

MODELLING AND SIMULATION OF 5-DOF ROBOT USING SIMMECHANICS

Essam Z. Fadhel^{1*}, Ameen M. Al Juboori², Mustafa Turki hussein³

¹University of Babylon, College of Engineering, Department of Mechanical Engineering, Hillah, Iraq

²Al-Mustaqbal University College, Biomedical Engineering Department, Hillah, Iraq

³University of Babylon, College of Engineering, Department of Mechanical Engineering, Hillah, Iraq

The main idea of this paper is to present the modeling and simulation of an individual 5DOF degrees of freedom robotic arm. The main axis of the modeling design will be up to the branch of science that makes up Mechatronics. The methods that have been used for the 3D model construction are presented as well as the methods for the simulation. The kinematics problem for the robot system under study is elaborated. Both forward and inverse kinematics is addressed in detail. The CAD model design is also presented. The model parameters are defined for the 3D model and the parameters are used as a base to diverge for the real system. thanks to CAD model translation to SimMechanics model, there is no need for the dynamic equations of motion derivation. The model in SimMechanics is adjusted properly for the trajectory tracking simulation. Several simulations are presented to validate the kinematics solution and the dynamics requirements for the motion. The results presented in the article shows the effectivity of the new softwares in modeling and control of engineering systems.

Key words: robot arm, manipulator, kinematics, cad model, simmechanics

INTRODUCTION

Electronics and automation are moving in today's world. It started with smartphones, continues with smartwatches, cars, houses, but also smart factories or whole cities. Sometimes it is automation purely at the level of software in the form of artificial intelligence and machine learning, other times it is an intervention in the real world such as automated production or control of a car and locking of houses. All these examples meet an important condition that the concept of automation includes, namely automatic system. Such systems perform their function with minimal or none the proportion of the human factor. But in spite of everything, this is more of a trend in the industry and not for granted. It is already dominant in most productions today; in others it is a dominant element of mechanization. Productions without these elements are rather exceptions. Mechanization includes methods that make people's work easier, such as a pneumatic press hammer, directly controlled handling equipment, etc. Automation is a term for automatic systems that do not need much human factor to run. Rather, man regulates and controls the work of the production machine. If necessary, the needs of the decision are decided by the person. In recent years, the term robot has been increasingly talked about. Thanks to its versatility, robotic are the most widely used in Industry. It is possible to change it by simply changing the work tool and the control program on a device with a different function. The different number of degrees of freedom then determines both mobility and access to hard-to-reach places, as well as its price. With sensor connection and Thanks to the design, intelligent control units can also handle considerable variability, e.g. machining more product variants on one line

(saving resources on machines and space). The method of simulation for an adaptive controller was suggested for the system of a human robot based on Solidworks, ADAMS and Simulink joint simulation [1]. A complicated arithmetical modeling process was avoided using this method; also the true time for the Simulink of the moving simulating function was exerted in enough modes. The adopted method was depended upon the building and analysis of the human robot mode. The design of adaptive controller and path tracking was progress depending on this method. The fitting curve method of the least square method was used to evaluate path tracking. The performance of this robot was developed during the comparative analysis. The investigation of the kinematics and dynamics of the active needle was achieved by [2]. Using the flexible swimming wave moving trajectory, the best needle insertion trajectory was determined by using the method of minimum energy. The method assumes that the surrounding tissue will receive smaller amount of energy due to insertion trajectory, which gave less tissue damage. Using a computer aided to develop the model which designed for the active needle by Solidworks. The results agreed with the suggested needle insertion trajectory concluded from the trajectory planning algorithm. By using multi actuators for forward and swimming wave motions to motorize the active needle. Comparison between two robot statuses for the same path with equal length of time was established by [3], and determined a code of computing to calculate the parameters of the kinematic and dynamic. The verification of the theory and the robot movement simulation was used by SolidWorks and MATLAB/Simulink software. Fedák et al. [4] was presented an implementation of active method of the robot technique and its movable simulation by MATLAB

*eng.osam.zuhair@uobabylon.edu.iq

software. Also, a Graphical User Interface model of the robot design had been sophisticated to employ various virtual experiments, therefore, it could be getting various outputs: angles, force and torque. The Graphical User Interface could be getting the inverse and direct kinematics tasks. Youcef et al. [5] presented a simulation of manufacturing robot manipulator Kawsaki FS03N with six degrees of freedom and the graphic Human Machine interaction with 3-dimensions modeling. It developed a direct and inverse geometric model with kinematics model. A movement robot model was sophisticated with SolidWorks and MatLab SimMechanics techniques with a connection between them. The purpose of this study was to test various control kinds before applying to the real robot. The virtual simulation system of manipulator was built by [6], which improved the design efficiency of manipulator control system. Firstly, the 3D model of the manipulator was established in Solidworks, and the industrial six degrees of freedom manipulator was taken as the research object, the Solidworks model was introduced into Simulink, and the joint simulation control system of the manipulator was built by the SimMechanics module in MATLAB/SIMULINK, and the model of the manipulator was added in the motion simulation platform. The kinematic and dynamic analysis of the manipulator was carried out by adding the terminal pose signal and the joint signal (angle function and torque value). The CAD model was designed considering the kinematic and dynamic position of the robot system [7]; also, the relevant material characteristics are selected using the CAD software. This CAD model was used to refine a mathematical model that could be used to implement the control laws and appropriate controllers of the system. Furthermore, the CAD Model is used to create a physical model as a result of translating the CAD model to MATLAB/Simulink (Simscape/SimMechanics library) to simulate the system works. When comparing the CAD physical model with the mathematical model, the same result was obtained. The main advantage of this modeling process was the different approaches in determining the physical 3D model (SimMechanics). The model developed can be used for a multipurpose simulation process which represents a valuable technique for robot system modeling. The direct and inverse kinematics of the robot manipulator that had been created with SolidWorks to obtain the 3dimension model and utilizing Simscape Multibody Link first generation to sending the CAD file to obtain the XML and STL file format was solved by [8]. Sophisticate the Graphical User Interfaces utilizing MATLAB GUIDE to control and find the robot kinematics value. A graphic of the robot could be drawn with all the dimensions labeled. The result of Homogeneous Transformation Matrix would give the position and orientation of the end effector. Besides, inverse kinematics solving would also carry out for two assumed points.

