Association Journal of Interdisciplinary Technics in Engineering Mechanics, Vol. 3, No. 3, 2025
ISSN: 3049-0898

Robotic Motion Planning Using the Jacobian Matrix-Based Inverse Kinematics Algorithm

Dr. Snehal Patong[1], and Dr. Nandan Suresh[2]

[1,2]Akriti College of Mechanics & Design, Aurangabad, India.

Abstract--- Planning a robot's motion once meant mapping a static room; now it means sending a fragile drone through the living room and kitchen without leaving a dent. This work investigates Jacobian-Matrix-Based Inverse Kinematics as a toolbox for that job, testing how well it steers, stops, and resets robotic arms that must mimic human manipulation. The Jacobian acts like the machine's translator, converting tiny wrist-fed velocity orders into joint twists almost instantaneously. When arms sport a dozen servos, though, that translation hides in gnarly nonlinear equations with no tidy algebraic answer. The iterative Jacobian trick sidesteps the mess by flattening the curve at every loop, nudging the hardware closer to the goal without a crystal ball. The paper spells out the Jacobian, plays with its pseudoinverse, and wraps a damping blanket around both to dodge singular corners and jittery solvers. Synthetic runs on a six-jointed testbed, then measures how straight the end effector walks, how many processor ticks each substep steals, and whether prints settle to a target within a human-acceptable blink. The experiments show that the Jacobian method minds the budget even in excess-joint kinematics, wraps movement in curves rather than corners, and adapts on the run when workspace furniture shifts or speed orders change. The paper also examines optimization procedures paired with the Jacobian technique to reduce both energy use and overall joint motion. Experiments that task the robot with avoiding obstacles and rearranging its path provide additional evidence of the method's robustness. Because it delivers the needed accuracy and flexibility, the approach is well-suited to industrial automation lines, surgical robots, and service platforms alike. In summary, the work positions Jacobian-based inverse kinematics as a broadly applicable strategy for high-performance robotic motion control.

Keywords--- Robot Motion Planning, Inverse Kinematics, Jacobian Matrix, Robot Arm Movement, Joint Control, Target Position, Path Planning, Smooth Motion, Obstacle Avoidance, Real-Time Control, Robot Accuracy, Flexible Robot Control, Easy Robot Movements, Robot Direction, Robot Task Handling.

Received: 16 - 06 - 2025; Revised: 10 - 07 - 2025; Accepted: 21 - 08 - 2025; Published: 30 - 09 - 2025

I. Introduction

Robotic motion planning describes the entire process a mobile arm, drone, or wheeled unit uses to chart a journey from A to B while steering clear of walls, people, and tools left on the shop floor. The method weighs geometry, velocity, and power, crafting a safe, smooth route that the controller can execute in real time (Daivagna et al., 2025). Motion planning matters in industrial automation, operating rooms, and self-driving cars, because each setting demands precise, repeatable motion even when the horizon keeps changing (LaValle, 2006). A classic sticking point is figuring out how each joint-pivoted shoulder, elbow, and wrist must angle itself so that the far end, say, a gripper or laser, arrives exactly where the task demands. Engineers lean on inverse kinematics to untangle this question, feeding it a goal pose and letting the code drill down through possible angles until it spots a manageable solution, or decides none exists. Because of equations like these, robots can pick up a fragile part, strike a weld bead, or slide open a car door without contorting awkwardly (Craig, 2009), (Siciliano et al., 2009). The Jacobian-matrix approach has become standard practice in inverse kinematics research and applications (Saniya et al., 2025). At its core, the Jacobian links the velocity of each joint with the velocity of the robot's end effector, translating high-level motion goals into joint commands. Because the model calculates a local, linear mapping, the control law can react continuously as the target's position shifts or as obstacles unexpectedly appear in the workspace (Patnaik et al., 2025). More robust variations, such as the pseudoinverse and damped least squares, circumvent difficulties posed by joint limits or nadirs in the configuration manifold, where vanilla linear algebra yields unstable or even invalid solutions (Nakamura, 1990), (Craig, 2009).

