

Article

Not peer-reviewed version

---

# Multimodal Control of Manipulators: Coupling Kinematics and Vision for Self- Driving Laboratory Operations

---

[Shifa Sulaiman](#)\*, [Amarnath A H](#), [Simon Bøgh](#), [Naresh Marturi](#)

Posted Date: 4 December 2025

doi: 10.20944/preprints202512.0336.v1

Keywords: self-driving laboratory; vision-based object tracking; Jacobian inverse kinematics; trajectory planning (RRT\*); autonomous robotic manipulation



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# Multimodal Control of Manipulators: Coupling Kinematics and Vision for Self-Driving Laboratory Operations

Shifa Sulaiman <sup>1,\*</sup> , Amarnath A H <sup>2</sup>, Simon Bøgh <sup>1</sup> , and Naresh Marturi <sup>3</sup> 

<sup>1</sup> Department of Electronics Systems, Aalborg University, Denmark

<sup>2</sup> Nawe Robotics, Calicut, India

<sup>3</sup> Extreme Robotics Laboratory, School of Metallurgy & Materials, University of Birmingham, Birmingham, United Kingdom

\* Correspondence: ssajmech@gmail.com

## Abstract

Self-driving laboratories are redefining autonomous experimentation by integrating robotic manipulation, computer vision, and intelligent planning to accelerate scientific discovery. This work presents a vision-guided motion planning framework for robotic manipulators operating in dynamic laboratory environments, with a focus on evaluating motion smoothness and control stability. The framework enables autonomous detection, tracking, and interaction with textured objects through a hybrid scheme that couples advanced motion planning algorithms with real-time visual feedback. Kinematic modeling of the manipulator is carried out using the screw theory formulations, which provides a rigorous foundation for deriving forward kinematics and the space Jacobian. These formulations are further employed to compute inverse kinematic solutions via the Damped Least Squares (DLS) method, ensuring stable and continuous joint trajectories even in the presence of redundancy and singularities. Motion trajectories toward target objects are generated using the RRT\* algorithm, offering optimal path planning under dynamic constraints. Object pose estimation is achieved through a vision pipeline that integrates feature-based detection with homography-driven depth analysis, enabling adaptive tracking and dynamic grasping of textured objects. The manipulator's performance is quantitatively evaluated using smoothness metrics, RMSE pose errors, and joint motion profiles including velocity continuity, acceleration, jerk, and snap. Simulation studies demonstrate the robustness and adaptability of the proposed framework in autonomous experimentation workflows, highlighting its potential to enhance precision, scalability, and efficiency in next-generation self-driving laboratories.

**Keywords:** self-driving laboratory; vision-based object tracking; Jacobian inverse kinematics; trajectory planning (RRT\*); autonomous robotic manipulation

## 1. Introduction

Self-driving laboratories (SDL) are transforming the landscape of autonomous experimentation by integrating robotics, computer vision, and intelligent planning to accelerate scientific workflows. Within these environments, dexterous manipulators equipped with sophisticated grippers play a pivotal role in executing precise and adaptive tasks, often outperforming human operators in terms of reliability and efficiency [1,2]. A key challenge in such dynamic settings lies in enabling robotic systems to interact with textured objects in real time, requiring seamless coordination between vision-based perception and motion planning. This coordination must not only ensure accurate object tracking but also generate smooth and feasible joint trajectories for safe and efficient manipulation.

This work presents a foundational control framework designed to address this challenge by coupling real-time object tracking with smooth and stable motion execution. The system integrates a hybrid vision pipeline based on feature detection and homography-driven pose estimation with Jacobian-based motion planning for a 7-DOF manipulator. While the vision module is validated

using a highly-textured, opaque object to ensure reliable tracking, the primary contribution lies in the rigorous quantitative analysis of joint motion dynamics. Specifically, we evaluate velocity continuity, acceleration, jerk, snap, and smoothness cost functions to benchmark trajectory quality and mechanical stability. Rather than claiming architectural novelty, this study focuses on validating the control pipeline's ability to generate smooth, feasible trajectories in response to dynamic visual input. Depth information is leveraged to interpret object orientation in 3D space, guiding the manipulator's end-effector toward dynamically changing targets. Trajectories are generated using the RRT\* algorithm, while inverse kinematic solutions are computed using Damped Least Squares (DLS), chosen for its potential to optimize joint smoothness and precision [3]. By establishing a modular and reproducible baseline, this framework lays the groundwork for future extensions involving more complex vision modalities and real-world SDL objects such as transparent glassware, reflective surfaces, and low-texture microplates.

The remainder of this paper is organized as follows. Section 2 focuses on previous work related to kinematic analysis, motion planning schemes, and vision algorithms. Kinematic modeling and workspace analyses of the manipulator, motion planning schemes and vision algorithm are presented in Section 3. Experimental results, including simulation and comparison studies of the proposed motion schemes, are presented in Section 4.

## 2. Related Work

In the context of SDLs, mobile manipulators that combine a dexterous arm with a mobile base are proving essential for automating complex experimental workflows. These integrated platforms offer both spatial mobility and fine-grained manipulation capabilities, allowing them to navigate dynamic lab environments and interact with diverse instruments and materials. Their deployment in chemical research settings introduces distinct challenges and opportunities, particularly due to the delicate and potentially hazardous nature of lab operations. Tasks such as handling fragile glassware, precisely dispensing reagents, and interfacing with analytical equipment require high levels of accuracy, repeatability, and safety making mobile manipulators a cornerstone of autonomous scientific discovery. A motion planning scheme associated with a manipulator typically involves several phases, including kinematic modeling, workspace analysis, trajectory planning, and collision avoidance methods. For a Jacobian-based motion planning scheme, a manipulator's kinematic model is essential to determine joint solutions. Kinematic equations for a manipulator can be derived using various techniques, such as the geometric approach, the DH method, the screw theory approach, and iterative methods. Kinematic modeling of a manipulator establishes the relationship between the pose of an end effector and its corresponding joint configurations. It also defines the mapping between the linear and angular velocities of the end effector and the manipulator's joints.

Screw theory formulations are widely used for the kinematic modeling of robotic systems with higher number of degrees-of-freedom (dof). This approach proved to be particularly flexible for modeling complex systems with coupled and offset joints. Liao et al. [4] demonstrated the application of screw theory formulations in determining inverse kinematic solutions for a 6-DOF manipulator with offset joints. They utilized the Paden-Kahen (PK) method in conjunction with screw theory to establish the relationship between joint angular variations and end effector positions. Another approach to kinematic modeling, combining screw theory and PK methods, was presented by Gandhi et al. [5]. They applied this methodology to model a 3-DOF street cleaning robotic manipulator, obtaining inverse solutions using the PK method. A comparison of joint solutions obtained through the PK method and numerical inverse kinematic approaches was conducted to assess computational efficiency. Additionally, Wang et al. [6] illustrated a derivation and solution strategy for solving kinematic equations of a free-floating robot arm. Their method involved a combined approach of screw theory and conservation of linear momentum theory. They employed Newton's iterative method during inverse kinematic analysis and compared results with other iterative methods to validate the accuracy of the approach.

Screw theory formulations implemented for determining inverse kinematic solutions for a 6 dof manipulator with offset joints was demonstrated by Liao et al. [4]. Paden-Kahen (PK) method was used along with screw theory approach to map a relation between joint angular variations and end effector positions. Another kinematic modeling approach combining screw theory and PK methods were given in [5]. A 3 dof street cleaning robotic manipulator were modeled using screw theory approach and inverse solutions were obtained using PK method. Joint solutions determined using PK method were compared with joint solutions obtained using a numerical inverse kinematic approach to study a computational efficiency of the proposed approach. Wang et al. [6] illustrated a derivation and a solution strategy for solving kinematic equations of a free floating robot arm using a combined approach of screw theory method and conservation of linear momentum theory. Newton's iterative method was also used during inverse kinematic analysis and results were compared with iterative methods to prove an accuracy of the method.

Liu et al. [7] introduced a kinematic modeling approach for a 6-DOF industrial robot utilizing screw theory formulations. They employed a Particle Swarm Optimization (PSO)-based algorithm to minimize synthesis errors while considering kinematic and dynamic constraints. Another method for kinematic modeling of a redundant manipulator, combining screw theory with the Newton-Raphson method, was presented by Ge et al. [8]. They derived forward kinematic equations based on screw theory formulations and obtained joint solutions using the Newton-Raphson method. Screw theory-based kinematic modeling and motion analysis of a fixed-base dual-arm robot were demonstrated by Sulaiman et al. [9]. They utilized screw theory-derived kinematic equations to plot the robot's workspace. Additionally, Sulaiman et al. [10] derived kinematic equations for a 10-DOF dual-arm robot with a wheelbase using screw theory. These equations were employed to evaluate singularities and dexterous regions within the robot's workspace. An iterative method for determining forward kinematic equations using screw theory formulations was demonstrated by Medrano et al. [11]. They applied this approach to model a 6-DOF manipulator and conducted simulation studies to demonstrate the advantages of their method. Sun et al. [12] illustrated a kinematic modeling method based on screw theory applicable for hybrid mechanisms. They computed Jacobian and Hessian matrices using a recursive method and used the proposed approach to derive the kinematic model of a hybrid robot. Simulation results indicated the accuracy of the obtained kinematic equations, demonstrating the practical feasibility of the proposed method for kinematic analysis of hybrid mechanisms.

