Robotics and Mechatronics

Mini Project Two

Mohammad Montazeri
School of Mechanical Engineering
College of Engineering, University of Tehran
Tehran, Iran; 810699269
mohammadmontazeri@ut.ac.ir

***Abstract*—This mini project is designed to address the kinematics and dynamics of robotic arms using MATLAB and *Simscape* environment. The given case study in this project is the known *Scara* robotic arm.**

***Index Terms*—Scara, dynamics, kinematics, simulation, jacobian**

I. Introduction

SCARA (Selective Compliance Articulated Robot Arm) robots, known for their precise X-Y plane movements, excel in material handling tasks and assembly work. Their versatility and cost-effectiveness make them popular in manufacturing environments. This report discusses a specific kind of this robot, as a valuable asset in various industries.

II. Denavit-Hartenberg Parameters

According to the vectors shown in Figure 1, the following D-H table can be drawn.

TABLE I
The D-H parameters of SCARA robot

i	a_i	b_i	α_i	θ_i	$q_i^{initial}$
1	0	b_1	90°	0	800mm
2	400	257.7	0	θ_2	0
3	250	0	0	θ_3	90°
4	0	$-b_4$	180°	0	100mm

III. Forward Kinematic Problem

$$Q_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix} \tag{1}$$

$$\vec{\mathbf{a_i}} = \begin{bmatrix} a_i \cos\theta_i \\ a_i \sin\theta_i \\ b_i \end{bmatrix} \qquad Q = Q_1 Q_2 Q_3 Q_4 \tag{2}$$

$$\vec{P} = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \mathbf{a_1} + Q_1\mathbf{a_2} + Q_1Q_2\mathbf{a_3} + Q_1Q_2Q_3\mathbf{a_4} \tag{3}$$

$$T = \begin{bmatrix} Q & \vec{P} \\ \vec{0}_{1\times3} & 1 \end{bmatrix} \qquad \phi = \arccos\left(\frac{tr(Q)-1}{2}\right) \qquad \vec{e} = \frac{vect(Q)}{\sin\phi} \tag{4}$$

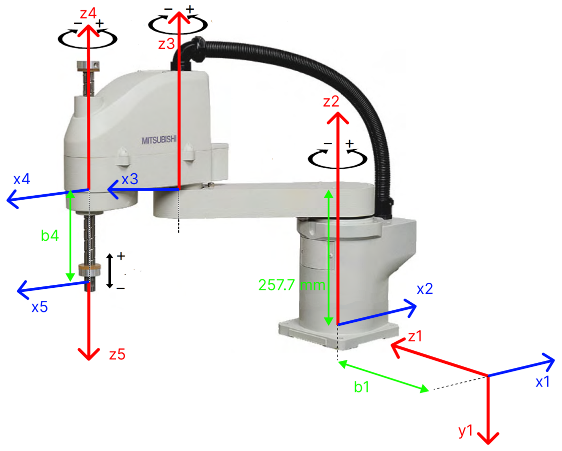

Fig. 1. SCARA robot marked up with D-H vectors. Raw image courtesy of Wikipedia [3].

Having D-H variables of the robot, all parameters above can be calculated via MATLAB. Here's its final results:

$$x = 250.0\,\cos(\theta_2+\theta_3) + 400.0\,\cos(\theta_2)$$
$$y = b_4 + 1.5\,10^{-14}\,\sin(\theta_2+\theta_3) + 2.4\,10^{-14}\,\sin(\theta_2) - 257.7$$
$$z = b_1 - 6.1\,10^{-17}\,b_4 + 250.0\,\sin(\theta_2+\theta_3) + 400.0\,\sin(\theta_2) + 1.6\,10^{-14}$$

Omitting the terms close to zero (of order 10^{-14} or lower), we gain the same values as analytical solution, according to the chosen x_1, y_1, y_1 axes shown in Figure 1.

$$x = 400\cos\theta_2 + 250\cos(\theta_2+\theta_3) \tag{5}$$
$$y = -(257.7 - b_4) \tag{6}$$
$$z = b_1 + 400\sin\theta_2 + 250\sin(\theta_2+\theta_3) \tag{7}$$
$$\phi = \theta_2 + \theta_3 \tag{8}$$

The rotation matrix after being filtered by removing the terms close to zero, can be shown as below. Using equation 4, the final rotation angle can also be derived from the obtained Q.