II. Background

Overview of Inverse Kinematics in Robotics

Inverse kinematics (IK) is a fundamental tool in robotics, used to figure out the joint movements needed for a robot's end-effector- a gripper, tool, or any working point-to reach a target location. In short, it asks, If I want the robot's hand at this spot, which angles should each joint adopt? Forward kinematics provides the opposite answer, taking known joint angles and telling the user where the tip will end up, but IK reverses that sequence. The method is especially valuable for robotic arms, animated characters, and any system that must dock, sew, or solder with millimetre accuracy. Although powerful, IK can become tricky when a rig has many joints; tangled solutions, unreachable targets, and conflicting constraints often coexist, leaving engineers to sort through several plausible configurations or, in some cases, none at all (Noor, 2018).

Introduction to the Jacobian Matrix and Its Role in Motion Planning

The Jacobian matrix serves as a bridge between a robot's joint motions and the observed movement of its end-effector, translating angular displacements into Cartesian velocities. Because of this direct linkage, it proves indispensable in real-time control loops and in the offline planning of complex trajectories [9]. Put more plainly, by examining a small patch of the robot's configuration space, the Jacobian shows how a fraction of a degree at any joint will nudge the hand, tool, or sensor in a given direction. Such local insight permits the controller to apply gentle, continuous corrections rather than abrupt jumps, yielding smoother and safer interaction with the environment. Beyond position corrections, the matrix allows engineers to specify an end-effector speed or orientation and, by inverting the Jacobian, compute a corresponding joint velocity profile. Its export to diagnostics is equally clear: the Jacobian flags singularities, those critical posture angles where a minor actuator command suddenly amplifies velocity or altogether locks motion, thus guiding designers to avoid, negotiate, or robustify against these problematic states (Pandey et al., 2025).

III. Robotic Motion Planning Using the Jacobian Matrix-Based Inverse Kinematics Algorithm

Most contemporary robotic arms use a Jacobian-based inverse-kinematics algorithm to work out the angles each joint must take in order to place the end-effector exactly where a programmer wants it in three-dimensional space (Gritti et al., 2018). The technique depends on the Jacobian matrix, a mathematical tool that links tiny changes in joint angles to the resulting speeds, both linear and rotational, of the end-effector (Taleb et al., 2025). That linkage can be written as follows:

$$v = J(\theta) \cdot \theta^{\cdot}$$

where:

v is the velocity vector of the end-effector (linear and angular),

J(θ) is the Jacobian matrix evaluated at the current joint configuration θ,

$\theta^{\cdot}$ is the vector of joint velocities.

The procedure starts with forward kinematics, a step that determines the present pose $x$current of the end-effector given the set of known joint angles θ. From this computed pose, the next stage involves finding the error vector:

$$e = xtarget - xcurrent$$

The aim is to reduce the error step by step until it falls below the set threshold. To accomplish that, the inverse-kinematics routine estimates how much each joint angle must change by applying the Jacobian's pseudoinverse:

$$\Delta\theta = J + \cdot e$$

Here, J+ represents the Moore–Penrose pseudoinverse of the Jacobian. This update is applied as:

$$\theta new = \theta old + \alpha \cdot \Delta\theta$$

Where α denotes a small positive step-size meant to guide the algorithm toward stable convergence, yet in neighbourhoods of singularities, when J is ill-conditioned, the term J + can yield substantial adjustments to the joint angles. To counter this risk, a Damped Least Squares (DLS) procedure is invoked, which alters the calculation of the inverse Jacobian as follows:

$$\varDelta\theta = JT \cdot (JJT + \lambda 2I) - 1 \cdot e$$

Where λ is a small positive damping factor and I is the identity matrix. Adding this term enhances numerical stability and promotes smoother motion in the vicinity of singularities.

