Redundant Manipulators and Singularity Avoidance: Strategies for Optimized Performance

Mullainathan V.H. *Department of Robotics andAutomation*
PSG College of Technology
Coimbatore, India

Srimathi P.
Department of Robotics and Automation
PSG College of Technology
Coimbatore, India

Anbarasi M .P.
Department of Robotics and Automation
PSG College of Technology
Coimbatore, India

Abstract - Robotic manipulators equipped with redundant degrees of freedom offer increased flexibility and dexterity but are prone to encountering singular configurations that hinder performance and control. This paper introduces novel strategies for singularity avoidance in redundant manipulators, focusing on optimizing their performance. Drawing inspiration from existing research, an innovative approach based on the Monte Carlo method for singularity analysis is proposed. By randomly mapping the robot's workspace and identifying points closest to a given trajectory, detection and characterization of singularity states is determined. Unlike traditional inverse calculation methods, this approach reducescomputational demands while providing a clear graphical representation of singularity occurrences. Through rigorous mathematical analysis and illustrative graph ,the effectiveness of this method in singularity avoidance is presented. Furthermore, exploring practical applications through case studies involving various robot configurations is possible. This research contributes valuable insights and tools for improving the operational efficiency and adaptability of redundant manipulators in diverse robotic applications.

Keywords - Redundant manipulators, Singular configurations, Universal Robots UR5e, Kinematic redundancy,Jacobian matrix, Null space, Geometric constraints, Configuration space, Mathematical properties, Joint angle solutions, Finite State Machine (FSM), Monte Carlo method, Probability assessment, Workspace analysis, Robot system design, Operational efficiency, Human-robot collaboration, Robotic adaptability, Industrial automation.

A. INTRODUCTION

In the domain of robotics and automation, the UniversalRobots UR5e has emerged as a versatile collaborative robot (cobot) known for its seamless integration into diverse industrial applications. This cobot, with its 5 kg payload capacity, excels in handling lightweight to medium-sized objects and extends its reach beyond that of an average human worker. However, the efficiency of the UR5e, like many redundant manipulators, is challenged by singular configurations—positions where the robot loses degrees of freedom and encounters uncontrollable movements.

This paper delves into the realm of redundant manipulators, singularity analysis, and avoidance strategies,with a focus on enhancing the UR5e's performance. The aim is to identify and address challenges posed by singular configurations, streamline practical applications, and reduce computational complexities. By doing so, it contributes to the knowledge base in robotics and improve the operationalefficiency of cobots like the UR5e.

B. FUNDAMENTAL CAUSES OF SINGULARCONFIGURATIONS

Kinematic Redundancy: Singularities often arise in redundant manipulators, such as the UR5e, because they have more degrees of freedom (DOF) than are necessary to perform a given task. This redundancy allows the robot to adopt multiple configurations to achieve the same end-effector position.

Jacobian Matrix: Singularities are closely tied to the Jacobian matrix, a mathematical construct used to relate joint velocities to end-effector velocities. When the Jacobian matrix becomes singular, it means that there are infinite solutions to the inverse kinematics problem, making the robot unable to uniquely determine joint velocities for a given end-effector velocity.

Null Space: In redundant manipulators, the null space represents the subspace of joint configurations that donot affect the end-effector position. Singularities often occur when the robot approaches regions within its null space. These are known as "self-motion singularities."

Geometric Constraints: Singularities can also be influenced by geometric constraints in the robot's workspace. For instance, when the robot approaches extreme joint limits or tries to reach a position outside its reach, singularities may occur.

Configuration Space: Singularities can be visualized in the configuration space, where each point represents a unique joint configuration. In singular configurations, certain directions in the configuration space become unattainable, leading to limited motion or loss of DOF.

Mathematical Properties: Singularities are often characterized by mathematical properties such as a determinant of the Jacobian matrix becoming zero or joint axes aligning. These conditions indicate that the robot has reached a singular configuration.

C. KINEMATIC REDUNDANCY AND SINGULARITY IN UR5E

Kinematic redundancy in robotic manipulators means having more degrees of freedom (DOF) than strictly neededfor a task, evident in extra joints or axes. This redundancy is valuable for adaptability, enabling robots to find multiple solutions for positioning. It aids obstacle avoidance, makingmotion planning in cluttered spaces safer and more efficient. However, redundancy also poses challenges, particularly related to singular configurations, where some DOF are lost,complicating control. Despite complexities, redundancy allows optimization and enhances human-robot collaboration, making it a vital aspect of robotic versatility and adaptability.