Recent advancements in vision-based grasping have significantly enhanced the capabilities of robotic manipulators in dynamic and unstructured environments. H el enon et al. [13] introduced a plug-and-play vision-based grasping framework that leverages Quality-Diversity (QD) algorithms to generate diverse sets of open-loop grasping trajectories. Their system integrates multiple vision modules for 6-DoF object detection and tracking, allowing trajectory generalization across different manipulators such as the Franka Research 3 and UR5 arms. This modular approach improves adaptability and reproducibility in robotic grasping tasks. Kushwaha et al. [14] proposed a vision-based intelligent grasping system using sparse neural networks to reduce computational overhead while maintaining high grasp accuracy. Their Sparse-GRConvNet and Sparse-GINNet architectures utilize the Edge-PopUp algorithm to identify high-quality grasp poses in real time. Extensive experiments on benchmark datasets and the a cobot validated the models' effectiveness in manipulating unfamiliar objects with minimal network parameters. Wang et al. [15] developed a trajectory planning method for manipulator grasping under visual occlusion using monocular vision and multi-layer neural networks. Their approach combines Gaussian sampling with Hopfield neural networks to optimize grasp paths in cluttered environments. The proposed method achieved a 99.5% identification accuracy and demonstrated significant improvements in motion smoothness and efficiency.

Zhang et al. [16] presented a comprehensive survey of robotic grasping techniques, tracing developments from classical analytical methods to modern deep learning-based approaches. Their work highlights the evolution of grasp synthesis and the integration of vision algorithms in robotic manipulation. Similarly, Newbury et al. [17] reviewed deep learning approaches to grasp synthesis,

emphasizing the role of convolutional neural networks and transformer models in improving grasp reliability. Du et al. [18] provided a detailed review of vision-based robotic grasping, covering object localization, pose estimation, and grasp inference for parallel grippers. Their analysis underscores the importance of combining RGB-D data with machine learning models to enhance grasp precision in real-world scenarios. These findings align with the growing trend of integrating vision and motion planning for dexterous manipulation. In addition to grasp synthesis, trajectory planning remains a critical component of robotic manipulation. Zhang et al. [19] proposed a time-optimal trajectory planning strategy that incorporates dynamic constraints and input shaping algorithms to improve motion speed and smoothness. Jose et al. introduced a screw theory-based method for generating singularity-free waypoints, validated through experiments on parallel manipulators. Finally, Ortenzi et al. [20] developed an iterative method for determining joint solutions of redundant manipulators performing telemanipulation tasks. Their approach avoids singularities and joint limits, enabling smooth and reliable motion execution. These contributions collectively demonstrate the importance of integrating vision-based perception with robust motion planning schemes to enhance the adaptability, precision, and safety of robotic manipulators in complex environments.

Existing research in SDLs has made notable progress in robotic manipulation, motion planning, and vision-based perception, yet several key limitations persist. Many frameworks lack real-time adaptability in dynamic lab environments, particularly when interacting with textured or moving objects. Vision systems are often limited to static object detection or rely on depth sensors, without integrating feature-based tracking and homography-driven depth estimation for textured surfaces. Furthermore, while screw theory has been widely applied for kinematic modeling, its use in deriving both forward and inverse kinematics with stability guarantees under redundancy and singularities is still underexplored. The screw theory with the Damped Least Squares (DLS) method addresses this gap by enabling smooth and continuous joint trajectories. In motion planning, although algorithms like RRT\* are known for optimality, their coupling with real-time visual feedback for adaptive trajectory generation remains limited in current literature. The proposed framework bridges this gap by combining RRT\*-based planning with a vision pipeline that supports dynamic pose estimation and grasping. Additionally, few studies offer a unified simulation-based evaluation of these components using quantitative metrics such as RMSE pose errors, velocity continuity, and higher-order motion profiles. By addressing these gaps, our work contributes a cohesive and scalable solution for autonomous experimentation in next-generation SDLs.

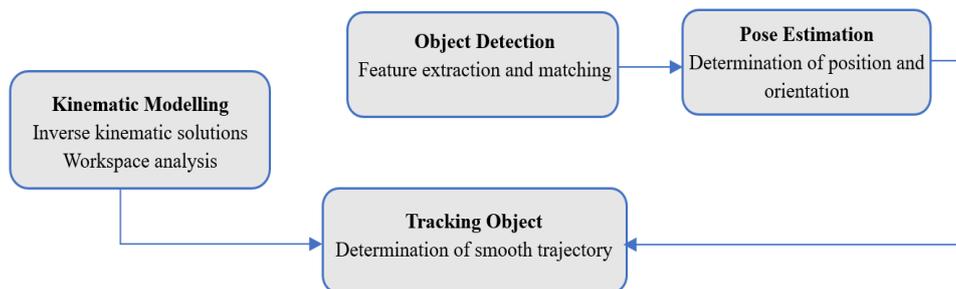
### 3. Methodologies

We performed kinematic analysis to model the motion constraints of the tracking system and assess the robot's operational reach. This included solving inverse kinematics to determine joint configurations that enable the system to accurately follow an object's path, thereby supporting effective grasping and manipulation. We implemented a structured planar pose estimation algorithm for object tracking, which unfolds through several key stages. The process begins with extracting and matching features to reliably identify distinct characteristics of the target object. This is followed by homography estimation and perspective transformation, which establish spatial correspondence and alignment. Directional vectors are then computed on the object's surface to analyze its orientation. Finally, depth data is integrated to estimate the object's planar pose, ensuring precise interaction and grasping capabilities. For object detection, the system first identifies salient features such as edges, corners, blobs, and ridges from images of planar objects using SIFT [21]. These features are then matched with those captured by the camera to determine object presence. Once converted into feature vectors, matches are evaluated to confirm detection. To address the computational demands of high-dimensional feature matching, we utilized the FLANN-based K-d Nearest Neighbor Search [22], which offers efficient performance for real-time applications. Homography estimation relies on the matched features, though some may be incorrect and introduce noise. To overcome this, we applied the RANSAC algorithm [23], which filters out false matches by iteratively selecting inliers

from minimal subsets of data. This approach improves both speed and accuracy, making it ideal for dynamic environments. Perspective transformation is then used to estimate corresponding points in the test image, allowing us to derive basis vectors on the object's surface. Depth information is subsequently incorporated to calculate the surface normal, enabling accurate 3D pose estimation of the planar object.

### 3.1. Approach Overview

This work employs a KUKA LBR iiwa 14 manipulator and Robotiq Hand-E gripper to evaluate the developed vision based motion planning scheme as shown in Fig 1. The kinematic equations governing the manipulator are derived using the screw theory approach. These equations facilitate the analysis of the manipulator's Cartesian workspace, which is essential for motion planning. Subsequently, motion planning is conducted to identify an optimal set of joint solutions for navigating trajectories within the manipulator and gripper workspaces. Trajectories are generated using the RRT\* algorithm, while joint solutions are determined using the Damped Least Square (DLS) method. Although the simulation environments used in this study are obstacle-free, the RRT\* algorithm was deliberately chosen over simpler interpolation methods to ensure generalization to more complex, cluttered environments typical of real-world SDL. RRT\* offers asymptotic optimality and the ability to handle high-dimensional configuration spaces with dynamic constraints, making it well-suited for future extensions involving obstacle avoidance, constrained motion, and workspace reconfiguration. By integrating RRT\* into the current framework, we establish a scalable and modular planning backbone that can be readily adapted to more realistic scenarios without requiring fundamental changes to the motion planning architecture.



**Figure 1.** Methodology adopted in this work.

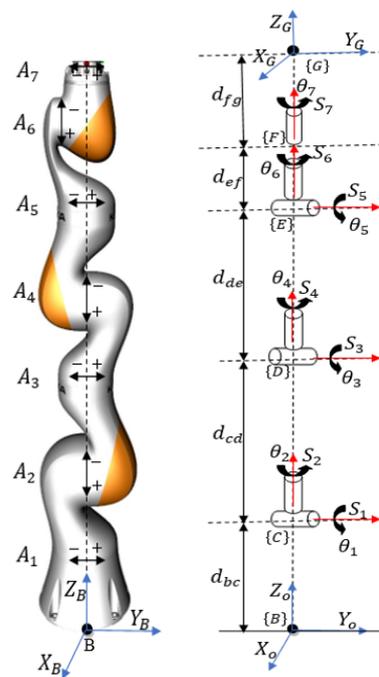
The kinematic equations obtained through the screw theory method enable the derivation of a set of feasible solutions. A unique set of configurations is selected for traversing a given trajectory using the DLS method. To ensure smooth and optimal trajectories, an objective function based on the accelerations of Cartesian motions is incorporated alongside the trajectory function. The smoothness of trajectory motions is assessed using this function. Along with the smoothness function, some other characteristics of the joint motions such as velocities, accelerations, jerk and snap values are evaluated. To further validate the performance of motion planning schemes, errors along the X, Y, and Z directions along with orientation errors of trajectory waypoints are compared. Simulation studies are conducted to evaluate the efficacy of each method in determining joint solutions for trajectories traversing various locations within the workspace. Building on this framework, object detection is achieved through feature extraction and matching, allowing the system to accurately identify and localize target objects within the environment. Once an object is detected, pose estimation determines its precise position and orientation, supplying essential spatial data for manipulation. In the final stage, adaptive tracking enables the robot to dynamically refine its motion strategy, mitigating singularities and ensuring stable interaction with the object. This sequential pipeline supports robust and flexible autonomous handling across complex and dynamic scenarios.

