Modeling Manipulator by Using Dual Quaternions from DH Form

Supervisor

Prof. Georgios Chamilothoris

PhD candidate

Eng. Abdullah Laalou

Abstract:

Robotic serial manipulators are essential tools in industrial automation and manufacturing processes. Accurate representation and analysis of their kinematics are crucial for precise control and efficient motion planning. Dual quaternions have emerged as a powerful mathematical framework for modeling the kinematics of robotic serial manipulators. Additionally, it offers a method to represent the modeling from the conventional Denavit-Hartenberg (DH) to Dual Quaternion. The quaternion can present the orientation without suffering from a singularity which can be found by DH.

1. Introduction

Robotic serial manipulators play a crucial role in numerous industries, facilitating automation, manufacturing, and other tasks requiring precise motion control. Accurate representation and analysis of their kinematics are essential for optimal performance. Dual quaternions have emerged as a powerful mathematical framework for modeling the kinematics of robotic serial manipulators. In this article, we provide a comprehensive exploration of the application of dual quaternions in robotic serial manipulators,
Singularity problem justifies the need of a robust and precise approach to perform the kinematic of robots. We Dual quaternion to avoid singularity problems.

2. Background Mathematical

Before going into the topic of quaternions, it is important to review some kind Number systems:

- Complex Number $\mathbb{C}$
- Quaternion Number System $\mathbb{H}$
- Dual Number $\mathbb{D}$
- Dual Quaternion $\mathscr{H}$

Complex Numbers $\mathbb{C}$

The set of complex numbers $\mathbb{C}$, can be understood as an extension of the set of real numbers $\mathbb{R}$. Any complex number can always be written in the form

$$c = a + b\,\hat{\imath}$$

where $a, b \in \mathbb{R}$. In addition, the imaginary unit, $\hat{\imath}$ has the following property $\hat{\imath}^2 = -1$

conjugation

$$(c)^* = a - b\,\hat{\imath}$$

Norm

$$\|c\| = \sqrt{c\,c^*} = \sqrt{(a + b\,\hat{\imath})(a - b\,\hat{\imath})} = \sqrt{a^2 + b^2}$$

Quaternion $\mathbb{H}$

$$h = a + b\;\hat{\imath} + c\;\hat{\jmath} + d\;\hat{k}$$

where $a, b, c, d \in \mathbb{R}$

and $\hat{\imath}, \hat{\jmath}, \hat{k}$ imaginary units have the following properties

$$\hat{\imath}\,\hat{\jmath} = -\hat{\jmath}\hat{\imath} = \hat{k}, \qquad \hat{\jmath}\,\hat{k} = -\hat{k}\hat{\jmath} = \hat{\imath}, \qquad \hat{k}\,\hat{\imath} = -\hat{\imath}\hat{k} = \hat{\jmath}$$

$\hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = \hat{\imath}\,\hat{\jmath}\,\hat{k} = -1$

Operation on quaternion

$$h_1 = a_1 + b_1\ \hat{\imath} + c_1\ \hat{\jmath} + d_1\ \hat{k}$$

$$h_2 = a_2 + b_2\ \hat{\imath} + c_2\ \hat{\jmath} + d_2\ \hat{k}$$

Sum/Subtraction

$$h_1 \pm h_1 = (a_1 \pm a_2) + (b_1 \pm b_2)\ \hat{\imath} + (c_1 \pm c_2)\ \hat{\jmath} + (d_1 \pm d_2)\ \hat{k}$$

Multiplication

$$h_1 h_2 = (a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2) + (a_1a_2 + b_1b_2 + c_1c_2 - d_1d_2)\ \hat{\imath} + (a_1a_2 - b_1b_2 + c_1c_2 + d_1d_2)\ \hat{\jmath} + (a_1a_2 + b_1b_2 - c_1c_2 + d_1d_2)\ \hat{k}$$

Where the real part, $Re(h) = a$.

if the real part equal zero, its s called a pure quaternion $\mathbb{H}_p$

Imaginary part, $Im(h) = b\ \hat{\imath} + c\ \hat{\jmath} + d\ \hat{k}$.

Conjugation

$$(h)^* = Re(h) - Im(h) = a - b\ \hat{\imath} - c\ \hat{\jmath} - d\ \hat{k}$$

Norm

$$\|h\| = \sqrt{h\ h^*}$$

Dual Number $\mathbb{D}$

Any dual number can be written in the form:

$$\underline{d} = a + \varepsilon\ b$$

where $a, b \in \mathbb{R}$. The algebra of dual numbers is based on the following property of the nilpotent dual unit $\varepsilon \neq 0$, $\varepsilon^2 = 0$.

Unit Quaternion and rotation of rigid bodies

When we extend the concept of dual numbers to be composed of two quaternions, we have what we call dual quaternions.

$$\underline{\boldsymbol{h}} = h + \varepsilon\ h'$$

Where $\underline{\boldsymbol{h}} \in \mathcal{H}$

Where $h, h' \in \mathbb{H}$

Note that $\mathbb{R} \subset \mathcal{H}$, $\mathbb{C} \subset \mathcal{H}$, $\mathbb{H} \subset \mathcal{H}$, $\mathbb{D} \subset \mathcal{H}$. That is, the real numbers, the complex numbers, the quaternions, and the dual numbers are all subsets of dual quaternions.

The primary part of the dual number

$$P(\underline{\boldsymbol{h}}) = h$$

The dual part of the dual number h'

$$D(\underline{\boldsymbol{h}}) = h'$$

Sum/Subtraction

$$\underline{\boldsymbol{h}}_{\mathbf{1}} \pm \underline{\boldsymbol{h}}_{\mathbf{2}} = (h_1 \pm h_2) + \varepsilon(h_1' \pm h_1')$$

Multiplication

$$\underline{\boldsymbol{h}}_{\mathbf{1}}\, \underline{\boldsymbol{h}}_{\mathbf{2}} = (h_1 h_2) + \varepsilon(h_1 h_2' + h_2 h_1')$$

Conjugation

$$(\underline{\boldsymbol{h}})^* = (h)^* + \varepsilon\,(h')^*$$

Norm

$$\|\underline{\boldsymbol{h}}\| = \sqrt{\underline{\boldsymbol{h}}\,\underline{\boldsymbol{h}}^*}$$

Real part

$$Re(\underline{\boldsymbol{h}}) = Re(h) + \varepsilon\, Re(h')$$

Imaginary part

$$Im(\underline{\boldsymbol{h}}) = Im(h) + \varepsilon\, Im(h')$$

Rigid Displacement:

Let $r = \cos\left(\frac{\vartheta}{2}\right) + v \sin\left(\frac{\vartheta}{2}\right)$ that represent a rotation

Let $t = (t_x\,, t_y\,, t_z)$ that represent the translation.

