UNIVERSITY OF BIRMINGHAM

Dual quaternion-based visual servoing for grasping moving objects

De Farias, Cristiana; Adjigble, Maxime; Tamadazte, Brahim; Stolkin, Rustam; Marturi, Naresh

[Link to publication on Research at Birmingham portal](#)

Dual Quaternion-Based Visual Servoing for Grasping Moving Objects

Cristiana de Farias[1,†], Maxime Adjigble[1], Brahim Tamadazte[2], Rustam Stolkin[1], Naresh Marturi[1]

Abstract— This paper presents a new dual quaternion-based formulation for pose-based visual servoing. Extending our previous work on local contact moment (LoCoMo) based grasp planning, we demonstrate grasping of arbitrarily moving objects in 3D space. Instead of using the conventional axis-angle parameterization, dual quaternions allow designing the visual servoing task in a more compact manner and provide robustness to manipulator singularities. Given an object point cloud, LoCoMo generates a ranked list of grasp and pre-grasp poses, which are used as desired poses for visual servoing. Whenever the object moves (tracked by visual marker tracking), the desired pose updates automatically. For this, capitalising on the dual quaternion spatial distance error, we propose a dynamic grasp re-ranking metric to select the best feasible grasp for the moving object. This allows the robot to readily track and grasp arbitrarily moving objects. In addition, we also explore the robot null-space with our controller to avoid joint limits so as to achieve smooth trajectories while following moving objects. We evaluate the performance of the proposed visual servoing by conducting simulation experiments of grasping various objects using a 7-axis robot fitted with a 2-finger gripper. Obtained results demonstrate the efficiency of our proposed visual servoing.

I. INTRODUCTION

Currently, many industries are increasingly using robotic manipulators on their production lines to perform pick and place of objects, e.g. bin picking, box packaging, etc. Over the years, grasping and manipulation of static objects has been well-explored, and a plethora of methodologies are presented in the literature [1]–[4]. With the recent advancements in computing for industrial automation, few industries have adopted to use vision systems to localize, track and grasp planar-moving objects, e.g., from a moving conveyor. However, most of these systems still rely on vacuum suction and assume constant moving speed where uncertainties in perception (e.g., error in predicted object pose due to varied speed) or inaccurate robot modeling may lead to pick-up failures. Nevertheless, handling arbitrarily moving objects in a 3D space is an open challenge to solve mainly due to the difficulties associated with planning arm plus hand motions while simultaneously tracking the object trajectory. The methodologies designed to grasp static objects are not feasible and robots need to be integrated with controllers bearing superior decision-making abilities to cope with uncertainties in perceived exteroceptive information.

[1] Extreme Robotics Laboratory, School of Metallurgy and Materials, University of Birmingham, Edgbaston, B15 2TT, UK. [2] Sorbonne Université, CNRS UMR 7222, INSERM U1150, ISIR, F-75005, Paris, France.
†Corresponding Author: {CXM1029}@student.bham.ac.uk
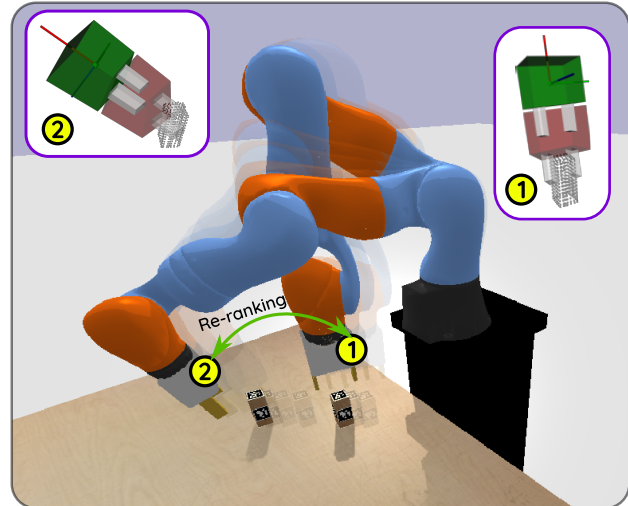
Fig. 1. Grasping arbitrarily moving objects in 3D space using the proposed dual quaternion based visual servoing. Arrow represents grasp re-ranking between two poses marked with circled numbers. (inset) Screenshots showing grasp candidates generated by LoCoMo at these two poses. Green gripper represents pre-grasp and red gripper is the grasp.

In this context, we present a singularity-robust dual quaternion-based visual servoing integrated with an efficient grasping algorithm to grasp arbitrarily moving objects in the robot's task space as shown in Fig. 1. A variety of visual servoing methods have emerged in the literature, which are broadly classified into image-based (IBVS), position-based (PBVS) and hybrid approaches [5]. In this work, we are particularly interested in PBVS where the object's relative position, estimated using perceived vision information, is used to control the manipulator's movements. Multiple methods are available in the computer vision literature to estimate the full six Degrees of Freedom (DoF) pose (3 rotations and 3 translations) of an object in 3D space [6]. This is a vital step in any PBVS where the estimated object pose is used to regulate the error between current and desired positions of the robot's end-effector. Consequently, any grasp planned on the object can be robustly updated or re-planned to the dynamic movements and perturbations.

Most of the existing state-of-the-art in grasping are *one-shot*, *i.e.*, they perceive once and plan grasps assuming the object is stationary; the arm is then commanded to the grasp pose without using any feedback [3]. However, they often fail in case of unplanned object movements or when there are uncertainties in the estimated hand to object relative poses. Few works have considered visual feedback to correct hand poses while reaching to grasp [7]–[10]. In [7],

Marturi et al. presented an approach to dynamically replan to grasp a moving object based on the vision information from two depth cameras, one hand mounted and the other scene camera. In [10], Kim et al. proposed a prediction-based programming by demonstration approach to catch objects in flight. An IBVS technique integrated with shared control grasping is presented in [8]. Although, most of these methods have shown satisfactory performance in grasping moving objects, their efficiency depends on the availability of inverse kinematic solutions throughout the trajectory. In addition, especially in the presence of redundant joints, manipulators might get caught in singular configurations.

Camera placement also plays a significant role in grasping moving objects. Two types of configurations are possible, *i.e.*, *eye-in-hand* where the camera is positioned on the robot end-effector; and *eye-to-hand*, where the camera is placed somewhere in the scene observing both robot end-effector and task object. For dynamic grasping tasks, the latter is often preferred as it does not suffer "blind spots" while visual tracking and is more biologically inspired. However, it still requires simultaneous tracking and pose estimation of both hand and object, which makes it challenging to use due to the lack of robust online tracking methodologies. As the main focus of this work is to design a new visual servoing controller, object pose estimation and visual tracking are not in its scope. For the sake of tracking object movements, we have used the conventional fiducial markers.

