

Author-accepted manuscript

Published citation

Z. Wu, Z. Wang, H. Zhang, and H. Yan, "Quaternion-Based Optimal Interpolation of Similarity Transformations for Multi-Agent Formation," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5783–5790, Jun. 2024. doi: [10.1109/LRA.2024.3396662](https://doi.org/10.1109/LRA.2024.3396662)

This is the author's accepted manuscript. Please cite the published IEEE article using the DOI above. The manuscript follows this notice page.

Quaternion-based Optimal Interpolation of Similarity Transformation for Multi-agent Formation

Zeming Wu¹, Zhuping Wang¹, Hao Zhang¹, Huaicheng Yan²

Abstract—This paper investigates the optimal motion interpolation problem for multi-agent formation. Specifically, this paper aims to generate trajectories of similarity transformations that minimize a series of metrics, including but not limited to traveled distance, kinetic energy consumption, and smoothness. A novel quaternion-based representation is proposed for similarity transformations to circumvent the complex map between Lie group elements and Lie algebra elements. With this quaternion-based representation, the time derivatives of agents can be calculated with finite algebraic operations, which is essential for motion interpolation. The optimal motion interpolation problem is formulated as a variational problem. We derive the necessary conditions for two points and multiple points optimal interpolations. Furthermore, a polynomial approximation method that preserves the structure of the optimal solution is proposed to improve the computational efficiency. Finally, the proposed method is illustrated with examples of ellipsoid formation. Comparisons are also conducted to show the optimality of the proposed method.

Index Terms—Multi-agent Formation, Optimal Interpolation, Similarity Transformation, Motion Generation.

I. INTRODUCTION

Formation has been a popular topic in the robotics and control community over the past years. In recent years, the rapid development of sensor and communication technology enables the application of multi-agent coordination in complex environments, such as geographical exploration [1], search and rescue [2], cooperative reconnaissance [3] and monitoring [4]. Since moving in a formation when executing a mission can be more effective in the above scenarios, significant efforts have been devoted to the formation navigation of multi-agent systems [5–10]. For example, in [5], a bearing-based formation tracking controller is designed following the leader-follower framework. By designing the motion of leaders, the swarm achieves scaling maneuver to pass through narrow passages. In [6, 10], the swarm are expected to keep in a prescribed geometry pattern during the whole navigation, and avoid obstacles by using scaling, rotational and translational maneuvers. In all these paradigms,

the swarm should move in a pre-defined geometry pattern. That is, the motion of the swarm should be modeled as similarity transformations, which includes the gross scaling, rotation and translation motion of the geometry pattern. Consequently, we are primarily interested in motion interpolation for the similarity transformations in this paper.

In literature, the problem of motion interpolation has been studied extensively, especially the motion interpolations in vector space. For example, in [11], B-spline are used to interpolate multiple waypoints to generate smooth trajectories for UAVs. In [12], clothoids are used to generate feasible and curvature continuous paths for AGV. However, the motion interpolation methods in vector space can not be directly apply to the interpolations of similarity transformations. When using the Lie group $\text{SIM}(3)$ expression, the similarity transformations is actually a manifolds on $\mathbb{R}^{4 \times 4}$. That is, the linear combinations of elements of $\text{SIM}(3)$ do not necessary belong to $\text{SIM}(3)$. Hence, directly interpolating $\text{SIM}(3)$ with the methods in vector space will lead to infeasible elements.

There are two main paradigms of motion interpolations on manifolds, one of which is the projection-based methods. The projection-based methods directly interpolates the elements in euclidean spaces, and then enforce the manifold constraints by projecting it back to the corresponding manifold [13, 14]. In [13], an algorithm is proposed for generating smooth trajectories for a moving rigid body with specified boundary. The method involves two steps, the first step: 1) generating an optimal trajectories in $\mathbb{R}^{4 \times 4}$ and then 2) projecting the optimal trajectories back to $\text{SE}(3)$. The projection-based method is computational efficient, but the optimality is lost after projection. Another paradigm is to re-parameterize the manifolds with vector space, and interpolate in vector space. For examples, it is well known that every Lie algebra element $\mathfrak{g}$ is associate with a Lie group element G by the corresponding exponential map. Consequently, the motion interpolation on Lie group (generally not a vector space) can be achieved in two steps: 1) interpolating on Lie algebra (vector space), and 2) mapping the Lie algebra trajectory back to Lie group using the exponential map. The motion interpolation on Lie group $\text{SE}(3)$ and $\text{SIM}(3)$ based on this paradigm can be found in [15] and [16], respectively. However, the exponential map and the logarithm map can not be expressed by finite algebraic operations in general. This property makes it challenging to calculate the time derivative, which is essential in motion planning.

The trajectories interpolating multiple points are generally not unique, additional properties are thus required. In prac-

This work is supported by in part by the National Natural Science Foundation of China (61922063, 62273255, 62150026), in part by the Shanghai International Science and Technology Cooperation Project (21550760900, 22510712000), in part by the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0100) and in part by the Fundamental Research Funds for the Central Universities.

¹Z. Wu, Z. Wang and H. Zhang are with the Department of Control Science and Engineering, Tongji University, Shanghai, China, e-mail: {zemingwu, elewzp, zhang_hao}@tongji.edu.cn.

²H. Yan is with the Key Laboratory of Advanced Control and Optimization for Chemical Processes of Ministry of Education, School of Information Science and Engineering, East China University of Science and Technology, Shanghai, 200237, P. R. China, e-mail: hcyang@ecust.edu.cn

tice, selecting the optimal trajectory under certain metrics is highly desirable. For example, in vector spaces, the linear interpolation is the trajectories that minimize the distance. The 7-th order polynomials correspond to the minimum-snap spline of quadrotors systems [17]. Only a few works focus on optimal motion interpolations on manifolds. In [18], a cost-functional based on covariant derivatives is proposed for 3D rigid body motion, and the calculus of variational (CoV) approach is used to obtain the necessary conditions for optimality. In [19], a semi-definite programming (SDP) method is proposed to generate minimum-jerk trajectories for rotational motions, but the proposed method can not handle the terminal conditions of rotation. In summary, the existing works mainly focus on the motion interpolation of rigid body motion. The optimal interpolation problem of similarity transformations is still an open problem. Moreover, most existing works [15, 18, 19] only investigate the optimal interpolation between two given states. The conditions for optimal interpolation of multiple given states must be investigated.

Motivated by the above observations, the optimal interpolation of multiple similarity transformations is studied in this paper. Compared with the previous works, the main contributions of this paper are as follows:

- 1) A novel quaternion-based representation for similarity transformation is proposed, whose time derivatives can be calculated with finite algebraic operations.
- 2) The necessary conditions of optimal similarity transformations interpolation are derived for both two points interpolation and multiple points interpolation.
- 3) A polynomial interpolation method is proposed to approximate the optimal trajectories efficiently.

The remainder of this paper is structure as follows. In Section II, the quaternion-based representation for similarity transformations is proposed, and the problem of optimal interpolation is formulated. Section III derives the necessary conditions of optimal interpolation for two and multiple points using the calculus of variations method. A polynomial approximation method for the optimal trajectories is proposed in Section IV. The effectiveness of the proposed method is validated by simulations and comparisons in Section V. In Section VI, we conclude the paper.