**Table 1.** Manufacturer provided link dimensions for the KUKA iiwa 14 manipulator.

Link	Dimension(mm)
$d_{bc}$	340
$d_{cd}$	740
$d_{de}$	400
$d_{eg}$	126
$d_{gf}$	126

### 3.2. Kinematic Modeling of the Manipulator

The kinematic modeling of the robot establishes a relationship between the angular variations of the manipulator's joints and the pose of the gripper. As stated before, the screw theory approach [24] has been utilized for this purpose. The corresponding kinematic frames of the manipulator are shown in Figure 2. In this formulation, two inertial frames are employed to determine the poses of the end effector. One frame is designated to the manipulator base, and the other to the end effector joint. In this method, the displacement of rigid bodies is represented as the motion of a screw with a defined pitch. First, we present the modeling for the robot, followed by the modeling for the hand.

**Figure 2.** Illustration of various frames and screw axes of the used KUKA iiwa manipulator.

As shown in Figure 2, all the joints of the robot are assigned with a screw axis with zero pitch for the rotary joints.  $A_i$  and  $d_{ij}$  represent various manipulator joints and links, respectively. Link lengths of the used KUKA iiwa manipulator are given in Table 1. Rodrigues formula for finding rotation matrix is given in (1)

$$\mathbf{R}(\hat{\omega}, \theta) = e^{[\hat{\omega}]\theta} = \mathbf{I} + \sin\theta[\hat{\omega}] + (1 - \cos\theta)[\hat{\omega}]^2 \quad (1)$$

where  $\hat{\omega}$  represents the  $3 \times 3$  skew-symmetric matrix of angular velocity matrix,  $\omega$ .  $\theta$  denotes the angles of respective joints. Let  $S_i$  represents the screw axes of various joints and  $4 \times 4$  transformation matrix,  $e^{[S]\theta}$  obtained using screw theory is given in (2).

$$e^{[S]\theta} = \begin{bmatrix} e^{[\hat{\omega}]\theta} & \theta \mathbf{I} + (1 - \cos\theta)[\hat{\omega}] + (\theta - \sin\theta)[\hat{\omega}]^2 \mathbf{v} \\ 0 & 1 \end{bmatrix} \quad (2)$$

where,  $\mathbf{v}$  is the 3x1 linear velocity vector and  $\mathbf{I}$  is the 3x3 identity matrix. Final transformation matrix  $\mathbf{T}(\theta)$  from base frame to end effector frame with  $n$  number of joints is given in (3).

$$\mathbf{T}(\theta) = e^{[S_1]\theta_1} e^{[S_2]\theta_2} \dots e^{[S_n]\theta_n} \mathbf{N} \quad (3)$$

where  $\mathbf{N}$  is a 4 x 4 transformation matrix, which defines the initial pose of end effector frame with respect to base frame. Transformation matrix for the used 7-dof manipulator,  $\mathbf{T}(\theta)_m$  is given in (4).

$$\mathbf{T}(\theta)_m = e^{[S_1]\theta_1} e^{[S_2]\theta_2} e^{[S_3]\theta_3} e^{[S_4]\theta_4} e^{[S_5]\theta_5} e^{[S_6]\theta_6} e^{[S_7]\theta_7} \mathbf{N}_m \quad (4)$$

where,

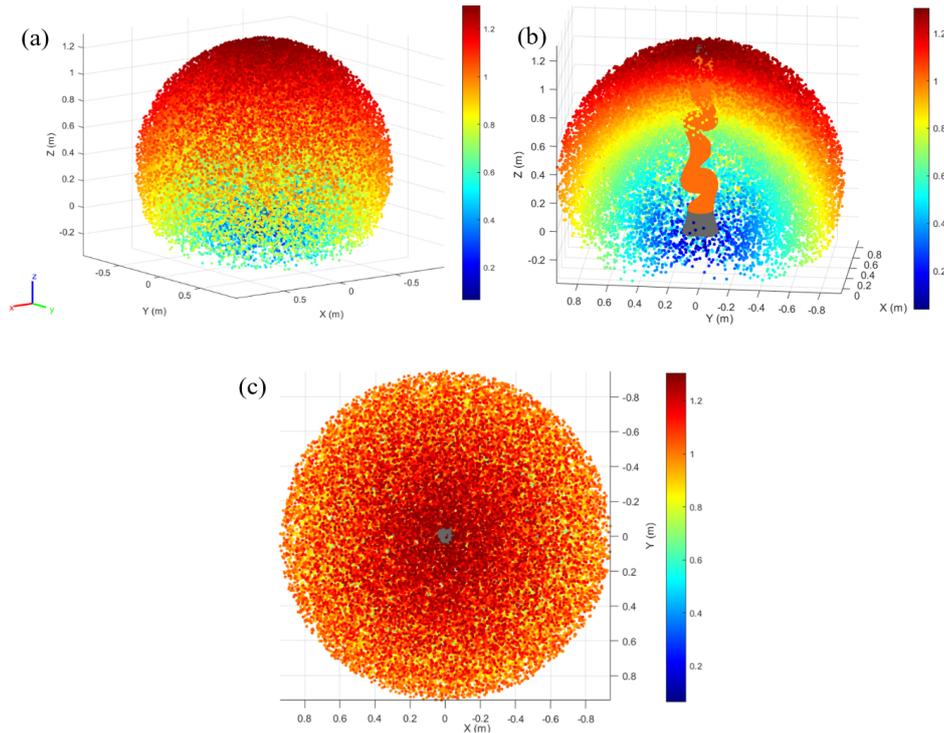
$$\mathbf{N}_m = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & D \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Distance between base frame and end effector frame,  $D$  is given in (6).

$$D = d_{bc} + d_{cd} + d_{de} + d_{ef} \quad (6)$$

### 3.3. Workspace Analysis

An analysis of the workspace was undertaken to identify singularities and unoccupied areas within the manipulator's operational space. Identifying distinct regions within the workspace facilitated the optimization of motion planning for various tasks. The combined Cartesian workspace of the manipulator and hand was determined using equation (4), considering joint limits. Figure 3 illustrates views of the workspace of the manipulator, showing all feasible positions of the manipulator and hand. This visualization accounted for joint limitations. Figures 3 (a), (b), and (c) represent 3D, sectioned, and top views respectively of the cartesian workspace. Notably, the maximum reachable point within the workspace was found to be 0.78 meters. Moreover, the manipulator's dexterous workspace volume, including the gripper, was calculated to be 1.93 cubic meters. The obtained workspace serves as a basis for identifying the reachable areas of the manipulator during trajectory planning. By leveraging this information, any unreachable spaces traversed by the trajectory were excluded to prevent issues during the motion planning scheme.



**Figure 3.** Views of combined workspace of the manipulator with hand (a)3D (b)Sectioned (c)Top.

#### 3.4. Determining Jacobian via Screw Theory

The space Jacobian matrix,  $J_s(\theta)$ , in fixed (space) frame coordinates, is a function of the joint angles  $\theta$ . Given the joint velocities  $\dot{\theta}$ , the task space (Cartesian) velocity  $V_s$  of an end-effector is given in (7)

$$V_s = J_s(\theta) \cdot \dot{\theta} \quad (7)$$

where  $J_s(\theta)$  is the  $6 \times n$  Jacobian matrix, with  $3 \times n$  linear Jacobian elements,  $J_v$ , and  $3 \times n$  angular Jacobian elements,  $J_\omega$  as given in (8).

$$J_s(\theta) = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \quad (8)$$

Expanding (7) for the 7-dof manipulator model, we obtain (9)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \frac{dh(\theta)}{d(\theta)} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \end{bmatrix} \quad (9)$$

where,  $\frac{dh(\theta)}{d(\theta)}$  is given in equation (10)

$$\frac{dh(\theta)}{d(\theta)} = \begin{bmatrix} J_{11} & J_{12} & \cdot & \cdot & \cdot & J_{17} \\ J_{21} & J_{22} & \cdot & \cdot & \cdot & J_{27} \\ \vdots & \dots & \vdots & \vdots & \vdots & \vdots \\ \cdot & \dots & \vdots & \vdots & \vdots & \vdots \\ J_{61} & \dots & \vdots & \vdots & \vdots & J_{67} \end{bmatrix} \quad (10)$$

The space Jacobian,  $J_s(\theta)$ , for the KUKA iiwa manipulator with  $n = 7$  joints, is determined using screw theory formulation given in (11).

$$J_s(\theta) = [J_s(\theta)_1 J_s(\theta)_2 J_s(\theta)_3 \dots J_s(\theta)_n] \quad (11)$$

where each element  $J_s(\theta)_i$  represents the  $6 \times 1$  Jacobian of each joint. The Jacobian of the first joint,  $J_s(\theta)_1$ , is equal to the screw axis,  $S_1$ , as given in (12).

$$J_s(\theta)_1 = S_1 \quad (12)$$

Jacobian of other joints are computed using (13).

$$J_s(\theta)_i = [Ad_{e^{[S_1] \dots e^{[S_{i-1}] \theta_{i-1}}]} S_i, \text{ for } i = 2, 3, \dots, n \quad (13)$$

### 3.5. Inverse Kinematics Techniques - Damped Least Squares (DLS) Strategy

Inverse kinematics (IK) algorithms are essential for determining joint parameters that guide a multi-link robotic system to a specified target pose. These methods compute joint angles that reposition the end-effector from its current location  $P$  to a desired target  $T$ . The positional error is defined is given in (14):

$$\vec{e} = T - P = 0 \quad (14)$$

To minimize this error, the joint configuration  $\theta$  is iteratively updated using (15):

$$\vec{e} = J_s(\theta) \Delta \theta \quad (15)$$

The Jacobian matrix  $J_s(\theta)$ , derived via screw theory as outlined in [25], is central to all three IK strategies discussed below. However, solving (15) may not always be feasible, especially when the manipulator possesses redundant degrees of freedom (DoFs). In singular configurations, the Jacobian becomes non-invertible, leading to an infinite number of solutions. To address this, alternative Jacobian formulations, as described in [26], can be employed. These alternatives improve performance by mitigating oscillations and overshoot but may occasionally produce non-feasible or suboptimal solutions in terms of energy efficiency and computational time. In singular configurations, the Jacobian is non-invertible, resulting in having infinite solutions. To avoid this issue, alternative Jacobians can be used as described in [26]. These can improve performance by reducing oscillation and overshoot. However, they may sometimes result in non-feasible or non-optimal solutions in terms of energy and computational time.