The UR5e's 6-DOF design (Fig 1) offers extra joint variables beyond the minimum needed for precise end- effector positioning. This inherent kinematic redundancy grants the UR5e flexibility to adapt in dynamic environments and optimize tasks. While advantageous, redundancy introduces control complexities and singularity challenges, necessitating careful management.

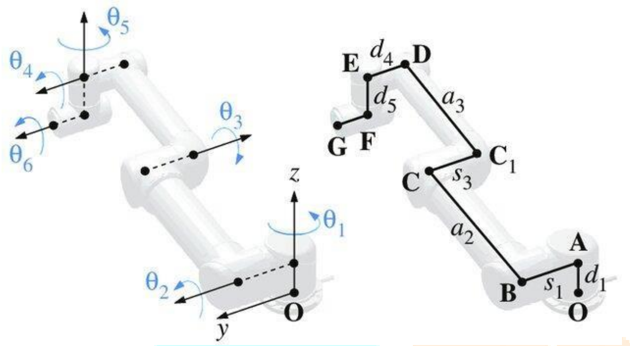


Fig 1 Axis Notation

Kinematic redundancy in robots like the UR5eoffers diverse advantages. It enables multiple solutions for a single task, facilitating adaptability and obstacle avoidance. Redundant robots reach inaccessible areas and perform complex tasks with enhanced dexterity, optimizing energy usage. Efficient path planning,human-robot collaboration, and safety features add to their versatility, making them suitable for evolving tasks.

i) Definition of Singularity in Robotics:

In robotics and manipulators, singularity refers to a specific configuration or state where the robot loses some ofits degrees of freedom (DOF). In simpler terms, it's a criticalpoint in the robot's motion space where its motion becomes constrained, leading to a significant reduction in its ability to move freely. Singularities are often characterized by the robot's joints aligning in a way that limits its motion. Theseconfigurations are mathematically significant because they result in a non-invertible Jacobian matrix, making it impossible to compute the robot's joint velocities uniquelyfrom its end-effector velocities.

ii) Challenges of Singular Configurations:

Singular configurations are problematic for several reasons. Firstly, they lead to a loss of degrees of freedom, meaning the robot can't move in all desired directions, limiting its maneuverability. Secondly, at or near singularities, small changes in the robot's joint angles can lead to significant changes in its end-effector position and orientation, making control challenging. This increased sensitivity can result in jerky and unpredictable robot movements. Additionally, singularities can make path planning and trajectory generation difficult, as the robot must actively avoid or transition through these configurations to perform tasks accurately and efficiently.

D. Robot Workspace

The UR arm's workspace is spherical, depicted in working area diagrams as consisting of two concentric circles. The inner circle is labeled as the "Recommended Reach," while the slightly larger outer one is the "Max. Working Area." At the center of this spherical workspace, precisely above and below the base joint, there exists a column. This column imposes certain movement restrictionson the robot within its interior.

