automation

MDPI

Article

A Smart Four-DOF SCARA Robot: Design, Kinematic Modeling, and Machine Learning-Based Performance Evaluation

Ahmed G. Mahmoud A. Aziz [1,*], Saleh Al Dawsari [2,3,*], Amr E. Rafaat [4], Ayat G. Abo El-Magd [1] and Ahmed A. Zaki Diab [5,6,*]

[1] Electrical and Computers Engineering Department, Higher Institute of Engineering and Technology, New Minia 61784, Egypt; ayat.abo_elmagd@mhiet.edu.eg
[2] School of Engineering, Cardiff University, Cardiff CF24 3AA, UK
[3] Electrical Engineering Department, College of Engineering, Najran University, Najran 1988, Saudi Arabia
[4] Mechatronics Engineering Department, Higher Institute of Engineering and Technology, New Minia 61784, Egypt; engamremad89@mhiet.edu.eg
[5] Electrical Engineering Department, Faculty of Engineering, Minia University, Minia 61111, Egypt
[6] Department of Mechatronics Engineering, Faculty of Engineering, Nahda University, Beni Suef 62764, Egypt
* Correspondence: a.g.mahmoud@mhiet.edu.eg (A.G.M.A.A.); aldawsarisa@cardiff.ac.uk (S.A.D.); a.diab@mu.edu.eg (A.A.Z.D.)

Abstract

Robotics is increasingly used in higher education laboratories, but most commercial robots are costly and designed for industrial use. This paper presents the design, modeling, and experimental evaluation of a low-cost four-degree-of-freedom (DOF) SCARA robot for educational and research purposes. The robot supports pick-and-place and laser engraving tasks. Direct and inverse kinematics were developed using Denavit–Hartenberg parameters, and the mechanical structure was validated through the dynamic analyses. A new machine learning (ML) framework integrating Support Vector Machine (SVM) and Random Forest (RF) models was implemented to enhance motion precision, predict task success, and compensate positioning errors in real time. Experimental tests over 360 cyles under varying speeds, payloads, and object types show that the SVM predicts grasp success with 94.4% accuracy, while the RF model estimates XY positioning error with an RMSE of 1.84 mm and cycle time error with an RMSE of 0.41 s. Moreover, a novel approach in this work that combines it with a laser engraving machine has been suggested. Repeatability experiments report 0.97 mm ISO-standard repeatability, and laser engraving trials yield mean positional errors of 0.45 mm, with maximum deviation of 0.90 mm. Compared to a baseline PID controller, the ML-enhanced strategy reduces RMS positioning error from 3.30 mm to 1.83 mm and improves repeatability by 36.5%, while slightly decreasing cycle time. These results demonstrate that the proposed SCARA robot achieves high-precision, consistent, and flexible operation suitable for both academic and light-duty practical applications.

Keywords: SCARA robot; simulation; drives; machine learning; kinematic

Check for updates

Academic Editor: Shuai (Steven) Li

Received: 21 October 2025
Revised: 13 December 2025
Accepted: 15 December 2025
Published: 1 January 2026

1. Introduction

These days, technology is developing in parallel with rapidly increasing human demands. Robotic arms are increasingly used to meet those demands and to perform tasks previously performed by humans in modern industries that require high-precision. They can handle various tasks, explore their surroundings, and interact with users thanks to their actuators, sensors, and control software. Over the past three decades, assembly automation

https://doi.org/10.3390/automation7010011

has advanced significantly with the development of powerful computing and robotics technologies. Robotic arms can be controlled by an external operator or by following preset instructions. Medical and industrial sectors now have the most advanced robot arm technology [1]. Robotic control has become a major area of interest for numerous academics due to its increasing capacity [2].

The Selective Compliance Assembly Robot Arm (SCARA) is an industrial manipulator designed with horizontal joints that enable fast motion within the X–Y plane and controlled vertical movement along the Z axis. Its structure supports efficient pick-and-place operations (PAPOs) and provides a circular working area suitable for high-speed planar tasks. This configuration gives the robot a high level of compliance in the horizontal plane, allowing its arm to adjust naturally during assembly and achieve accurate positioning [3]. SCARA robots offer four-degrees-of-freedom (4-DOF) that enable fast planar motion and controlled vertical rotation, making them suitable for precise and repeatable tasks. Thanks to their speed, accuracy, and compact structure, they are commonly used in industrial and educational applications, especially for efficient PAPOs. SCARA robots have significantly accelerated assembly operations in industrial settings. Their design supports fast, precise handling of small components, making them suitable for applications ranging from automotive manufacturing to electronics production. With proper sealing, their joints can operate underwater, and their ability to be controlled remotely makes them valuable for tasks that pose risks to human workers, such as chemical handling or work in high-temperature environments.

The first SCARA robot was developed in 1979 under the supervision of Professor Makino at Yamanashi University [4,5]. Since then, SCARA robots have been widely adopted in industry, especially for PAPOs. Numerous models have been designed and produced by researchers and are now commercially available [2,6–11]. For instance, Bhatia proposed a SCARA robot design based on an expert system [8]. A Jacobian model for a four-axis domestically produced SCARA robot was created by Manjunath [12]. In the meantime, Omari concentrated on creating a delicate motion mechanism for a very precise mounting SCARA robot system [10]. Some studies focus on making SCARA robots easier to control. Ali, for example, used a polytopic linear parameter-changing technique to study the H_∞ control of a SCARA robot [7]. Recently, the integration of machine learning (ML) techniques into robotic systems has become a major trend in modern automation. In PAPOs, which require high-precision, dynamic adaptation, and real-time decision making, ML algorithms provide significant advantages over traditional control methods [13,14]. Intelligent models are able to anticipate ideal motion trajectories, identify anomalies, and adjust to changing payloads or workspace circumstances by learning from sensory inputs such as position, force, and visual-feedback. In the case of the 4-DOF SCARA Robot, ML-based control can enhance path optimization, minimize positioning errors, and improve overall cycle time efficiency. Furthermore, deep learning (DL) models like Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks [15], as well as ML algorithms like Support Vector-Machines (SVMs) and Random Forest (RF), can be used to classify different kinds of objects, optimize grasping parameters, and guarantee reliable performance in unstructured environments [16,17]. By combining data-driven intelligence with traditional kinematic control, the SCARA robot is transformed from a pre-programmed manipulator into an industry-ready, self-improving system. Sensor technology, particularly vision sensors, has greatly enhanced robot capabilities [2,12]. Modern robots can operate in both structured and unstructured environments by sensing their surroundings. Machine vision systems capture images of target objects and process them to extract key information, which is then sent to the robot for motion control [13–16]. For instance, in an automated PAPO, cameras detect object positions and the robot arm moves to grasp and relocate them.

Vision-based robots have been increasingly applied in various industrial automation tasks due to their ability to adapt to changing environments [18–21]. Classification, detection, and identification are common challenges in automated production [3,22]. Using cameras for classification offers advantages such as consistent accuracy, speed, and repeatability. Various image processing methods have been developed to distinguish different objects. Among them, the SVM is widely applied for classification tasks and has often been found to outperform ANNs [23].

Laser engraving removes thin layers of material from a surface using a focused laser beam to create visible patterns or text. The interaction among the laser and the material depends on both the type of material and the laser's properties. This process requires high-precision and accuracy to produce detailed designs or inscriptions. However, most commercial laser engravers are rigid and lack flexibility, limiting their ability to adapt to different patterns or materials. Therefore, a more versatile and adjustable laser engraver is needed to efficiently handle a wide variety of designs across multiple materials. Vo and Hiep's study introduced an object classification technique for a robotic arm based on SVM. With a basic mechanical setup and no real-time motion optimization or in-depth kinematic analysis, their work was primarily focused on picture recognition and sorting tasks [16]. In contrast, the current work achieves greater adaptability in PAPOs by combining ML with thorough kinematic modeling and experimental validation on a 4-DOF SCARA robot. Li et al. [14] developed and tested a high-speed parallel-robot. Although the research included real mechanical implementation and performance testing, it did not employ ML or laser-based processes. Moreover, its kinematic structure differs significantly from the SCARA configuration adopted in this study, which provides a simpler and more flexible framework for industrial PAPOs. In the work of Han et al. [15] on multimodal intelligent-logistics-robots combining 3D CNN, LSTM, and visual SLAM, the main emphasis was on perception and navigation for logistics applications. While the study successfully integrated DL for environment understanding, it lacked a detailed analysis of manipulator kinematics or end-effector control. The current research instead integrates both mechanical modeling and intelligent learning, bridging the gap among motion control and perception. The review Artificial Intelligence (AI), ML, and DL in advanced robotics [17] outlines the progression of AI methods in robotics but remains largely theoretical, offering no practical implementation or experimental validation. In contrast, the current work provides a complete design-to-deployment pipeline on an actual SCARA system, demonstrating clear and measurable performance gains. Likewise, Jahanshahi [13] presents a conceptual survey of ML techniques for robotic grasping in space applications. This research focused primarily on the challenges of controlling the robot in microgravity conditions without attention to terrestrial PAPOs or kinematic derivations. Finally, the work entitled "Trajectory tracking of a SCARA robot using intelligent active-force control" [24] investigated motion accuracy improvement using AI-based controllers. The study included experimental results on a SCARA robot, yet did not address laser processing or provide the comprehensive kinematic derivations presented in our work. The incorporation of detailed modeling, enhanced PAPOs, and the introduction of a new laser engraving approach clearly sets this work apart from recent SCARA-related studies. Although SCARA robots are widely available, most commercial systems remain expensive and primarily designed for industrial environments, limiting their use in academic settings. Inexpensive and safe models are essential for enabling students to engage practically with core robotics topics such as kinematics, dynamics, sensing, design, and control. In response to this need, the present study introduces a low-cost, 4-DOF SCARA robot developed for teaching, research, and demonstration purposes. The system includes a MATLAB-based kinematic model constructed using Denavit–Hartenberg parameters, supported by workspace and schematic analyses to facilitate understanding and validation. The following are the primary contributions of this paper:

- Based on robot architecture, a SCARA robot model is produced by analyzing its workspace, generating a schematic diagram and matching DH characteristics, and creating a kinematic model in MATLAB with related reference frames.
- A low-cost four-DOF SCARA robot's design and development process, including conceptual design, specification definition, product development, and testing.
- A mechanical structure for a 4-DOF SCARA robot that can carry out specific duties for demonstration reasons like research, educational, and exhibition purposes like PAPOs has been designed and built.
- The integration of SVM and RF algorithms for optimizing task precision and motion efficiency in the SCARA robot. The results demonstrate that the AI-based control improved the robot's accuracy and repeatability in pick-and-place operations.
- Experimental validation of the robot's performance, including laser engraving tasks.
- The system's technical features have been examined, including the assembly and calibration procedures, the engraving's accuracy and precision, and the safety precautions.

The remainder of the paper is organized as follows. Section 2 presents a thorough study of SCARA robot kinematics and provides a solution to the proposed robot's inverse kinematics problem via MATLAB. Section 3 introduces, in detail, the implementation steps of the proposed robot, starting from mechanical assembly, through control and electronic configuration, and ending with the robot testing phase in PAPOs. Section 4 presents laser engraving with SCARA robot configuration with a detailed presentation of the proposed robot electronic configuration and ends with a presentation of the practical results of the implemented robot.

Figures 1 and 2 illustrate the design-to-implementation workflow and the final suggested SCARA robot prototype, respectively. The system allows students and researchers to gain hands-on experience, bridging theory and practical robotics applications.

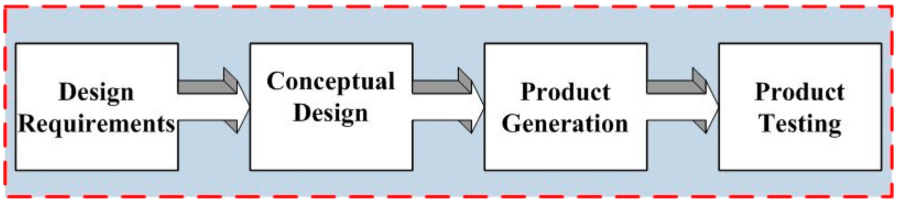


Figure 1. Design and implementation flow of SCARA robot.

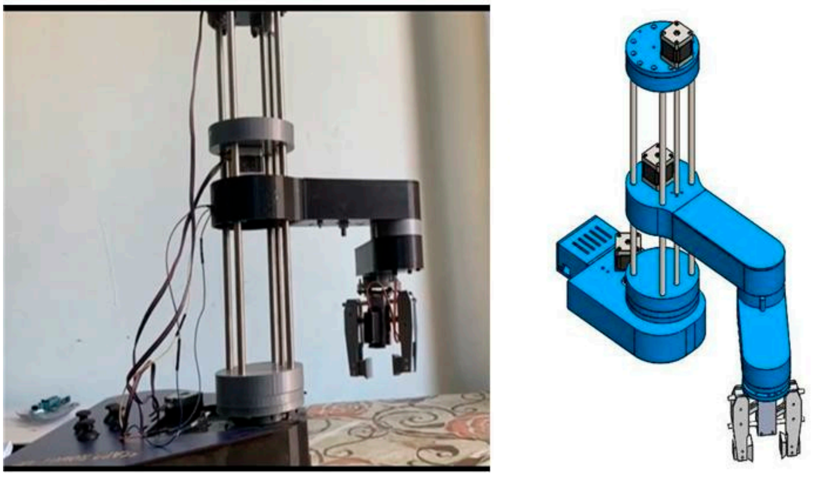


Figure 2. SCARA robot SolidWorks model design and its completed implementation.

