

Comparative Study of Iterative Inverse Kinematics Methods for Serial Manipulators

Manan Kalasariya
Research Scholar

Mechanical Engineering Department,
BVM Engineering College,
V. V. Nagar, Anand, Gujarat, India

Dr Vinay Patel
Professor

Mechanical Engineering Department,
BVM Engineering College,
V. V. Nagar, Anand, Gujarat, India

Ashish Thakkar
Assistant Professor

Mechanical Engineering Department,
BVM Engineering College,
V. V. Nagar, Anand, Gujarat, India

Abstract— It is not possible to formulate the closed form solution of inverse kinematics for all kind of robot configuration. This paper presents the comparative study for the performance of different iterative inverse kinematic method utilized for a serial manipulator which can be applicable for all kind of robot configuration. Jacobian inverse, Jacobian transpose, Jacobian pseudoinverse and damped least square (DLS) methods are compared in the point of view of convergence rate, computation requirement, approximation accuracy and behavior near the singularity. A 3-DOF spherical robot is analyzed using these methods. Formulation of the inverse kinematic algorithm is programmed using MATLAB Simulink and simulated in the MATLAB Simscape environment which utilizes the 3D model of robot created using PTC Creo.

Keywords— Inverse kinematics, Jacobian inverse method, Jacobian transpose method, Jacobian Pseudoinverse method, Damped Least Square method

I. INTRODUCTION

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. Closed form or analytic method is the simplest solution for inverse kinematics. But, it is not possible to obtain the analytic relation between joint variables and robot end effector position and orientation for all robot configurations, especially in case of high degree of freedom manipulator due to its non-linear behavior. 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 [1]. 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.

A. Jacobian

Jacobian is the multidimensional form of derivative which calculates the derivative of any function e.g. Y_i as a function of differentials of X_j . From forward kinematics, it is inferred that the position and orientation are the function of the joint variable (q) as shown in (1). So, by differentiating position and orientation of end effector (P_e) will give the linear and rotational velocity (q).

$$P_e = F(q) \quad (1)$$

By partial differentiating (1) w.r.t joint variable (q).

$$V_e = \frac{\partial F}{\partial q} \delta q \quad (2)$$

Where V_e = Cartesian velocity vector.

$$V_e = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \dot{d} \\ \dot{\theta} \end{bmatrix} \quad (3)$$

Note the function F is non-linear, so partial derivative of F will be a function of q_i . The $\frac{\partial F}{\partial q}$ term in (2) is known as Jacobian $J(q)$. So, Jacobian is the mapping matrix which maps between the joint space and the Cartesian space.

$$V_e = J(q)\delta q \quad (4)$$

Where, $J(q) = [J_1(q) \ J_2(q) \ \dots \ J_n(q)]$

Rewriting (4) for δq by inverting Jacobian as (5), can be used to map the position and orientation form the Cartesian space to the joint space that helps in the inverse kinematics of robot [2].

$$\delta q = J(q)^{-1} \cdot V_e \quad (5)$$

II. ITERATIVE INVERSE KINEMATICS METHODS

Iterative inverse kinematics are Jacobian based method and gives differential control to robot manipulator. Each iteration gives a short motion towards target constrained by the short time interval. The various iterative inverse kinematics methods are discussed in this section.

The simplest Jacobian transpose method utilizes the Jacobian transpose instead of the Jacobian inverse. This reduces the computation of matrix inversion. The change in joint velocity $\Delta\dot{q}$ is calculated as (6)

$$\Delta\dot{q} = \alpha \cdot J(q)^T \cdot e \quad (6)$$

Where α is the scalar constant effects the change in q during each iteration [1] and e is the error vector (6×1) defining position and orientation error between current pose and target pose. The value of scalar α should small enough to minimize the new value of the error vector after the update [1]. Choose α so as to make this value as close as possible to error vector. The main advantage of this method is faster convergence, minimum computation, and less complexity.

H. Das [3] 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 Newton Raphson method, NR method required more computation per step then Jacobian transpose method and also NR method becomes unstable near the singular values. The new developed iterative method is implemented for 10 DOF planer robot with obstacle avoidance.

Jacobian inverse method directly utilizes the inverse of Jacobian to calculate the change in joint variables. But Jacobian inversion is only possible with the full rank Jacobian matrix. So, the greatest limitation of this method is the requirement of a square matrix. The value of Δq can be calculated as(7).

$$\Delta\dot{q} = J(q)^{-1} \cdot V_e \quad (7)$$

The Jacobian inverse method requires large computation for matrix inversion but it has fast convergence property [4]. This method is not very popular due to limitation as stated before. And has another limitation during singularity, when Jacobian loss its full-rank and Jacobian inversion become impossible or gives large velocity near to singular configuration.