II. PRELIMINARIES AND PROBLEM FORMULATION

A. Notation

In what follows, $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}^n$ denotes the set of n -dimension column vector over $\mathbb{R}$, and $\mathbb{R}^{m \times n}$ denotes the set of m -by- n matrices over $\mathbb{R}$. Note that, vectors are represented by bold lowercase letters, and matrices are represented by bold uppercase letters. For example, c is a scalar, $\mathbf{c}$ is a vector and $\mathbf{C}$ is a matrix.

τ denotes the time. And the time derivative of a variable is denoted by the superscript with parenthesis $(\cdot)$, for example, $\mathbf{c}^{(n)} = \frac{d^n \mathbf{c}}{d\tau^n}$ is the n -th order time derivative of vector $\mathbf{c}$. For simplicity, given a vector $\mathbf{c} \in \mathbb{R}^d$, we denote the superscript

with bracket $[\cdot]$ as:

$$\mathbf{c}^{[n]} = (\mathbf{c}^T, \mathbf{c}^{(1)T}, \dots, \mathbf{c}^{(n)T})^T \in \mathbb{R}^{d(n+1)}. \quad (1)$$

$\mathbf{SO}(3)$ denotes the 3-dimension rotation group defined as $\mathbf{SO}(3) = \{\mathbf{R} | \mathbf{R} \in \mathbb{R}^{3 \times 3}, \mathbf{R}^T \mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1\}$. $\|\cdot\|_2$ denotes the Euclidean norm of vector, $\|\cdot\|_F$ denotes the Forbenius norm of matrices.

B. Quaternion-based Similarity Transformations

The similarity transformation is defined as the map form $\mathbb{R}^3$ to $\mathbb{R}^3$:

$$\mathbf{p}(\mathbf{c}) = \alpha \mathbf{R} \mathbf{c} + \mathbf{t}, \quad (2)$$

where $s \in \mathbb{R}^+$ is the scaling factor, $\mathbf{R} \in \mathbf{SO}(3)$ is the rotation matrix and $\mathbf{t} \in \mathbb{R}^3$ is the translation vector. $\mathbf{c} = (c_x, c_y, c_z)^T \in \mathbb{R}^3$ is the position in the local frame and $\mathbf{p} = (p_x, p_y, p_z)^T \in \mathbb{R}^3$ is the position in the inertial frame. Various method has been used to describe similarity transformations in the existing works, which includes the Lie group $\mathbf{SIM}(3)$ method [16], the corresponding Lie algebra $\mathfrak{sim}(3)$ method [16] and the quaternion-based method [20].

The Lie group of 3D similarity transformations is the following set in $\mathbb{R}^{4 \times 4}$:

$$\mathbf{SIM}(3) = \left\{ \begin{pmatrix} \alpha \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \mid \alpha \in \mathbb{R}^+, \mathbf{R} \in \mathbf{SO}(3), \mathbf{t} \in \mathbb{R}^3 \right\}. \quad (3)$$

The $\mathbf{SIM}(3)$ is a nontrivial manifold on $\mathbb{R}^{4 \times 4}$, which is not a vector space, the linear combinations of elements of $\mathbf{SIM}(3)$ do not necessary belong to $\mathbf{SIM}(3)$. As a result, the Lie group $\mathbf{SIM}(3)$ is unsuitable for interpolation.

One way to circumvent this difficulty is to interpolate with Lie algebra $\mathfrak{sim}(3)$, which is a vector space. However, calculating the time derivative of similarity transformations with Lie algebra $\mathfrak{sim}(3)$ involves the calculation of right jacobian of the exponential coordinates [19], which makes it unsuitable for optimal interpolation.

A non-unit quaternion $\mathbf{q} = (q_w, q_x, q_y, q_z)^T \in \mathbb{R}^4$ together with a translation vector $\mathbf{t} = (t_x, t_y, t_z)^T \in \mathbb{R}^3$ also define a similarity transformation. Denote $\mathbf{q}_u$ as the unit direction vector of quaternion $\mathbf{q}$, and denote $\beta = \|\mathbf{q}\|$ as the norm of quaternion, such that $\mathbf{q} = \beta \mathbf{q}_u$. The similarity transformation described by $\mathbf{q}$ and $\mathbf{t}$ is:

$$\begin{aligned} \mathbf{p} &= \mathbf{q} \circ \mathbf{c} \circ \bar{\mathbf{q}} + \mathbf{t} \\ &= \beta^2 \cdot \mathbf{q}_u \circ \mathbf{c} \circ \bar{\mathbf{q}}_u + \mathbf{t}, \end{aligned} \quad (4)$$

where $\bar{\mathbf{q}}$ denotes the conjugate of $\mathbf{q}$, and $\circ$ denotes the product of quaternion. Noting that for a unit quaternion $\mathbf{q}_u$, the term $\mathbf{q}_u \circ \mathbf{c} \circ \bar{\mathbf{q}}_u$ corresponds to the rotation term $\mathbf{R} \mathbf{c}$ in (2). And the β^2 corresponds to the scaling term s in (2).

The product of quaternion is a linear operation, which is used twice in (4). As a result, each elements of $\mathbf{p}$ is a quadratic form of the quaternion $\mathbf{q}$. More precisely:

$$\begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} = \begin{pmatrix} 2\mathbf{q}^T \mathbf{H}_x \mathbf{q} \\ 2\mathbf{q}^T \mathbf{H}_y \mathbf{q} \\ 2\mathbf{q}^T \mathbf{H}_z \mathbf{q} \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}, \quad (5)$$

where matrix $\mathbf{H}_x, \mathbf{H}_y, \mathbf{H}_z \in \mathbb{R}^{4 \times 4}$ are matrix depend on $\mathbf{c}$, whose detailed expression are given in Appendix I.

The quaternion-based representation $(\mathbf{q}, \mathbf{t})$ for similarity transformations enjoys the following advantages. Firstly, $(\mathbf{q}, \mathbf{t}) \in \mathbb{R}^7$ is a vector space, which can be easily interpolated. Secondly, the quaternion-based representation is not overdetermined, which only has 7 parameters, while the $(\alpha, \mathbf{R}, \mathbf{t})$ representation has 13 parameters. Thirdly, there are no singularities in the representation except the one at zero, which would mean that the scale factor is zero and is thus not valid for similarity transformations. Finally, the time derivative can also be easily calculated according to product rule and binomial theorem. Take p_x as an example:

$$p_x^{(n)} = \frac{d^n p_x}{d\tau^n} = \mathbf{t}^{(n)} + 2 \sum_{i=0}^n \binom{n}{i} \mathbf{q}^{(n-i)T} \mathbf{H}_x \mathbf{q}^{(i)}, \quad (6)$$

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}. \quad (7)$$

As a result, the quaternion-based method will be used in this paper for interpolation. For simplicity, $\mathbf{s} = (\mathbf{q}, \mathbf{t}) \in \mathbb{R}^7$ will be used to denote a quaternion-based similarity transformations.

