# Case of Study: Kinematics Analysis, Workspace, Simulation and Trajectory Planning of a 4-Axis Serial Robot

ABDELILAH EL MERINI[1,*], CHAIMAE OUCHICHA[2], OUAFAE HAMDOUN[3]
Physics Department, Faculty of Sciences, Department of Management Assistance Techniques, FAMISDS Lab, Department of New Technologies
Abdelmalek Essaadi University, ENCG Hassan 1st University, Sultan Moulay Slimane University
Tetouan, Settat, Beni Mellal
MOROCCO

*Abstract:* - Selective Compliance Assembly Robot Arm (SCARA) manipulators are extensively employed in automated manufacturing due to their ability to deliver high-speed, precise, and repeatable motion in assembly tasks. This study presents kinematic modeling and validation of a four-degree-of-freedom SCARA with an RRPR (Revolute-Revolute-Prismatic-Revolute) configuration. The forward and inverse kinematic equations were formulated by applying the Denavit-Hartenberg convention, with algebraic and geometric approaches to ensure accurate mapping between joint parameters and end-effector positions. The accuracy of the kinematical models was validated in a MATLAB-based simulation environment with the GUIDE interface, enabling visualization of joint behavior and end-effector trajectory in a 3D workspace. A full factorial design experiment ($3^3$), consisting of 27 joint parameter combinations, assessed the positional output (Px, Py, Pz) under different configurations. In addition, optimal trajectory planning moved the robot between specified points in 4.5 seconds using a fixed time step and 451 interpolated points. Cross-validation using RoboAnalyzer demonstrated high fidelity in spatial and temporal trajectory generation. Finally, a workspace simulation with 250,000 random joint variable samples confirmed the complete workspace and mechanical constraints of the manipulator. The MATLAB workspace boundary closely agreed with RoboAnalyzer, validating the derived kinematic model for robotic motion analysis and industrial automation applications.

*Key-Words:* - Kinematics modelling, D-H parameters, SCARA Robot, Robotics, Workspace, Trajectory planning, Design experiment.

Received: January 25, 2026. Revised: April 9, 2026. Accepted: May 2, 2026. Published: September 9, 2026.

## 1 Introduction

Selective Compliance Assembly Robot Arms (SCARA) represent one type of serial manipulator distinguished by a planar structure and selective compliance horizontally, making them appropriate for tasks needing precision and speed such as pick-and-place, assembly, and surface processing, [1]. These robots typically feature a configuration composed of revolute and prismatic joints and operate in an open-loop kinematic chain, allowing for straightforward modeling and control.

In industrial robots, commands are typically given in Cartesian space, however, control signals are expressed in joint space. This requires solid knowledge of the forward and inverse kinematics to solve the transformation between joint values and the position-orientation of the end effector. Forward kinematics (FK) finds the pose of the end-effector given joint parameters, while inverse kinematics (IK) solves the inverse problem, which is usually represented by nonlinear equations with possibly multiple solutions, [2]. Understanding both is essential for applications including path planning, workspace analysis, and robot control. Simulation tools (e.g., MATLAB, CAD-based) are a key factor in the development of a robot system, since engineers can validate the theoretical model and the capabilities of a motion planning strategy before its implementation in a real scenario.

In [3], a detailed mathematical model for the SCARA manipulator, which takes into account the dynamics of the servomotor, was introduced, and its dynamic simulation was showcased. Besides the equation of motion, forward and inverse kinematics are included. The software utilized, Solid Dynamics, was implemented to design a SCARA robot for drilling, while the efficiency of the robot-actuator system was tested using MATLAB/Simulink.

In [4], the authors focused on the readjustment of a five-degree-of-freedom robot originally built in the

1980s for educational purposes. They addressed the inverse kinematics problem using a geometric approach and developed a control system based on a Cypress PSoC 5LP platform. The refurbished robot serves as a practical tool for teaching robotics, enabling students to understand the structure of the robot and its inverse kinematics problem.

In [5], a new redundant SCARA robot, which is at the conceptual design stage, was investigated. Computer simulations with typical pick-and-place tasks were used for comparison of kinematic and dynamic manipulability of the new concept with the conventional system of the same manufacture. An extra revolute joint was added in the SCARA robot kinematics for enhanced kinematic and dynamic performance at increased complexity. The authors pointed out the potential of a redundant SCARA construction to decrease cycle times for multiple applications and pointed out that optimization of manipulator, mechanics, and control should be considered.

In [6], the authors suggested an integrated treatment of robotics education which fuses offline simulation with physical prototyping of robots. They implemented a motion planning module that was new to RoboAnalyzer, a 3D model-based software, which is used to teach robotics. Kinematic analysis and motion planning were realized by the module to realize the linear and circular motion of the end-effector point. When the off-line simulation was satisfactory, sending the joint trajectories to an Arduino IDE through serial port communication controlled the motors of a physical prototype, which was constructed with Dynamixel motors. This strategy was designed to give students practical knowledge of both simulated and real-world implementations, as well as the operation of a ROS-based robot.

Building upon advancements in robotic control systems, significant strides have been made in the development and analysis of SCARA robots. Notably, a comprehensive mathematical model for the SCARA robot was developed, incorporating the design and implementation of an Adaptive Neuro-Fuzzy Inference System controller. This model allows accurate control and adapts to diverse operating conditions, [7]. Moreover, four ANFMs have also been developed to address the inverse kinematics problem of the SCARA robot, [8], where the solution returned is both accurate and relatively fast when compared to solving the set of highly nonlinear equations. In addition to these works, kinematic and design analysis studies of an RRPR robot arm based on Denavit-Hartenberg (DH) notation and MATLAB Simulink were also reported, [9], which presented important references for the kinematic behavior and design considerations by which the robotic system was implemented.