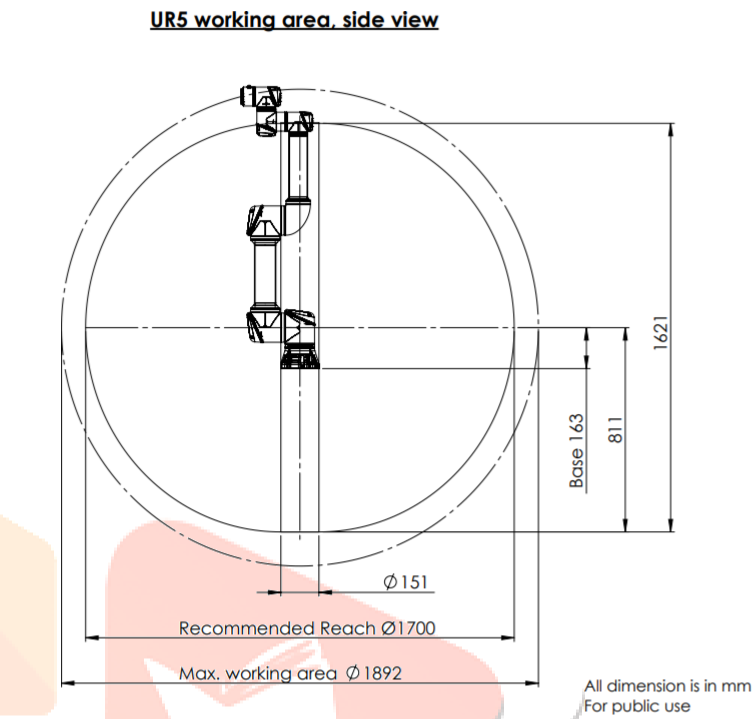


Fig 2 Reach representation of UR5

Boundary of the Outer Workspace: The illustration below demonstrates that within the suggested reach sphere (depicted in blue), the robot can precisely control the tool's movement to virtually any position with considerable flexibility in orientation. As the work extends beyond the suggested reach but remains within the maximum working area (shown in grey), the robot can access most positions; however, there are constraints on the tool's orientation. This limitation arises because, in certain situations, the robot lacks the physical reach to extend far enough.

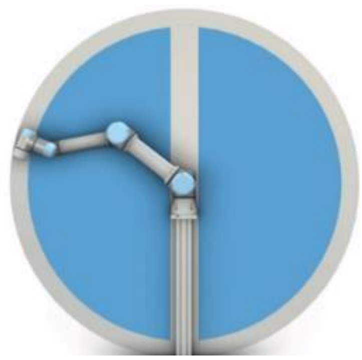


Fig 3 Reach and Work Envelop

To prevent operating outside the recommended workspace, arrange the equipment in proximity to the robot.In cases where this arrangement is not feasible, consider opting for a UR robot with an extended reach to accommodate your specific operational needs.

Boundary of the Inner Workspace: It is advisable to refrain from executing robot motions within the column situated directly above and below the robot's base (as depicted in grey in the visual representation). Within this region, a multitude of positions and orientations becomes

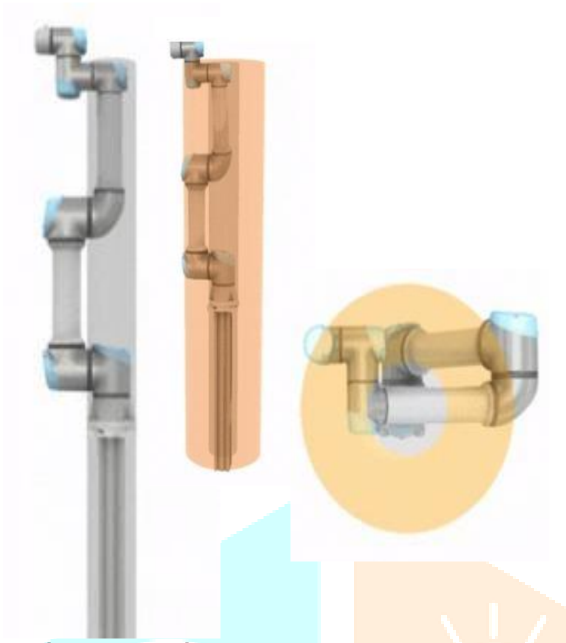


Fig 4 Regions Robot cannot Reach

physically unreachable due to the specific arrangement of the robot arm's joints. Furthermore, you may encounterchallenges when attempting linear movements in the space immediately outside of this cylindrical area (as portrayed inorange). This is because maintaining a relatively slow tool speed necessitates an exceptionally high base joint rotation speed, rendering certain tool movements unattainable or unsafe.

Plan the robot task layout to minimize the need foroperations within or in close proximity to the central cylinder. If working in this area is unavoidable, opt for the "Use joint angles" option with MoveJ instead of MoveL whenever feasible. This choice eliminates the need for kinematic conversion and is less impacted by singularities

Another consideration is mounting the robot base on a horizontal surface, which can reorient the central cylinder from a vertical to a horizontal position, potentially relocating it away from the task's critical areas. Wrist Alignment Singularity: In UR robots, the shoulder, elbow, and wrist 1 joints share a common rotational plane, as indicated by the numbered arrows 1, 2, and 3 in the visual representation.
However, when the motion of wrist joint 2 is aligned (numbered 4) with this same plane, achieved by setting it to an angle of 0 or 180 degrees,constraints on the robot's range of motion is imposed.