The pose transformation (Unit dual quaternions) can be presented by a translation followed by a rotation can always be written in the form

$$\underline{x} = r + \varepsilon\,\frac{1}{2}\,t\,r$$

We can represent DH to Dual Quaternion

The first is the rotation of $\boldsymbol{\theta}_{\mathbf{i}}$

$$\underline{\mathbf{x}}_{\mathbf{i}}^{\mathbf{i-1}} = \cos\left(\frac{\boldsymbol{\theta}_{\mathbf{i}}}{2}\right) + \hat{\mathrm{k}} \sin\left(\frac{\boldsymbol{\theta}_{\mathbf{i}}}{2}\right)$$

the second is a translation of $\mathbf{d}_{\mathbf{i}}$

$$\underline{\mathrm{x}}_{\mathrm{i}''}^{\mathrm{i}'}(\mathrm{d}_{\mathrm{i}}) = 1 + \varepsilon\frac{1}{2}\,\hat{\mathrm{k}}\,\mathrm{d}_{\mathrm{i}}$$

the third is the translation of $\mathbf{a_i}$

$$\underline{x}_{i'''}^{i''}(a_i) = 1 + \varepsilon \frac{1}{2}\, \hat{\imath}\, a_i$$

the fourth, and last, is the rotation of $\boldsymbol{\alpha_i}$

$$\underline{\mathbf{x}}_{\mathbf{i}}''' = \cos\left(\frac{\boldsymbol{\alpha_i}}{2}\right) + \hat{\imath}\sin\left(\frac{\boldsymbol{\alpha_i}}{2}\right)$$

So we can find the form from DH parameters:

$$\underline{x}_i^{i-1} = \begin{bmatrix} \cos(\boldsymbol{\alpha_i}/2)\cos(\boldsymbol{\theta_i}/2) \\ \sin(\boldsymbol{\alpha_i}/2)\cos(\boldsymbol{\theta_i}/2) \\ \sin(\boldsymbol{\alpha_i}/2)\sin(\boldsymbol{\theta_i}/2) \\ \cos(\boldsymbol{\alpha_i}/2)\sin(\boldsymbol{\theta_i}/2) \\ -\frac{a_i}{2}\sin(\boldsymbol{\alpha_i}/2)\cos\left(\boldsymbol{\theta_i}/2\right) - \frac{\mathbf{d_i}}{2}\cos(\boldsymbol{\alpha_i}/2)\sin(\boldsymbol{\theta_i}/2) \\ \frac{a_i}{2}\cos(\boldsymbol{\alpha_i}/2)\sin\left(\boldsymbol{\theta_i}/2\right) - \frac{\mathbf{d_i}}{2}\sin(\boldsymbol{\alpha_i}/2)\sin(\boldsymbol{\theta_i}/2) \\ \frac{a_i}{2}\cos(\boldsymbol{\alpha_i}/2)\sin\left(\boldsymbol{\theta_i}/2\right) + \frac{\mathbf{d_i}}{2}\sin(\boldsymbol{\alpha_i}/2)\cos(\boldsymbol{\theta_i}/2) \\ -\frac{a_i}{2}\sin(\boldsymbol{\alpha_i}/2)\sin\left(\boldsymbol{\theta_i}/2\right) + \frac{\mathbf{d_i}}{2}\cos(\boldsymbol{\alpha_i}/2)\cos(\boldsymbol{\theta_i}/2) \end{bmatrix}$$

References

[1] K. Erleben and S. Andrews, “Solving inverse kinematics using exact Hessian matrices,” *Computers & Graphics*, vol. 78, pp. 1–11, Feb. 2019, doi: https://doi.org/10.1016/j.cag.2018.10.012.

[2] X. Wang, X. Liu, L. Chen, and H. Hu, “Deeplearning damped least squares method for inverse kinematics of redundant robots,” *Measurement*, vol. 171, p. 108821, 2021, doi: https://doi.org/10.1016/j.measurement.2020.108821.

[3] F. Marić, M. Giamou, A. W. Hall, S. Khoubyarian, I. Petrović, and J. Kelly, “Riemannian Optimization for DistanceGeometric Inverse Kinematics,” *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1703–1722, doi: https://doi.org/10.1109/TRO.2021.3123841.

[4] A. Dahlin and Y. Karayiannidis, “Adaptive Trajectory Generation Under Velocity Constraints Using Dynamical Movement Primitives,” *IEEE Control Systems Letters*, vol. 4, no. 2, pp. 438–443, doi: https://doi.org/10.1109/LCSYS.2019.2946761.

[5] N. Zivkovic, J. Vidakovic, S. Mitrovic, and M. Lazarevic, “Implementation of Dual Quaternionbased Robot Forward Kinematics Algorithm in ROS,” in *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1–4. doi: https://doi.org/10.1109/MECO55406.2022.9797160.

[6] L. LechugaGutierrez, E. MaciasGarcia, G. MartínezTerán, J. ZamoraEsquivel, and E. BayroCorrochano, “Iterative inverse kinematics for robot manipulators using quaternion algebra and conformal geometric algebra,” *Meccanica*, vol. 57, no. 6, pp. 1413–1428, 2022, doi: https://doi.org/10.1007/s1101202201512w.

[7] A. Valverde and P. Tsiotras, “Spacecraft Robot Kinematics Using Dual Quaternions,” *Robotics*, vol. 7, no. 4, 2018, doi: https://doi.org/10.3390/robotics7040064.

[8] M. M. Schill and M. Buss, “Kinematic Trajectory Planning for Dynamically Unconstrained Nonprehensile Joints,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 728–734, 2017, doi: https://doi.org/10.1109/LRA.2017.2788197.

[9] H. Albedran and K. Jarmai, *Optimization for Robot Modelling with MATLAB*. 2020. doi: https://doi.org/10.1007/9783030404109.

Towards a Computationally Efficient Solution of the Inverse Kinematics problem using Machine Learning

Abdullah Laalou [1], Abdulkader Joukhadar [2], Hassan Alnaddaf [2] and Georgios Chamilothoris[3]

[1] Doctoral Candidate, Dpt. of Control and Automation, University of Aleppo, University Sq., Alshahaba District, Aleppo, Syria **abdullah.laalou@gmail.com**
[2] Dpt. of mechatronics, University of Aleppo, University Sq., Alshahaba District, Aleppo, Syria
[3] Dpt. Mechanical Engineering, University of West Attica, Petrou Ralli & Thivon 250, 12241 Athens, Greece **thor@uniwa.gr**

Abstract. The paper reports work-in-progress towards exploiting Machine Learning methods to solve the inverse kinematics (IK) problem in real-time, in order to control the pose (position and orientation) of the end-effector of a robotic arm. The work explores the use of Unit Dual Quartenion to the kinematic of a robot arm of serial architecture with six degrees of freedom (DOFs). Non-linear constrained optimization is applied to solve the IK problem taking into account actuation (motor) limitations in every DOF. The results show that this approach overcomes singularities and achieves a continuous "best realisable" solution, even when the target path is outside the reachable workspace.This sets the basis for further research work, namely building a off-line datasets that can be used to train a computationally efficient Machine-Learning model running in real-time.

