

SCARA Robot Links Length Optimization by Using MATLAB and Verification with SimMechanics and Solidworks

Ph.D. Mustafa Mashali^{1*}, Miftah Addeif², Mohamed Embarak³

¹ mhmashali@elmergib.edu.ly, ² moftaheddeif@gmail.com, ³
mohamed.embarak197@gmail.com

^{1,2,3} Department of Mechanical Engineering, College of Engineering, Elmergib University, Libya

*Corresponding author email:

ABSTRACT

This paper aims for optimizing links length that consumed the minimum energy, for a customized SCARA robot. Nine link length combinations are tested and simulated. This research is a part of a project of designing a robotic arm for a packing task. Kinematic and dynamic studies are performed for a 2R robotic arm. The results of kinematic study which are angular displacement, angular velocity and angular acceleration for each joint are determined and exported to the dynamic study to obtain the torque and power consumed. The dynamic study is performed with the aid of MATLAB code, MATLAB/SimMechanics and Solidworks are used to simulate and analyze the dynamic of the robotic arm. The energy consumed for each link length combination using the three methods is calculated.

Keywords: Solidworks Motion Study, MATLAB/SimMechanics, SCARA Robot, Link length optimization.

1 Introduction

In light of technological development and the increased demand for a huge amount of products of all kinds, it has become very important to increase reliance on robots in factories. This is because of their speed and accuracy that are essential in industrial sectors. Using robots can protect the laborers from multiple hazards such as toxic chemicals, high temperature and physical pollution. In addition, they lead to reduce the density of workers and maintain safe distances among the operators that can significantly mitigate the risk of spreading epidemics and infectious diseases such as the Corona pandemic (COVID-19).

This research is a part of a project that aims of designing a robotic arm for packing Tuna-cans at Al-Wafa Tuna factory at Al-Khums Libya. This paper focuses on finding the optimal links length of robotic arm for packing process. Figure 1 shows the packing unit of Tuna-cans in the factory. The packing unit consists of a table with a conveyer belt and six workers at six substations. The robotic arm is attached to the one of the substations. The designed robotic arm is SCARA (Selective Compliant Assembly Robot Arm) configuration. That is because

SCARA robot is characterized by speed and accuracy that are required for assembly and packing process [1].