Artificial Neural Networks (ANN) have attracted a lot of attention due to their flexibility and great capability to approximate a nonlinear function. The capability of ANNs in solving complex multidimensional and multi-criteria problems has made them a primary option in computer modeling engineering, [10]. Neural Networks have been widely employed in serial robots with the aid of ANN and ANFIS, [11]. ANN in the form of a multi-layer perceptron (MLP) has been used to plot the forward and inverse kinematic problems of a six-degree-of-freedom (DOF) robot manipulator with the aim of reducing the system error, [12]. Here, some authors have introduced a structured ANN method based on minimizing the mean square error to solve the inverse kinematic problem of the 4-axes SCARA robot manipulator, [13]. Other recent studies have explored various aspects of SCARA manipulators, including kinematic modeling, trajectory planning, and workspace analysis, [14], [15], [16].

The primary objective of this study is to model a comprehensive kinematic model for a 4-degree-of-freedom SCARA manipulator with an RRPR (Revolute-Revolute-Prismatic-Revolute) configuration. The Denavit-Hartenberg (D-H) parameterization serves to establish both forward and inverse kinematic equations that link joint parameters to end-effector positions.

A user-friendly graphical user interface (GUI) is implemented within MATLAB's GUIDE environment to simulate and visualize the manipulator's motion. This interface allows for real-time interaction, enabling users to input joint parameters and observe the corresponding end-effector positions in a 3D workspace. Additionally, the Jacobian matrix of the manipulator is examined to support trajectory planning and workspace analysis.

Trajectory planning is a critical aspect of robotic motion. In this study, time-parameterized paths are generated to guide the manipulator's movement through specified waypoints. The workspace is thoroughly analyzed by simulating a vast number of joint configurations, allowing for the visualization of reachable areas and the assessment of potential obstacles.

To ensure the accuracy and reliability of the model, simulation results from the MATLAB environment are compared with those obtained from specialized robotic simulation software.

The remainder of the paper proceeds as follows: Section 2 outlines the SCARA kinematics modeling process, including the Denavit-Hartenberg parameterization, forward kinematic equations, inverse kinematics formulation, and implementation of a GUI-based simulator. Section 3 presents the experimental study using a full factorial design to analyze the influence of joint parameters on end-effector position, followed by a numerical confirmation of inverse kinematics solutions. Section 4 details the trajectory planning methodology, illustrating the generation and validation of a multi-point path using MATLAB and RoboAnalyzer. Section 5 focuses on workspace simulation using 250,000 joint configurations, comparing the MATLAB-generated results with those from RoboAnalyzer for validation. Finally, Section 6 summarizes the findings and proposes directions for future research.

# 2 Scara Kinematics Modeling

## 2.1 DH Convention

The Denavit-Hartenberg convention is a widely employed method in the field of robotics that aims to assign a coordinate frame to each joint in a robotic arm. It links two adjacent coordinate systems to describe their relative position by means of a homogeneous transformation matrix, [17]. This method consists of determining four parameters to build a DH table as shown in Table 1, where joint angles θi and the third link offset d3 are joint variables. The link lengths are assigned as follows: L1= 250 mm; L2 = 150 mm; d1=300 mm and d4= 50mm.

Table 1. DH parameters of SCARA Robot

*Source: created by the authors*

| Joints (i) | θi (°) | $d_i$(mm) | $a_i$(mm) | $\alpha_i$(°) |
|---|---|---|---|---|
| 1 | $\theta_1$ | $d_1$ | $L_1$ | 0 |
| 2 | $\theta_2$ | 0 | $L_2$ | π |
| 3 | 0 | $d_3$ | 0 | 0 |
| 4 | $\theta_4$ | $d_4$ | 0 | 0 |

The configuration depicted in Figure 1 is an RRPR SCARA manipulator that has three revolute, joints θ1, θ2, and θ4, and a prismatic joint, d3, as variable joints. The specific coordinate systems are established using the Denavit-Hartenberg notation.

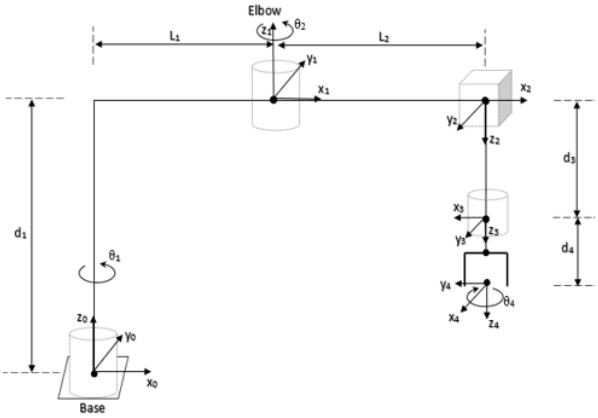


Fig. 1: Coordinate systems of 4DOF SCARA Robot

*Source: created by the authors*

## 2.2 Forward Kinematics Modeling Equations

Forward kinematics is a transformation matrix manipulation, which proceeds link by link using the Denavit-Hartenberg notation and frames. For a rigid arm with $n$ degrees of freedom, the nonlinear relationship between joint coordinates $q$ and end-effector position/orientation x at an instant of time can be expressed by the forward kinematic equation given in Eq. (1):

$$x = f(q) \quad (1)$$

where $f(.)$ is a nonlinear differential function.