C. The Problem of Optimal Interpolation

Considering a swarm that consists of N robots, the geometry pattern of a swarm is defined as $\mathbf{C} = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_N) \in \mathbb{R}^{3 \times N}$, where $\mathbf{c}_i \in \mathbb{R}^3$ is the position of the i -th agent in local frame. A swarm is said to be moving in *formation* if the swarm preserves the geometry pattern, i.e. moving with only similarity transformations. More precisely, the position of the i -th agent in the inertial frame can be described as:

$$\mathbf{p}_i(\tau) = \mathbf{q}(\tau) \circ \mathbf{c}_i \circ \mathbf{q}(\tau) + \mathbf{t}(\tau), \quad \forall i = 1, 2, \dots, N. \quad (8)$$

In some applications, e.g., formation navigation in cluttered environment [10], the swarm are desired to navigation through a sequence of similarity transformations $\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_M$ at time knots $\tau_0, \tau_1, \dots, \tau_M$. The interpolation problem is to find a curve $\mathbf{s} : [\tau_0, \tau_M] \mapsto \mathbb{R}^7$ that satisfies $\mathbf{s}(\tau_i) = \mathbf{s}_i, \forall i = 0, \dots, M$. Moreover, it is highly desirable in practice that the curve $\mathbf{s}(\tau)$ is optimal under some performance index. For example, the swarm should navigate through the prescribed similarity transformations with minimum traveled distance, minimum kinetic energy, etc. Given a scalar function $V(\mathbf{s}, \tau) \in \mathbb{R}$, the general optimal interpolation problem can be formulated as follows:

$$\min_{\mathbf{s}(\tau)} J(\mathbf{s}(\tau)) = \int_{\tau_0}^{\tau_M} V(\mathbf{s}(\tau), \tau) d\tau \quad (9a)$$

$$s.t. \quad \mathbf{s}^{[n-1]}(\tau_0) = \mathbf{s}_0^{[n-1]}, \quad \mathbf{s}^{[n-1]}(\tau_M) = \mathbf{s}_M^{[n-1]}, \quad (9b)$$

$$\mathbf{s}^{[d_m-1]}(\tau_i) = \mathbf{s}_i^{[d_m-1]}, \quad m = 1, 2, \dots, M-1. \quad (9c)$$

where $\mathbf{s}_0^{[n-1]}, \mathbf{s}_M^{[n-1]} \in \mathbb{R}^{7n}$ are the constant initial state and terminal state up to $(n-1)$ -th order time derivative that should be satisfied, respectively. $\mathbf{s}_i^{[d_m-1]} \in \mathbb{R}^{7d_m}$ is the constant intermediate condition up to (d_m-1) -order time derivative that should be satisfied at time knot t_i , and note

that $d_m \leq n$. In this paper, the scalar functions with the following form will be studied:

$$V(\mathbf{s}, \tau) = \sum_{i=1}^N \|\mathbf{p}_i^{(n)}\|_2^2, \quad (10)$$

where $n \in \mathbb{Z}^+$ is a selectable parameter. When $n = 1$, $V(\mathbf{s}, \tau)$ is the summation of the squared velocity of each agent, and the physical meaning of (9a) is to minimize the total traveled distance of swarm. When $n = 2$, $V(\mathbf{s}, \tau)$ is the summation of the squared acceleration of each agent, the physical meaning of (9a) corresponds to minimizing the total spend kinetic energy [14]. There are also some examples [10, 14, 17] that minimize the jerk ($n = 3$) or snap ($n = 4$) when generating trajectories for quadrotors, and the corresponding physical meaning of $V(\mathbf{s}, \tau)$ is a measurement of smoothness.

By selecting the geometry centre of the formation as the origin of the local frame, the optimal interpolation problem can be separated into two subproblems, i.e., the optimal interpolation of quaternion and the optimal interpolation of translation vector. Expanding (10) with (8):

$$\begin{aligned} V(\mathbf{s}, \tau) &= \sum_{i=1}^N \|\mathbf{p}_i^{(n)}\|_2^2 \\ &= \sum_{i=1}^N \|(\mathbf{q} \circ \mathbf{c}_i \circ \mathbf{q})^{(n)} + \mathbf{t}^{(n)}\|_2^2 \\ &= \sum_{i=1}^N \|(\mathbf{q} \circ \mathbf{c}_i \circ \mathbf{q})^{(n)}\|_2^2 + \sum_{i=1}^N \|\mathbf{t}^{(n)}\|_2^2 \\ &\quad + \sum_{i=1}^N 2\mathbf{t}^{(n)T} (\mathbf{q} \circ \mathbf{c}_i \circ \mathbf{q})^{(n)}, \end{aligned} \quad (11)$$

Noting that the product of quaternion is a linear operation, and the origin of the local frame is the same as the geometry centre of the formation, that is, $\sum_{i=1}^N \mathbf{c}_i = \mathbf{0}$. We have:

$$\begin{aligned} &\sum_{i=1}^N 2\mathbf{t}^{(n)T} (\mathbf{q} \circ \mathbf{c}_i \circ \mathbf{q})^{(n)} \\ &= 2\mathbf{t}^{(n)T} (\mathbf{q} \circ (\sum_{i=1}^N \mathbf{c}_i) \circ \mathbf{q})^{(n)} \\ &= 2\mathbf{t}^{(n)T} (\mathbf{q} \circ \mathbf{0} \circ \mathbf{q})^{(n)} = 0. \end{aligned}$$

As a result, the optimal interpolation problem can be separated as the following two subproblems:

Definition 1 ($\mathcal{P}_1$, Optimal Translation Interpolation).

$$\begin{aligned} \min_{\mathbf{t}(\tau)} \quad & J_1(\mathbf{t}(\tau)) = \int_{\tau_0}^{\tau_M} N \|\mathbf{t}^{(n)}\|_2^2 d\tau \\ s.t. \quad & \mathbf{t}^{[n-1]}(\tau_0) = \mathbf{t}_0^{[n-1]}, \quad \mathbf{t}^{[n-1]}(\tau_M) = \mathbf{t}_M^{[n-1]}, \\ & \mathbf{t}^{[d_m-1]}(\tau_m) = \mathbf{t}_m^{[d_m-1]}, \quad m = 1, 2, \dots, M-1. \end{aligned}$$

where $\mathbf{t}_0^{[n-1]}$ and $\mathbf{t}_M^{[n-1]}$ are boundary conditions, $\mathbf{t}_m^{[d_m-1]}, \tau_m, m = 1, \dots, M$ are intermediate states and time knots that should be given.

The optimal translation interpolation problem $\mathcal{P}_1$ is actually the *Multi-Stage Minimum Control Effort Problem* of a n -order integrator system, which has been studied in [21].

Another subproblem corresponds to interpolate the scaling and rotation optimally:

Definition 2 ($\mathcal{P}_2$, Optimal Quaternion Interpolation).

$$\min_{\mathbf{q}(\tau)} J_2(\mathbf{q}(\tau)) = \int_{\tau_0}^{\tau_M} \sum_{i=0}^N \|(\mathbf{q} \circ \mathbf{c}_i \circ \mathbf{q})^{(n)}\|_2^2 d\tau \quad (12a)$$

$$s.t. \quad \mathbf{q}^{[n-1]}(\tau_0) = \mathbf{q}_0^{[n-1]}, \quad \mathbf{q}^{[n-1]}(\tau_M) = \mathbf{q}_M^{[n-1]}, \quad (12b)$$

$$\mathbf{q}^{[d_m-1]}(\tau_m) = \mathbf{q}_m^{[d_m-1]}, \quad i = 1, 2, \dots, M-1. \quad (12c)$$

where $\mathbf{q}_0^{[n-1]}$ and $\mathbf{q}_M^{[n-1]}$ are boundary conditions, $\mathbf{q}_m^{[d_m-1]}$, $\tau_m, m = 1, \dots, M$ are intermediate quaternions and time knots that should be given.