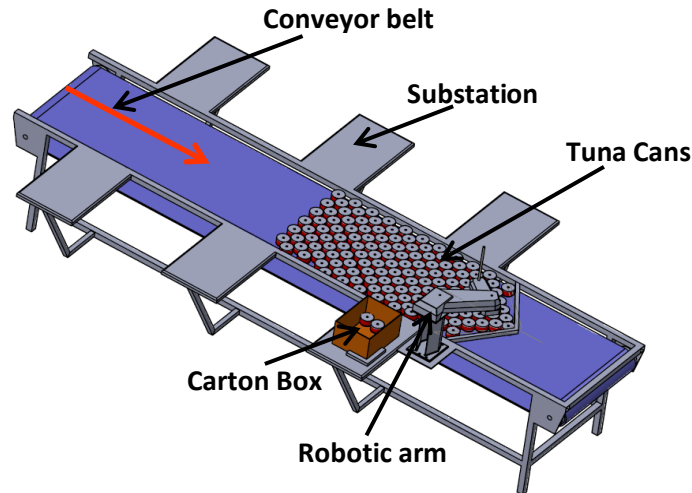


Figure 1: SCARA robot in the packing process

This arm was designed by Japanese engineer Hiroshi Makino in 1979 [1, 2]. A lot of studies has been carried out for this arm. A problem of verifying the kinematic model and the technical specifications of the four degrees of freedom SCARA robot by applying the D-H method was addressed by Fang et.al. The kinematic model was created in a MATLAB environment and simulated to monitor the movement of each joint in the robot to verify the validity of the model[3]. Alshamasin et.al presented the design and development of the four-axis SCARA robot system for drilling tasks. They worked on developing and simplifying the mathematical model. They developed it by deriving depending on the forward and inverse kinematic equations from the D-H (Denavit–Hartenber) model. The Solid Dynamic design program and the MATLAB Simulink program were used for mathematical analysis to confirm the results. The results are compatible with both programs[4]. AVCHAT et.al demonstrated the design of a SCARA three degree of freedom robot with working area dimensions of 1200 x 500mm. This arm can maintain torque of 60 N-m, at the end of the clutch. Ansys software was used in simulating the design[5]. Elaikh et.al presented the kinematic analysis by giving an angle at each joint and analyzing vibration using the Lagrange method. This is to obtain the frequency equation and analyze the effects of vibration loading on the dynamic stability of the robotic arm. The vibration analysis was for the possibility of improving the robotics system performance of the type of SCARA robots[6]. Mariappan et.al presented four-degree-freedom SCARA robot. It designed for drilling tasks and developed using Solidworks software. The CAD files were converted into a block diagram in a MATLAB/SimMechanics environment in the second generation technology. A simulation was conducted to monitor performance and dynamic variables. The results indicated that the structure variables affect the dynamic variables[7].

Fernini presented the design and development of the robot system using Solidworks and modeling using MATLAB/Simulink. Analysis was also performed for this robot to study the dynamic behavior when moving in a straight line by comparing the elbow movement up or down, which showed in the results that there is no difference in the energy exerted[8]. Gouasmi.et.al presented two degrees of freedom robot design for pick and place tasks. Solidworks software was used to design and develop the robot. MATLAB/Simulink software was used to simulate and verify the results. This was done by giving an arc path for movement, and testing the movement of the elbow up and down. By reviewing the results, it was found that the movement of the elbow down requires the least torque for the robot joints[9]. Subhashini presented method to find the optimal link lengths of SCARA robot while moving in a rectangular path to remove the edges of the rectangular pieces. He also addressed the topic of finding the optimal link lengths of the robot when using the circular path by setting different lengths and analyzing the energy used in each case. The lengths that give the least energy possible were determined[10, 11].

In this paper, the problem of choosing the optimal links length for straight line trajectory is presented. This problem is for a modified SCARA robot arm. The joint configuration of the robot is (R-R-P) where R is a revolute joint and P is a prismatic joint. Solidworks motion and MATLAB/SimMechanics software is used for dynamic study. The straight-line path is generated. Torque, power and energy consumption are measured and analyzed at each joint to obtain the optimal links length in which the overall energy consumption is minimized.

2 Kinematic and Dynamic Model

As it can be seen from Figure 1, the designed robotic arm has a complicated geometry. The aim of this research is to determine the optimal length for the first and second links that can be carried out without using the actual geometry. Therefore, a simple arm of 2R is enough for performing the experiments. Figure 2 shows 3D model of the simple arm created in Solidworks. Kinematic model gives a consideration to the motion of the arm neglecting forces that cause this motion. The kinematic model can be determined using D-H method. To define the D-H parameters for the simple arm, first, the frames have to be attached to the arm as shown in Figure 2A. Second, The D-H parameters can be determined according to the methodology explained in[12]. D-H parameters are illustrated in Table 1.

2.1 Forward Kinematic Model

The forward kinematic model calculates the end-effector position and orientation of a robot from joint angles and link lengths. Equation (1) is the general equation for determining the position and orientation of the end-effector for an arm with N joints. In this case $N = 2$.