Keywords: Robotics, Dual Quaternion, Inverse Kinematics, Constrained Optimization, Machine Learning

1. Introduction

Practically every practical robot application incorporates solving the inverse kinematics (IK) in real-time, in order to control the pose (position and orientation) of the end-effector. In general, the IK problem, i.e. the relationship between actuation and resulting pose, is highly non-linear and incorporates singularities and multiple solutions. The problem is further complicated when actuation constraints must be taken into account - as is typical in real-life robot mechanisms. In all cases, computing efficiency is critical in solving the IK problem, in order to to meet responsiveness and accuracy goals imposed by the robotic application.

Investigating mathematical approaches that facilitate the formulation and efficient solution of the IK problem, recent research has turned to unit dual quaternion (UDQ) representation of the robot pose recently [3]. Apparent advantages of the unit dual quaternion include its immunity to singularities, robustness to numerical errors, and higher storage and computational efficiencies. These features have attracted the interest of other ares, such as computer graphics and computer vision [2].

UDQ requires seven parameters to describe pose transformation in successive coordinate systems, instead of the 12 parameters required in the conventional homogeneous transformation matrix (HTM) approach. In addition, multiplications of dual quaternions are computationally less expensive than the matricx multiplications required for HTMs. Moreover, UDQ coefficients may be employed directly in control laws without the requirement for additional parameters to be extracted, as is the typical case when using HTM [4].

Standard approaches to solving the IK problem require the computation of the Jacobian matrix using the kinematic equation [5]. Often, the general IK problem is expressed as a constrained Quadratic Programming (QP) problem seeking to minimize a distance between the target and actual trajectory, while respectong actuation and other limitations [5][6].

After presenting the Unit Dual Quartenion in Section 1, the paper examines its use for modelling the kinematic of a robot arm of serial architecture with six degrees of freedom (DOFs), described in Section 3. Section 4 applies a non-linear constrained optimization method to solve the IK problem taking into account actuation (motor) limitations in every DOF. The results, in Section5, show that the method achieves a continuous "best realisable" solution, even when the target path is outside the reachable workspace. As discussed in Section 6, this sets the basis for further research work, namely building a off-line datasets that can be used to train a computationally efficient Machine-Learning model running in real-time.

2. Background

2.1 Quaternion $\mathbb{H}$:

Quaternion was introduced by Hamilton in 1843 which extended complex numbers in 4 dimensions. It has 3 imaginary parts and one real part.

$$h = q_1 + q_2\ \hat{\imath} + q_3\ \hat{\jmath} + q_4\ \hat{k} \tag{1}$$

Where $h \in \mathbb{H}$

where $q_1, q_2, q_3, q_4 \in \mathbb{R}$

and $\hat{\imath}, \hat{\jmath}, \hat{k}$ imaginary units have the following properties

$$\hat{\imath}\,\hat{\jmath} = -\hat{\jmath}\hat{\imath} = \hat{k}, \quad \hat{\jmath}\,\hat{k} = -\hat{k}\hat{\jmath} = \hat{\imath}, \quad \hat{k}\,\hat{\imath} = -\hat{\imath}\hat{k} = \hat{\jmath} \tag{2}$$

$$\hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = \hat{\imath}\,\hat{\jmath}\,\hat{k} = -1 \tag{3}$$

The unit quaternion can present the orientation without suffering from a singularity which makes it more robust but it is more complicated.

The unit quaternion represents rotation of the reference frame of rigid bodies in the three-dimensional space [9]

$$r = \cos\left(\frac{\vartheta}{2}\right) + v \sin\left(\frac{\vartheta}{2}\right) \tag{4}$$

Where ϑ is rotation angle around the rotation axis v.

2.2 The Dual Quaternion $\mathscr{H}$

The unit quaternion can describe only orientation with disregarding the translation, so we extended the concept of dual number $\mathbb{D}$ to be composed of two quaternion what we called dual quaternion.

The qual quaternion can be written in the form

$$\underline{h} = h + \varepsilon h' \tag{5}$$

$$\underline{h} = \left\{(q_1 + q_2\ \hat{\imath} + q_3\ \hat{\jmath} + q_4\ \hat{k}) + \varepsilon\left(q_5 + q_6\ \hat{\imath} + q_7\ \hat{\jmath} + q_8\ \hat{k}\right)\right\} \tag{6}$$

Where $\underline{h} \in \mathscr{H}$

Where $h, h' \in \mathbb{H}$

Where ε is dual unit (also known as Clifford unit) [10], $\varepsilon \neq 0, \varepsilon^2 = 0$

We called h primary part, h'dual part

Note that $\mathbb{R} \subset \mathcal{H}, \mathbb{C} \subset \mathcal{H}, \mathbb{H} \subset \mathcal{H}, \mathbb{D} \subset \mathcal{H}$. That is, the real numbers, the complex numbers, the quaternions, and the dual numbers are all subsets of dual quaternions.

A dual quaternion in expanded form $\underline{h} = \left\{(q_1 + q_2\ \hat{\imath} + q_3\ \hat{\jmath} + q_4\ \hat{k}) + \varepsilon\left(q_5 + q_6\ \hat{\imath} + q_7\ \hat{\jmath} + q_8\ \hat{k}\right)\right\}$. it can define the operator $\mathrm{vec}(\underline{h}) = [q_1 + q_2 + q_3 + q_4 + q_5 + q_6 + q_7 + q_8]^T$ that maps a dual quaternion into an 8-dimensionals vector[11].

3. Modeling Serial Robots

The rigid body are described by the position of the origin of the local frame and the orientation of its axes where the position of the origin of coordinate frame $\mathcal{F}_i$ relative to coordinate frame $\mathcal{F}_j$ can be denoted by the 3 × 1 vector. The most common way to represent the attitude of rigid body in minimal representation is a set of three Euler angles. [13]

The Euler angles represent a rotation in a specific sequence and it's commonly used because it presents orientation physically and easy to understand, On the other hand, it is suffered from singularity. For instance, if we use a sequence of rotation by fixed angle such 'XYZ' (called RPY roll pitch yaw angles) used fixed angle so we rotate about X axis then by Y axis then by Z axis. If the rotation about y equals 90° then we will have singularity (Gimbal Lock) which occurs when two rotational axis align and lose one degree of freedom.

3.1 Modeling Kinematic Using Denavit-Hartenberg Parameters:

The DH (Denavit-Hartenberg) parameters are the most often used form for describing a robot model. This convention was founded in 1955 by Jacques Denavit and Richard Hardenberg in order to standardize the coordinate frames for spatial links [1][13]. DH method ensures that the position and orientation of each frame can be described by only four parameters, which remove the complexity from the kinematic analysis and allow us to derive the equations of motion using the kinetic.

