Kinematic and Dynamic Analysis of an Industrial Six-Axis Robotic Manipulator

Ashwin Misra and Ghanvir Singh

October 22, 2019

Kinematic and Dynamic Analysis of an Industrial Six-Axis Robotic Manipulator

Ashwin Misra[1], Ghanvir Singh[2]
Department of Mechanical Engineering
Delhi Technological University
New Delhi, India
ashwin_bt2k16@dtu.ac.in[1] and

Abstract

Industrial robots are the most widely manufactured and used types of robots in the production sector. To achieve a high degree of performance, various parameters and characteristics of robots should be known. Nowadays various tools are used for simulation, modeling, and analysis of a robot to assist in the enhancement and improvement of the robotic operations. The first objective of this article is to derive the complete (forward and inverse) kinematic and dynamic model of a 6DoF robotic manipulator through both analytical and software-numeric approaches. The second objective is to study the results of the combination of investigative tools across different domains to perform the same analysis such as 3D CAD Modelling, kinematic analysis using Robotic Toolbox in MATLAB and dynamic analysis using Robo-Analyzer. Hence, the novelty of this research lies in plotting a simplified complete analysis for early stage robotic researchers of any robotic manipulator that can be easily derived from the specific manipulator used here for ease of explanation. This can act as a precursor to teach pendant and robotic learning software development.

Industrial robots, Kinematic analysis, Dynamic analysis, Modeling, Denavit-Hartenberg parameters

1 Introduction

Kinematics is the branch of classical mechanics which deals with the position, velocity and relative orientation of points or bodies. In robotics, it is used to describe the motion of multi-link systems such as a 6DoF robotic arm. The parameters of each links such as the link length, twist angle etc are used to set up the governing equations. These equations are non-linear in nature and are used to correlate the joint parameters to the position and configuration of the robot. Denavit and Hartenbe‘rg used a standard methodology to achieve this purpose which uses a mathematical matrix to determine the pose of one link relative to another [1].

Kinematic analysis of robots further consists of analysis of two component models- Forward Kinematic model and Inverse kinematic model. Forward Kinematic model deals with the link properties as inputs and gives us the relative pose of the end-effector of the robotic arm. The inverse kinematic model is the inverse of the forward kinematic model i.e. the pose of the end-effector is known and we find out the link parameters of the robotic arm.

Dynamics is the branch of classical mechanics which deals with the study of the effect of forces on the motion of an object. In dynamic analysis of robots, we will look at dynamics as it relates to acceleration, loads, masses and inertias. To be able to accelerate a mass, we need to exert a force on it. Similarly, to impart an angular acceleration to a mass we need to exert a non-zero torque on it.

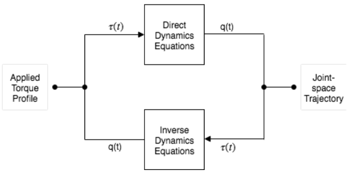


Figure 1: Direct and Inverse Dynamics

The article [2] follow only forward and kinematic analysis of a KUKA KR6 robot and not the dynamic analysis. Also, there is no hint for a method to import CAD model of a robot in analysis software. It is very specific in nature and do not validate their process with numeric software approaches.

Another article [3] follows the method to import CAD Models which are not pre-installed in Robo-analyzer. But this article also covers only the Kinematic analysis and then results in a introduction to the development of a MATLAB code for a teach pendant software. This is again for a lower degree of freedom robotic manipulator and not covers a complete end-to-end analysis procedure.

This article presents a simplified approach to tackle the complete kinematic and dynamic analysis of a robotic manipulator. The authors were motivated due to the lack of articles from an introductory level to approach this problem through a simplified methodology for early stage researchers and students. Due to the presence of lower DoF analysis articles and the lack of those of 6 DoF manipulators with combination of CAD which are widely used in manufacturing industries, This article focuses on a transparent and clear process to understand the CAD import method and analysis of an industrial six axis manipulator which can be used in various applications following the same methodology. An example with a sample position and orientation is also used to enhance the visualization of the process.

