Mechatronics & Robotics
Mini Project 2

1st Alireza Kamali Ardakani
School of Mechanical Engineering
University of Tehran
Tehran, Iran
alirezakamali@ut.ac.ir

2nd Mehdi Tale Masouleh
School of Electrical and Computer Engineering
University of Tehran
Tehran, Iran
m.t.masouleh@ut.ac.ir

***Abstract*—In this course project, the Scara robot model IRB910SC was analyzed in terms of kinematics (solving direct and inverse kinematics) and dynamics (in the form of modeling and simulation in the Simulink/Simscape environment). There were challenges in obtaining accurate calculations, which were tried with a lot of trial and error so that there is minimal error between modeling and simulation.**

***Index Terms*—Forward Kinematic, Inverse Kinematic, Jacobian Matrix, Dynamic Modeling, DH Parameters**

I. Introduction

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(Fig 1). In this project, the simulink file for the robotic arm and its physical parameters are given and attached.

Fig. 1. ABB IRB 910 SC Scara Robot

To obtain the kinematic equations, the analysis of the DH parameters is required. This will be done according to the given dimensions of the robot. In the following, using the relationships in the lesson, the equations are extracted in order. It should be noted that some equations are written according to the convention set in the attached article. Results can also be seen in the attached files; Simulink-Kamali.m for generating joint variables according to 4567 trajectory motion, ComparingPlotResults.m for comparing the generated output dynamic variable from the simulation, and also the model. FKP-IKP-JACOBIAN-KAMALI.m contain equations that generate the solution of kinematic. some assumptions (for example conj(theta) = theta) are considered to simplify answers in all codes and cause the program to be fast-generating.

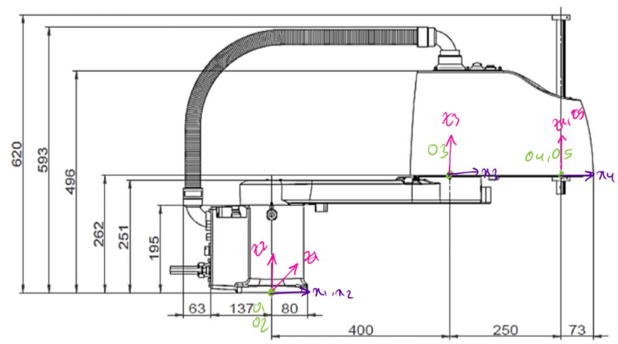

Fig. 2. DH axis and Frames

II. DH (Denavit Hartenberg) Parameters

First of all, the coordinate axes are displayed as follows based on the similarity of the z-axis with the simulation and also the rules in the book (Fig 2). After deciding on the directions of the axes, it is time to adjust and prepare the DH table (I)

i	a_i	b_i	α_i	θ_i
1	0	b_1	$\pi/2$	0
2	400	251.1	0	θ_2
3	250	0	0	θ_3
4	0	b_4	0	0

TABLE I
DH Table

According to the following formula in direct kinematics, the rotation matrices and vectors a_i 's are calculated.

$$\mathbf{Q}_i = \begin{bmatrix} \cos\theta_i & -\lambda_i \sin\theta_i & \mu_i \sin\theta_i \\ \sin\theta_i & \lambda_i \cos\theta_i & -\mu_i \cos\theta_i \\ 0 & \mu_i & \lambda_i \end{bmatrix}, \quad \mathbf{a}_i = \begin{bmatrix} a_i \cos\theta_i \\ a_i \sin\theta_i \\ b_i \end{bmatrix}$$

Now, these results from MATLAB coding are obtained:

$$Q_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$Q_2 = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$Q_3 = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$Q_4 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$a_1 = \begin{pmatrix} 0 \\ 0 \\ \frac{b_1}{1000} \end{pmatrix}$$

$$a_2 = \begin{pmatrix} \frac{2\cos(\theta_2)}{5} \\ \frac{2\sin(\theta_2)}{5} \\ \frac{2511}{10000} \end{pmatrix}$$

$$a_3 = \begin{pmatrix} \frac{\cos(\theta_3)}{4} \\ \frac{\sin(\theta_3)}{4} \\ 0 \end{pmatrix}$$

$$a_4 = \begin{pmatrix} 0 \\ 0 \\ \frac{-b_4}{1000} \end{pmatrix}$$

III. Forward Kinemtaic Problem(FKP)

$$[\,\mathbf{Q}_6\,]_1[\,\mathbf{Q}_5\,]_1[\,\mathbf{Q}_4\,]_1[\,\mathbf{Q}_3\,]_1[\,\mathbf{Q}_2\,]_1[\,\mathbf{Q}_1\,]_1 = [\,\mathbf{Q}\,]_1$$

$$[\,\mathbf{a}_1\,]_1 + [\,\mathbf{a}_2\,]_1 + [\,\mathbf{a}_3\,]_1 + [\,\mathbf{a}_4\,]_1 + [\,\mathbf{a}_5\,]_1 + [\,\mathbf{a}_6\,]_1 = [\,\mathbf{p}\,]_1$$