$$Q = \begin{bmatrix} \cos(\theta_2+\theta_3) & \sin(\theta_2+\theta_3) & 0.0 \\ 0.0 & 0.0 & 1.0 \\ \sin(\theta_2+\theta_3) & 1.0\,\cos(\theta_2+\theta_3) & 0 \end{bmatrix} \tag{9}$$

$$\phi = \cos^{-1}(0.5\cos(\theta_2+\theta_3) - 0.5) \tag{10}$$

The overall homogenous rotation-displacement transformation matrix with the initial conditions[1] chosen for this problem is calculated as:

$$\begin{pmatrix} 0 & 1 & 0 & 400 \\ 0 & 0 & 1 & -158 \\ 1 & 0 & 0 & 1050 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In another simpler representation, the following position and orientation is achieved with the mentioned initial conditions.

$$x = 400mm \quad y = -157.7mm \quad z = 1050mm \quad \phi = 90^\circ$$

IV. Inverse kinematic Problem

Playing with equations 5 to 8 we can find the IKP equations as below. Having the task-space parameters $x, y, z \,\&\, \phi$, we can calculate joint-space parameters $b_1, b_4, \theta_2 \,\&\, \theta_3$ as:

$$\rightarrow b_4 = y + 257.7 \tag{11}$$

$$x = 400\cos\theta_2 + 250\cos\phi \ , \ z = b_1 + 400\sin\theta_2 + 250\sin\phi$$

$$\rightarrow \theta_2 = \arccos\left(\frac{x - 250\cos\phi}{400}\right) \tag{12}$$

$$\rightarrow b_1 = z - 400\sin\theta_2 - 250\sin\phi \tag{13}$$

$$\rightarrow \theta_3 = \phi - \theta_2 \tag{14}$$

With a glance of what's achieved, it can be concluded that there's always a unique answer for b_4, while the *arccos* term in θ_2 results two answers, leading to two answers for each of θ_2, θ_3 and b_1.

$$Answers \longrightarrow \begin{cases} \theta_2^+ \longrightarrow \theta_4^+ \longrightarrow b_1^+ \longrightarrow b_4 \\ \\ \theta_2^- \longrightarrow \theta_1^- \longrightarrow b_1^- \longrightarrow b_4 \end{cases}$$

V. Jacobian Matrix

Since the first and the last joints are prismatic, we'd have

$$\underset{\sim}{\mathcal{J}} = \begin{bmatrix} \vec{O} & \vec{e_2} & \vec{e_3} & \vec{O} \\ \vec{e_1} & \vec{e_2} \times \vec{r_2} & \vec{e_3} \times \vec{r_3} & \vec{e_4} \end{bmatrix} \tag{15}$$

However, since we're dealing with a 4-DoF robot, its Jacobian matrix must have a square (4×4) shape. This robot is meant to access 3 positional (x, y, z) and one angular (ϕ) degrees of freedom. Thus the first row of the mentioned matrix will be reduced to

$$\underset{\sim}{\mathcal{J}} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ \vec{e_1} & \vec{e_2} \times \vec{r_2} & \vec{e_3} \times \vec{r_3} & \vec{e_4} \end{bmatrix}_{4\times4} \tag{16}$$

Deriving the Jacobian in the first joint frame requires calculating each e_i and r_i in the first frame. That can be done using the following transformations.

[1] $\theta_2 = 0, \quad \theta_3 = 90^\circ, \quad b_1 = 800mm, \quad b_4 = 100mm$

$$[\vec{a_1}]_1 = \begin{bmatrix} 0 \\ 0 \\ b_1 \end{bmatrix} \ [\vec{a_2}]_2 = \begin{bmatrix} 400\cos\theta_2 \\ 400\sin\theta_2 \\ 257.7 \end{bmatrix} \ [\vec{a_3}]_3 = \begin{bmatrix} 250\cos\theta_3 \\ 250\sin\theta_3 \\ 0 \end{bmatrix} \ [\vec{a_4}]_4 = \begin{bmatrix} 0 \\ 0 \\ -b_4 \end{bmatrix}$$

$$[\vec{e_2}]_1 = \underset{\sim}{Q_1}\,[\vec{e_2}]_2 \qquad [\vec{e_3}]_1 = \underset{\sim}{Q_1}\underset{\sim}{Q_2}\,[\vec{e_3}]_3 \qquad [\vec{e_4}]_1 = \underset{\sim}{Q_1}\underset{\sim}{Q_2}\underset{\sim}{Q_3}\,[\vec{e_4}]_4$$

$$[\vec{a_2}]_1 = \underset{\sim}{Q_1}\,[\vec{a_2}]_2 \qquad [\vec{a_3}]_1 = \underset{\sim}{Q_1}\underset{\sim}{Q_2}\,[\vec{a_3}]_3 \qquad [\vec{a_4}]_1 = \underset{\sim}{Q_1}\underset{\sim}{Q_2}\underset{\sim}{Q_3}\,[\vec{a_4}]_4$$

$$[\vec{r_2}]_1 = [\vec{a_2}]_1 + [\vec{a_3}]_1 + [\vec{a_4}]_1 \quad [\vec{r_3}]_1 = [\vec{a_3}]_1 + [\vec{a_4}]_1 \quad [\vec{r_4}]_1 = [\vec{a_4}]_1$$

$$[\vec{r_1}]_1 = [\vec{a_1}]_1 + [\vec{a_2}]_1 + [\vec{a_3}]_1 + [\vec{a_4}]_1$$