2 Analysis

Fanuc S430-IW is an industrial robot manufactured by Fanuc Robotics and designed for a variety of manufacturing and system processes. It is most widely used in welding process due to various favourable factors. Fig.2 represents the allowed motion along and respective axes and Fig.4 represents the co-ordinate systems attached to each of the links The S-430-IW robot has a work envelope that allows processing of large fabrications and high motion speeds to reduce time between welds. It has a payload capacity of 130 kg and total robot weight is 1300kg with reach of 2643mm with Articulated structure. This robot is ideal for fast and precise applications in all environments[4]. Its wide usage and applications in the automation industry led to it's selection with the focus to efficiently model this robot using various platforms. As evident from figure 2, the robotic arm is a serial 6 degree of freedom manipulator with Six revolute joints. The coordinate axes for each of the links are shown with the allowed motion. If a revolute joint is denoted by R. Then -

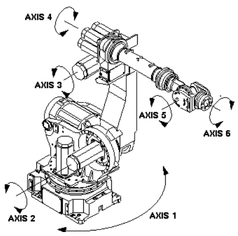


Figure 2: Joints of FANUC S430i.

$R \perp R \parallel R \perp R \perp R \perp R$

The Denavit-Hartenberg (D-H) Model of representation is a very simple way of modeling robot links and joints that can be used for any robot configuration.In Table.1, the D-H parameters for each link are summarized. The D-H parameters are further discussed in-depth in the forward kinematic analysis section. Parameters for Table 1:-

θ = joint angle, d = link length

Table 1: D-H Parameters for FANUC S430- IW

$Link_i$	θ_i	d_i	a_i	α_i
1	θ_1	d_1	a_1	$\pi/2$
2	θ_2	0	a_2	2π
3	θ_3	0	a_3	$\pi/2$
4	θ_4	d_4	0	$\pi/2$
5	θ_5	0	0	$\pi/2$
6	θ_6	d_6	0	2π

a = link offset, α = twist angle

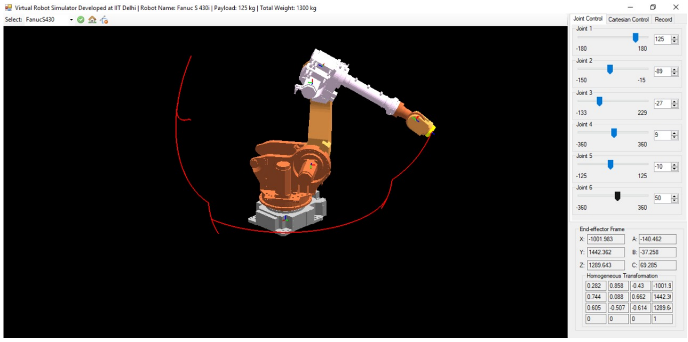
This article uses traditional numerical methods along with software counterparts for validation and better results. The robotic toolbox is used for the Kinematic analysis and Robo-Analyzer is used for the dynamic analysis. The reason behind the usage of two similar soft wares is presenting a more informative and intuitive result to enhance the understanding of the process. For using the MATLAB robotic toolbox/simulink, one must be well versed with MATLAB and Simulink to perform the analysis, whereas Robo-Analyzer presents a user friendly and easy to use analysis method which also visualizes the motion and plots the path of the end-effector. The latter method is more suitable for early stage robotic students and researchers, but due to the wider application and control of the robotic toolbox, it is also used in the article to provide a holistic approach which can be tailored to various levels.

2.1 Computer Aided Design

The CAD Assembly and figures of widely used robotic manipulators such as that of Fanuc, KUKA etc. are present on respective manufacturers website, which can be used and imported to end-user applications such as Virtual Robot module after making necessary changes. If not present, the CAD model can be designed and added to VRM using a Solid Modelling software with the necessary addin. The following generic procedure can be adopted for every robotic manipulator whose CAD is not available through the manufacturer.
Carefully measure all the dimensions of each link and model each part in AutoDesk Inventor and validate the distance between joint axes with the DH table values.Import CAD part of each link in AutoDesk Inventor CAD assembly. Link

0 should be placed such that the base of the assembly is along the XY Plane.Insert constraints between the consecutive links.(revolute joints).At the end effector, a User coordinate system should be placed to represent the axes of the system, the system should be such that the direction of the last joint axis should be parallel with the Z-axis.The tool center point is the origin of the User Coordinate system's origin which should be always placed at the centre of the tool then Launch the inventor addin which reads joint axes data and determines the D-H Parameters One User Coordinate system is added to assembly where the D-H Frames are to be situated also to export the CAD file, these UCS are to be moved within each part file.

