



DOI: 10.71762/rsjj-am29

Research Paper

Simulation, Analysis, and Performance Evaluation of a Three-degree-of-freedom SCARA Robot for Industrial Assembly Applications

Amir Hossein Badami¹, Mohamad Shahgholi^{1*}

¹Department of Mechanical Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran

*Email of the Corresponding Author: mohamad.shahgholi@iau.ac.ir

Received: December 2, 2024; Accepted: February 27, 2025

Abstract

The prevalence of robots in contemporary society is steadily increasing, significantly impacting various aspects of daily life and industry. Robots have become crucial components in flexible manufacturing systems due to their adaptability and efficiency in performing repetitive tasks. Human-robot collaboration has garnered widespread interest and is being extensively applied across both domestic and industrial sectors, with robots serving as reliable partners to enhance productivity and safety. Among these robotic systems, the SCARA robot is particularly valued for its high-speed and precision performance, especially in assembly lines. This paper provides a comprehensive analysis and simulation of SCARA robots with two and three degrees of freedom tailored for production line applications. Initially, the Denavit-Hartenberg (D-H) parameters are identified to structure the robot's configuration, followed by the derivation of its forward kinematic equations. Furthermore, the dynamic equations of the robot are formulated based on these mathematical models to understand its movement and response. Ultimately, the performance of the SCARA robot is analyzed and simulated using MATLAB software in the Simulink environment.

Keywords

Scara Robot, Torque, Denavit Hartenberg Parameters, Simulink

1. Introduction

Robotic science primarily impacts the industry, which is evident in the reduced prices of products resulting from automation [1]. Robots are typically employed in tasks where they outperform humans or operate in hazardous environments, such as volcanic explorations, thus safeguarding human lives [2]. The Scara robot, developed in 1981 under Professor Hiroshi Makino at Yamanashi University, exemplifies such innovation. Scara, an acronym for Selective Compliance Assembly Robot Arm, is designed for flexibility and ease in assembly operations [3-4].

A robotic arm, generally programmable, mimics the functionality of a human arm [5-6]. It can either be an autonomous robot or part of a larger robotic system. The joints of a robotic arm enable rotational or translational movement, creating a chain of motion [7-8]. Scara robots are known for their cost-effectiveness and versatility, optimized for applications in transport and assembly due to their design principles [9-10].

The Scara robot typically has three rotational degrees of freedom, although variants with different degrees of freedom exist [11-15]. The three-degree-of-freedom Scara robot is preferred due to its high maneuverability in the xy-plane and the ability to position along the z-axis, making it ideal for industrial assembly operations. Its speed, rigidity, and durability make Scara robots a favored choice among manufacturers [16-17].

In a study, the workspace of a 3DOF SCARA robot using the Denavit-Hartenberg method was analyzed. They examined how design parameters, including link lengths and joint configurations, influence workspace efficiency. Their findings indicated notable improvements in the robot's flexibility and coverage, which enhance its applicability in industrial environments [18]. In another study, a hybrid intelligent control system for a SCARA robot was proposed to integrate virtual reality modeling to enhance control precision and adaptability. Their findings demonstrated the effectiveness of the approach in improving the robot's performance in complex industrial tasks, particularly in assembly line applications [19]. Also, a multibody dynamic model for a SCARA manipulator tailored for in-space applications was developed in another study [20]. The study emphasized the manipulator's suitability for precision assembly tasks in microgravity environments. Their findings underline the importance of advanced modeling techniques for optimizing robotic performance in unique operational conditions [20].

This paper presents a simulation of a three-degree-of-freedom Scara robot using Matlab and Simulink. Initially, the Denavit-Hartenberg parameters are defined, followed by the extraction of the robot's direct and inverse kinematic equations. The results demonstrate the robot's suitability for industrial applications.

2. Material and Method

The chosen robot in this article is the Scara robot, which has three degrees of freedom. This robot is the same as the RRP robot, which has one prismatic joint and two rotary joints, and there are three links between the joints. Figure 1 shows a screen of this robot. According to this figure, the length of the first link is h , the second is d_1 , and the third is d_2 . θ_1 is the rotation value of the first rotary joint, θ_2 is for the second torsional joint and d is the displacement created by the prismatic joint.

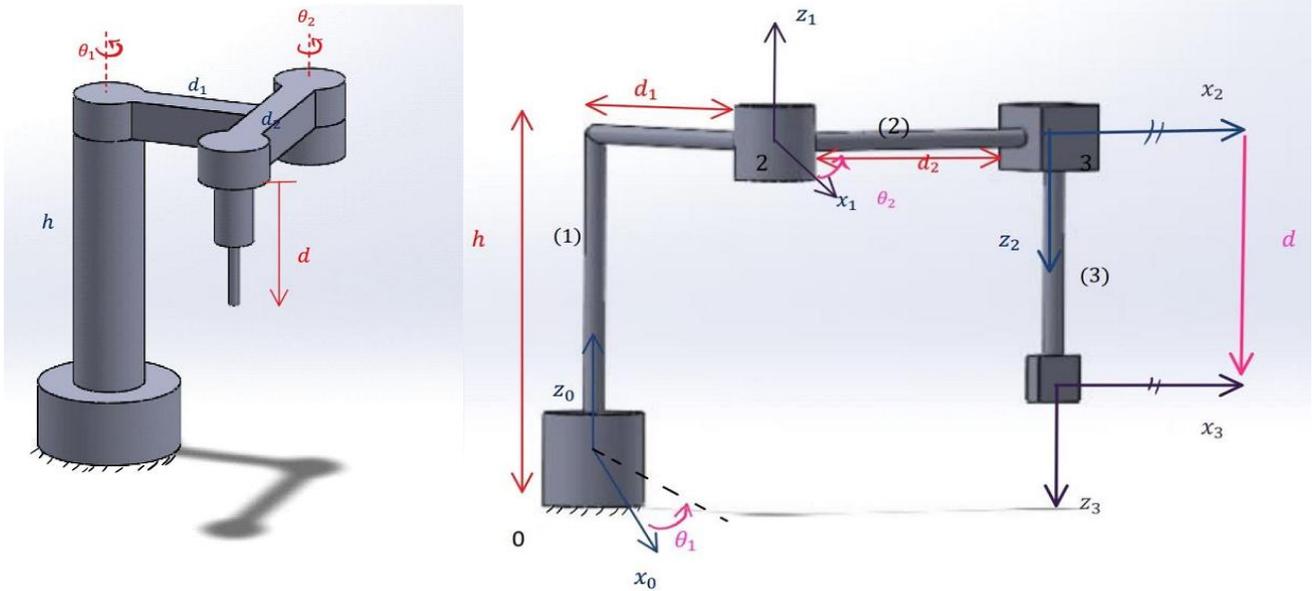


Figure 1. Scara robot with three degrees of freedom and its parameters (left-sided), Defining links and joints in the Scara robot with three degrees of freedom (right-sided)