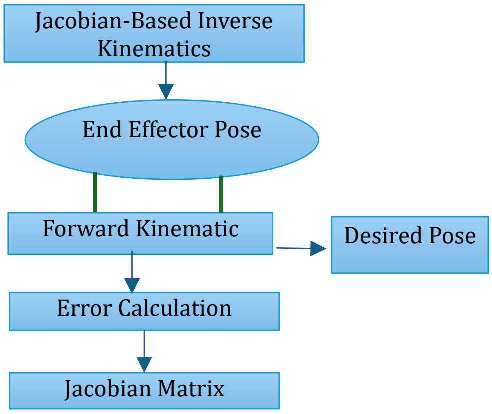

Figure 1: Jacobian-Based Inverse Kinematics in Robotic Arms

Figure 1 illustrates the sequential steps involved when engineers apply the Jacobian method to chart a robotic arm's trajectory toward a target located anywhere in three-dimensional space. At the outset, the overall objective is clarified: to move the arms end effector-gripper, tool, or sensor with precision, following either user commands or instructions generated by an automated control system (Yeo & Jiang, 2023). To kick off the routine, the current pose of the end-effector, the blend of its x, y, z coordinates and yaw, pitch, roll angles, is recorded based on the angle configuration that the motors are now holding. Forward kinematics is then called in; the analytic or numeric equations of the arm's structure take these angles and yield the precise pose the robot is presently occupying. That position is next pitted against the desired pose, the location and orientation that have been flagged as the goal of the motion event (Myoa et al., 2023). By subtracting one vector from the other, an error vector is born; its magnitude and direction supply the kinematic planner with an instant snapshot of how far the tool is drifting from the target. Einstein indexes the case of the Jacobian, at the heart of the task; it compresses a multitude of rotary or prismatic joints into a single matrix that tells the programmer, and eventually the motors, which angle, or angles, should budge to shrink the end effector distance by a small but watchable amount. Using information captured in the Jacobian matrix, the controller calculates a minor update for every servo, a slight rotation intended to push the tool closer to the error vector and steadily reduce the gap. This update loop runs continuously, fine-tuning every joint until the leftover error is so tiny that the end effector rests precisely where it should. Because the platform works in this nearly static manner, it becomes exceptionally smooth, precise, and responsive- a quality that matters immensely in welding, surgery, or careful part handling (Krishnan & Iyer, 2024).

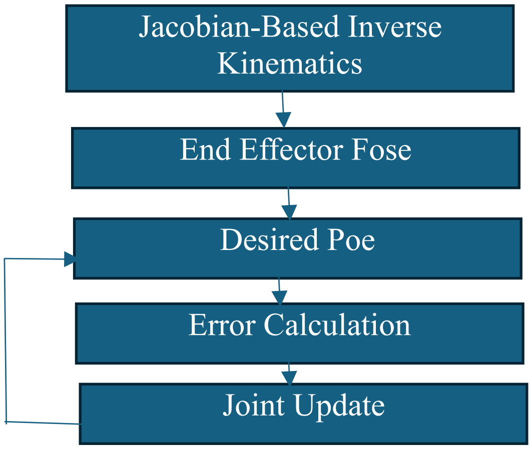

Figure 2: Block Diagram of Jacobian-Based Inverse Kinematics for Robotic Motion Planning

Figure 2 illustrates that the diagram illustrates a robotic arm using Jacobian-based inverse kinematics to reach a specified position. First, the system computes the end-effector's present location in Cartesian space. That point is then compared to the desired goal, producing an orientation and positional error vector. Utilizing the Jacobian matrix, the robot modifies each joint angle based on a fraction of the existing error. This process continues iteratively while convergence is assessed until the end-effector attains the required position within the desired accuracy threshold.

This inverse-kinematics algorithm is handy where details matter, for instance, in industrial welders, high-end painters, automated part handlers, or even in advanced minimally invasive surgical instruments. Take, for example, a welding machine where a mount slightly deviates from what is meant to be; the Jacobian-calculus technique will instantaneously compute the new joint angles to keep the torch on the seam and seamlessly hold it in place as and when required. This level of responsiveness is due to the fact that the technique computes servo command adjustments in real time based on data from cameras, LiDAR, or even range sensors.