Jacobian matrix is of size ($6 \times n$), where n is the number of degree of freedom. So, when the degree of freedom is less than 6 the Jacobian matrix becomes rectangular or not of full rank. As a result, calculation of Jacobian inverse become impossible. So, the best possible approximate solution is given by Moore-Penrose inversion also known as pseudo inversion, denoted as J^+ . The Pseudoinverse method sets the value of Δq as in (8).

$$\Delta\dot{q} = J(q)^+ \cdot e \quad (8)$$

The above equation gives least square solution for $\Delta\dot{q}$, which is the unique vector of smallest magnitude which minimizes $\|J\Delta q - e\|$ or equivalently minimizes $\|J\Delta q - e\|^2$. For under-constrained system or redundant system, the Pseudoinverse is given by (9)

$$J^+ = J^T (JJ^T)^{-1} \quad (9)$$

S R Buss [1] 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 also when the target is beyond its reachability. M. R. de Gier [4] controlled the 6 DOF robot for application of 3D printing by the Jacobian Pseudoinverse method. It is remarked that, complexity in getting the closed form solution increase with increase in DOF. In Jacobian Pseudoinverse 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.

The major common problem in the above-stated methods is a poor performance during singularity. Which can be efficiently handled by Damped Least Square (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 $\Delta\dot{q}$. It is also called the Levenberg-Marquardt method and was first implemented for inverse kinematics by Wampler [5]. The quantity minimizes in pseudoinverse method is changed to (10) by introducing damping factor to the pure least square solution.

$$\|J\Delta q - e\|^2 + \lambda^2 \|\dot{q}\|^2 \quad (10)$$

Where λ is non zero damping constant. So, the damped least square solution is given by (11)

$$\Delta\dot{q} = J^T (JJ^T + \lambda^2 I)^{-1} \cdot e \quad (11)$$

The damping factor need to choose very carefully to make the (11) numerically stable. Higher the value of damping factor, higher the robustness against singularity and lower tracking accuracy. And on the other side lower value of damping factor makes the system less robust against singularity [1] [6]. The value of damping factor should be greater than the smallest singular value [6]. The estimation of singular value can be obtained by singular value decomposition of Jacobian matrix i.e.

$$J = \sum_{i=0}^n \sigma_i u_i v_i \quad (12)$$

Where u_i, v_i are input and output singular vector and σ_i is singular value. On the basis of (12), the damped least solution in (13) can be written as

$$\Delta\dot{q} = \sum_{i=0}^n \frac{\sigma_i}{\sigma_i^2 + \lambda_i^2} v_i u_i^T \cdot e \quad (13)$$

So, it is inferred from the above (13) when $\sigma_i \gg \lambda_i$ there is little influence of damping factor since

$$\frac{\sigma_i}{\sigma_i^2 + \lambda_i^2} \approx \frac{1}{\sigma_i} \quad (14)$$

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 $\frac{\sigma_i}{\lambda_i}$, this progressively reduces the joint velocity.

By selecting constant damping factor, constant damping factor would not allow taking full advantage of DLS method. So, it is required to select the dynamic damping factor based on the nearness to singular configuration [7]. Nearness to the singular configuration can be efficiently estimated by the smallest singular value of Jacobian matrix [6] [7]. Bruno Siciliano [6] implemented the dynamic damping factor to control 6-DOF robotic arm.

S R Buss [1] introduces the Damped least square method as superior to all other methods. It avoids the problem of Pseudoinverse method and damps the high velocities near the singularity. The DLS method worked substantially better than the Jacobian transpose method and Pseudoinverse method, although it is somewhat slower. But at the point where the error was minimized, there was a lot of oscillation and shaking.

C. W. Wempler [5] had determined the corrective joint motion for error in end effector position and orientation. The DLS method is formulated to avoid the singularity. He has also suggested for selecting the damping factor. As damping factor reduce the accuracy of the method and on the other hand it makes the system robust against singularity. It is suggested to divide the DOF into parts and applying DLS solution reduce considerable computation which will help to achieve the required accuracy efficiently.

