5-DOF Manipulator Simulation based on MATLAB/Simulink Methodology

[1]Sagar K. Soni, [2]Vishal A. Pandya

[1]M. Tech Research Scholar, [2]Assistant Professor
Mechanical Engineering Department
Birla Vishvakarma Mahavidyalaya Engineering College, Vallabh Vidyanagar-388120, Anand, Gujarat, India.

Abstract : Robot technology is advancing rapidly, which leads to moving of industry from automation to robotization. The serial link manipulator-type robot is highly versatile for performing various industrial tasks. Closed form solution of inverse kinematics for all robot configuration is not possible. Therefore, to eliminate this problem jacobian based iterative method used for inverse kinematics. In this paper present inverse kinematics algorithm for damped least square method (DLS), pseudoinverse and jacobian transpose method. These methods compare based on approximation error and behavior near singularity. Inverse kinematics algorithm develop in MATLAB Simulink and Simulated in MATLAB Simscape environment. For simulation, 3D CAD model of robotic arm is prepared in CREO 3.0

***keywords* – kinematics, Matlab/Simulink, jacobian, iterative**

I. INTRODUCTION

Robotics is the field of engineering and technology, which deals with the study of robot that includes design, controlling, programming etc., which are substantially different from those of existing techniques. Now a day robotics is dealing with research and development of interdisciplinary areas, which includes kinematics, dynamics, control, motion planning, sensing, programming and machine intelligence. [1]

The robot technology is advancing rapidly. The industry is moving from the current state of automation to robotization, to increase productivity and deliver uniform quality. Robot and robot-like manipulators are now commonly employed in hostile environment, such as atomic plant, space station and satellites. There are now increasing number of applications of robot in medical. [2]

One type of robot commonly used in industry is robotic manipulator or robotic arm. It is an open or closed kinematic chain of rigid links interconnected by movable joints. In some configurations, links can be considered to correspond to human anatomy as waist, upper arm, and forearm with joints at shoulder and elbow. At the end of arm end effector, it may be tool and its fixture or gripper. [2]

Kinematic analysis is the first step for controlling the robot to perform the desired task. Robotic manipulators are of many types; here only open chain serial manipulator is considered. Firstly, forward kinematics is evaluated, which can be easily analyzed using the Denavit-Hartenberg parameter. Next step is to evaluate inverse kinematics for obtaining joint variable for the defined pose of the robot.

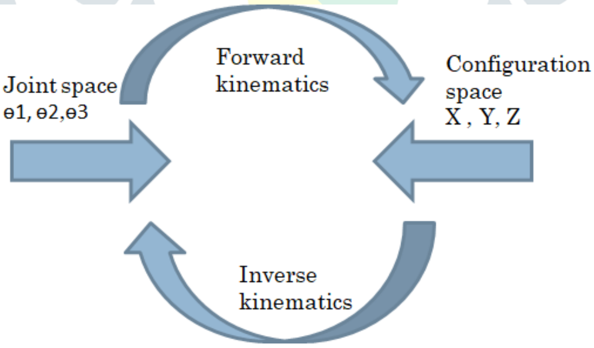

Figure 1: Forward kinematics and inverse kinematics

Closed form or analytic method is the easy solution for inverse kinematics. But, this method is not applicable for all tupe robot configuration, Due to high degree of freedom and non linear behavior near singularity. So, it is required to solve the inverse kinematic problem using an iterative method, which can be easily utilized for all robot configurations.

Many iterative inverse kinematic methods available that are Newton-Raphson method, Jacobian inverse method, Jacobian transpose, Jacobian pseudoinverse method, damped least square (DLS) method etc. to solve the non-linear equation. Iterative methods do not give an exact solution but an approximate solution and also does not guarantees the singularity. All these methods approximate least square solution to reduce the error between current and target pose of the robot. All the iterative methods are Jacobian based and computationally expensive. Each iteration approximates the differential joint velocity, which in turns give differential motion on integration. The joint variables are updated at end of each iteration by adding differential joint values obtained in each iteration.