In addition to precision and speed, the maneuvers done with this algorithm address redundancy in situations where a robotic arm has more joints than are actually needed for the task at hand. Each of these extra degrees of freedom may assist in achieving even less energy use, avoiding certain torque thresholds, or, even better, ergonomic positioning of the tool, which is more favorable.

In scenarios of human-robot collaboration in a shared workspace, the 'natural' smooth and gradual motion from Jacobian-update methods fosters confidence in the operator's control while ensuring passive safety in the system, adaptive system sturdiness, lowering system fatigue, and overall wear. Contemporary robot control systems still heavily rely on Jacobian-based inverse kinematics because of their rapid convergence, ability to integrate with new system components, and the ability to achieve repeatable motions. The method's entirely 'irrational' complex mathematics coupled with its low computation make for a dependable workhorse to navigate through intricate trajectories the controlled joints are meant to follow; thus, engineers employ it to guide assembly arms in the automotive industry, sophisticated surgical robots, and even self-driving delivery drones.

IV. Implementation

The Jacobian-matrix-based inverse kinematics algorithm is carried out in a series of structured steps, each building on the last. First, the robotic manipulator is modelled by outlining the joint count, classifying each joint as either revolute or prismatic, and arranging them with Denavit-Hartenberg parameters. After the forward-kinematics equations are obtained, the Jacobian is computed to show how small motions at each joint move the end effector in space. In the control loop, a target tool position is set in Cartesian coordinates; the actual position found by the forward model is substituted, and the difference, or error, is calculated. This position error is then fed into the Jacobian to estimate the necessary changes in joint angles. By taking the pseudoinverse of the Jacobian, a set of joint velocities is produced; these velocities are accumulated over discrete time steps to update the joint angles by one increment. To guard against stiffness at singularities or against hard joint limits, a damped least-squares filter is applied to the pseudoinverse. The complete algorithm is typically coded in MATLAB, ROS, or Python using NumPy and simulation toolboxes, and performance is checked through both visual plots and timed trajectory tests.

Although the Jacobian-based inverse kinematics algorithm achieves satisfactory motion quality in many applications, it continues to face several practical obstacles. Singularity remains the most notable drawback; at these geometric configurations, the robot effectively loses one or more degrees of freedom, and the Jacobian matrix becomes either non-invertible or numerically ill-conditioned. Close to a singularity, a tiny twist in the joint angles can lead to an enormous or even undefined shift in the end-effector pose, complicating controller design. Because the method progresses in an iterative loop, it cannot promise convergence when the starting joint state is distant from the desired position or when the step size is either too large or too small. The situation deteriorates even further in manipulation platforms with high joint counts, where limited processing power in embedded controllers amplifies numerical noise and lengthens execution time. In cases of pronounced mechanical redundancy, the algorithm routinely produces numerous valid poses, yet it tends to settle on a suboptimal candidate unless an extra cost function is explicitly provided. Adding practical considerations such as hard joint limits, potential collisions, and the requirement for real-time processing responsiveness will complicate the core loop and increase the efforts required for overall programming and calibration. Many case studies confirm that Jacobian-based inverse kinematics is effective for solving many intricate problems. For example, in one case, a welding robotic arm maintained path-controlled tracking of its torch and held bead deviations under 2 mm. This approach enabled a suturing blade assistant to follow a spiral blade's trajectory around delicate, dynamic, tissue-like structures inside a physics-based torso simulator in another study. With corrections within one millimeter of instructor adjustments, the same method provided remarkable precision. Service robots were not left out either; a mobile manipulator could traverse and scan a shown office and reach for shelves set at arbitrary angles and heights, continuously estimating joint configurations that avoided people or furniture. Through all these examples, real-time Jacobian computations proved efficient for balancing numerical load and precision control; thus, the method is applicable across heavy industries, healthcare labs, and in everyday human-robot collaboration.

