Attitude synchronization control for multi-agent systems on unit Quaternions

Pieter van Goor, Zhiyong Sun, and Changbin (Brad) Yu

Abstract— **This paper discusses an attitude synchronization problem for multi-agent systems in 3-D spaces and proposes a synchronization control law based on the quaternion representation of 3D rotations. The proposed control strategy takes advantage of the geometry of the unit quaternion space and synchronizes agents to have representative quaternions that map to the same physical 3D rotation. Each agent's attitude state is represented by a unit quaternion, and the attitude synchronization law ensures the states remain in the unit quaternion space. We show that this control law is stable for the desired equilibrium using a Lyapunov approach, and provide a simulation of nine agents interacting in an undirected graph to justify our results. We also provide a detailed analysis of the two-agent system to provide intuition behind the control law's operation.**

I. Introduction

In the last decade, multi-agent consensus and coordination control has been an active topic of research in the control community; we refer the reader to [1], [2] for an overview of this area. In this paper, we consider a particular multi-agent coordination problem relating to agents' attitude control and synchronization. Attitude synchronization control has been related with and motivated by vast engineering applications, including motion control of rigid bodies, satellites' orientation alignments, and spacecraft formation. In contrast to the consensus and coordination dynamics in linear vector spaces such as $\mathbb{R}^n$ as reviewed in [1], [2], system dynamics for attitude synchronization involve fundamental *nonlinear* characteristics in the representation of attitude variables, such as rotation matrices (the special orthogonal matrix group SO(d)), Euler angles and quaternions [3].

In this paper we consider the multi-agent attitude synchronization problem in 3-D spaces when agents' attitudes are presented by unit quaternions. Unit quaternions are often used to parametrize SO(3) (the 3-D rotation matrix group), while they also feature several advantages. For example, unit quaternion representation of an attitude or rotation has clear physical interpretation, in the sense that the independent definition of an axis of rotation and an angle is directly related to a rigid body's physical attitude quantities. Furthermore, unit quaternion gives a globally nonsingular representation of rigid-body attitude [4], which avoids the inherent singularity of other representations while a continuous state-feedback control law for global stabilization of attitude dynamics may not be possible [5].

This work was supported by the Australian Research Council under grant DP160104500.

Pieter van Goor, Zhiyong Sun and Changbin (Brad) Yu are with College of Engineering and Computer Science, The Australian National University, Canberra ACT 2601.

The key challenge in dealing with attitude control with unit-quaternion space or SO(3) as opposed to $\mathbb{R}^n$, for which multi-agent consensus is a largely well solved problem, is the non-linearity of the attitude space and the constraints its elements must abide by [6]. In this paper, we represent agents' attitudes or rotations using the unit quaternions, which can be thought of as the elements of the 3-sphere $\mathbb{S}^3 = \{\mathbf{x} \in \mathbb{R}^4 : ||\mathbf{x}|| = 1\}$. Interestingly, there have been investigations into multi-agent synchronization where agent states lie in the 1-sphere such as in [7] and in the 2-sphere such as in [8], or in general higher-dimensional spheres [9]. However, the issue with extending either of these controllers to the 3-sphere is non-trivial, while the nonlinear space of unit-quaternion representations features its own topological properties as discussed in e.g., [10], [11] which studied the attitude stabilization control for a single rigid body using quaternion representation. However, there are relatively few works in the literature on multi-agent attitude control using unit quaternion representations.

The focus of this paper is to propose a feasible and novel control law to reach multi-agent attitude synchronization on unit quaternions. We begin by providing preliminary results pertaining to quaternion operations and their relationship with 3-D rotations, and then formulate the problem. Following this we show the main results and provide proof that the attitude synchronization dynamics are stable with respect to the agreement equilibrium using a Lyapunov function argument. Finally, we show the results of simulations on a 9-agent system to demonstrate and justify the proposed attitude synchronization law.

II. Preliminaries

A. Graph theory

Graphs provide useful representations of multi-agent systems, where each of the n agents in a system is considered as a vertex of the graph, and each communication link is considered as an edge of the graph. These communication links can be directed or undirected. In this paper, we only consider undirected graph modelings.

Definition 1 (Undirected Graph [12])**.** *An undirected graph $\mathcal{G}$ can be represented by a pair of two sets $\mathcal{G} = (\mathcal{V}, E)$ where $V = \{1, ..., n\}$ is called the vertex set, and $E \subset \mathcal{V} \times \mathcal{V}$ is called the edge set and for any $i, j \in \{1, ..., n\}$ we have $(i, j) \in E \Leftrightarrow (j, i) \in E$.*

Definition 2 (Connected Graph [12])**.** *An undirected graph $\mathcal{G}$ is connected when, for any two vertices $i, j \in \mathcal{V}(\mathcal{G})$ there is a sequence of edges in $E(\mathcal{G})$ that link i and j.*

Definition 3 (Neighbourhood [12]). *The neighborhood N_i of the agent i is defined as*

$$N_i = \{j \in \mathcal{V} | (i,j) \in E\}$$

B. Quaternions and rotations

In many ways, quaternions are an extension of the complex numbers. This results in them lending themselves well to describing 3D physical operations in much the same way that complex numbers can be used to describe 2D physical operations.