II. LITERATURE SURVEY

M. A. Qassem [3] have done experiment on AL5B Robot arm. It utilizes Mat lab/ Simulink and AutoCAD as the tools for testing motional characteristics of the AL5B Robot arm. A robot model will be developed; forward – inverse - kinematics will be implemented & tested transformation matrix Equation will be used to calculate inverse kinematics equations. Graphical User Interface (GUI) was developed in Mat lab to test and simulate the motional characteristics of the Robot arm. A physical interface between the AL5B robot arm and the GUI will be designed.

Z Chen [4] have presented Inverse kinematics solution done on the base of D-H parameter & the transformation matrix. By using direct kinematics method for solving the equation and get solution to reach particular selected point. Using MATLAB robot toolbox to build a mathematical model to draw three-dimensional robot arm. Using MATLAB programming to achieve the forward solution and the inverse solution of the manipulator Verify the correctness of solution by single point and continuous path verification.

V.N. Iliukhina [5] has develop Numerical Method-Bruit force iterative method to solve the system of nonlinear equations. The solution of position coordinates is obtained using brute force iterative method in embedded c and the algorithm prepared. It is complex to solve the system of nonlinear equations having trigonometric functions analytically. Hence, the analysis carried out uses iterative numerical method. The algorithm is tested for over 20 sample test cases and is forfend to be working efficiently.

Charles w. wampler [6] conclude that the Jacobian matrix, are inefficient and fail near kinematic singularities. Vector formulations of inverse kinematic problems are developed that lead to efficient computer algorithms. To overcome the difficulties encountered near kinematic singularities, the exact inverse problem is reformulated as a damped least-squares problem, which balances the error in the solution against the size of the solution.

K W Chin [7] has discussed the algebraic method, Newton Raphson method with many other modified iterative methods for the 6DOF robot. The comparison is shown among them and concluded that NR method is superior to all other used. Algebraic method is significantly faster but gives more than one solution for the same configuration of robot. Newton's method converges quickly with a small number of objective function evaluations, but requires that the initial guess be close to the target configuration.

H. Das [8] has developed a new iterative method for the solution of inverse kinematics of redundant system using the transpose of Jacobian matrix instead of Pseudoinverse. The two new multiplying factor is implemented to increase the convergence rate. Then the same is compared to NR method, NR method required more computation per step then Jacobian transpose method and NR method becomes unstable near the singular values. The new developed iterative method is implemented for 10 DOF planer robot with obstacle avoidance.

S R Buss [9] Introduced various Jacobian based methods explaining Pseudoinverse method being fast but the poor quality of approximation. It gives very large end effector velocity near the singularity and when the target is beyond its reachability. It is remarked that, complexity in getting the closed form solution increase with increase in DOF. In Jacobian Pseudo inverse method, velocity is integrated with the position due to that, there would be a small drift in the solution, but it can reduce by reducing step size. S R Buss introduces the Damped least square method as superior to all other methods. It avoids the problem of Pseudoinverse method and damps the high velocities.

Ignacy Dule [10] has made a comparative study for various Jacobian based methods. The comparison is done because of a number of elementary operations to compute an iteration. It is shown that Pseudoinverse requires large operation followed by damped least square method and followed by Jacobian transpose method. Jacobian transpose method is computationally cheapest and damped least square method is more better with bit more computational cost. Similarly, in comparison based on time consumed for computation, Pseudoinverse takes maximum computation time followed by DLS and Jacobian transpose method for movement of the end effector on a specific path. The number of methods based on the first and the second order approximations along with Jacobian transpose method, the modified Levenberg–Marquardt method (DLS) of the inverse Jacobian matrix have been tested. It is concluded from the simulations of three different.