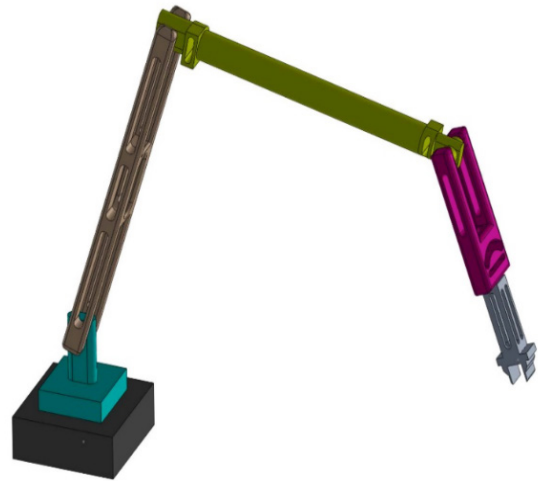


Figure 1: Robot CAD

In this work, a CAD model of 5dof robot system is designed. The forward and inverse kinematics for this system model is solved. For the dynamic motion simulation, the Model is translated to SimMechanics software in order to avoid the derivation of the dynamic equations. Different end-effector trajectory tracking paths are used to ensure the inverse kinematics solution effectiveness. Furthermore, the SimMechanics simulations provided the full dynamic requirements for the robot system in the real world, the required torques for the motion, velocity and accelerations of the joints are induced.

KINEMATICS MODEL

In this article, the robot shown in figure 1, has 5 rotational joints and a grip. For the sake of simplicity, the CAD Model is designed as shown in the figure only to show the rotational joint and make the kinematics analysis clear. Furthermore, the CAD model later transferred into SimMechanics program to simulate the dynamics of the robot without using the traditional dynamic equations. In this section the kinematics problem is detailed and solved.

FOREWORD KINEMATICS

The direct kinematics problem is based on the knowledge of the joint coordinates and searches for the position of the zero-end point. In practice, it is usually not possible to directly measure the position of the end-effector often due to the inability to place sufficiently accurate sensors. A direct kinematic problem can be described:

$$P = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = f(\theta)$$

Which represents the Direct kinematics problem. The so-called homogeneous coordinates can be used to describe the overall position of coordinate systems. The introduction of homogeneous coordinates is related to the problem of geometric projection. Basically, it consists of a rotation matrix and a translation vector supplemented by a zero and a unit vector. Transformation matrix This notation is used by other methods for solving geometric projections. The best-known convention used to describe the geometry of serial robots is Denavit-Hartenberg Convention [9]. In the case of a pair of solids link $i-1$ and link i , which are connected by a joint with one degree of freedom in this case, it is a rotational constraint. In practice, however, both cases of kinematic pairs (translation and rotation) can occur. The definition of the $T_i = O_i - [x_i \ y_i \ z_i]$, knowing the $T_{i-1} = O_{i-1} - [x_{i-1} \ y_{i-1} \ z_{i-1}]$, by the Denavit-Hartenberg convention, is as follows:

$$T_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \cos(\alpha_i)\sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i) & -d_i * \sin(\alpha_i) \\ \sin(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\cos(\theta_i) & \cos(\alpha_i) & d_i * \cos(\alpha_i) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

a = distance in the x, y plane, d = distance in the z direction.

Table 1: DH-Parameters of the Robot

i	α_i	a_{i-1}	d_i	θ_i
1	0	0	d_1	θ_1
2	90	0	0	θ_2
3	0	a_3	0	θ_3
4	0	a_4	0	θ_4
5	-90	0	d_5	θ_5
6	0	0	0	Gripper

According to the parameters in table 1. the forward kinematics Eqs. can be written as

$$T_0^1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_1^2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & a_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^4 = \begin{bmatrix} \sin\theta_4 & \cos\theta_4 & 0 & a_4 \\ -\cos\theta_4 & \sin\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^5 = \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ -\sin\theta_5 & -\cos\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_0^5 = T_0^1 \times T_1^2 \times T_2^3 \times T_3^4 \times T_4^5 = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}$$

Here the A_{ij} inside the overall kinematics matrix is expressed as:

$$A_{11} = \cos\theta_1 \sin(\theta_2 + \theta_3 + \theta_4) \cos\theta_5 - \sin\theta_1 \sin\theta_5$$

$$A_{12} = -\cos\theta_1 \sin(\theta_2 + \theta_3 + \theta_4) \sin\theta_5 - \sin\theta_1 \cos\theta_5$$

$$A_{13} = \cos\theta_1 \cos(\theta_2 + \theta_3 + \theta_4)$$

$$A_{14} = \cos\theta_1 [d_5 \cos(\theta_2 + \theta_3 + \theta_4) + a_3 \cos\theta_2 + a_4 \cos(\theta_2 + \theta_3)]$$

$$A_{21} = \sin\theta_1 \sin(\theta_2 + \theta_3 + \theta_4) \cos\theta_5 + \cos\theta_1 \sin\theta_5$$

$$A_{22} = -\sin\theta_1 \sin(\theta_2 + \theta_3 + \theta_4) \sin\theta_5 + \cos\theta_1 \cos\theta_5$$

$$A_{23} = \sin\theta_1 \cos(\theta_2 + \theta_3 + \theta_4)$$

$$A_{24} = \sin\theta_1 [d_5 \cos(\theta_2 + \theta_3 + \theta_4) + a_3 \cos\theta_2 + a_4 \cos(\theta_2 + \theta_3)]$$

$$A_{31} = -\cos(\theta_2 + \theta_3 + \theta_4) \cos\theta_5$$

$$A_{32} = \cos(\theta_2 + \theta_3 + \theta_4) \sin\theta_5$$

$$A_{33} = \sin(\theta_2 + \theta_3 + \theta_4)$$

$$A_{34} = d_1 + a_3 \sin\theta_2 + d_5 \sin(\theta_2 + \theta_3 + \theta_4) + a_4 \sin(\theta_2 + \theta_3)$$

$$A_{41} = A_{42} = A_{43} = 0$$

$$A_{44} = 1$$

The end effector position vector concluded from the Eq. 3 as:

$$P = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} A_{14} \\ A_{24} \\ A_{34} \end{bmatrix}$$