We use dual quaternions as our main basis for the visual servoing in this paper. Similar to homogeneous transformation matrices or angle-axis representations, dual quaternion can be used to represent full 6 DoF. Moreover, unit dual quaternions are singularity-free and have the benefit of being more compact and less computationally demanding than traditional representations, which makes them more suitable for visual servoing tasks. More specifically, they have proven to be an efficient and useful representation for many topics related to vision, such as hand-eye calibration [11], simultaneous mapping and localisation [12] and pose estimation and tracking [13]–[15]. Despite the increased interest on using dual quaternion for both robotic control and vision applications, literature which combines both is still scarce. No known instances of using dual quaternion for visual servoing are reported in the literature until very recently a method has been presented by Saltus et al. in [13]. There, the authors presented a PBVS schema to regulate the camera velocity with pose tracking achieved using an Extended Kalman filter.

In this paper, we take a different approach for the design of a dual quaternion-based visual controller by explicitly integrating grasping into our visual servoing framework. With this aim, we explore our previous local contact moment (LoCoMo)-based unknown object grasping [16] to grasp arbitrarily moving objects in a 3D space. Firstly, LoCoMo will generate a ranked list of stable grasp poses on the perceived point cloud data of the scene object. Combining this with dual quaternion-based visual servoing, we will demonstrate the task of tracking and grasping arbitrarily

moving objects. Capitalising on the dual quaternion spatial distance error between current end-effector pose and LoCoMo-provided pre-grasp pose, we built a dynamic grasp re-ranking metric to select the best feasible grasp for a moving object. In other words, the proposed visual servoing framework simultaneously analyses the quality of multiple grasp configurations and their distance to the gripper. Such analysis is performed in real-time, which allows the robot to switch between grasps as the object moves. Finally, as the robot dynamically tracks the object and switches among feasible grasps, it may encounter undesired configurations, such as singularities and joint limits. Therefore, the developed dual quaternion-based control strategy (while ensuring a singularity free representation) explores the robot null-space to provide a smooth motion avoiding joint limits. Overall, the main contributions of this paper are summarised as follows:

- A dual quaternion-based PBVS scheme described on the manipulator's joint space, which is robust to singularities. This ensures smooth motion for the manipulator while tracking moving objects.
- We extend our previous work on stationary unknown object grasping based on local contact moments to grasp arbitrarily moving objects. For this purpose, we designed a new ranking metric based on dual quaternion error and vantage-point tree (vp-tree) to dynamically select the best feasible grasp during tracking trajectories.

The remainder of this paper is organised as follows. In Section II, we introduce the problem and present the developed method pipeline. Section III presents the technical details of the developed method. Thereafter, in Section IV, we discuss the simulation experiments conducted using a 7-DoF robot fitted with a parallel-jaw gripper.

II. PROBLEM DESCRIPTION

A. Problem Statement

In general, PBVS problems aim at regulating the pose of a system based on visual information. Traditionally, these servoing systems are described using the axis-angle notation for rotations and have position and orientation decoupled. In this work, we tackle the problem of developing an efficient and generic robotic visual servoing strategy by designing a joint space visual regulator based on dual quaternion algebra.

Following dual quaternion notations, let us assume that the current pose is $\underline{x}$ and the desired pose is $\underline{x}_d$, the visual servoing problem is defined as

$$\arg\min \left[\underline{e}(\underline{x}, \underline{x}_d) \right] \tag{1}$$

where, $\underline{e}$ is the dual quaternion error function for the pose. Hence, our problem becomes finding a control strategy which controls the manipulator velocities to minimise $\underline{e}$. Given the perceived point cloud of an object, LoCoMo provides feasible grasp ($\underline{x}_g$) and pre-grasp ($\underline{x}_{pg}$) poses. These are then used as visual servoing reference poses $\underline{x}_d = \underline{x}_{pg} | \underline{x}_g$. More details are presented in the next section. The desired manipulator poses are automatically updated from the pose
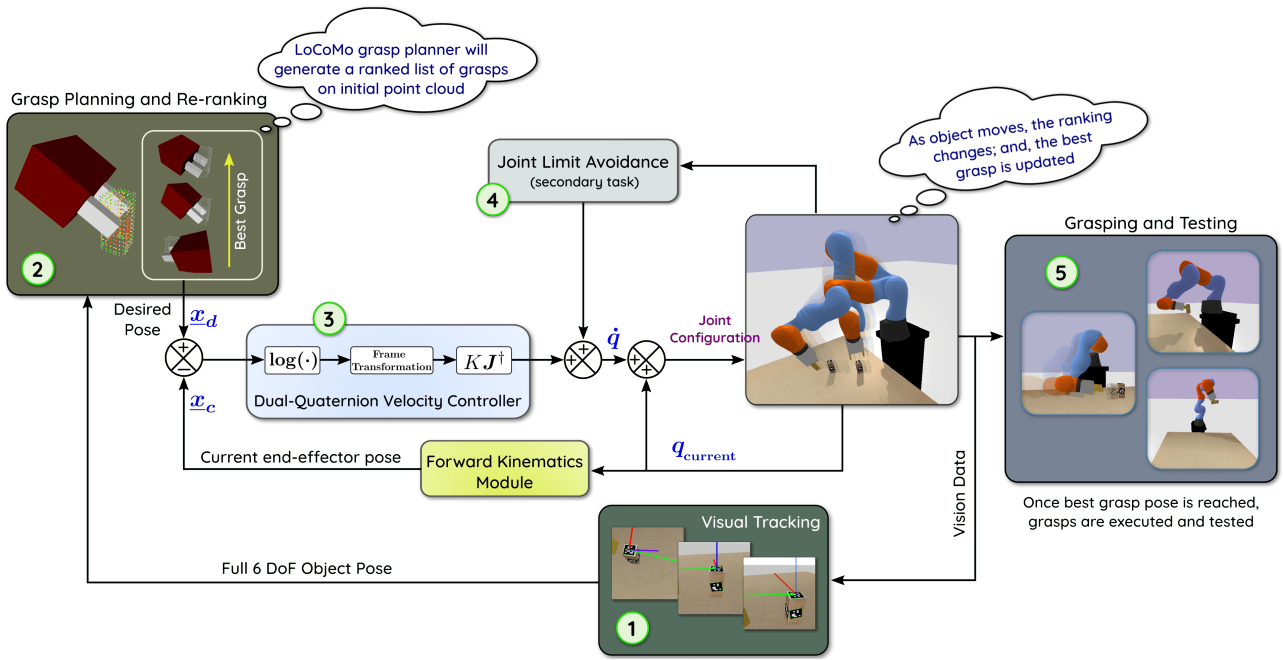
Fig. 2. Control flow of the proposed method to grasp moving objects. It consists of five stages, which are marked with circled numbers. Using point cloud captured by a scene-mounted depth camera, LoCoMo grasp planner provides initial best grasp candidates of the object. Simultaneously, object pose is estimated by tracking fiducial markers placed on the object. Whenever the object moves, the desired pose for visual servoing, *i.e.*, the pre-grasp pose computed by the grasp planner, is updated via re-ranking. Once the visual servoing converged and the object is grasped, the grasp stability is tested.