III. OPTIMAL INTERPOLATION OF QUATERNION

In this section, the optimal quaternion interpolation problem $\mathcal{P}_2$ will be discussed. Denote $V_i(\mathbf{q})$ as:

$$V_i(\mathbf{q}) = \|(\mathbf{q} \circ \mathbf{c}_i \circ \mathbf{q})^{(n)}\|_2^2 = \sum_{j \in \{x, y, z\}} \|2(\mathbf{q}^T \mathbf{H}_{ij} \mathbf{q})^{(n)}\|_2^2 \quad (13)$$

And denote $V_{ij}(\mathbf{q})$ as:

$$V_{ij} = \|2(\mathbf{q}^T \mathbf{H}_{ij} \mathbf{q})^{(n)}\|_2^2. \quad (14)$$

Then J_2 becomes:

$$J_2 = \sum_{i=0}^N \sum_{j \in \{x, y, z\}} \int_{\tau_0}^{\tau_M} V_{ij}(\mathbf{q}) d\tau. \quad (15)$$

A. Optimal Interpolation of Two Quaternions

We solve $\mathcal{P}_2$ with the calculus of variations method. For simplicity, the intermediate quaternion constraints (12c) will be removed temporarily. Denote $\mathbf{q}^*$ as the optimal solution of $\mathcal{P}_2$. And define the admissible trajectories as $\mathbf{q} = \mathbf{q}^* + \delta\mathbf{q}$, where $\delta\mathbf{q}$ is the variation of $\mathbf{q}^*$. To satisfies (12b), the following conditions should be satisfied:

$$\mathbf{q}^{[n-1]}(\tau_0) = \mathbf{q}_0^{[n-1]}, \quad \mathbf{q}^{[n-1]}(\tau_M) = \mathbf{q}_M^{[n-1]}, \quad (16)$$

$$\delta\mathbf{q}^{[n-1]}(\tau_0) = \mathbf{0}, \quad \delta\mathbf{q}^{[n-1]}(\tau_M) = \mathbf{0}. \quad (17)$$

The first variations of J_2 is:

$$\begin{aligned} \delta J_2 &= \sum_{i=0}^N \sum_{j \in \{x, y, z\}} \int_{\tau_0}^{\tau_M} V_{ij}(\mathbf{q}^* + \delta\mathbf{q}) - V_{ij}(\mathbf{q}^*) d\tau \\ &\approx \sum_{i=0}^N \sum_{j \in \{x, y, z\}} \sum_{k=0}^n \int_{\tau_0}^{\tau_M} \left(\frac{\partial V_{ij}}{\partial \mathbf{q}^{(k)}} \right)^T \delta\mathbf{q}^{(k)} d\tau. \end{aligned} \quad (18)$$

By using the integration by parts formula, we have:

$$\begin{aligned} &\int_{\tau_0}^{\tau_M} \left(\frac{\partial V_{ij}}{\partial \mathbf{q}^{(k)}} \right)^T \delta\mathbf{q}^{(k)} d\tau \\ &= \frac{\partial V_{ij}}{\partial \mathbf{q}^{(k-1)}} \delta\mathbf{q}^{(k)} \Big|_{\tau_0}^{\tau_M} - \int_{\tau_0}^{\tau_M} \left(\frac{d}{d\tau} \frac{\partial V_{ij}}{\partial \mathbf{q}^{(k)}} \right)^T \delta\mathbf{q}^{(k)} d\tau \end{aligned} \quad (19)$$

Using integration by parts for $k-1$ times, we have:

$$\begin{aligned} &\int_{\tau_0}^{\tau_M} \left(\frac{\partial V_{ij}}{\partial \mathbf{q}^{(k)}} \right)^T \delta\mathbf{q}^{(k)} d\tau \\ &= \sum_{l=0}^{k-1} (-1)^l \left(\frac{d^l}{d\tau^l} \frac{\partial V_{ij}}{\partial \mathbf{q}^{(k-l)}} \right)^T \delta\mathbf{q}^{(k-l-1)} \Big|_{\tau_0}^{\tau_M} \\ &\quad + \int_{\tau_0}^{\tau_M} (-1)^k \left(\frac{d^k}{d\tau^k} \frac{\partial V_{ij}}{\partial \mathbf{q}^{(k)}} \right)^T \delta\mathbf{q} d\tau \end{aligned} \quad (20)$$

From (17), the first term of (20) is zero, thus:

$$\int_{\tau_0}^{\tau_M} \left(\frac{\partial V_{ij}}{\partial \mathbf{q}^{(k)}} \right)^T \delta\mathbf{q}^{(k)} d\tau = \int_{\tau_0}^{\tau_M} (-1)^k \left(\frac{d^k}{d\tau^k} \frac{\partial V_{ij}}{\partial \mathbf{q}^{(k)}} \right)^T \delta\mathbf{q} d\tau. \quad (21)$$

As a result, the first variations of J_2 is finally calculated as:

$$\delta J_2 = \sum_{i=0}^N \sum_{j \in \{x, y, z\}} \sum_{k=0}^n \int_{\tau_0}^{\tau_M} (-1)^k \left(\frac{d^k}{d\tau^k} \frac{\partial V_{ij}}{\partial \mathbf{q}^{(k)}} \right)^T \delta\mathbf{q} d\tau. \quad (22)$$

The necessary condition for an extrema of functional J_2 is that its first variations δJ_2 vanish. Since the $\delta\mathbf{q}$ is arbitrary, so the following condition should be satisfied:

$$\sum_{i=0}^N \sum_{j \in \{x, y, z\}} \sum_{k=0}^n (-1)^k \frac{d^k}{d\tau^k} \frac{\partial V_{ij}}{\partial \mathbf{q}^{(k)}} = \mathbf{0}. \quad (23)$$

By submitting (14) into (23) we can obtained the necessary conditions of solution to $\mathcal{P}_2$, which are summarized in the following theorem.

Theorem 1 (necessary Conditions for Two Quaternions Optimal Interpolation). *A trajectory, denoted by $\mathbf{q}^*(t) : [\tau_0, \tau_M] \mapsto \mathbb{R}^4$, is the optimal for the cost-funtional (12a) with boundary constraints (12b), if the following conditions are satisfied:*

- 1) *Boundary conditions (16) are satisfied.*
- 2) *The following differential equations are satisfied:*

$$\sum_{i=0}^N \sum_{j \in \{x, y, z\}} \mathbf{H}_{ij} \mathbf{q}^* (\mathbf{q}^{*T} \mathbf{H}_{ij} \mathbf{q}^*)^{(2n)} = \mathbf{0}. \quad (24)$$

Proof. The proof is detailed in Appendix II. ■

The differential equations (24) together with boundary conditions (16) is a *Two-Point Boundary Value Problem* (TPBVP). However, (24) are $2n$ -order non-linear differential equations, which do not admit an analytical solution in general. As a result, the TPBVP is solved with some numerical method, e.g., Finite Difference Method (FDM). FDM requires convert the high-order differential equation (24) to a first-order differential equation. A new variable is introduced as:

$$\mathbf{y} = \mathbf{q}^{[2n-1]} = (\mathbf{q}^T, \mathbf{q}^{(1)T}, \dots, \mathbf{q}^{(2n-1)T})^T, \quad (25)$$

whose time derivative is:

$$\dot{\mathbf{y}} = (\mathbf{q}^{(1)T}, \mathbf{q}^{(2)T}, \dots, \mathbf{q}^{(2n)T})^T. \quad (26)$$