It is clear from Eq. 4, that the end-effector position does not depend on the θ_5 . Which simplifies the problem of inverse kinematics.

INVERSE KINEMATICS

Inverse kinematics is a method for determining the configuration of a robot (i.e. position and orientation of all joints) for a given resultant coordinate in the workspace [9]. Thanks to this method, the target motion coordinates in the form (x, y, z) can be input to the interface. The software application or the robot control directly then retrieves the coordinates using an algorithm inverse kinematics algorithm that converts the coordinates into the necessary configuration of the individual robot parts. In addition to determining the position of the robot in operation, the method can be used to find the working the robot's workspace or its constraints, e.g. due to an obstacle, different height levels of the base robot and the working area, etc. In addition, gripping possibilities must also be considered the load. The load can take on different angles to the base or rotations and their combinations. The variation of the gripping angle and rotation is determined by the number of degrees of freedom, the configuration of the joints and also the actual position of the robot - some joints used to achieve position of the load cannot be used to achieve the orientation of the load [10]. To solve the inverse kinematics, the dimensions of the different parts of the robot need to be known, the maximum possible deflections of the joints and, above all, the axis of their rotation/displacement. It is also useful to create a simplified model of the kinematic chain. These consist of individual members (arm) and links (kinematic pair, joint). Industrial robots use only two kinds of kinematic pairs. These are the linkages Rotary (R) and sliding (T) [10]. For inverse kinematics tasks, nowadays, specialized programs for its simulation or it is possible to use directly modeling applications, in which robots are designed and use an accurate model to track collisions of themselves at the points of coupling. In the case of computing, the algorithm for computing inverse kinematics is calculated automatically from the dimensions and number of terms and from the types and number of kinematic pairs. To determine the inverse kinematics in this work, the simplification by transferring the individual terms and constraints into 2D space for manual calculation. In these resulting formulas of inverse kinematics will be used in the application to control the implemented robotic arm and in the arm control simulation.

INVERSE KINEMATICS OF THE ROBOT MODEL

The robotic arm model solved and implemented in this paper is a 5 DOF robot due to its 5 joints. All these degrees of freedom (kinematic twos) are of rotational type, none of the kinematic pairs is sliding. Half of them have a rotary axis parallel to the axis of the arm and the other three are perpendicular to it. To determine the position of anybody in 3D space, the solution of

$$\begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} = f(P)$$

here, n represents the number of DOF of the robot system. Most of the calculations are based on a two-dimensional representation of the individual kinematic pairs as two sides of triangles. The triangles are folded in various ways and through the partial angles we try to get the final value of the angle of the two mutually placed kinematic pairs. For the calculations, the global coordinates/axes are denoted as follows: - x - right/left, y - forward/backward, z - up/down

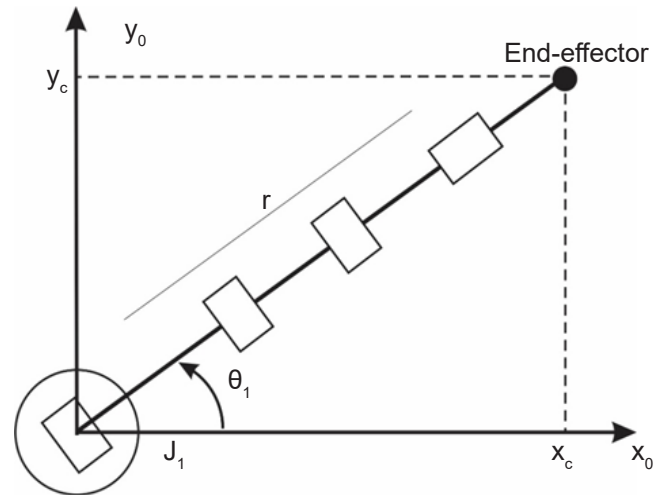


Figure 2: Top View of Robot Kinematics Model

The kinematic of J_1 is the first in order from the base, as shown in figure 2. Its axis is the only one invariant in any configuration and is parallel to the z-axis. orientation of the arm parallel to the y-axis (straight ahead). The calculation of the configuration of this link is simple as it does not depend on other constraints or point distance. It is sufficient to use the x, y coordinates as the dimensions of the axes of the right triangle and calculate the deflection according to formula:

$$\theta_1 = a \tan 2(x_c, y_c)$$

where θ_1 is the rotation angle (the rotation direction is determined by the sign of the x-coordinate). The rest of the joint angles can be calculated using the robot system planar view shown in figure 3. First the angle ψ assumed to be as $\psi = \theta_2 + \theta_3 + \theta_4$, will help us in other operations - it simplifies the determines of the angles. The angle θ_3 can be calculated as follows, using the equation (4), summing the $d_x^2 + d_y^2$ and taking the square root of the equation yields:

$$a_3 \cos \theta_2 + a_4 \cos(\theta_2 + \theta_3) = \pm \sqrt{d_x^2 + d_y^2} - d_5 \cos \psi$$