Figure 2 illustrates the finalized SCARA robot, highlighting the improvements made during the product development phase.

2. SCARA Robot Kinematics Analysis

A compact, customized SCARA robot was developed to perform tasks within the targeted operational environment. Its Denavit–Hartenberg (DH) parameters were derived directly from the robot's kinematic structure. Using these parameters, a MATLAB 2023b-based kinematic model was built to enable detailed workspace analysis and motion simulation [19,21]. With this particular customized architecture, these derived kinematic models offer a crucial technical reference for putting into practice efficient motion control strategies for SCARA robots. Table 1 lists the conventional DH characteristics for robots, which are derived from the manipulator kinematics.

Table 1. SCARA robot DH parameters.

Link	a (mm)	α (Deg)	d (mm)	θ (Deg)
L0	0	0	d_1 = 50 (fixed)	θ_1
L1	220	0	0	θ_2
L2	200	0	d_3 (prismatic)	0
L3	0	0	d_4 = 40 (fixed offset)	θ_4

The simulated kinematic model of the SCARA robot is shown in Figure 3 after these DH parameters are implemented in the MATLAB environment using the traditional DH modeling approach and joint motion constraints [18,19].

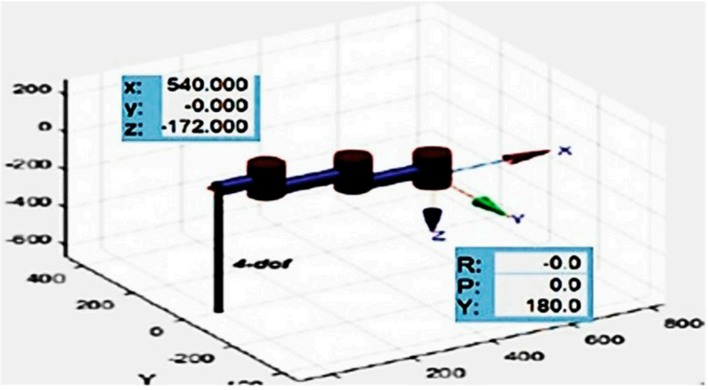


Figure 3. Kinematic SCARA robot modeling.

To confirm the accuracy of the inverse kinematic equations, computational modeling in MATLAB allows the development of transformation matrices and related manipulator postures during subsequent SCARA kinematic analysis.

The screw theory and homogeneous transformation method are the most often used approaches for describing end-effector posture in relation to the robot base frame. The SCARA robot's forward and inverse kinematics were developed using homogeneous transformation matrices, which provide a simple and computationally efficient method for pose description in this robot's straightforward kinematic structure. While screw theory is another approach, it is better suited for complex manipulators with multiple degrees of freedom [11,19,21]. The DH transformation matrices for every joint can be found to solve direct and inverse kinematics once the DH parameters have been determined, as illustrated below.

$$^{0}T_{1} = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & a_1C\theta_1 \\ S\theta_1 & C\theta_1 & 0 & a_1S\theta_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$^{1}T_{2} = \begin{bmatrix} C\theta_2 & S\theta_2 & 0 & a_2C\theta_2 \\ S\theta_2 & -C\theta_2 & 0 & a_2S\theta_2 \\ 0 & 0 & -1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$^{2}T_{3} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$^{3}T_{4} = \begin{bmatrix} C\theta_4 & -S\theta_4 & 0 & 0 \\ S\theta_4 & C\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

When all of the transformation matrices for every joint are combined, the SCARA robot's overall transformation matrix looks like this:

$$^{0}T_{3} = \begin{bmatrix} C\theta_1C\theta_2 - S\theta_1S\theta_2 & -C\theta_1S\theta_2 - S\theta_1C\theta_2 & 0 & a_2(C\theta_1C\theta_2 - S\theta_1S\theta_2) + a_1C\theta_1 \\ S\theta_1C\theta_2 - C\theta_1S\theta_2 & -S\theta_1S\theta_2 + C\theta_1C\theta_2 & 0 & a_2(S\theta_1C\theta_2 + C\theta_1S\theta_2) + a_1S\theta_1 \\ 0 & 0 & 1 & -d_3 + d_1 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

The SCARA robot's kinematics analysis can be carried out using the transformation matrix as a guide.

2.1. SCARA Robot Forward Kinematics

To determine the mapping between the joint displacements and the end effector's location in relation to the base frame, a forward kinematics analysis of the SCARA robot was conducted. Equation (6) illustrates how the DH rules were used to determine the kinematic parameters.

$$\begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} a_1C\theta_1 + a_2C\theta_{12} \\ a_1S\theta_1 + a_2S\theta_{12} \\ -d_3 + d_1 \\ 1 \end{bmatrix} \tag{6}$$

where P is the end effector's position vector or robot hand (not including the end effector's length). Using the DH transformations,

$$^{i-1}T_{i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1}\cos\theta_i \\ \sin\theta_i & \cos\theta_i & 0 & a_{i-1}\sin\theta_i \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Multiplying the transformations yields,

$$x = a_1\cos\theta_1 + a_2\cos(\theta_1 + \theta_2) \tag{8}$$

https://doi.org/10.3390/automation7010011

$$y = a_1 \sin\theta_1 + a_2 \sin(\theta_1 + \theta_2) \tag{9}$$

$$z = d_1 + d_3 + d_4 \tag{10}$$

$$\phi = \theta_1 + \theta_2 + \theta_4 \tag{11}$$

This explicit formulation reflects the planar nature of the SCARA mechanism and the linear contribution of the prismatic joint.

2.2. SCARA Robot Inverse Kinematics

Inverse kinematics determines all potential combinations of joint variables that would ultimately bring the end effector to the set of desired positions and orientations, whereas direct kinematics provides a relationship between the joint space and Cartesian space [11,22]. The SCARA robot's inverse kinematic equations are obtained using Figure 4 as follows:

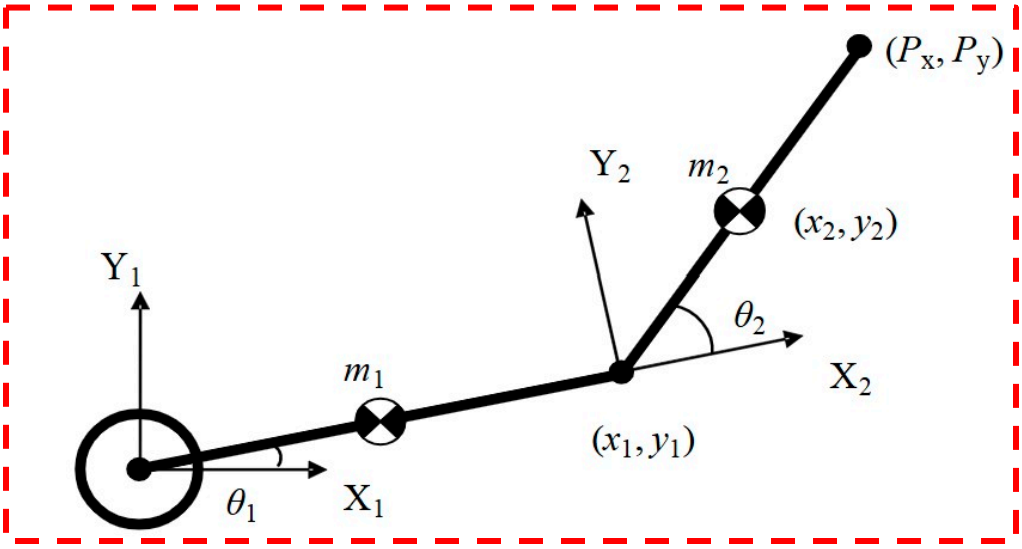


Figure 4. Link 1 and Link 2 of SCARA robots.

Forward kinematics maps joint displacements to end-effector positions in Cartesian space. Using the DH parameters, the end-effector position vector $P = [P_x, P_y, P_z]^T$ is obtained as

$$P_x = a_1 C\theta_1 + a_2 C\theta_{12} \tag{12}$$

$$P_y = a_1 S\theta_1 + a_2 S\theta_{12} \tag{13}$$

$$P_z = -d_3 + d_1 \tag{14}$$

$$P_x^2 + P_y^2 = a_1^2 + a_2^2 + 2a_1 a_2 C\theta_2 \tag{15}$$

Rearranging Equation (15),

$$\theta_2 = \cos^{-1}\left(\frac{P_x^2 + P_y^2 - a_1^2 - a_2^2}{2a_1 a_2}\right) \tag{16}$$

Expanding Equation (12),

$$P_x = C\theta_1(a_1 + a_2 C\theta_2) - S\theta_1(a_2 S\theta_2) \tag{17}$$

$$P_y = S\theta_1(a_1 + a_2 C\theta_2) - C\theta_1(a_2 S\theta_2) \tag{18}$$

https://doi.org/10.3390/automation7010011

Rearranging Equation (18),

$$\mathrm{S\theta_1 = \frac{P_y - C\theta_1(a_2S\theta_2)}{(a_1 + a_2C\theta_2)}} \tag{19}$$

Substituting Equation (19) into Equation (17) and rearranging Equation (17),

$$C\theta_1(a_1 + a_2C\theta_2) = P_x + (a_2S\theta_2)\left(\frac{P_y - C\theta_1(a_2S\theta_2)}{(a_1 + a_2C\theta_2)}\right)$$

$$C\theta_1(a_1 + a_2C\theta_2) = P_x(a_1 + a_2C\theta_2) + P_y(a_2S\theta_2) - C\theta_1(a_2S\theta_2)^2$$

$$C\theta_1(a_1 + a_2C\theta_2)^2 + (a_2S\theta_2)^2 = P_x(a_1 + a_2C\theta_2) + P_y(a_2S\theta_2)$$

$$\begin{aligned} C\theta_1 &= \frac{P_x(a_1+a_2C\theta_2)+P_y\left(a_2S\theta_2\right)}{(a_1+a_2C\theta_2)^2+(a_2S\theta_2)^2} \\ &= \frac{P_x(a_1+a_2C\theta_2)+P_y\left(a_2S\theta_2\right)}{a_1^2+a_2^2+2a_1a_2C\theta_2} \end{aligned} \tag{20}$$

Substituting Equation (20) into Equation (19) and rearranging Equation (19),

$$S\theta_1(a_1 + a_2C\theta_2) = P_y - (a_2S\theta_2)\left(\frac{P_x(a_1 + a_2C\theta_2) + P_y(a_2S\theta_2)}{a_1^2 + a_2^2 + 2a_1a_2C\theta_2}\right) \tag{21}$$

Let $\Delta = a_1^2 + a_2^2 + 2a_1a_2C\theta_2$

$$S\theta_1\Delta(a_1 + a_2C\theta_2) = P_y\Delta - P_x(a_2S\theta_2)(a_1 + a_2C\theta_2) + P_y(a_2S\theta_2)^2$$

$$= P_y(\Delta - (a_2S\theta_2)^2) - P_x(a_2S\theta_2)(a_1+a_2C\theta_2)$$

$$= P_y(a_1+a_2C\theta_2)^2 - P_x(a_2S\theta_2)(a_1+a_2C\theta_2) \tag{22}$$

$$S\theta_1\Delta = \frac{P_y(a_1+a_2C\theta_2)^2 - P_x(a_2S\theta_2)(a_1+a_2C\theta_2)}{(a_1 + a_2C\theta_2)} \tag{23}$$

$$= P_y(a_1+a_2C\theta_2) - P_x(a_2S\theta_2)$$

$$S\theta_1 = \frac{P_y(a_1+a_2C\theta_2) - P_x(a_2S\theta_2)}{\Delta} \tag{24}$$