Robo-Analyzer expects the input data to be in XML Format. Fanuc S430-IW is a fixed base manipulator with the following CAD model- The assembly contains seven parts and six insert constraints. The insert constraints should be always defined from the base to the end-effector link to conform with the order of the D-H Table.


Virtual Robot Simulator Developed at IIT Delhi | Robot Name: Fanuc S 430i | Payload: 125 kg | Total Weight: 1300 kg

Select: FanucS430

Joint 1: 125 (-180 to 180)
Joint 2: -89 (-150 to -15)
Joint 3: -27 (-133 to 229)
Joint 4: 9 (-360 to 360)
Joint 5: -10 (-125 to 125)
Joint 6: 50 (-360 to 360)

End-effector Frame

X: -1001.983
A: -140.462
Y: 1442.362
B: -37.258
Z: 1289.643
C: 69.285

Homogeneous Transfomation

0.282	0.858	-0.43	-1001.9
0.744	0.088	0.662	1442.3
0.605	-0.507	-0.614	1289.6
0	0	0	1

Figure 3: Imported CAD Model of FANUC S430i in VRS of Robo-Analyzer.

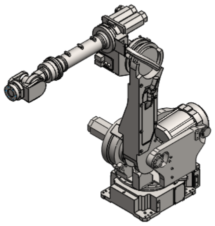


Figure 4: CAD Model of FANUC S430i.

2.2 Forward Kinematics

Calculating the orientation and position of the end-effector of the robot is called forward kinematic analysis. In order to model a forward kinematic analysis for the robot, there is usually a methodology to be followed. Firstly, number the links and joints along with attaching local coordinate reference systems to each links (figure 4) [5] . Now, establish each link's D-H Parameters for every link referencing from figure 6. Using the attached frames (D-H), the four parameters that locate one frame of reference relative to another are defined as:
θ_i (joint angle) is the angle between the x_{i-1} and $x - i$ axes about the z_{i-1} axis; d_i(link offset) is the distance from the origin of frame i -1 to the x_i axis along the z_{i-1} axis; a_i(link length) is the distance between the z_{i-1} and z_i axes along the x_i axis; for intersecting axes is parallel to $z_{i-1}z_i$; α_i(link twist) is the angle between the z_{i-1} and z_i axes about the x_i axis.

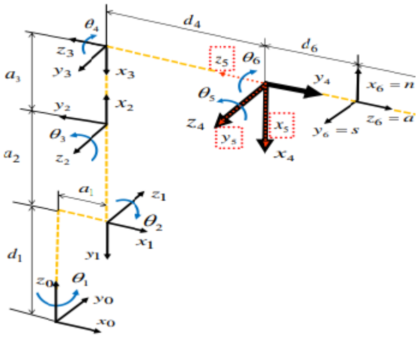


Figure 5: Kinematic model structure

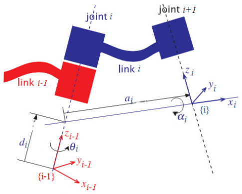


Figure 6: D-H Frame parameterized allocation [6]

Now, homogeneous transformation matrices are calculated for each link and then the kinematic equations of the manipulator are computed. The equivalent homogeneous transformation as a final result of all individual concatenations:-

$$^{i-1}T_i(\theta_i, d_i, a_i, \alpha_i) = R_z(\theta_i).T_z(d_i).T_x(a_i).R_x(\alpha_i) \tag{1}$$

Where $R()$ and $T()$ are the $4x4$ homogeneous transformation matrix for translatory and rotatory motion.

$$^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i.c\alpha_i & s\theta_i.s\alpha_i & a_i.c\theta_i \\ s\theta_i & c\theta_i.c\alpha_i & -c\theta_i.c\alpha_i & a_i.s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where $c\theta_i$ is $cos(\theta_i)$ and $s\theta_i$ is $sin(\theta_i)$
The kinematic equation of a serial robot containing m links is