obtained via the object's visual tracking. Throughout this process, by using visual servoing instead of conventional path planning, we are able to accomplish moving object grasping.

B. Method Overview

Fig. 2 depicts the pipeline of our dual quaternion-based visual servoing to grasp arbitrarily moving objects. It consists of five different stages (numbers inside circles in figure): ① Visual tracking from fiducial markers, ② grasp planning from LoCoMo [16], ③ velocity regulator and robot joint space visual servo control, ④ joint limit avoidance as a secondary task projected at the robot null-space and ⑤ grasping and testing. The process starts with ①, where we capture the object point cloud and an image to estimate initial object pose. It is worth noting that we use a scene mounted and *hand-eye* calibrated RGB-D camera for this purpose. As mentioned earlier, we compute the pose by detecting the fiducial markers that are placed on the object's surface [17]. Alternatively, any other visual object pose tracker can be used. At each subsequent step, the object pose is updated and is passed to the grasp planning module. In ②, for the first iteration, a list of best grasping and pre-grasping poses are generated by our LoCoMo grasp planner on the object point cloud. Pre-grasping poses are selected as the poses with a fixed offset on the approach axis, as shown in Fig. 1. For the subsequent iterations, as the object moves, the best grasping pose may change due to kinematic constraints of the robot. Therefore, given the current manipulator configuration, at each step, the grasping poses are dynamically re-ranked and the best pre-grasping pose is used by the visual servoing as the desired pose $\underline{x}_d$ to reach. Once the robot is at pre-grasping pose, the grasping

pose is used as desired pose such that the robot can proceed to grasp the object. This two-point reach reduces the chance of collision between the end-effector and the object on its approach. In ③, we have the main controller that minimises the dual quaternion error as in (1). The error here is defined as the dual quaternion spatial distance between the end-effector's current pose and the desired pose from previous step. Additionally, to avoid joint limits, we add a secondary task projected to the robot's null space. This is depicted by ④ in Fig. 2. Finally, in ⑤, we execute the grasp and test its stability by performing a series of tests as described in [18]. All the technical details regarding our dual quaternion visual controller, grasp generation by LoCoMo and our re-ranking schema are described in the next sections.

III. METHODOLOGY

This section provides compact formulations of our PBVS schema using dual quaternion algebra instead of the traditional axis-angle notation [5].

A. Mathematical Preliminaries

Quaternions are an extension of imaginary numbers, often used to define three-dimensional rotations. First, let us consider $\hat{i}, \hat{j}, \hat{k}$ be the three quaternion unit vectors such that $\hat{i}^2 = \hat{j}^2 = \hat{k}^2 = \hat{i}\hat{j}\hat{k} = -1$. The algebra of quaternions [19] is generated by the basis elements 1, $\hat{i}$, $\hat{j}$, and $\hat{k}$, yielding the non-commutative set

$$\mathbb{H} \triangleq \left\{ \eta + \boldsymbol{\mu} : \ \boldsymbol{\mu} = \mu_1 \hat{i} + \mu_2 \hat{j} + \mu_3 \hat{k}, \ \eta, \mu_1, \mu_2, \mu_3 \in \mathbb{R} \right\}$$
(2)

where, for any quaternion $\boldsymbol{x} \in \mathbb{H}$, η and $\boldsymbol{\mu}$ are respectively the real and imaginary (or vector) parts. The quaternion

conjugate is then defined as $x^* \triangleq \eta - \mu$, which in turn allows to define the quaternion norm as $\|x\|^2 = xx^* = x^*x$. Particularly, we define the subset of unit norm quaternions $\mathcal{S}^3 \triangleq \{x \in \mathbb{H} : \|x\| = 1\}$ under multiplication to represent spatial rotations, and the pure quaternions subset $\mathbb{H}_0 \triangleq \{x_0 : \eta + \mu \in \mathbb{H}, \eta = 0\}$, to represent spatial translations. Those two subsets have important properties in representing rigid body motions. The vector isomorphism under $\mathbb{H}_0$ allows for operations such as inner and vector products and the group multiplication of unit quaternions. This guarantees that, for any $x_1, x_2 \in \mathcal{S}^3$, the quaternions multiplication $x_3 = x_1 x_2 \in \mathcal{S}^3$. Besides, a unit quaternion with rotation ϕ around the axis $n \in \mathbb{H}_0$, can be given by

$$x = \cos\left(\frac{\phi}{2}\right) + n\sin\left(\frac{\phi}{2}\right) \tag{3}$$

Let $x_1, x_2 \in \mathbb{H}_0$. The quaternion inner and vector products are

$$\langle x_1, x_2 \rangle \triangleq -\frac{x_1 x_2 + x_2 x_1}{2}$$
$$x_1 \times x_2 \triangleq \frac{x_1 x_2 - x_2 x_1}{2} \tag{4}$$