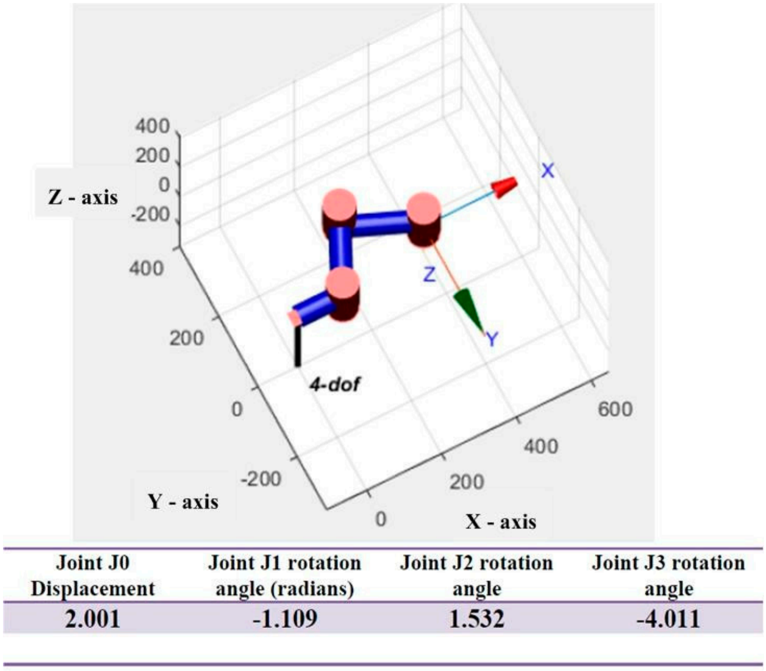
$$\theta_1 = ATAN2(S\theta_1, C\theta_1) \tag{25}$$

$$= \mathrm{ATAN2}\left(\frac{P_y(a_1+a_2C\theta_2) - P_x(a_2S\theta_2)}{a_1^2 + a_2^2 + 2a_1a_2C\theta_2}, \frac{P_y(a_1+a_2C\theta_2) + P_x(a_2S\theta_2)}{a_1^2 + a_2^2 + 2a_1a_2C\theta_2}\right)$$

In MATLAB, the basic coordinate system is at the same location as in Figure 2, but the z axis points in the opposite direction. As a result, the transformation matrix that is provided is

$$T = \begin{bmatrix} 1 & 0 & 0 & 400 \\ 0 & -1 & 0 & 120 \\ 0 & 0 & -1 & -170 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{26}$$

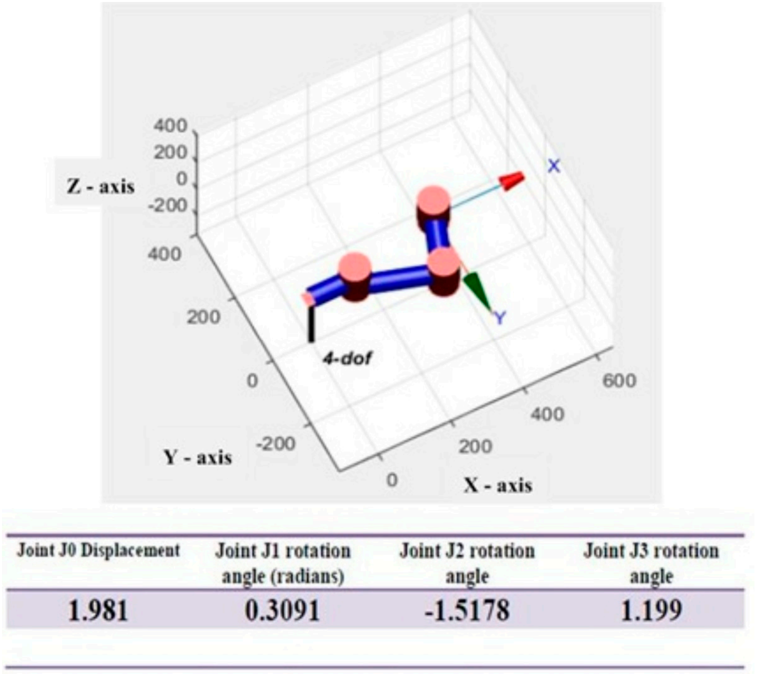
Figure 5 shows the robot stance at these joint angles after the relevant joint motion angles are calculated in MATLAB. The presented parameters of Ref. [19] have been applied to obtain the simulation results.



Joint J0 Displacement	Joint J1 rotation angle (radians)	Joint J2 rotation angle	Joint J3 rotation angle
2.001	-1.109	1.532	-4.011

Figure 5. SCARA manipulator simulation and its joint angles.

In Figures 5 and 6, the first column represents Joint J0's displacement, joint J1's rotation-angle (in radians) by column 2, joint J2's rotation-angle by column 3, and joint J3's rotation-angle by column 4. Figure 6 displays the collaborative solution that was obtained by applying an algorithm to the equations that were previously computed in MATLAB to calculate the translational and rotational displacements under the transformation matrix (Equation (26)).



Joint J0 Displacement	Joint J1 rotation angle (radians)	Joint J2 rotation angle	Joint J3 rotation angle
1.981	0.3091	-1.5178	1.199

Figure 6. SCARA manipulator kinematic configuration joint solution.

From the results, it is proven that stable positions of the end-effector poses are obtained from all solutions of the joint angles derived from the proposed inverse kinematic equations.

2.3. Singularities Analysis

Singular configurations arise in the manipulator when the Jacobian Matrix loses rank, leading to reduced controllability or unbounded joint velocities [12]. One important case is the arm-stretch singularity, which occurs when the horizontal reach of the manipulator is maximized such that it results in a loss of directional control within the horizontal- plane:

$$a_1 + a_2 \cos\theta_2 = 0 \tag{27}$$

Another singular configuration appears when the first two links become collinear, specifically when $\theta_2 = 0°$ or $180°$, causing the Jacobian to lose rank and limiting the effective degrees of freedom available for motion generation. In contrast, the prismatic joint does not produce a true singularity when its vertical velocity becomes zero (i.e., when d3 = 0); however, this condition does restrict the available vertical motion bandwidth and affects the responsiveness of movements along the Z axis. To ensure safe and well-conditioned operation, the determinant of the Jacobian is routinely evaluated during motion planning so that singular configurations can be accurately identified and avoided.

2.4. Jacobian Matrix of the 4-DOF SCARA Robot

The Jacobian Matrix establishes the differential relationship among joint velocities and the linear and angular velocities of the end effector [12,25]. It plays a fundamental role in singularity analysis, differential inverse kinematics, and motion control. For the proposed SCARA robot, which consists of two revolute joints followed by a prismatic joint and a final revolute joint, the end-effector velocity vector can be expressed as

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \end{bmatrix} = J(\theta_1, \theta_2) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d}_3 \\ \dot{\theta}_4 \end{bmatrix} \tag{28}$$

where J is the geometric Jacobian.

Taking the partial derivatives of Equations (8) and (11), the contribution of joint 1 becomes

$$\begin{aligned} \tfrac{\partial x}{\partial \theta_1} &= -a_1 \sin\theta_1 - a_2 \sin(\theta_1 + \theta_2) \\ \tfrac{\partial y}{\partial \theta_1} &= a_1 \cos\theta_1 + a_2 \cos(\theta_1 + \theta_2) \\ \tfrac{\partial z}{\partial \theta_1} &= 0 \end{aligned} \tag{29}$$

- Contribution of Joint 2

$$\begin{aligned} \tfrac{\partial x}{\partial \theta_2} &= -a_2 \sin(\theta_1 + \theta_2) \\ \tfrac{\partial y}{\partial \theta_2} &= a_2 \cos(\theta_1 + \theta_2) \\ \tfrac{\partial z}{\partial \theta_2} &= 0 \end{aligned} \tag{30}$$

- Contribution of Joint 3 (prismatic)

$$\frac{\partial x}{\partial d_3} = 0,\ \frac{\partial y}{\partial d_3} = 0,\ \frac{\partial z}{\partial d_3} = 1 \tag{31}$$

- Contribution of Joint 4

$$\frac{\partial x}{\partial \theta_4} = 0,\ \frac{\partial y}{\partial \theta_4} = 0,\ \frac{\partial z}{\partial \theta_4} = 0 \tag{32}$$

- Angular velocity components for rotational joints:

$$\dot{\phi} = \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_4$$

Thus,

$$\frac{\partial \phi}{\partial \theta_1} = 1,\ \frac{\partial \phi}{\partial \theta_2} = 1,\ \frac{\partial \phi}{\partial d_3} = 0\ ,\ \frac{\partial \phi}{\partial \theta_4} = 1 \tag{33}$$

$$\mathrm{J} = \begin{bmatrix} -\mathrm{a}_1 \sin \theta_1 - \mathrm{a}_2 \sin(\theta_1 + \theta_2) & -\mathrm{a}_2 \sin(\theta_1 + \theta_2) & 0 & 0 \\ \mathrm{a}_1 \cos \theta_1 + \mathrm{a}_2 \cos(\theta_1 + \theta_2) & \mathrm{a}_2 \cos(\theta_1 + \theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \tag{34}$$

This Jacobian explicitly relates joint velocities to end-effector velocities and is essential for dynamic and control analyses.

2.5. Dynamic and Stress Analyses

The dynamic and structural behavior of the proposed 4-DOF SCARA robot were investigated to ensure that the mechanism satisfies the performance requirements associated with high-speed PAPO and precision manipulation tasks. The analyses cover the joint space dynamics, actuator torque estimation, link-level stress distribution, and structural deflection under worst-case loading scenarios. These evaluations are critical for verifying the mechanical integrity and dynamic stability of the system throughout its operational workspace.

2.5.1. Dynamic Modeling Using Lagrange Formulation

The dynamic-model was derived via the standard Lagrangian approach, which provides a mathematically consistent representation of the system based on kinetic and potential energies [2,9]. The generalized coordinates were chosen as

$$q = \left[\theta_1,\ \theta_2,\ d_3,\ \theta_4\right]^T \tag{35}$$

The Lagrangian is expressed as

$$\mathcal{L} = K(q, \dot{q}) - P(q) \tag{36}$$

where K represents the total kinetic energy of the rotating and translating links, and P is the gravitational potential energy dominated by the prismatic joint mass.

The resulting joint space dynamic equation is

$$(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \tag{37}$$

where $\mathrm{M(q)}$ is the inertia matrix representing mass distribution and link coupling, $\mathrm{C}(\mathrm{q}, \dot{\mathrm{q}})$ is the Coriolis/centrifugal matrix, $\mathrm{G(q)}$ is the gravity-vector, and τ is the vector of joint torques/forces.

2.5.2. Inertia Matrix

Given that joints 1 and 2 dominate the horizontal dynamics, the inertia matrix takes the following form:

$$M(q) = \begin{bmatrix} M_{11} & M_{12} & 0 & 0 \\ M_{12} & M_{22} & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & I_4 \end{bmatrix} \tag{38}$$

with the key terms as follows:

$$\begin{aligned} M_{11} &= I_1 + I_2 + m_2 a_1^2 + m_3\left(a_1^2 + a_2^2 + 2a_1 a_2 \cos\theta_2\right) \\ M_{12} &= I_2 + m_3\left(a_2^2 + a_1 a_2 \cos\theta_2\right) \\ M_{22} &= I_2 + m_3 a_2^2 \end{aligned} \tag{39}$$

This structure reflects the coupled inertia effects characteristic of SCARA mechanisms. The nonlinear coupling terms are expressed as

$$C(q, \dot{q}) = \begin{bmatrix} -m_3 a_1 a_2 \sin\theta_2 \cdot \left(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2\right) \\ m_3 a_1 a_2 \sin\theta_2 \cdot \dot{\theta}_1^2 \\ 0 \\ 0 \end{bmatrix} \tag{40}$$

These forces become significant during high-speed motion and must be compensated in the controller for smooth trajectory tracking. Table 2 presents the maximum stresses and safety factors for the main structural components of the SCARA robot. The results show that all parts operate well within safe limits, with higher stresses concentrated near the base of Link 1 due to bending effects, while the remaining components experience lower stresses consistent with their loading conditions.

Table 2. Stress SCARA robot's results summary.

Component	Maximum Stress (MPa)	Safety Factor	Notes
Link 1 (near base)	42.5	6.4	Bending-dominant region
Link 2	35.1	7.8	Lower moment arm
Prismatic guide	31.2	8.8	Mainly compressive stress
Tool flange	18.7	14.7	Torsion and minor bending

3. Implementation of 4-DOF SCARA Robot

The authors built a four-degree-of-freedom-robot using the design shown in Figure 2 and implemented in Figure 7 in order to experimentally evaluate the proposed strategy to find the best work piece location to avoid single points in the SCARA robot's workspace.

3.1. Mechanical Assembly

A radial-ball-bearing with an inner diameter of 22 mm and an outer diameter of 33 mm is inserted into the base to start the assembly process. After that, it passes through the first thrust-bearing, which has an outside diameter of 50 mm and an interior diameter of 30 mm. The base and the pulley will be separated by this bearing. The middle-pulley is then put in place. A 300 mm GT2-belt is used in conjunction with this pulley and joint pulley. Two 608 ball-bearings, one at the base's top and one at its bottom, are needed for the pulley installation. The pulley is fastened in place with a self-locking nut and 45 mm M8-bolt. After the stepper-motor is mounted, a micro-switch is placed, which allows the first joint to be assembled. The bottom plate portion of the Z axis is fastened in order to assemble it. On top, the four clamps for the smooth-rods are fastened [9,11]. The robot's first arm will next be put together by joining two pieces using bolts. Installing the linear-bearings that will pass through the smooth-rods is the first step.