The kinematic modeling of a robot manipulator describes the relationship between the links and joints that compose its kinematic chain. The most popular methods use the Denavit-Hartenberg convention, which uses a minimal parameter representation of the kinematic chain, but on the other hand has some limitations. The direct kinematics analysis is the process of calculating the end-effector posture from the joint positions given, while the inverse kinematics analysis is the process of obtaining the joint positions necessary to establish a desired end-effector posture. Both analyses are important in motion study. [7]

The equation for DH convention (7)

$$A = \mathrm{Rot}_z(\theta_i)\ \mathrm{Tran}_z(d_i)\ \mathrm{Tran}_x(a_i)\ \mathrm{Rot}_x(\alpha_i) \tag{7}$$

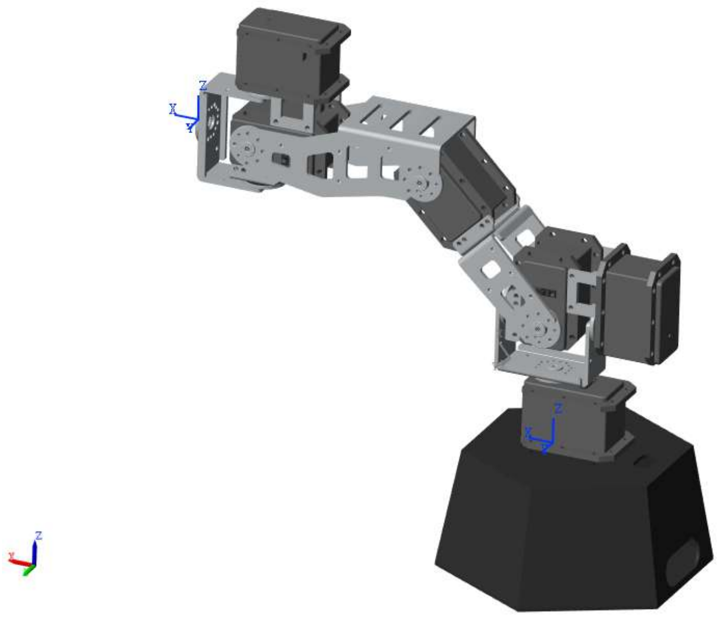

Fig 1. Manipulator with 6DOFs drawing by SolidWorks

Table1. DH parameters

Link	$\boldsymbol{\theta_i}$ (rad)	$\mathbf{d_i}$ (cm)	$\mathbf{a_i}$ (cm)	$\boldsymbol{\alpha_i}$ (rad)
1	$(\theta_1 + \frac{\pi}{2})$	d_1	0	$+\,pi/2$
2	$(\theta_2 + \frac{\pi}{2})$	0	0	$+\,pi/2$
3	$(\theta_3 + \frac{\pi}{2})$	0	a_3	0
4	θ_4	0	a_4	0
5	θ_5	0	0	$+\,pi/2$
6	θ_6	0	a_6	0

Note: $d_1 = 7$ cm, $a_3 = 9.3$ cm, $a_4 = 9.3$ cm, $a_6 = 3.635$ cm

3.2 Modeling Kinematic using Dual Quaternion:

Rigid Displacement:

Unit dual quaternions compose the set $\underline{\zeta}$, which represent pose transformations of the reference frame of rigid bodies in three-dimensional space.

Let $r = \cos\left(\frac{\vartheta}{2}\right) + v \sin\left(\frac{\vartheta}{2}\right)$ that represent a rotation about the unit vector v through ϑ where $\|r\| = 1$

Let $t = (t_x, t_y, t_z)$ that represent the translation.

The pose transformation (Unit dual quaternions) can be presented by a translation followed by a rotation can always be written in the form

$$\boxed{\underline{x} = r + \varepsilon \frac{1}{2} t r \qquad (8)}$$

Where $\underline{x} \in \underline{\zeta}$, $\|\underline{x}\| = 1$.

Forward Kinematics Model (FKM)

The forward kinematic model (FKM), concerns finding the function that maps the joint space variables into the task space variables.

Note the kinematic model of a robot models the motion of the robot without considering forces. It concerns only the geometry of the robot. This means that weight, inertia, and so on are not considered in kinematic modeling

Let $\underline{x}_0^w \in \underline{\zeta}$ be reference frame at the base of the robot. For convenience, it can coincide with the reference frame of the first joint of the robot.

So, the pose of the end effector will be:

$$\underline{x}_E^W = \underline{x}_1^0 \underline{x}_2^1 \underline{x}_3^2 \dots \underline{x}_n^{n-1} \tag{9}$$

The DH parameters provide a systematic way to calculate each individual joint transformation of any n-DoF serial manipulator. Each joint transformation $\underline{x}_i^{i-1}(\theta_i) \in \underline{\zeta}$,where $i = 1,2, \ldots\ldots n$ is composed of four intermediate transformations, as follows

$$\underline{x}_i^{i-1} = \underline{x}_{i'}^{i-1}(\theta_i)\, \underline{x}_{i''}^{i'}(d_i)\, \underline{x}_{i'''}^{i''}(a_i)\, \underline{x}_i^{i'''}(\alpha_i) \tag{10}$$

where the DH parameters, for each joint, are$\boldsymbol{\theta_i}$, $\mathbf{d_i}, \mathbf{a_i}, \boldsymbol{\alpha_i} \in \mathbb{R}$. Each of those parameters is related to one transformation. The first is the rotation of $\boldsymbol{\theta_i}$ about the z-axis of frame $\mathcal{F}_{i-1}$.

$$\underline{\mathbf{x}}_{\mathbf{i}}^{\mathbf{i-1}} = \cos\left(\frac{\boldsymbol{\theta_i}}{2}\right) + \hat{k}\sin\left(\frac{\boldsymbol{\theta_i}}{2}\right) \tag{11}$$

the second is a translation of $\mathbf{d_i}$ about the z-axis of frame $\mathcal{F}_{\mathbf{i}'}$.

$$\underline{x}_{i''}^{i'}(d_i) = 1 + \varepsilon\frac{1}{2}\,\hat{k}\,d_i \tag{12}$$

the third is the translation of $\mathbf{a_i}$ about the x-axis of frame $\mathcal{F}_{\mathbf{i}''}$,

$$\underline{x}_{i'''}^{i''}(a_i) = 1 + \varepsilon\frac{1}{2}\,\hat{\imath}\,a_i \tag{13}$$

the fourth, and last, is the rotation of $\boldsymbol{\alpha_i}$ about the x-axis of frame $\mathcal{F}_{\mathbf{i}'''}$

$$\underline{\mathbf{x}}_{\mathbf{i}}^{\prime\prime\prime} = \cos\left(\frac{\boldsymbol{\alpha_i}}{2}\right) + \hat{\imath}\sin\left(\frac{\boldsymbol{\alpha_i}}{2}\right) \tag{14}$$