And by rearranging Eq. d_z as:

$$d_z - d_1 - d_5 \sin(\psi) = a_3 \sin \theta_2 + a_4 \sin(\theta_2 + \theta_3)$$

From equations (7&8)

$$\cos\theta_3 = \frac{(d_z - d_5 \sin(\psi) - d_1)^2 + (\pm\sqrt{d_x^2 + d_y^2} - d_5 \cos(\psi))^2 - a_3^2 - a_4^2}{2a_3a_4}$$

$$\sin\theta_3 = \pm\sqrt{1 - (\cos\theta_3)^2}$$

$$\theta_3 = a \tan 2(\sin\theta_3, \cos\theta_3)$$

From figure 3 also, α & β angles can be expressed as:

$$\alpha = a \tan 2(d_z - d_5 \sin(\psi) - d_1, \pm\sqrt{d_x^2 + d_y^2} - d_5 \cos(\psi))$$

$$\beta = a \tan 2(a_4 \sin\theta_3, a_3 + a_4 \cos\theta_3)$$

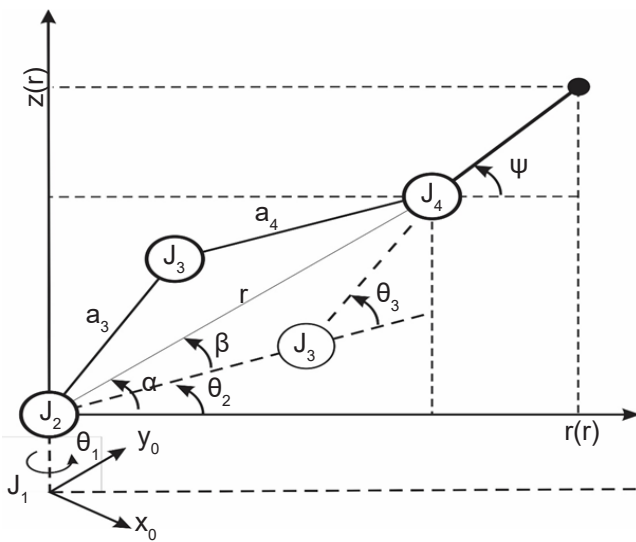


Figure 3: Planar View of Robot Kinematics Model

$$\theta_2 = \alpha - \beta$$

The joint angle of rotation Θ_4 is determined from ψ

$$\theta_\psi = \theta_2 + \theta_3 + \theta_4 \rightarrow \theta_4 = \theta_\psi - \theta_2 - \theta_3$$

Finally, the joint angle Θ_5 by using total transformation matrix in equation (3):

$$s_5 = s_1 r_{11} - c_1 r_{21}$$

$$c_5 = s_1 r_{12} - c_1 r_{22}$$

$$r_{11} = ((c_1 c_2 c_3 - c_1 s_2 s_3) c_4 + (-c_1 c_2 s_3 - c_1 s_2 c_3) s_4) c_5 - s_1 s_5$$

$$r_{12} = -((c_1 c_2 c_3 - c_1 s_2 s_3) c_4 + (-c_1 c_2 s_3 - c_1 s_2 c_3) s_4) s_5 - s_1 c_5$$

$$r_{21} = ((s_1 c_2 c_3 - s_1 s_2 s_3) c_4 + (-s_1 c_2 s_3 - s_1 s_2 c_3) s_4) c_5 - c_1 s_5$$

$$r_{22} = -((s_1 c_2 c_3 - s_1 s_2 s_3) c_4 + (-s_1 c_2 s_3 - s_1 s_2 c_3) s_4) s_5 - c_1 c_5$$

$$\theta_5 = a \tan 2(s_5, c_5)$$

Where, $s_i = \sin(\Theta_i)$ and $c_i = \cos(\Theta_i)$.

CAD ROBOT MODEL

The 3D CAD program SolidWorks is a tool for modelling engineering systems, it has a wide range of features for creating CAD models for a variety of disciplines [11]. This tool allows the engineers to create three main document types. The first type of document is parts, which represent the individual parts of a mechanism, and the second type of document is assemblies, which group individual parts into a functional mechanism - an assembly, and the third document type is drawings, which represent drawings of parts and assemblies necessary for their manufacture. SolidWorks applications are currently one of the most widely used CAD tools and can provide a range of additional functions using the so-called "add-on modules". The power of the CAD models lies in the ability to transfer these models into another form of models in another software. These translations of the models serve the purpose of simulation or printing process. The material properties and the inertia parameters are determined automatically through CAD model material selection. And these properties can be transferred also. So, in this case the CAD model provides several benefits over the traditional methods of simulations.

MATLAB SIMULINK SIMMECHANICS

The MATLAB development program is designed for complex modelling mechatronic systems and provides a large number of computational tools for the implementation of control and identification of systems. It also includes the graphical programming environment Simulink for creating control algorithms and for solving the dynamics of controlled systems. The SimMechanics simulation program is part of the MatLab/Simulink family of programs. This entire family of programs is designed for scientific and engineering calculations based on matrix-based calculations [12]. Nowadays, it provides many specific toolboxes containing specific functions for solving problems of numerical computation and algorithm development in these areas. The core of MatLab itself provides the numerical and programming apparatus for performing various numerical computations. In cooperation with this kernel, it is then possible to implement various dynamic processes in the Simulink environment, providing a block-based graphical interface [13]. A dedicated toolbox used in this work for mechanical dynamic calculations and modelling of mechanical systems is called SimMechanics. SimMechanics extends the capabilities of Simulink by means for modelling and simulation of mechanical devices and their control systems. This software contains libraries of simulation blocks that correspond to real components of real mechanical systems. Several blocks representing various solids, joint connections, motion screws, dampers, springs, sensors, actuators, etc can be found in the library. These blocks can be used to intuitively create graphical representations of complex mechanical sys-

