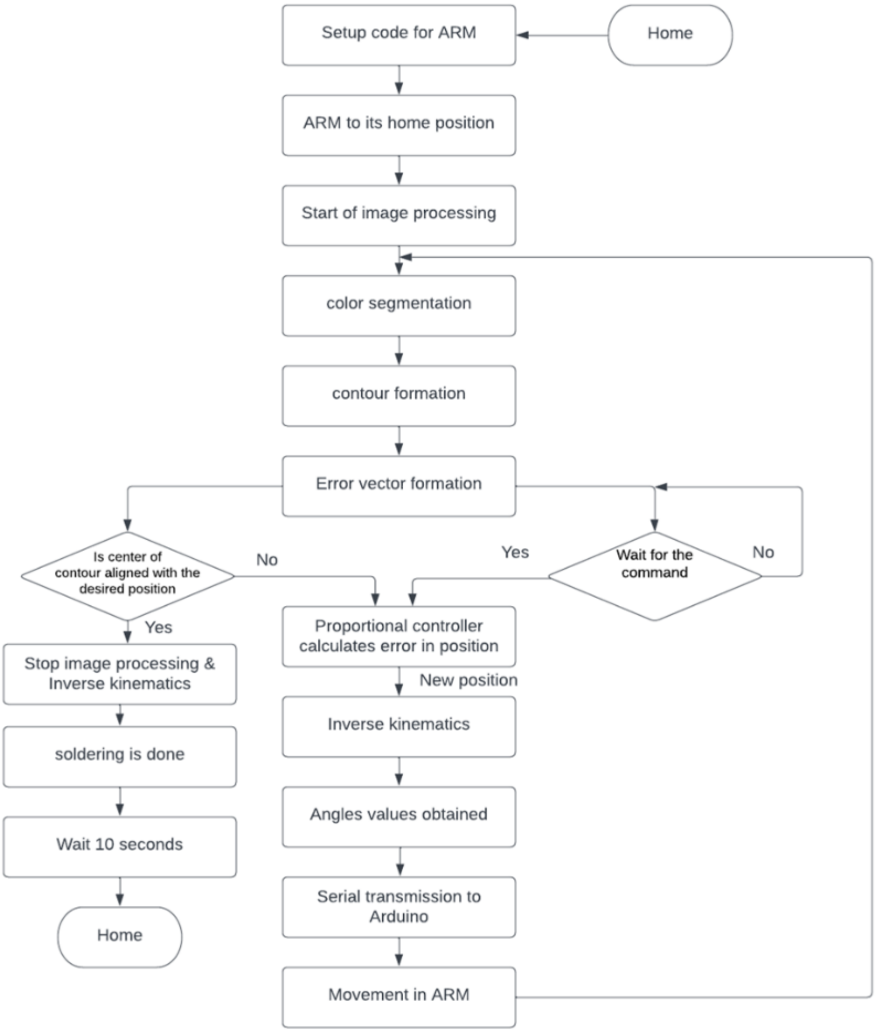

Fig. 2. Workflow diagram of the proposed model

The flow diagram in Figure 2 explains the abstract-level implementation. Modular-level implementation is considered to consolidate the overall work. Fig x describes the arm's integrated working of hardware and software modules.

The algorithms developed for robotic arm control are,

- Proportional controller with adaptive gain
- Image Processing using OpenCV
- Inverse kinematics

3.3 Arm Control

The primary processes involved in arms control are image processing and inverse kinematics.

Image Processing

The captured frame from the camera is processed, and the target position is deduced from the captured frame. Hence, the obtained dynamic feed is processed for every frame, and the arm movement is calculated accordingly. In the processing of images, color detection, and contour segmentation are considered.

Color Detection

Under color detection, the desired color applied to the target site is detected from the entire frame. On detecting the required color, the identification and separation of the pixels are made. For the target site identification, the color red is considered. The parameters considered for color detection are the lower and upper bounds for the Hue Saturation Value (HSV) values.