A further extension of imaginary numbers is the dual quaternion, which is introduced as non-minimal, non-commutative, singularity-free description of rigid body motions [20]. Dual quaternions are defined as the set

$$\mathbb{H} \otimes \mathbb{D} \triangleq \{\underline{x} = x_P + \varepsilon x_D \mid x_P, x_D \in \mathbb{H}\} \tag{5}$$

where, ε is called dual unit with $\varepsilon^2 = 0$, $\varepsilon \neq 0$, and x_P and x_D are the primary and dual parts of the dual quaternion $\underline{x}$, respectively. Furthermore, analogous to quaternions, the dual quaternion conjugate is $\underline{x}^* \triangleq x_P^* + \varepsilon x_D^*$ and subsequently, $\|\underline{x}\|^2 = \underline{x}\underline{x}^*$. Next, the unit dual quaternion subset under multiplication, $\underline{S} \triangleq \{\underline{x} \in \mathbb{H} \otimes \mathbb{D} : \|\underline{x}\| = 1\}$, forms the Lie group $\mathrm{Spin}(3) \ltimes \mathbb{R}^3$ that double covers $SE(3)$. The group inverse of $\underline{x} \in \underline{S}$ is $\underline{x}^*$. Finally, we define the pure dual quaternion as $\mathbb{H}_0 \otimes \mathbb{D} \triangleq \{\underline{x}_0 = x_P + \varepsilon x_D \mid x_P, x_D \in \mathbb{H}_0\}$ such that for $\underline{x}_0 \in \mathbb{H}_0 \otimes \mathbb{D}$, $\underline{x}_0^* = -\underline{x}_0$.

Let us consider $x_t \in \mathbb{H}_0$ and $x_r \in \mathcal{S}^3$ being an arbitrary translation and rotation of a rigid body. The full rigid body displacement can then be described as follows

$$\underline{x} = x_r + \varepsilon \frac{1}{2} x_t x_r, \tag{6}$$

and the kinematic equation for the body twist will be

$$\underline{\dot{x}} = \frac{1}{2}\underline{x}\, \underline{\omega}^B, \tag{7}$$

where, $\underline{\omega}^B \in \mathbb{H}_0 \otimes \mathbb{D}$ is the body twist and $\underline{\dot{x}}$ is the time derivative of $\underline{x}$. When dealing with rigid body motions, it is important to efficiently describe transformations between different frames. Let $^I\underline{x}_B \in \underline{S}$ be the transformation between the inertial frame $\mathbb{F}^I$ and the moving body frame $\mathbb{F}^B$. Then inertial twist is given by,

$$\underline{\omega}_I = {}^I\underline{x}_B\underline{\omega}^{BI}\underline{x}_B^* \tag{8}$$

The adjoint transformation is given by

$$\mathrm{Ad}(\underline{x})\underline{y} \triangleq \underline{x}\underline{y}\underline{x}^*, \tag{9}$$

with $\underline{x} \in \underline{S}$ and $\underline{y} \in \mathbb{H}_0 \otimes \mathbb{D}$. Using (9), the inertial twist can be re-written as $\underline{\omega}_I = \mathrm{Ad}(^I\underline{x}_B)\underline{\omega}^B$, or in (7), as $\underline{\dot{x}} = \frac{1}{2}\underline{\omega}^I\underline{x}$. Additionally, the logarithm operation involved within unit dual quaternion is

$$\log(\underline{x}) = \frac{\phi n}{2} + \varepsilon\frac{p}{2}, \tag{10}$$

where, p represents translation. Finally, we define an operator for one-by-one mapping $\mathrm{vec}_6 : \mathbb{H}_0 \otimes \mathbb{D} \rightarrow \mathbb{R}^6$ and its inverse operation $\underline{\mathrm{vec}}_6 : \mathbb{R}^6 \rightarrow \mathbb{H}_0 \otimes \mathbb{D}$.

B. Controller Design

Now, we show how dual quaternions are used to express the visual servoing controller. Let us consider $\underline{x}_c$ to be the current pose of the end-effector, obtained from the robot's forward kinematic model. Following (1), the error $\underline{e}$ to be regulated is given by

$$\underline{e} = (\underline{x}_c)^* \underline{x}_d. \tag{11}$$

where, $\underline{x}_d$ is the reference pose. Selection of the reference pose is explained in the next section. It is worth noting that all operations are performed with respect to the robot base coordinate frame. For this purpose, the visually estimated poses (grasps) are transformed to the base frame using camera-robot calibration. For more details, we refer the readers to [21].

In order to minimise the error and ensure convergence, we define a logarithmic controller that leads to a closed loop system, such as $\underline{\dot{e}} = -\frac{1}{2}\log(\underline{e})\underline{e}$. The stability of such a system is discussed in [22]. Thereby, using (7),

$$\underline{\dot{x}}_c = \frac{1}{2}\underline{x}_c\,\underline{\omega}_c^B \tag{12}$$

in which $\underline{\omega}_c^B$ is the current end-effector twist. Hence, from (11), the error's first order time derivative is $\underline{\dot{e}} = (\underline{\dot{x}}_c)^* \underline{x}_d$. Now, substituting for $\underline{\dot{x}}_c$ using (12), we get

$$\underline{\dot{e}} = \left(\frac{1}{2}\underline{x}_c\,\underline{\omega}_c^B\right)^* \underline{x}_d = -\frac{1}{2}\,\underline{\omega}_c^B\underline{x}_c^*\underline{x}_d = -\frac{1}{2}\,\underline{\omega}_c^B\underline{e} \tag{13}$$

Finally, given the desired closed loop dynamics, the current twist in the end-effector body frame can be defined as

$$-\frac{1}{2}\log(\underline{e})\underline{e} = -\frac{1}{2}\,\underline{\omega}_c^B\underline{e}$$
$$\underline{\omega}_c^B = \log(\underline{e}) \tag{14}$$

To perform visual servoing with a robotic manipulator we need to apply the control input in (14) to the robot joint space. With our design, we choose to control the robot's global pose; thus, we work with the input twist in the robot base frame, i.e., using (9), $\underline{\omega}_c^I = \mathrm{Ad}(\underline{x}_c)\underline{\omega}_c^B$. In Fig. 2, this operation is represented by the "frame transformation block" in stage ③. For the sake of simplicity, from now on, inertial twist will be simply denoted as $\underline{\omega}$ and unless stated, all variables are in the robot base frame. Considering $J \in \mathbb{R}^{6 \times N_{\mathrm{DoF}}}$ the dual quaternion-based geometric Jacobian,

which maps the coupled dual quaternion twist to the robot joint space as in [23], the joint space controller is defined as

$$q_{k+1} = q_k + K J^\dagger \operatorname{vec}_6 (\boldsymbol{\omega})$$
$$q_{k+1} = q_k + K J^\dagger \operatorname{vec}_6 (\operatorname{Ad}(\underline{\boldsymbol{x}}_c) \log(\underline{\boldsymbol{e}}))$$
(15)

where, q_{k+1} and q_k are the commanded and the current joint configurations, respectively, $K > 0$ is the proportional gain, and $J^\dagger$ is the Jacobian damped pseudo-inverse computed as $J^\dagger = J^T (J J^T + \lambda I_6)^{-1}$, with λ being a small scalar gain.