After doing the math via MATLAB and erasing the negligible terms[2], here's the final result:

$$\mathbf{J} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & -250\sin(\theta_2+\theta_3) - 400\sin(\theta_2) & -250\sin(\theta_2+\theta_3) & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 250\cos(\theta_2+\theta_3) + 400\cos(\theta_2) & 250\cos(\theta_2+\theta_3) & 0 \end{bmatrix}$$

VI. Dynamic Model

To use the Euler-Lagrange method, we need to find the kinetic energy (T) and potential energy (V) of the system.

A. Kinetic Energy

For each link, the kinetic energy is a combination of translational and rotational kinetic energy.

$$T = \sum_{1}^{4} \left(\frac{1}{2} m_i \, \dot{\vec{c}}_i^{\,T} \, \dot{\vec{c}_i} + \frac{1}{2} \vec{\omega}_i^{\,T} \, \mathbf{I}_i \, \vec{c_i} \right) \tag{17}$$

where

- m_i is the mass of the i-th link.
- $\dot{\mathbf{c}}_i^T$ is the linear velocity of the center of mass of the i-th link.
- $\boldsymbol{\omega}_i$ is the angular velocity of the i-th link.
- $\mathbf{I}_i$ is the inertia tensor of the i-th link.

B. Potential Energy

The potential energy is due to gravity:

$$V = \sum_{1}^{4} (m_i g h_i) \tag{18}$$

where h_i is the height of the center of mass of the i-th link. It must be reminded that the obtained potential energy is not going to be directly used in this procedure, rather, its derivatives will be of use (see equation 21). Thus, gravitational potential energy of links with no height alternation, which would be constant along any movement of the robot, will vanish after differentiation. Therefore, we are allowed to ignore them from the very beginning.

It can be put as by taking the ground level at O_4, we would only have the potential energy of the last (prismatic) link. That's because the centers of mass of the other links always have a constant height, invariant of any degree of freedom of

[2] terms of order 10^{-14} or less, caused due to the computational errors which have occurred in the calculation process.

the system. In other words, their height wouldn't change and since the potential energy terms are used after differentiation, these constant values would be omitted. Accordingly, we only calculate the term related to the forth joint.

$$V = U_{g_4} = m_4 g h_4 = -0.15 \times 9.81 \times b_4 \tag{19}$$

C. Euler-Lagrange Equation

The Lagrangian is defined as:

$$\mathcal{L} = T - V \tag{20}$$

Using the Euler-Lagrange equation for each generalized coordinate (q_i):

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i \tag{21}$$

where τ_i represents the generalized forces/torques.

Computing these derivatives for each DoF (q_i) will yield the dynamic equations of motion for the SCARA robot. All this procedure is handled via *Matlab* in a file called `torques_using_model.m` which accompanies this report file. Its results are also plotted and compared with the *later-explained* simulation results via `plotting_results.m` whose figures are attached in the Results section. It must also be mentioned that the following equations have a fundamental influence on the algorithm used to solve the dynamic model.

$$[\mathrm{CoM}]_{DH} = T\,[\mathrm{CoM}]_{\text{link}} \tag{22}$$

$$[I]_{DH} = Q\,[I]_{\text{link}}\,Q^{-1} \tag{23}$$

$$T = \begin{pmatrix} Q & b \\ 0 & 1 \end{pmatrix} \tag{24}$$

D. Simulation

Along with this report comes a series of *STL*, *Simulink* and *Matlab* files which together carry the burden of simulating this robot in *Simscape* environment of *Matlab*. For this matter, the STL files first need to be imported in `my_second_part_R2018b.slx` for precise representation of the robot along its movement. After that, the input variables of the system must be given to the model. Therefore, a subsystem (Figure 3) is added to the simulink file which imports inputs of the system. These inputs include each of the joint-space parameters $(b_1, b_4, \theta_2, \theta_3)$ and their first 2 derivatives.[3] To provide these variables, first the matlab codes must be run, so that the mentioned variables get saved in the workspace. It's suggested to run `parameters.m`, `torques_using_model.m`, `motions_for_simulink.m` and `my_second_part_R2018b.slx` sequentially. It's also worth mentioning that importing the inputs could have been handled in both of the two ways explained below; while both of them are implemented in the provided Simulink file, the latter is currently used and the first method is commented out for now.