By using the principles of the standard DH method and coordinate systems, the homogeneous transformation matrix represents the orientation and position of the end-effector in frame i relative to its previous coordinate frame i-1, as given in Eq. (2), [18].

$$^{i-1}T_i = Trans(z_{i-1}, d_i) Rot(z_{i-1}, \theta_i) Trans(x_{i-1}, a_i) Rot(x_{i-1}, \theta_i)$$

$$^{i-1}T_i = \begin{bmatrix} C\theta_i & -S\theta_i\, C\alpha_i & S\theta_i\, S\alpha_i & a_i\, C\theta_i \\ S\theta_i & C\theta_i\, S\alpha_i & -C\theta_i\, S\alpha_i & a_i\, S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Through the DH table shown in Table 1, we can obtain four transformation matrices between each pair of adjacent frames.

$$^{0}T_1 = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & a_1\, C\theta_1 \\ S\theta_1 & C\theta_1 & 0 & a_1\, S\theta_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{1}T_2 = \begin{bmatrix} C\theta_2 & S\theta_2 & 0 & a_2\, C\theta_2 \\ S\theta_2 & -C\theta_2 & 0 & a_2\, S\theta_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$ {}^{2}T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{3}T_4 = \begin{bmatrix} C\,\theta_4 & -S\,\theta_4 & 0 & 0 \\ S\,\theta_4 & C\,\theta_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3) $$

From the given joint parameters, we can get the SCARA end-effector position (Px, Py, Pz) and orientation {(nx, ny, nz), (ox, oy, oz), (ax, ay, az)} by multiplying the four transformation matrices from the first joint to the last one, as expressed in Eq. (3), resulting in Eq. (4).

$$ {}^{0}T_4 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4) $$

$$ {}^{0}T_4 = {}^{0}T_1\,{}^{1}T_2\,{}^{2}T_3\,{}^{3}T_4 $$

$$ = \begin{bmatrix} C(\theta_1+\theta_2-\theta_4) & S(\theta_1+\theta_2-\theta_4) & 0 & a_1\,C\theta_1 + a_2\,C(\theta_1+\theta_2) \\ S(\theta_1+\theta_2-\theta_4) & -C(\theta_1+\theta_2-\theta_4) & 0 & a_1\,S\theta_1 + a_2\,S(\theta_1+\theta_2) \\ 0 & 0 & -1 & d_1-d_3-d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Moreover, our research study is focused on verifying the position of the end-effector in the reference coordinate system as well as the results of joint angles by means of the Robotics GUIDE interface. Therefore, the following equations represent the position of the 4-DOF SCARA arm for forward kinematics, as given in Eq. 5, Eq. 6 and Eq. 7.

$$ P_x = a_1\,\cos\theta_1 + a_2\,\cos(\theta_1+\theta_2) \quad (5) $$

$$ P_y = a_1\,\sin\theta_1 + a_2\,\sin(\theta_1+\theta_2) \quad (6) $$

$$ P_z = d_1 - d_3 - d_4 \quad (7) $$

## 2.3 Inverse Kinematics Equations of Scara Manipulator

The desired end-effector position and orientation x in the inverse kinematics can be solved by Eq. 8

$$ q = f^{-1}(x) \quad (8) $$

where $q$ is the desired end effector position.

There are several solutions to the inverse kinematics problem. By analyzing the different conditions, a comprehensive analytical solution to the inverse kinematics of the manipulator has been obtained. To determine the correct solution for studying inverse kinematics, the variables of all the joints are calculated and compared using a numerical forward kinematics solution, as given in Eq. 9, Eq. 10 and Eq. 11.

$$ \theta_2 = \pm\,\text{Arccos}\left(\frac{P_x{}^2 + P_y{}^2 - a1^2 - a2^2}{2*a1*a2}\right) \quad (9) $$

$$ \theta_1 = \pm\text{Arcsin}\left(\frac{P_y(a1 + a2*C2) - P_x*a2*a2^2}{P_x{}^2 + P_y{}^2}\right) \quad (10) $$

$$ \theta_4 = \pm\,Tan^{-1}\left(\frac{n_x*S12 + n_y*C12}{n_x*C12 + n_y*S12}\right) \quad (11) $$

## 2.4 Robot Kinematics Simulation and Guide Interface

In order to simulate our studied manipulator, we used the Robotic Toolbox, [14], which deals with the kinematics and dynamics of manipulators by integrating a simple programming syntax. Figure 2 illustrates the validation flowchart of forward kinematics model in the Matlab program.

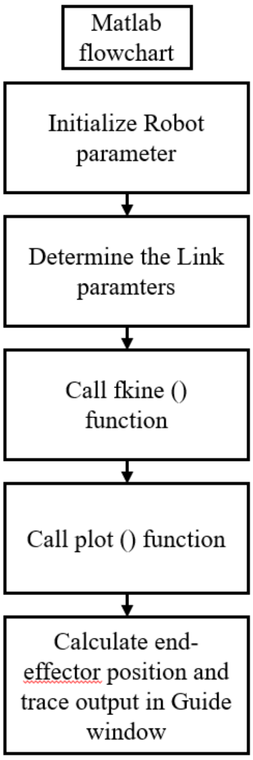


Fig. 2: The SCARA forward kinematic model flowchart

*Source: created by the authors*

The Guide window was used to calculate the end-effector position and to generate the three-dimensional simulation graphics of the SCARA robot, as shown in Figure 3.