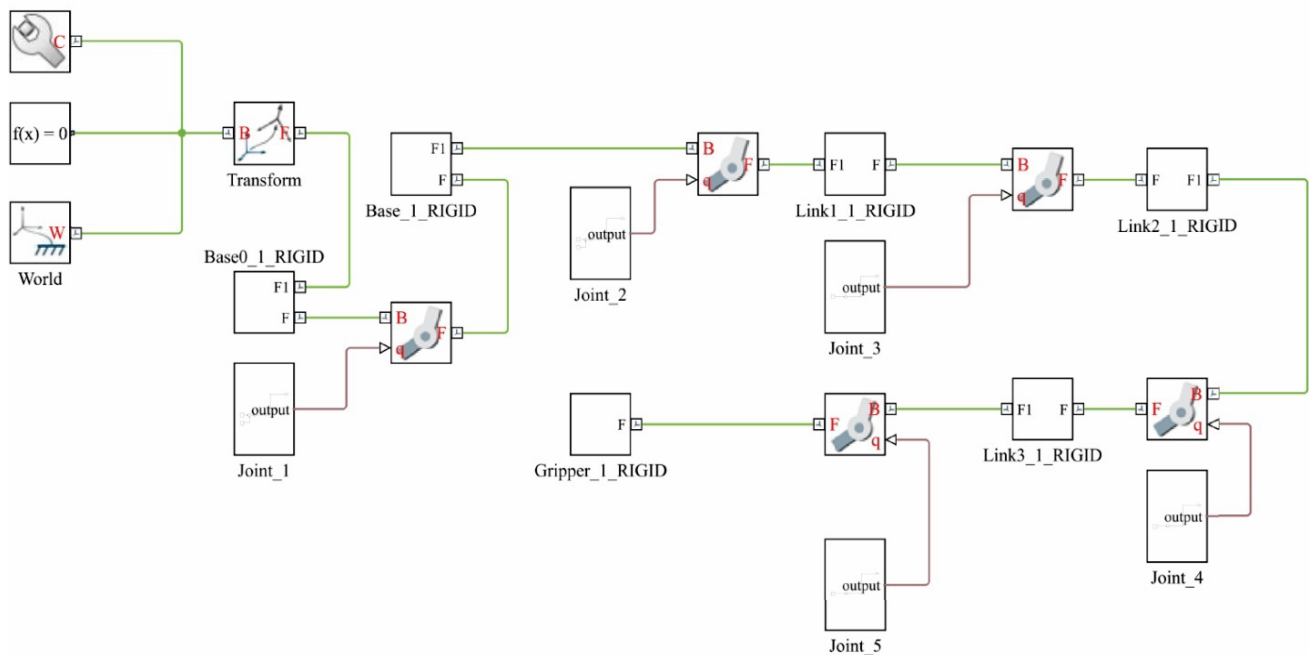


Figure 4: The SimMechanics Model of the Robot

tems (Figure 4), connect them to the environment and control systems also modelled by Simulink tools. In this way, it is possible to simulate the interaction of mechanical parts and them control units and the effect of the whole system on the surrounding environment and vice versa [14]. SimMechanics offers kinematics and dynamic analysis of mechanical systems. The SimMechanics offers the ability to import models from CAD softwares for better and easier method for simulation, which is used in this work. The model exported from solidworks software to SimMechanics. And the simulation is done to calculate the joint torques required for the motion.

SIMULATION RESULTS

The simulation presented in this works used the CAD model and the simulation software for dynamic motion simulation of the robot system. The main idea of the simulation includes the solution of the inverse kinematics problem and dynamic simulation depending on CAD model. The first step in the simulation is the suggesting paths for the end-effector to follow during the simulation. The paths suggested in this work are listed in Table 2. Three paths are suggested the first is linear path, second is circular path, and the third is helical path. The equations of the paths are described in table. 2. The paths used for simulation are shown in figure 5. For the linear path shown in figure 5-(a), the solution of the inverse kinematics is shown in figure 6, the results are calculated without θ_5 because this joint angle does not affect the End-effector position. The solution of inverse kinematics was used to find the required joint angles to follow path perfectly with very small error around 10^{-5} . For the circular path shown in figure 5-(b), the joint angles calculated from inverse kinematics procedure are used in SimMechanics for dynamic simulation. The resulted joint

Table. 2: Paths used in the Simulations.

	x	y	z
1 Linear Path	0.2	0.2	[0.18,0.5]
	0.2	[0.2,-0.2]	0.5
	[0.2,-0.2]	-0.2	0.5
2 Circular Path	$\left. \begin{aligned} x &= 0.2 \\ y &= R \sin(\vartheta) \\ z &= R \cos(\vartheta) \end{aligned} \right\} \begin{aligned} R &= 0.2 \\ \vartheta &= [0, 2\pi] \end{aligned}$		
	$\left. \begin{aligned} x &= 0.2 + (R/2) \\ y &= R \sin(\vartheta) \\ z &= R \cos(\vartheta) + 0.15 \end{aligned} \right\} \begin{aligned} R &= 0.02 \\ \vartheta_r &= [0, 2\pi] \\ \vartheta &= [0, 10\pi] \end{aligned}$		
3 Helical Path			

angles and their derivative are shown in figure 7. In this figure, the derivative of joint angles estimated from the simulation inside SimMechanics. These derivatives are used to estimate the dynamic required by the joints in order to move the links to prescribed angles from the inverse kinematics solutions. For the helical path, the inverse kinematics solution joint angles are shown in the figure 8. The joint angles estimated very effectively. These joint angles are used in the SimMechanics simulation to estimate the torques required by the joints. The joint torques calculated in the simulation are shown in figure 9. The notches shown in the curves for torques are due to the coupling dynamics between links. After getting the data points of the paths the inverse kinematics problem is solved to calculate the required joint angles for the End-effector to follow the suggested (desired) path. Later the calculated joint angles are fed to SimMechanics software in order to simulation the imported CAD model using the joint angles for each path.