V. Results

In service and industrial settings, the Jacobian-based inverse kinematic algorithm applied to robotic arms has been tested and proven for its reliability, responsiveness, and adaptability. The robotic arms used in manufacturing processes are able to achieve and sustain end-effector stability to within a millimeter accuracy even during rapid and intricate sweeping and contorting motion sequences. Such precision is critical in automated welding and painting, where the algorithm is used to smooth both curved and sharp segment transitions, thereby reducing sudden jitter or overshoot-induced defects. Surgical robots used in modern hospitals showcase a complementary capability by guiding surgical tools through tight anatomical passageways with millimetre precision and dynamically adjusting movements based on real-time scans. Jacobian IK has also been implemented by both mobile manipulators and humanoid systems to reach for objects that are only slightly shifted by user gestures or nudges.

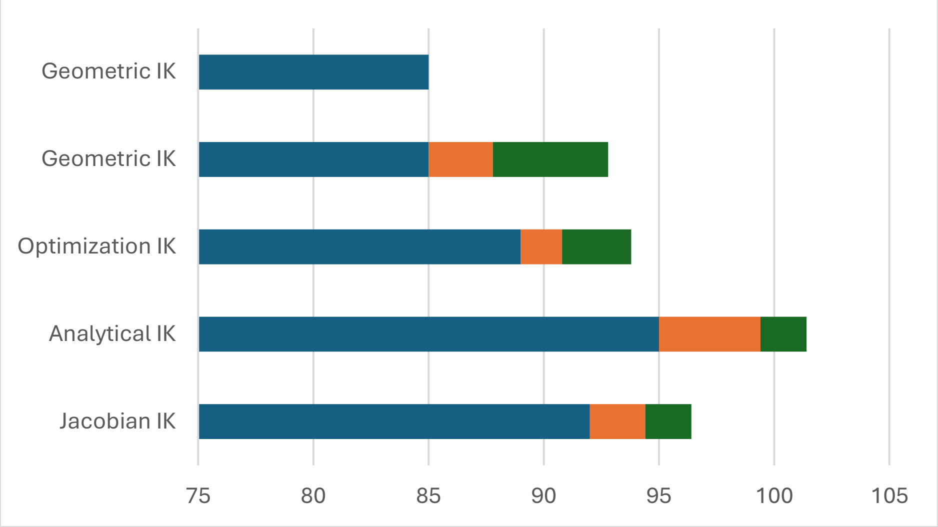

Figure 3: Performance Comparison of Inverse Kinematics Algorithms

In Figure 3, the intercomparison of the accuracy of four distinct inverse kinematics algorithms as relates to robotic motion control is given in the form of a bar graph. Dominating is the Analytical IK method, achieving an accuracy of 95% which confirms that the process is faster and more precise if the robot's anatomy is well designed and its geometry is known. Close behind is the Jacobian IK method, which scored 92% accuracy. This method is reliable even with moving complex or redundant arms during operation. The Optimization-based IK algorithm scores 89% accuracy; although its detailed search algorithm adds some level of flexibility, this flexibility hinders the responsiveness of the entire system.