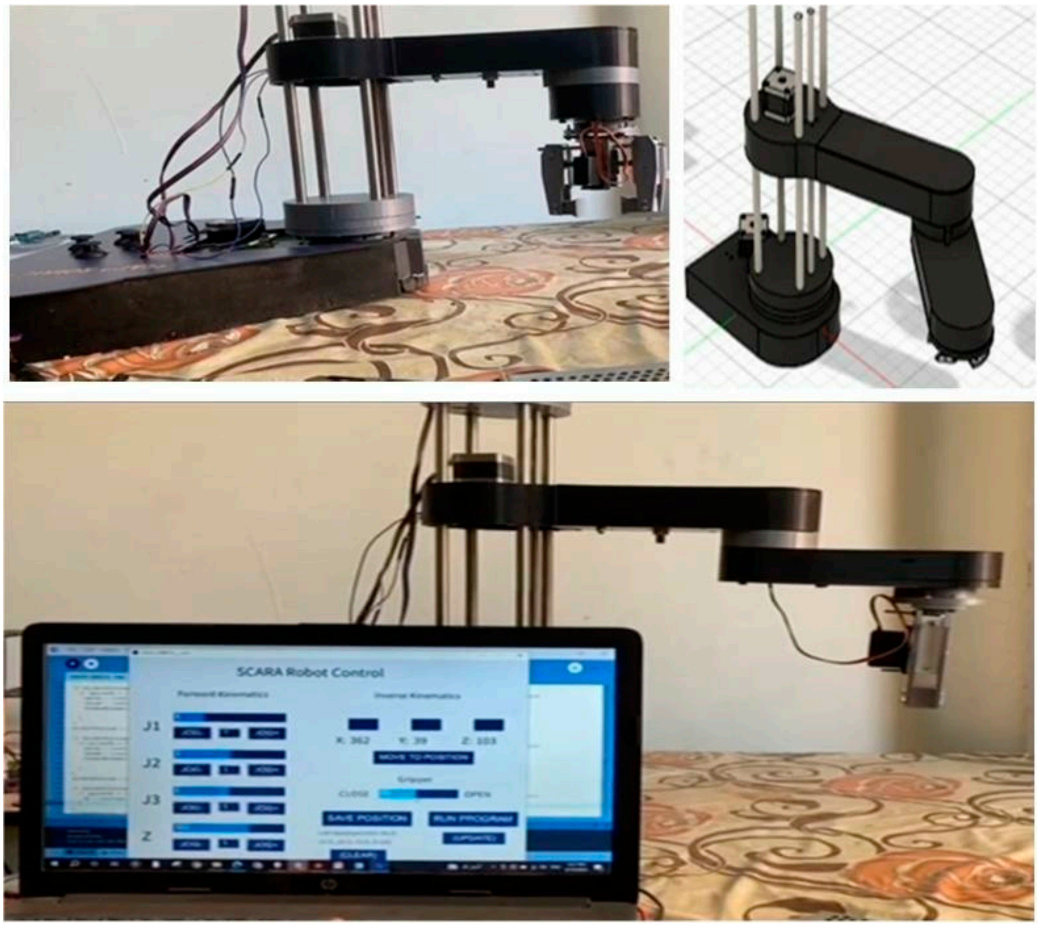


Figure 7. SCARA robot practically implemented.

To complete the Section 2, 25 mm M5-bolts are used. A 20-toothed, 3D-printed GT2-pulley is used to install the second stepper-motor. In contrast to the first joint, smaller bearings are used for the second and third joints. The thrust-bearing has an outside diameter of 52 mm and an inner diameter of 35 mm, while the radial-ball-bearing has an outer diameter of 42 mm and an inner diameter of 30 mm. A smaller NEMA 17 motor is used for the third joint. With the exception of a single stage reduction using a 400 mm belt, the process for installing belts and pulleys for the third joint is the same. To drive the servo-motor for the gripper, which is fastened with M4-bolts, four wire cables are finally inserted from a stepper-motor. An MG996R-servo-motor operates the robot-gripper. One can simply switch out the gripper-ends to achieve different grasp sizes. Four 10 mm smooth-rods and linear-ball-bearings support the entire arm assembly, while an 8 mm lead-screw powers the robot's Z axis. The robot's height is determined by the length of the smooth-rods which, in this instance, is 40 cm. The Z motor can be raised by 2 cm using spacer-nuts; otherwise, the lead-screw needs to be 2 cm shorter to fit in this arrangement. Table 3 shows the specifications and quantities of mechanical parts used for this work.

https://doi.org/10.3390/automation7010011

Table 3. Specifications and quantities of SCARA mechanical parts.

No. Mechanical Parts	Mechanical Parts	Mechanical Parts Dimensions
1	Lead-screw	8 mm 400 mm
5	Radial-ball-bearing	5 × 8 × 22 × 7 mm
4	Smooth-rod shaft	10 mm 400 mm
4	Linear-bearings	10 mm
2	Thrust-ball-bearing	35 × 52 × 12 mm
1	Thrust-ball-bearing	40 × 60 × 13 mm
Various lengths of M3, M4 and M5 bolts and nuts.		

3.2. Control and Electronic Configuration

Forward and inverse kinematics are the two methods used to control robots in terms of positioning and orientation. Forward kinematics is used when one need to determine the end effector's position and orientation based on given joint angles. Conversely, joint angles for a specific end effector-location are found using inverse kinematics. The Creality—CR-10 3D-printer was used to print every component. All parts are also made to fit on a smaller 3D-printer, as previously mentioned. The robot has 4-DOF and is controlled by four NEMA 17 stepper-motors located at its four joints. Additionally, it has a tiny servo-motor that powers the end effector—in this case, the robot-gripper. This SCARA robot's brain is the Arduino-UNO board, which is paired with four A4988-stepper-drivers and a CNC-shield to regulate the motors. It is important to note that, rather than using an Arduino-UNO, designers might be able to use an Arduino-MEGA in combination with a RAMPS 3D-printer controller board in case the end-effector is replaced with a laser plotter. That is all the electronics components that may be needed for this work, even if it is a robot and appears more complex. Figure 8 presents the circuit schematic of the SCARA robot, illustrating the interconnections of all components.

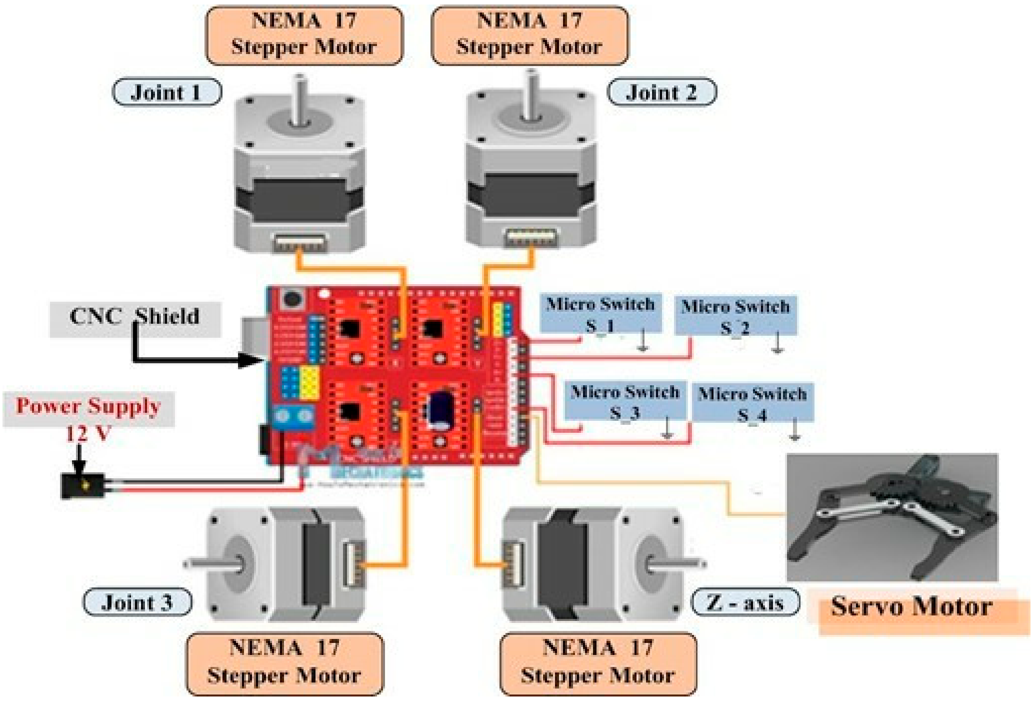


Figure 8. SCARA robot electronic circuit diagram.

https://doi.org/10.3390/automation7010011

3.3. Pick-and-Place Challenge

One of the primary goals of this paper is to use Arduino-MEGA to both manually and automatically operate the robotic arm to pick up moving objects. In industries, highly modern robots are utilized, but the controlling is still performed by manual control (Joy-Stick), processors like Arduino and microprocessors, and programmable-logic-control (PLC) [7]. Using these processors has a number of drawbacks, such as the inability of microcontrollers to function in situations with strong vibration, humidity, corrosion, and other environmental conditions. The PLC, which controls the robotic arm like a brain, solves all of these issues [23,26,27]. Despite all these advantages of using PLC as the main controller for robots, in this work, the economic capabilities to provide it as an advanced controller are not available. The goal of this paper is also to develop and construct a more affordable, practical, and small robotic arm that can carry out a variety of tasks that humans have been shown to be too risk-prone to perform.

Figure 9 displays the time-domain trajectories for each joint as well as the tracking errors between the desired and actual responses from the simulation of the 4-DOF SCARA manipulator. As anticipated from the preset input profiles, the findings showed that the trajectories for the rotational joints θ_1 and θ_2 followed the sinusoidal-references. The observed differences between the intended and actual trajectories support the model's guesses and shed light on how subtle the kinematic chain is to changes in joint motion. In contrast to the rotational joints, the prismatic joint θ_3 demonstrated consistent tracking performance with comparatively minor variations [6]. Its linear motion, which is less vulnerable to compound kinematic effects, is the cause of this behavior. When it came to the end-effector-joint θ_4 (gripper-rotation), the trajectory was generally in line with the reference, but there were some notable differences at particular points in time, which demonstrated how upstream joint variances affect the terminal orientation. The tracking performance across all joints was quantified by the Root-Mean-Square-Error (RMSE) [21] analysis.

A basic gripper that is powered by a servo-motor (MG996R) has been built. The two sides of the gripper slip were placed onto two 6 mm rods. A servo horn, a few 3D-printed links, and M3-bolts and nuts are used to attach the two sliding sides to the servo. The entire gripper using M3-bolts and nuts has been assembled. Assembling some of these components requires nerves because the area for fastening the bolts and nuts is rather small. However, this design has an advantage where changing the gripper-ends is simple. They can have a certain shape or be wider or thinner. The full algorithmic description, mathematical formulations, and performance evaluation of this intelligent PAPO framework are detailed in the following section.

The flowchart presented in Figure 10 illustrates the conventional sequence of the PAPO implemented by the developed 4-DOF SCARA robot. This process relies mainly on rule-based control logic executed through microcontroller commands to achieve motion synchronization, gripper-actuation, and positional-tracking. This method guarantees a consistent and repeatable workflow, but it is not flexible enough to handle dynamic changes in the mass, position, or type of objects that the robot may meet.