2.1 Direct kinematic analysis

According to Figure 1, first, the links and joints, which are marked with black color are named. The fixed axis $x_0 z_0$ is connected to link 0 or ground. Also, the z -axis of link 1 is placed along the positive direction of rotation of joint 2 as well as the x -axis along link 1. The z -axis of link 2 is placed along the prismatic joint 3 and its direction is opposite to the z -axis of the ground coordinates. The x -axis of this link is considered along it. For the final operator, the z -axis is taken along link 3 and in the direct direction of coordinate 2 in the z -axis, and also x is as coordinates of the link 2.

According to Figure 1 and the explanations given in the placement of coordinates, Table 1 shows the parameters of Denavit-Hartenberg, where θ is the angle of rotation around the z -axis until the alignment of the x -axis, d is the distance between the x -axis of a coordinate and the next coordinate, l is the distance between the z -axis of one coordinate and the next coordinate, and α is the rotation around the x -axis until the z -axis aligns between the two coordinates.

Table 1: Danavit Hartenberg parameters

	θ	d	l	α
1	θ_1	h	d_1	0
2	θ_2	0	d_2	π
3	0	d	0	0

Unknown values are θ_1 , θ_2 , and d . Therefore, the coordinate transformation matrix of the final operator to the origin is equal to [21-22]:

$$T_3^0 = R_{z,\theta_1} T_{z,h} T_{x,d_1} R_{z,\theta_2} T_{x,d_2} R_{x,\pi} T_{z,d} \quad (1)$$

Where $R_{z,\theta}$ and $R_{x,\alpha}$ are defined as follows:

$$R_{z,\theta} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_{z,d} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$R_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_{x,l} = \begin{bmatrix} 1 & 0 & 0 & l \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Finally, Equation 1 is converted to Equation 3 using the put the top values in Equation 1

$$\begin{bmatrix} \cos(\theta_1 + \theta_2) & \sin(\theta_1 + \theta_2) & 0 & d_2 \cos(\theta_1 + \theta_2) + d_1 \cos \theta_1 \\ \sin(\theta_1 + \theta_2) & -\cos(\theta_1 + \theta_2) & 0 & d_2 \sin(\theta_1 + \theta_2) + d_1 \sin \theta_1 \\ 0 & 0 & -1 & h - d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

2.2 Inverse kinematic analysis

For inverse kinematics, the goal is to be able to obtain θ_1 , θ_2 , and d by having x , y , and z (Figure 2). The values of h , d_1 , and d_2 are considered as before to simplify the calculation process. The point P is considered on the final operator and it is assumed that $P_0^0 =$. So according to the x_0y_0 axis [23-24].

$$l^2 = X^2 + Y^2 \quad (5)$$

$$l^2 = 30^2 + 25^2 + 2 * 25 * 30 * \cos \theta_2$$

$$\rightarrow \theta_2 = \cos^{-1} \frac{X^2 + Y^2 - 30^2 - 25^2}{2 * 25 * 30}$$

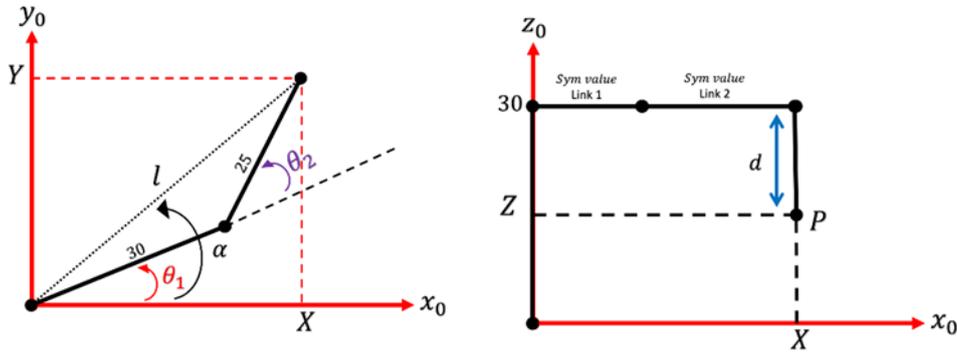


Figure 2. Inverse kinematics analysis for Scara robot with three degrees of freedom (left-sided). Obtaining d with the help of inverse kinematics analysis for the Scara robot with three degrees of freedom (right-sided)

There are 2 solutions for θ_2 . Accordingly, it is obvious:

$$\sin \alpha = \frac{Y}{l}, \cos \alpha = \frac{X}{l} \rightarrow \alpha$$

$$\sin \alpha - \theta_1 = \frac{25 \sin \theta_2}{l}, \cos \alpha - \theta_1 = \frac{30 + 25 \cos \theta_2}{l}$$

$$\rightarrow \alpha - \theta_1 = \text{atan2} \left(\frac{25 \sin \theta_2}{l}, \frac{30 + 25 \cos \theta_2}{l} \right) \quad (6)$$

So θ_1 is obtained according to θ_2 . According to x_0z_0 :

$$Z = 30 - d \rightarrow d = 30 - z \quad (7)$$

In this way, d is also obtained.

2.3 Inverse kinematic analysis

The purpose of this part is that the movement speed of each joint and location of the robot to the kinematics is considered and finally, the movement and rotation speed of the final operator or vice versa is determined. For this, first, the Jacobian matrix of the robot is obtained [25-27].