Based on the literature review conclude that the major common problem in the above-stated methods is a poor performance during singularity. Which can be efficiently handled by DLS method by introducing the damping factor. The damping factor damps the large velocity near singularity, which makes the system robust against singularity and can give a numerically stable method for approximating Δq. It is also called the Levenberg-Marquardt method and was first implemented for inverse kinematics by Wampler [6]. The quantity minimizes in pseudoinverse method is changed to by introducing damping factor to the pure least square solution.

III. MATHEMATICAL MODEL OF ROBOT KINEMATICS

To define kinematics of the manipulator axes need to be placed according to the Denavit-Hartenberg method the basic rules of which are:

- Z-axis is in the direction of the joint axis
- X-axis is parallel to the common normal
- ɵi is the angle from xi-1 to xi along zi-1
- di is the distance from the intersection of zi-1 with xi to the origin of (i-1) system of axes

1. Forward kinematics

FK of manipulator is to find out the position of end effector in work space based on the given joint angle parameter. One common method to done FK is DH parameter based. To find out the overall transformation matrix of 4 x 4. In that last column first three elements give the end effector position in work space in X, Y, Z coordinate and top left 3 x 3 is orientation matrix of end effector.

Frame assignment of 5-DoF robot is shown in fig 2.

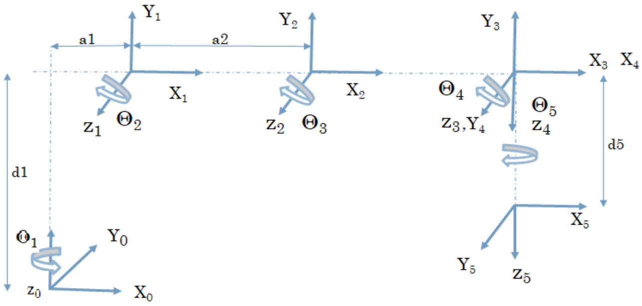

Figure 2: Frame assignment of 5-DOF robotic manipulator

based on the frame assignment prepared the DH parameter table is shown below.

Table 1: D-H parameter of 5-DOF robotic manipulator

Joint	Link twist α_i	Link length a_i	Joint offset d_i	Joint angle θ_i
1	**90**	**101.25**	**334.25**	θ_1
2	**0**	**220**	**0**	θ_2
3	**0**	**220**	**0**	θ_3
4	**90**	**0**	**0**	θ_4
5	**0**	**0**	**137.35**	θ_5

Calculation of the homogeneous transformation matrix 0T5 that connects the system of axis X5Y5Z5 with the absolute system of axes X0Y0Z0 is given below:

0T5 = 0T1 1T2 2T3 3T4 4T5 (1)

$$
{}^{0}T_{5} = \begin{bmatrix} C_1C_{234}C_5 + S_1S_5 & C_1C_{234}S_5 + S_1C_5 & C_1S_{234} & C_1(d_5S_{234} + a_3C_{23} + a_2C_2 + a_1) \\ S_1C_{234}C_5 - C_1S_5 & -S_1C_{234}S_5 - C_1C_5 & S_1S_{234} & S_1(d_5C_{234} + a_3S_{23} + a_2C_2 + a_1) \\ S_{234}C_5 & -S_{234}S_5 & -C_{234} & (-d_5C_{234} + a_3S_{23} + a_2S_2 + d_1) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)
$$

Here, C1 =cos(θ1), S1 = sin(θ1),C23 = cos(θ2 + θ3), C234 = cos(θ2 + θ3 + θ4), S234 = sin(θ2 + θ3 + θ4)

2. Inverse kinematics by iterative method

Iterative methods are very well known for their applicability to any robotic configuration. Iterative methods are also known as numerical solution as it uses numerical mathematics to solve. These methods are computationally extensive and by nature slower compared to the closed formed method. Iterative methods do not give an exact solution but an approximate solution and also does not guarantees the singularity, with certain iterative methods. All the iterative methods are Jacobian based which maps the joint variable to the Cartesian space or vice-versa on its inversion.

As shown in below fig. e is an error. In each iteration this error minimize upto user define set limit.