Given any $\mathbf{y}$, only $\mathbf{q}^{(2n)}$ in $\dot{\mathbf{y}}$ is unknown, which can be calculated with differential equation (24). Expanding (24) with (6), we have:

$$\sum_{i=0}^N \sum_{j \in \{x, y, z\}} \sum_{k=0}^{2n} \binom{2n}{k} \mathbf{H}_{ij}^T \mathbf{q} (\mathbf{q}^{(2n-k)})^T \mathbf{H}_{ij} \mathbf{q}^{(k)} = \mathbf{0}. \quad (27)$$

Noting that, (27) is linear w.r.t $\mathbf{q}^{(2n)}$ when $\mathbf{y}$ is given. As a result, $\mathbf{q}^{(2n)}$ can be obtained by solving the following linear equation:

$$\mathbf{A} \mathbf{q}^{(2n)} = \mathbf{b}, \quad (28)$$

where $\mathbf{A}$ and $\mathbf{b}$ are calculated as:

$$\mathbf{A} = \sum_i^N \sum_{j \in \{x, y, z\}} \mathbf{H}_{ij}^T \mathbf{q} \mathbf{q}^T \mathbf{H}_{ij}, \quad (29)$$

$$\mathbf{b} = - \sum_i^N \sum_{j \in \{x, y, z\}} \left(\sum_{k=1}^{2n-1} \mathbf{q}^{(2n-k)} \mathbf{H}_{ij} \mathbf{q}^{(k)} \right) \mathbf{H}_{ij} \mathbf{q}. \quad (30)$$

To solve the linear equation (28), the invertibility of $\mathbf{A}$ should be determined, which is discussed by the following theorem.

Theorem 2. $\text{rank}(\mathbf{A}) = 4$ if and only if $\text{rank}(\mathbf{C}) \geq 2$.

Proof. Matrix $\mathbf{A}$ can be expressed as:

$$\mathbf{A} = \mathbf{U} \mathbf{U}^T \in \mathbb{R}^4, \\ \mathbf{U} = (\cdots, \mathbf{H}_{ij}^T \mathbf{q}, \cdots) \in \mathbb{R}^{4 \times 3N}.$$

Consequently, $\text{rank}(\mathbf{A}) = 4$ if and only if $\text{rank}(\mathbf{U}) = 4$. For any $\mathbf{C}$ that satisfied $\text{rank}(\mathbf{C}) \geq 2$, there exist a matrix $\mathbf{P} = (p_{ij}) \in \mathbb{R}^{N \times N}$ that:

$$\mathbf{C}^* = \mathbf{C} \mathbf{P} = \begin{pmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \quad (31)$$

where $\mathbf{I}_2$ is the identity matrix of $\mathbb{R}^{2 \times 2}$, $\mathbf{0}$ is the zero matrix with proper dimension.

Denote the matrix $\mathbf{U}$ after transformation as:

$$\mathbf{U}^* = (\cdots, \mathbf{H}_{ij}^{*T} \mathbf{q}, \cdots), \\ \mathbf{H}_{ij}^* = \mathbf{H}_j(\mathbf{c}_i^*) = \sum_{k=0}^N p_{ik} \mathbf{H}_{kj}.$$

The rank will not change after the transformation $\mathbf{P}$, that is, $\text{rank}(\mathbf{U}^*) = \text{rank}(\mathbf{U})$. It can be verified easily that $\forall \mathbf{q} \neq \mathbf{0}$, $\text{rank}(\mathbf{U}^*) = 4$.

On the contrary, when $\text{rank}(\mathbf{C}) < 2$, we can verified that $\forall \mathbf{q} \neq \mathbf{0}$, $\text{rank}(\mathbf{U}^*) \leq 3$. The proof is completed. ■

The condition $\text{rank}(\mathbf{C}) \geq 2$ implies that $\mathbf{c}_i, i = 1, \cdots, N$ are not collinear, that is, the desired geometry pattern is not a line. The Theorem 2 states that $\mathbf{q}^{(2n)}$ can be determined uniquely when the desired geometry pattern is not a line.

B. Optimal Interpolation of Multiple Quaternions

In this subsection, the intermediate quaternions constraints (12c) are taken into accounted.

For the multi-stage optimal interpolation, in addition to the constraints (16) and (17), the following constraints are also required:

$$\mathbf{q}^{*[d_m-1]}(\tau_i^-) = \mathbf{q}^{*[d_m-1]}(\tau_i^+) = \mathbf{q}_i^{[d_m-1]}, \quad (32)$$

$$\delta \mathbf{q}^{[d_m-1]}(\tau_i^-) = \delta \mathbf{q}^{[d_m-1]}(\tau_i^+) = \mathbf{0}, \quad (33)$$

where τ_i^- and τ_i^+ denote the left and right limit, respectively.

What's more, due to the form of cost-function (12a), which contains positive definite quadratic form of $\mathbf{q}^{(n)}$. It implies that $\mathbf{q}^{(n)}$ is bounded. As a result, the optimal trajectory should at least be $n - 1$ times continuously differentiable:

$$\mathbf{q}^{*[n-1]}(\tau_i^-) = \mathbf{q}^{*[n-1]}(\tau_i^+), \quad (34)$$

$$\delta \mathbf{q}^{[n-1]}(\tau_i^-) = \delta \mathbf{q}^{[n-1]}(\tau_i^+). \quad (35)$$

Similar to (20), use integration by parts in each interval:

$$\begin{aligned} & \sum_{m=0}^{M-1} \int_{\tau_m^+}^{\tau_{m+1}^-} \left(\frac{\partial V_{ij}}{\partial \mathbf{q}^{(k)}} \right)^T \delta \mathbf{q}^{(k)} d\tau \\ &= \sum_{m=0}^{M-1} \sum_{l=0}^{k-1} (-1)^l \left(\frac{d^l}{d\tau^l} \frac{\partial V_{ij}}{\partial \mathbf{q}^{(k-l)}} \right)^T \delta \mathbf{q}^{(k-l-1)} \Big|_{\tau_m^+}^{\tau_{m+1}^-} \\ &+ \sum_{m=0}^{M-1} \int_{\tau_m^+}^{\tau_{m+1}^-} (-1)^k \left(\frac{d^k}{d\tau^k} \frac{\partial V_{ij}}{\partial \mathbf{q}} \right)^T \delta \mathbf{q} d\tau. \end{aligned} \quad (36)$$

The conditions of the second term of (36) becomes zero are the same as two points interpolation. The only difference is the first term of (36). Due to boundary constraints (17) and intermediate constraints (33), the first term of (36) can be re-ordered as:

$$\begin{aligned} & \sum_{m=0}^{M-1} \sum_{l=0}^{k-1} (-1)^l \left(\frac{d^l}{d\tau^l} \frac{\partial V_{ij}}{\partial \mathbf{q}^{(k-l)}} \right)^T \delta \mathbf{q}^{(k-l-1)} \Big|_{\tau_m^+}^{\tau_{m+1}^-} \\ &= \sum_{m=1}^{M-1} \sum_{l=0}^{k-1} (-1)^l \left(\frac{d^l}{d\tau^l} \frac{\partial V_{ij}}{\partial \mathbf{q}^{(k-l)}} \right)^T \delta \mathbf{q}^{(k-l-1)} \Big|_{\tau_m^+}^{\tau_m^-} \\ &= \sum_{m=1}^{M-1} \sum_{l=0}^{k-d_m} (-1)^l \left(\frac{d^l}{d\tau^l} \frac{\partial V_{ij}}{\partial \mathbf{q}^{(k-l)}} \right)^T \delta \mathbf{q}^{(k-l-1)} \Big|_{\tau_m^+}^{\tau_m^-}. \end{aligned} \quad (37)$$