Definition 4 (Quaternion [13]). *The set of quaternions is denoted $\mathbb{H}$. Each quaternion consists of a scalar (real) part $r \in \mathbb{R}$ and a vector (imaginary) part $\mathbf{u} = (x, y, z) \in \mathbb{R}^3$. Two common representations used for a quaternion $q \in \mathbb{H}$ are*

$$q \equiv [r, \mathbf{u}]$$

and

$$q \equiv r + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are quaternion bases, with $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$, $\mathbf{ij} = \mathbf{k}$, and $\mathbf{ji} = -\mathbf{k}$.

Proposition 1 (Quaternion Addition and Multiplication, Quaternion Conjugate [13]). *Given two quaternions $q_1 = [r_1, \mathbf{u}_1], q_2 = [r_2, \mathbf{u}_2] \in \mathbb{H}$, addition and multiplication can be expressed as follows.*

$$q_1 q_2 = [r_1 r_2 - \mathbf{u}_1 \cdot \mathbf{u}_2, \mathbf{u}_1 \times \mathbf{u}_2 + r_1 \mathbf{u}_2 + r_2 \mathbf{u}_1]$$

$$q_1 + q_2 = [r_1 + r_2, \mathbf{u}_1 + \mathbf{u}_2]$$

From this it can be seen that addition is commutative, but multiplication is not. Quaternion multiplication is however associative, and distributive over addition.
The conjugate of a quaternion $q = [r, \mathbf{u}]$ is defined as $q^* = [r, -\mathbf{u}]$.

Definition 5 (Quaternion Inner Product [14]). *Given two quaternions $q_i = [r_i, \mathbf{u}_i], q_j = [r_j, \mathbf{u}_j]$ we define their inner product as if they were vectors in $\mathbb{R}^4$:*

$$\langle q_i, q_j \rangle = r_i r_j + \langle \mathbf{u}_i, \mathbf{u}_j \rangle$$

where $\langle \mathbf{u}_i, \mathbf{u}_j \rangle$ is the standard inner product between vectors in $\mathbb{R}^3$.

Definition 6 (Unit Quaternion [13]). *A unit quaternion $q = [r, \mathbf{u}]$ has the property that $||q|| = \sqrt{r^2 + ||\mathbf{u}||^2} = 1$.*

Immediately, we can observe that the product of two unit quaternions is still a unit quaternion, whereas the sum is, in general, not. Thus the set of unit quaternions is clearly a group under multiplication, much alike the 3D rotation group SO(3). The identity element of this group is simply $q_I = [1, \mathbf{0}_3]$. Inverses in this group are provided by the conjugates: $q^* q = [r^2 + ||\mathbf{u}||^2, \mathbf{u} \times \mathbf{u} + r\mathbf{u} - r\mathbf{u}] = [1, \mathbf{0}] = q_I$.

Definition 7 (The Special Orthogonal Group SO(3), or 3D Rotation Group [13]). *The 3D Rotation Group SO(3) contains all possible matrices that cause a rotation about the origin of $\mathbb{R}^3$. It can be defined as SO(3) $= \{R \in \mathbb{R}^{3\times3} | R^T R = I_3\}$ and is a group under matrix multiplication.*

Proposition 2 (Unit Quaternions as Representations of Rotation [15]). *Each matrix $R \in$ SO(3) can be represented by an axis of rotation and an angle of rotation by Euler's Rotation Theorem (see e.g., [13, Chapter 2]). Given the axis angle representation of a rotation as $\mathbf{v}$ and θ respectively, we may represent the rotation using the unit quaternion $q = [cos(\theta/2), \mathbf{v} sin(\theta/2)]$. Moreover, every unit quaternion represents exactly one rotation matrix.*

While every unit quaternion has a unique rotation it corresponds to and we can map 3D rotations into the unit quaternion group, the inverse map from the group of unit quaternions to SO(3) is not injective. In particular, consider a unit quaternion $q = [r, \mathbf{u}]$ and its negative $-q = [-r, \mathbf{u}]$. Given that q has axis $\mathbf{v}$ and angle θ, we can write $r = cos(\theta/2)$ and $\mathbf{u} = \mathbf{v} sin(\theta/2)$. Now we analyse what $-q$ represents. We have $-r = -cos(\theta/2) = cos(\pi - \theta/2) = cos((2\pi - \theta)/2)$ and $-\mathbf{u} = -\mathbf{v} sin(\theta/2) = -\mathbf{v} sin((2\pi - \theta)/2)$. Noting that a rotation by $2\pi - \theta$ is the same as a rotation by $-\theta$, we see that $-q$ is a rotation by $-\theta$ on the axis $-\mathbf{v}$, and therefore $-q$ represents the same rotation as q.