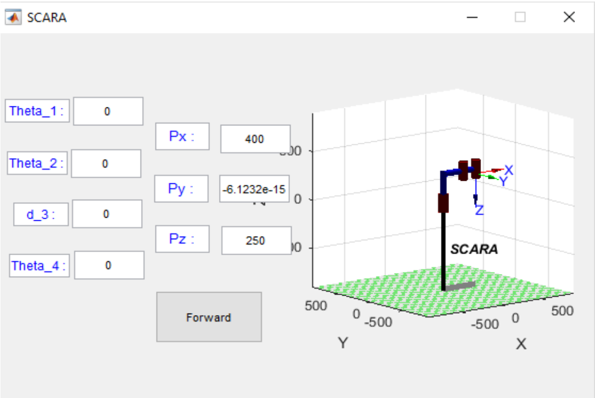


Fig. 3: SCARA home position in three-dimensional simulation

*Source: created by the authors*

## 3 Experimental Study and Px, Py, Pz Response

In this study, we opted for a full factorial design 3k, where k denotes the number of factors, and three is the number of levels per factor. It measures the response of every possible combination of all factors, [19]. We used as factors the first three quantitative parameters of a 4-axis SCARA arm, namely: A. the first joint θ1 (in degree - °); B. the second joint θ2 (in degree - °); C. the longitudinal position of the gripper d3 (in millimeters-mm). As the fourth joint has no influence on the final position vector, and only affects the rotation matrix, it was not included as a factor. Table 2 displays the levels/values for each parameter.

Table 2: Levels and factors selected for the main experiments. *Source: created by the authors*

[a] Unit.

| Level | Factors | | |
|---|---|---|---|
| | A (° [a]) | B (° [a]) | C (mm [a]) |
| 1 (-1) | -120 | -130 | 0 |
| 2 (0) | 0 | 0 | 75 |
| 3 (+1) | 120 | 130 | 150 |

The experiment consisted of 27 tests according to the previous planning (full factorial design). For each parameter setting, nine combinations were performed sequentially for each factor. The numerical responses of the final position vector as well as the inverse kinematics results were calculated using the Guide Interface of the SCARA robot.

Table 1 (Appendix) provides the end-effector position response and the respective values of the sampling. For the 27 tested process parameter settings, it is observed that the data vs. type of response are well distributed to a certain extent.

### 3.1 Three-Dimensional Response Analysis of TCP Position

This subsection focuses on the analysis of the three-dimensional response of the Tool Center Point (TCP) position in an RRPR robot manipulator. The manipulator is characterized by four inputs, namely Joint1, Joint2, d3, and Joint4. The objective is to investigate the relationship between these inputs and the TCP position coordinates Px, Py, and Pz. The analysis revealed that the 3D response of the TCP position is primarily influenced by Joint1 and Joint2. However, it is evident that the linear position of the end-effector along the Z-axis, Pz, is solely dependent on the longitudinal position of the gripper, d3, and exhibits no significant correlation with Joint1 and/or Joint2. Figure 4, Figure 5 and Figure 6 illustrate the three-dimensional responses of Px, Py, and Pz, respectively, based on various input combinations.

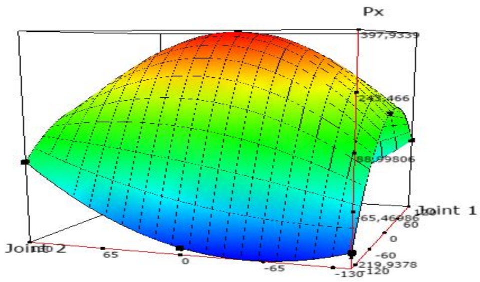


Fig. 4: 3D response for Px based on Joint1 and joint 2. *Source: created by the authors*

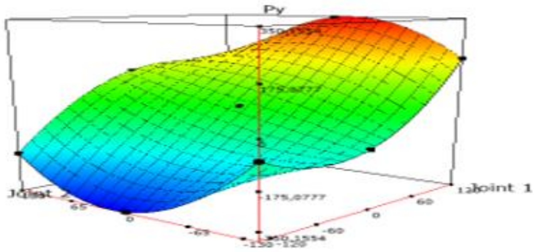


Fig. 5: 3D Response graphs for Py based on Joint1 and Joint 2. *Source: created by the authors*

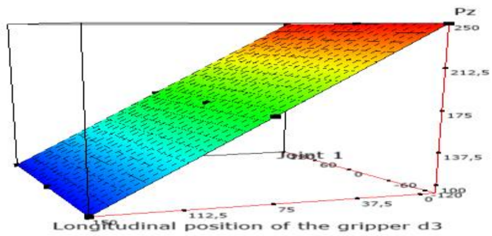


Fig. 6: 3D response for Pz based on joint1 and d3. *Source: created by the authors*

The findings show that the 3D response of the TCP position is mainly affected by Joint1 and Joint2. These two revolute joints are crucial for determining how the end-effector is oriented and positioned. By adjusting Joint1 and Joint2, it is possible to achieve various orientations and positions of the TCP within the workspace.

This analysis offers valuable insights for fine-tuning control strategies and designing robotic systems that need precise manipulation and positioning of the end-effector within a three-dimensional workspace. Future research could delve into additional optimization techniques and examine how various input parameters affect the TCP position in RRPR manipulators.

### 3.2 Confirmation of Experiment using Inverse Numerical Solution

In order to simulate the numerical inverse Kinematics problem of the studied manipulator, we used the robotic toolbox, [20], which deals with kinematics and dynamics of manipulators by integrating a programming syntax.

Table 3 presents a validation of the L27 factorial experiment design solving the inverse kinematics problem (IKP) through a numerical inverse solution developed in Matlab using Guide Interface.