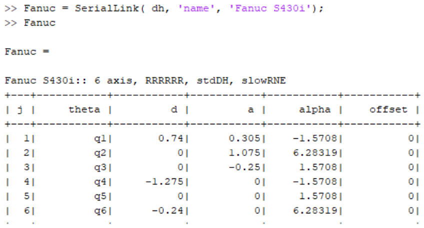
$$^{0}T_n = \prod_{i=1}^{m} {}^{i-1}T_i \tag{3}$$

Where $^{i-1}T_i$ is the homogeneous transformation matrix of frame i relative to the frame i-1

$$\begin{aligned} ^{0}T_6 = & \begin{bmatrix} c_1 & 0 & -s_1 & a_1c_1 \\ s_1 & 0 & c_1 & a_1s_1 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} c_2 & s_2 & 0 & a_2c_2 \\ s_2 & -c_2 & 0 & a_2s_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ & \times \begin{bmatrix} c_3 & 0 & s_3 & a_3c_3 \\ s_3 & 0 & -c_3 & a_3s_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ & \times \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} c_6 & s_6 & 0 & 0 \\ s_6 & -c_6 & 0 & 0 \\ 0 & 0 & -1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ & = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4) \end{aligned}$$

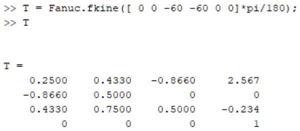
where :- $c_i = cos\theta_i$, $s_i = sin\theta_i$, $a_i =$ offset length, $d_i =$ link length

The Robotics toolbox in MATLAB was used for the software approach to the numerical forward kinematics problem. The Toolbox contains many functions that are useful for the study and simulation of classical arm-type robotic studies such as kinematic analysis, dynamic analysis and trajectory generation. It provides functions for manipulating and converting between data types such as: vectors, homogeneous transformations, roll-pitch-yaw, Euler angles and unit-quaternions which are necessary to represent Three dimensional position and orientation. The Robotic Toolbox is an open-source free software with controls in a straightforward manner which allows for easy understanding.[7] . In order to create a model of Fanuc S430-IW robot using Robotic Toolbox, the D-H parameters from Table.1 have been used as inputs. After creating a simulink model from this code using the integrated function using the D-H Parameters as input denoted in Figure.7, another function is used for a particular momentary position of joints to obtain the uniform transformation matrix corresponding to the last link of the manipulator further shown by Figure 8. The translation matrix produced by this software numerical approach and that obtained previously by the analytical approach will yield the same results. Hence this acts as an exemplified approach which can be generalized for any manipulator.



```
>> Fanuc = SerialLink( dh, 'name', 'Fanuc S430i');
>> Fanuc

Fanuc =

Fanuc S430i:: 6 axis, RRRRRR, stdDH, slowRNE
+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|       0.74|      0.305|    -1.5708|          0|
|  2|         q2|          0|      1.075|    6.28319|          0|
|  3|         q3|          0|      -0.25|     1.5708|          0|
|  4|         q4|     -1.275|          0|    -1.5708|          0|
|  5|         q5|          0|          0|     1.5708|          0|
|  6|         q6|      -0.24|          0|    6.28319|          0|
```

Figure 7: Creating SimuLink Model of the Robot in Robotics Toolbox



```
>> T = Fanuc.fkine([ 0 0 -60 -60 0 0]*pi/180);
>> T


T =
    0.2500    0.4330   -0.8660     2.567
   -0.8660    0.5000         0         0
    0.4330    0.7500    0.5000    -0.234
         0         0         0         1
```

Figure 8: forward kinematics transformation

The momentary position and orientation of the manipulator depending upon the selected parameters is shown in the fig 9 and fig 12. In this section we will discuss different temporal graphs obtained for forward kinematic analysis of Fanuc S430-IW in Robo-Analyzer Simulation Software by creating 6Dof Skeleton robot, with with initial joint values [**0 0 -60 -60 0 0**] and final joint values [**0 0 0 0 0 0**]. As we see in fig 10 a significant change in coordinates and Joint values of only links 3 and 4 are noted . fig 11 is a plot for joints torque (consider only gravitational forces) vs time in which we noticed that torque on joints 1 and

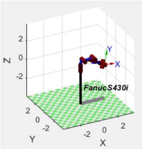


Figure 9: Robot POSE in Forward Kinematic variables

4 is changes with time rapidly during period of 2.4 to 3.4 seconds.similarly like this we can plot graphs for velocity and acceleration of the links From the above discussion we can conclude that much of the change we see only in joint 4 and slightly in joint 3 and in 1 during force analysis on joints with time. The time values depends on the manipulator used.