From the above equations end effector orientation and position can be written. Also, notice that Q0 in Matlab Code shows a zero rotation matrix that generates joint zero orientation.

$$\mathbf{P}_{\mathrm{EE}} = \begin{pmatrix} 0.25\cos(\theta_2+\theta_3) - 0.001\,b_4 + 0.4\cos(\theta_2) + 0.2511 \\ 0.001\,b_1 + 0.25\sin(\theta_2+\theta_3) + 0.4\sin(\theta_2) \\ 0.2511 - 0.001\,b_4 \end{pmatrix}$$

$$\mathbf{Q}_{\mathrm{EE}} = \begin{pmatrix} \cos(\theta_2+\theta_3) & -1.0\sin(\theta_2+\theta_3) & 1.0 \\ \sin(\theta_2+\theta_3) & \cos(\theta_2+\theta_3) & 0 \\ 0 & 0 & 1.0 \end{pmatrix}$$

IV. Inverse Kinematic Problem(IKP)

$$\mathrm{xEE} = 0.25\cos(\theta_2+\theta_3) - 0.001\,b_4 + 0.4\cos(\theta_2) + 0.2511$$

$$\mathrm{yEE} = 0.001\,b_1 + 0.25\sin(\theta_2+\theta_3) + 0.4\sin(\theta_2)$$

$$\mathrm{zEE} = 0.2511 - 0.001\,b_4$$

$$\phi = \theta_2 + \theta_3$$

Now, from the 'solve' function in MATLAB, IKP will be obtained explicitly and no numerical method or simplification is needed. here are the results:

$$b_1 = \begin{pmatrix} 1000.0\,\mathrm{yEE} - 250.0\sin(\phi) - 400.0\sqrt{1-(2.5\,\mathrm{zEE} - 2.5\,\mathrm{xEE} + 0.625\cos(\phi))^2} \\ 1000.0\,\mathrm{yEE} - 250.0\sin(\phi) + 400.0\sqrt{1-(2.5\,\mathrm{zEE} - 2.5\,\mathrm{xEE} + 0.625\cos(\phi))^2} \end{pmatrix} \tag{1}$$

$$\theta_2 = \begin{pmatrix} \mathrm{acos}(2.5\,\mathrm{xEE} - 2.5\,\mathrm{zEE} - 0.625\cos(\phi)) \\ -1.0\,\mathrm{acos}(2.5\,\mathrm{xEE} - 2.5\,\mathrm{zEE} - 0.625\cos(\phi)) \end{pmatrix} \tag{2}$$

$$\theta_3 = \begin{pmatrix} \mathrm{acos}(2.5\,\mathrm{xEE} - 2.5\,\mathrm{zEE} - 0.625\cos(\phi)) \\ -1.0\,\mathrm{acos}(2.5\,\mathrm{xEE} - 2.5\,\mathrm{zEE} - 0.625\cos(\phi)) \end{pmatrix} \tag{3}$$

$$b_4 = \begin{pmatrix} 251.1 - 1000.0\,\mathrm{zEE} \\ 251.1 - 1000.0\,\mathrm{zEE} \end{pmatrix} \tag{4}$$

V. Jacobian Matrix(J)

For obtaining Jacobian matrix that shows relation between twist and change rates of joint variable, this relations can be written.

$$t = J\dot{\theta} \tag{5}$$

$$J = \begin{pmatrix} 0 & 1.0 & 1.0 & 0 \\ 0 & -0.25\sin(\theta_2+\theta_3) - 0.4\sin(\theta_2) & -0.25\sin(\theta_2+\theta_3) & 1.0 \\ 1.0 & 0.25\cos(\theta_2+\theta_3) + 0.4\cos(\theta_2) & 0.25\cos(\theta_2+\theta_3) & 0 \\ 0 & 0.25\sin(\theta_2+\theta_3) + 0.4\sin(\theta_2) & 0.25\sin(\theta_2+\theta_3) & 1.0 \end{pmatrix} \tag{6}$$