In the previous parts, T_3^0 is obtained by using a direct kinematics solution. According to that, the displacement of the final operator relative to the ground coordinates is obtained from the first three rows of the last column:

In the previous parts, T_3^0 is obtained by using a direct kinematics solution. According to that, the displacement of the final operator relative to the ground coordinates is obtained from the first three rows of the last column:

$$o_0o_3 = \begin{bmatrix} d_2 \cos(\theta_1 + \theta_2) + d_1 \cos \theta_1 \\ d_2 \sin(\theta_1 + \theta_2) + d_1 \sin \theta_1 \\ h - d \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (8)$$

According to the previous values of d_1 , d_2 , and h , the speed Jacobian is calculated as follows:

$$J_v = \begin{bmatrix} -30 \sin \theta_1 - 25 \sin(\theta_1 + \theta_2) & -25 \sin(\theta_1 + \theta_2) & 0 \\ 30 \cos \theta_1 + 25 \cos(\theta_1 + \theta_2) & 25 \cos(\theta_1 + \theta_2) & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (9)$$

And finally, the V_e is defined in the following equation:

$$V_e = J_v \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d} \end{bmatrix} \quad (10)$$

The speed of the final operator relative to the ground coordinates is obtained for the case where the movement speed of the joints and the location of the robot.

By considering the speed of the final operator and also the speed of movement of the joints in the desired location, the following equation is obtained:

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d} \end{bmatrix} = J_v^{-1} V_e \quad (11)$$

Which has an answer if the determinant of velocity Jacobian is not zero at a particular location and there is an inverse matrix, which is called a singular point. The periodic Jacobian is also obtained from the following relation:

$$J_w = [\alpha_1 k_0 \quad \alpha_2 k_1 \quad \alpha_3 k_2] \quad (12)$$

Where $\alpha_1=\alpha_2=1$ because they are rotating and $\alpha_3=0$ because it is a prismatic joint. Also $k_0 = k_1 = k_2 = k_3 = 1$ according to the direct kinematics of the robot. Accordingly, the rotation angle of the robot is defined as:

$$\omega_e = J_w \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d} \end{bmatrix} \quad (13)$$

3. Results and discussion

3.1 Verification of direct kinematic equations

For validation, the simulation in Simulink MATLAB and Simscape is performed. First, the links in another software such as Solidworks are designed, and then, by placing them in the model, the simulation is done [28]. For simulation, h and d_1 are considered with 30 cm and d_2 as much as 25 cm. In the Simscape section of Simulink, first, in the section of the mechanism configuration, the gravity value in the z-axis direction of the ground coordinates is set to zero so that the moments of inertia do not hurt the robot's performance.

Then, the torsional and prismatic joints are placed between the coordinate axes of Figure 1 and links 1 to 3 are connected in certain places. The goal of this paper is to take an arbitrary initial value inside each joint and use that to report the position of the final operator. For this reason, a Transform Sensor is connected on coordinate 3, and using PS-Simulink Converter, x, y, and z values are displayed relative to ground coordinates.

Figure 3 shows the Simulink block diagram and Figure 4 shows the design of the robot. Table 2 also shows the simulation results for the arbitrary values of the joints, which are verified by placing the first three rows of the right column in the relation T_3^0 , and the values are correct.

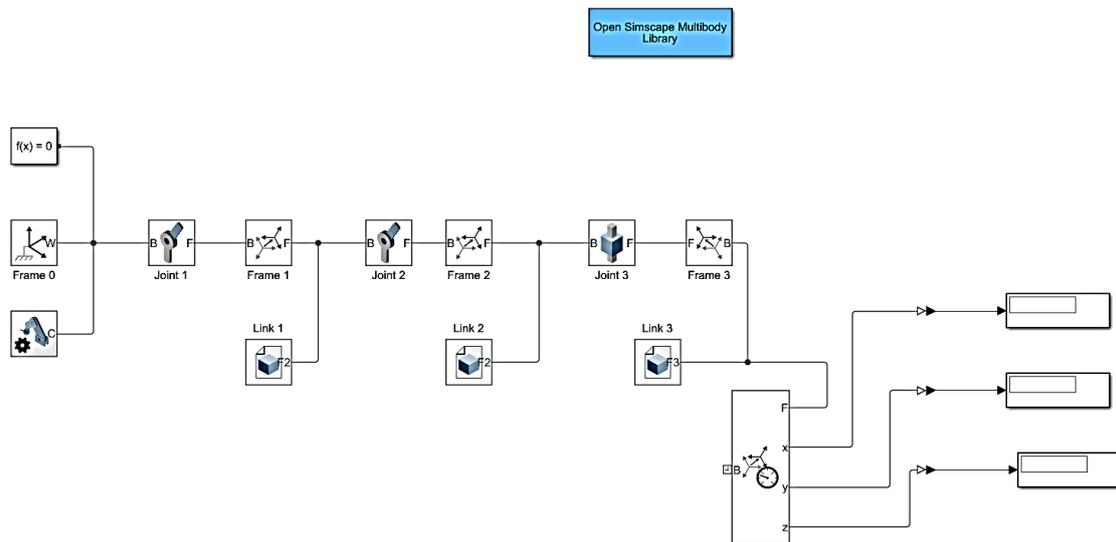


Figure 3. Simulink block diagram for Scara robot of three degrees of freedom

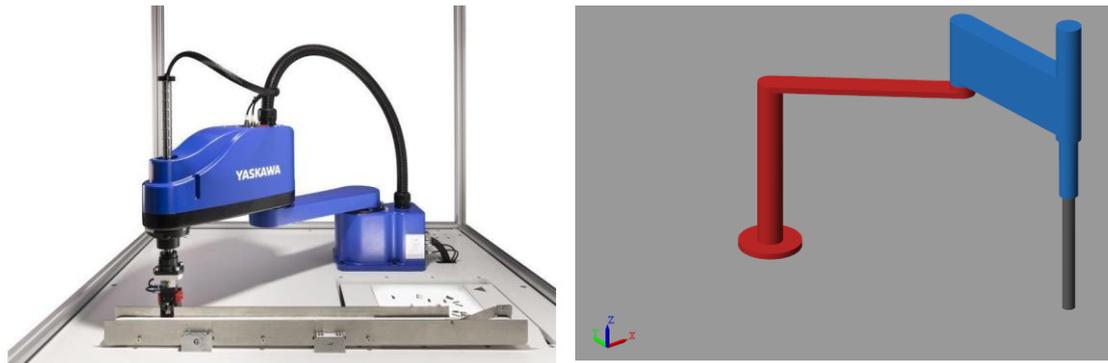


Figure 4. An industrial Scara robot (Left side) and view of the designed Scara robot with three degrees of freedom (right side)