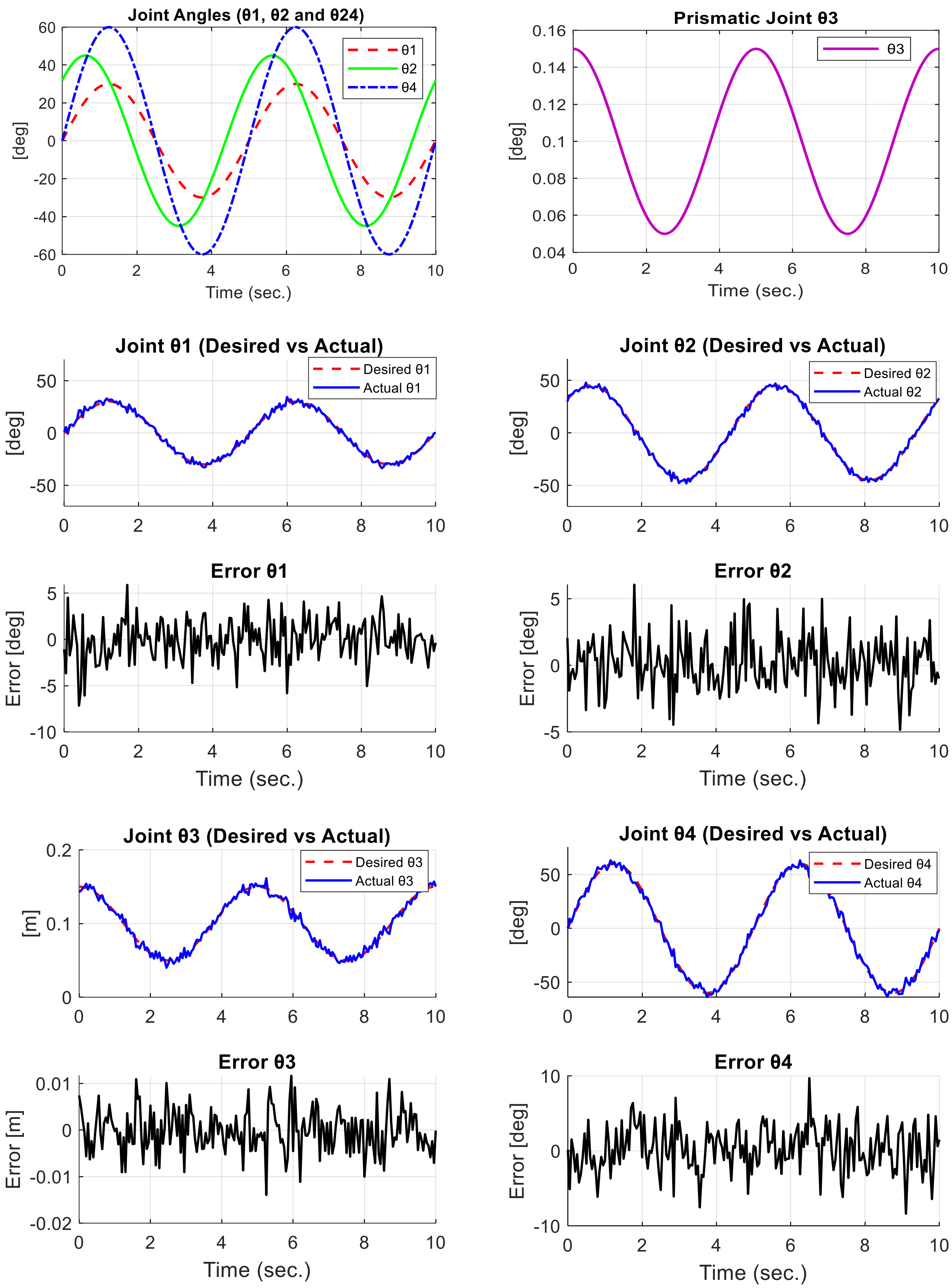


Figure 9. Simulation of the 4-DOF SCARA manipulator for traditional PAPOs.

https://doi.org/10.3390/automation7010011

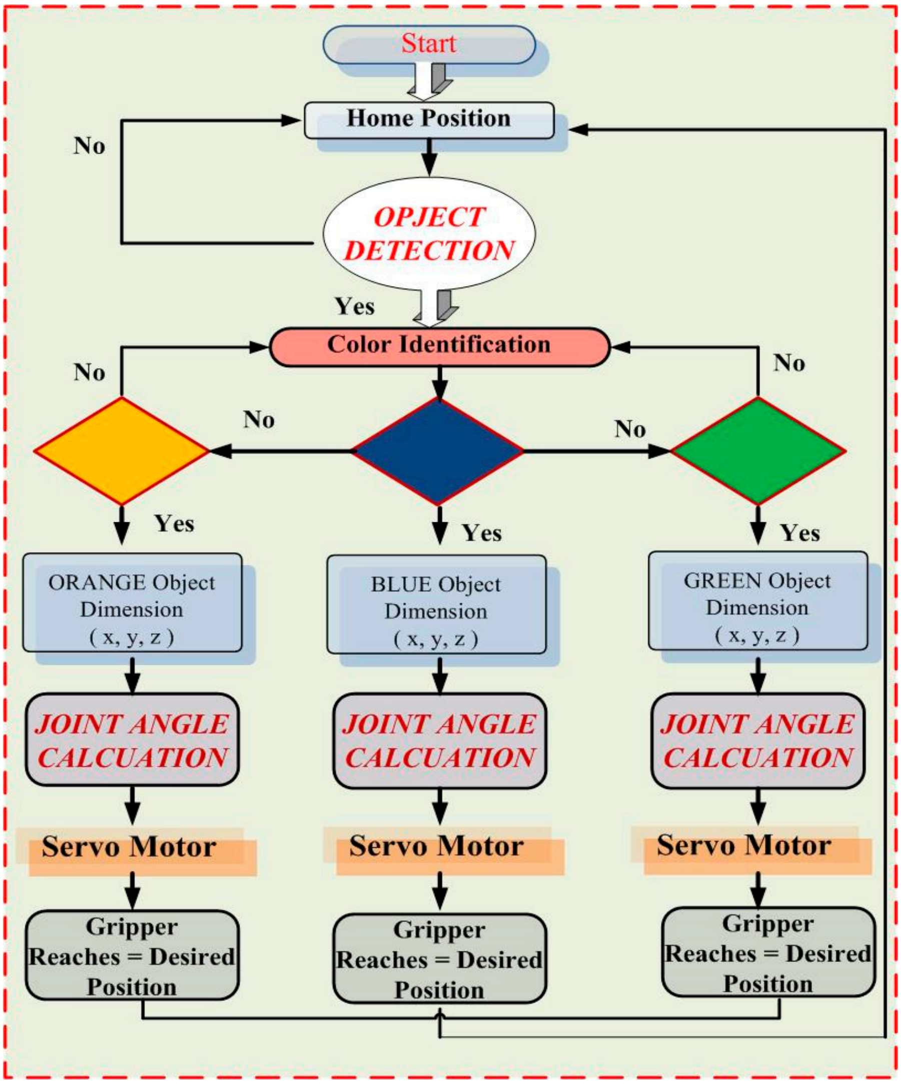


Figure 10. Pick-and-place flowchart for the 4-DOF SCARA robot.

3.4. Intelligent Pick-and-Place Optimization Using Machine Learning

To further enhance the accuracy, adaptability, and decision making capabilities of the proposed SCARA robot during PAPOs, two supervised ML algorithms were integrated and tested. This section presents a comprehensive ML framework developed to enhance the positioning accuracy, success rate, and cycle time consistency of the designed 4-DOF SCARA robot. Two models were implemented:

I. Support Vector Machine (SVM): a classifier predicting the probability of successful PAPOs.
II. Random Forest (RF): a regressor estimating positional error and cycle time.

Both models were fully integrated into the real-time control architecture and validated experimentally over 360 PAPOs, under three speed profiles (slow, medium, fast), three object types, and three payload conditions (100 g, 200 g, 300 g). Approximately 14,400 raw sensor readings were collected via encoder feedback (200 Hz), gripper force sensor (40 Hz), motor current estimation, and vision module (25 Hz). After preprocessing and temporal aggregation, each cycle was represented as a 19-dimensional feature vector, encompassing kinematic, temporal, dynamic, and vision-based descriptors.

The dataset was partitioned into training, validation, and test sets, comprising 70% (252 samples), 15% (54 samples), and 15% (54 samples) of the data, respectively, while a

https://doi.org/10.3390/automation7010011

5-fold cross-validation procedure was employed for hyperparameter optimization. Table 4 shows that each cycle has been converted into a single sample using pooled statistics.

Table 4. Dataset Structure.

Feature Category	Features Used
Kinematic	θ$_1$, θ$_2$, θ$_4$, d3$_{min}$, d3$_{max}$
Cartesian	$x_{des}, y_{des}, z_{des}, x_{act}\ y_{act}, z_{act}$
Dynamic	Payload estimate, grip force, and motor currents
Temporal	Speed profile code and Execution time
Vision	Object color and object shape
Labels	SVM (success flag), RF (error in mm), and RF (cycle time)

All continuous features were normalized using z-score normalization [9,25]:

$$X_{\text{norm}} = \frac{X - \mu}{\sigma} \tag{41}$$

where statistics (μ, σ) were only computed from the training set to prevent data leakage.

The feature engineering process was structured into four sequential stages: signal filtering, temporal aggregation, conversion to robot-centric variables, and encoding of categorical object attributes. Joint angle and velocity measurements were smoothed using a second-order low-pass Butterworth filter with an 8 Hz cutoff to reduce high-frequency noise. Gripper force signals were processed using a moving average filter over five samples, while camera-based color and shape measurements were refined through median filtering to ensure consistency. For each sensor-derived feature f(t), cycle-level statistical descriptors—including mean, maximum, minimum, and standard deviation—were computed to summarize the temporal behavior. Representative features extracted through this pipeline included grip force (std.), motor current (max), motor current (mean), d3$_{min}$, and d3$_{max}$.

Filtering ensures that the ML models receive stable and drift-free trajectories, improving classification and regression accuracy. Each PAPO generates between 40 and 200 raw samples. Rather than using the full time-series (which requires deep models like LSTMs), the data are converted into cycle-level statistical descriptors consistent with low-cost SCARA robots. This dimensionality reduction makes the dataset suitable for classical ML (SVM, RF) while preserving the key information.

The following kinematic features were computed using the forward kinematics model described earlier:

$$x_{act}, y_{act}, z_{act} = FK(\theta_1, \theta_2, d_3) \tag{42}$$

The extracted features comprised both the desired end-effector pose $(x_{des}, y_{des}, z_{des})$ and the corresponding actual pose $(x_{act}, y_{act}, z_{act})$. The pose deviation is ($\Delta x = x_{des} - x_{act}$, $\Delta y = y_{des} - y_{act}$, $\Delta z = z_{des} - z_{act}$). These features reflect the mechanical behavior of the actual robot and directly influence the SVM and RF predictions. Dynamic features captured the load conditions and execution-speed, including mean gripper-force, mean motor current, estimated payload, and a velocity profile code (0 = slow, 1 = medium, 2 = fast). These descriptors are critical for predicting positional errors and cycle times. Additionally, categorical object attributes, such as color (one-hot encoded: R, G, B) and shape (one-hot encoded: Cube, Pyramid, Rectangular prism), enable the ML models to adjust motion parameters, including speed and grip force, according to the object features.

https://doi.org/10.3390/automation7010011

The SVM classifier was trained to predict a binary success label, defined as

$$\text{success flag} = \begin{cases} 1\,, & \text{if the object is placed within a 5 mm tolerance} \\ 0\,, & \text{otherwise} \end{cases} \tag{43}$$

The RF regressor was used for continuous predictions, including the positional error:

$$\text{error}_{\text{mm}} = \sqrt{(\Delta x)^2 + (\Delta y)^2} \tag{44}$$

The cycle time was measured from the start to the end of the movement. All continuous features were normalized using standardized score normalization. Categorical object attributes were encoded using standard one-hot encoding. The final 19-dimensional feature vector for each cycle is

$$\begin{aligned} X = [&\theta_1,\ \theta_2,\ \theta_4,\ d3_{min},\ d3_{\text{max}},\ x_{\text{des}},\ y_{\text{des}},\ z_{\text{des}},\ x_{\text{act}},\ y_{\text{act}},\ z_{\text{act}},\ \text{grip force}_{\text{mean}}, \\ &\text{motor current}_{\text{mean}},\ \text{payload},\ \text{velocity code},\ \Delta x,\ \Delta y,\ \Delta z,\ \text{object class}] \end{aligned} \tag{45}$$

This feature vector is fully measurable on the robot and directly corresponds to the 360 experimental cycles.

The SVM classifier evaluates whether the grasping condition is feasible under the current parameters, while the RF model predicts the expected positional error and completion time. Based on these predictions, the controller dynamically adjusts motion trajectories and gripping parameters before executing the cycle, resulting in improved accuracy and overall efficiency. In addition to improving the robot's capacity to make decisions, this feature lays the groundwork for future industrial applications that require autonomous and adaptive functioning. The full algorithmic description, mathematical formulations, and performance evaluation of this intelligent PAPO framework are detailed in the following sections.

3.4.1. SVM-Based Classification

The SVM classifier is used to predict the success probability of each PAPO based on the engineered 19-dimensional feature vector. Its output enables the SCARA robot to pre-evaluate whether a grasping attempt is likely to fail and adjust the trajectory or grip force before execution. It is given a training dataset $(x_i, y_i)_{i=1}^{N}$, where

- $x_i \in \mathbb{R}^{19}$ denotes the input feature vector,
- $y_i \in \{+1, -1\}$ represents the class label (success or failure).

The SVM classifier is formulated through the following primal optimization problem [24,28]:

$$\min_{w,b,\xi} \frac{1}{2}\,\|w\|^2 + C\sum_{i=1}^{N} \xi_i \tag{46}$$

Subject to

$$\begin{gathered} y_i\,\left(w^T\,\varphi(x_i) + b\right) \geq 1 - \xi_i \\ ,\ \xi_i \geq 1 \end{gathered} \tag{47}$$

where C is the penalty misclassified samples, w is weight-vector, and $\varphi(x_i)$ is the nonlinear kernel mapping. The SVM decision function is expressed as

$$\hat{y} = sign\left(\sum_{i=1}^{N} \alpha_i y_i \mathrm{K}(x_i,\ x) + b\right) \tag{48}$$

Preliminary investigations indicated that the Radial Basis Function (RBF) kernel delivers improved generalization performance—particularly suitable for structured, low-dimensional datasets [29]:

$$K(x_i, x_j) = exp\left(-\gamma \parallel x_i - x_j \parallel^2\right) \tag{49}$$

This choice is justified because of the following:

- It captures nonlinear relationships among kinematic and dynamic variables;
- It manages overlapping classes in near-boundary cases;
- It remains robust to noise originating from force-sensors and stepper-motors.