1) **Importing the pre-derivated timeseries**: In the `motions_for_simulink.m` file, along with each 100001-sample position/orientation variable, its first two derivatives are also calculated and available. We can directly import all of them into simscape.
2) **Derivating using simulink**: It's also an option to only import the basic joint-space variables and use *Simulink-PS Converter* block to find their derivates numerically via *Input Handling* tab of the block.

Using the combination of mentioned files, a smooth 4-5-6-7 trajectory planing pick-and-place is modeled whose initial and final situations are shown in Figure 4. It's worth mentioning that the initial and final conditions of the simulation is different from those of analytical solution and are defined as:

$$\theta_{init}^{sim} = [0, 0, 0, 0]$$
$$\theta_{fin}^{sim} = \theta_{init}^{sim} + [0.5, \pi/3, -\pi/6, -0.05]$$

After all, the output of the simulation, which is the collection of joint-space forces and torques, is both plotted and passed to the Matlab workspace for further comparison of the results. Figure 5 shows the overall result of the simulation. Note that both analytical solution and simulation use a "4-5-6-7" pick-and-place trajectory with the following formulation:

$$\underline{\mathbf{s}(\tau) = -\mathbf{20}\tau^{\mathbf{7}} + \mathbf{70}\tau^{\mathbf{6}} - \mathbf{84}\tau^{\mathbf{5}} + \mathbf{35}\tau^{\mathbf{4}}} \tag{25}$$

There are also some miscellaneous functions like Q_x, Q_y, Q_z, Q_{func}, and a_{func} alongside other files, which are used in the context of provided code scripts and are defined separately to organize the codes more neatly.

At the end, The obtained results from both of the mentioned methods are plotted together to enable a thorough comparison. Figure 6 shows that both methods almost predict the same values for the forces acting in the prismatic joints and the torques acting on the revolute ones. The negligible differences can also be due to the different initial states two models commence their motion from. The imprecision of considered D-H parameters, geometrical inaccuracy of the STL model and computational errors might also be responsible of the errors. Overall, it can be concluded that the theoretical expectations mostly confirm with the simulation.

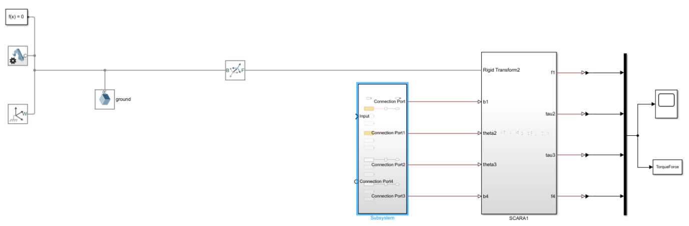

Fig. 2. An overview of the provided simulink environment

[3]In other words, the position/orientation, velocity and acceleration of each joint.

VII. Results

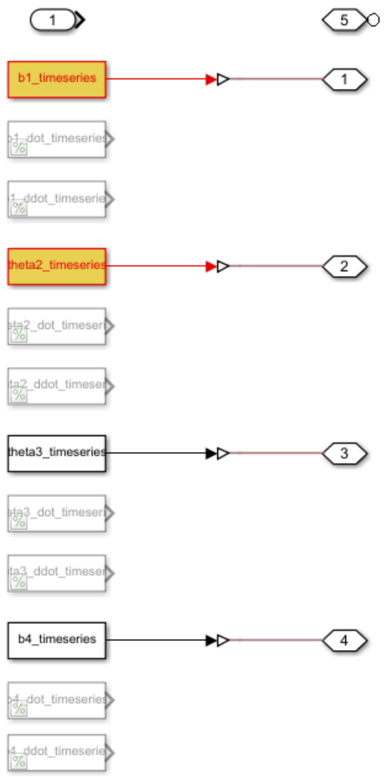

Fig. 3. Modified inputs defined as subsystem added to the simulink file

VIII. Conclusion

SCARA robots offer valuable advantages in manufacturing and assembly tasks due to their speed, precision, and cost-effectiveness. However, their limited range of motion restricts their suitability for certain applications. In conclusion, this report has delved into the *kinematics* and *dynamics* of a specific SCARA robot, deriving its D-H parameters, solving its Forward and Inverse kinematic problem, finding its Jacobian, achieving its dynamic model and verifying the results with MATLAB *Simscape* simulations.