Table 2. Simulated results for arbitrary values of joints

θ_1		45	180	320
θ_2		45	-90	-45
d		13	30	20
<i>Loc.</i>	x	21.21	-30	25.16
	y	46.21	25	-44.19
	z	17	0	10

3.2 Verification of inverse kinematics

First, arbitrary angles (different from direct kinematics) are considered, and using Simulink presented in Figure 3, the final position is obtained as shown in Table 3.

Table 3. Simulated results for arbitrary values of joints with the help of direct kinematics

θ_1		35	-40	90
θ_2		40	15	-90
d		5	10	-5
<i>Loc.</i>	x	31.05	45.64	25
	y	41.36	-29.85	30
	z	25	20	35

Now, these final values of P are put in the inverse kinematics function and the degrees of freedom are gained:

Table 4. Simulated results for arbitrary values of joints with the help of indirect kinematics

<i>Loc.</i>	x	31.05	45.64	25
	y	41.36	-29.85	30
	z	25	20	35
θ_1		35.02	-26.38	10.39
θ_2		39.96	-14.98	90
d		5	10	-5

The reason for the small difference in the degrees of freedom is the trigonometric functions in the function and also the fact that the smallest angle of the first joint is considered the answer. According to these results, inverse kinematics works well.

3.3 Jacobin validation and speed

For the simplicity of the work, a command is written in the MATLAB software that calculates the Jacobian according to the position of the Scara robot and gives the final speed of the operator according to the movement speed of the joints or vice versa [28-29]. For validation, $\theta_1=35$ degrees, $\theta_2=40$ degrees, and $d=5$ cm. The speed of these joints should be 3 radians/second, 2 radians/second, and 1 cm/second, respectively. It should be noted that for the software using deg2rad, first the angles in radians are calculated and then entered. Therefore, the speed of the final operator is equal to:

$$V = \begin{bmatrix} -172.36 \text{ cm/s} \\ 106.07 \text{ cm/s} \\ -1 \text{ cm/s} \end{bmatrix}, W = \begin{bmatrix} 0 \\ 0 \\ 5 \text{ rad/s} \end{bmatrix} \quad (14)$$

If the inverse of this equation is considered, as a result

$$\dot{Q} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d} \end{bmatrix} = \begin{bmatrix} 3 \text{ rad/s} \\ 2 \text{ rad/s} \\ 1 \text{ cm/s} \end{bmatrix} \quad (15)$$

Which gives the correctness of the written code.

3.4 Design of path for Robot motion

To design the path of the robot, an example is considered. It is imagined that the location of the robot's final operator changes from point (40,5,15) to point (10,38,25-) within one second and then stands there for two seconds and finally reaches its starting place in one second. Therefore, the route design has three parts: start, stop, and return. Here route design is performed in two modes.

a) Unrestricted route

Limitation means not having a limit for the movement of motors on the joints, and it is assumed that there is no limitation in this part. Since there is no limitation, a polynomial is made for the movement of each joint, considering the part where the movement happens in one second. Because it is assumed that the acceleration derivative is continuous along the path (not at the beginning and end), six boundary conditions are, so the function is of the 5th order [30].

$$\theta_{(t)} = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (16)$$

The Scara robot has torsional freedom of two degrees and prismatic freedom of 1 degree, and for each of these degrees ($\theta_{1(t)}$, $\theta_{2(t)}$, $d_{(t)}$), coefficients should be calculated and some terms made. Because the outputs of the function are in degrees and centimeters, path functions are also of the same kind. These values should be checked in three intervals (a time between 0 and 1), (b time between 1 and 3

stationary), and (c time between 3 and 4), and functions should be obtained by calculating the coefficients for each joint in each interval [30].

$$\theta_{(0)} = \theta_S, \theta_{(1)} = \theta_E, \dot{\theta}_{(0)} = 0, \dot{\theta}_{(1)} = 0, \ddot{\theta}_{(0)} = 0, \ddot{\theta}_{(1)} = 0 \quad (17)$$

The values of θ_s and θ_E for each degree of freedom are obtained with the inverse kinematics function. Based on these boundary conditions, the polynomial coefficients are all obtained, and as a result, the displacement of the variables relative to the origin in terms of t is given in three relationships for the interval a.

For an interval of b, from where there is no movement and there are 2 seconds of stop, the values of the functions are the same as the final values of θ_E obtained for each of them, and these three relations are also obtained simply.

For an interval of c with the rules of transfer of functions, since the reverse movement is the movement in an interval of a, by changing t inside the function to $-t+4$, the three relations of this interval for variables are easily obtained and there is no need to obtain relations from the new boundary conditions and their changes.

After obtaining all the relationships, the speed relationship of each variable and then the acceleration of the variable in each interval are obtained by deriving them.

The reason for this is that to apply these functions in the Simulink environment, numerical calculations are required and they are placed in the form of MATLAB functions, and having these relationships greatly increases the accuracy of the work. All these calculations and their results are in the Trajectory_a.m file, and when running, the relationship of each variable is displayed for each interval, this file is used only for this purpose and it is not used directly in the Simulink file. The Scara model file with these path functions is placed as Scara_a.slx and its diagrams and analysis are given in the following. In this file, Trajectory_1 takes time as input and obtains the movement, speed, and acceleration for each joint as output.