Since we consider a redundant robot in this work, we explore manipulator's null-space to avoid joint limits. For this purpose, we define a secondary task joint-space mapping as $J_s = q - \bar{q}$, with $\bar{q}$ being the mean position for each joint. The secondary task cost to be minimized can be defined as $\mathcal{C}_{\text{null}} = \frac{1}{2} \sum (q - \bar{q})^2$. Using the Jacobian J, we define a null-space projector as $P = (I_7 - J^\dagger J)$. Now, (15) with the added null-space term becomes

$$q_{k+1} = q_k + K J^\dagger \operatorname{vec}_6 (\boldsymbol{\omega}) + K_s P J_s^{-1} \mathcal{C}_{\text{null}},$$
(16)

with $K_s < 0$ being the secondary task gain. Note that the secondary task will not have any effect on the primary visual servoing task.

C. Grasp Planning and Re-ranking

Moving object grasping is performed by combining a grasp generator with the proposed visual servoing control scheme. As mentioned earlier, we have used LoCoMo-based grasp planner [16] to synthesise grasps on a task object. The advantage of using the LoCoMo grasp planner is two-fold. Firstly, the method neither requires any object 3D models nor offline training, making it easy to integrate with our visual servoing. Secondly, the grasp ranking scheme can be extended to provide dynamic grasp selection [24], which is ideal for moving object grasping. Given an object point cloud, an initial set of grasp and pre-grasp poses are generated using the LoCoMo grasp planner. The top-ranked grasp is selected and the visual servoing controller is initialized using this pose as the reference $\underline{\boldsymbol{x}}_d$. Grasp computation and ranking is given by

$$\mathcal{C}_i = \frac{1}{N_s} \sum_{j=1}^n ((2\pi)^n |\Sigma|)^{\frac{1}{2}} \mathcal{N}(\Psi_j; \vec{0}, \Sigma)$$
$$\mathcal{R} = \gamma \prod_{i=1}^{N_f} \mathcal{C}_i^{\omega_i}$$
(17)

where, $\mathcal{C}_i$ is the contact moment of each gripper finger i, $\mathcal{R}$ is the ranking score, N_s is a normalizing term, n is the number of point cloud patches sampled around the current location of the finger, $\mathcal{N}(\Psi_j; \vec{0}, \Sigma)$ is the multivariate Gaussian density function centered at $\vec{0}$ with co-variance Σ, Ψ_j is the difference of zero-moment shift vectors between object and gripper fingers surface, ω_i and γ are weighting factors, and N_f is the total number of fingers. More details can be found in [16].

While the object is moving, we visually track its pose, given by $\underline{\boldsymbol{x}}_o$, with a scene mounted camera. This in turn is used to update feasible grasps accordingly. To enable our controller to be as efficient as possible for dynamic grasping, a grasp re-ranking scheme is implemented. At each iteration, we select the $\mathcal{K}$ feasible grasps to the current gripper position using the dual quaternion distance. By feasible, we mean that there is an inverse kinematics solution from the current robot configuration. These set of grasps are then re-ranked and the top ranked grasp is used for visual servoing.

Even though all the generated grasps could be used for the re-ranking stage, using the $\mathcal{K}$ closest grasps allow to devise a more efficient algorithm, especially for real-time applications. We use a vantage-point tree (vp-tree) [25] for searching the nearest grasps to the robot's gripper. The vp-tree provides a fast and efficient nearest neighbor algorithm that can be used with any arbitrary metric space. For our method, we use the norm of the dual quaternion spatial error between two dual quaternions, e.g., for two poses $\underline{\boldsymbol{x}}_1, \underline{\boldsymbol{x}}_2$, this is computed as $\underline{\boldsymbol{x}}_e = 1 - \underline{\boldsymbol{x}}_1^* \underline{\boldsymbol{x}}_2$ (then mapped to $\mathbb{R}^8$ where we take the Euclidean norm in conjunction with the vp-tree to find the closest grasps. Using the $\mathcal{K}$ closest grasps and dual quaternion distances dq_i of the grasp pose i from the current gripper position, the grasp re-ranking Υ is defined as

$$\Upsilon = \frac{dq_i}{dq_{max} - dq_{min}} \text{, with } dq_{max} \neq dq_{min}$$
(18)

where, dq_{min} and dq_{max} are respectively the minimum and maximum dual quaternion distances between grasps and gripper pose among the $\mathcal{K}$ closest grasp candidates. Υ is updated at each iteration and the highest ranked grasp is used for the visual servoing controller. This technique, however, can create chattering problem in the controller if the best grasp changes too frequently due to noise. To account for that we have set a hysteresis to the switching strategy, *i.e.*, we only change the control reference pose if the new grasp ranking has improved at least by a switching threshold δ [7]. This strategy allows to have a smooth and dynamic reach-to-grasp trajectory by using the closest best grasp candidate available.

IV. EXPERIMENTAL RESULTS

In this section we present the simulation experiments conducted to validate the performance of our proposed dual quaternion based visual servoing and its ability in grasping arbitrarily moving objects.