The DLS method seeks to minimize both the Cartesian error and the joint velocity norm. It is particularly effective near singularities and for underactuated systems. The objective function with a constant parameter,  $W$  is given in (16):

$$\min \|V_s - J_s \dot{\theta}\|_{W_1}^2 + \|\Delta \theta\| \quad (16)$$

The solution is obtained as given in (17):

$$\dot{\theta} = J_s^+ V_s \quad (17)$$

where the damped inverse  $J^+$  is defined as given in (18):

$$J_s^+ = (J_s^T W_1 J_s + W_2)^{-1} J_s^T W_1 \quad (18)$$

Here,  $W_1 = I$  and  $W_2 = \alpha I$ , with  $\alpha$  being a damping coefficient that balances stability and responsiveness. The weighting matrices  $W_1$  and  $W_2$  are positive definite matrices and non-singular in nature. The weighting matrix  $W_1$  can be chosen to add priority to the task vector. This is very important in redundancy resolution approaches, as might alternatively be chosen to give the cost function energy

units and so produce consistent results regardless of units or scale. The formulation given in (18) works well for a rank-deficient Jacobian and can avoid generating high joint velocities near singularity regions. It blends joint velocity damping with minimizing the inaccuracy in following the given trajectory, resulting in a trade-off between the viability of the joint velocities and the precision with which the intended end-effector trajectory is followed. The joint update is computed using (19)

$$\Delta\theta = [J_s^T (J_s J_s^T + \alpha^2 I)^{-1}]_{7 \times 6} \vec{e}_{6 \times 1} \quad (19)$$

When  $\alpha = 0$ , the method simplifies to the pseudoinverse approach. Larger values of  $\alpha$  reduce joint velocity magnitudes but may introduce trajectory deviations. A larger value of  $\alpha$  indicates a reduced joint velocity norm, although there can still be some deviation from the intended end-effector trajectory. Given that the DLS formulation must weigh the viability of the inverse kinematic solution against its precision, it is evident that the damping factor  $\alpha$  is crucial. Therefore, it is important to set this parameter's value appropriately to guarantee that joint velocities are feasible in all configurations.

### 3.6. Vision-Based Grasping Framework

To enable reliable object manipulation, we implemented a vision-guided grasping method [27] that integrates perceptual input with adaptive planning strategies. The architecture comprises two core modules:

- Planar object localization and pose inference
- Grasp synthesis based on dynamic pose updates

#### 3.6.1. Planar Object Localization and Pose Inference

Robust pose estimation is a prerequisite for effective manipulation. Our method follows a four-stage pipeline:

1. Extraction of visual features and descriptor matching
2. Homography-based transformation and perspective alignment
3. Derivation of object-centric coordinate axes
4. Pose refinement using depth-enhanced feedback

The pose estimation pipeline fuses 2D homography with RGB-D measurements to recover the full 6-DoF pose of planar, textured objects. First, the RGB image is processed with SIFT to extract keypoints and descriptors that are robust to viewpoint and scale changes. Descriptor matching against a pre-registered template is performed using FLANN, and RANSAC is subsequently applied to reject outliers and compute a homography that maps template coordinates to the current image plane. The homography is used to localize 2D corner (or keypoint) pixel coordinates in the RGB image, and then the depth image is generated at those pixel coordinates to obtain metric 3D points that are used for pose estimation. The homography is used purely for 2D localization: it projects the four template corners to pixel coordinates in the current frame, yielding high-confidence 2D points that anchor the object's footprint. Depth fusion is then performed at these localized pixels. The RGB and depth frames are hardware-aligned using the camera intrinsics and distortion parameters obtained from calibration. For each corner pixel, the corresponding depth value is obtained from the synchronized depth image. These values are back-projected into 3D camera coordinates via the pinhole model, producing a set of 3D corner points that transform the 2D homography correspondences into metric space. If any corner depth is invalid (missing or beyond sensor range), nearest-neighbor interpolation over the local depth patch or planar fitting on inliers is used to recover a consistent 3D estimate; frames with insufficient valid corners are discarded to preserve reliability. The object pose is constructed from these 3D corners. The centroid defines the translation, while orientation is obtained by fitting a local orthonormal frame to the planar surface: the surface normal is computed from cross products of non-collinear corner edges, and in-plane axes are derived by orthogonalizing one edge direction against the normal. The resulting rotation matrix is refined by enforcing orthonormality. For numerical

stability, a minimal set of geometric constraints is applied to reject degenerate configurations (nearly collinear corners or extremely acute aspect ratios), and the final pose is expressed either as Euler angles or a quaternion depending on downstream controller requirements. To mitigate measurement noise and ensure smooth control inputs, the estimated pose is temporally filtered before publication.

All poses are transformed from the camera frame to the robot base frame using the fixed extrinsic calibration between the sensor and the manipulator, ensuring consistency with the kinematic chain. The final output is published as a message comprising the filtered 6-DoF pose and timestamp, which the motion stack consists of two phases: RRT\* for the initial approach to the vicinity of the object, followed by DLS-based visual servoing that resolves incremental pose errors in real time without global re-planning. This design makes the roles of homography and depth explicit and complementary: homography provides robust 2D corner localization under texture and perspective changes, while depth supplies metric scale to recover 3D structure. Their fusion yields stable, accurate 6-DoF poses suitable for dynamic manipulation, with clearly defined fallbacks, filtering, and frame transforms to maintain smooth end-effector trajectories. Planar objects are detected using the SIFT algorithm, which identifies distinctive keypoints and encodes local image structure through descriptors. Matching is performed using FLANN for floating-point descriptors or Hamming distance for binary descriptors, establishing correspondences between the input image and a known template. From the matched keypoints, a homography matrix  $\mathbf{H}$  is computed to model the planar transformation as given in (20)

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \mathbf{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (20)$$

The projected coordinates  $(x', y')$  are obtained using (21)

$$x' = \frac{a}{c}, \quad y' = \frac{b}{c} \quad (21)$$

RANSAC is employed to eliminate outliers and ensure robust estimation. To define the object's local frame, three reference points are selected based on (22)

$$P_c = (w/2, h/2), \quad P_x = (w, h/2), \quad P_y = (w/2, 0) \quad (22)$$

where  $w$  and  $h$  denote the object's width and height. These points are projected into 3D using RGB-D data from a RealSense camera, yielding directional vectors using (23)

$$\vec{i} = \frac{\vec{x}}{\|\vec{x}\|}, \quad \vec{j} = \frac{\vec{y}}{\|\vec{y}\|}, \quad \vec{k} = \frac{\vec{x} \times \vec{y}}{\|\vec{x} \times \vec{y}\|} \quad (23)$$

The orthonormal basis  $(\vec{i}, \vec{j}, \vec{k})$  defines the object's orientation. The rotation matrix  $R$  is constructed from these vectors given in (24)

$$R = \begin{bmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \\ i_z & j_z & k_z \end{bmatrix} \quad (24)$$

Euler angles  $(\phi, \theta, \psi)$  are derived as given in (25)

$$\theta = \tan^{-1}(j_z), \quad \phi = \sin^{-1}(-i_z), \quad \psi = \tan^{-1}\left(\frac{i_y}{i_x}\right) \quad (25)$$

These angles encode the object's spatial orientation and used for planning the grasping strategy.

### 3.7. Motion Evaluation Metrics

In this section, we describe the motion planning evaluation metrics adopted for analysing the motion smoothness. The evaluation metrics encompassed both motion smoothness and trajectory tracking performance. Motion smoothness was assessed using velocity, acceleration, jerk, and snap profiles, while tracking accuracy was quantified through Root Mean Square Error (RMSE) of the end-effector trajectory. To analyze dynamic consistency, velocity continuity and higher-order motion profiles were computed by evaluating the maximum differences between successive joint values across the trajectory.