$${}^0T_N = {}^0T_1 {}^1T_2 \dots \dots {}^{N-1}T_N \quad (1)$$

Where N is the number of joints

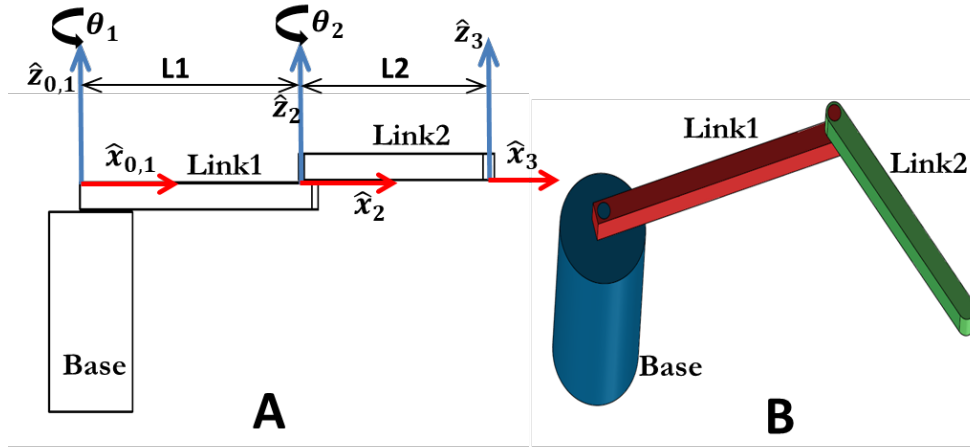


Figure 2: demonstrate 2-D and 3-D model of the RR simple arm and frame assignment.

Table 1: D-H parameters.

I	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	l_1	0	θ_2
3	0	l_2	0	0

$$\begin{aligned}
 {}^0_1T &= \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad {}^0_2T = \begin{bmatrix} C_{12} & -S_{12} & 0 & l_1 C_1 \\ S_{12} & C_{12} & 0 & l_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^0_3T &= \begin{bmatrix} C_{12} & -S_{12} & 0 & l_1 C_1 + l_2 C_{12} \\ S_{12} & C_{12} & 0 & l_1 S_1 + l_2 S_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{2}$$

where C_1 is $\cos(\theta_1)$, S_1 is $\sin(\theta_1)$, C_{12} is $\cos(\theta_1 + \theta_2)$, S_{12} is $\sin(\theta_1 + \theta_2)$, l_1 is the length of the first link, l_2 is the length of the second link.

2.2 Jacobian Matrix

The Jacobian matrix ($J(\theta)$) expresses the relationship between joints velocity ($\dot{\theta}$) relative to linear velocity (V) and angular velocity (ω) of end-effector. In the case of simple arm, there is no orientation so the equation will be as follow:

$$V = J(\theta)\dot{\theta} \tag{3}$$

The Jacobian matrix in simple arm case is as follow:

$$J(\theta) = \begin{bmatrix} -l_1 S_1 - l_2 S_{12} & -l_2 S_{12} & 0 \\ l_1 C_1 + l_2 C_{12} & l_2 C_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

2.3 Dynamic Model

The dynamic model studies the behaviour of moving objects and the forces acting on them. The equations of motion are derived using Newton Euler's method as it is explained in[12]. This is to determine the torque at each joint (τ_1, τ_2) .

$$\tau_1 = \left(I_1 + I_2 - l_1 S_2 - \frac{1}{2} l_2 + l_1 C_2 + \frac{m_1}{4} l_1^2 + \frac{1}{2} l_2 + \frac{l_1 C_2}{2} \right) \ddot{\theta}_1 + I_2 \ddot{\theta}_2 + (l_1 C_2 - m_2 l_1 C_2 S_2 + l_1 l_2 m_2 S_2) \dot{\theta}_1^2 + \left(\frac{l_1 l_2 m_2 S_2}{2} \right) (\dot{\theta}_1 + \dot{\theta}_2)^2 \quad (5)$$

$$\tau_2 = \left(I_2 + \frac{l_2}{2} + \frac{l_1 C_2}{2} \right) \ddot{\theta}_1 + \left(I_2 + \frac{l_2}{2} \right) \ddot{\theta}_2 + (l_1 l_2 m_2 S_2) \dot{\theta}_1^2 \quad (6)$$