So, we can find the form from DH parameters:

$$\underline{x}_i^{i-1} = \begin{bmatrix} \cos(\boldsymbol{\alpha_i}/2)\cos(\boldsymbol{\theta_i}/2) \\ \sin(\boldsymbol{\alpha_i}/2)\cos(\boldsymbol{\theta_i}/2) \\ \sin(\boldsymbol{\alpha_i}/2)\sin(\boldsymbol{\theta_i}/2) \\ \cos(\boldsymbol{\alpha_i}/2)\sin(\boldsymbol{\theta_i}/2) \\ -\frac{a_i}{2}\sin(\boldsymbol{\alpha_i}/2)\cos(\boldsymbol{\theta_i}/2) - \frac{\mathbf{d_i}}{2}\cos(\boldsymbol{\alpha_i}/2)\sin(\boldsymbol{\theta_i}/2) \\ \frac{a_i}{2}\cos(\boldsymbol{\alpha_i}/2)\sin(\boldsymbol{\theta_i}/2) - \frac{\mathbf{d_i}}{2}\sin(\boldsymbol{\alpha_i}/2)\sin(\boldsymbol{\theta_i}/2) \\ \frac{a_i}{2}\cos(\boldsymbol{\alpha_i}/2)\sin(\boldsymbol{\theta_i}/2) + \frac{\mathbf{d_i}}{2}\sin(\boldsymbol{\alpha_i}/2)\cos(\boldsymbol{\theta_i}/2) \\ -\frac{a_i}{2}\sin(\boldsymbol{\alpha_i}/2)\sin(\boldsymbol{\theta_i}/2) + \frac{\mathbf{d_i}}{2}\cos(\boldsymbol{\alpha_i}/2)\cos(\boldsymbol{\theta_i}/2) \end{bmatrix} \quad (15)$$

Inverse Kinematic Model (IKM)

The inverse kinematic model (IKM), concerns finding the opposite mapping of the FKM, that is, a function that maps the task space variables into joint space variables.

Differential Kinematics Model (DKM)

The first-order time-derivative of the FKM is called Differential Kinematics Model (DKM). It maps the joint-space velocities into task-space velocities in the general form (17).

$$gx = f(\boldsymbol{\theta}) \quad (16)$$

$$\dot{x} = J\,\dot{\boldsymbol{\theta}} \quad (17)$$

Where $\boldsymbol{\theta} \triangleq \boldsymbol{\theta}(t) \in \mathbb{R}^{\mathbf{n}}$ is the vector of the manipulator joint configuration, $x \triangleq x(t) \in \mathbb{R}^{\mathbf{m}}$ is the vector of the m task variables, $J \triangleq J(\boldsymbol{\theta}) \in \mathbb{R}^{\mathbf{m\,x\,n}}$ is a Jacobian matrix.

In Dual Quaternion the Differential Kinematics will be ().[10]

$$\mathrm{vec}_8\, \dot{\underline{x}} = J_{\underline{x}}\, \dot{\boldsymbol{\theta}} \quad (18)$$

Where $J_{\underline{x}} \in \mathbb{R}^{\mathbf{8\,x\,n}}$ is dual quaternion analytical Jacobian,

4. Constrained Optimization:

There are many types of optimization problems one of them Unconstrained optimization which have the form (19)

$$u \in \underset{\dot{\boldsymbol{\theta}}}{\arg\min}\, f(\dot{\boldsymbol{\theta}}) \tag{19}$$

Which mean find all decision variable $\dot{\boldsymbol{\theta}}$ that minimize objective function $f(\dot{\boldsymbol{\theta}})$ and return one of them as u.

Using unconstrained optimization to solve robot-control tasks is useful in some cases, but most tasks in practice have what we call hard limits. They are called that way in the sense that those limits are joint-space or task-space limits that cannot be trespassed. For instance, joint limits are a physical type of hard limit. The controller cannot command the robot to move over its joint limits.

When performing kinematic control, the decision variable are the joint velocities and not the joint positions. Therefore, we need to write the joint limits as a linear inequality that depends on the joint velocities.

We have used in this paper quadratic programming (QP), which is a linearly-constrained quadratic optimization problem, in the form (20)

$$\begin{aligned} u \in\ & \underset{\dot{\boldsymbol{\theta}}}{\arg\min}\, f(\dot{\boldsymbol{\theta}}) \\ & \text{subject to } W\dot{\boldsymbol{\theta}} \leq \mathbf{w} \end{aligned} \tag{20}$$

Where $W \in \mathbb{R}^{\mathbf{n} \times \mathbf{p}}$ and $\mathbf{w} \in \mathbb{R}^{\mathbf{p}}$represent $\mathbf{p}$ linear constraints related to the n degrees-of-freedom of the robot. So, we have to find decision variable $\dot{\boldsymbol{\theta}}$ that minimize objective function $f(\dot{\boldsymbol{\theta}})$, subject to $W\dot{\boldsymbol{\theta}}$ less or equal to $\mathbf{w}$, and return as u.

Suppose that the lower joint limits are $\boldsymbol{\theta}^- \in \mathbb{R}^{\mathbf{n}}$ and the upper joint limits are $\boldsymbol{\theta}^+ \in \mathbb{R}^{\mathbf{n}}$. The most common way of implementing joint limits using a QP is as follows [10]

$$\begin{aligned} u \in\ & \underset{\dot{\boldsymbol{\theta}}}{\arg\min} \left\| J\,\dot{\boldsymbol{\theta}} + \operatorname{vec}_8(\underline{x} - \underline{x}_d) \right\|^2 + \lambda^2 \left\|\dot{\boldsymbol{\theta}}\right\|^2 \\ & \text{subject to } W\dot{\boldsymbol{\theta}} \leq \mathbf{w} \end{aligned} \tag{21}$$

Where

$W = \begin{bmatrix} -I \\ +I \end{bmatrix}$, I = Identity matrix, and $\mathbf{w} = \begin{bmatrix} -(\boldsymbol{\theta}^- - \boldsymbol{\theta}) \\ +(\boldsymbol{\theta}^+ - \boldsymbol{\theta}) \end{bmatrix}$

The main idea behind this constraint is to constrain the velocity of the joint in the direction of the joint limit.