For the color red,

Lower bound = [0, 120, 120]

Upper bound = [5, 255, 255]

Contour Segmentation

The second image processing stage is contour segmentation, where the contours are drawn for the required color detected from the previous process. Boundaries for the color detected are drawn. Multiple contours are drawn initially due to the presence of multiple target sites. An approximation is considered to detect the actual target site. The contour of the maximum area is considered as the actual target site, as only a single target implementation is considered. On choosing the contour with the maximum area, the error vector is drawn from the center of the captured frame to the center of the maximum area contour. Hence, this error vector forms the basis for implementing the inverse kinematics module.

Inverse Kinematics

Inverse kinematics involves finding the joint angles or parameters that will achieve a specific position and orientation for the end effector of a robot. The desired job site is captured dynamically for analyzing the target position. Hence, the position to which the arm should move from the initial processing phase is obtained. The inverse kinematics is implemented from the Image processed data. The arm's physical parameters, the end-effector's current position, and the possible movement range of angles for each motor, in addition to the desired action site, are considered for the angle generation. This module finds the angles to which the arm joints must move to reach the desired position.

4 Implementation

The soldering of electronic components using a robotic arm is the application selected. The end-effector is fitted with a soldering iron, and the camera is placed such that the tip of the iron is visible in the frame. The work site is marked on the PCBs with the proposed color and size.

Initially, a virtual model of the arm is created in SOLIDWORKS based on the design parameters of the arm, and it is imported into MATLAB's Robotics Toolbox. Simulation of the model is done to analyze various cases of arm movement. The SOLIDWORKS design is shown in Figure 3, and the MATLAB simulation of the 5DoF robot arm is shown in Figure 4. The hardware schematic diagram is depicted in Figure 5. After completing simulations, the Robotic Arm is fabricated, and the hardware is installed as depicted in Figure 6.

The hardware modules and software libraries used are given in Tables 2 and 3.