Table 3: Main results for Inverse Kinematics based on Factorial Experimental Design. *Source: created by the authors*

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | Px | Py | Pz | θ1 | θ2 | d3 | θ4 |
| 2 | -0.176303 | -0.075552 | 0.25 | -120 | -130 | 0 | 0 |
| 3 | -0.176303 | -0.075552 | 0.1 | -120 | -130 | 150 | 180 |
| 4 | -0.176303 | -0.075552 | 0.1 | -120 | -130 | 75 | 90 |
| 5 | -0.2 | -0.34641 | 0.25 | -120 | 0 | 0 | 0 |
| 6 | -0.2 | -0.34641 | 0.175 | -120 | 0 | 75 | 90 |
| 7 | -0.2 | -0.34641 | 0.1 | -120 | 0 | 150 | 180 |
| 8 | 0.022721 | -0.190459 | 0.25 | -120 | 130 | 0 | 0 |
| 9 | 0.022721 | -0.190459 | 0.175 | -120 | 130 | 75 | 90 |
| 10 | 0.022721 | -0.190459 | 0.1 | -120 | 130 | 150 | 180 |
| 11 | 0.153582 | -0.114907 | 0.25 | 0 | -130 | 0 | 0 |
| 12 | 0.153582 | -0.114907 | 0.175 | 0 | -130 | 75 | 90 |
| 13 | 0.153582 | -0.114907 | 0.1 | 0 | -130 | 150 | 180 |
| 14 | 0.4 | 0 | 0.25 | 0 | 0 | 0 | 0 |
| 15 | 0.4 | 0 | 0.175 | 0 | 0 | 75 | 90 |
| 16 | 0.4 | 0 | 0.1 | 0 | 0 | 150 | 180 |
| 17 | 0.153582 | 0.114907 | 0.25 | 0 | 130 | 0 | 0 |
| 18 | 0.153582 | 0.114907 | 0.175 | 0 | 130 | 75 | 90 |
| 19 | 0.153582 | 0.114907 | 0.1 | 0 | 130 | 150 | 180 |
| 20 | 0.022721 | 0.190459 | 0.25 | 120 | -130 | 0 | 0 |
| 21 | 0.022721 | 0.190459 | 0.175 | 120 | -130 | 75 | 90 |
| 22 | 0.022721 | 0.190459 | 0.1 | 120 | -130 | 150 | 180 |
| 23 | -0.2 | 0.34641 | 0.25 | 120 | 0 | 0 | 0 |
| 24 | -0.2 | 0.34641 | 0.175 | 120 | 0 | 75 | 90 |
| 25 | -0.2 | 0.34641 | 0.1 | 120 | 0 | 150 | 180 |
| 26 | -0.176303 | 0.075552 | 0.25 | 120 | 130 | 0 | 0 |
| 27 | -0.176303 | 0.075552 | 0.175 | 120 | 130 | 75 | 90 |
| 28 | -0.176303 | 0.075552 | 0.1 | 120 | 130 | 150 | 180 |
| 29 | (IKP) = 3 Inputs | | | 4 Outputs = θi | | | |

The Figure 7 illustrates the validation flowchart of inverse kinematics model on Matlab program.

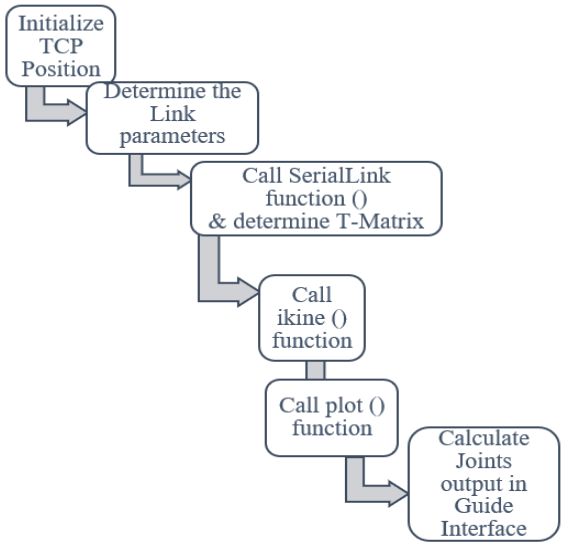


Fig. 7: Inverse kinematics model flowchart in Matlab. *Source: created by the authors*

In Figure 8, it is shown that three inputs Px, Py and Pz are required for the Inverse Kinematics Problem (IKP), which generates four outputs: θ1, θ2, d3, and θ4.

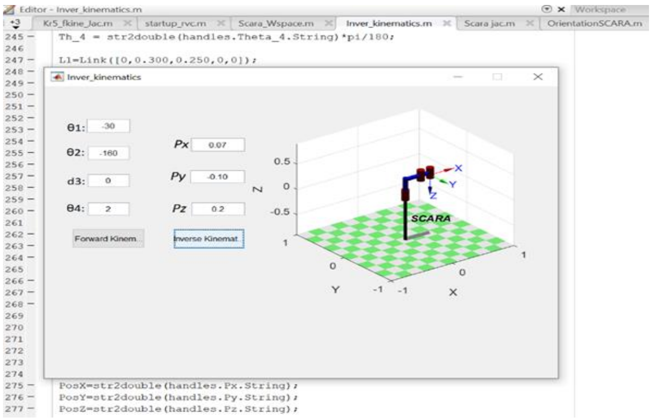


Fig. 8: Main results for Inverse Kinematics based on Numerical Solution. *Source: created by the authors*

## 4 Trajectory Planning of SCARA Robot

Trajectory planning plays a vital role in making sure of unique and efficient motion control of robot structures. The primary objective of trajectory generation is to provide inputs for the motion control machine that assures the execution of the desired trajectory. To be able to achieve this, users typically describe the desired trajectory using specific parameters. These parameters may also include the initial and final factors for point-to-point control or a

finite collection of factors along the route for movement through a series of points.
In the case of SCARA robots, trajectory planning involves determining the path that the end-effector should follow to reach the desired positions. To illustrate this, an example of a trajectory is considered as follows: Starting from an initial point represented by = [0° 0° 0m 0], we aim to reach a final point represented by = [120° 130° 0.15m 90°]. The desired motion is to be completed within a time interval of t = 4.5s, with a step size of 0.01s. This results in the generation of a path consisting of 451 points.
To implement and validate the trajectory planning, the syntax provided in MATLAB can be utilized. Figure 9 and Figure 10 showcase the implementation syntax in MATLAB, along with the validation of the matching results using RoboAnalyzer, [21]. This enables the user to visually inspect and verify the accuracy and feasibility of the generated trajectory.