For simplicity, let $V_{ij,kl} = (-1)^l \frac{d^l}{d\tau^l} \frac{\partial V_{ij}}{\partial \mathbf{q}^{(k-l)}}$. According to the continuously differentiable constraints (35), the above formula is further expanded as:

$$\begin{aligned} & (V_{ij,kl})^T \delta \mathbf{q}^{(k-l-1)} \Big|_{\tau_m^+}^{\tau_m^-} \\ &= \left(V_{ij,kl}(\tau_m^-) - V_{ij,kl}(\tau_m^+) \right)^T \delta \mathbf{q}^{(k-l-1)}(\tau_m). \end{aligned} \quad (38)$$

The first variation of J_2 vanish if $V_{ij,kl}(\tau_m^-) = V_{ij,kl}(\tau_m^+)$. And note that, $V_{ij,kl}$ is a function with $\mathbf{q}^{[n+l-1]}$ as parameter. Consequently, the necessary condition for extrema is:

$$\mathbf{q}^{[n+l-1]}(\tau_m^-) = \mathbf{q}^{[n+l-1]}(\tau_m^+), \quad (39)$$

where $n \leq n + l - 1 \leq 2n - d_m - 1$, the second equal sign establishes when $k = n$ and $l = n - d_m$. The equation (39) means that the optimal trajectory $\mathbf{q}^*$ should be $2n - d_m - 1$ times continuously differentiable at time knot τ_m . We summary the above result in the following Theorem.

Theorem 3 (necessary Conditions for Multiple Quaternions Optimal Interpolation). *A trajectory, denoted by $\mathbf{q}^*(t) : [\tau_0, \tau_M] \mapsto \mathbb{R}^4$, is the optimal for the problem $\mathcal{P}_2$ if the following conditions are satisfied:*

- 1) *Boundary conditions (16) and intermediate constraints (32) are satisfied.*
- 2) *The differential equations (24) are satisfied in each time interval $[\tau_m, \tau_{m+1})$, $m = 0, 1, \dots, M - 1$.*
- 3) *The optimal trajectory $\mathbf{q}^*(\tau)$ is $d_m - 1$ times continuously differentiable at each time knot τ_m , $m = 1, \dots, M - 1$.*

Proof. The first and second conditions can be directly concluded from Theorem 1. And the third condition can be concluded from (39). The proof is completed. ■

According to Theorem (3), the problem $\mathcal{P}_2$ is a Multi-Point Boundary Value Problem (MPBVP). The MPBVP can be solved using the same numerical methods as TPBVP.

IV. POLYNOMIAL APPROXIMATION

Solving the TPBVP and MPBVP numerically is to obtain a high-resolution optimal trajectory is computational expensive, thus not a practical option for applications like real-time motion generation of multi-agent formation, interactive CAD applications, etc. In this section, an algorithm that approximates the optimal trajectory is proposed.

To begin with, the time complexity of numerical method is analyzed. The FDM method involves the inverse of a $HMn \times HMn$ matrix in each iteration, whose time complexity is of order $O(H^3 M^3 n^3)$. The time complexity increase mainly due to the increase of H . We circumvent this problem by finding a low-resolution solution first. And then interpolating the low-resolution solution using a M -piece polynomial which preserves the structure of the optimal solution.

An M -piece polynomial is used to represent the Optimal trajectory. The m -th piece polynomial is defined as:

$$\mathbf{q}_m(\tau) = \beta(\tau)^T \kappa_m, \quad \forall \tau \in [0, T_m), \quad m = 1, \dots, M. \quad (40)$$

where $T_m = \tau_m - \tau_{m-1}$ is the duration of polynomial, $\kappa_m \in \mathbb{R}^{(d+1) \times 4}$ is the coefficient of polynomial and $\beta(t) = [1, \tau, \dots, \tau^d]^T \in \mathbb{R}^{d+1}$ is the natural basis. And d is the degree of polynomial.

To preserve the structure of optimal solution, the constraints that a approximation trajectory $\mathbf{q}$ should satisfies excluding differential equation (24) are listed as follows:

- 1) $4 \times 2n$ constraints from (16).
- 2) $4 \times \sum_{m=1}^{M-1} d_m$ constraints from (32).
- 3) $4 \times \sum_{m=1}^{M-1} 2n - d_m$ constraints from (39).

The number of constraints is $4 \times (2n + \sum_{m=1}^{M-1} (d_m + 2n - d_m)) = 4 \times 2nM$. Consequently, if an M -piece polynomial is used to approximate the optimal trajectory, the degree of polynomial in each piece should be at least $2n - 1$. And when a higher degree polynomial is choosen, more degrees of freedom are provided for approximation. In this paper, H extra points are added into constraints to approximate the differentiable equation (24) in each piece.

The main steps of algorithm are summarized as follows: Firstly, a coarse trajectory with H extra points is calculated using FDM. The extra points $\hat{\mathbf{q}}_{m,h}$, $h = 1, \dots, H$ are evenly distributed in time domain, as shown in Fig. 3. Secondly, the optimal trajectory is approximated using M -piece polynomial with $2n + H - 1$ degree.

Denote the coefficients of M -piece polynomial as $\mathcal{K} = (\kappa_1, \kappa_2, \dots, \kappa_M) \in \mathbb{R}^{M(2n+H) \times 4}$. The m -th polynomial should passthrough $\hat{\mathbf{q}}_{m,h}$ and $\mathbf{q}_m^{[d_m-1]}$, hence, the following constraints are needed:

$$\beta\left(\frac{h}{H+1}T_m\right)^T \kappa_m = \hat{\mathbf{q}}_{m,h}, \quad h = 1, \dots, H \quad (41)$$

$$\beta^{(i)}(T_m)^T \kappa_m = \mathbf{q}_m^{(i)}, \quad i = 0, 1, \dots, d_m - 1. \quad (42)$$

To enforce the continuous constraints (39), the following constraints are needed:

$$\beta^{(i)}(T_m)^T \kappa_m = \beta^{(i)}(0)^T \kappa_{m+1}, \quad i = 0, \dots, 2n - d_m - 1. \quad (43)$$

There are $4 \times (2n + H)M$ independence linear constraints in (41)-(43). Thus, the coefficient $\mathcal{K}$ can be solved uniquely. What is more, due to the sparsity of constraints, the time complexity of solving $\mathcal{K}$ is $O(M(2n + H)^3)$ [21].

As a result the total time complexity of polynomial is $O(H^3 M^3 n^3 + M(2n + H)^3)$. Although the order of time complexity do not decrease, we show in Sec V that the proposed method can approximate the optimal trajectory with small H . Additionally, since the optimal trajectories are approximated by polynomials, the trajectories can be evaluated continuously, that is, evaluated in arbitrary resolution. And this property makes it suitable for motion generation.

V. SIMULATION AND COMPARISON

In this section, we will present some examples of the proposed method for motion interpolation in similarity transformations. We also compare the proposed method in Sec IV with numerical method and the method proposed in [16].