VI. Dynamic Modeling

To obtain the dynamic equations, all the mass of the links, their moment of inertia, and also their center of mass should be converted to the said frames with the relationships mentioned in the project text, and then use those findings in the equations. **For dynamic modeling, 2 models with Different equations and approach are derived and both results are reported in continued.**

$$[CoM]_{DH} = T[CoM]_{link}$$

$$[I]_{DH} = Q[I]_{link}Q^{-1}$$

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

$$\boldsymbol{r}_{11} = \boldsymbol{Q}_z(\theta_1 - \theta_1^{\mathrm{init}})\boldsymbol{c}_1$$

$$\boldsymbol{r}_{ij} = \sum_{k=i}^{j-1}[\boldsymbol{a}_k]_1 + [\boldsymbol{Q}_z(\theta_j - \theta_j^{\mathrm{init}})\boldsymbol{c}_j]_1;\ \text{for } j \neq 1 \text{ and } i \leq j$$

$$\left[\mathbf{Q}_z\left(\theta_j-\theta_j^{\text{init}}\right)\mathbf{c}_j\right]_1=\left(\prod_{k=1}^{j-1}\mathbf{Q}_k\right)\mathbf{Q}_z\left(\theta_j-\theta_j^{\text{init}}\right)\mathbf{c}_j.$$

$$[W_1]_1=\begin{bmatrix}z & 0 & 0 & 0\end{bmatrix}$$

$$[W_2]_2=\begin{bmatrix}Q_1^T z & z & 0 & 0\end{bmatrix}$$

$$[W_3]_3=\begin{bmatrix}Q_2^T Q_1^T z & Q_2^T z & z & 0\end{bmatrix}$$

$$[W_4]_4=\begin{bmatrix}Q_3^T Q_2^T Q_1^T z & Q_3^T Q_2^T z & Q_3^T z & z\end{bmatrix}$$

$$W_i=\begin{bmatrix}e_1 & e_2 & \cdots & e_i & 0_{3\times 1} & 0_{3\times 1}\end{bmatrix}$$

$$T_i=\frac{1}{2}\dot{\theta}^T M_i\dot{\theta} \tag{7}$$

$$V_i=m_i g h_i \tag{8}$$

$$\tau=M\ddot{\theta}+\dot{M}\dot{\theta}-\frac{\partial T}{\partial\theta}+\frac{\partial V}{\partial\theta} \tag{9}$$

After simplification in the MATLAB environment, the below results are obtained. note that transformations of vectors from one frame to another frame were done by a rotation matrix. some assumptions also have been considered in MATLAB which can be seen in the attached text code. For example, for filtering numbers lower than 0.0001 a code algorithm was implemented for this purpose. This algorithm happens with filterMatrix and zeroSmallNumbers function in Matlab. 1

```
Function filterMatrix(inputMatrix):
    threshold = 0.000001;
    filteredMatrix = sym(zeros(size(inputMatrix)));
    for i = 1:size(inputMatrix, 1) do
        currentRow = inputMatrix(i, :);
        for j = 1:numel(currentRow) do
            filteredMatrix(i, j) =
             zeroSmallNumbers(currentRow(j),
             threshold);
            filteredMatrix(i, j) =
             simplify(filteredMatrix(i, j));
            filteredMatrix(i, j) = vpa(filteredMatrix(i, j),
             3)
        end
    end
    return filteredMatrix;
Function zeroSmallNumbers(expr, threshold):
    expr = mapSymType(expr, 'constant', @(c) c *
     (abs(double(c)) ≥ threshold))
```

Algorithm 1: Filter Matrix

$$\begin{aligned}F_1 = {} & 3.0\,b_{\ddot{1}}-0.1164\,{\theta_{\dot{2}}}^2\cos(\theta_2)-0.7324\,{\theta_{\dot{2}}}^2\sin(\theta_2)\\
&+0.5145\,\theta_{\ddot{2}}\cos(\theta_2+\theta_3)+0.5145\,\theta_{\ddot{3}}\cos(\theta_2+\theta_3)\\
&-0.0955\,\theta_{\ddot{2}}\sin(\theta_2+\theta_3)-0.0955\,\theta_{\ddot{3}}\sin(\theta_2+\theta_3)\\
&+0.7324\,\theta_{\ddot{2}}\cos(\theta_2)-0.1164\,\theta_{\ddot{2}}\sin(\theta_2)\\
&-0.0955\,{\theta_{\dot{2}}}^2\cos(\theta_2+\theta_3)-0.0955\,{\theta_{\dot{3}}}^2\cos(\theta_2+\theta_3)\\
&-0.5145\,{\theta_{\dot{2}}}^2\sin(\theta_2+\theta_3)-0.5145\,{\theta_{\dot{3}}}^2\sin(\theta_2+\theta_3)\\
&-1.029\,\theta_{\dot{2}}\,\theta_{\dot{3}}\sin(\theta_2+\theta_3)-0.191\,\theta_{\dot{2}}\,\theta_{\dot{3}}\cos(\theta_2+\theta_3)\end{aligned} \tag{10}$$