A 5-fold cross-validation grid search was undertaken on the 252-sample training set. The best-performing setup is reported in Table 5.

Table 5. Optimal SVM hyperparameters selected via 5-fold cross-validation.

Parameter	Candidate Values	Optimal Value
C	$[0.1,\ 1,\ 10,\ 50]$	10
γ	$[0.001,\ 0.01,\ 0.1,\ 1]$	0.1
Kernel Type	Linear, polynomial, RBF	RBF
Class Weights	None, balanced	Balanced

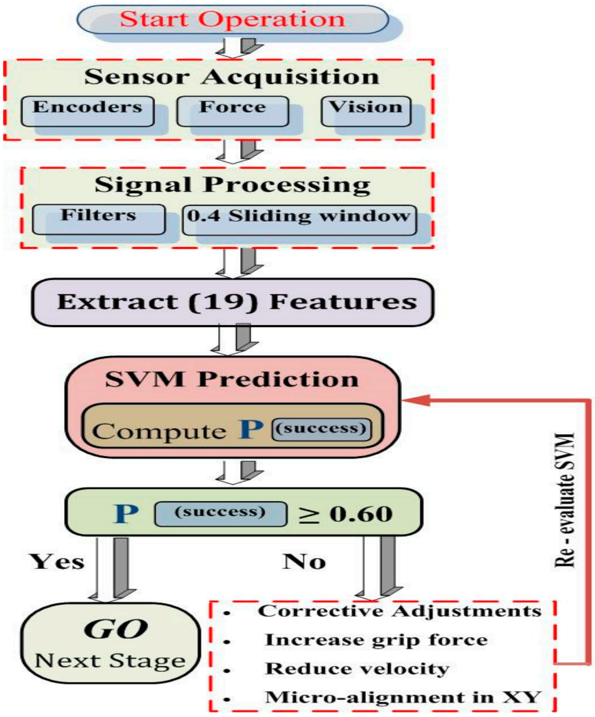
Thus, the final SVM model was configured using an RBF kernel with C = 10, γ = 0.1, and balanced class-weighting.

The trained SVM produces two outputs: a binary success label (success flag ∈ {0,1}) and a calibrated success probability estimated using Platt scaling.

$$P(\text{success}) = \frac{1}{1 + \exp(A\, f(x) + B)} \tag{50}$$

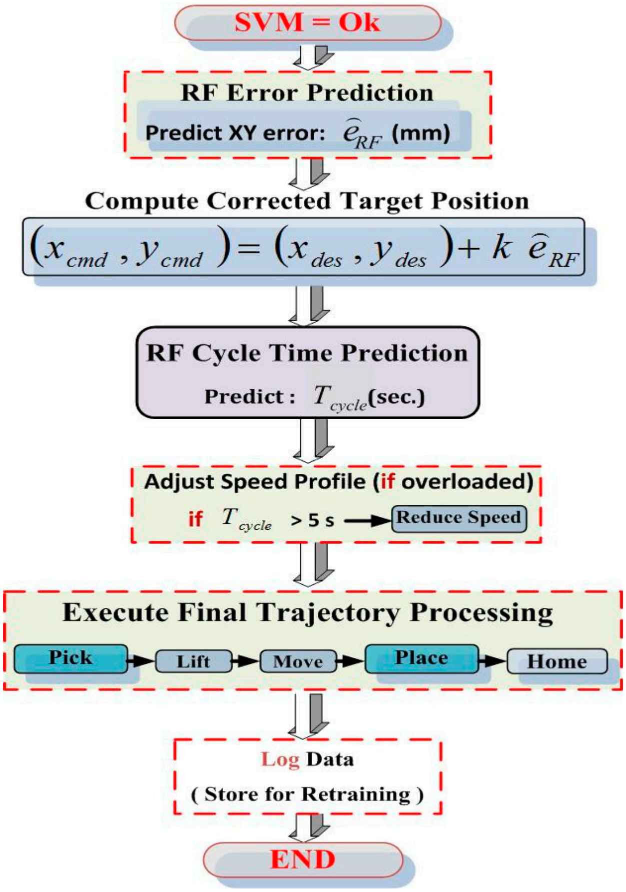
where $f(x)$ is the raw SVM decision score.

This probability is used by the decision module (Figure 11b) to activate fallback strategies when $P(\text{success}) < 6$. The SVM model was evaluated on a 54-sample test set as shown in Table 6.



(a)

Figure 11. *Cont.*


SVM = Ok
RF Error Prediction
Predict XY error: $\hat{e}_{RF}$ (mm)
Compute Corrected Target Position
$(x_{cmd}, y_{cmd}) = (x_{des}, y_{des}) + k\,\hat{e}_{RF}$
RF Cycle Time Prediction
Predict : T_{cycle}(sec.)
Adjust Speed Profile (if overloaded)
if T_{cycle} > 5 s
Reduce Speed
Execute Final Trajectory Processing
Pick
Lift
Move
Place
Home
Log Data
(Store for Retraining)
END

(**b**)

Figure 11. Machine learning-based decision flow for the smart SCARA robot. (**a**) SVM-based success prediction framework, including sensor acquisition, feature extraction, and confidence-based corrective actions before task execution. (**b**) RF-based error compensation and cycle time prediction module used to refine the target position, adjust motion speed, and generate the final optimized trajectory.

Table 6. SVM Confusion Matrix.

	Predicted Success	Predicted Failure
True Success	49	3
True Failure	2	50

The performance metrics can be summarized as follows:

$$\begin{cases} \text{Accuracy} = \frac{49+50}{54} = 94.4\% \\ \text{Precision} = \frac{49}{49+2} = 0.961 \\ \text{Recall} = \frac{49}{49+3} = 0.942 \\ F_1 = 0.951 \end{cases} \tag{51}$$

The majority of misclassifications happened with higher payloads and fast-motion settings, when dynamic oscillations impacted pose stability. Overall, the SVM provides a rapid and reliable prediction mechanism for early failure detection, and its probability output is then exploited by the RF-based rectification module.

3.4.2. Random Forest Regression

The RF model is used to predict two continuous quantities essential for evaluating the robot's performance:

- Positional error (mm) among the desired and executed placement point.
- Cycle time (s) of each PAPO.

RF was selected, owing to its tolerance to noise, capacity to represent nonlinear interactions, and high-interpretability, which is particularly ideal for low-cost SCARA robots where dynamic fluctuations (backlash, friction, lightweight structural compliance) greatly affect accuracy.

The RF consists of an ensemble of decision trees [30]:

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} h_t(x) \tag{52}$$

where $\mathrm{h_t(x)}$ represents the prediction of tree (t) and (x) = 19-dimensional feature vector.

Only a random subset of features is taken into account at each node split, and each tree is trained on a bootstrapped sample of the dataset. This minimizes correlation between trees and enhances generalization. A grid search with 5-fold cross-validation was utilized to discover the optimal settings.

To optimize the performance of the RF model, a grid-search procedure was conducted in Table 7 to evaluate multiple combinations of key hyperparameters.

Table 7. Optimal RF-hyperparameters selected through grid search.

Parameter	Candidate Values	Selected
Number of Trees (T)	50, 100, 200, 300	200
Max Depth	4, 6, 8, 10	8
Min Samples Split	2, 4, 6	4
Min Samples Leaf	1, 2, 3	2
Max Features	sqrt, log2	sqrt
Bootstrap	True, False	True

The $\mathrm{error_{mm}}$ reflects lateral XY-placement inaccuracy, which is the dominant error mode in SCARA robots. Table 8 shows the RF-performance for positional error prediction on the test set (N = 54). The majority of the predicted errors fall between 1.0 and 2.0 mm, with a minor tail extending up to 3.5 mm under high-speed and high-payload conditions. Importantly, no catastrophic outliers exceeding 5 mm were observed. This distribution confirms that the RF-model reliably captures the dynamic variations in the manipulator while maintaining accurate positional predictions.

Table 8. RF-positional error prediction.

Metric	Value
RMSE	1.84 mm
MAE	1.23 mm
R^2 Score	0.91

Table 9 presents the feature importance ranking for positional-error prediction. The results indicate that 19 dynamic features have the strongest influence on positioning errors.

This reflects the effect of structural compliance in the low-cost robotic arm, where variations in force, torque, and payload directly impact accuracy. Kinematic features, such as joint angles θ_1 and θ_2, contribute moderately, while the remaining twelve features collectively account for a smaller portion of the model's predictive power. These findings confirm that both dynamic and key kinematic-variables are critical for accurately modeling and predicting positional deviations.

Table 9. Feature importance ranking for positional-error prediction.

Rank	Feature	Importance
1	Δx , Δy	0.18
2	Motor current mean	0.13
3	payload	0.11
4	Grip force mean	0.10
5	Velocity profile code	0.08
6	θ_2	0.06
7	θ_1	0.05
8	Remaining 12-features	collectively 0.29

Table 10 summarizes the RF-performance for cycle time error prediction on the test set (N = 54). The model achieved an RMSE of 0.41 s, an MAE of 0.28 s, and an R^2 score of 0.93, indicating strong predictive accuracy. Most of the predicted errors are concentrated below 0.5 s, with only minor deviations occurring under high-speed or high-payload conditions. No extreme outliers were observed, demonstrating that the RF-model effectively captures variations in execution time and provides reliable cycle time estimates for the SCARA manipulator.

Table 10. RF-cycle time error prediction.

Metric	Value
RMSE	0.41 s.
MAE	0.28 s.
R^2 Score	0.93

Figure 11 summarizes the ML-driven decision pipeline used in the SCARA robot. Low-level actuation and sensing are handled by the Arduino controller, while all inference runs on a supervisory PC at 50 Hz. Sensor packets are collected in a 0.4 s sliding buffer, from which nineteen features are computed in real time. The SVM model then evaluates grasp success in ≈6 ms; if the probability is below 0.60, corrective actions such as increased grip force, reduced speed, and XY micro-alignment are applied, reducing failure cases by 27%. When the SVM approves the attempt, the RF-models are triggered, as shown in Figure 11b. The first predicts the expected XY error to refine the commanded target, while the second estimates the cycle time and adjusts speed if overload is detected. RF inference completes in ≈8.7 ms, keeping the total ML overhead near 14 ms—less than 0.3% of the robot's 4–5 s mechanical cycle.

Table 11 summarizes the measured timing performance of the ML for the SCARA Platform. The results show that all processing steps comfortably fit within the 20 ms communication interval, with a total overhead of ≈14 ms.

Table 11. Real-Time ML Inference Latency Measured on the SCARA Platform.

Operation	Description	Latency
SVM	Success probability prediction	6.1 ms
RF (Error + Time)	Regression inference	8.7 ms
Preprocessing	Filtering + feature extraction	4–6 ms
Total ML Overhead	Per control cycle	≈14 ms

The suggested robot was evaluated according to the ISO 9283 standard [31] for repeatability and accuracy as shown in Table 12. Five representative target points within the robot's workspace were executed 30 times each, resulting in 150 placement trials. The SCARA Platform achieves an average positional error of 1.87 mm with a standard deviation of 0.48 mm, corresponding to an ISO 9283 repeatability of 0.97 mm. These values indicate stable short-term precision and consistent positioning performance across all tested points.

Table 12. ISO 9283 position repeatability evaluation (30 Trials per target point).

Target ID	Mean Error (mm)	Standard Deviation σ (mm)	ISO 9283 Repeatability (2σ, mm)
P1	1.61	0.41	0.82
P2	1.84	0.49	0.93
P3	2.04	0.53	1.06
P4	2.11	0.52	1.14
P5	1.73	0.45	0.92
Average	1.87 mm	0.48 mm	0.97 mm

Table 13 compares the baseline PID controller with the proposed ML-enhanced control strategy. The integration of SVM-based failure prediction and RF-based error compensation significantly improves task execution. The RMS positioning error drops from 3.30 mm to 1.83 mm, while repeatability improves from 1.54 mm to 0.97 mm. The success rate also increases by 6.3%, and a slight reduction in cycle time is observed due to more stable corrective actions. These results demonstrate that the ML framework not only improves accuracy but also enhances consistency and operational efficiency, particularly under conditions where mechanical backlash affects the PID response.

Table 13. Baseline Comparison between PID-control and the proposed ML-control.

Metric	PID Only	ML-Enhanced	Improvement
XY Error (RMS)	3.30 mm	1.83 mm	40.8% reduction
Success Rate	88.5%	94.6%	+6.3%
Cycle Time	4.70 s	4.62 s	0.08 s faster
Repeatability (2σ)	1.54 mm	0.97 mm	36.5% improvement

The improved model, depicted in Figure 11, introduces a data-driven layer employing SVM and RF algorithms to optimize motion parameters and predict the success rate of each PAPO attempt in real time.

Integrating ML algorithms into the PAPO introduces a self-adaptive layer that enables the SCARA robot to handle variability in object size, mass, and position with improved precision. The hybrid use of SVM and RF allows both categorical classification (task success) and continuous prediction (expected error and completion time). This combination effectively improves decision quality and reduces manual-tuning efforts, which are common in purely rule-based control systems. Figure 12 shows the PAPO for the implemented SCARA robot test where the implementing robot is recognized and trained to recognize different colors and geometric shapes.