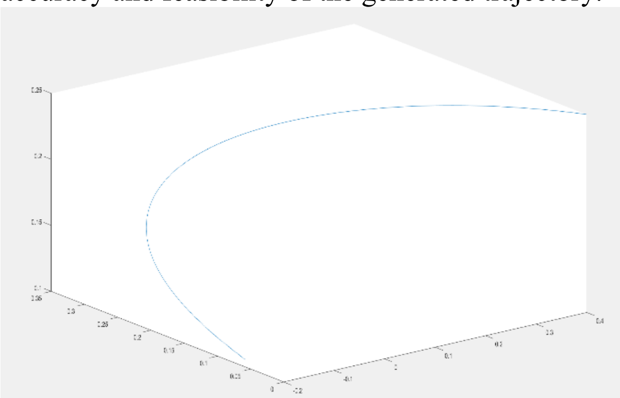

Fig. 9: Matlab Trajectory Path Planning from qA to qAB. *Source: created by the authors*

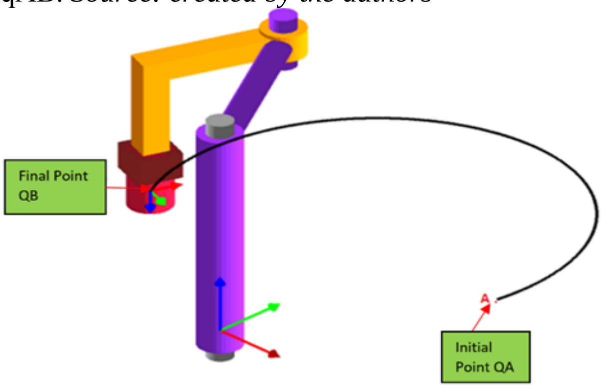


Fig. 10: Matched Results in Trajectory Planning. *Source: created by the authors.*

In conclusion, SCARA trajectory planning is essential for achieving precise motion control in robotic systems. By specifying the initial and final points or a sequence of points, users can generate trajectories that ensure the desired path is followed. Implementing the trajectory-planning algorithm in MATLAB and validating the results using tools like RoboAnalyzer aid in fine-tuning the trajectory and ensuring its successful execution.

## 5 Manipulator Workspace Simulation

The space point simulation experiment of the SCARA RRPR (4-Dof) manipulator is carried out, using the DH parameters shown in Table 1 according to Coordinate systems of 4DOF SCARA Robot. Using the Equations in Section 2.3 and the 50*50*100 randomly selected points of θ1, θ2, and d3, respectively, in the MATLAB software environment. These 250 000 samples points in the SCARA arm are used to simulate the workspace (Figure 11 and Figure 12, Own source), [14].

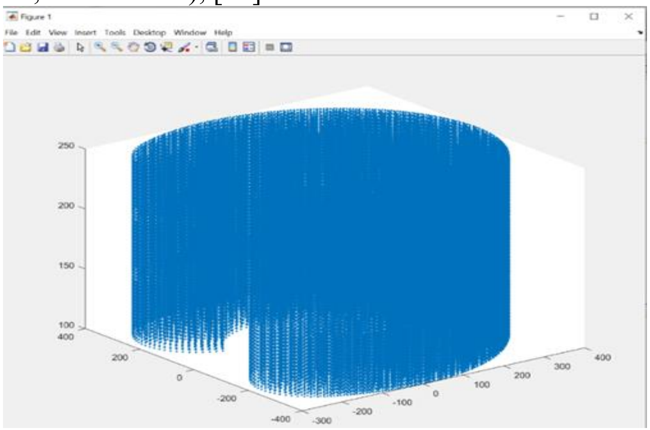

Fig. 11: SCARA Workspace simulation in Matlab.
*Source: created by the authors*

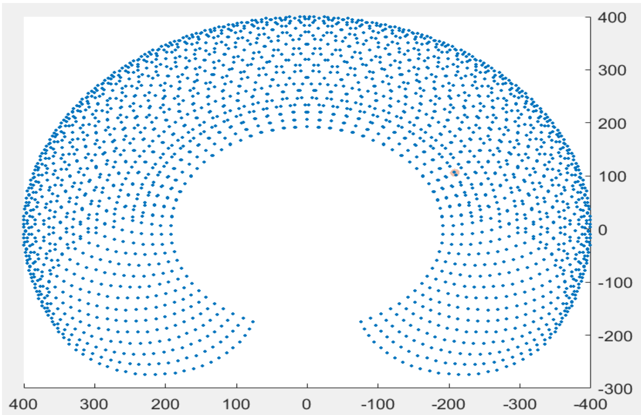


Fig. 12: Border of workspace outside area.
*Source: created by the authors.*

To validate the results obtained from our mathematical modeling and MATLAB simulations, we employed RoboAnalyzer, [21], a widely recognized educational and research tool for robotic visualization and kinematic analysis. Among the various software options available for robot motion analysis, RoboAnalyzer was selected for its intuitive

graphical interface and specialized features tailored to serial manipulators. It supports the input of Denavit–Hartenberg parameters, provides 3D visualization of robot structures, and enables simulation of both forward and inverse kinematics as well as end-effector trajectory planning. These capabilities offer essential visual feedback that facilitates verification of simulation outputs, ensuring that the modeled motion corresponds to realistic robotic behavior. This cross-validation step enhances the credibility and robustness of the proposed approach.
The kinematic simulations were conducted using Peter Corke's Robotics Toolbox v10.3 for MATLAB R2018a (64-bit). The computations were performed on a Lenovo ThinkPad X1 Yoga (2nd Gen) equipped with an Intel Core i7-7500U processor, 16 GB RAM, 512 GB SSD, and Intel HD Graphics 620. The operating system used was Windows 10 Pro (64-bit). This configuration was sufficient to perform real-time simulations and graphical renderings without performance limitations, [22].