We generate a trajectory for ellipsoid formation. More precisely, $N = 10$ agents are evenly distributed on the surface of an ellipsoid. The parameters are set as follows: $n = 3$, $H = 2$, $M = 3$. The initial states $\mathbf{q}_0 = (-0.5, 0.4, 0.3, 0.6)^T$, final state $\mathbf{q}_3 = (0.8, 0.6, 0.4, 0.4)^T$ and $\mathbf{q}_0^{(i)} = \mathbf{q}_3^{(i)} = \mathbf{0}$, $i = 1, 2$. $d_1 = d_2 = 1$, and the intermediate states are $\mathbf{q}_1 = (-0.45, 0.55, 0.45, 0.25)^T$, $\mathbf{q}_2 = (0.3, 0.4, 0.4, 0.5)^T$. And the time knots are, $\tau_0 = 0$, $\tau_1 = 4s$, $\tau_2 = 8s$, $\tau_3 = 12s$. The trajectory of optimal quaternion is shown in Fig. 3. And the corresponding trajectories of desired geometry pattern and different agents are shown in Fig. 1-2

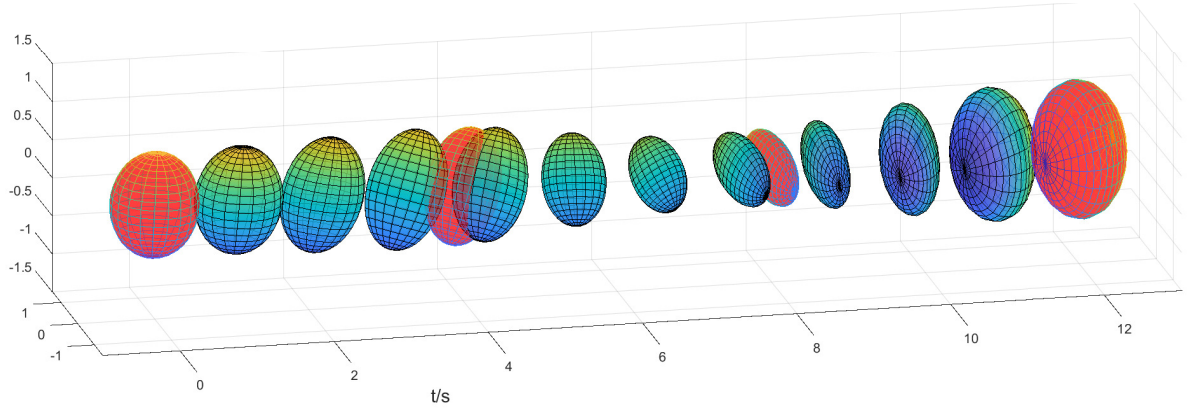


Fig. 1. The optimality trajectory of geometry pattern. The red ellipsoids are the states that should be passthrough.

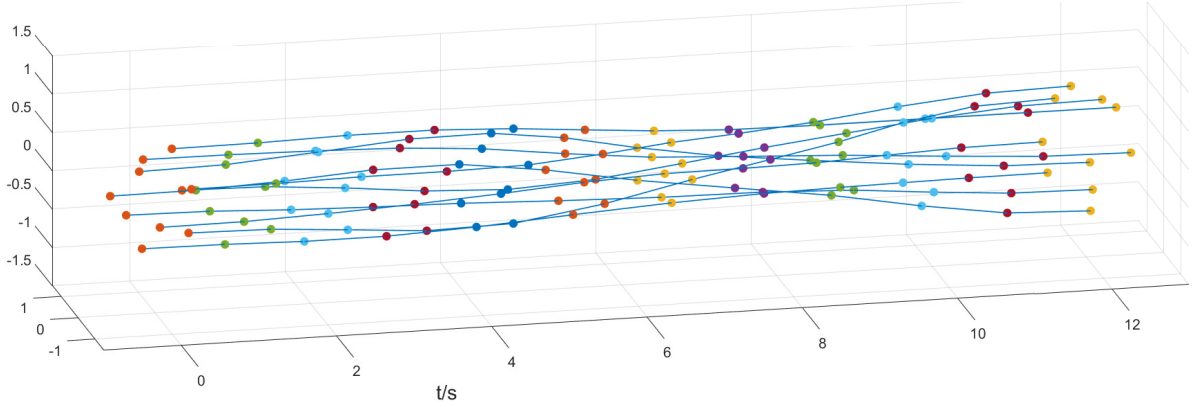


Fig. 2. The optimality trajectories of different agents. The points with the same color are the positions of agents with the same time stamp.

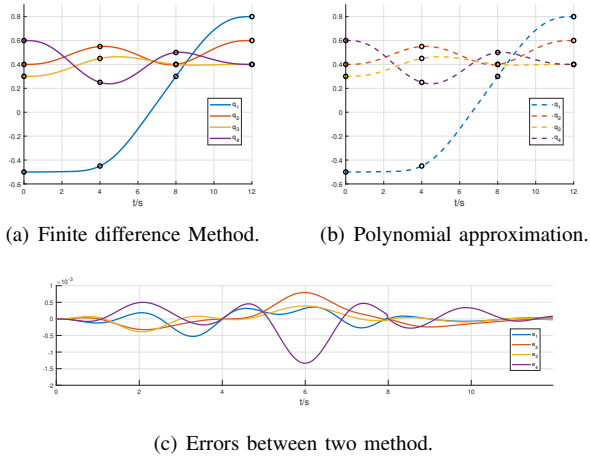


Fig. 3. Comparison between FDM and polynomial approximation.

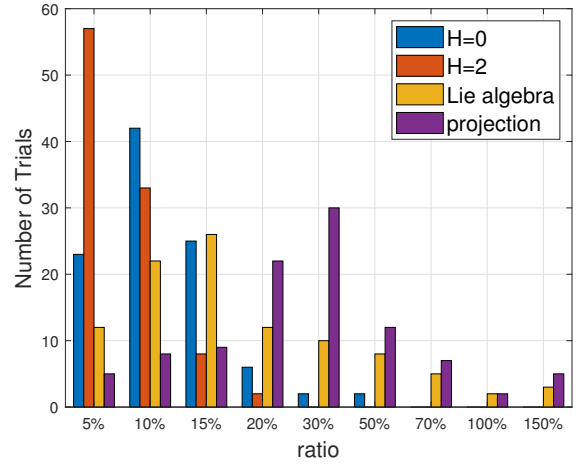


Fig. 4. The distribution of the ratio of J_2 to optimal cost J_2^* .

To show the efficiency of the proposed polynomial approximation method, we compare the proposed method with FDM, the Lie algebra interpolation method [16] and the projection method [13]. The boundary and intermediate quaternions are generated randomly, and the time derivatives of boundary quaternion are all set to zero.

The distribution of the ratio of J_2 to J_2^* (minimum cost calculated by numerical method) are summarized in Fig. 4. It can be seen that our method outperforms the Lie algebra interpolation method [16] and projection method [13], even

when $H = 0$. And by selecting a larger H , the proposed method will approximate the optimal trajectory better.

However, as analyzed in Sec. IV, selecting a larger H will result in higher computational cost. In practice, the parameters should be selected by trading off optimality and computational cost.

VI. CONCLUSIONS

This paper investigates the optimal motion interpolation problem of similarity transformations. A novel quaternion-

based representation for similarity transformations is proposed to ease the evaluation of the trajectories and their time derivatives. We formulate the optimal interpolation of similarity transformations as a variational problem. Furthermore, the necessary conditions for two points interpolations and multiple points interpolations are derived. A polynomial approximation method is proposed to improve the computational efficiency. The simulations and comparisons show the efficiency of the proposed method.