Fig. 3. SOLIDWORKS model designed for 5DoF robot arm

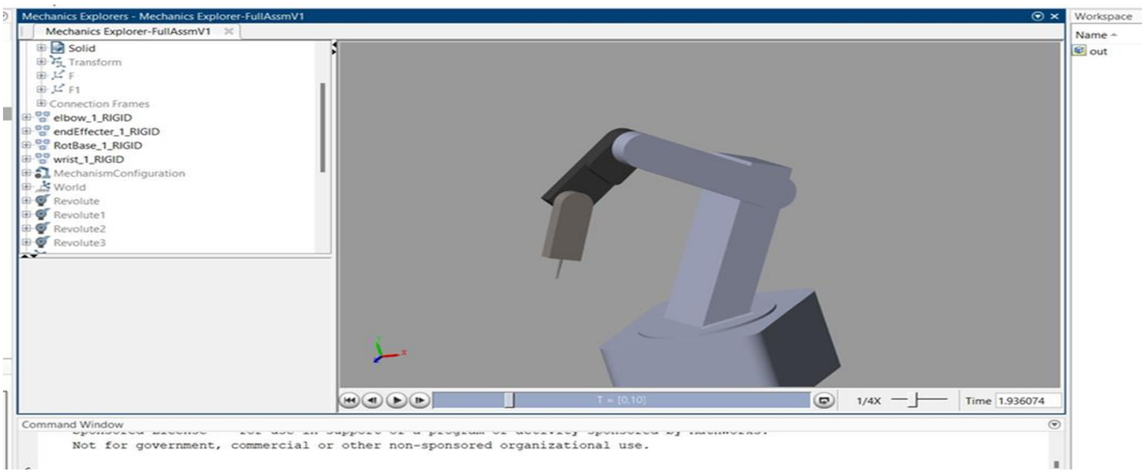

Fig. 4. Simulation of 5DoF robot arm in MATLAB

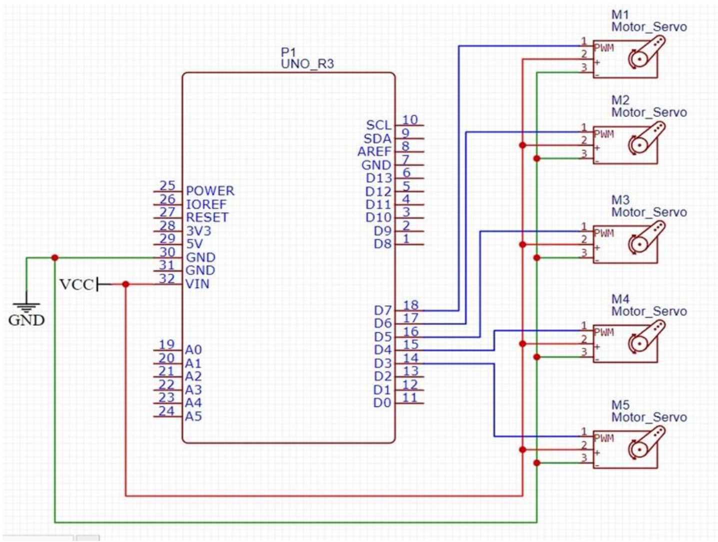

Fig. 5. Schematic of hardware implementation

Table 2. Hardware modules used

Microcontroller	Actuators	Sensing element
Arduino uno R3	SG90s servos MG995 servos Soldering iron	¼ CMOS 640*480 USB Camera

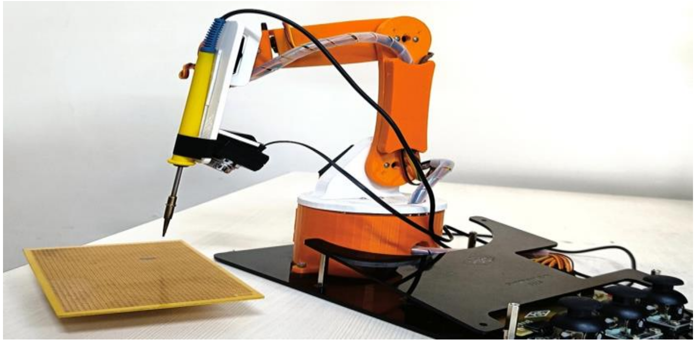

Fig. 6. Fabricated hardware setup

Table 3. Modules Required at the processing site

Library	Usage
Tinyik	Tinyik is a simple inverse kinematics solver package for Python. Given a robotic arm as a series of linkages, inverse kinematic calculations are performed for a given coordinate, and it outputs the required desired
OpenCV	It allows us to perform image processing and computer vision tasks.

4.1 Simulation of Inverse Kinematics

From the inverse kinematics simulation, the initial position and the arm's position change according to the camera feed are shown in Figures 7a and 7b.

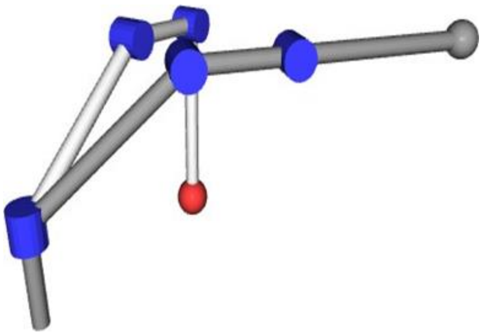

Fig. 7(a) Initial position of the arm

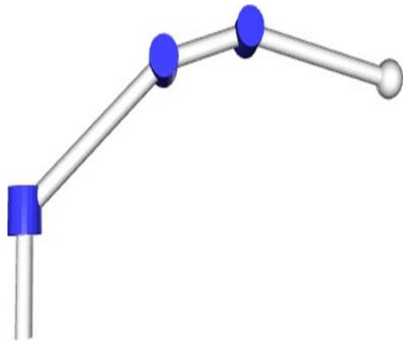

Fig. 7(b) Arm position changed based on the camera feed

4.2 Simulation Results

The simulation results of machine vision in detecting the contouring sites, identifying the active target, and the error vector formed are shown in Figures 8a, 8b, and 9.