A. Simulation Setup

All experiments are performed using the open source Python wrapper for Bullet physics engine, Pybullet, which has been proved to be an efficient and stable simulation tool for robotics [26]. Our robotic setup consists of a 7-DoF robot arm fitted with a parallel jaw gripper. For experiments we have selected three different objects from the YCB object dataset [27] that are both suitable with our gripper and allow for marker placement on their surfaces. In order to make the visual servoing system more realistic, we have also simulated a virtual rgb-d camera for both pose tracking and point cloud acquisition. Initially, we move the camera to three different
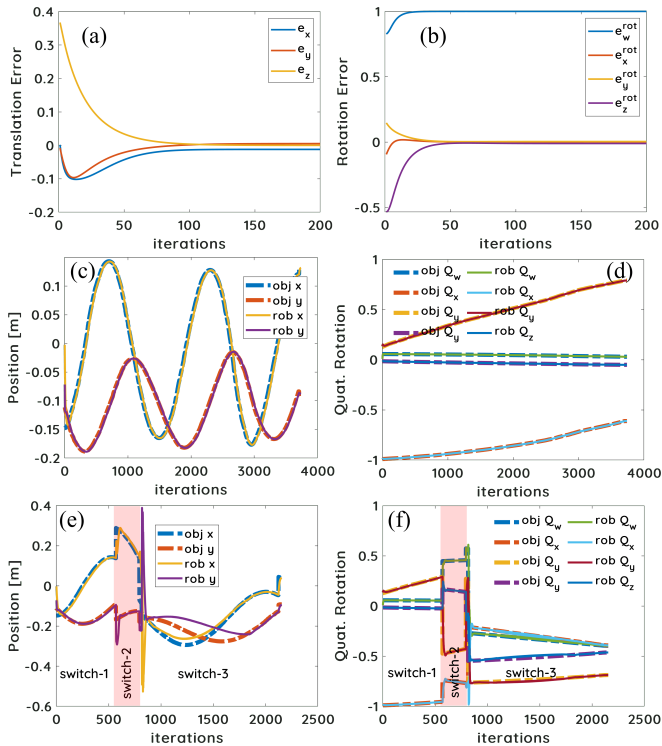
Fig. 3. Plots describing the convergence of the proposed method. Top row shows the (a) translation and (b) rotation errors convergence when approaching the desired grasp configuration with respect to a stationary object. In the second row, we show the robot tracking an object moving in a circular trajectory with the grasp re-ranking disabled; (c) and (d) shows respectively position and orientation errors between the end-effector and desired pre-grasp pose. In the last row, we show (e) the position and (f) orientation errors with the proposed method when the grasp re-ranking is enabled. The displayed trajectory contains three grasp switches, *i.e.,* re-ranking module adjusted the feasible grasp pose three times.

locations around the object and stitch the clouds captured at these locations to obtain a task point cloud for LoCoMo. Besides, for the sake of visual object pose tracking, we have used ArUco fiducial marker tracking [17] with markers placed on each object. Object CAD models downloaded from the YCB dataset with altered textures to include the makers have been used for the simulation. We report the results following the evaluation protocol presented in [18]. That is, we simulate different trajectories for each object to move and attempt to dynamically grasp them. Once grasped, we perform various tests as in [18] to evaluate grasp stability.

B. Controller Convergence Analysis

With this set of experiments, performed using "woodblock" object (YCB ID: 45), we test the convergence of our visual servoing in case of both static and moving objects. The plots shown in Fig. 3(a) and 3(b) respectively depict the translational and rotational evolution of our method for a stationary object. These results clearly show smooth convergence. Next tests are done to validate and demonstrate the convergence while following moving objects. Here, we test the behaviour in case of both grasp re-ranking enabled and disabled. When the re-ranking is disabled, the system might run close to singularities and joint limits, and we analyse the performance of our controller in such harsh conditions.

Plots in Fig. 3(c) and 3(d) show how our velocity regulator given by (16) can simultaneously follow same grasp without re-ranking. Comprehensively, it performed well where robot closely followed the trajectory. Plots in Fig. 3(e) and 3(f) show the behaviour when re-ranking given by (18) is enabled. Particularly, we show the desired configuration switching to three different grasps, with the last one remaining stable until the moment of grasping (at the end of plot). These results clearly demonstrate the efficiency of our controller. In case of grasp switching, the controller successfully stabilised mid-trajectories and continued tracking the object. Particularly, we note that the orientation error in both Fig. 3(d) and Fig. 3(f) remain low. That is one of the advantages of using the dual quaternion representation, which has coupled rotation and translation. Overall, the quick stabilisation to perturbations and robustness to singularities and joint limits make our dual quaternion-based visual servoing well suited for the problem of grasping moving objects.

C. Grasping Arbitrarily Moving Objects

The second set of experiments are conducted to evaluate the performance of our method in grasping arbitrarily moving objects. For this, we have identified 5 different trajectories for the object to move: (i) horizontal line from left side of the table to the right; (ii) vertical line, *i.e.*, object moving away from the robot; (iii) diagonal line, from left bottom to right upper corner; (iv) ellipsoid with radius $(15, 8)$ cm; (v) sine wave with an amplitude and frequency of 15 cm and 0.02 Hz, respectively. For each trajectory, we do two different tests: (i) we move the object in the space without any rotations; and (ii) we move the object with added arbitrary rotations.

Our visual servoing is initialised when the LoCoMo provides initial best grasp and pre-grasp poses. By using the pre-grasp pose as a reference pose, the robot starts moving towards it. This is when the object is moved in one of the five aforementioned trajectories. While the object moves, we constantly monitor the error between the desired pre-grasp configuration and the current end-effector pose. Once the sum of the squared error reaches a threshold, we update the desired pose to the best grasp pose. When the controller converges, the robot will execute the grasp with a force of 50 N. As mentioned, this two-point approach reduces the chances that the robot collides with objects. Once grasped, the object movement is stopped, and the grasp is evaluated for stability. For this, we conduct the stability tests defined in [18]. They are: (i) lifting test, in which we lift the object 20 cm above the table at the speed of 10 cm/s; (ii) rotation test, where we move the manipulator to be in $90 - 90$ configuration [18] and rotate the object from $90°$ to $-90°$ with a speed of 45 deg/s; and finally, (iii) shake test, where we shake the object in a sinusoidal pattern with an amplitude of 0.25 m and a peak acceleration of 10 m/s². These three tests are performed sequentially as in the order mentioned, and if the object slips from the gripper at any point, they are deemed failure and the next test is not performed.