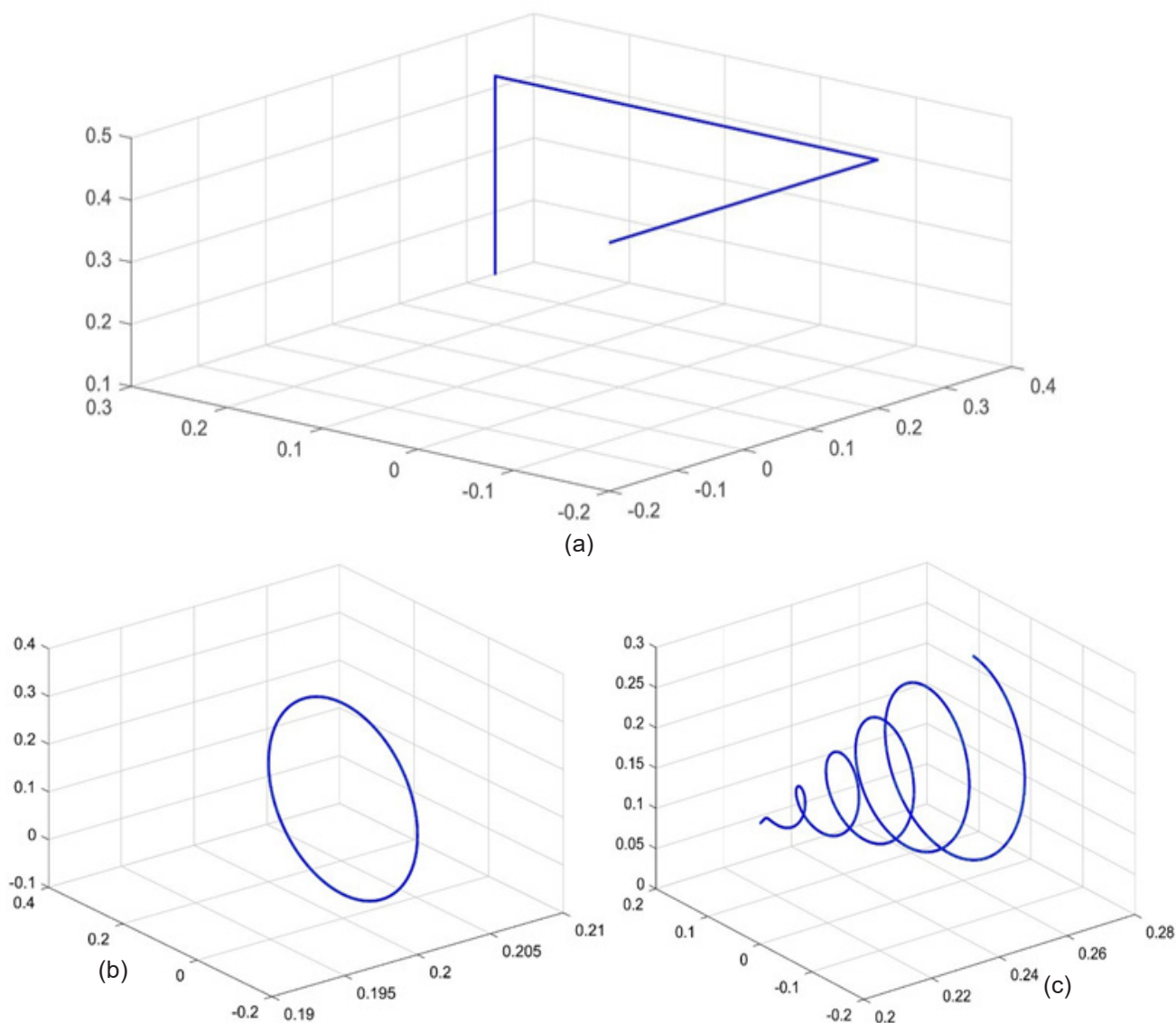


Figure 5: End-Effector Trajectory Paths, (a) Linear Path, (b) Circular Path, (c) Helical Path