$$m_i = \rho * h^2 * l_i \quad (7)$$

where I_i is the moment of inertia for the link i , m_i is the mass of the link i (kg), ρ is the density of the material (kg/m^3). The power consumed by the joint motors is determined as follow:

$$P_j = \dot{\theta}_j * \tau_j \quad (8)$$

$$E = \sum_{k=1}^{n-1} \frac{P_j(k+1) + P_j(k)}{2} * dt \quad (9)$$

where: P_j is power consumed in the joint j , $n = \frac{t}{dt}$; t is the complete period of time, dt : is the time interval, E is the energy consumption which is calculated using numerical integration (trapezoidal method) of the power versus time curve.

3 Problem formulation

To minimize the energy consumed, the robot's end-effector has to move in a straight line trajectory to pick and place Tuna cans. As shown in Figure 3, the robot moves from picking position, in which the robot arm is shown in red color, to the placing position in which the robot arm are shown in blue color. The picking position is fixed by cans' guides. This is to eliminate the cost of vision system for detecting the cans position. So in order to design the robot to do this type of tasks, you have to give a consideration in some of the robot's mechanical properties, and one of them is links length. Selecting the appropriate links length leads to the minimum energy consumption due to motors motion. This also leads to smooth motion of the end-effector.

4 Problem Solving Methodology

4.1 Data Used in Analysis

As mentioned before, the end-effector travels from the picking to the placing position in a straight line trajectory. As shown in Figure 4, the picking position is at a distance of 570 mm along X-axis and a distance of 200 mm along Y-axis. The placing positions are varying relative to the carton box. Therefore, the center of the carton box is chosen as the placing position

which is at a distance of -130 mm along X-axis -minus sign indicates the opposite direction- and a distance of 600 mm along Y-axis. The travelling distance of the end-effector is 807 mm. The time in which the end-effector finishes the path is $t = 5$ seconds. The path is divided to points every $dt = 0.1$ second. Therefore, the path has 50 points equally spaced. The Cartesian velocities of the end effector (V_x, V_y) can be calculated as follows:

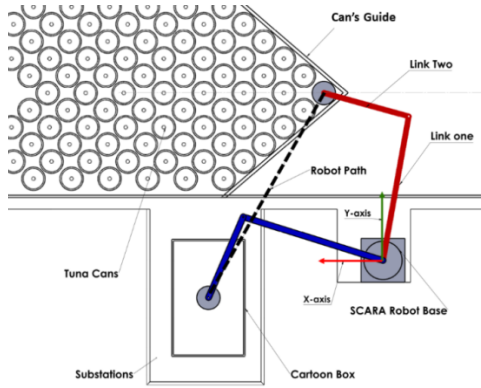


Figure 3: SCARA robot with important dimensions.

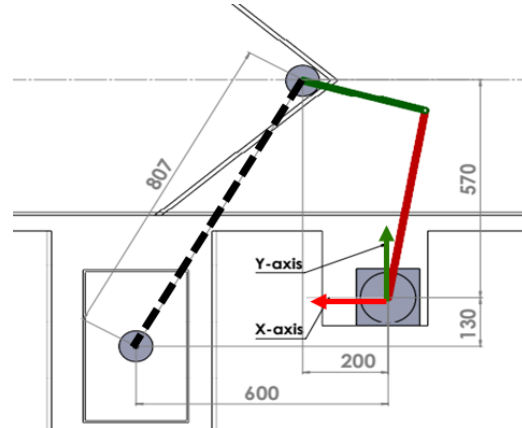


Figure 4: The robot path during the packing processes of the Tuna cans.

$$V_x = \frac{Px_i - Px_{i-1}}{dt} \quad (10)$$

$$V_y = \frac{Py_i - Py_{i-1}}{dt} \quad (11)$$

where i is the point number.