This limitation applies universally across the workspace, irrespective of its specific area.

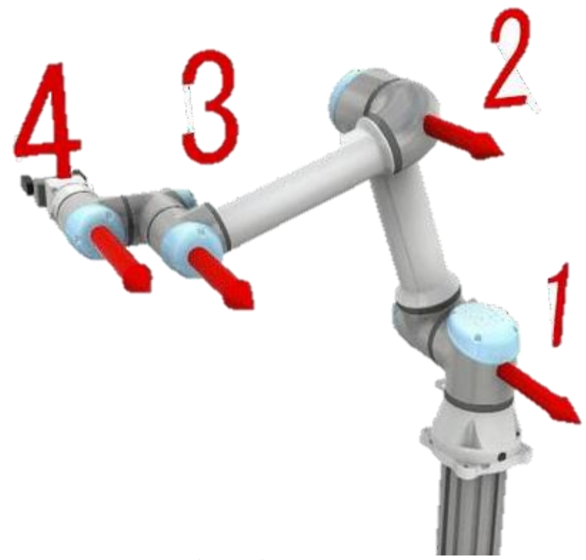


Fig 5 Joints with Same direction of axis of Rotation

Design the robot task layout to avoid the need for aligning the robot wrist joints in this specific manner. Alternatively, adjust the tool's direction to allow it to point horizontally without requiring the problematic wrist alignment. Additionally, if linear motion isn't essential, consider using MoveJ with the "Use joint angles" option to mitigate the singularity issue.

E. SINGULARITIES AND THE JACOBIAN MATRIX:

The ur5 robot's inverse kinematics involve determining the joint angles required to position its end effector accurately. The robot used in the experiments is described as a 6-degree-of-freedom (dof) cobot with rotational joints, and its kinematics are expressed usingdenavit-hartenberg (dh) parameters. These dh parameters provide a geometrical description of the robot's joints and links.

i	α_i(rad)	$a_i(mm)$	$d_i(mm)$	θ_i
1	$\pi/2$	0	$d_1 = 89.2$	θ_1
2	0	$a_2 = 425.0$	0	θ_2
3	0	$a_3 = 392.0$	0	θ_3
4	$\pi/2$	0	$d_4 = 109.3$	θ_4
5	$-\pi/2$	0	$d_5 = 94.75$	θ_5
6	-	-	$d_6 = 82.5$	θ_6

Table 1 DH Parameter

The transformation matrix ^0T is fundamental for IK calculations, representing the position and orientation of the robot's end-effector relative to the base frame. Theelements of this matrix are computed using trigonometric functions and the DH parameters. The IK solution for the UR5 involves a set of equations that relate various joint angles and trigonometric functions.

$$^0_6T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Shoulder Singularity: Occurs when the endeffector can't move along the $z6$ direction. It's associated with $\theta2$, $\theta3$, and $\theta4$ angles.

Wrist Singularity: Happens when $s5$ (sin of $\theta5$) equals zero, meaning $\theta5$ is either 0 or $\pm\pi$. This results in $z4$ and $z6$ being parallel.

Elbow Singularity: Occurs when $s3$ (sin of $\theta3$) equals zero, which happens when $\theta3$ is 0 or $\pm\pi$. In practice, only the case where $\theta3$ equals 0 is physically possible.

F. ALGORITHMS TO SELECT VALID SOLUTIONS

Two algorithms are provided to determine suitablejoint angle solutions that steer clear of singularities. Thesealgorithms are engineered to calculate a set of angles that position the end-effector as desired while keeping joint motion to a minimum.They take into account several criteria for validation:

a) Complex angles are rejected.

b) $\theta5$ is considered invalid if $|s5|$ is too small.

c) $\theta3$ is considered invalid if $|s3|$ is too small or if it's complex.

d) $\theta2$ and $\theta4$ are not valid if a specific condition involving trigonometric functions holds.

e) Outer workspace limits are checked, ensuring that $\theta3 \neq 0$ and $\theta4 \neq \pi/2$.

i) **Algorithm 1** is designed to find a set of joint angles that positions the robot's end effector to a specified location while minimizing the overall movement of its joints. Unlike some other methods that consider various objectives such as avoiding singularities, staying within joint limits, or navigating around obstacles, this algorithm focuses on minimizing joint displacement. While different criteria can be used for selecting the final joint angles based on specific objectives, this algorithm defaults to minimizing thetotal joint movement.