After finalizing the trajectory waypoints, the joint angles of the manipulator and gripper were provided to achieve the desired end-effector motions. Smoothness of joint motions was evaluated based on the rate of change of acceleration over time. The smoothness function  $S_{func}$ , derived from the third derivative of position, serves as an effective metric for smoothness. Since smoothness is inversely proportional to the rate of change of acceleration, the reciprocal of  $S_{func}$  was used as a measure of smoothness. The performance of above-mentioned methods with respect to x, y and z coordinates within a time period,  $t$  was evaluated using a reciprocal of smoothness function [28] given in (26).

$$S_{func} = \frac{1}{\sqrt{\frac{1}{2} \int_{t_1}^{t_2} \left( \left( \frac{d^3x}{dt^3} \right)^2 + \left( \frac{d^3y}{dt^3} \right)^2 + \left( \frac{d^3z}{dt^3} \right)^2 dt} \times \frac{(t_2 - t_1)^5}{l^2}} \quad (26)$$

where, length of trajectory,  $l$  is given in (27)

$$l = \sum_{i=1}^{n-1} \sqrt{\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2} \quad (27)$$

As the rate of change of acceleration increases, smoothness decreases. Therefore, smoother motions correspond to lower rates of change in acceleration. Additionally, positional errors at the waypoints of the end-effector were analyzed to assess the accuracy of the inverse joint solutions. To provide a comprehensive measure of overall error, this work introduces an RMSE metric that combines both positional and orientation errors. To effectively represent these combined errors as a single measure, positional errors (x, y, z) and orientation errors were calculated separately, normalized, and then integrated into a unified metric. For position, the RMSE can be computed using (28):

$$RMSE_{pos} = \sqrt{\left( \frac{1}{N} \right) * \sum_{i=1}^N \left( (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2 \right)} \quad (28)$$

For orientation errors, since there are three angles (roll, pitch and yaw), the RMSE for each angle is computed separately as given in (29) - (31):

$$RMSE_{roll} = \sqrt{\left( \frac{1}{N} \right) * \sum_{i=1}^N \left( (\theta_{roll,i} - \hat{\theta}_{roll,i})^2 \right)} \quad (29)$$

$$RMSE_{pitch} = \sqrt{\left( \frac{1}{N} \right) * \sum_{i=1}^N \left( (\theta_{pitch,i} - \hat{\theta}_{pitch,i})^2 \right)} \quad (30)$$

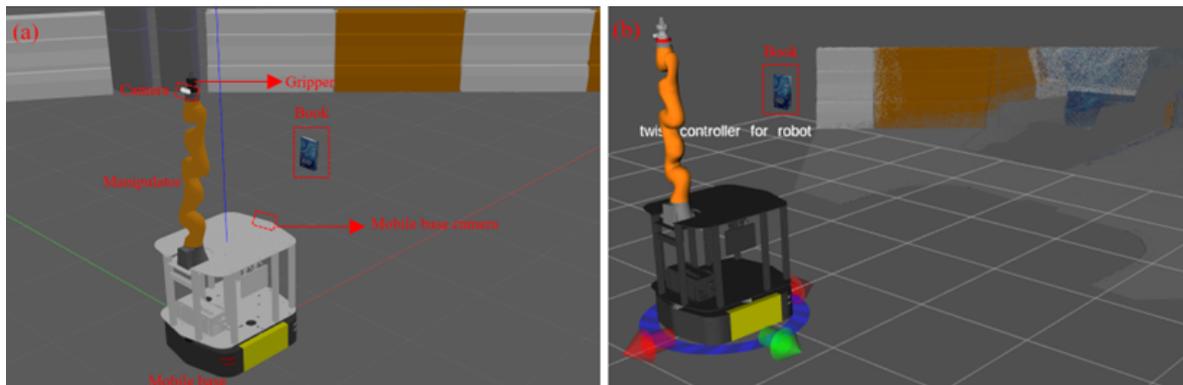
$$RMSE_{yaw} = \sqrt{\left( \frac{1}{N} \right) * \sum_{i=1}^N \left( (\theta_{yaw,i} - \hat{\theta}_{yaw,i})^2 \right)} \quad (31)$$

To represent the overall orientation error as a single metric in radians, we combined the individual RMSE values for roll, pitch, and yaw into a single orientation RMSE using (32):

$$RMSE_{orient} = \sqrt{\left(\frac{1}{3}\right) * (RMSE_{roll})^2 + (RMSE_{pitch})^2 + (RMSE_{yaw})^2} \quad (32)$$

#### 4. Results and Discussion

The process of robotic manipulation in this work begins with kinematic modeling, which involved solving inverse kinematics and conducting workspace analysis to determine feasible configurations for the robot's end effector. This foundational stage ensured that the manipulator can reach and interact with objects within its operational domain. Following this, object detection is performed through feature extraction and matching techniques, enabling the system to identify and localize target objects within the environment. Once detected, pose estimation is employed to determine the precise position and orientation of the object, providing critical spatial information for manipulation. The final stage involved tracking the textured object, in which the robot dynamically adjusts its position and orientation to follow the object. This sequential framework facilitates reliable and flexible autonomous handling in complex and variable settings. The proposed object recognition and pose estimation algorithm were implemented within the Robot Operating System (ROS) framework, leveraging OpenCV on an Ubuntu 20.04 platform. The system was deployed on a 3.0 GHz Intel Core i7-7400 CPU with 16GB of RAM. To evaluate the algorithm, a simulation environment was constructed using Rviz and Gazebo, featuring a mobile manipulator interacting with a textured book cover, as illustrated in Figure 4. The simulation setup, including both Gazebo and Rviz visualizations, is depicted in Figures 4(a) and 4(b), respectively.



**Figure 4.** Simulation environment (a) Gazebo (b) Rviz.

The motion strategies were evaluated based on the smoothness of joint motions using a smoothness function and the error range relative to the desired end-effector trajectory. Additionally, the velocity, acceleration, jerk, and snap values of the motions were determined to analyse the joint motions. The proposed method aimed to gradually adjust joint positions and orientations from a stable state. However, due to the arbitrary number of iterations, the time required to find an inverse kinematics (IK) solution for a given end-effector pose was variable. Despite this, the duration of a single iteration remained constant with respect to the dimensionality of  $J$  and  $\theta$ , and was unaffected by the full algorithm's completion. To address this variability, a maximum time limit for the algorithm was enforced by setting an upper bound on the number of iterations. In Jacobian-based inverse kinematics solvers, the dimensionality of  $J$  in 3-dimensional space is typically either 3 or 6. A 3-dimensional  $J$  encodes only the positional information for the end-effector, while a 6-dimensional  $J$  is often preferred as it includes both positional and orientation information. In this work, a 6-dimensional  $J$  vector was chosen to account for both positional and orientation components. The system was deemed repeatable if a given goal vector  $\vec{g}$  consistently produced the same pose vector. However, achieving repeatability in redundant systems requires special measures, as this consistency is not guaranteed inherently. An

alternative approach involves resetting the system to a predefined default pose, ensuring repeatable solutions. However, this method may introduce sharp discontinuities in the solution trajectory. For every inverse solution technique employed in this work, the error matrix  $\vec{e}$  was assigned values as described in (33):

$$\vec{e} = \begin{bmatrix} X_{\text{target}} - X_{\text{endeffector}} \\ Y_{\text{target}} - Y_{\text{endeffector}} \\ Z_{\text{target}} - Z_{\text{endeffector}} \\ \alpha_{\text{target}} - \alpha_{\text{endeffector}} \\ \beta_{\text{target}} - \beta_{\text{endeffector}} \\ \gamma_{\text{target}} - \gamma_{\text{endeffector}} \end{bmatrix} = \begin{bmatrix} 0.01 \\ 0.01 \\ 0.01 \\ 0.01 \\ 0.01 \\ 0.01 \end{bmatrix} \quad (33)$$

$\Delta\vec{e}$  was determined such that it moves  $\vec{e}$  closer to  $\vec{g}$ . The starting iteration assumes  $\Delta\vec{e}$  as  $\vec{g} - \vec{e}$ . The stopping conditions were implemented to improve the performance of the method and reduce computational effort during the iterations. In this work, the stopping criteria were as follows:

- Finding a solution within the error limits given in (34).
- Convergence to a local minimum.
- Non-convergence after the allotted time.
- Maximum iterations reached.

If the solution converges to a local minimum, pose vectors can be randomized to avoid recurrence in future iterations. An allotted time can be specified for each step to prevent the method from exceeding a predetermined duration. Additionally, the maximum number of steps can be set to limit computational time. The iteration step size,  $\beta$ , was determined using the desired  $\vec{e}$  and  $\vec{g}$  as given in (34):

$$\Delta\vec{e} = \beta(\vec{g} - \vec{e}), \quad 0 \leq \beta \leq 1 \quad (34)$$

The step size was limited by scaling it with  $\beta$ . The determination of  $\beta$  was carried out after computing  $\Delta\theta$  for better approximation. A common approach involved limiting joint rotation increments to a maximum of 5 degrees per iteration. Initially,  $\Delta\theta$  was computed without including  $\beta$ , and later checked to ensure no  $\Delta\theta$  values exceeded the threshold  $\beta_h$ , calculated using equation (35):

$$\beta_h = \frac{\text{Threshold}}{\max(\text{Threshold}, \max(|\Delta\theta_i|))} \quad (35)$$

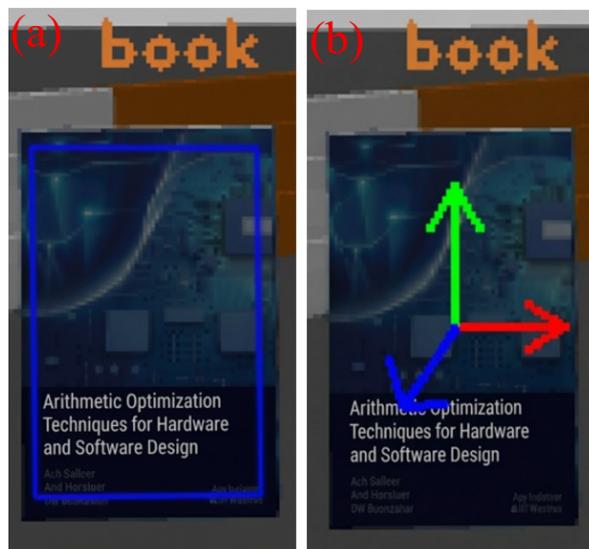
After finalizing  $\beta$ , the  $\Delta\theta$  values were updated during iterations using (36):

$$\theta = \theta + \beta(\Delta\theta) \quad (36)$$

The dampening constant of DLS method was determined by multiplying the size of the desired change in the end effector by a small constant that was set at each iteration. The small constant was taken as 1/100 of the total sum of the segments from base to end effector. Small constant was added to prevent the oscillating of errors due to desirable change was too small to dampen. The problem occurs because the target positions are nearly close to each other. The inclusion of  $\alpha$  avoids the convergence of jacobian matrix to a singular space at any time. An orthogonal set of vectors, which are equal to the length of the desirable change in vector,  $\vec{e}$  will result in a good damping effect. However, the effect resulted in increase in computational time for calculating the joint values. Each iteration tried to lower the value of errors to the desired  $\vec{e}$  value. The iteration stopped when the stopping criteria specified in section were met.