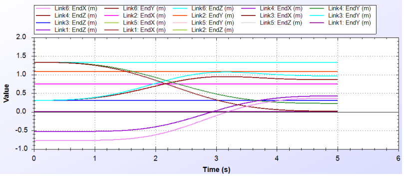


Figure 10: Link coordinate vs Time

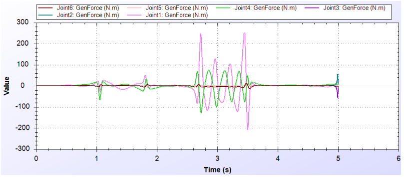


Figure 11: Joint torque vs Time

2.3 Inverse kinematics

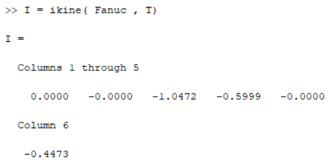
Inverse kinematics is the problem of finding the robot joint coordinates, given a homogeneous transform representing the pose of the end-effector. It is very much helpful, when the path followed is pre-planned in the cartesian space. For instance, any geometrical dimension or figure can be solved by analytical as well as simulation software approaches. Firstly, we show the inverse kinematic model with the analytical process using numerical equations in which we find the value joint (1-6) angles. The calculation of joint angles is required to move this system to a point on the component surface. This approach is solved to find the first three joint angles followed by the last three joint angles. The joint angle solution equations are shown in Eq(5). The values of p_x, p_y and p_z are calculated using Eq. (5). p_x , p_y and p_z are the last column of matrix Eq (6).

$$
\begin{array}{ll}
p_x = p_{x6} - (d_6) * a_x & \\
p_y = p_{y6} - (d_6) * a_y & \\
p_z = p_{z6} - (d_6) * a_z & \\
\theta_1 = a * tan2(p_y, p_x)) * 180/\pi & -\pi \leq \theta_1 \geq \pi \\
\theta_2 = tan^{-1}(\frac{sin\theta_2}{cos\theta_3}) & -\pi \leq \theta_2 \geq \pi \\
\theta_3 = tan^{-1}(\frac{sin\theta_3}{cos\theta_3}) & -\pi \leq \theta_3 \geq \pi \\
\theta_4 = tan^{-1}(\frac{sin\theta_4}{cos\theta_4}) * 180/pi & \\
for\ orientation\ \omega\ and\ wrist\ indicators & \\
\theta_5 = tan^{-1}(\frac{sin\theta_5}{cos\theta_5}) * 180/pi & -\pi \leq \theta_5 \geq \pi \\
\theta_6 = tan^{-1}(\frac{sin\theta_6}{cos\theta_6}) * 180/pi & -\pi \leq \theta_6 \geq \pi
\end{array}
$$

---

(5)

Through the Robotics Toolbox software approach, First step is to generate the transform corresponding to particular joint coordinates, and then find the corresponding joint angles using the function ikine(), which yields coherent values with the analytical method. The inverse kinematic procedure for any specific robot can be derived symbolically [8] and commonly, an efficient closed-form solution can be obtained. However the Toolbox gives only a generalized description of the robotic manipulator in terms of kinematic parameters so an iterative numerical process must be used. The starting point for the solution may be input, or else it defaults to zero, and provides limited control over the particular solution that will be obtained.



```
>> I = ikine( Fanuc , T)

I =

  Columns 1 through 5

    0.0000   -0.0000   -1.0472   -0.5999   -0.0000

  Column 6

   -0.4473
```

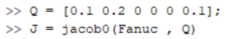
Figure 12: Inverse kinematics transformation

The manipulators Jacobian matrix, J_q, maps differential motion or velocity between configuration and Cartesian space. For an m-axis manipulator the end-effector Cartesian velocity is given by-

$$
{}^{0}\vec{X}_m =^{0} J_q(\vec{q})\vec{q} \tag{6}
$$

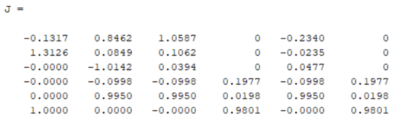
$$ {}^{T_n}\dot{\vec{X}}_m = J^T m_q(\vec{q})\vec{q} \tag{7} $$