Here are the steps of Algorithm 1:

1. Calculate both possible solutions for $\theta1$, anddiscard any complex angles.
2. Use the previously computed $\theta1$ values to calculate $\theta5$. Reject sets of $\theta5$ values that are consideredinvalid.
3. Compute $\theta6$ for the remaining sets.
4. Calculate $\theta3$ values and validate them. Discard solutions with unacceptable angles.
5. Finally, compute $\theta2$ and $\theta4$, and eliminate sets with invalid angles.

The algorithm then selects the solution with the least difference from the current joint positions, determined using the following equations:

$$\Delta\theta_i = \theta_{i,p} - \theta_{i,j}$$

$$diff_j = \sqrt{\sum_{i=1}^{6} \triangle\, \theta_i^2}$$

In these equations, "p" represents the previous joint positions, "i" denotes a specific joint, and "j" represents the computed joint angle set.

ii) **Algorithm 2** is designed to compute a set of joint angles that positions the robot's end effector at a specified location while minimizing joint movement and ensuring that the solution does not result in singular configurations. It employs a Finite State Machine (FSM) to systematically compute and validate these angles.

The FSM, illustrated in Fig.6, guides the algorithm through a sequence of states to achieve this task. Each state computes one or two joint angles, selects the option that minimizes joint movement, and checks for angle validity.If an angle is deemed invalid, the algorithm revisits a previous state and adjusts the angle.

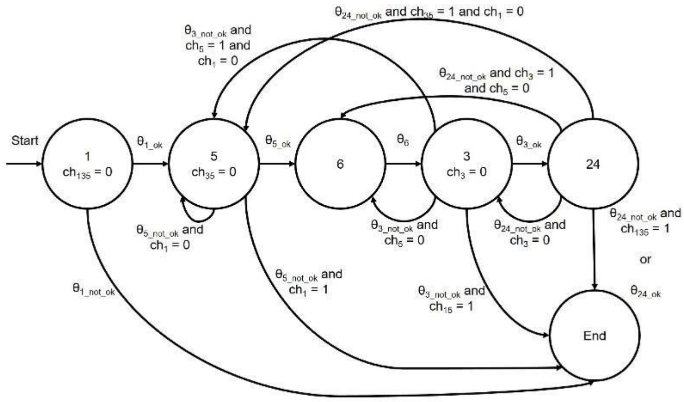


Fig 6 Finite state machine

Here's a breakdown of the states in Algorithm 2:

1. **State 1:** Compute and validate $\theta1$ values, selectingthe one closer to the previous $\theta1$. If a valid $\theta1$ is found, proceed to State 5.
2. **State 5:** Similarly, to $\theta1$, compute and validate the two possible angles for $\theta5$, choosing the one closerto the previous value. If an acceptable $\theta5$ is found,move to State 6. If computed angles are invalid, retry State 5 using the other $\theta1$ (if it hasn't been modified).
3. **State 6:** Compute $\theta6$, and then transition to State 3
4. **State 3:** Calculate and validate both $\theta3$ values, selecting the one closer to the previous $\theta3$. If an acceptable $\theta3$ exists, move to State 24. It's important to note that verifying one of thecalculated angles is sufficient. If $\theta3$ is invalid,proceed to one of the following states:

State 6: If $\theta5$ hasn't been modified, use the other $\theta5$ value.

State 5: If $\theta5$ has changed but $\theta1$ hasn't, test the other possible angle for $\theta1$.

End: If, after changing $\theta1$, both possibilities for $\theta5$ result in unacceptablevalues for $\theta3$.

5. **State 24:** In this state, calculate $\theta2$ and $\theta4$. If the resulting set of angles is valid, the algorithm has found a solution, and the next state is "End." Otherwise, proceed to one of the following states:

State 24: If $\theta3$ hasn't been modified, adjust it, and repeat the state.

State 6: If both $\theta3$ angles have been used,and only one $\theta5$ has been used, test the other possible $\theta5$.