5. Results

5.1 Constrained Optimization using Dual Quaternion for Single Pose (pitch=90°)

We will solve the inverse kinematics of Pose target has gimbal lock(pitch=90°) by Constrained Optimization

Table2. Pose Target

	X (cm)	Y (cm)	Z (cm)	Roll (deg)	Pitch (deg)	Yaw (deg)	
Pose Target = [	15	5	3	15	90	75	]

Where the limit of the joints:

Theta Limit min= [-179°,-32°, -105°, -5°, -25°, -68°]

and Theta Limit max= [179°, 32°, 45°, 100°, 120°, 68°]

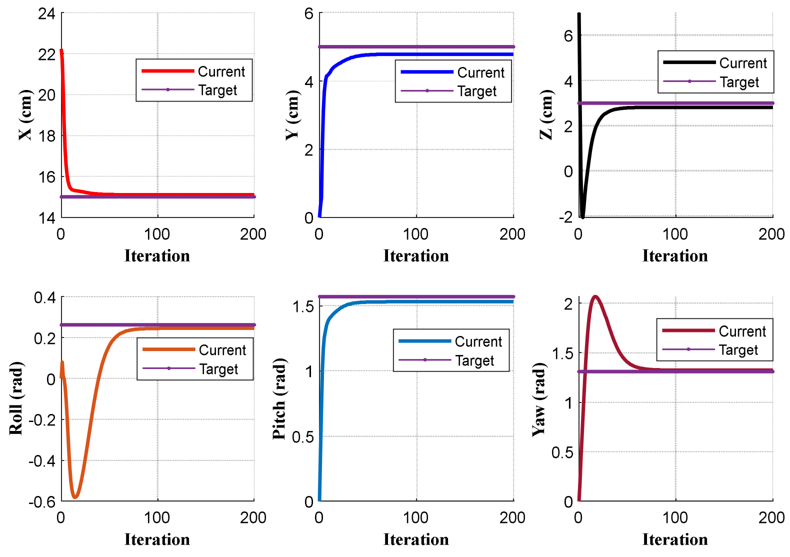

Fig 2. Pose target and Pose Actual

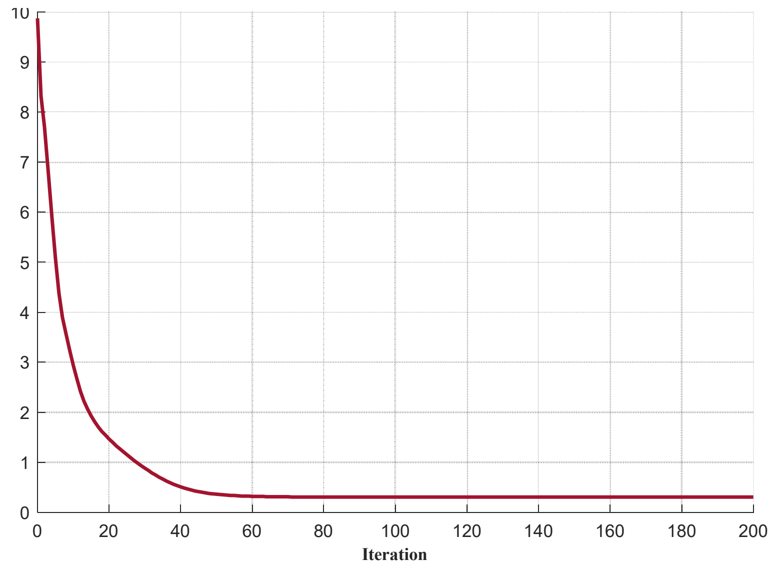

Fig 3. Norm Error for Pose Target in every iteration

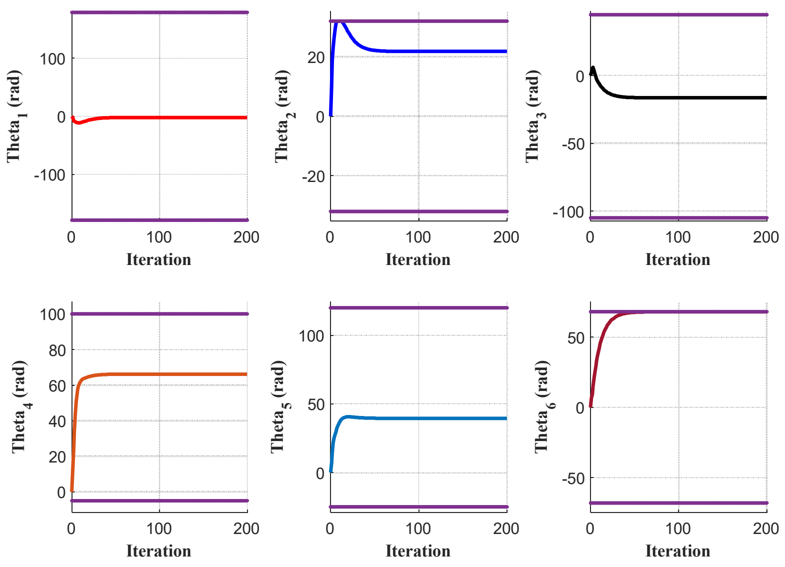

Fig 4. Theta of all joints of manipulator

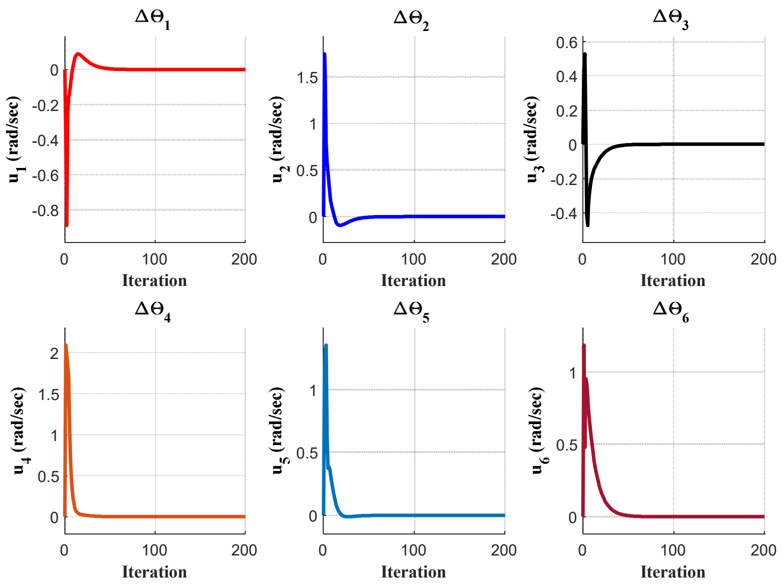

Fig 5. Speed in every iteration

5.2 Constrained Optimization using Dual Quaternion for Tracking Predefined Trajectory

We will solve the inverse kinematics for defined trajectory by Constrained Optimization, the Trajectory path which we defined contains 40 poses.

first Half Circle with R=7 and contain 20 poses.