A RealSense camera mounted on the mobile base (shown in Figure 4) was employed to detect the initial pose of a textured book cover. Once the pose was identified, the RRT\* motion planning algorithm utilized this information to generate a trajectory that allowed the manipulator to follow the book's movement while maintaining a grasp-ready posture. DLS method was used to determine joint angles to traverse the trajectory. The outputs of the vision algorithm are presented in Figures 5(a) and

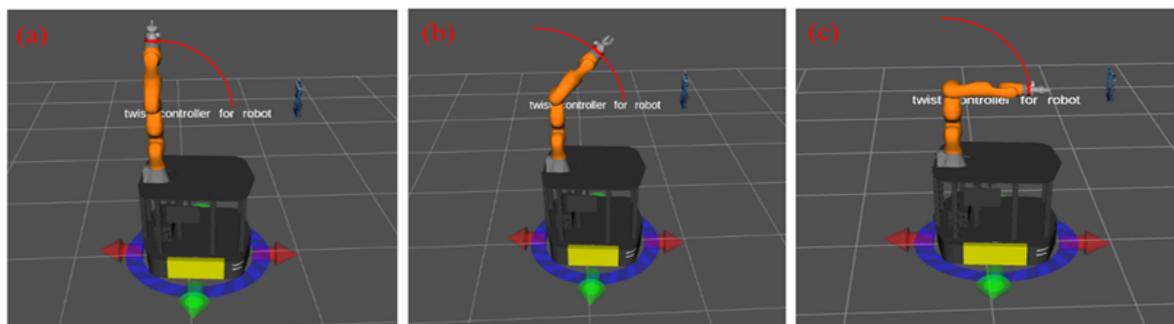
(b), where Figure 5(a) shows the detected bounding box around the object, and Figure 5(b) illustrates the estimated pose. These visual results confirmed the algorithm's effectiveness in accurately localizing and orienting the target, providing essential input for subsequent manipulation.



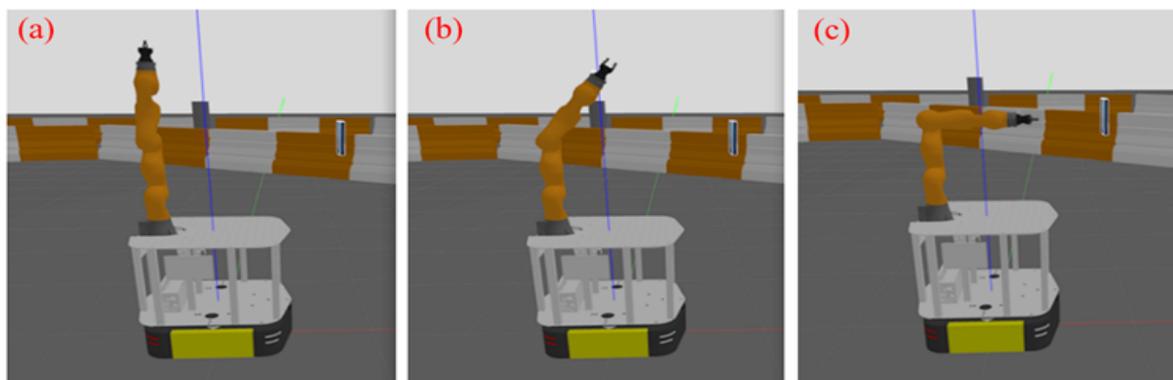
**Figure 5.** Vision algorithm outputs: (a) Bounding box, (b) Pose of object

The mobile manipulator was placed in front of a book as shown in Figure 4. The manipulator follows a trajectory computed via the Rapidly-exploring Random Tree Star (RRT\*) algorithm, with corresponding joint configurations derived using the Damped Least Squares (DLS) inverse kinematics method. The end-effector pose is defined such that the manipulator's gripper approaches the book frontally, maintaining a face-to-face orientation at a distance of 15 cm in x-axis as shown in Figures 6 and 7. Figures 6 and 7 illustrate the manipulator's progression through its initial, intermediate, and final configurations as it approaches the target pose, visualized in Rviz and Gazebo environments, respectively.

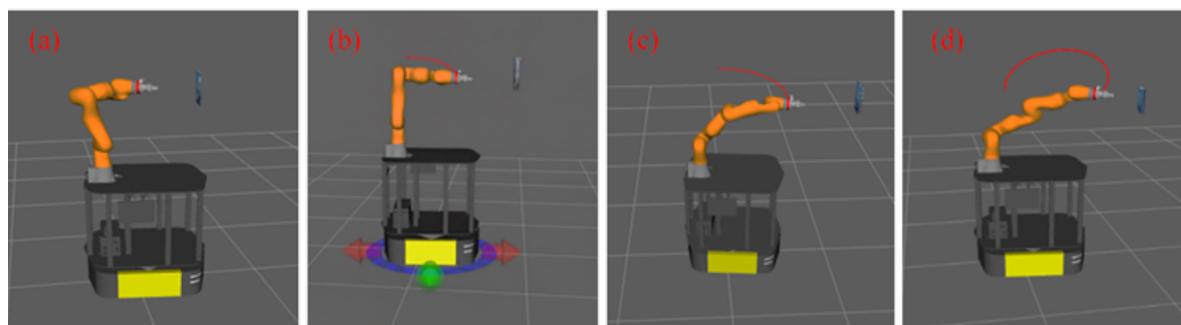
To further validate the system's manipulation capabilities, we performed tracking experiments following the motions of the book. During tracking, camera placed on the manipulator was used for detecting the updated poses of the book. Screenshots of the manipulator's motion in response to changes in the book's pose in Rviz and Gazebo are shown in Figures 8 – 9 respectively. These frames capture the adaptive motion planning and execution as the robotic arm reconfigures its joints and end effector to reach the target object. Mounted on a stationary mobile platform, the manipulator dynamically adjusts its configuration based on updated pose estimates from the vision system. The colored coordinate axes in each frame represent the estimated pose of the book, highlighting the system's ability to track and align with the object throughout the approach. This visual evidence underscores the effectiveness of the integrated perception and planning framework in achieving precise and responsive object interaction.



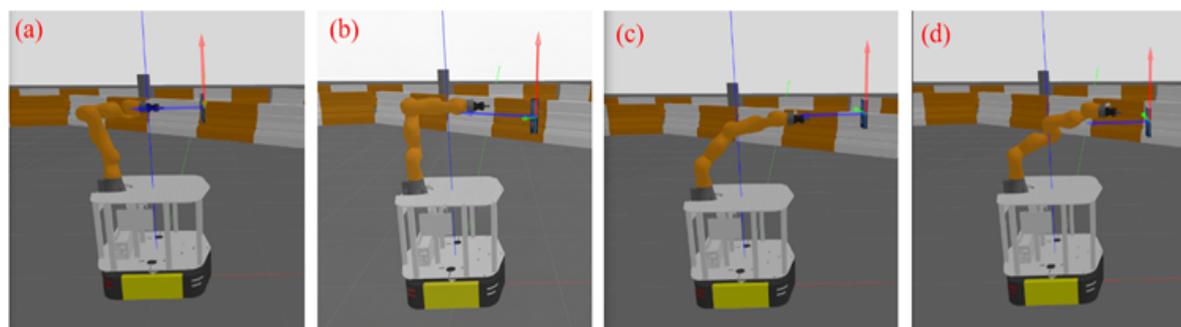
**Figure 6.** Manipulator motions to reach in front of a book in Rviz (a)Initial (b)Intermediate (c)Final



**Figure 7.** Manipulator motions to reach in front of a book in Gazebo (a)Initial (b)Intermediate (c)Final



**Figure 8.** Manipulator motions to track a book in Rviz



**Figure 9.** Manipulator motions to track a book in Gazebo

The RRT\* algorithm is employed exclusively during the initial approach phase to generate a collision-free, optimal trajectory from the manipulator's current configuration to the vicinity of the target object. Once the end-effector reaches the location, the system transitions into a dynamic tracking mode, where continuous pose updates from the vision pipeline (at 13 FPS) are used to compute incremental corrections via the DLS based inverse kinematics method. During this phase, the system is not re-planning full RRT\* trajectories at each frame. Instead, the DLS controller operates in a visual-servoing loop, resolving the instantaneous pose error between the current end-effector position and the updated target pose. This separation between global planning (RRT\*) and local tracking (DLS-based servoing) ensures computational efficiency while maintaining smooth and responsive motion. These experiments demonstrated the framework's ability to transition seamlessly from visual pose estimation and motion planning to physical interaction, validating its robustness in both tracking and manipulation tasks. All motion planning and execution were carried out within the ROS MoveIt framework, which ensured smooth and adaptive path optimization for precise object handling.