C. Problem formulation

This paper aims to develop a control law that can be used by a multi-agent system to achieve rotation consensus. In particular, we wish to use the quaternion representation of 3D rotations and define a control law such that each agent modeled in an undirected network can be driven to reach rotation synchronization. The state of agent i is defined by a unit quaternion q_i, and the relative quaternion between two neighboring agents i and j is defined as $q_{ij} = [r_{ij}, \mathbf{u}_{ij}] := q_i^* q_j$. Given that the graph $\mathcal{G} = (V, E)$ represents the agents and their connections, we aim to design a control law to achieve rotation synchronization, in the sense that

$$\forall (i,j) \in E, q_{ij}(t) \to \pm q_I, \text{as } \ t \to +\infty$$

III. Main results

A. Rotation control system represented by unit quaternions

1) Defining agent states: We describe the orientation of a given agent i by using a quaternion $q_i := q_i(t) = [cos(\frac{\theta_i(t)}{2}), \mathbf{v}_i(t) sin(\frac{\theta_i(t)}{2})]$ as above, where $\theta_i(t) \in [0, 2\pi]$ and $\mathbf{v}_i(t) \in \mathbb{R}^3$ refer to the angle and axis of rotation in 3D space respectively. We further simplify our notation by writing $r_i = cos(\frac{\theta_i(t)}{2})$ and $\mathbf{u}_i = \mathbf{v}_i(t) sin(\frac{\theta_i(t)}{2})$ such that $q_i = [r_i, \mathbf{u}_i]$.

2) Defining the control law: A common approach to consensus control in multi-agent systems is to adopt the nearest-neighbor algorithm [16], which causes each agent to move to the average relative position of all its neighboring agents in Euclidean space. The space of unit quaternions is, however, non-linear, and this requires that our control law respects the topological properties of the unit quaternion space. To accomplish this, we utilize a function that transforms each relative quaternion into a purely imaginary quaternion, so that when left-multiplied by q_i, we obtain a motion perpendicular to q_i, similarly to how multiplying a purely imaginary complex number ic by another complex number $a+ib$ results in a complex number perpendicular to $a+ib$ in the complex plane. Moreover, the function we define preserves the direction of the imaginary component of the input quaternion, and this results in q_i tending towards the relative average state of its neighboring agent quaternions. Thus, given an arbitrary quaternion $q=[r,\mathbf{u}]$, we define the function $f(q)=[0,r\mathbf{u}]$, which represents a purely imaginary quaternion capturing the necessary information from q. Then we define our control law to be

$$\dot{q}_i = q_i \sum_{j\in N_i} f(q_i^* q_j) \tag{1}$$

To show that this control law indeed preserves the unit length of the quaternions to which it is applied, we show that the vectors $q_i f(q_i^* q_j)$ are perpendicular to q_i by calculating their inner product. To accomplish this, we begin by finding a simplified expression for $q_i f(q_i^* q_j)$.

$$q_i f(q_i^* q_j) = [r_i, \mathbf{u}_i][0, r_{ij}\mathbf{u}_{ij}]$$

By standard quaternion multiplication this gives us

$$q_i f(q_i^* q_j) = [-r_{ij}\langle \mathbf{u}_i, \mathbf{u}_{ij}\rangle, r_{ij}\mathbf{u}_i \times \mathbf{u}_{ij} + r_i r_{ij}\mathbf{u}_{ij}]$$

Expanding $\mathbf{u}_{ij}$

$$\begin{aligned} q_i f(q_i^* q_j) = [&-r_{ij}\langle \mathbf{u}_i, (-\mathbf{u}_i \times \mathbf{u}_j + r_i\mathbf{u}_j - r_j\mathbf{u}_i)\rangle, \\ &r_{ij}\mathbf{u}_i \times (-\mathbf{u}_i \times \mathbf{u}_j + r_i\mathbf{u}_j - r_j\mathbf{u}_i) \\ &+ r_i r_{ij}(-\mathbf{u}_i \times \mathbf{u}_j + r_i\mathbf{u}_j - r_j\mathbf{u}_i)] \end{aligned}$$

Now we place r_{ij} outside the vector as a factor, and resolve the expressions inside. Note we use the properties of the vector triple product to resolve $\mathbf{u}_i \times (\mathbf{u}_i \times \mathbf{u}_j)$.