Bruno Siciliano [6] reviewed damped least square method for 6 DOF robot. Similarly reviewed by Wempler [5] that large damping factor reduces the tracking accuracy even when there is the possibility of an accurate feasible solution. He has used two smallest singular values to estimate the damping factor for making the robot more robust. And also introduced the user-defined accuracy depends on the task to reduce the computation. The feedback correction strategy is adopted to avoid numerical drifts in solution due to discrete-time implementation in the inverse differential kinematics. Then this modified algorithm is implemented for the robot and obtained a considerable reduction in translation and orientation error.

Ignacy Dule *et.al.* [8] had made a comparative study for various Jacobian based methods i.e. Jacobian transpose, Jacobian pseudo-inverse, Damped least square method (DLS), Modified Damped least square method and approximation methods. The comparison is done on the basis 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 manipulators that the modified Levenberg–Marquardt method is much better than the Jacobian

transpose. Other than DLS method, all methods are more sensitive to the singularity.

Iterative methods for inverse kinematics is used generally for redundant robots when the closed form solution is not possible. Here an example of the 3-DOF spherical robotic arm is presented to simplify the problem for understanding and for performance analysis of the different iterative methods.

III. KINEMATIC ANALYSIS OF 3-DOF SPHERICAL ARM

A. Forward kinematics

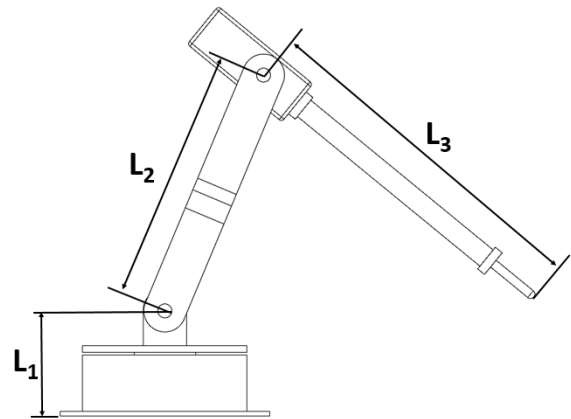


Fig. 1. 3-DOF spherical robotic arm

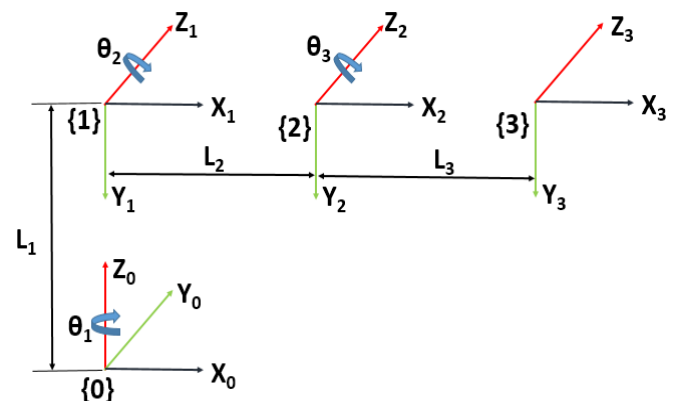


Fig. 2. Frame assignment for 3-DOF spherical robotic arm

TABLE I. D-H PARAMETER FOR 3-DOF SPHERICAL ROBOTIC ARM

i	a_i	α_i	θ_i	d_i
1	0	-90°	θ_1	$L_1 = 60$ mm
2	$L_2 = 146$ mm	0	θ_2	0
3	$L_3 = 200$ mm	0	θ_3	0

Fig. 1 shows 3-DOF spherical robotic arm having a first joint axis perpendicular to the second joint axis. Second and third joint axis are parallel to one another. Denavit and Hartenberg convention system is used to define coordinate systems and D-H parameters are defined as shown in TABLE I. The transformation matrix is determined using D-H parameters and overall transformation 0T_3 is obtain by (15). Equation (16) maps joint variables to Cartesian space represents orientation and position of the end effector. The standard convention for trigonometric functions in (16), and later is utilized, $C_1 = \cos(\theta_1)$, $S_{12} = \sin(\theta_1 + \theta_2)$.

$${}^0T_3 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \quad (15)$$

$${}^0T_5 = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & -S_1 & C_1(L_2 C_2 + L_3 C_{23}) \\ S_1 C_{23} & -S_1 S_{23} & C_1 & S_1(L_2 C_2 + L_3 C_{23}) \\ -S_{23} & -C_{23} & 0 & L_1 - L_2 S_2 - L_3 S_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