**Table 2.** Performance Metrics for Object Tracking and Grasping

Metric	Value	Description
Tracking Accuracy	96.7%	Proportion of frames with accurate pose estimation prior to occlusion caused by end-effector constraints.
Pose Estimation Error	$\pm 0.63$ cm	Mean spatial deviation between predicted and ground-truth object poses.
Detection Latency	75 ms	Average per-frame processing time for object detection and pose update, supporting real-time operation.
Detection Precision	97.1%	Fraction of correctly identified objects among all detections.
Detection Recall	96.5%	Fraction of actual objects successfully detected.
Runtime Performance	$\sim 13.30$ FPS	Average frame rate during continuous tracking and grasping.

To obtain quantitative performance insights, we conducted a series of trials using a textured book cover placed at multiple positions within the camera's field of view, under varying environmental conditions. The system achieved a tracking accuracy of 96.7% as given in Table 2, indicating consistent pose estimation until occlusion occurred due to end-effector interference. Pose estimation error remained within  $\pm 0.63$  cm, confirming the system's suitability for precision grasping tasks. Pose estimation error ( $\pm 0.63$  cm) was computed by comparing the output of the vision pipeline to the ground-truth object pose provided by the Gazebo simulation environment. The detection pipeline maintained an average latency of 75 ms per frame, enabling real-time responsiveness at approximately 13.30 FPS. Detection precision and recall were measured at 97.1% and 96.5%, respectively, validating the system's robustness in identifying and localizing target objects under challenging lighting and background conditions. These results affirm the system's applicability to dynamic manipulation tasks in semi-structured environments, with strong generalization across object types and camera perspectives. The demonstrated performance highlights its potential for deployment in practical domains such as automated sorting, assistive robotics, and mobile manipulation.

#### 4.0.1. Evaluation of Trajectory Motions

Maximum errors and RMSE of cartesian motions obtained using DLS method were shown in Table 3.

**Table 3.** Maximum error and RMSE of Cartesian motions obtained using DLS method

Error	x	y	z	$\alpha$	$\beta$	$\gamma$
Maximum error(mm/deg)	1.06	1.11	0.74	1.75	0.94	1.05
RMSE(mm/deg)	0.97	0.81	0.66	1.05	0.71	0.88

The maximum translational errors range from 0.74 mm to 1.11 mm, while rotational errors peak at 1.75 deg. The RMSE values indicate consistent performance, with translational deviations remaining below 1 mm and rotational deviations under 1.10 deg. These results demonstrate the DLS method's effectiveness in maintaining low pose estimation errors across the trajectory. Velocity Continuity (VC), Acceleration Profile (AP), jerk, snap, smoothness and RMSE errors of end effector were evaluated. Maximum values of AP, jerk, and snap values of motions were calculated to analyse the behaviour of motions. VC, AP, jerk, and snap values of manipulator joints are given in Table 4.

Table 4 presents the dynamic motion metrics of the manipulator joints, including velocity continuity (VC), acceleration profile (AP), jerk, and snap, evaluated across all seven joints ( $\theta_1$  to  $\theta_7$ ). The VC values range from 0.08 deg/s to 0.30 deg/s, indicating smooth transitions in joint velocities without abrupt discontinuities. The acceleration profiles span from 0.75 deg/s<sup>2</sup> to 1.9 deg/s<sup>2</sup>, reflecting the rate of change in velocity during motion execution. Jerk values, which quantify the variation in acceleration, remain below 0.41 deg/s<sup>3</sup>, suggesting well-regulated dynamic behavior. Similarly, snap

values representing the fourth derivative of position are maintained below  $0.46 \text{ deg/s}^4$ , confirming the overall smoothness and stability of the joint trajectories. These metrics collectively demonstrate that the motion planning and control strategies employed yield dynamically consistent and mechanically safe joint movements suitable for precise and compliant manipulation tasks.

**Table 4.** Velocity Continuity( $\text{deg/s}$ ) , Acceleration Profile( $\text{deg/s}^2$ ) , jerk( $\text{deg/s}^3$ ) and snap( $\text{deg/s}^4$ ) values of manipulator joints

Metrics	$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$	$\theta_5$	$\theta_6$	$\theta_7$
VC	0.29	0.24	0.11	0.30	0.26	0.08	0.24
AP	1.49	1.08	0.88	1.9	1.58	0.75	1.88
Jerk	0.39	0.33	0.28	0.41	0.30	0.19	0.37
Snap	0.46	0.40	0.37	0.44	0.32	0.24	0.43

DLS performed better in terms of VC, AP, jerk and snap values of motions. Motions obtained using JT method exhibited lower VC, unstable AP, higher jerk and higher snap values.

Table 5 presents the smoothness values associated with the individual joint motions of the manipulator, denoted as  $\theta_1$  through  $\theta_7$ . The smoothness metric serves as a quantitative indicator of trajectory continuity, where lower values correspond to reduced dynamic fluctuations in joint motion specifically in velocity, acceleration, jerk, and snap thereby reflecting smoother and more mechanically stable trajectories. These values quantify the continuity and fluidity of joint trajectories during execution. The results indicate that most joints exhibit smooth motion profiles, with values ranging from 0.78 to 1.57. Notably, joint  $\theta_7$  shows a higher smoothness value of 2.14, suggesting increased variability or dynamic complexity in its motion. Overall, the smoothness metrics reflect well-conditioned joint behavior, contributing to stable and precise end-effector control throughout the manipulation task.

**Table 5.** Smoothness value of manipulator motions

Joint	Values
$\theta_1$	0.78
$\theta_2$	1.57
$\theta_3$	0.81
$\theta_4$	1.03
$\theta_5$	0.86
$\theta_6$	0.79
$\theta_7$	2.14

#### 4.1. Discussion

DLS method determined feasible joint motions with optimum accuracy and feasibility. It prevented the occurrences of high joint velocities and provide a smooth motion even near to singular regions by employing the appropriate damping factor. DLS method was employed to compute joint configurations for the manipulator, particularly in scenarios involving near-singular configurations and unreachable target poses. The core logic of the DLS approach involves assessing the contribution of each joint angle to the end-effector's motion, comparing this influence to the remaining distance to the goal, and applying damping proportionally. When a joint's influence significantly exceeds the required motion, its update is more aggressively damped. This is achieved by constraining the maximum allowable change in joint angles, thereby preventing abrupt or non-feasible motions. In regions close to kinematic singularities, selecting an appropriate damping factor is critical. A poorly tuned damping constant can lead to excessive joint velocities or oscillatory behavior, especially when the manipulator attempts to reach unreachable targets. In such cases, the manipulator tends to extend toward the goal, approach a singular configuration, and oscillate due to the limitations of the Jacobian matrix, which

provides only a first-order approximation of the end-effector's motion. These higher-order effects become increasingly significant near singularities. To mitigate this, a linear model of the forward kinematics was constructed using screw theory. While effective in most regions, this model may lose accuracy near the workspace boundaries, where second-order corrections could be necessary.

Empirical tuning of the damping constant revealed a trade-off between tracking precision and motion stability. Lower damping values minimized positional errors but introduced oscillations, while higher damping suppressed oscillations at the cost of accuracy. A compromise value was selected to balance these effects, resulting in smooth joint trajectories with tolerable error margins. The DLS method proved computationally efficient and capable of converging to feasible joint solutions without excessive effort. Overall, the DLS approach demonstrated robustness in handling ill-conditioned Jacobians and singular configurations. The joint solutions obtained were considered optimal for the seven degrees of freedom of the manipulator, enabling stable and precise motion planning across diverse task scenarios.

While the experiment utilized a textured book cover as a proxy object, the demonstrated capabilities particularly robust tracking of textured planar surfaces and smooth, redundant motion planning are directly transferable to high-value tasks in a SDL environment. We acknowledge that the current vision pipeline, which relies on feature-based detection and homography-driven pose estimation, cannot generalize to more challenging SDL objects such as transparent glassware, reflective instrument panels, or low-texture plastic microplates. These object classes often lack sufficient visual features for reliable SIFT-based tracking and require alternative sensing modalities (e.g., depth fusion, polarization imaging, or thermal vision) or learning-based approaches (e.g., deep neural networks trained on domain-specific datasets) to ensure robust perception. However, the primary objective of this study is to establish a foundational motion and control framework that is modular and sensor-oriented. The architecture is designed to accommodate future upgrades to the perception stack without requiring changes to the underlying kinematic modeling or trajectory optimization methods. In this regard, the current implementation serves as a proof-of-concept that validates the motion planning and control pipeline under ideal visual conditions, providing a baseline for future extensions.