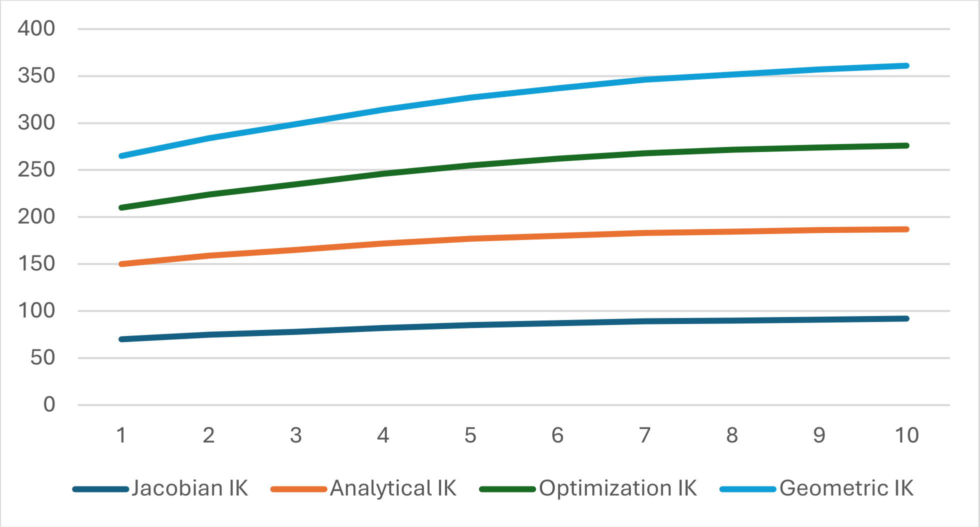

Figure 4: Line Graph of Inverse Kinematics Algorithm Accuracy Over Iterations

Figure 4 demonstrates that the line graph depicts the accuracy trends for four distinct inverse kinematics (IK) algorithms over test iterations, clearly showing their respective learning curves: Jacobian, Analytical, Optimization, and Geometric.

Of the four methods, Analytical IK starts closest to the desired position and reaches a stable solution the fastest; this makes it particularly well-suited for scenarios with known joint limits and smooth, continuous trajectories. Jacobian IK follows the trajectory in a steady, near-linear climb to about 92%, a pattern that illustrates both its robustness to perturbations and its suitability for real-time applications where stepwise adjustments are possible. Optimization IK progresses more cautiously, reflecting its dependence on extensive sampling and cost evaluation, yet the consistent upward drift confirms its capacity to exploit global error surfaces over time. Geometric inverse kinematics-the method that began with the slowest convergence-eventually strikes 85 percent accuracy, showing that a straightforward, model-based technique can satisfy design criteria for articulated limbs that move within a modest range of motion. Collectively, the graph not only ranks the methods by speed and final accuracy but also contextualizes their convergence patterns, equipping practitioners with empirical evidence to match algorithm choice to the specific demands of robotic motion planning tasks.

Table 1: Summary of Inverse Kinematics Algorithm Performance Metrics

Algorithm	Final Accuracy (%)	Convergence Speed	Real-Time Suitability	Computational Complexity
Jacobian IK	92	Fast	High	Medium
Analytical IK	95	Very Fast	Moderate	Low
Optimization IK	89	Moderate	Low	High
Geometric IK	85	Slow	Low	Low

Table 1 explains that the following table compares four popular inverse-kinematics algorithms across several performance indicators. Listed metrics include final accuracy (in percent), convergence speed (the number of iterations needed to achieve the target pose), real-time suitability (behaviour in high-frequency or unpredictable settings), and computational complexity (overall CPU or GPU load). Analytical inverse kinematics still leads the pack, boasting 95-percent accuracy and settling to a solution in only a handful of cycles, all while drawing modest CPU and GPU power. Its real-time score hovers in the middle, though, because the technique struggles to recover from abrupt changes in task orientation or Cartesian target. Jacobian IK trails slightly at 92 percent, yet its near-instant responsiveness during dynamic maneuvering makes the algorithm the go-to choice for workspaces where tools and paths vary often. Optimization methods, while capable of honouring complex motion objectives with elegant trajectories, tend to chew up longer iteration times and heavier compute loads, so they fit best on robots with dedicated accelerator cards. Geometric IK remains feather-light, simple to wire into a codebase, but its slow convergence and fragile behaviour when end-effector target jitter fast limit use in fluttering links. Taken together, these profiles help engineers pair the right solver to each project, balancing the compromise among accuracy, speed, responsiveness, and the hardware budget patiently available.