B. Inverse Kinematics

The iterative method requires Jacobian for computation of inverse kinematics solution. Jacobian for each joint is calculated as (4) and assembled to form manipulator Jacobian as in (17)

$$J = \begin{bmatrix} -S_1(L_2 C_2 + L_3 C_{23}) & -C_1(L_2 S_2 + L_3 S_{23}) & -C_1 L_3 S_{23} \\ C_1(L_2 C_2 + L_3 C_{23}) & -S_1(L_2 S_2 + L_3 S_{23}) & -S_1 L_3 S_{23} \\ 0 & -(L_2 C_2 + L_3 C_{23}) & -L_3 C_{23} \\ 0 & -S_1 & -S_1 \\ 0 & C_1 & C_1 \\ 1 & 0 & 0 \end{bmatrix} \quad (17)$$

Inverse kinematics algorithm is shown in Fig. 3. The iterative inverse kinematics approximates the best least square solution for joint variables to reduce the error between target and current position in each iteration. User define accuracy is to be set to stop iterating program. Joint variables are updated at each iteration to calculate forward kinematics and update Jacobian. Different inverse kinematics methods are implemented for a 3-DOF spherical robotic arm for performance analysis of inverse kinematics methods.

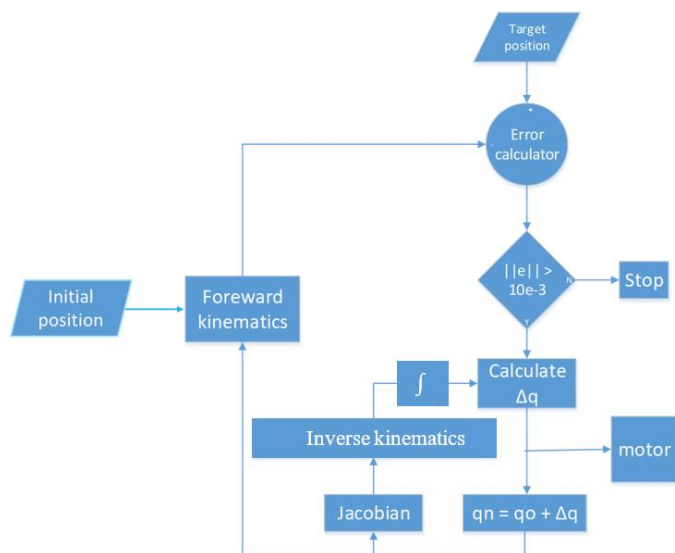


Fig. 3. Inverse kinematics algorithm

IV. RESULT AND COMPARISON

The inverse kinematics algorithm is programmed in MATLAB Simulink and simulated using MATLAB Simscape. Four methods are programmed i.e. Jacobian inverse, Jacobian transpose, Jacobian pseudoinverse and damped least square method.

The program is prepared in such a way that any method can be selected to solve inverse kinematics for performance study. The inverse kinematics is done to achieve arbitrary position (133,162,53) from the home position with $q = [0, -2\pi/3, -\pi/2]^T$. The user-defined accuracy is set to 10^{-3} mm. Results of the inverse kinematics solution by all methods are shown in TABLE II. Joint velocity during each iteration for each method is plotted as shown in Fig. 4.

TABLE II. RESULT TABLE

IK method	No. of iteration	Computation time (sec)
Jacobian Inverse	839	4.27
Pseudoinverse	839	3.77
Jacobian transpose	291	2.91
Damped least square	779	4.02