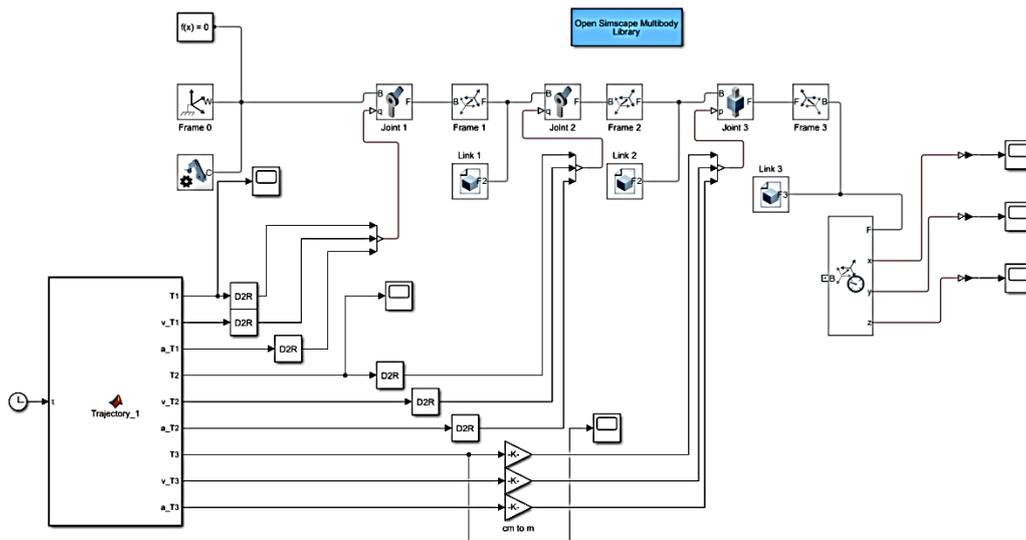


Figure 5. Simulink block diagram in Scara robot with three degrees of freedom for Trajectory_1

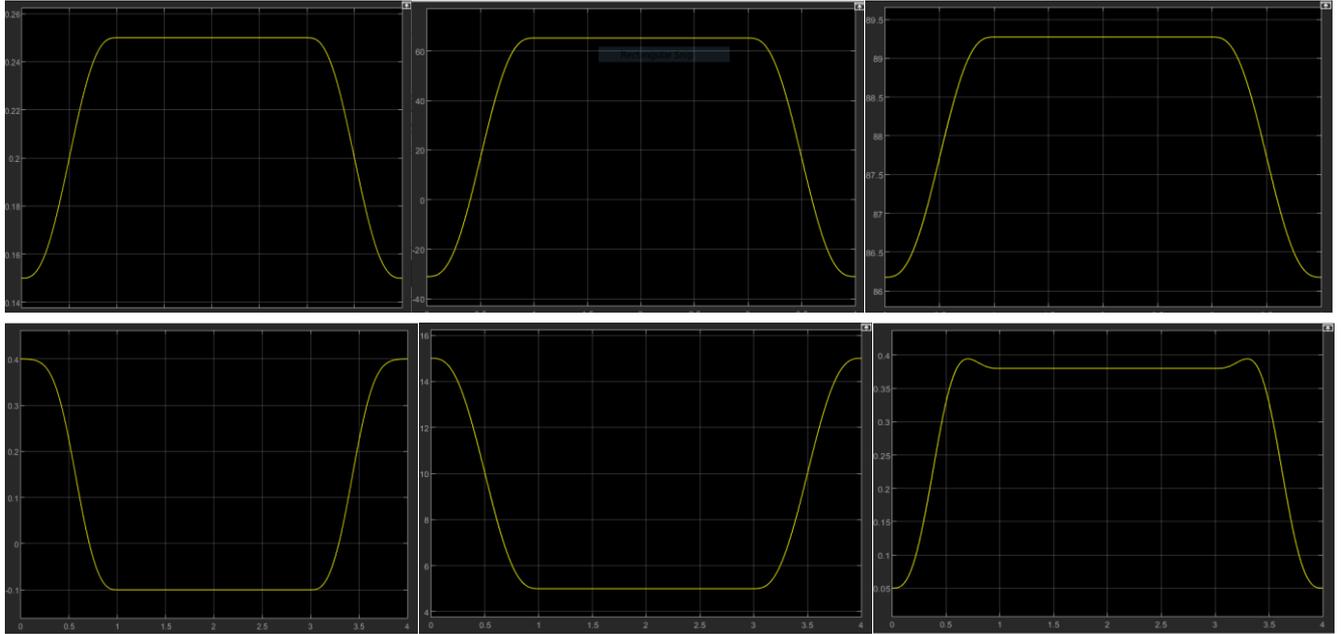


Figure 6. Changes of θ_1 , θ_2 , and d during movement from right to left, respectively, without limitation in speed and acceleration (Top). The changes of x , y , and z relative to the origin of the ground coordinates from right to left, respectively, without limitation in speed and acceleration (Bottom)

The robot movement animation is displayed by running the Simulink file. Figure 6 shows the changes of θ_1 , θ_2 , and d from right to left during movement. Figures 6 also show the changes of x , y , and z about the origin of the earth's coordinates.

b) Movement with limitations in speed and acceleration

According to the mentioned example, the movement in a straight line is considered with the speed and acceleration limits of $v_{max}=0.3$ m/s and $a_{max}=\pm g/2$ m/s², where g is approximately 9.8 and S_{max} is the distance between the initial location and the joint and its end location. In this case, the operator stops at the destination for two seconds and then returns to the first place. According to Figure 7 and the trapezoidal profile, 8 ranges for speed are defined.

In the solution path, first, the change times of the slope for the velocity graph are obtained using the LSPB relations. The values of t_a , t_d , and t_f for the state (a b c) are easily obtained with the following relations [30]:

$$t_a = \frac{v_{max}}{a_{max}}, t_d = \frac{S_{max}}{v_{max}}, t_f = t_d + t_a \quad (18)$$

These intervals, for example, are the end of the interval of a, the end of the interval of b, and the end of the interval of c, respectively. Next, the zero interval is two seconds and the speed is zero, and for d, e, and f the constants of the return time are also obtained by changing the variables.