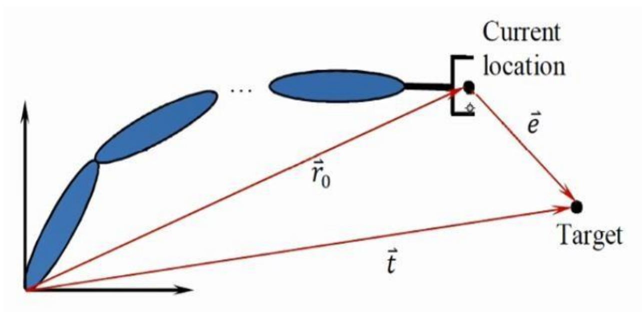

Figure 3: Concept of iterative method

Inverse kinematics by using jacobian base iterative method require jacobian matrix of our robot configuration. Calculate the individual jacobian of each link, and then combine to form an overall manipulator jacobian.

Jacobian is known as multidimensional form of derivative, which calculates the derivative of any function. . From forward kinematics, the position and orientation are the function of the joint variable (q) as shown in eq. (2)

$$P_e = F(q) \tag{2}$$

By partial differentiating equation (2) w.r.t joint variable (q),

$$V_e = \frac{\partial F}{\partial q}\delta q \tag{3}$$

V_e = Cartesian velocity vector

$$V_e = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{d} \\ \dot{\theta} \end{bmatrix} \tag{4}$$

$$V_e = J(q).\delta q \tag{5}$$

Rewriting equation (5)

$$\delta q = J(q)^{-1} \cdot V_e \tag{6}$$

$$J(q) = \begin{bmatrix} J_1(q) & J_2(q) & J_3(q) & J_4(q) & J_5(q) \end{bmatrix} \tag{7}$$

2.1 Damped least square method

The major common problem in the other jacobian based method has poor performance during singularity. Which can be efficiently handled by DLS method by introducing the damping factor. Damping factor damps the large velocity near singularity, which makes the system robust against singularity and can give a numerically stable method for approximating $\Delta\dot{q}$. It is also known as Levenberg Marquardt method and was first implemented for inverse kinematics by Wampler [6].

$$\Delta\dot{q} = (JJ^T + \lambda^2 I)^{-1}.e \tag{8}$$

Select the damping factor carefully for numerically stable equation. Higher value of damping factor make system robust so it should be optimum for stable movement. . The estimation of singular value can be obtained by singular value decomposition of Jacobian matrix.

$$J = \sum_{i=0}^{n} \sigma_i u_i v_i \tag{9}$$

Where ui, vi are input and output singular vector and $\boldsymbol{\sigma i}$ is singular value. On the basis of equation (9) there is little influence of damping factor

$$\frac{\sigma_i}{\sigma_i^2 + \lambda_i^2} \approx \frac{1}{\sigma_i} \tag{10}$$

On the other hand, when the singularity is approached, the smallest singular value tends to be zero while the associated component of the solution is driven to zero by the factor $\frac{\boldsymbol{\sigma_i}}{\boldsymbol{\lambda_i}}$ this progressively reduces the joint velocity.

Algorithm for inverse kinematics is shown in Fig.(3). This method give best least square solution for joint variables to reduce the error between target and current position in each iteration. To stop the program set user define accuracy. Updated joint variable at each iteration calculate forward kinematics and update Jacobian.