Table 2 displays the Cartesian velocities of the end-effector throughout the entire path. Using Jacobian matrix ($J(\theta)$) and Cartesian velocity ($V \text{ mm/sec}$) the joint velocity ($\dot{\theta} \text{ rad/sec}$) can be determined using Equation (3). As the joint velocity is evaluated, the joint displacement ($\theta \text{ rad}$) and joint acceleration $\ddot{\theta} \text{ rad/second}^2$ can be calculated as follow:

$$\theta_i = \dot{\theta}_i * dt \quad (12)$$

$$\ddot{\theta}_i = \frac{\dot{\theta}_i - \dot{\theta}_{i-1}}{dt} \quad (13)$$

where i is the point number.

The joint displacement (θ), joint velocity ($\dot{\theta}$), and joint acceleration ($\ddot{\theta}$) are used as inputs to the torque equations

(5),(6).

Table 2: The position and velocity of the end-effector.

Point	Time t	P_x	P_y	V_x	V_y
1	0	570	200	-139.97	79.99
2	0.1	556	208	-138.26	80.24
3	0.2	542	216	-138.40	80.34
4	0.3	528	224	-138.49	80.42
5	0.4	514	232	-138.57	80.50
6	0.5	500	240	-138.64	80.59
⋮	⋮	⋮	⋮	⋮	⋮
47	4.6	-88	576	-135.72	81.96
48	4.7	-102	584	-135.72	82.1
49	4.8	-116	592	-135.73	82.26
50	4.9	-130	600	-135.76	82.43

The links in the simple arm has square cross section area with 20mm for each side. These dimensions are determined according to the maximum bending moment in the stress equation using Solidworks. Links lengths vary from (200mm to 600mm) with a step of 50 mm. The total link lengths are $l_1 + l_2 = 800$ mm. There are 9 links length combinations. The total link length is chosen in the way to prevent the arm from getting into a singularity configuration or near it. If the arm is in singularity or near it, the Jacobian matrix has no inverse which makes the arm to move with a high velocity.

4.2 Dynamic Model Analysis Methods

In this paper, three methods are used to solve the dynamic model. These methods are: MATLAB program, MATLAB/SimMechanics and Solidworks motion.

4.2.1 MATLAB Program

MATLAB code is written to program the simple arm equations of motion to get joints torque in each link length combination and to get the joint displacement θ , velocity $\dot{\theta}$, and acceleration $\ddot{\theta}$. These are the inputs of the MATLAB/SimMechanics and Solidworks motion models. The steps used in MATLAB program to solve the dynamic model and visualize the results are illustrated below.

Set equation of motion inputs.

Set time;

Generate straight line trajectory.

Do for all link length combination

Defining l_1 and l_2

Get the joint velocities from Jacobian matrix and Cartesian end-effector velocities.

Get joint displacement and joint acceleration from joint velocity $(\theta, \dot{\theta}, \ddot{\theta})$;

Determine the Torque in each joint motor using torque equations (τ);
 Determine the Power consumed in each joint using power equation (P);
 Display torque and power data Vs time;
 Determine the Energy consumed in each joint using energy equation (E);
 End loop
 Check for link length combination that consumes less Energy.
 End program

4.2.2 MATLAB/SimMechanics Model

As it was mentioned before, the actual designed robotic arm is simplified to a simple 2R robotic arm. The simple robotic arm is designed in Solidworks software. In order to get the MATLAB/SimMechanics model, the CAD model is imported from Solidworks software as an (xml) file to MATLAB/ SimMechanics. This generated a block diagram and 3D model in Simulink environment as shown in Figure 5. All 9 link length combinations are designed in Solidworks and imported to MATLAB/ SimMechanics.