$$\begin{aligned}\tau_2 = {} &\\
&0.8937\,\theta_{\ddot{2}}+0.4622\,\theta_{\ddot{3}}-0.0573\,{\theta_{\dot{3}}}^2\cos(\theta_3)\\
&-0.3087\,{\theta_{\dot{3}}}^2\sin(\theta_3)+0.04856\,{\theta_{\dot{2}}}^2\cos(\sigma_3)\\
&+0.04856\,{\theta_{\dot{3}}}^2\cos(\sigma_3)-0.2058\,\theta_{\ddot{2}}\,\sigma_2\\
&-0.1029\,\theta_{\ddot{3}}\,\sigma_2+0.1407\,{\theta_{\dot{2}}}^2\sin(\sigma_3)\\
&+0.1407\,{\theta_{\dot{3}}}^2\sin(\sigma_3)+0.0382\,\theta_{\ddot{2}}\,\sigma_1\\
&+0.0191\,\theta_{\ddot{3}}\,\sigma_1-0.1204\,\theta_{\ddot{2}}\cos(2.0\,\theta_2)\\
&+0.03224\,\theta_{\ddot{2}}\sin(2.0\,\theta_2)+0.5145\,b_{\ddot{1}}\cos(\theta_2+\theta_3)\\
&-0.0955\,b_{\ddot{1}}\sin(\theta_2+\theta_3)+0.0382\,{\theta_{\dot{2}}}^2\sigma_2\\
&+0.0191\,{\theta_{\dot{3}}}^2\sigma_2+0.2058\,{\theta_{\dot{2}}}^2\sigma_1\\
&+0.1029\,{\theta_{\dot{3}}}^2\sigma_1+0.7324\,b_{\ddot{1}}\cos(\theta_2)\\
&-0.1164\,b_{\ddot{1}}\sin(\theta_2)+0.03224\,{\theta_{\dot{2}}}^2\cos(2.0\,\theta_2)\\
&+0.6174\,\theta_{\ddot{2}}\cos(\theta_3)+0.3087\,\theta_{\ddot{3}}\cos(\theta_3)\\
&+0.1204\,{\theta_{\dot{2}}}^2\sin(2.0\,\theta_2)-0.1146\,\theta_{\ddot{2}}\sin(\theta_3)\\
&-0.0573\,\theta_{\ddot{3}}\sin(\theta_3)-0.1407\,\theta_{\ddot{2}}\cos(\sigma_3)\\
&-0.1407\,\theta_{\ddot{3}}\cos(\sigma_3)+0.04856\,\theta_{\ddot{2}}\sin(\sigma_3)\\
&+0.04856\,\theta_{\ddot{3}}\sin(\sigma_3)-0.1146\,\theta_{\dot{2}}\,\theta_{\dot{3}}\cos(\theta_3)\\
&-0.6174\,\theta_{\dot{2}}\,\theta_{\dot{3}}\sin(\theta_3)+0.09712\,\theta_{\dot{2}}\,\theta_{\dot{3}}\cos(\sigma_3)\\
&+0.2814\,\theta_{\dot{2}}\,\theta_{\dot{3}}\sin(\sigma_3)+0.0382\,\theta_{\dot{2}}\,\theta_{\dot{3}}\,\sigma_2\\
&+0.2058\,\theta_{\dot{2}}\,\theta_{\dot{3}}\,\sigma_1\\
&\text{where}\\
&\sigma_1=\sin(2.0\,\theta_2+\theta_3)\\
&\sigma_2=\cos(2.0\,\theta_2+\theta_3)\\
&\sigma_3=2.0\,\theta_2+2.0\,\theta_3\end{aligned} \tag{11}$$