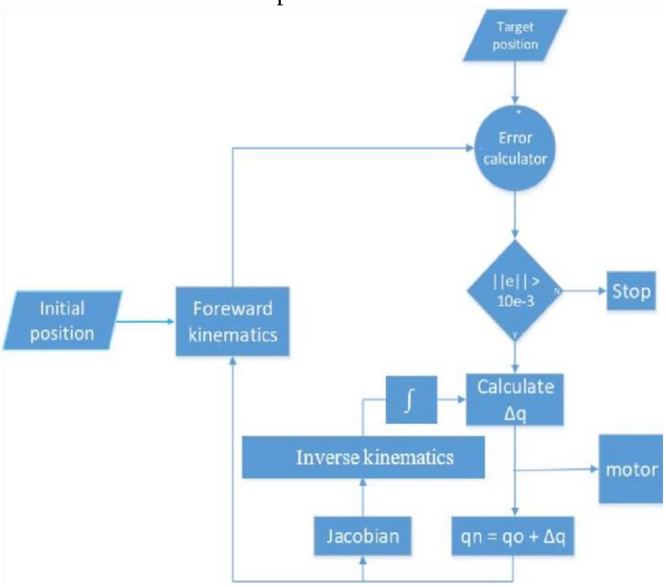

Figure 4: Inverse kinematics algorithm flow chart

2.2 Calculation for checking position and orientation error in program

Position error :- Program develop in such a way that it compare value, one which is calculate from the forward kinematics element (x, y, z) and the target position in robot work space (X, Y, Z). When this position error within acceptable limit, program automatically stop calculating iteration. Position limit is set to o.o1 mm.

Orientation error :- Orientation error calculated on the angle-axis representation of orientation. In program Rd is desire orientation matrix and Re is rotation matrix computed from joint variables.

Equation for orientation error:-

$$e = r\sin(\upsilon) \tag{11}$$

$$R(v,r) = R_d.\mathrm{Re}^T(q) \tag{12}$$

$$v = \cos^{-1}\left(\frac{r_{11}+r_{22}+r_{33}-1}{2}\right) \tag{13}$$

$$r = \frac{1}{2\sin v}\begin{bmatrix} r_{32}-r_{23} \\ r_{13}-r_{31} \\ r_{21}-r_{12} \end{bmatrix} \tag{14}$$

So the value of ***e*** is 3 x 1 vector which give the orientation error about x, y, z axis. r_{11}, r_{21}, r_{31}, r_{23}, r_{32}, r_{12}, r_{13}, r_{33}, r_{22} are the component of rotation matrix $R(v,r)$.

IV. RESULT AND DISCUSSION

MATLAB Simulink is used to prepare inverse kinematics program and to visualize the movement of the robotic arm, simulation is done in the MATLAB simscape environment. Damped least square method is used for inverse kinematics. Inverse kinematics is done to reach the arbitrary position in the workspace. Position (300, 400, 200) value of the joint space variable q (53.15, -8.13, 56.94, -48.83, 90).

Damped least square method

Position error in terms of average squar root :- 0.018

Orientation error in terms of average squar root :- 0.0002614

Pseudoinverse method

Position error in terms of average squar root:- 0.018

Orientation error in terms of average squar root:-0.0002614

Jacobian transpose method

Jacobian transpose method is fail for above robot configuration, because in calculation of joint variable it gives high fluctuation in successive values due to mathematical constrain. So end effector not reach at the target position. Because calculation require 6 x 5 jacobian function.

Based on the simulation end effector reach the desire position in the workspace with a smooth velocity flow. The Damped least square and pseudoinverse solution dams the high velocity near singularity and efficiently controls the joint velocity. Compare result of DLS and pseudoinverse, both work equally efficient for above robotic configuration and jacobian transpose is failed to calculate joint variable.