6. **State 5:** If $\theta5$ has changed, and both $\theta3$ options have been tested but $\theta1$ hasn'tbeen modified, use the other angle for $\theta1$.**End:** If it's impossible to find a valid set of angles to reach the desired pose even after changing $\theta1$,$\theta5$, and $\theta3$.**End:** This final state is reached when avalid set of angles is found or when it becomes impossible to find one.

G. MONTE CARLO METHOD

Monte Carlo methods are a class of computational techniques used to solve complex problems through randomsampling. In these methods, a large number of random samples or simulations are generated to approximate a solution or evaluate the behavior of a system. Each sample represents a possible outcome, and by aggregating the results of many such samples, statistical estimates or solutions to intricate problems can be obtained. Monte Carlomethods are incredibly versatile and find applications in various fields, including physics, finance, engineering, and even robotics.

Practical Implication for Robot System Design:
In the context of assessing the probability of singular point occurrence, the Monte Carlo method proves invaluable. This approach leverages random sampling to generate configurations of the robot's arms based on even probability distributions within defined movement intervals. Each configuration is then statistically evaluated for itsproximity to a singular state, a critical area that hinges on the precision used to determine the Jacobian determinant's quasi-zero status. The precision value, typically set at 0.01, defines this critical area and serves as a reference for each robot mechanism type. If the calculated Jacobian determinant surpasses this threshold precision, thecorresponding point in the workspace is deemed singular. The Monte Carlo analysis results in a map of singularities, showcasing their distribution in the workspace.

For instance, in Figure 7, the workspace of a planar mechanism with rotational and prismatic joints is displayed. Here, precision is maintained at 0.01, and the

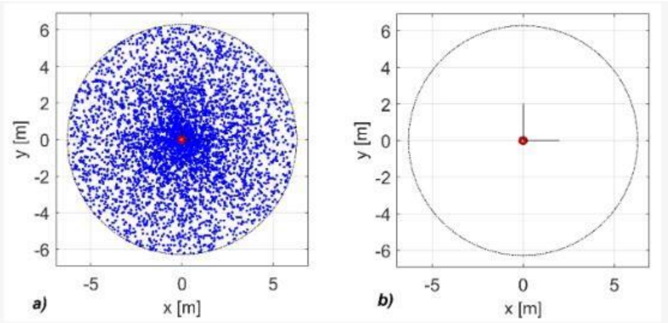


Fig 7 Workspace of Planar Mechanism (MatLab)

size of the tolerance field approximately equals one-twentieth of the workspace span. This precision standard is upheld across various mechanisms

Fig 8 illustrates the workspace of a planar mechanism with two rotational joints, depicting both singular and non- singular points, with the Monte Carlo analysis revealing a map of these singularities. The generation of numerous samples allows for assessing the ratio and distribution of singular versus non-singular states, providing valuable insights into the behavior of these robotic systems.