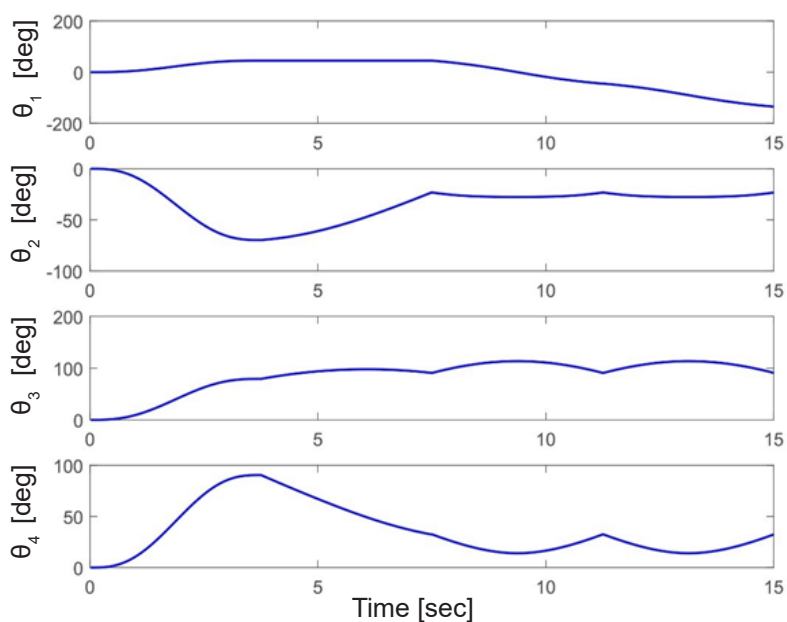


Figure 6: Inverse Kinematics Solution for Linear Path Trajectory

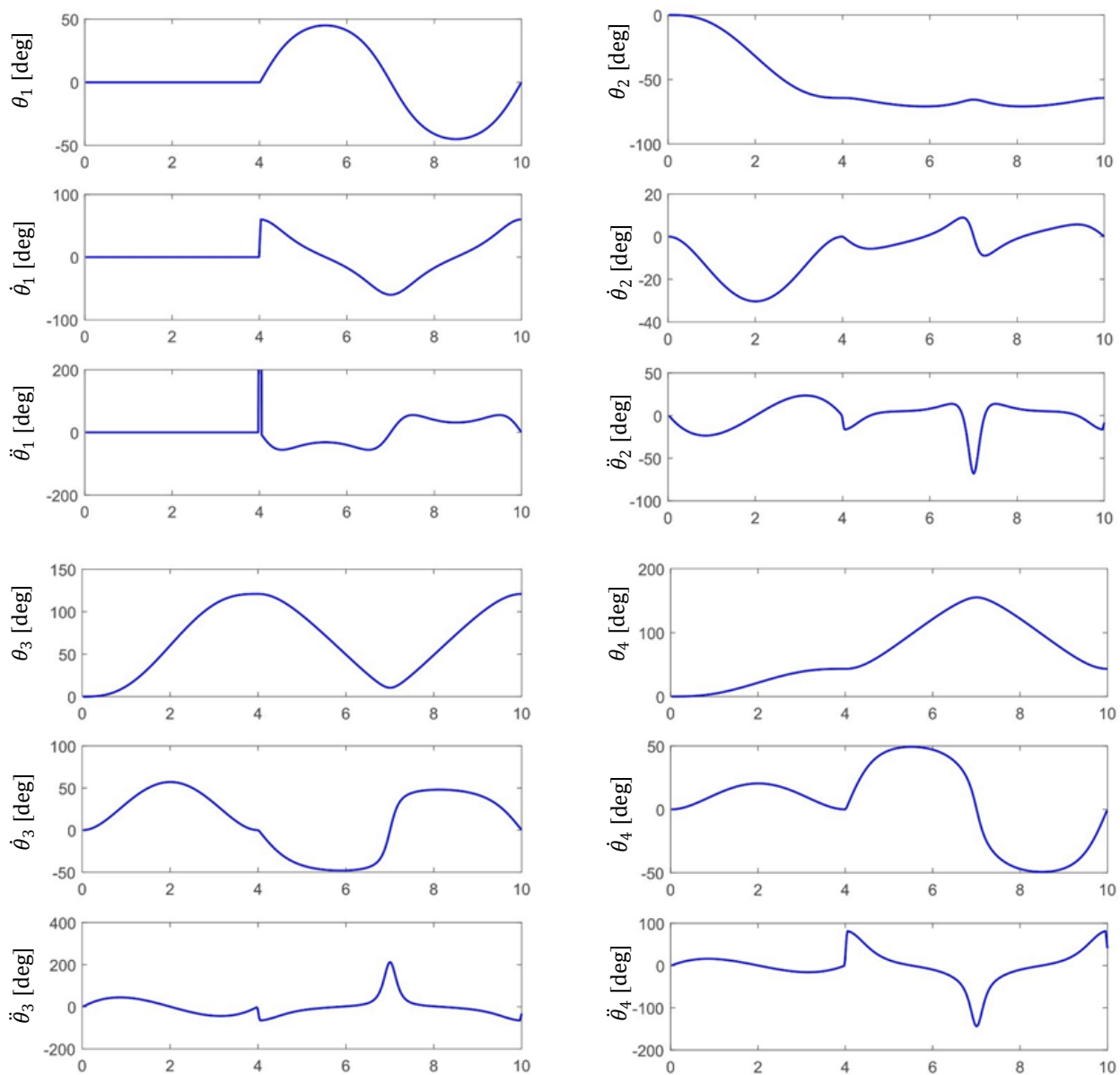


Figure 7: Joint angles simulation for the circular Path in SimMechanics

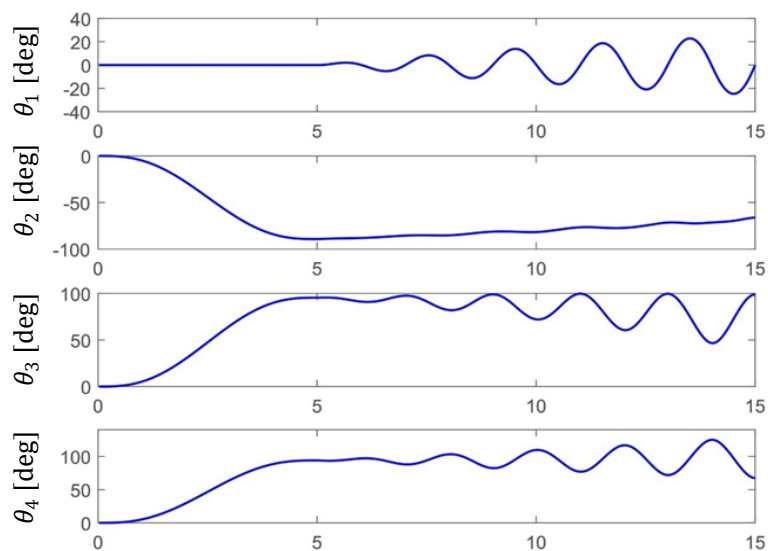


Figure 8: Inverse Kinematics Solution for Linear Path Trajectory

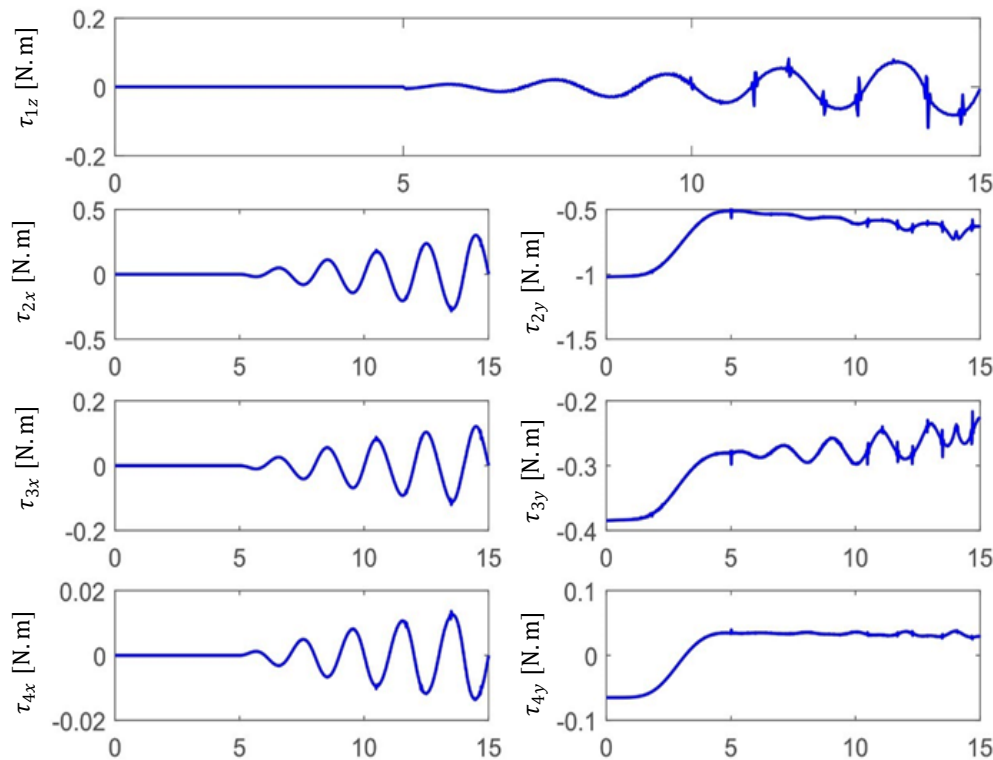


Figure 9: Joint Torque Estimated for the Helical Path simulation

CONCLUSIONS

The main aim of this work is to study and simulate an individual 5DoF robot system. It can be noted that this work can be used for all simulation control work in 3D with real constraints for all industrial robots. The both kinematics problem 'forward and Inverse' for this system are solved for the 5DoF robot. A CAD model is developed using the Solidworks software, and the model translated into an appropriate form for the sake of simulation. The simulation model is built in SimMechanics software. These models are used to simulate the robot in 3D using MatLab and SolidWorks. From the obtained results we noted that the simulations (MatLab based on Solidworks Models) are almost perfectly perform the Realtime behavior of the robot system, which allow researchers to develop further advanced control laws.

REFERENCES

1. Song, D., Zheng, L., Wang, L., Qi, W., Li, Y. (2009). The co-simulation of humanoid robot based on Solidworks, Adams and Simulink. In: Progress in Robotics. FIRA 2009. Communications in Computer and Information Science, vol 44. Springer, Berlin, Heidelberg. DOI:10.1007/978-3-642-03986-7_2
2. Zarghani, N. (2009). Modeling and simulation of an active robotic device for flexible needle insertion. MSc thesis, National University of Singapore.
3. 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. DOI:10.5772/50203
4. Fedák, V., Ďurovský, F., Üveges, R. (2014). Analysis of robotic system motion in SimMechanics and matlab gui environment. MATLAB Applications for the Practical Engineer, Kelly Bennett. IntechOpen, DOI: 10.5772/58371. Available from: <https://www.intechopen.com/chapters/46873>