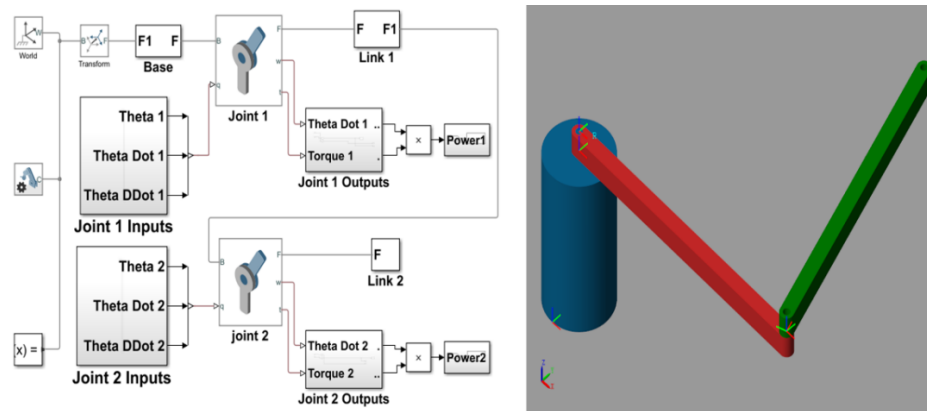


Figure 5: The MATLAB/SimMechanics block diagram and 3D model of the simple arm in the Mechanic Explorer.

4.2.3 Solidworks motion study

Solidwork motion is an add-on to the solidworks software. Solidworks motion is used to study and analyze the simple arm motion. Motion study for the 9 simple arm CAD models are carried out. These studies are generated by attaching a rotary motor for each joint. Data point file of joint displacement θ and time was imported to the joints motors. The outputs of the motion study are kinematic and dynamics results such as: Displacement, velocity, acceleration, forces, and power. Also, it provides animation for the motion study as shown in Figure 6. The results were exported from graphs as a (CSV) file for the 9 links length combinations.

5 Results and Discussion

In this work, three methods of solving the Kinematics and dynamic models are presented: MATLAB, MATLAB/SimMechanics and Solidworks motion. These three methods were used

to calculate joint velocity $\dot{\theta}$, torque τ and power consumption P for each joint. It means that, 18 different result have been analyzed for each links length combination.

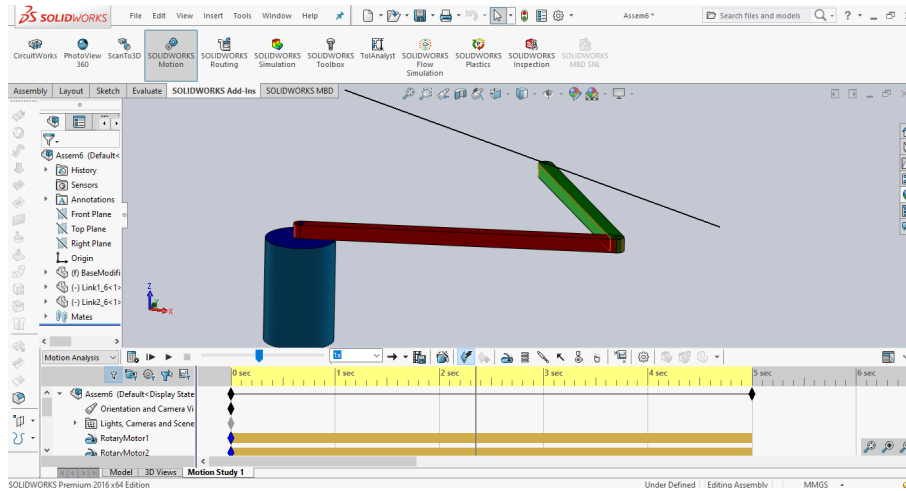


Figure 6: Simple Arm in Solidworks Motion study set.