## 6 Conclusion and future work

This study focused on the analytical investigation of the forward and inverse kinematics of a 4-degree-of-freedom SCARA manipulator with an RRPR configuration, aiming to accurately model and simulate the motion of the end-effector. The kinematic modeling was carried out using the Denavit-Hartenberg convention, combined with algebraic and geometric methods to precisely solve the inverse kinematics problem. To validate these theoretical models, a graphical user interface was developed in MATLAB GUIDE, providing real-time visualization of the end-effector's position and trajectories within a simulated 3D workspace. The results obtained were compared with those from the RoboAnalyzer software, demonstrating excellent agreement in both workspace definition and time-based trajectory planning. A comprehensive experimental study, based on 27 scenarios testing different configurations of the first three joints, was conducted to evaluate the influence of these parameters on the Cartesian position of the end-effector, thereby confirming the robustness of the model. Additionally, an extensive workspace simulation, relying on 250,000 randomly generated joint configurations, validated the predicted spatial coverage in accordance with RoboAnalyzer outcomes. Point-to-point trajectory planning was also successfully implemented, producing smooth and accurate motion of the end-effector from an initial to a target position through interpolation of 451 points, thus ensuring a continuous and controlled trajectory.
These results demonstrate that the analytical approach provides a reliable framework for modeling and simulating SCARA manipulators. Our future work includes incorporating a Back Propagation Artificial Neural Network method with hyper-parameters optimized by Design of Experiments. This model provides an improved prediction of the dynamic response of serial robots, applicable in an industrial environment. It can be considered as an alternative method of trajectory design by using the Jacobian inverse for a real-time calculation of the joint positions when analytic solutions are either too complicated or resource consuming.

*References:*

[1] Beni, A. (2011) Kinematics of Adept Three Robot Arm, in Robot Arms, InTech, ISBN: 978-953-307-160-2, https://doi.org/10.5772/17732

[2] Gouasmi, M., Ouali, M., Fernini, B., Meghatria, M. (2012) Kinematic Modelling and Simulation of a 2-R Robot Using SolidWorks and Verification by MATLAB/Simulink, International Journal of Advanced Robotic Systems, Vol. 9, No. 6, pp. 1–13, https://doi.org/10.5772/50203

[3] Alshamasin, M. S., Ionescu, F., Al-Kasasbeh, R. T. (2012) Modelling and simulation of a SCARA robot using solid dynamics and verification by MATLAB/Simulink, International Journal of Modelling, Identification and Control, Vol. 15, No. 1, pp. 28–38. Available: https://www.inderscience.com/info/inarticle.php?artid=43938 [Access Date : 01/01/2026]

[4] Szabó, S. I., Rónai, L. (2023) Readjustment of a Robot for Educational Purposes, International Review of Applied Sciences and Engineering, Vol. 14, No. 2, pp. 212–219, https://doi.org/10.1556/1848.2022.00506

[5] Bussola, R., Legnani, G., Callegari, M., Palmieri, G., Palpacelli, M.-C. (2019) Simulation Assessment of the Performance of a Redundant SCARA, Robotics, Vol. 8, No. 2, Article 45, https://doi.org/10.3390/robotics8020045

[6] Shruti, M., Chittawadigi, R. G. (2021) Offline Simulation of Motion Planning of a Planar Manipulator in RoboAnalyzer and its Integration with a Physical Prototype, in 2021 IEEE 9th Region 10 Humanitarian Technology Conference (R10-HTC), Bangalore, India, pp. 1–6, https://doi.org/10.1109/R10-HTC53172.2021.9641544

[7] Mohammed, Y. I., & Hussein, S. M. (2015). Modeling and Simulation of Industrial SCARA Robot Arm. International Journal of Engineering and Advanced Technology (IJEAT), Vol. 4, No. 4, pp. 220–229. Available at: https://www.ijeat.org/portfolio-item/d3944044415/ [Access Date : 01/01/2026]

[8] Jasim, W. M. (2011) Solution of Inverse Kinematics for SCARA Manipulator Using Adaptive Neuro-Fuzzy Network, International Journal on Soft Computing, Vol. 2, No. 4, pp. 59–66, https://doi.org/10.5121/ijsc.2011.2406

[9] Yasar, S. A., Korkut, I. (2016) Design And Kinematic Analysis Of A Rrpr Robot Arm, International Journal of Innovative Research in Engineering & Management, Vol. 3, No. 6, pp. 490–493, https://doi.org/10.21276/ijirem.2016.3.6.7

[10] Wilamowski, B. M., Irwin, J. D. (2011) Intelligent systems (Part I: Chapter 1), in Intelligent Control Systems (Neural Networks), CRC Press, New York, USA, Vol. 1, pp. 1–12, https://doi.org/10.1201/9781315218427

[11] Hasan, A. T., Ismail, N., Hamouda, A. M. S., Aris, I., Marhaban, M. H., Al-Assadi, H. M. A. A. (2010) Artificial neural network-based kinematics Jacobian solution for serial manipulator passing through singular configurations, Advances in Engineering Software, Vol. 41, No. 2, pp. 359–367, https://doi.org/10.1016/j.advengsoft.2009.06.006