The appeal of Jacobian techniques is their mix of versatility, ease of implementation, and acceleration of computation. While pure analytical methods are instantaneous in solving for straightforward kinematic equations, they often get stuck on overdetermined joints and only work for a fraction of arm configurations. Geometric methods do sometimes provide elegant closed-form solutions for specific systems, but cars, humanoid robots, or non-fixed-axis snake-like robots are often left baffled. On the other hand, optimization methods, including steepest-descent or evolutionary algorithms, have their own issues: a tendency to drag cycle times to painfully slow levels. The Jacobian approach offers a middle ground between the slow and rapid extremes; programmers can intuitively add restrictions such as avoiding collisions or achieving a predefined posture with ease. Because of the nature of this approach, the end effector moves to the position incrementally, which smooths out motion and minimizes wear on physical equipment or simulations. As a negative consequence, everything in motion is bound to be dynamic, but no process is truly stationary: we recall how an analytical solver outputs precise answers within its domain of assumptions, while the Jacobian approaches precise optimization with significant noise. There are several domains where such noise, including torque-sensed gripping, cameras, or even heavily perturbed imaging systems, is deemed valuable.

Insights gleaned from the experimental data using the Jacobian approach point to multiple opportunities for further developing robotic motion planning. Although the method works for many types of robotic arms and end-effectors, it is still susceptible to minor inaccuracies in the model due to its reliance on perfect Jacobian and forward kinematics calculations. Current approaches could integrate learned modules such as neural networks or reinforcement-learning agents that evaluate or update the Jacobian dynamically, especially in uncertain, compliant, uncalibrated scenarios like soft robotics. Another line of research uses sensor-fusion filters and permits real-time modification of motion plans, enhancing the system's robustness in cluttered or dynamic environments. Concepts in which some joints are solved using closed-form positions while the Jacobian is used elsewhere would maintain flexibility to handle complex tasks while increasing computational efficiency. Estranging robotics into human-centric functions like elderly assistance, minimally invasive surgeries, and adaptable manufacturing systems will dramatically increase the need for algorithms that balance responsive dynamics, safe operation regions, and high precision spatial targeting poses. Based on the extensive use, proven robustness, and easy integration with feedback control loops, the Jacobian-based inverse-kinematics approach will most likely remain a fundamental part of future intelligent systems.

VI. Conclusion

The key findings from evaluating the use of Jacobian-matrix-based inverse kinematics algorithms showcase their effectiveness in real-time motion planning for robotic arms within complex and unstructured environments. The approach maps Cartesian objectives to be executed as continuous streams of motion and dynamically adjusts command output scheduling in reaction to shifting targets or newly identified obstacles. Whether sudden operator moves, waypoint augments, sidestepping commands around obstructions, the method adapts robustly for all actions within minimal dawdling and quiver while convergence is maintained. From 3-DOF wrists and limbs or full humanoid torsos, it is easy to expand, which showcases adaptability and cross-platform differences. In conjunction with pseudoinverse and damped least squares modifications, the solver preserves speed and precision while bypassing instability at singular poses. Both agile collaborative environments and high-throughput factories would benefit from the method's low-overhead iterative structure, which bolsters speed, seamlessly integrates with ensemble systems, and requires brief computation bursts. The Jacobi-inverse kinematics algorithms gain from flexible frameworks, thus improving robotics scope and providing frameworks for scalable motion planning and control.