Due to space limitation, the only sixth links length combination results will be demonstrated. Using MATLAB program, the joint angular velocity $\dot{\theta}$ was calculated for each joint. Then Joint displacement θ was determined by multiplying $\dot{\theta}$ with time. Figure 7 shows the simple robotic arm movements from the picking position at $t = 0$ second to the placing position at $t = 5$ seconds

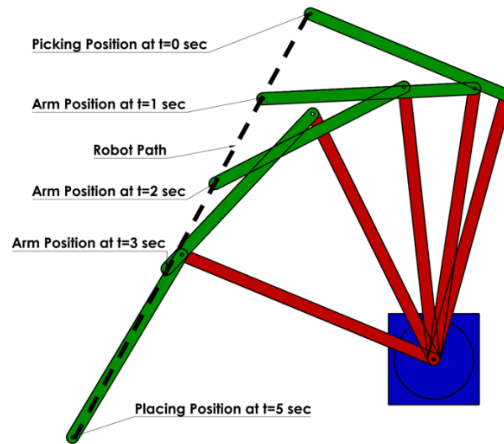


Figure 7: Time sequence of the simple robotic arm.

Figure 8 shows the joint velocity for both of the simple arm joints at the sixth links length combination. It shows that, there is a very small difference between the two joint velocities. For joint one the difference disappears as time progresses. The MATLAB/SimMechanics joint angular velocity is the same as MATLAB program. Figure 9 shows the total power consumed by the two simple arm joints' motors in the sixth links length combination. The total power consumed in the two joints was computed by integrating the joint torque with the joint angular velocity as it was stated in equation (8). Figure 10 demonstrates the total power consumed in

each link length combination for whole period of time. To obtain the total power for each link length combination, the power consumed was numerically integrated for each method results. Three power consumed values corresponding to the three methods were obtained for each link length combination. It shows clearly that the sixth link length combination has the minimum power consumption. It means that the optimal links length are $l_1 = 450 \text{ mm}$ and $l_2 = 350 \text{ mm}$.