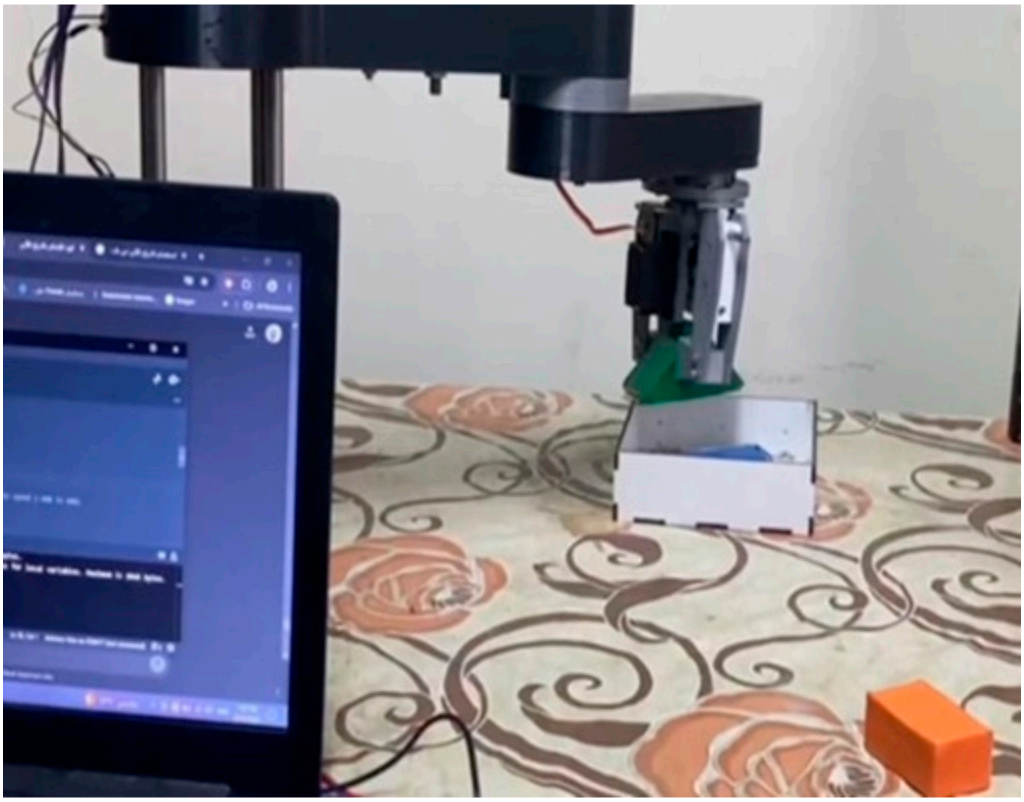


Figure 12. Pick-and-place SCARA robot test.

Some M4-bolts and nuts to secure the gripper to the robot arm can be used. A robotic system including three stepper-motors and a servo-motor is intended to be controlled by the PAPO code. It makes it possible to

- Use limit switches to homing the motors.
- Use serial input to receive movement and position commands.
- Manage a servo-gripper-motor.
- Store and carry out movement patterns.

As it continuously checks for fresh input, the code is set up to parse commands, store positions, regulate serial communication, and execute those positions.

4. Laser Engraving with SCARA Robot Configuration

High levels of precision and accuracy are required for the elaborate designs or letters that can be engraved on a variety of materials using the laser engraving method [32]. This essay also discusses engraving on wood surfaces, which requires adaptability and agility to enable the creation of various designs. The four-DOF SCARA robot and the laser engraver are used together for this purpose. By employing the G-code and M-code instructions for the engraving path, it will assist in creating precise and exact design paths that the laser will trace and engrave on the wood surface.

A variety of power levels are available for laser modules, ranging from low-power (5 mW) for applications involving alignment and pointing to high-power (e.g., 40 W) for industrial engraving and cutting. The particular needs of the application, such as the material being processed, the needed speed, and the level of precision, will determine which laser module is best. Quality, dependability, and suitability for the intended use are all crucial considerations when choosing a laser module. Laser modules from trusted manufacturers deliver consistent performance and adhere to high performance standards [23,27].

Another end effector, in this case a laser plotter, will be utilized in this section of the SCARA robot's practical application to carry out extremely precise drilling and engraving tasks. The detailed explanation of the design, mechanical assembly, and implementation of the practical part of the implementing robot was previously provided. Therefore, in this part, one will only address the control part and the practical implementation of using the laser in engraving, in order not to prolong the process. The limit switches, which are typically closed with two wires going to the end-stops S and pins for each axis, respectively, are directly above the stepper-motors. Our specific laser module operates at 12 V and requires only two wires—Ground and VCC—to be controlled by a PWM signal.

The three-pin connector that is included with the module to attach the laser module has been utilized. In this situation, the white-wire in the middle is the VCC and the black- wire on the left is ground. To make it simple to connect these wires to the end effector's wires, two male-jumper wire connectors have been connected to them. After that, the connector is inserted; the wire through one of the holder's apertures can be inserted and fastened to the laser module with two M3-bolts. The laser module to the robot with four M4-bolts has been fastened. Figure 13 displays laser engraving with the SCARA robot setup.

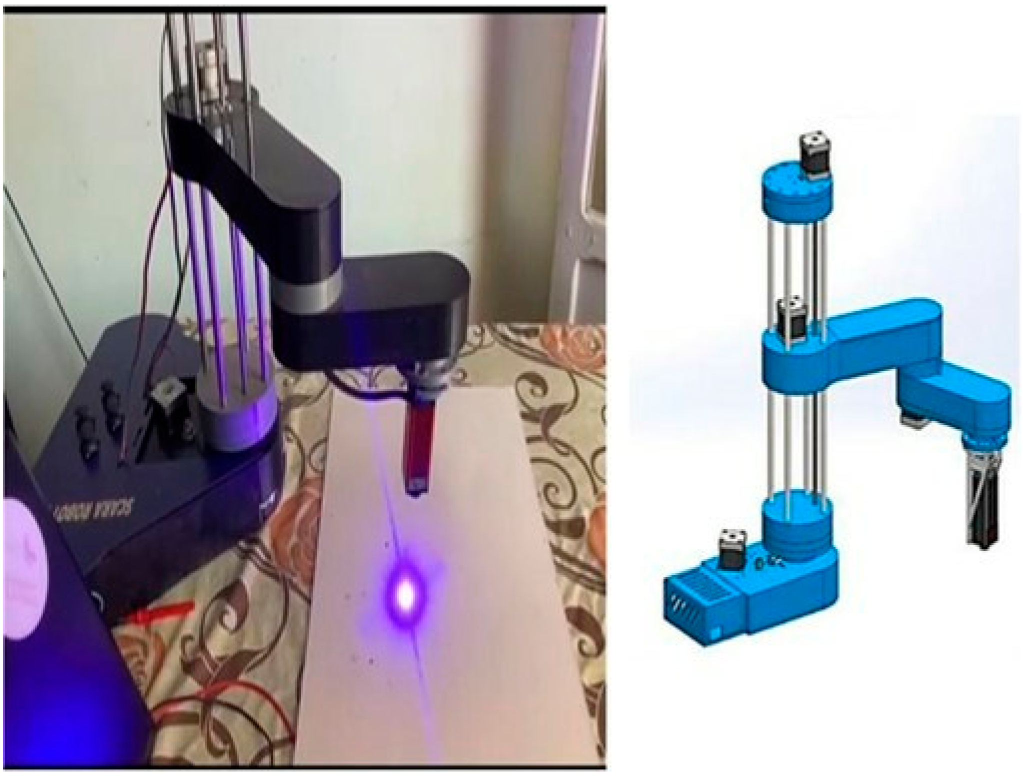


Figure 13. Laser engraver configuration with SCARA Robot.

4.1. Laser Engraver Electronic Configuration

An Arduino-MEGA-board is needed to control and handle the motors and sensors in this case, along with 4 NEMA 17 stepper-motors [33] for movement on the x, y, and z axes, a RAMPS 1.6 board, an Arduino-shield that can control stepper-motor drivers and power-heating elements, take temperature readings, and perform other tasks required for

3D-printing, and finally, a laser module for engraving. A well-liked open-source electronics control board, RAMPS 1.6, is frequently seen in CNC-machines and 3D-printers. Based on the Arduino-Mega (2560), it offers a platform for managing fans, heaters, stepper-motors, and more. Pin D9, which controls a 3D-printer setup, will be used to control the laser module. Because this pin is PWM enabled, it can be used to adjust the laser's intensity. Make sure the heat sinks are free before inserting the cables into the casing. A 12 V power-supply with a 4 A minimum current rate to power everything is required.

Here, firstly, the drivers' step resolution has been chosen. Naturally, three jumpers have been added for each stepper-driver to choose the highest resolution for smoother motion. The 16th step resolution A4988 stepper-driver has been used. For either personal or professional use, the robot (final product) can be used to carve designs into wood. The addition of a Z Axis Focus-Master-Kit, Belt-Tensioner-Set, and Retainer-feet could further enhance the robot. Figure 14 shows the laser-engraving SCARA robot circuit diagram.

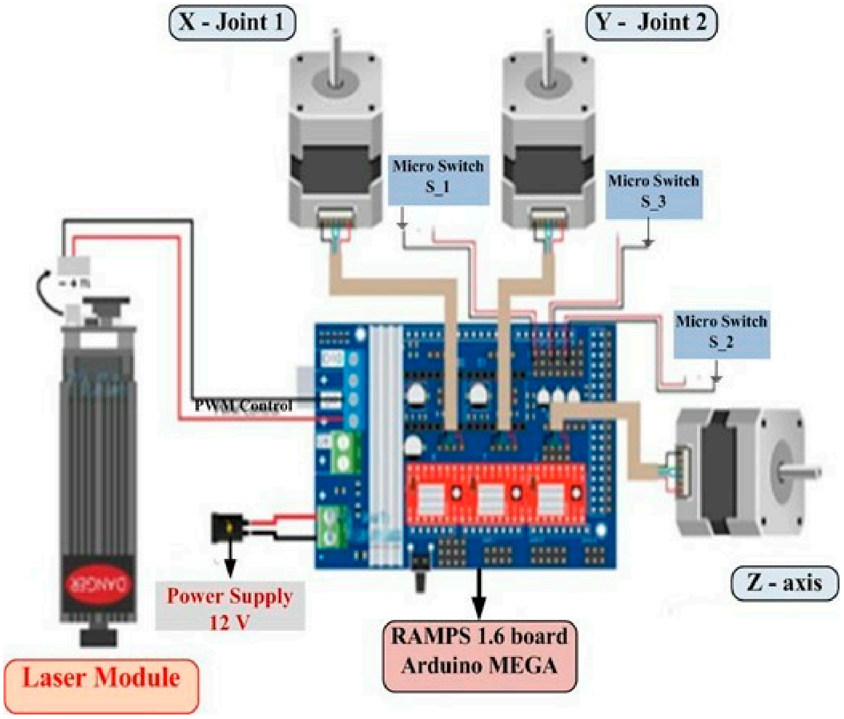


Figure 14. Laser-engraving SCARA robot circuit diagram.

4.2. Laser Engraving Accuracy, Repeatability, and Compliance Evaluation

An experiment was conducted to measure engraving accuracy, reproducibility, and the effect of SCARA standard compliance on mark quality. A 5 × 5 calibration grid was printed at 10 mm spacing onto a wooden board. The implemented robot was instructed to engrave 25 points corresponding to the centers of the printed grid cells. The engraved pattern was photographed, using a metric ruler to determine the scale, and the offsets between the printed reference points and the engraved points were measured. To assess the positioning fidelity of the engraving process, the Euclidean distance was evaluated for every point by comparing the intended reference coordinates with the corresponding engraved marks. The quantitative assessment revealed a mean positional deviation of 0.45 mm, while the maximum recorded discrepancy reached 0.90 mm, with a standard deviation of 0.28 mm. Taken together, these metrics indicate that the developed SCARA-platform maintains a stable and controlled engraving performance, providing precision levels that align well with the requirements of light duty and educational laser engraving tasks.

The engraving sequence was repeated under identical conditions, and the displacement among corresponding points was used to assess motion consistency. The results, summarized in Table 14, show a mean repeatability error of 0.25 mm and a maximum deviation of 0.62 mm, indicating that the robot can reproduce consecutive paths with stable behavior.

Table 14. Quantitative evaluation of laser engraving performance.

Metric	Value
Number of engraved points	25
Grid spacing	10 mm
Mean accuracy error	0.45 mm
Maximum accuracy error	0.90 mm
Accuracy standard deviation	0.28 mm
Repeatability error (mean)	0.25 mm
Maximum repeatability deviation	0.62 mm
Compliance effect (outer region vs. center)	≈20% increaseinerror

Workspace compliance was examined by comparing accuracy at the outer and central regions. Points engraved near the boundary showed roughly a 20% increase in error due to reduced stiffness at full extension, whereas central points exhibited only small-deviations. As also reported in Table 14, the achieved accuracy range of 0.45–0.90 mm is comparable to typical low-cost Cartesian engravers, while the SCARA system provides greater workspace flexibility.