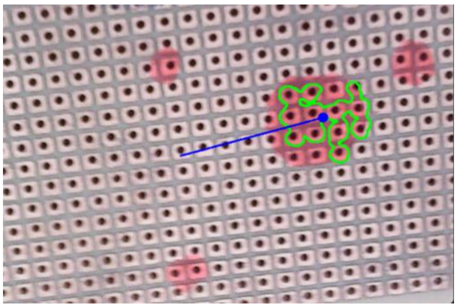

Fig. 8(a). Contouring of the sites detected

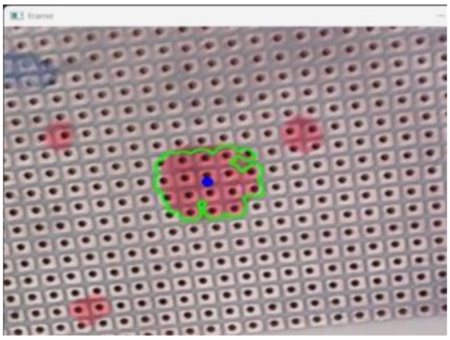

Fig. 8(b). Active target site identification

4.3 Hardware Result

The Image of the board is captured using the camera and fed to the server where the image processing is performed. The specified color is identified from the image frame, and those pixels are passed to the contouring. From all the contoured sites, the work site is determined by comparing the areas of the contours. The selected contour's center is compared with the center of the frame. The error vector is formed and given to the proportional controller to determine the target position.

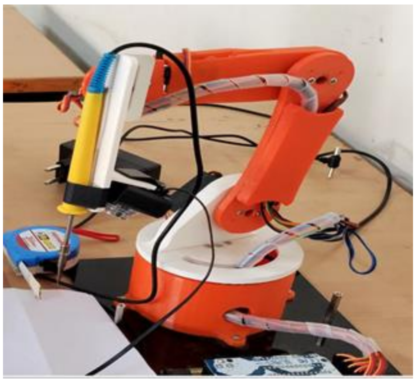

Fig. 10. Initial Arm position

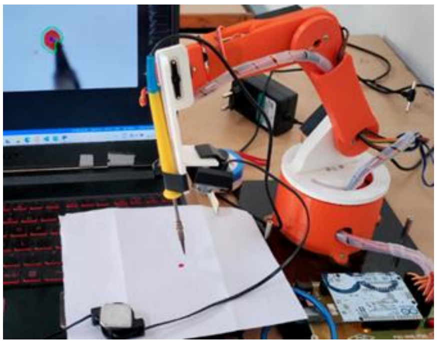

Fig. 11(a). Arm changed position

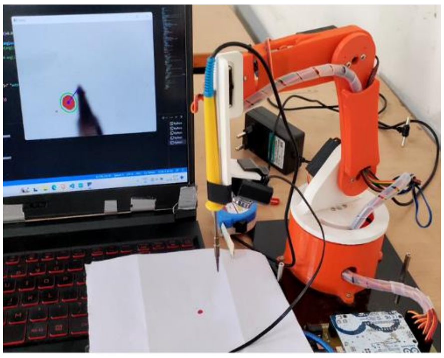

Fig. 11(b). Arm final position

The obtained positional values are passed to the inverse kinematics Tinyik module to obtain the joint angles. The joint angles are serially transmitted to the robotic arm controller. The controller processes the obtained data and controls the servos such that the end-effector reaches the desired position. Hence, for every arm movement, both processes (Image Processing and Inverse Kinematics) continue until the arm reaches the desired target position. The testing shows that the Convergence time of the arm to its target position takes around 1 minute 43 seconds.

5 Summary

Thus, machine vision-based robotic arms would revolutionize the manufacturing industry in the sector of automation by improving accuracy, speed, and productivity. With the help of advanced machine vision technology, robotic arms can detect and process visual information in real-time, allowing them to make precise and accurate movements. Machine vision-based robotic arms have many applications, including assembly, packaging, sorting, and inspection tasks. The limitations of the proposed work include the following.