In the next step, it is necessary to obtain the distance of S in each part for each of the back, zero, and fourth intervals. Based on this, first, for the back interval, the movement paths of S in three intervals a, b, and c are created with LSPB relations [31]:

$$a: s(t) = \frac{1}{2} a_{max} t^2, \dot{s}(t) = a_{max} t, \ddot{s}(t) = a_{max} \quad (19)$$

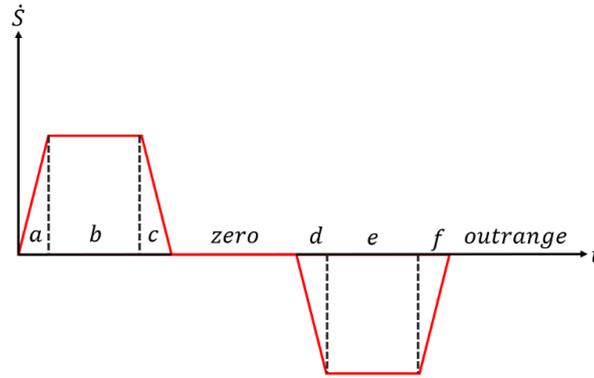


Figure 7. Trapezoidal profile for speed that includes 8 steps

$$b: s(t) = v_{max} \left(t - \frac{1}{2} t_a \right), \dot{s}(t) = v_{max}, \ddot{s}(t) = 0 \quad (20)$$

$$c: s(t) = v_{max} (t - t_d) - \frac{1}{2} a_{max} (t^2 - t_d^2) + a_{max} t_d (t - t_d) + v_{max} \left(t_d - \frac{1}{2} t_a \right), \dot{s}(t) = v_{max} - a_{max} (t - t_d), \ddot{s}(t) = -a_{max}$$

So, signals of motion, speed, and acceleration can be designed by placing the value of constant time. In the zero interval, the value of S is the same as the path length or S_max. This value is the distance between the coordinates of the start and end points, considering the linearity of the movement and its linear scanning. For the return interval, due to the constant slope of the trapezoidal edges, S is obtained by subtracting S from S_max of the same interval in the back route. Of course, it should be noted that in this case, for the same S, the t value has been changed to t-tf-2. In the outrange mode, the value of the derivative of S is set to zero. After obtaining S according to the input time, this value is superimposed on the vector of the start and end distance, and X(s) is formed [31].

$$X(S) = X_1 + \frac{X_2 - X_1}{S_{max}} S \quad (21)$$

So by placing S in it, the coordinates of the location of the operator in time are gained. Now, with the function of the inverse kinematics (Inv_Kin_Scarax.m), the displacement of each joint relative to its axis is obtained. The difference between this function of the inverse kinematics for Scara and the previous function is that because this function is supposed to be in the Simulink environment, it needs to be changed. By applying these places to the joints, it has been seen the change of speed and acceleration at every moment results in the proper performance of the model in its tasks.

The Trajectory_b.m file is placed only to obtain the back time constants and the model file whose input is written in the MATLAB function of Simulink. The obtained results are given in Figure 8. According to the form of the graphs, there is an error value during the movement in the graphs.

The reason for this is that the value of step is more than t_a , which cannot be seen as the maximum value of the acceleration by drawing the graph of the second derivative S .

As the step increases, this value becomes more realistic, but in the end, this step is not practical for the functions, and a higher step must be taken so that the system has its function until the end.

By looking at the S -speed diagram, the shape of the trapezoids is almost correct, and with the correct initial and final location, a good performance can be seen. Nonetheless, by reducing the step, the movement becomes smoother, but in the end, the device is placed in the final position with a small amount of error, which can be ignored.

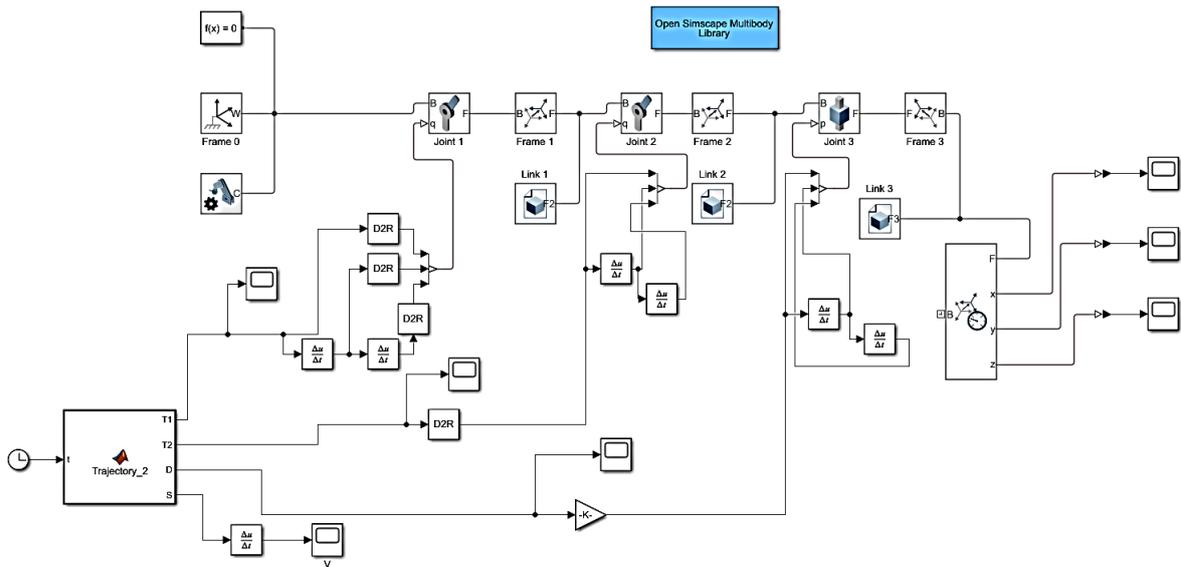


Figure 8. Simulink block diagram for Scara robot with three degrees of freedom for Trajectory_2

The Trajectory_2 block actually receives the time and after calculating S at that moment, performs scanning and then outputs the degrees of freedom with inverse kinematics. Finally, these degrees become acceleration and speed. Figure 9 shows the changes of θ_1 , θ_2 , and d respectively, and also shows the changes of x , y , and z during the movement.