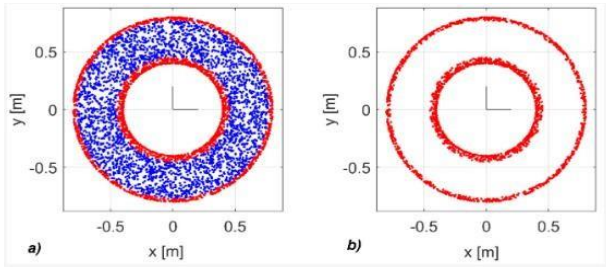


Fig 8 Workspace of 2R

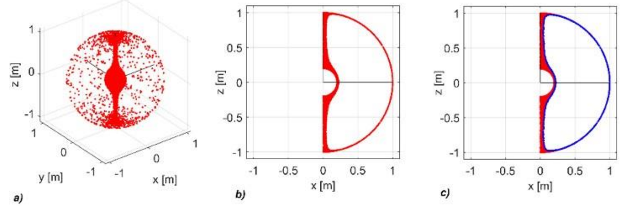


Fig 9 Singular Points

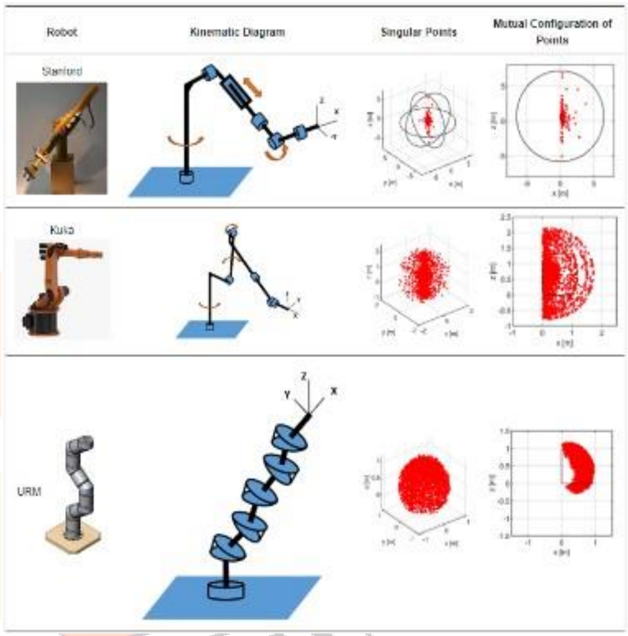
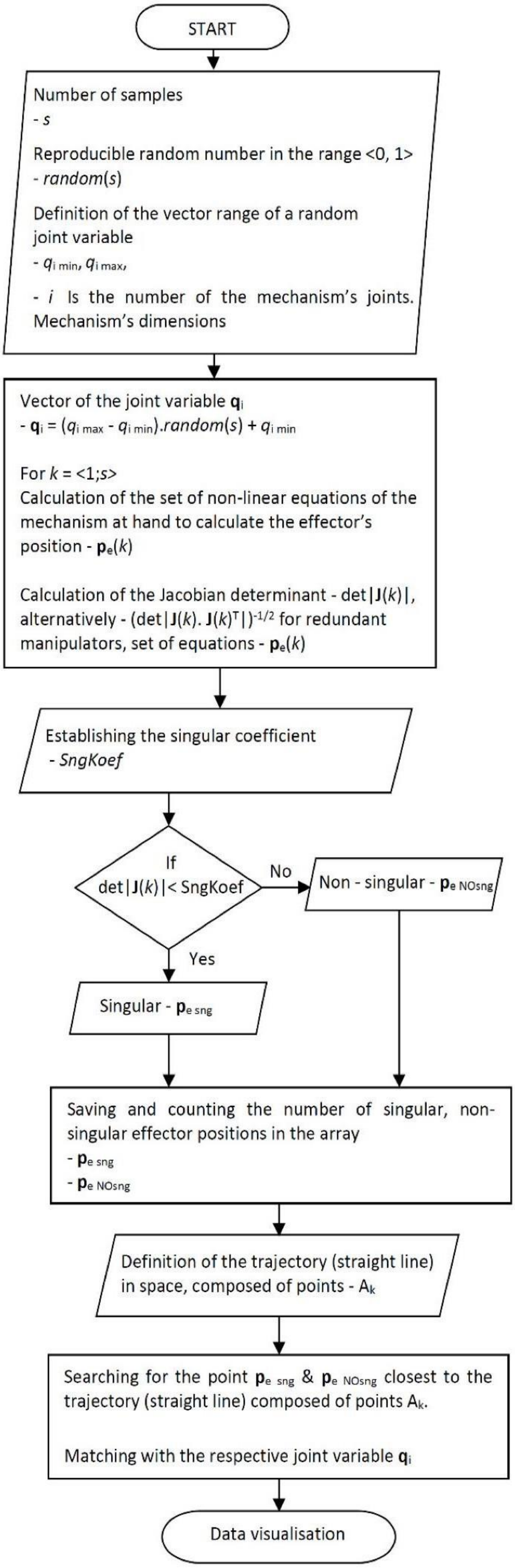


Fig 10 Comparison of Stanford , Kuka, and URM

Moreover, Figure 9 showcases the visualization of singular points, their distribution in the workspace's diagonal section, and the configurations of robotic arms that can be achieved.Notably, the analysis reveals that non-singular configurations can be achieved in close proximity to singular points for all robot types, raising questions about the criticality of the established precision in Jacobian determinant calculations for each mechanism type. The study compares different robot designs, including Stanford, Kuka, and URM (Unlimited Rotational Module), shedding light on their singular point characteristics (Fig10)

Fig 11 Algorithm to determine Singular Points



Algorithm that Determines Singular Points Generation:

Variables:
s—Number of samples (indicates the number of samplesgenerated
random(*s*)—Reproducible random number (the Gauss distribution)
$q_{i\ min}$—Minimal value of a random joint variable
$q_{i\ max}$—Maximal value of a random joint variable
i—The number of the mechanism's joints (degree offreedom)
$\mathbf{q}_i$—The vector of the joint variable is the function of thesample—s, $\mathbf{q}_i(s)$
k—Step of cycle
$\mathbf{p}_e(k)$—End effector's position
$\mathbf{J}(k)$—Jacobian matrix
det|$\mathbf{J}(k)$|—Jacobian determinant in its absolute value
SngKoef—Singular coefficient (any value in the range <0; 0.1>, in this case 0.01
singular state is assessed based on this value)
$\mathbf{p}_{e\ NOsng}$—End effector's position under non—singularity $\mathbf{p}_{e\ sng}$—End effector's position under singularity A_k—Points of the trajectory (straight line in this case) in space.

INFERENCE

Algorithm 1 and Algorithm 2, both armed with the Monte Carlo method, offer different approaches to solve the puzzle of positioning a robot's end effector while keeping joint angles in check. Think of Algorithm 1 as your robot's GPS, initially exploring both possible paths for $\theta 1$ angles and filtering out any confusing choices. It then calculates $\theta 5$, $\theta 6$, $\theta 3$, $\theta 2$, and $\theta 4$, making sure they're within acceptable boundaries. The algorithm picks the final set of joint angles by minimizing the difference from your robot's current joint positions, like your GPS picking the quickest route.

On the other hand, Algorithm 2 operates like a seasoned tour guide, following a structured roadmap through a series of steps. It uses a Finite State Machine (FSM) to smartly calculate and validate joint angles while also avoiding tricky situations (singular configurations) and ensuring smooth moves. The FSM guides the algorithm, step by step, in finding the best path. So, in a nutshell, Algorithm 1 is your GPS for simplicity and efficiency, while Algorithm 2 is your expert tour guide for systematic planning and avoiding tricky spots on the journey.

CONCLUSION

This research delves into the realm of redundant manipulators, with a specific focus on the Universal Robots UR5e, a versatile collaborative robot. The study investigates the critical issue of singular configurations that impede the performance of the UR5e and similar robots, delving into their fundamental causes, such as kinematic redundancy and geometric constraints. To mitigate these challenges, two algorithms are introduced to optimize joint angle solutions while minimizing movement. The Monte Carlo method is also employed to assess the likelihood of singular point occurrence. This research not only advances the robotics field but also has practical implications, enhancing operational efficiency, safety, and adaptability of collaborative robots, contributing to progress in automation and human-robot collaboration.

REFERENCES

1. Stejskal, T.; Svetlík, J.; Ondočko, Š. Mapping Robot Singularities through the Monte Carlo Method. Appl. Sci. 2022, 12, 8330. https://doi.org/10.3390/ app12168330

2. M. Weyrer, M. Brandstötter, and M. Husty, "Singularity Avoidance Control of a Non- Holonomic Mobile Manipulator for Intuitive HandGuidance," *Robotics*, vol. 8, no. 1, p. 14, Feb. 2019, doi: 10.3390/robotics8010014.

3. Villalobos, J.; Sanchez, I.Y.; Martell, F. Singularity Analysis and Complete Methods to Compute the Inverse Kinematics for a 6-DOF UR/TM-Type Robot. Robotics 2022, 11, 137. https://doi.org/10.3390/ robotics11060137

4. Yahya, Samer & Moghavvemi, Mahmoud & Mohamed, HAF. (2012). Singularity avoidance of a six degree of freedom three dimensional redundant planar manipulator.. Computers & Mathematics with Applications. 64. 856-868. 10.1016/j.camwa.2011.12.073.

5. Takla, Abdullah & Layous, Jamil. (2023). Joints limits and singularity avoidance. 10.13140/RG.2.2.36062.20809.

6. Weyrer, Matthias & Brandstötter, Mathias & Husty, Manfred. (2019). Singularity Avoidance Control of a Non-Holonomic Mobile Manipulator for Intuitive Hand Guidance. Robotics. 8. 14. 10.3390/robotics8010014.