Fig. 4 shows some screenshots of grasping moving objects using our method. Obtained results for all three objects, *i.e.,*
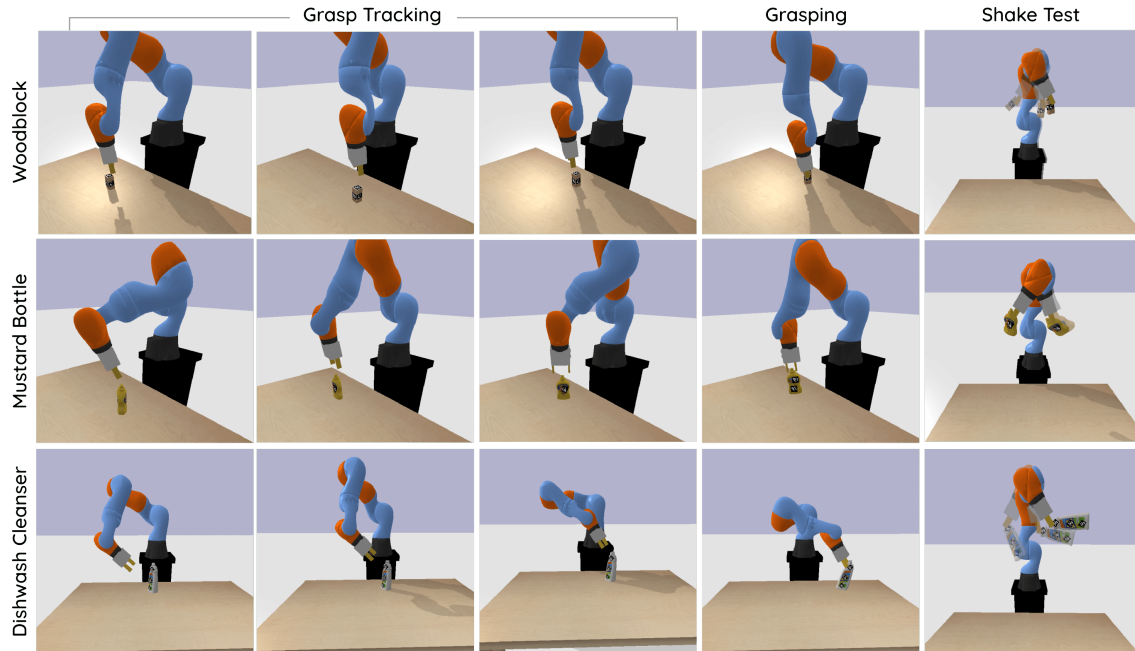
Fig. 4. Screenshots depicting the robotic grasping of moving objects. The first three columns show various positions during the trajectory and the fourth column images show object being grasped and the final column images show a grasp stability test.

woodblock (YCB id: 45), mustard bottle (YCB id: 9), and scrub cleanser (YCB id: 20), are summarised in the Tables I, II and III, respectively. We report the number of times the controller has switched the grasping configuration (re-ranking), the time from the start of the control loop until grasping, and the grasp success evaluation. Furthermore, we present the results with and without the null space control activated as well as the results with and without grasp re-ranking enabled.

Undoubtedly, the full method with null space control activated and re-ranking enabled, outperformed the others. On average, for all three objects respectively, the full method achieved $[\mathbf{80}\%, \mathbf{80}\%, \mathbf{70}\%]$ success while the method without null space achieved $[60\%, 60\%, 40\%]$ success and the method without re-ranking enabled achieved $[50\%, 30\%, 30\%]$ success. With full method, for the first two objects we got $\mathbf{100}\%$ success rate following and grasping the objects. We note that, due to joint constraints, the controller without the null space did not converge to grasp the object. We also note that the number of switches is higher without the null space. That is because when the null space control is deactivated, the re-ranking compensates unfeasible configurations. Out of all cases, the method without re-ranking showcased poor performance. We believe that this is due to the lack of inverse kinematics during trajectories. Overall, the obtained results clearly demonstrate the efficiency of our proposed dual quaternion-based visual servoing in grasping objects that are moving arbitrarily in a 3D space.

V. CONCLUSION

In this paper, we have presented a method for grasping moving objects by formulating a visual servoing controller based on dual quaternion algebra. For the problem of grasping we have incorporated a novel re-ranking strategy to the LoCoMo grasp planner, so that grasp candidates automatically update whenever the object moves. The dual quaternion proves to be advantageous when performing visual servoing for grasping. The coupled rotation and translation allow the manipulator to quickly converge to a desired grasp pose even in the presence of perturbations. Through simulation experiments, we have demonstrated that in combination with joint limit avoidance and dynamic re-ranking, our method demonstrates high success rate in grasping moving objects. In future works we plan on improving the control strategy with optimisation techniques. Furthermore, we also plan on extending this work by performing real robot experiments and integrating the object tracking with dual quaternions.

REFERENCES

[1] A. Bicchi and V. Kumar, "Robotic grasping and contact: a review," in *IEEE Int. Conf. on Rob. and Auto.*, vol. 1, 2000, pp. 348–353.
[2] A. Sahbani, *et al.*, "An overview of 3d object grasp synthesis algorithms," *Rob. Auton. Syst.*, vol. 60, no. 3, pp. 326–336, 2012.
[3] J. Bohg *et al.*, "Data-driven grasp synthesis—a survey," *IEEE Trans. on Rob.*, vol. 30, no. 2, pp. 289–309, 2013.
[4] C. de Farias *et al.*, "Simultaneous tactile exploration and grasp refinement for unknown objects," *IEEE Rob. and Auto. Lett.*, vol. 6, no. 2, pp. 3349–3356, 2021.
[5] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robot. Autom. Mag*, vol. 13, pp. 82–90, 2006.
[6] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: a hands-on survey," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 12, pp. 2633–2651, 2015.
[7] N. Marturi *et al.*, "Dynamic grasp and trajectory planning for moving objects," *Autonomous Robots*, vol. 43, no. 5, pp. 1241–1256, 2019.
[8] M. Gridseth, *et al.*, "On visual servoing to improve performance of robotic grasping," in *Conf. on Comp. and Rob. Vis.* 2015, pp. 245–252.
[9] F. Husain *et al.*, "Realtime tracking and grasping of a moving object from range video," in *IEEE Int. Conf. Rob. Auto.* 2014, pp. 2617–2622.
[10] S. Kim, A. Shukla, and A. Billard, "Catching objects in flight," *IEEE Trans. on Rob.*, vol. 30, no. 5, pp. 1049–1065, 2014.
[11] K. Daniilidis, "Hand-eye calibration using dual quaternions," *The Int. J. Rob. Res*, vol. 18, no. 3, pp. 286–298, 1999.

TABLE I

DYNAMIC GRASPING RESULTS WITH THE "WOODBLOCK" OBJECT FOR VARIOUS TESTING TRAJECTORIES AND FOR VARIOUS TEST CASES.