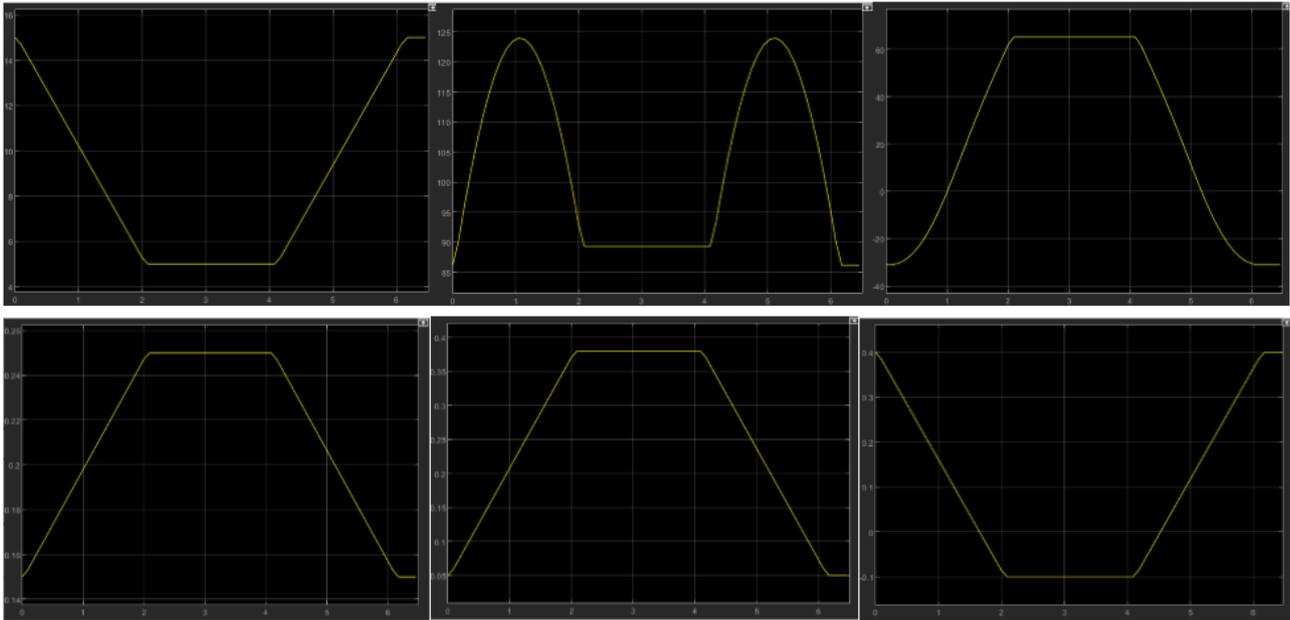


Figure 9: Changes of θ_1 , θ_2 , and d during movement from right to left, respectively, with limitations in speed and acceleration (top). The changes of x , y , and z relative to the origin of the ground coordinates from right to left, respectively, with limitations in speed and acceleration (bottom)

4. Conclusions

In this paper, the equations of motion in the Scara robot by using the kinematic equations of robots were obtained. For this purpose, first, the Denavit-Hartenberg parameters were determined and then the direct kinematic equations of the robot were extracted. In the following, the dynamic equations of the robot were extracted and according to these equations, the analysis and simulation of the Scara robot were performed in the Simulink environment of MATLAB software. The basic issue in the design of any control system is to ensure the operation of the closed circuit system according to certain technical specifications the most important of these criteria is the stability of the system, which can be seen in the simulated robot that it can be used for industrial applications according to the results obtained.

5. References

- [1] Li, K., Huo, Y., Liu, Y., Shi, Y., He, Z. and Cui, Y. 2022. Design of a lightweight robotic arm for kiwifruit pollination. *Computers and Electronics in Agriculture*. 198: 107114. doi:10.1016/j.compag.2022.107114.
- [2] Huang, R.Y., Cheng, C.W., Tsai, M.C. and Lee, A.C., 2022. Acceleration-based FIR filters for trajectory interpolation with different kinematic constraints and target velocities for NC systems. *Control Engineering Practice*. 124: 105204. doi:10.1016/j.conengprac.2022.105204.
- [3] Mahdieh, M.S. 2024. Investigation of Steel Slabs Marker Robotic Arm Control Parameters in Noise and Disturbance Absence/Presence. *Journal of Modern Processes in Manufacturing and Production*. 13(2): 85 – 96. doi: 10.71762/2967-8504.
- [4] Smith, A., Contreras, G.A.T., Brisbois, M.C., Lacey-Barnacle, M. and Sovacool, B.K. 2023. Inclusive innovation in just transitions: the case of smart local energy systems in the UK. *Environmental Innovation and Societal Transitions*. 47:100719. doi:10.1016/j.eist.2023.100719.