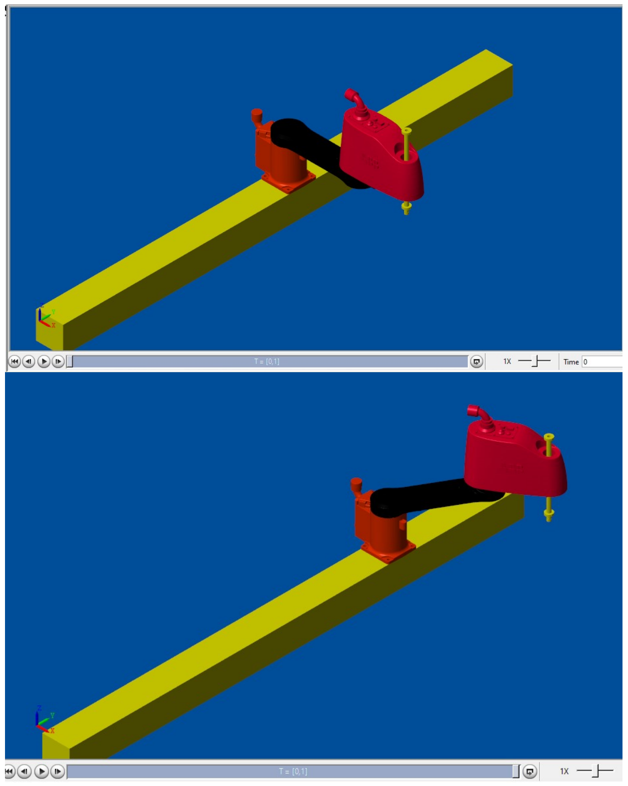

Fig. 4. Initial and final condition of the robot in the simulation

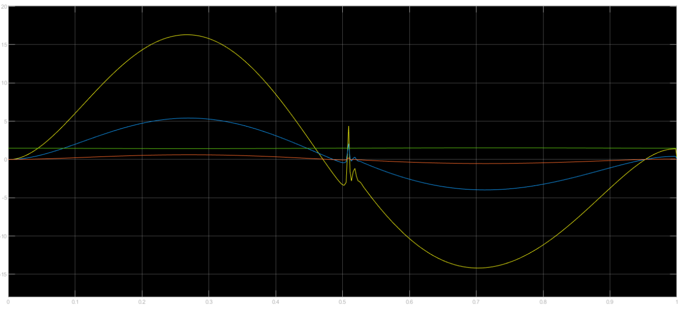

Fig. 5. Overall result of the simulation on the joint forces and torques

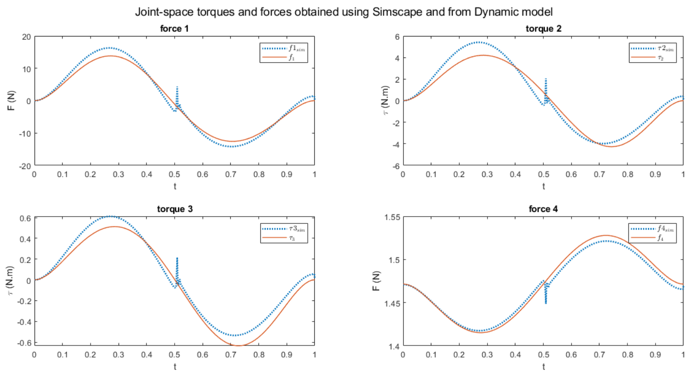

Fig. 6. Comparison of the simulation and analytical results in dynamic modeling of the SCARA robot under review

References

[1] J. Angeles, “Fundamentals of Robotic Mechanical Systems”, Theory, Methods, and Algorithms, 4th edition, Springer.

[2] J. J. Craig, “Introduction to Robotics”, 3d edition, Pearson Education, Inc.

[3] Wikipedia, 2023, “SCARA”, Wikimedia Foundation, Last modified July 4, 2023. https://en.wikipedia.org/wiki/SCARA.

[4] M. Kazim, J. Hong, M. Kim, K. K. Kim, (2023). “Recent advances in path integral control for trajectory optimization: An overview in theoretical and algorithmic perspectives”. Annual Reviews in Control, 57, 100931. [Online]. Available: https://doi.org/10.1016/j.arcontrol.2023.100931

[5] B. Nadir, O. Mohammed, N. Minh-Tuan, et al. “Optimal trajectory generation method to find a smooth robot joint trajectory based on multiquadric radial basis functions”. Int J Adv Manuf Technol 120, 297-312 (2022). [Online]. Available: https://doi.org/10.1007/s00170-022-08696-1