- The movement of the arm determined by the control algorithm is implemented using a proportional controller, and it causes overshoots in the motion of the arm, thereby affecting the convergence time to a significant extent. The consideration and optimization of convergence time are necessary factors. This limitation can be rectified by implementing a Proportional Integral Differential (PID) Controller.

- This implementation considers only a single job site, thereby limiting the presence of multiple job areas for a single application.

The scope of the proposed work can be enhanced with the following features.

- Modularization of the head can pave the way for implementing various applications using a single manipulator. Conversion of the server to a central hub can also be done to implement various applications using multiple manipulator arms.

● Implementing Machine Learning (ML) techniques enable the higher scope to identify the work site in the view of various applications, enabling an individual arm to function for multiple industrial applications.

● Integration of Industrial IoT to the Arm can be done to access, control, and analyze the data from remote sites.

Overall, the benefits of machine vision-based robotic arms are likely to improve in the upcoming years as technology continues to evolve.

References

[1] Awf Abdulrahman and Serkan Varol, “A Review of Image Segmentation Using MATLAB Environment”, 2020 8th International Symposium on Digital Forensics and Security (ISDFS), pp. 1-5, 2020.

[2] M. Gül, "Modeling of Inverse Kinematic Analysis of Open-Source Medical Assist Robot Arm by Python", Celal Bayar University Journal of Science, vol. 17, no. 1, pp. 31-34, Dec. 2020, doi:10.18466/cbayarfbe.776697

[3] X. Chen, X. Huang, Y. Wang, and X. Gao, "Combination of augmented reality-based brain-computer interface and computer vision for high-level control of a robotic arm", IEEE Trans. Neural Syst. Rehabil. Eng., vol. 28, no. 12, pp. 3140-3147, Dec. 2020

[4] V. Kumar, Q. Wang, W. Minghua, S. Rizwan, S. M. Shaikh and X. Liu, "Computer vision-based object grasping 6DoF robotic arm using picamera," 2018 4th International Conference on Control, Automation and Robotics (ICCAR), Auckland, New Zealand, 2018, pp. 111-115, doi: 10.1109/ICCAR.2018.8384653.

[5] Shaikat, A. S., Akter, S., & Salma, U. . (2020). Computer Vision-Based Industrial Robotic Arm for Sorting Objects by Color and Height. Journal of Engineering Advancements, 1(04), 116–122. https://doi.org/10.38032/jea.2020.04.002

[6] Angelidis, A., Vosniakos, GC. Prediction and compensation of relative position error along industrial robot end-effector paths. Int. J. Precis. Eng. Manuf. 15, 63–73 (2014). https://doi.org/10.1007/s12541-013-0306-5

[7] Szabo R, Ricman R-S. Robotic Arm Position Computing Method in the 2D and 3D Spaces. Actuators. 2023; 12(3):112. https://doi.org/10.3390/act12030112

[8] J. Iqbal, R.U. Islam, H. Khan, “Modeling and analysis of a 6 DoF robotic arm manipulator”, Can. J. Electr. Electron. Eng. 3 (6) (2012) 300–306.

[9] T. P. Cabré, M. T. Cairol, D. F. Calafell, M. T. Ribes and J. P. Roca, "Project-Based Learning Example: Controlling an Educational Robotic Arm With Computer Vision," in IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, vol. 8, no. 3, pp. 135-142, Aug. 2013, doi: 10.1109/RITA.2013.2273114.

[10] Tang Y, Chen M, Wang C, Luo L, Li J, Lian G, Zou X. “Recognition and Localization Methods for Vision-Based Fruit Picking Robots: A Review”. Front Plant Sci. 2020 May 19;11:510. doi: 10.3389/fpls.2020.00510. PMID: 32508853; PMCID: PMC7250149.