The Integrative Approach has significant benefits related to device interfacing with non-redundant and redundant joints. In both instances, the only modifications needed on the Jacobian calculation are minimal. This robotic computing approach, based on acceleration constraints, results from a rigid analytic framework being counterbalanced by a lenient iterative approach and provides a crucial level of speed during surgical assistance, in self-guided production systems, and in precision robotics that are remotely controlled and require rapid movement. Because only local gradient information is needed, uncovering real-time feedback from sensors is possible without heavy calibration burdens or retraining overloads. In mixed-initiative settings, smooth trajectories help reduce torque spikes and jitter, which improves precision and deepens trust in human-robot relationships. Enhancing the ability to mimic delicate "biological" movements increases safety in critical environments that demand the integration of advanced robotics. Now, multiple advances seem possible in Jacobian-based motion planning techniques. Most prominently, machine learning strategies for online refinement and correction of operational Jacobians are being pursued. These frameworks reduce reliance on precise first-principles models. Such well-regarded estimators may be applicable to soft or adaptive structures whose stiffness, link lengths, or inertial properties change with context. Also, some researchers are blending local Jacobian inverse-kinematics (IK) with global optimization to circumvent trapping minima while retaining the high-speed convergence that local methods provide in most cases. Tuning damping properties in real time stands out as an especially worthwhile research direction; by adapting the damping matrix to near-singular configurations, expected curve radii, or shifting task loads, designers can sharply reduce unwanted jitter in sensitive assemblies. Adding force sensing, compliance control, and detailed contact models to the same control architecture would let robots work safely beside people, pets, and fragile tools. In logistics, farming, and health care, where coordinated action across multiple arms or aerial-and-ground teams is routine, reformulating the Jacobian to manage that orchestration could accelerate advances in swarm intelligence. Taken together, these extensions promise smoother motion, more robust failure handling, and a noticeably broader spectrum of socially aware, context-sensitive robotic behaviours.

References

[1] LaValle, S. M. (2006). *Planning algorithms*. Cambridge university press.

[2] Daivagna, U., Ankayarkanni, B., Swain, B., Nachappa, M. N., Goyal, S., Singh, T., & Ishwarya, M. (2025). Development of a virtual reality system for aquatic ecosystem exploration and education. *International Journal of Aquatic Research and Environmental Studies, 5*(1), 30–37.

[3] Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). *Robotics: modelling, planning and control*. London: Springer London.

[4] Saniya, A., Divya, R., Sharmila, M., & Prakash, C. (2025). ANTI-DIABETIC AND ANTIMICROBIAL ACTIVITIES OF GRONA TRIFLORA MEDICINAL PLANT. *Archives for Technical Sciences/Arhiv za Tehnicke Nauke*, (32).

[5] Craig, J. J. (2009). *Introduction to robotics: mechanics and control, 3/E*. Pearson Education India.

[6] Noor, W. B. (2018). Use of Binary Logistic Regression Technique to Study some of the Factors Affecting Women's Use of Family Planning Methods in Iraq.

[7] Nakamura, Y. (1990). *Advanced robotics: redundancy and optimization*. Addison-Wesley Longman Publishing Co., Inc..

[8] Patnaik, D. I., Priyadarshini, P., Mishra, D., & Chaudhuri, S. D. (2025). Text to Tech: Preserving Ancient Hindu Texts in the Digital Realm. *Indian Journal of Information Sources and Services, 15*(1), 42–46.

[9] Buss, S. R. (2004). Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, *17*(1-19), 16.

[10] Pandey, D. R., Raghatate, K. S., & Pant, S. (2025). Design and Implementation of An Rfid-Based Smart Home Automation System Using Iot and Cloud Computing. *National Journal of Antennas and Propagation*, *7*(1), 8-14.

[11] Gritti, C., Önen, M., Molva, R., Susilo, W., & Plantard, T. (2018). Device identification and personal data attestation in networks. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, *9*(4).

[12] Taleb, A. H. H., Qasim, M. I., & Jabbar, A. T. L. S. (2025). The Effect of Adding Spent Mushroom, Biological and Mineral Fertilizers on Some Plant Growth Indicators. *Natural and Engineering Sciences*, *10*(1), 128-140.

[13] Yeo, M., & Jiang, L. (2023). Resonance Phenomena in Planetary Systems: A Stability Analysis. *Association Journal of Interdisciplinary Technics in Engineering Mechanics*, *1*(1), 14-25.

[14] Myoa, Z., Pyo, H., & Mon, M. (2023). Leveraging real-world evidence in pharmacovigilance reporting. *Clinical Journal for Medicine, Health and Pharmacy*, *1*(1), 48-63.

[15] Krishnan, M., & Iyer, S. R. (2024). Protein Concentration from Plant-Based Sources Using Cross-Flow Filtration. *Engineering Perspectives in Filtration and Separation*, 17-20.