$\tau_3 =$

$$
\begin{gathered}
0.4622\,\theta_{\ddot{2}} + 0.4622\,\theta_{\ddot{3}} \\
+ 0.0573\,\theta_{\dot{2}}{}^2\,\cos(\theta_3) \\
+ 0.3087\,\theta_{\dot{2}}{}^2\,\sin(\theta_3) \\
+ 0.04856\,\theta_{\dot{2}}{}^2\,\cos(\sigma_1) \\
+ 0.04856\,\theta_{\dot{3}}{}^2\,\cos(\sigma_1) \\
- 0.1029\,\theta_{\ddot{2}}\,\sigma_3 \\
+ 0.1407\,\theta_{\dot{2}}{}^2\,\sin(\sigma_1) \\
+ 0.1407\,\theta_{\dot{3}}{}^2\,\sin(\sigma_1) \\
+ 0.0191\,\theta_{\ddot{2}}\,\sigma_2 \\
+ 0.5145\,b_{\ddot{1}}\,\cos(\theta_2 + \theta_3) \\
- 0.0955\,b_{\ddot{1}}\,\sin(\theta_2 + \theta_3) \\
+ 0.0191\,\theta_{\dot{2}}{}^2\,\sigma_3 \\
+ 0.1029\,\theta_{\dot{2}}{}^2\,\sigma_2 \\
+ 0.3087\,\theta_{\ddot{2}}\,\cos(\theta_3) \\
- 0.0573\,\theta_{\ddot{2}}\,\sin(\theta_3) \\
- 0.1407\,\theta_{\ddot{2}}\,\cos(\sigma_1) \\
- 0.1407\,\theta_{\ddot{3}}\,\cos(\sigma_1) \\
+ 0.04856\,\theta_{\ddot{2}}\,\sin(\sigma_1) \\
+ 0.04856\,\theta_{\ddot{3}}\,\sin(\sigma_1) \\
+ 0.09712\,\theta_{\dot{2}}\,\theta_{\dot{3}}\,\cos(\sigma_1) \\
+ 0.2814\,\theta_{\dot{2}}\,\theta_{\dot{3}}\,\sin(\sigma_1) \\
\text{where} \\
\sigma_1 = 2.0\,\theta_2 + 2.0\,\theta_3 \\
\sigma_2 = \sin(2.0\,\theta_2 + \theta_3) \\
\sigma_3 = \cos(2.0\,\theta_2 + \theta_3)
\end{gathered}
\tag{12}
$$

$$F_4 = 0.3\,b_{\ddot{4}} + 1.471 \tag{13}$$

VII. Conclusion

To be able to compare the output of the modeling and its obtained equations with the simulation performed in the Simscape environment, first a polynomial 4 5 6 7 should be considered for the smooth movement of the joint variables and then their values in different iterations, in a specific operation pick and place time is specifically saved. These values are given to the dynamic function and the output which are (F_1 , tau_2 , tau_3 , F_4 is plotted in another file. Now, the results for one of two models are reported:

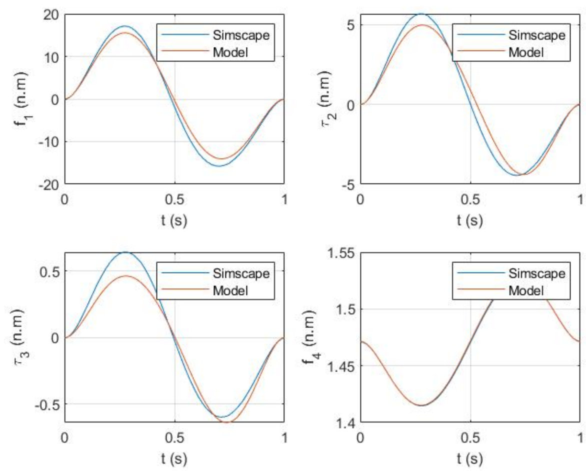

Fig. 3. Resutls For Dynamic Model number 1

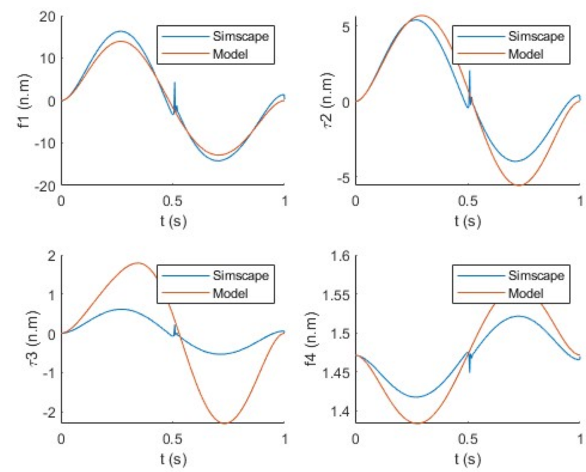

Fig. 4. Resutls For Dynamic Model number 2

In this project, which finally led to the verification of the modeling results with the dynamic simulation results, many assumptions were made to achieve the results of the equations, and finally, according to the two models written in the Matlab file, the correctness of the equations was accepted.

References

[1] J. Angeles, Fundamentals of Robotic Mechanical Systems — Theory, Methods, and Algorithms, 01 2007, vol. 124