Trajectory	Full Method					Without NullSpace					Without Re-rank				
	Switch[1]	Time[2]	Lift[3]	Rot.[3]	Shake[3]	Switch[1]	Time[2]	Lift[3]	Rot.[3]	Shake[3]	Switch[1]	Time[2]	Lift[3]	Rot.[3]	Shake[3]
H. Line	1	28.81	100	100	100	0	11.07	100	100	100	NA	26.54	100	100	100
H. Line + Rot.	1	32.63	100	100	100	3	-	-	-	-	NA	30.42	100	100	100
V. Line	2	19.85	100	0	0	1	72.42	0	0	0	NA	10.93	0	0	0
V. Line + Rot.	1	19.313	100	100	100	1	29.71	0	0	0	NA	-	-	-	-
Diag.	1	19.63	100	100	100	3	17.15	100	100	100	NA	-	-	-	-
Diag. + Rot.	1	22.28	100	100	100	0	20.82	100	100	100	NA	29.4	100	100	100
Circle	0	88.29	100	100	100	0	11.32	100	100	100	NA	81.11	100	100	100
Circle + Rot.	3	91.32	100	100	100	4	97.1	0	0	0	NA	79.27	100	100	100
Wave	2	52.929	100	0	0	9	161.77	100	100	100	NA	-	-	-	-
Wave + Rot.	4	77.35	100	100	100	7	134.11	100	100	100	NA	-	-	-	-

[1] Number of times the re-ranking switched the reference pose.
[2] Time in seconds from the start pose until the object is grasped.
[3] Percentage of success (%) of lifting, Rotational and Shaking tests.

TABLE II

DYNAMIC GRASPING RESULTS WITH THE "SCRUB CLEANSER" OBJECT FOR VARIOUS TESTING TRAJECTORIES AND FOR VARIOUS TEST CASES.

Trajectory	Full Method					Without NullSpace					Without Re-rank				
	Switch	Time	Lift	Rot.	Shake	Switch	Time	Lift	Rot.	Shake	Switch	Time	Lift	Rot.	Shake
H. Line	2	30.85	100	100	100	3	48.15	100	100	100	NA	48.8	100	100	100
H. Line + Rot.	2	39.92	100	100	100	3	38.39	100	100	100	NA	-	-	-	-
V. Line	2	43.66	100	0	0	2	49.87	0	0	0	NA	-	-	-	-
V. Line + Rot.	2	37.76	100	100	0	2	31,08	100	100	0	NA	-	-	-	-
Diag.	4	110.26	100	100	100	3	39.55	100	100	100	NA	-	-	-	-
Diag. + Rot.	2	31.45	100	100	100	2	50.41	100	100	100	NA	-	100	-	-
Circle	3	76.81	100	100	100	0	62.44	0	0	0	NA	81.11	100	100	100
Circle + Rot.	5	88.07	100	100	100	4	214.04	100	0	0	NA	82.438	100	100	100
Wave	0	82.438	100	100	100	2	46.29	100	100	100	NA	41.29	100	100	100
Wave + Rot.	2	35.73	100	100	100	2	56.1	100	100	100	NA	156.32	0	0	0

TABLE III

DYNAMIC GRASPING RESULTS WITH THE "MUSTARD BOTTLE" OBJECT FOR VARIOUS TESTING TRAJECTORIES AND FOR VARIOUS TEST CASES.

Trajectory	Full Method					Without NullSpace					Without Re-rank				
	Switch	Time	Lift	Rot.	Shake	Switch	Time	Lift	Rot.	Shake	Switch	Time	Lift	Rot.	Shake
H. Line	0	26.681	100	0	0	2	59.09	100	100	100	NA	49.9	100	100	0
H. Line + Rot.	0	40.94	100	100	100	3	38.39	100	100	100	NA	37.88	100	100	0
V. Line	-	-	-	-	-	-	-	-	-	-	NA	52.32	0	0	0
V. Line + Rot.	2	91.75	100	100	100	2	97.56	100	100	100	NA	65.49	100	100	100
Diag.	2	45.31	100	100	100	5	83.55	100	0	0	NA	-	-	-	-
Diag. + Rot.	1	45.31	100	100	100	1	32.90	100	0	0	NA	-	-	-	-
Circle	5	88.07	100	100	100	7	197.1	0	0	0	NA	145.87	100	100	100
Circle + Rot.	3	179.42	100	100	100	3	66.4	100	100	100	NA	-	-	-	-
Wave	1	29.83	100	100	0	1	47.8	0	0	0	NA	48.09	100	100	0
Wave + Rot.	3	45.69	100	100	100	3	0	0	0	0	NA	48.57	100	100	100

[12] K. Li et al., "Simultaneous localization and mapping using a novel dual quaternion particle filter," in *Int. Conf. on Info. Fus.*, 2018, pp. 1668–1675.

[13] R. Saltus et al., "Dual quaternion visual servo control," in *IEEE Conf. Dec. Cont.*, 2020, pp. 5956–5961.

[14] Yuanxin Wu et al., "Strapdown inertial navigation system algorithms based on dual quaternions," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 1, pp. 110–132, 2005.

[15] B. Busam, T. Birdal, and N. Navab, "Camera pose filtering with local regression geodesics on the riemannian manifold of dual quaternions," in *IEEE Int. Conf. Comp. Vis. Workshops*, 2017, pp. 2436–2445.

[16] M. Adjigble et al., "Model-free and learning-free grasping by local contact moment matching," in *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 2018, pp. 2933–2940.

[17] S. Garrido-Jurado et al., "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[18] Y. Bekiroglu et al., "Benchmarking protocol for grasp planning algorithms," *IEEE Rob. Auto. Lett.*, vol. 5, no. 2, pp. 315–322, 2020.

[19] W. R. Hamilton, *Elements of quaternions*. Longmans, Green, & Company, 1866.

[20] J. M. Selig, *Geometric fundamentals of robotics*. Springer Science & Business Media, 2004.

[21] F. Chaumette, "Visual servoing," in *Robot Manipulators: Modeling, Performance Analysis and Control*, E. Dombre and W. Khalil, Eds. ISTE, 2007, ch. 6, pp. 279–336.

[22] D. Han, Q. Wei, and Z. Li, "A dual-quaternion method for control of spatial rigid body," in *IEEE Int. Conf. on Net., Sens. and Cont.*, 2008, pp. 1–6.

[23] L. F. C. Figueredo, B. V. Adorno, and J. Y. Ishihara, "Robust H-infinity kinematic control of manipulator robots using dual quaternion algebra," *Automatica*, 2021, pp.1–8.

[24] M. Adjigble et al., "An assisted telemanipulation approach: combining autonomous grasp planning with haptic cues," in *IEEE/RSJ Int. Conf. Intel. Rob. Sys.*, 2019, pp. 3164–3171.

[25] P. N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces," in *ACM-SIAM Symp. on Discrete algorithms*, 1993, pp. 311–321.

[26] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.

[27] B. Calli et al., "The ycb object and model set: Towards common benchmarks for manipulation research," in *Int. Conf. Adv. Robot.*, 2015, pp. 510–517.