The processing software can be executed once the G-code and M-code instructions have been uploaded to Arduino. The SCARA robot will move to its home-position once the electricity is supplied. After that, the robot can be utilized to work automatically or move reduce operator. The laser module and all joints were implemented and operated as expected and tested by engraving two designs with high-precision on two pieces of wood, as shown in Figure 15.

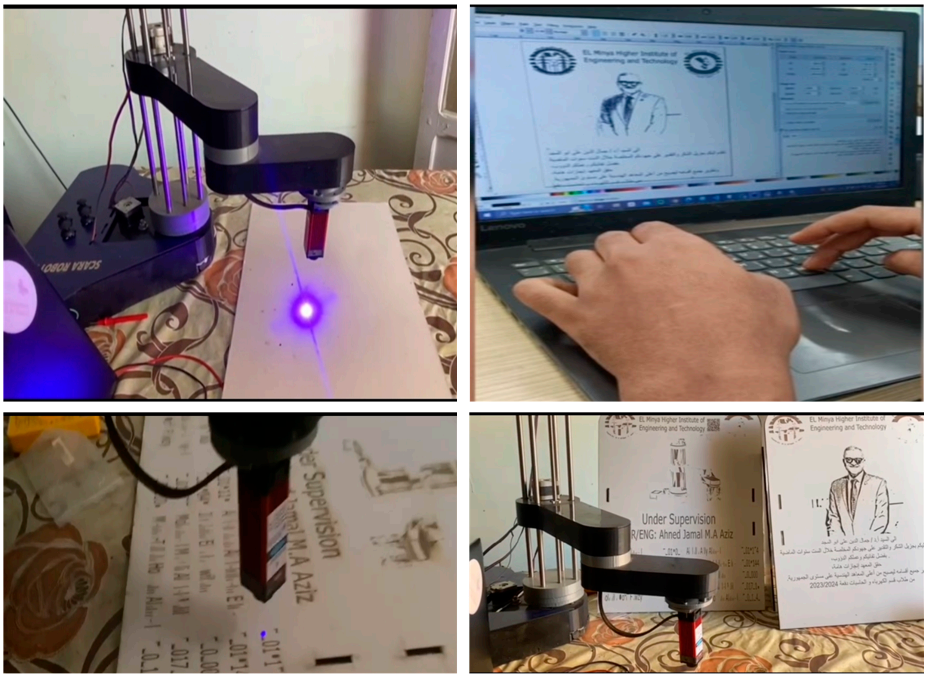

Under Supervision
R/ENG: Ahmed Jamal M.A Aziz
EL Minya Higher Institute of Engineering and Technology

Figure 15. Laser-engraving SCARA robot test.

5. Conclusions

The design, modeling, and experimental evaluation of the low-cost 4-DOF SCARA robot have demonstrated high-precision, repeatability, and operational reliability. Kinematic analyses and stress simulations ensured mechanical stability, while MATLAB-based modeling was validated against CAD simulations. The integration of ML, combining SVM for grasp success prediction and RF for XY error and cycle time estimation, significantly enhanced performance. Experimental results show 94.4% SVM classification accuracy, 1.84 mm RMSE in positional error prediction, and 0.41 s RMSE in cycle time prediction. Repeatability tests report 0.97 mm ISO-standard repeatability and laser engraving trials achieved mean positional errors of 0.45 mm with a maximum of 0.90 mm. Compared to a baseline PID controller, the ML-enhanced strategy reduced RMS positioning error by 40.8%, improved repeatability by 36.5%, and increased task success rate to 94.6%. These results confirm that the developed SCARA platform is capable of precise, consistent, and flexible operation for educational, research, and light-duty practical applications. Future work will focus on faster control hardware, expanded AI modules, and autonomous path planning to further improve accuracy and adaptability.

Author Contributions: Conceptualization, A.G.M.A.A., A.A.Z.D. and S.A.D.; methodology, A.G.M.A.A. and A.A.Z.D.; software, A.G.M.A.A.; validation, A.G.M.A.A., A.E.R., A.G.A.E.-M. and A.A.Z.D.; formal analysis, A.G.M.A.A.; investigation, A.G.M.A.A.; resources, A.G.M.A.A. and A.E.R.; data curation, A.G.M.A.A. and A.G.A.E.-M.; writing—original draft preparation, A.G.M.A.A. and A.A.Z.D.; writing—review and editing, A.G.M.A.A., S.A.D., A.E.R., A.G.A.E.-M. and A.A.Z.D.; visualization; supervision, A.G.M.A.A.; project administration, A.G.M.A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article.

Acknowledgments: The author, Ahmed G. Mahmoud A. Aziz, who is the supervisor of the project, would like to express his deepest gratitude to the fourth-year students of the Electrical and Computers Engineering Department at the Higher Institute of Engineering and Technology, New Minia, Minia, Egypt, for their participation in the practical part of this research. Additionally, he is grateful to the Higher Institute of Engineering and Technology, New Minia, Minia, Egypt for providing the necessary resources and academic environment that facilitated this work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhang, Z.; Meng, Q.; Cui, Z.; Yao, M.; Shao, Z.; Tao, B. Machine Learning Applications in Parallel Robots: A Brief Review. *Machines* **2025**, *13*, 565. [CrossRef]
2. Voglewede, P.; Smith, A.H.; Monti, A. Dynamic performance of a SCARA robot manipulator with uncertainty using polynomial chaos theory. *IEEE Trans. Robot.* **2009**, *25*, 206–210. [CrossRef]
3. Noshahi, S.F.; Farooq, A.; Irfan, M.; Ansar, T.; Chumuang, N. Design and fabrication of an affordable scara 4-dof robotic manipulator for pick and place objects. In Proceedings of the 2019 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), Chiang Mai, Thailand, 30 October–1 November 2019; pp. 1–5.
4. Westerlund, L. *The Extended Arm of Man: A History of Industrial Robot*; Informationsförlaget: Stockholm, Sweden, 2000.
5. Makino, H.; Furuya, N.; Soma, K.; Chin, E. Research and development of the scara robot. In Proceedings of the 4th International Conference on Production Engineering, Tokyo, Japan, 26–27 August 1980; pp. 885–890.
6. Jin, G.; Yu, X.; Chen, Y.; Li, J. Scara+ system: Bin picking system of revolution-symmetry objects. *IEEE Trans. Ind. Electron.* **2024**, *71*, 10976–10986. [CrossRef]
7. Ali, H.S.; Boutat-Baddas, L.; Becis-Aubry, Y.; Darouach, M. H∞ control of a SCARA robot using polytopic LPV approach. In Proceedings of the 2006 14th Mediterranean Conference on Control and Automation, Ancona, Italy, 28–30 June 2006; pp. 1–5.

https://doi.org/10.3390/automation7010011

8. Bhatia, P.; Thirunarayanan, J.; Dave, N. An expert system-based design of SCARA robot. *Expert Syst. Appl.* **1998**, *15*, 99–109. [CrossRef]
9. He, Y.; Mai, X.; Cui, C.; Gao, J.; Yang, Z.; Zhang, K.; Chen, X.; Chen, Y.; Tang, H. Dynamic modeling, simulation, and experimental verification of a wafer handling SCARA robot with decoupling servo control. *IEEE Access* **2019**, *7*, 47143–47153. [CrossRef]
10. Omari, A.; Ming, A.; Limam, G.; Kanamori, C.; Kajitani, M.; Nakamura, S. Development of a High Precision Mounting Robot System With Fine Motion Mechanism (3rd Report) Positioning Experiments of SCARA Robot with Fine Mechanism. *J. Jpn. Soc. Precis. Eng.* **2001**, *67*, 1101–1107. [CrossRef]
11. Sahari, K.S.M.; Weng, K.H.; Han, Y.W.; Anuar, A.; Baharuddin, M.Z.; Mohideen, S.S.K. Design and Development of a 4â€"Dof Scara Robot for Educational Purposes. *J. Teknol. (Sci. Eng.)* **2011**, *54*, 193–215.
12. Manjunath, T.C.; Ardil, C. Development of a Jacobean model for a 4-Axes indigenously developed SCARA system. *World Acad. Sci. Eng. Technol. Int. J. Mech. Aerosp. Ind. Mechatron. Manuf. Eng.* **2007**, *2*, 1180–1186.
13. Jahanshahi, H.; Zhu, Z.H. Review of machine learning in robotic grasping control in space application. *Acta Astronaut.* **2024**, *220*, 37–61. [CrossRef]
14. Li, Y.H.; Ma, Y.; Liu, S.T.; Luo, Z.J.; Mei, J.P.; Huang, T.; Chetwynd, D.G. Integrated design of a 4-DOF high-speed pick-and-place parallel robot. *CIRP Ann.* **2014**, *63*, 185–188. [CrossRef]
15. Han, Z. Multimodal intelligent logistics robot combining 3D CNN, LSTM, and visual SLAM for path planning and control. *Front. Neurorobotics* **2023**, *17*, 1285673. [CrossRef]
16. Cong, V.D.; Hiep, T.T. Support vector machine-based object classification for robot arm system. *Int. J. Electr. Comput. Eng.* **2023**, *13*, 5047–5053. [CrossRef]
17. Soori, M.; Arezoo, B.; Dastres, R. Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cogn. Robot.* **2023**, *3*, 54–70. [CrossRef]
18. Bouzid, R.; Gritli, H.; Narayan, J. ANN approach for SCARA robot inverse kinematics solutions with diverse datasets and optimisers. *Appl. Comput. Syst.* **2024**, *29*, 24–34. [CrossRef]
19. Peng, S.; Hu, T. Kinematic Analysis of SCARA Robots. *J. Comput. Electron. Inf. Manag.* **2025**, *17*, 51–57. [CrossRef]
20. Hong, K.-S.; Choi, K.-H.; Kim, J.-G.; Lee, S.J.R.; Manufacturing, C.-I. A PC-based open robot control system: PC-ORC. *Robot. Comput.-Integr. Manuf.* **2001**, *17*, 355–365. [CrossRef]
21. Er, M.J.; Lim, M.T.; Lim, H.S. Real-time hybrid adaptive fuzzy control of a SCARA robot. *Microprocess. Microsyst.* **2001**, *25*, 369–378. [CrossRef]
22. Tsai, L.-W. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1999; p. 505.
23. Bhargava, A.; Kumar, A. Arduino controlled robotic arm. In Proceedings of the 2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 20–22 April 2017; Volume 2, pp. 376–380.
24. Huang, H.; Arogbonlo, A.; Yu, S.; Kwek, L.C.; Lim, C.P. Trajectory tracking of a SCARA robot using intelligent active force control. *Neural Comput. Appl.* **2025**, *37*, 13345–13370. [CrossRef]
25. Nansai, S.; Kamamichi, N.; Naganawa, A. Implementation of Path-Following Control of Lizard-Inspired Single-Actuated Robot Utilizing Inverse Kinematics. *Automation* **2025**, *6*, 74. [CrossRef]
26. Aziz, A.G.M.A.; Abdelaziz, A.Y.; Ali, Z.M.; Diab, A.A.Z. A Comprehensive Examination of Vector-Controlled Induction Motor Drive Techniques. *Energies* **2023**, *16*, 2854. [CrossRef]
27. Lee, H.-W. The study of mechanical arm and intelligent robot. *IEEE Access* **2020**, *8*, 119624–119634. [CrossRef]
28. Pelossof, R.; Miller, A.; Allen, P.; Jebara, T. An SVM learning approach to robotic grasping. In Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA'04), New Orleans, LA, USA, 26 April–1 May 2004; Volume 4, pp. 3512–3518.
29. Xie, B.; Qi, Y.; Su, W. RBF Network Adaptive Control of SCARA Robot Based on Fuzzy Compensation. *J. Phys. Conf. Ser.* **2020**, *1650*, 032006. [CrossRef]
30. Aziz, A.G.M.A. A Robust Machine Learning Framework for Partial Discharge Diagnosis in Diverse Power Substations. *Int. J. Artif. Intell. Emerg. Technol.* **2024**, *7*, 42–52. [CrossRef]
31. *ISO 9283*; Manipulating Industrial Robots—Performance Criteria and Related Test Methods. International Organization for Standardization: Geneva, Switzerland, 1998.
32. Sanjeewa, N.; Wathudura, V.M.; Kahatapitiya, N.S.; Silva, B.N.; Subasinghage, K.; Wijesinghe, R.E. Real-Time Coordinate Estimation for SCARA Robots in PCB Repair Using Vision and Laser Triangulation. *Instruments* **2025**, *9*, 7. [CrossRef]
33. Chenchireddy, K.; Dora, R.; Mulla, G.B.; Jegathesan, V.; Sydu, S.A. Development of robotic arm control using Arduino controller. *Int. J. Robot. Autom.* **2024**, *264*, 1024–1064. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

https://doi.org/10.3390/automation7010011