REFERENCES

- [1] J. Han and Y. Chen, "Multiple uav formations for cooperative source seeking and contour mapping of a radiative signal field," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1, pp. 323–332, 2014.
- [2] J. J. Acevedo, B. C. Arrue, I. Maza, and A. Ollero, "A decentralized algorithm for area surveillance missions using a team of aerial robots with different sensing capabilities," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 4735–4740.
- [3] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Trans. Robot. Autom.*, vol. 14, no. 6, pp. 926–939, 1998.
- [4] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robot. Autom. Mag.*, vol. 19, no. 1, pp. 24–39, 2012.
- [5] S. Zhao and D. Zelazo, "Translational and scaling formation maneuver control via a bearing-based approach," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 3, pp. 429–438, 2017.
- [6] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *Int. J. Robot. Res.*, vol. 36, no. 9, pp. 1000–1021, 2017.
- [7] R. Van Parys and G. Pipeleers, "Distributed model predictive formation control with inter-vehicle collision avoidance," in *Proc. IEEE Asian Control Conf.*, 2017, pp. 2399–2404.
- [8] P. C. Lusk, X. Cai, S. Wadhwan, A. Paris, K. Fathian, and J. P. How, "A distributed pipeline for scalable, deconflicted formation flying," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5213–5220, 2020.
- [9] Y. H. Choi and D. Kim, "Distance-based formation control with goal assignment for global asymptotic stability of multi-robot systems," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2020–2027, 2021.
- [10] L. Quan, L. Yin, C. Xu, and F. Gao, "Distributed swarm trajectory optimization for formation flight in dense environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2022, pp. 4979–4985.
- [11] B. Zhou, J. Pan, F. Gao, and S. Shen, "Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1992–2009, 2021.
- [12] S. Zhang, R. Wang, Z. Jian, W. Zhan, N. Zheng, and M. Tomizuka, "Controllable clothoid path generation for autonomous vehicles," *IEEE Robotics and Automation Letters*, 2023.
- [13] C. Belta and V. Kumar, "An svd-based projection method for interpolation on se (3)," *IEEE transactions on Robotics and Automation*, vol. 18, no. 3, pp. 334–345, 2002.
- [14] C. Belta and V. Kumar, "Motion generation for formations of robots: A geometric approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, 2001, pp. 1245–1250 vol.2.
- [15] F. C. Park and B. Ravani, "Smooth invariant interpolation of rotations," *ACM Transactions on Graphics (TOG)*, vol. 16, no. 3, pp. 277–295, 1997.
- [16] S. Leonardos, C. Allen-Blanchette, and J. Gallier, "The exponential map for the group of similarity transformations and applications to motion interpolation," in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 2015, pp. 377–382.
- [17] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 2520–2525.
- [18] M. Zefran, V. Kumar, and C. Croke, "Metrics and connections for rigid-body kinematics," *Int. J. Robot. Res.*, vol. 18, no. 2, pp. 243–258, 1999.
- [19] M. Watterson, T. Smith, and V. Kumar, "Smooth trajectory generation on se (3) for a free flying space robot," in *Proc. IEEE Int. Conf. Intell. Robots Syst.* IEEE, 2016, pp. 5459–5466.
- [20] V. Kyrki, "Quaternion representation for similarity transformations in visual slam," in *Proc. IEEE Int. Conf. Intell. Robots Syst.* IEEE, 2008, pp. 2498–2503.
- [21] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Trans. Robot.*, pp. 1–10, 2022.
- [22] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Josa a*, vol. 4, no. 4, pp. 629–642, 1987.
- [23] J. Riordan, *Introduction to combinatorial analysis*. Courier Corporation, 2012.

APPENDIX I

The equation (4) can be expressed in matrix product form [22]:

$$\mathbf{p} = \mathbf{Q}\mathbf{c} + \mathbf{t},$$

$$\mathbf{Q} = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z + q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_z q_y + q_w q_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}$$

Expanding the (I), the following equations are obtained:

$$\mathbf{H}_x = \begin{bmatrix} c_x & 0 & c_z & -c_y \\ 0 & c_x & c_y & c_z \\ c_z & c_y & -c_x & 0 \\ c_y & c_z & 0 & -c_x \end{bmatrix},$$

$$\mathbf{H}_y = \begin{bmatrix} c_y & -c_z & 0 & c_x \\ -c_z & -c_y & c_x & 0 \\ 0 & c_x & c_y & c_z \\ c_x & 0 & c_z & -c_y \end{bmatrix},$$

$$\mathbf{H}_z = \begin{bmatrix} c_z & c_y & -c_x & 0 \\ c_y & -c_z & 0 & c_x \\ -c_x & 0 & -c_z & c_y \\ 0 & c_x & c_y & c_z \end{bmatrix}.$$

APPENDIX II

PROOF OF THEOREM 1

From (22) to (23), we only need to proof:

$$\sum_{k=0}^n (-1)^k \frac{d^k}{d\tau^k} \frac{\partial V_{ij}}{\partial \mathbf{q}^{(k)}} = \mathbf{H}_{ij} \mathbf{q} (\mathbf{q}^T \mathbf{H}_{ij} \mathbf{q})^{(2n)}. \quad (44)$$

To proof the above equation, a lemma is needed:

Lemma 4 (Chapter 1, [23]). *P(k) is a polynomial with k as parameter. And the degree of P(k) is less than n. Then the following equation holds.*

$$\sum_{k=0}^n (-1)^k \binom{n}{k} P(k) = 0. \quad (45)$$

According to (23), we have:

$$\frac{\partial V_{ij}}{\partial \mathbf{q}^{(k)}} = 4 \binom{n}{k} \mathbf{H}_{ij} \mathbf{q}^{(n-k)} (\mathbf{q}^T \mathbf{H}_{ij} \mathbf{q})^{(n)}.$$

The k-th order time derivative is calculated according to binomial theorem:

$$\frac{d^k}{d\tau^k} \frac{\partial V_{ij}}{\partial \mathbf{q}^{(k)}} = 4 \binom{n}{k} \mathbf{H}_{ij} \left\{ \sum_{l=0}^k \binom{k}{l} \mathbf{q}^{(n-l)} (\mathbf{q}^T \mathbf{H}_{ij} \mathbf{q})^{(n+l)} \right\}.$$

Note that the binomial coefficients

$$\binom{k}{l} = \frac{(n+k)!}{l!(k-l)!} = \frac{k(k-1) \cdots (k-l+1)}{l!}.$$

is a polynomial with k as parameter, whose degree is l.

Re-ordering the summation, we have:

$$\begin{aligned} & \sum_{k=0}^n (-1)^k \frac{d^k}{d\tau^k} \frac{\partial V_{ij}}{\partial \mathbf{q}^{(k)}} \\ &= \sum_{l=0}^n \sum_{k=l}^n (-1)^k \binom{n}{k} \binom{k}{l} \mathbf{H}_{ij} \mathbf{q}^{(n-l)} (\mathbf{q}^T \mathbf{H}_{ij} \mathbf{q})^{(n+l)}. \end{aligned} \quad (46)$$

By using the Lemma 4, all term with $l < n$ become zero. Only the term with $l = k = n$ remains, which is exactly the equation (44). The proof is completed.