These are stated in terms of base or end-effector coordinates respectively and where is the Cartesian velocity represented by a 6-vector as above. The two Jacobians are computed by the Toolbox functions jacob0() and jacobe() respectively. For an m-axis manipulator the Jacobian is a 6n matrix.



```
>> Q = [0.1 0.2 0 0 0 0.1];
>> J = jacob0(Fanuc , Q)
```

Figure 13: Jacobian Equation



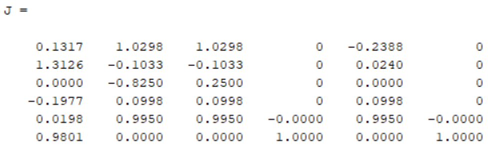
```
J =

   -0.1317    0.8462    1.0587         0   -0.2340         0
    1.3126    0.0849    0.1062         0   -0.0235         0
   -0.0000   -1.0142    0.0394         0    0.0477         0
   -0.0000   -0.0998   -0.0998    0.1977   -0.0998    0.1977
    0.0000    0.9950    0.9950    0.0198    0.9950    0.0198
    1.0000    0.0000   -0.0000    0.9801   -0.0000    0.9801
```

Figure 14: Jacobe Matrix

Note that in Figure 15, the top right 4 and 6 column indicate, correctly, that motion of joints 4 and 6 does not cause any translational motion of the robots end-effector. .

$$ \dot{q} = ({}^{0}J_q^{-1})({}^{0}\dot{x}_m) \tag{8} $$

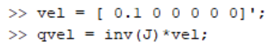
The article[9] which gives the joint velocities required to to achieve the desired Cartesian velocity. In this example, in order to achieve 0.1m/s translational motion in the end-effector X-direction the required joint velocities can be easily find from the velocity graph plot.which requires approximately equal and opposite motion of the shoulder and elbow joints.



```
J =

    0.1317    1.0298    1.0298         0   -0.2388         0
    1.3126   -0.1033   -0.1033         0    0.0240         0
    0.0000   -0.8250    0.2500         0    0.0000         0
   -0.1977    0.0998    0.0998         0    0.0998         0
    0.0198    0.9950    0.9950   -0.0000    0.9950   -0.0000
    0.9801    0.0000    0.0000    1.0000    0.0000    1.0000
```

Figure 15: Jacobn Matrix

The null space of this Jacobian indicates that equal and opposite motion of joints 4 and 6 will result in zero motion of the end-effector.



```
>> vel = [ 0.1 0 0 0 0 0]';
>> qvel = inv(J)*vel;
```

Figure 16: Velocity equation

There also exists a case of kinematic singularity in which a position is achieved in which two robot links align in same orientation resulting in a reduction in the degree of freedom of the manipulator.

2.4 Dynamic Analysis

Manipulator dynamics is concerned with the equations of motion, the way in which the manipulator moves in response to torques applied by the actuators, or external forces. The mathematics of the dynamics of serial link manipulators with valid arguments along with a brief history on the mathematical formulation are well presented in articles [8] [10]. The equations of motion for an m-axis manipulator are given by

$$Q = M(\vec{q})\vec{\ddot{q}} + C(\vec{q}, \vec{\dot{q}})\vec{\dot{q}} + F(\vec{\dot{q}}) + G(q) \tag{9}$$

q :vector of generalized joint coordinates describing the pose of the manipulator. $\vec{\dot{q}}$: vector of joint velocities. $\vec{\ddot{q}}$: the vector of joint accelerations M: manipulator tensor for inertia
C: describes Coriolis and centripetal effects centripetal moments are proportional to $\vec{\dot{q}}_i^2$, while the Coriolis moments are proportional to $\vec{\dot{q}}_i$, $\vec{\dot{q}}_j$
F : describes viscous and Coulomb friction
G : the gravity loading Q is the vector of generalized forces associated with the generalized coordinates q.