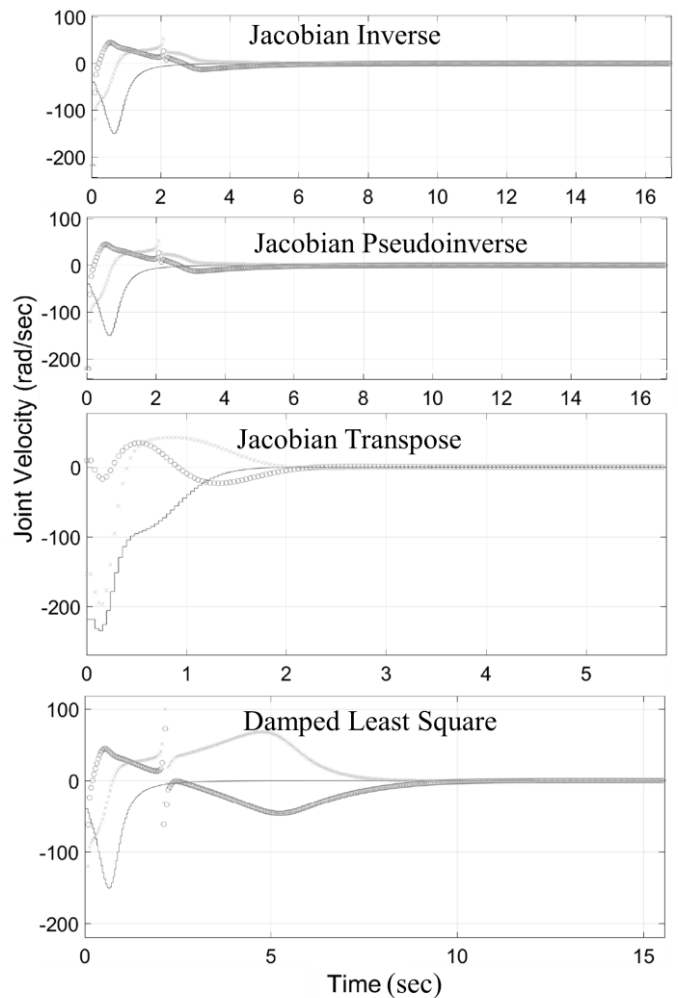


Fig. 4. Joint velocity during each iteration using different IK methods (— Joint-1, - - - Joint-2, . . . Joint-3)

It is inferred from TABLE II. that Jacobian transpose method is fast converging and also take less computation time comparatively, but it is not able to handle singularity. Jacobian inverse and pseudoinverse both take approximately same iteration (exactly same iteration in this case due to the square Jacobian matrix) but pseudoinverse takes less computation time compared to the Jacobian inverse method. Both these Jacobian inverse and pseudoinverse is unstable near singularity as inferred from Fig. 4, the joint velocity of joint-2 and joint-3 approaching

infinity near arm singularity. But, as seen in Fig. 4, the Damped least square solution damps the high velocity near singularity and efficiently controls the joint velocity. With the addition of velocity damping, DLS converge fast but requires a little more computations comparatively, as the number of elementary operations increases by incorporating damping factor. If the robotic configuration is far from the singularity, DLS give the approximately same result as pseudoinverse method. Iteration count in DLS is depended upon the nearness of robotic configuration to the singularity. It can be suggested that if dynamic damping factor is used in Damped least square method then the robot can be efficiently controlled superior to all other iterative methods.

V. CONCLUSION

All the iterative method has its own advantages but possesses common limitation for computation inverse Jacobian. Jacobian transpose method converges very fast with very less computation requirement. The Jacobian inverse method requires a full-rank Jacobian matrix, this limitation is overcome by the pseudoinverse method. Pseudoinverse best approximates the Jacobian inverse for non-square matrix and also require less computation time comparatively. Although it possessing many advantages, pseudoinverse does not perform well near the singularity. It is concluded from the comparison of different iterative kinematics method that the damped least square method is superior to all other methods that are reviewed in terms of convergence rate and robustness against computational

singularity but at the cost of reduction of accuracy if optimal damping or dynamic damping factor is not selected. And also damped least square method is computationally stable and can be efficiently implemented.

REFERENCES

- [1] Buss, Samule R, "Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods," University of California, San Diego, 2009.
- [2] R K Mittal, I J Nagrath, Robotics and Control, New Delhi: Mc Graw Hill Education, 2015.
- [3] H.Das, J-J. E. Slotine and T.B. Sheridan, "Inverse kinematic Algorithm for Redundent Systems," IEEE, Cambridgr, 1988.
- [4] Gier, M. R. de, "Control of a robotic arm: Application to on-surface 3D-printing," Delft University of Technology, 2015.
- [5] Wampler, Charles W., "Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods," *IEEE Transactions On Systems, Man, And Cybernetics*, vol. 16, no. 1, 1986.
- [6] Bruno Siciliano, Stefan Chiaverini, "Review of the Damped Least-Squares Inverse Kinematics with Experiments on an Industrial Robot Manipulator," *IEEE Transactions On Control Systems Technology*, vol. 2, no. 2, 1994.
- [7] Anthony A. Maciejewski, Charls A. Klein, "Numerical Filtering for the Operation of Robotic Manipulators through Kinematically Singular Configuration," *Journal of Robotic Systems, John Wiley & Sons*, vol. 5, no. 6, pp. 527-552, 1998.
- [8] Ignacy Duleba, Michal Opalka, "A Comparison Of Jacobian-Based Methods Of Inverse Kinematics For Serial Robot Manipulators," *International Journal of Applied Mathematics in Computer science*, vol. 23, no. 2, pp. 373-382, 2013.