Table3. First part of path trajectory

	X (cm)	Y (cm)	Z (cm)	Roll (deg)	Pitch (deg)	Yaw (deg)	
Pose Target 01 = [	17	3.5	6	15	30	45	]
Pose Target 20 = [	17	-3.5	6	15	30	45	]

Then Line between two points contain 20 poses

Table4. Second part of path trajectory

	X (cm)	Y (cm)	Z (cm)	Roll (deg)	Pitch (deg)	Yaw (deg)	
Pose Target 21 = [	17	-3.5	6	15	30	45	]
Pose Target 40 = [	17	3.5	6	15	30	45	]

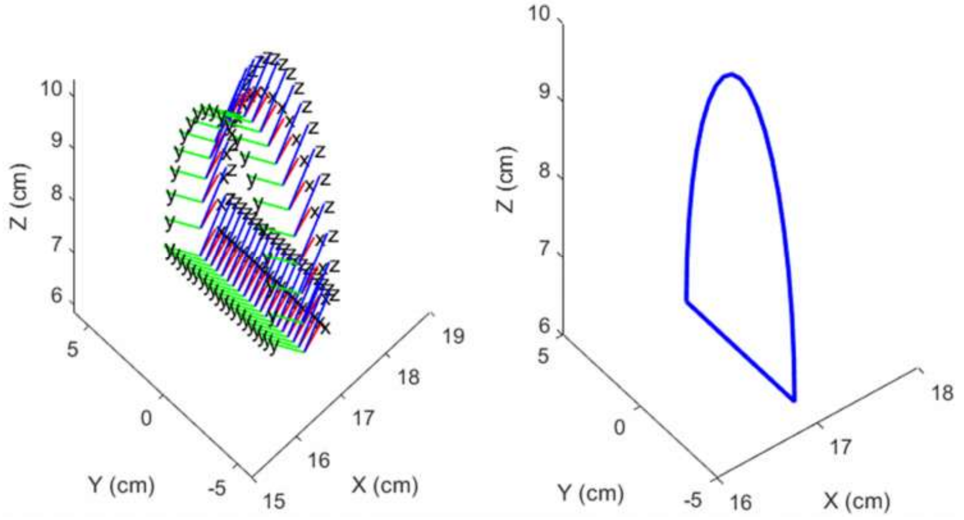

Fig 6. Poses of path trajectory

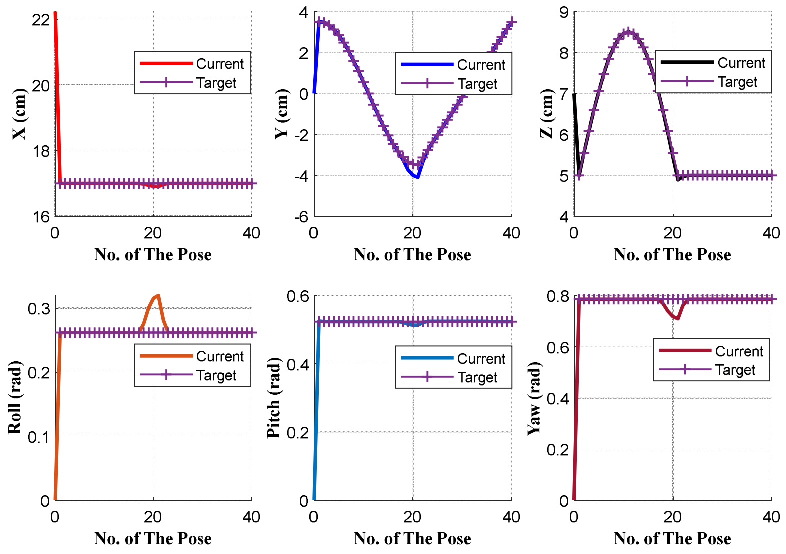

Fig 7. Poses Trajectory

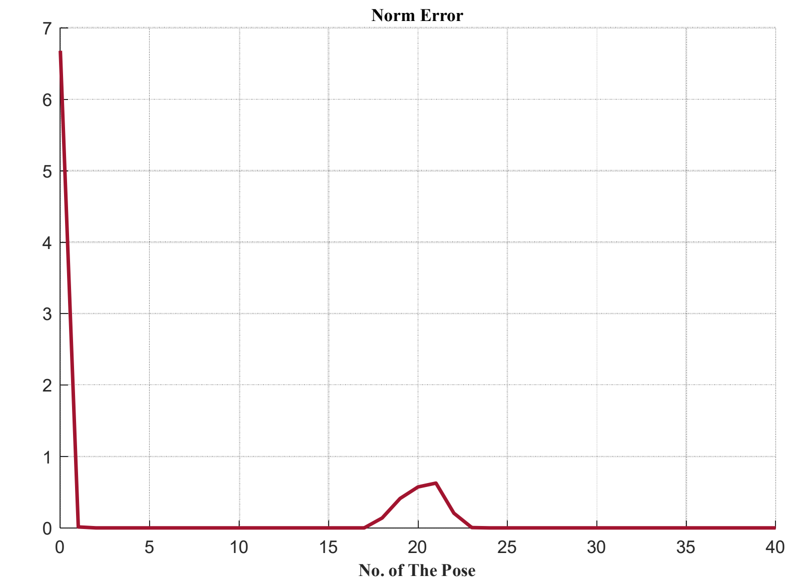

Fig 8. Norm Error for Poses

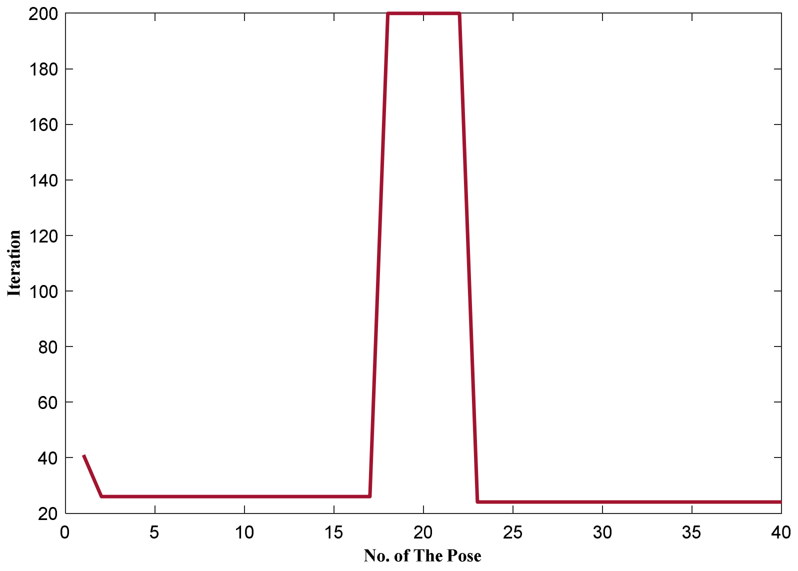

Fig 9. Number of iterations in every Pose

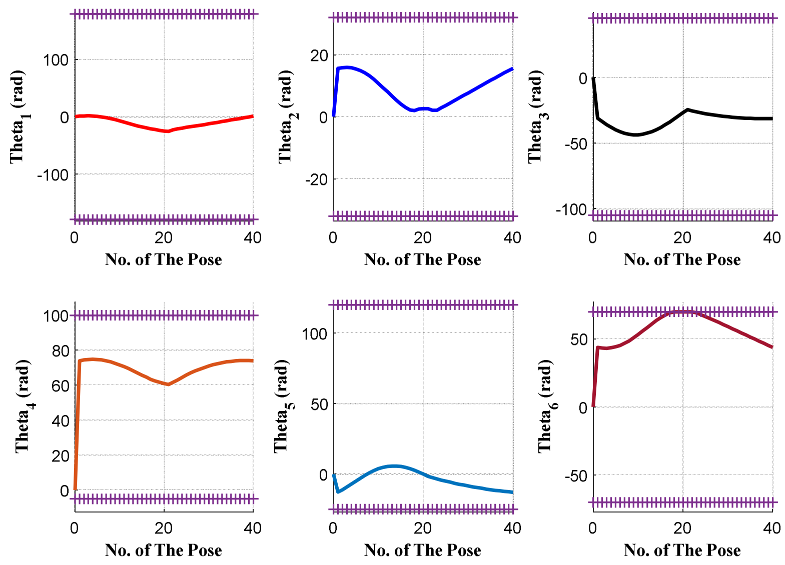

Fig 10. Theta of all joints of manipulator in every Pose

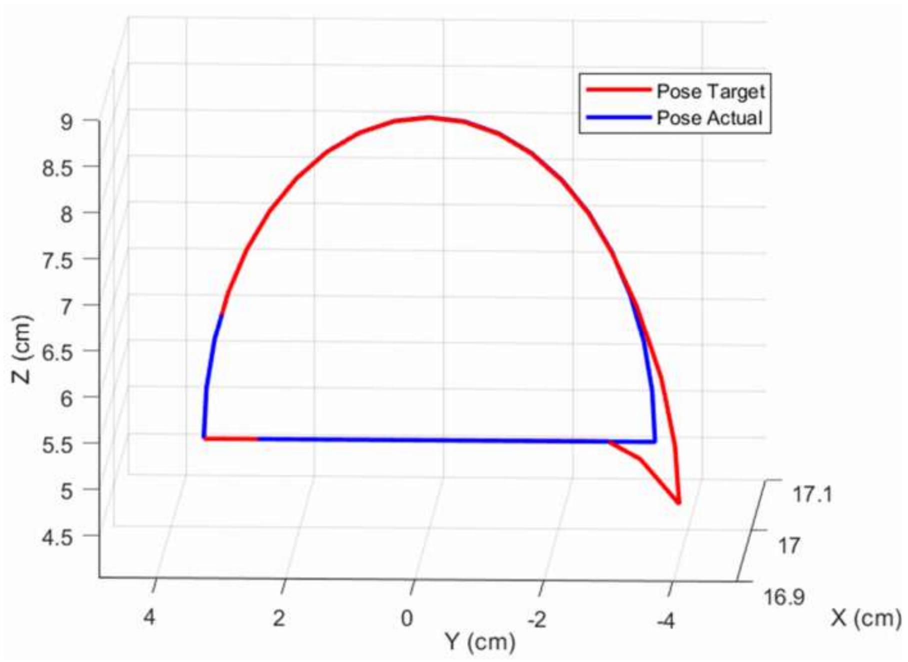

Fig 11. Distance between Actual Poses and Target Poses

6. Conclusions

Simulations show that the combination of the Unit Dual Quartenion and Constrained-Quadratic Optimisation provides acceptable solutions for the Inverse kinematic problem when actuation limitations are imposed. The resulting paths are smooth and achieve "best possible" trajectories even for target parths that lie beyond the reachable space. However, the approach requires significant computational effort.
This sets the basis for further work, in the context of Doctoral research, to exploit the approach off-line in order to tabulate extensive IK datasets. The results will be used to train Machine Learning systems that can reproduce the solution in a computationally efficient way and achieve the rapid and accurate response required in practical robotic applications.

References

[1] J. J. Craig, *Adaptive Control of Mechanical Manipulators*. Addison Wesley Publishing Company, 1988.

[2] Z. Zhao, T. Wang, and D. Wang, "Inverse kinematic analysis of the general 6R serial manipulators based on unit dual quaternion and Dixon resultant," in *2017 Chinese Automation Congress (CAC)*, pp. 2646–2650. doi: https://doi.org/10.1109/CAC.2017.8243223.

[3] H. Wang, Z. Zhou, X. Zhong, and Q. Chen, "Singular Configuration Analysis and Singularity Avoidance with Application in an Intelligent Robotic Manipulator," *Sensors*, vol. 22, no. 3, 2022, doi: https://doi.org/10.3390/s22031239.