[12] Köker, R. (2005) Reliability-based approach to the inverse kinematics solution of robots using Elman's networks, Engineering Applications of Artificial Intelligence, Vol. 18, No. 6, pp. 685–693, https://doi.org/10.1016/j.engappai.2005.01.004

[13] Jha, P., Biswal, B. B. (2014) A neural network approach for inverse kinematic of a SCARA manipulator, International Journal of Robotics and Automation, Vol. 3, No. 1, pp. 52–61. Available: https://ijra.iaescore.com/index.php/IJRA/article/view/933 [Access Date : 01/01/2026]

[14] Huang, H., Arogbonlo, A., Yu, S., Kwek, L. C., & Lim, C. P. (2025). Trajectory tracking of a SCARA robot using intelligent active force control. Neural Computing and Applications, Vol. 37, No. 19, pp. 13345–13370. https://doi.org/10.1007/s00521-025-11200-x

[15] Kumar, V., & Mistri, A. (2025). Fuzzy logic-based synchronization of trajectory planning and obstacle avoidance for RRP SCARA robot. International Journal on Interactive Design and Manufacturing, Vol. 19, pp. 3039–3052. https://doi.org/10.1007/s12008-024-02214-1

[16] Bouzid, R., Gritli, H., Narayan, J. (2024) ANN approach for SCARA robot inverse kinematics solutions with diverse datasets and optimisers, Applied Computer Systems, Vol. 29, No. 1, pp. 24–34, https://doi.org/10.2478/acss-2024-0004

[17] Denavit, J., Hartenberg, R. (1955) A kinematic notation for lower-pair mechanisms based on matrices, Journal of Applied Mechanics, Vol. 22, pp. 215–221, https://doi.org/10.1115/1.4011045

[18] Jazar, R. N. (2010) Theory of Applied Robotics, Springer US, Boston, MA, ISBN: 978-1-4419-1750-8, https://doi.org/10.1007/978-1-4419-1750-8

[19] Montgomery, D. C. (2012) Design and Analysis of Experiments, Wiley, New York, NY, ISBN: 978-1118146927, https://archive.org/details/designanalysisof0000mont [Access Date : 01/01/2026]

[20] Corke, P. I. (2017) Robotics, Vision and Control: Fundamental Algorithms in MATLAB®, Springer, Cham, ISBN: 978-3-319-54413-7, https://doi.org/10.1007/978-3-319-54413-7

[21] RoboAnalyzer. RoboAnalyzer: A 3D Model Based Robotics Simulation Software. Available at: http://www.roboanalyzer.com/ [Access Date : 01/01/2026].

[22] AM101-svg. Case_Study_Serial_Robot-1-: Serial Robot Case Study Repository. Available at: https://github.com/AM101-svg/Case_Study_Serial_Robot-1- [Access Date : 01/01/2026]

**Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)**

Abdelilah El Merini: Contributed to the theoretical development and practical implementation of the modeling framework, the simulation setup and experimental validation as an applied case study of the 4-DOF serial robot manipulator.

Chaimae Ouchicha: Provided scientific and methodological supervision of the research, ensuring rigor in the analytical approach, experimental design, and interpretation of results.

Ouafae Hamdoun: Contributed to the development and formal elaboration of the theoretical foundations, including the derivation and verification of the kinematic equations and the mathematical modeling of the RRPR manipulator configuration.

**Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself**

No funding was received for conducting this study.

**Conflict of Interest**

The authors have no conflicts of interest to declare that are relevant to the content of this article.

# APPENDIX

Table 1. Experimental main result

*Source: created by the authors*

| **Standard order test** | | | **Responses** | | |
|---|---|---|---|---|---|
| | | | **Px** | **Py** | **Pz** |
| 1 | 1 | 1 | -176.3 | -75.55 | 250 |
| 1 | 1 | 2 | -176.3 | -75.55 | 175 |
| 1 | 1 | 3 | -176.3 | -75.55 | 100 |
| 1 | 2 | 1 | -200 | -346.4 | 250 |
| 1 | 2 | 2 | -200 | -346.4 | 175 |
| 1 | 2 | 3 | -200 | -346.4 | 100 |
| 1 | 3 | 1 | 22.72 | -190.5 | 250 |
| 1 | 3 | 2 | 22.72 | -190.5 | 175 |
| 1 | 3 | 3 | 22.72 | -190.5 | 100 |
| 2 | 1 | 1 | 153.6 | -114.9 | 250 |
| 2 | 1 | 2 | 153.6 | -114.9 | 175 |
| 2 | 1 | 3 | 153.6 | -114.9 | 100 |
| 2 | 2 | 1 | 400 | 0 | 250 |
| 2 | 2 | 2 | 400 | 0 | 175 |
| 2 | 2 | 3 | 400 | 0 | 100 |
| 2 | 3 | 1 | 153.6 | 114.9 | 250 |
| 2 | 3 | 2 | 153.6 | 114.9 | 175 |
| 2 | 3 | 3 | 153.6 | 114.9 | 100 |
| 3 | 1 | 1 | 22.72 | 190.5 | 250 |
| 3 | 1 | 2 | 22.72 | 190.5 | 175 |
| 3 | 1 | 3 | 22.72 | 190.5 | 100 |
| 3 | 2 | 1 | -200 | 346.4 | 250 |
| 3 | 2 | 2 | -200 | 346.4 | 175 |
| 3 | 2 | 3 | -200 | 346.4 | 100 |
| 3 | 3 | 1 | -176.3 | 75.55 | 250 |
| 3 | 3 | 2 | -176.3 | 75.55 | 175 |
| 3 | 3 | 3 | -176.3 | 75.55 | 100 |