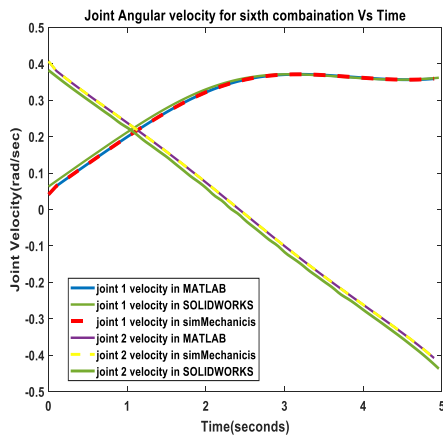


Figure 8: Joint velocity Vs Time

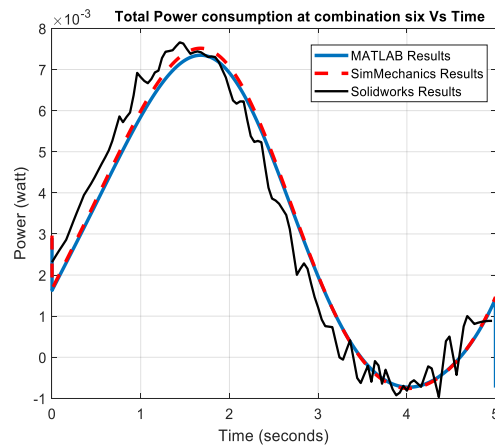


Figure 9: Power consumption by each joint for the sixth combination

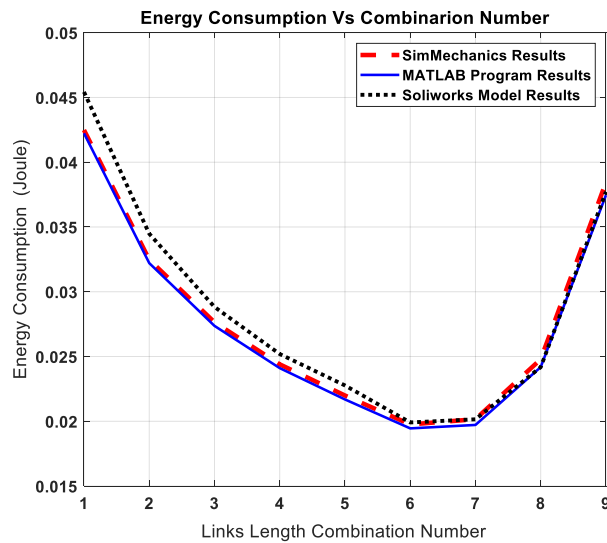


Figure 10: Energy consumption at each combination

6 Conclusion

In this paper, the optimal links length that give a minimum power consumption are determined for customized SCARA robot. Nine link length combinations are tested and simulated. This research is a part of a project of designing a robotic arm for packing task in tuna factory. The robotic arm end effector has to travel mainly in straight line for about 800 mm to pick up and

place the Tuna-Cans. The kinematic and dynamic models are built for the 2R robotic arm. The results of kinematic model which are angular displacement θ and angular velocity $\dot{\theta}$ and angular acceleration $\ddot{\theta}$ for each joint are determined and exported to the dynamic model to obtain the torque and power consumed. Three method which are MATLAB program, and MATLAB/SimMechanics and Solidworks are used to simulate and analyze the dynamic of the robotic arm. The power consumed for each link length combination using the three methods is calculated. From the results, it can be concluded that the sixth link length combination has the minimum power consumption.

References

- [1] M. Shariatee, A. Akbarzadeh, A. Mousavi, and S. Alimardani, "Design of an economical SCARA robot for industrial applications," in *2014 Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, 2014, pp. 534-539.
- [2] S. Y. Nof, *Handbook of industrial robotics*: John Wiley & Sons, 1999.
- [3] J. Fang and W. Li, "Four degrees of freedom SCARA robot kinematics modeling and simulation analysis," *International Journal of Computer, Consumer and Control*, vol. 2, pp. 20-27, 2013.
- [4] M. S. Alshamasin, F. Ionescu, and R. T. Al-Kasasbeh, "Kinematic modeling and simulation of a scara robot by using solid dynamics and verification by matlab/simulink," *European Journal of Scientific Research*, vol. 37, pp. 388-405, 2009.
- [5] V. AVCHAT, R. KALE, J. KODAM, and R. KOKATE, "DESIGN AND ANALYSIS OF 3-AXIS SCARA ROBOT TO SUSTAIN 60 Nm TORQUE BY NUTRUNNER," 2019.
- [6] T. E. Elaikh, H. J. Abed, K. M. Abed, and S. M. Swadi, "Vibration and kinematic analysis of scara robot structure," *Diyala Journal of Engineering Sciences*, vol. 6, pp. 127-143, 2013.
- [7] S. M. Mariappan and A. Veerabathiran, "Modelling and simulation of multi spindle drilling redundant SCARA robot using SolidWorks and MATLAB/SimMechanics," *Revista Facultad de Ingenieria Universidad de Antioquia*, pp. 63-72, 2016.
- [8] B. Fernini, "<Dynamic Behavior of a SCARA Robot by using N-E_met.pdf>."
- [9] M. Gouasmi, M. Ouali, B. Fernini, and M. h. Meghatria, "Kinematic modelling and simulation of a 2-R robot using solidworks and verification by MATLAB/Simulink," *International Journal of Advanced Robotic Systems*, vol. 9, p. 245, 2012.
- [10] P. Subhashini and K. Chinmayi, "Parametric Optimization of Link Lengths of A SCARA Robot for Deburring Of Rectangular Paths," *Materials Today: Proceedings*, vol. 5, pp. 11498-11506, 2018.
- [11] P. Subhashini, N. Raju, and G. V. Rao, "Parametric optimization of link lengths of a SCARA robot for deburring of circular paths," in *CAD/CAM, Robotics and Factories of the Future*, ed: Springer, 2016, pp. 127-135.
- [12] J. Craig, "Introduction to Robotics," 2018.