5. Youcef, Z.Y., Adel, M., Hamza, S. (2015). Dynamic and kinematic simulation of kawasaki manipulator industrial robot using Solidworks and MatLab SimMechanics. Proceedings of the European Modeling and Simulation Symposium, pp. 46-51.
6. Tang, Y., Li, X. (2018). Simulation research of manipulator control system based on Solidworks and SimMechanics. Academic Journal of Computing and Information Science, 1(1); 19-27. DOI: 10.25236/AJCIS.010003.
7. Zatopek, J., Ek, Z., Machado, J.A., & Sousa, J. (2018). Dynamic simulation of the CAD model in SimMechanics with multiple uses. Turkish Journal of Electrical Engineering and Computer Sciences, 26, 1278-1290. DOI:10.3906/elk-1712-217

8. AZHAR, M. W. (2019). Kinematics modelling of robot manipulator using solidworks. MSc thesis, Department of Electrical Engineering Faculty of Engineering, Muhammadiyah University of Surakarta.
9. Craig, J. (2018). Introduction to robotics - mechanics and control. Pearson, USA.
10. Corke, P. (2017) Robotics, Vision and Control Fundamental algorithms in MATLAB. Springer International Publishing, Germany.
11. Kurowski, P. (2021). Engineering analysis with Solidworks simulation. SDC Publications, USA.
12. Dung, L.T., Kang, H., Ro, Y. (2010). Robot manipulator modeling in MatLab-SimMechanics with PD control and online gravity compensation. International Forum on Strategic Technology/ Engineering, pp. 446-449.
13. Russell, K., Shen, Q., Sodhi, R.S. (2018). Kinematics and dynamics of mechanical systems implementation in MATLAB® and SimMechanics (2nd ed.). CRC Press. DOI:10.1201/9780429506253.
14. Gao, Q., liu, J., liu, Y., tian, X., (2019). Research on admittance control simulation based on SimMechanics. International Conference on Informatics, Control and Automation ICA 2019, pp. 155-159.

Paper submitted: 22.08.2021.

Paper accepted: 16.11.2021.

*This is an open access article distributed under the
CC BY 4.0 terms and conditions.*