[4] Silva and B. V. Adorno, "Wholebody control of a mobile manipulator using feedback linearization based on dual quaternions," in *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*, pp. 293–298. doi: https://doi.org/10.1109/LARSSBR.2016.56.

[5] H. Pham, V. Perdereau, B. V. Adorno, and P. Fraisse, "Position and orientation control of robot manipulators using dual quaternion feedback," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 658–663. doi: https://doi.org/10.1109/IROS.2010.5651097.

[6] K. Ayusawa, W. Suleiman, and E. Yoshida, "Predictive Inverse Kinematics: optimizing Future Trajectory through Implicit Time Integration and Future Jacobian Estimation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 566–573. doi: https://doi.org/10.1109/IROS40897.2019.8968110.

[7] K. Dufour, J. OcampoJimenez, and W. Suleiman, "Visual–spatial attention as a comfort measure in human–robot collaborative tasks," *Robotics and Autonomous Systems*, vol. 133, p. 103626, 2020, doi: https://doi.org/10.1016/j.robot.2020.103626.

[8] C. R. Rocha, C. P. Tonetto, and A. Dias, "A comparison between the Denavit–Hartenberg and the screwbased methods used in kinematic modeling of robot manipulators," *Conference papers of Flexible Automation and Intelligent Manufacturing*, vol. 27, no. 4, pp. 723–728, 2011, doi: https://doi.org/10.1016/j.rcim.2010.12.009.

[9] M. Fox, "Quaternions and Rotation Sequences by Jack B. Kuipers," *The Mathematical Gazette*, vol. 90, pp. 173–175, Jan. 2006, doi: https://doi.org/10.2307/3621452.

[10] J. M. Selig, *Geometric Fundamentals of Robotics*. Springer, 2005. doi: https://doi.org/10.1007/b138859.

[11] B. V. Adorno, “Two-arm Manipulation: From Manipulators to Enhanced Human-Robot Collaboration,” Automatic. Université Montpellier II - Sciences et Techniques du Languedoc, 2011.

[12] A. Escande, N. Mansard, and P. Wieber, “Hierarchical quadratic programming: Fast online humanoidrobot motion generation,” *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014, doi: https://doi.org/10.1177/0278364914521306.

[13] J. Vidakovic, M. Lazarević, V. Kvrgic, Z. Dancuo, and G. Ferenc, “Advanced Quaternion Forward Kinematics Algorithm Including Overview of Different Methods for Robot Kinematics,” *FME Transactions*, vol. 42, Jan. 2014, doi: https://doi.org/10.5937/fmet1403189v.