## 5. Conclusion

This study introduced a multimodal control framework that integrates vision-guided object tracking with Jacobian-based motion planning for autonomous manipulation in SDL environments. The proposed system combined feature-based detection and homography-driven pose estimation with RRT\*-based trajectory generation and inverse kinematics computed via the Damped Least Squares (DLS) method. Kinematic modeling using screw theory enabled robust workspace analysis and stable joint solutions, even in the presence of redundancy and near-singular configurations. Quantitative evaluations demonstrated the effectiveness of the framework. The manipulator achieved a maximum Cartesian pose error of 1.11 mm and a root mean square error (RMSE) of 0.97 mm across translational axes. Rotational RMSE remained below 1.10 deg, confirming precise orientation tracking. Joint motion profiles showed velocity continuity values ranging from 0.08 deg/s to 0.30 deg/s, with acceleration profiles peaking at 1.9 deg/s<sup>2</sup>. Higher-order metrics such as jerk and snap were maintained below 0.41 deg/s<sup>3</sup> and 0.46 deg/s<sup>4</sup>, respectively, indicating smooth and dynamically stable trajectories. Additionally, joint smoothness values ranged from 0.78 to 2.14, with most joints exhibiting consistent motion behavior. Despite these promising results, certain limitations persist particularly in handling edge-of-workspace scenarios and highly nonlinear object motions. Future work will focus on enhancing the linear kinematic model with second-order approximations to improve accuracy near workspace boundaries. Real-time feedback from tactile and force-torque sensors will be incorporated to enable compliant manipulation and safe human-robot interaction. Furthermore, extending the framework to support multi-arm coordination and mobile base integration will enhance scalability and operational reach. Finally, benchmarking the system across diverse laboratory tasks and deploying it in real-world

experimental settings will be essential for validating its generalizability and impact on autonomous scientific discovery.

This work demonstrated foundational capabilities that align closely with the operational demands of a SDL. By showcasing reliable tracking of textured planar objects and smooth, redundant motion execution, the system lays groundwork for autonomous manipulation tasks central to SDL workflows. These include handling labeled reagent bottles, engaging with instrument interfaces, and managing microplate logistics. The use of a textured book cover, while simplified, effectively illustrated the system's potential to generalize across SDL-relevant tasks. Future work will focus on enhancing the perception module to handle non-ideal laboratory objects by integrating multi-modal sensing and deep learning-based object detection and pose estimation. Additionally, we aim to evaluate the system's performance in real-world SDL scenarios involving diverse object geometries, material properties, and environmental conditions. These extensions will further validate the scalability and robustness of the proposed framework in practical autonomous experimentation workflows.

**Author Contributions:** Conceptualization, methodology, software implementation, and manuscript preparation were carried out by Shifa Sulaiman. Amarnath A H contributed to simulations of work. Simon Bøgh and Naresh Marturi provided supervision, critical review, and guidance throughout the research and manuscript refinement process. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Pioneer Center for Accelerating P2X Materials Discovery, CAPeX.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Dobrokvashina, A.; Sulaiman, S.; Gamberov, T.; Hsia, K.H.; Magid, E. New Features Implementation for Servosila Engineer Model in Gazebo Simulator for ROS Noetic. In Proceedings of the Proceedings of the International Conference on Artificial Life and Robotics. ALife Robotics Co., Ltd., 2023, Vol. 28, pp. 153–156.
2. Sultanov, R.; Sulaiman, S.; Tsoy, T.; Chebotareva, E. Virtual collaborative cells modeling for UR3 and UR5 robots in Gazebo simulator. In Proceedings of the Proceedings of the 2023 International Conference on Artificial Life and Robotics, 2023, pp. 149–152.
3. Sulaiman, S.; Harikumar, A.; Marturi, N.; Bøgh, S. Comparative Analysis of Jacobian-Based Motion Planning Methods for Redundant Manipulators. In Proceedings of the Proceedings of the 2025 IEEE International Conference on Advanced Robotics and its Social Impacts (ARSO). IEEE, July 2025, pp. 259–264.
4. Liao, Z.; Jiang, G.; Zhao, F.; Mei, X.; Yue, Y. A novel solution of inverse kinematic for 6R robot manipulator with offset joint based on screw theory. *International Journal of Advanced Robotic Systems* **2020**, *17*, 1729881420925645.
5. Gandhi, N.; Nagababu, G.; Vistapalli, J. Modeling and Kinematic Analysis of a Robotic Manipulator for Street Cleaning Applications Using Screw Theory. In Proceedings of the Advances in Industrial Machines and Mechanisms: Select Proceedings of IPROMM 2020. Springer, 2021, pp. 609–618.
6. Wang, Y.; Liang, X.; Gong, K.; Liao, Y. Kinematical research of free-floating space-robot system at position level based on screw theory. *International Journal of Aerospace Engineering* **2019**, *2019*.
7. Liu, Z.; Xu, J.; Cheng, Q.; Zhao, Y.; Pei, Y.; Yang, C. Trajectory planning with minimum synthesis error for industrial robots using screw theory. *International Journal of Precision Engineering and Manufacturing* **2018**, *19*, 183–193.
8. Ge, D. Kinematics modeling of redundant manipulator based on screw theory and Newton-Raphson method. *Journal of Physics: Conference Series* **2022**, *2246*, 012068.
9. Shifa, S.; Sudheer, A.P. Modelling of torso and dual arms for a humanoid robot with fixed base by using screw theory for dexterous applications. In Proceedings of the IOP Conference Series: Materials Science and Engineering, IOP Publishing, 2021, Vol. 1132, p. 012036.
10. Shifa, S.; Sudheer, A.P. Modeling of a wheeled humanoid robot and hybrid algorithm-based path planning of wheel base for the dynamic obstacles avoidance. *Industrial Robot: the international journal of robotics research and application* **2022**, *49*, 1058–1076.
11. Medrano-Hermosillo, J.A.; Lozoya-Ponce, R.; Ramírez-Quintana, J.; Baray-Arana, R. Forward Kinematics Analysis of 6-DoF Articulated Robot using Screw Theory and Geometric Algebra. In Proceedings of the 2022 XXIV Robotics Mexican Congress (COMRob). IEEE, 2022, pp. 1–6.

12. Sun, P.; Li, Y.; Chen, K.; Zhu, W.; Zhong, Q.; Chen, B. Generalized kinematics analysis of hybrid mechanisms based on screw theory and lie groups lie algebras. *Chinese Journal of Mechanical Engineering* **2021**, *34*, 1–14.
13. Hélénon, F.; Huber, J.; Amar, F.B.; Doncieux, S. Toward a Plug-and-Play Vision-Based Grasping Module for Robotics. *arXiv preprint arXiv:2310.04349* **2023**.
14. Kushwaha, V.; Shukla, P.; Nandi, G. Vision-based intelligent robot grasping using sparse neural network. *International Journal of Intelligent Robotics and Applications* **2025**, pp. 1–14.
15. Wang, H.; Zhao, Q.; Li, H.; 2, R.Z. Polynomial-based smooth trajectory planning for fruit-picking robot manipulator. *Information Processing in Agriculture* **2022**, *9*, 112–122.
16. Zhang, H.; Tang, J.; Sun, S.; Lan, X. Robotic grasping from classical to modern: A survey. *arXiv preprint arXiv:2202.03631* **2022**.
17. Newbury, R.; Gu, M.; Chumbley, L.; Mousavian, A.; Eppner, C.; Leitner, J.; Bohg, J.; Morales, A.; Asfour, T.; Kragic, D.; et al. Deep learning approaches to grasp synthesis: A review. *IEEE Transactions on Robotics* **2023**, *39*, 3994–4015.
18. Du, G.; Wang, K.; Lian, S.; Zhao, K. Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. *Artificial Intelligence Review* **2021**, *54*, 1677–1734.
19. Zhang, S.; Lyu, B.; Li, Y.; Sui, L.; Yang, J. Trajectory planning of a 6R manipulator around obstacles by using SRRT\* Algorithm. In Proceedings of the The 2nd International Conference on Computing and Data Science, 2021, pp. 1–4.
20. Ortenzi, V.; Marturi, N.; Rajasekaran, V.; Adjigble, M.; Stolkin, R. Singularity-robust inverse kinematics solver for tele-manipulation. In Proceedings of the 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE). IEEE, 2019, pp. 1821–1828.
21. Wu, J.; Cui, Z.; Sheng, V.S.; Zhao, P.; Su, D.; Gong, S. A Comparative Study of SIFT and its Variants. *Measurement science review* **2013**, *13*, 122.
22. Muja, M.; Lowe, D.G. Fast approximate nearest neighbors with automatic algorithm configuration. In Proceedings of the Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP), 2009, Vol. 1, pp. 331–340.
23. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM* **1981**, *24*, 381–395. <https://doi.org/10.1145/358669.358692>.
24. Pardos-Gotor, J. *Screw theory in robotics: an illustrated and practicable introduction to modern mechanics*; CRC Press, 2021.
25. Lynch, K.M.; Park, F.C. *Modern robotics*; Cambridge University Press, 2017.
26. Buss, S.R. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation* **2004**, *17*, 16.
27. Paul, S.K.; Chowdhury, M.T.; Nicolescu, M.; Nicolescu, M.; Feil-Seifer, D. Object detection and pose estimation from rgb and depth data for real-time, adaptive robotic grasping. In *Advances in Computer Vision and Computational Biology: Proceedings from IPCV'20, HIMS'20, BIOCAMP'20, and BIOENG'20*; Springer, 2021; pp. 121–142.
28. Kato, N.; Iuchi, T.; Murabayashi, K.; Tanaka, T. Comparison of Smoothness, Movement Speed and Trajectory during Reaching Movements in Real and Virtual Spaces Using a Head-Mounted Display. *Life* **2023**, *13*, 1618.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.