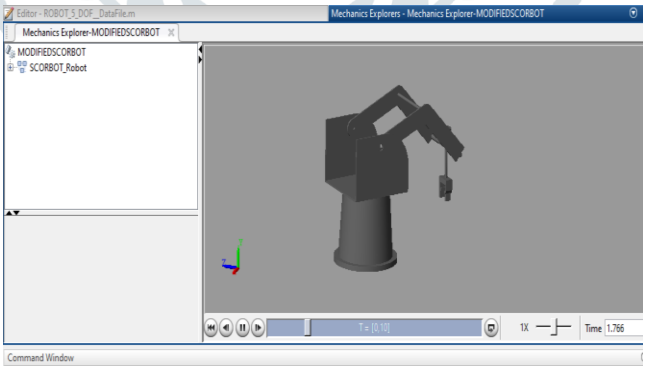

Figure 5: Result of modelling of inverse kinematics in MATLAB simscape

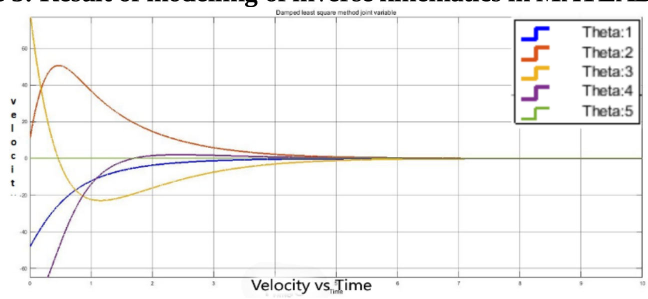

Figure 6: DLS - Joint velocity during each iteration of different theta value

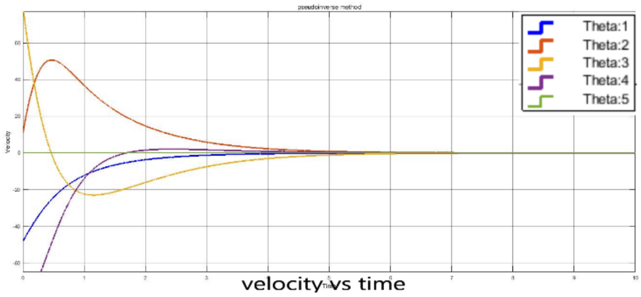

Figure 7: Pseudoinverse - Joint velocity during each iteration of different theta value

V. CONCLUSION

Based on the result conclude that DLS and pseudoinverse method give similar behavior for the above robotic configuration in terms of error and velocity near singularity and jacobian transpose is failed to calculate feasible joint value due to mathematical constraint. The inverse kinematics problem for the 5 DOF robotic manipulator is verify succesfully for iterative IK method. The kinematics program is modelled successfully in MATLAB Simulink and simulate the model in simscape environment succesfully. Direction for further research is the development of an algorithm for trajectory planning taken into account.

REFERENCES

[1] S. K. Saha, Introduction to Robotics, New Delhi: Mc graw hill education, 2014.

[2] R. K. Mittal and I. J. Nagrath, Robotics and Control, New Delhi: Mc Graw Hill Education, 2015.

[3] M. A. Qassem and I. Abuhadrous, "“Modeling and Simulation of 5-DoF Educational Robot Arm," *International Conference on Advanced Computer Control, IEEE,* april, 2007.

[4] Z. Chen, J. Yan and M. Li, "Kinematics Analysis and Application of 5-DoF Manipulator with Special Joint," in *Chinese Automation Congress*, 2017.

[5] V. N. Iliukhina, K. B. Mitkovskiib and D. A. Bizyanovaa, "The Modeling of Inverse Kinematics for 5-DoF Manipulator," *Procedia Engineering,* vol. 176, pp. 498-505, 2017.

[6] C. wampler, "“Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods," *IEEE transactions on systems, man, and cybernetics,* vol. 16, 1986.

[7] C. K. Wu, K. B. R and M. Andrew, "Closed-form And Generalized Inverse Kinematics Solutions for the Analysis of Human Motion," in *19th international conference IEEE*, Chicago, USA, Oct. 30 to Nov. 2, 1997.

[8] H..Das, E. Slotine and T. Sheridan, "Inverse kinematic Algorithm for Redundent Systems," in *IEEE, Conference on robotics and automation, Cambridgr, 1988.* , Cambridger, 1988.

[9] R. B. Samule, "Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods," in *IEEE, University of California, San Diego*, san diego, 2009.

[10] I. Duleba and M. Opalka, "A Comparison Of Jacobian–based Methods Of Inverse Kinematics For Serial Robot Manipulators," *International journal of Applied Mathematics in Computert science,* vol. 23, pp. 373-382, 2013.

[11] B. Siciliano and L. Sciavicco, Modelling, Planning and Control, London: Springer, 2010.