- [5] Goel, R. and Gupta, P. 2020. Robotics and industry 4.0. A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development. Springer.
- [6] Abbasi-Kesbi, R., Valipour, A. and Imani, K. 2018. Cardiorespiratory system monitoring using a developed acoustic sensor. *Healthcare Technology Letters*. 5(1):7-12. doi:10.1049/htl.2017.0012.
- [7] Foley, F., O'Donoghue, C. and Walsh, J., 2023. Optimizing the Implementation of a Robotic Welding System. *Journal of Modern Processes in Manufacturing and Production*. 12(2):39-52. doi: 20.1001.1.27170314.2023.12.2.3.8.
- [8] Abbasi-Kesbi, R. and Nikfarjam, A., 2017. Denoising MEMS accelerometer sensors based on L2-norm total variation algorithm. *Electronics Letters*. 53(5):322-324. doi:10.1049/el.2016.3811.
- [9] Winfield, A.F., Winkle, K., Webb, H., Lyngs, U., Jirotko, M. and Macrae, C. 2021. Robot Accident Investigation: A Case Study in Responsible Robotics, *Software Engineering for Robotics*. doi: 10.1007/978-3-030-66494-7_6.
- [10] Abbasi-Kesbi, R., Nikfarjam, A. and Akhavan Hezaveh, A. 2018. Developed wearable miniature sensor to diagnose initial perturbations of cardiorespiratory system. *Healthcare Technology Letters*, 5(6):231-235. doi:10.1049/htl.2018.5027.
- [11] Deylami, A., 2021. Tracking Control of Robots Revisited Based on Taylor Series and Asymptotic Expansion. *Journal of Modern Processes in Manufacturing and Production*. 10(2):63-70. doi: 20.1001.1.27170314.2021.10.2.6.7.
- [12] Mohammed Ali, H., Hashim, Y. and AL-Sakkal, G. 2022. Design and implementation of Arduino based robotic arm. *International Journal of Electrical and Computer Engineering*. 12(2):1411-1411. doi: 10.11591/ijece.v12i2.pp1411-1418.
- [13] Abbasi-Kesbi, R. and Nikfarjam, A. 2018. A miniature sensor system for precise hand position monitoring. *IEEE Sensors Journal*, 18(6):2577-2584. doi: 10.1109/JSEN.2018.2795751.
- [14] Abbasi-Kesbi, R. and Nikfarjam, A. 2015. October. A mini wearable wireless sensor for rehabilitation applications. In 2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM) (pp. 618-622). IEEE. doi: 10.1109/ICRoM.2015.7367854.
- [15] Hägele, M. and Schraft, R.D., 2017. Present State and Future Trends in Mechanical Systems Design for Robot Application, the *Mechanical Systems Design Handbook*. CRC Press.
- [16] Chen, D., Zeng, Z., Guan, Y., Zhu, H. and Zhang, T. 2020. SCARA robots developed with modular method. *IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE. doi: 10.1109/ICMA49215.2020.9233716.
- [17] Liyanage, M.H., Krouglicof, N. and Gosine, R. 2010. October. High speed electro-hydraulic actuator for a scara type robotic arm. *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. doi: 10.1109/IROS.2010.5650759.
- [18] Alnomani, S.N., Nassrullah, K.S. and Imran, A.M. 2021. Improving the workspace for a 3-DOF SCARA robot by investigating the main parameters of Denavit-Hartenberg approach. *Journal of Mechanical Engineering Research and Developments*. 44(6):1-8.
- [19] Al Mashhadany, Y., Abbas, A.K., Algburi, S. and Taha, B.A. 2024. Design and Analysis of a Hybrid Intelligent SCARA Robot Controller Based on a Virtual Reality Model. *Journal of Robotics and Control (JRC)*. 5(6):1722-1735. doi: 10.18196/jrc.v5i6.23158.

- [20] Veneziano, S. and De Simone, M.C. 2023. Multibody Modeling of a Serial Manipulator for In-Space Applications. International Conference New Technologies, Development and Applications. Switzerland. doi: 10.1007/978-3-031-31066-9_39
- [21] Karem, I.S. Wahab, T.A.J. and Yahyh, M.J. 2017. Design and implementation for 3-DOF SCARA Robot based PLC. Al-Khwarizmi engineering journal. 13(2):40-50. doi:10.22153/kej.2017.01.002.
- [22] Abaas, T. F., Khleif, and Abbood, M., Q. 2020. Inverse kinematics analysis and simulation of a 5 DOF robotic arm using MATLAB. Al-Khwarizmi Engineering Journal. 16(1): 1-10. doi:10.22153/kej.2020.12.001.
- [23] Shen, H., Liu, Y., Wu, H., Hu, C. and Wang, S. 2018. July. Forward and inverse kinematics for a novel double scara robot. IOP Conference Series: Earth and Environmental Science. 170(4): 042088.
- [24] Karem, I.S., Wahab, T.A.J. and Yahyh, M.J., 2017. Design and implementation for 3-DOF SCARA Robot based PLC. Al-Khwarizmi engineering journal, 13(2):40-50. doi: 10.22153/kej.2020.12.001.
- [25] Rania Idrees, R., Kareem, I. and Abdul-Lateef, W. E. 2024. Kinematic analysis model of 4-DOF robotic arm. AIP Conference Proceedings. doi:10.1063/5.0206140.
- [26] Abbasi-Kesbi, R., Nikfarjam, A. and Memarzadeh-Tehran, H. 2016. A patient-centric sensory system for in-home rehabilitation. IEEE Sensors Journal. 17(2):524-533. doi: 10.1109/JSEN.2016.2631464.
- [27] Krut, S., Nabat, V., Company, O. and Pierrot, F. 2004. A high-speed parallel robot for Scara motions. IEEE International Conference on Robotics and Automation. doi: 10.1109/ROBOT.2004.1308914.
- [28] Yao, C. and Voronin, S.S. 2018. The Kinematic Analysis of the SCARA on MATLAB. International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). doi: 10.1109/ICIEAM.2018.8728847.
- [29] Gulzar, M.M., Murtaza, A.F., Ling, Q., Javed, M.Y., Rizvi, S.T.H. and Rana, R.A. 2015. Kinematic modeling and simulation of an economical scara manipulator by Pro-E and verification using MATLAB/Simulink. International Conference on Open Source Systems & Technologies (ICOSST) doi: 0.1109/ICOSST.2015.7396410.
- [30] Ma, G.Q., Yu, Z.L., Cao, G.H., Zheng, Y.B. and Liu, L. 2014. The kinematic analysis and trajectory planning study of high-speed SCARA robot handling operation. Applied Mechanics and Materials. 687: 294-299. doi: 10.4028/www.scientific.net/AMM.687-691.294.
- [31] Chanal, H., Guyon, J.B., Koessler, A., Dechambre, Q., Boudon, B., Blaysat, B. and Bouton, N. 2021. Geometrical defect identification of a SCARA robot from a vector modeling of kinematic joints invariants. Mechanism and Machine Theory. 162:104339. doi:0.1016/j.mechmachtheory.2021.104339.