The manipulators kinematics and dynamics are represented in a general way within the robotics toolbox by a dyn matrix which is given as the first input argument to the dynamic functions. Each row shows one link of the manipulator and the columns are assigned according to Table 1 (D-H Table). The dyn matrix is actually a D-H matrix modified with additional columns for the mass and inertial parameters for each link, as well as that of motor inertia and friction parameters. The joint torques required are calculated by Inverse Dynamics to achieve the specified state of joint position, velocity and acceleration. The recursive Newton-Euler formula is an efficient matrix oriented algorithm for calculating the inverse dynamics, and implemented by the function rne(). The analysis process is very straight forward which further strengthens the reason to use this software in addition to MATLAB. The time steps and time for which the analysis is needed can be entered and the Inverse dynamic analysis button does the needful. In this paragraph, we will discuss different temporal graphs plotted for the Dynamic Analysis of Fanuc S430-IW in Robo-Analyzer Simulation Software by creating 6Dof skeleton robot using D-H table as of Fanuc S430-IW with the same initial and final joint values. Fig.17 show that link coordinates of link 4 and link 6 are changing relatively significantly with time. similarly like this we can easily plots the velocity and acceleration graphs of links. Joint torque values are same for dynamic analysis as in Fig.11. We conclude that Link 4 and 6 show major changes with minor changes in link 1.

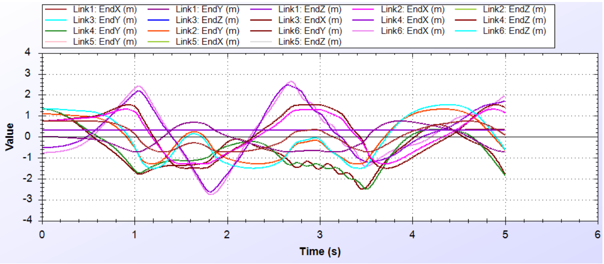


Figure 17: Link coordinate vs Time

3 Conclusion

Through a carefully methodical analytical approach validated with a numeric-software approach, a novel exemplified complete procedure is presented to carry out the complete Kinematic and Dynamic analysis of a 6 DoF Manipulator. A specific 6DoF manipulator (Fanuc S430-IW) is selected for a targeted approach but the methodology can be generalized for any 6 DoF robotic manipulator. Complete graphical data is provided to ease the transparency of the process and improve the analysis readability. With a few modifications, the method can also be adopted for m- DoF robotic manipulator. All the results of the analysis are discussed and explained in the article. Hence a cross-domain approach to Kinematic and dynamic modelling and analysis of a robot is investigated with supporting graphical and numerical results.

References

[1] R. Hartenberg Denavit. A kinematic notation for lower-pair mechanisms based on matrices. *ASME journal*, 1955.

[2] M. Dahari and J. Tan. Forward and inverse kinematics model for robotic welding process using kr-16ks kuka robot. *2011 Fourth International Conference on Modeling, Simulation and Applied Optimization*, 2011.

[3] Amogh Patwardhan, Aditya Prakash, Rajeev lochana, and G.Chittawadigi. Kinematic analysis and development of simulation software for nex dexter robotic manipulator. *International Conference on Robotics and Smart Manufacturing (RoSMa2018)*, 2018.

[4] Caitlin Baciu Dan-Ilie Buliga Daniel Constantin, Marin Lupoae. Forward kinematic analysis of an industrial robot. *new development in mechanics and mechanical engineering*, 2015.

[5] Jeremy L. Rickli Kiran Deshmukh and Ana Djuric. Kinematic modeling of an automated laser line point cloud scanning system. *Procedia Manufacturing*, 2016.

[6] Peter Corke. Robotics, vision and control: Fundamental algorithms in matlab. *Springer Tracts in Advanced Robotics,*, 2011.

[7] Peter Corke. Robotics toolbox for matlab. *pdf*, 2014.

[8] R.P Paul. Robot manipulators: Mathematics, programming, and control. *MIT Press*, 1981.

[9] D. Whitney and D. M. Gorinevskii. The mathematics of coordinated control of prosthetic arms and manipulators. *ASME Journal of Dynamic Systems, Measurement and Control,*, 1972.

[10] J. M. Hollerbach. Dynamics, in robot motion - planning and control. *MIT Press*, 1982.