$$\begin{aligned} q_i f(q_i^* q_j) = r_{ij}[&-r_i\langle \mathbf{u}_i \mathbf{u}_j\rangle + r_j||\mathbf{u}_i||^2, \\ &-\mathbf{u}_i\langle \mathbf{u}_i, \mathbf{u}_j\rangle + \mathbf{u}_j||\mathbf{u}_i||^2 + r_i^2\mathbf{u}_j - r_i r_j\mathbf{u}_i] \end{aligned}$$

Using this expression, we may show this vector is perpendicular to q_i by showing that the inner product $\langle q_i, q_i f(q_i^* q_j)\rangle = 0$ as follows.

$$\begin{aligned} \frac{\langle q_i, q_i f(q_i^* q_j)\rangle}{r_{ij}} &= -r_i^2\langle \mathbf{u}_i, \mathbf{u}_j\rangle + r_i r_j||\mathbf{u}_i||^2 \\ &\quad - ||\mathbf{u}_i||^2\langle \mathbf{u}_i, \mathbf{u}_j\rangle + \langle \mathbf{u}_i, \mathbf{u}_j\rangle||\mathbf{u}_i||^2 \\ &\quad + r_i^2\langle \mathbf{u}_i, \mathbf{u}_j\rangle - r_i r_j||\mathbf{u}_i||^2 \\ &= 0 \\ \therefore \qquad \langle q_i, q_i f(q_i^* q_j)\rangle &= 0 \end{aligned} \tag{2}$$

Therefore, we have shown that q_i is perpendicular to $q_i f(q_i^* q_j)$ independent of i and j. Thus, since the control law dictates that $\dot{q}_i$ is set equal to the sum of $q_i f(q_i^* q_j)$ for all $j \in N_i$, it is clear that $\dot{q}_i$ is set perpendicular to q_i, and that it therefore preserves the unit length constraint placed on q_i.

Remark 1. *For a unit quaternion $q=[cos(\theta), \mathbf{v}sin(\theta)]$, it holds $log(q)=[0,\mathbf{v}]$ (see e.g.,[17]), and the left multiplication of $[0,\mathbf{v}]$ by q renders a tangent vector at q of the unit quaternion space. The above argument actually shows that $q_i f(q_i^* q_j)$ is a velocity vector in the tangent space of unit quaternion space at point q_i, and therefore the composite vector $q_i \sum_{j\in N_i} f(q_i^* q_j)$ still lives in the tangent space at point q_i, which guarantees the solution of* (1) *is in the unit quaternion space.*

3) Relative quaternions and Lyapunov function: For the relative quaternion q_{ij}, by standard quaternion multiplication, we obtain $r_{ij} = r_i r_j + \langle \mathbf{u}_i, \mathbf{u}_j\rangle$ and $\mathbf{u}_{ij} = -\mathbf{u}_i \times \mathbf{u}_j + r_i\mathbf{u}_j - r_j\mathbf{u}_i$. There are also two useful identities here: $r_{ij} = r_i r_j + \langle \mathbf{u}_i, \mathbf{u}_j\rangle = r_j r_i + \langle \mathbf{u}_j, \mathbf{u}_i\rangle = r_{ji}$ and $\mathbf{u}_{ij} = -\mathbf{u}_i \times \mathbf{u}_j + r_i\mathbf{u}_j - r_j\mathbf{u}_i = \mathbf{u}_j \times \mathbf{u}_i - r_j\mathbf{u}_i + r_i\mathbf{u}_j = -\mathbf{u}_{ji}$. The relative quaternion q_{ij} is also representative of the relative rotation between agents i and j. As such, when agents i and j are synchronized (by which we mean that they represent the same rotation), we have $q_i = \pm q_j$ and therefore $q_{ij} = \pm q_i^* q_i = [\pm 1, \mathbf{0}_3]$. Hence we wish for any Lyapunov function for this system to have a zero value when $r_{ij} = \pm 1$ and so we define our Lyapunov function candidate as follows.

$$V = \frac{1}{2}\sum_{i=1}^{n}\sum_{j\in N_i}(1-r_{ij}^2) \tag{3}$$

B. Proof of stability

1) Expressing $\dot{r}_{ij}$: Our aim now is to show that the derivative of V with respect to time is less than or equal to zero for all trajectories of our system. Thus we calculate the derivative of the Lyapunov function (3) along the trajectories of the rotation control system (1) as

$$\dot{V} = -\sum_{i=1}^{n}\sum_{j\in N_i}(\dot{r}_{ij} r_{ij}) \tag{4}$$

From here, in order to find an expression for $\dot{r}_{ij}$, we will need to evaluate $\dot{q}_{ij}$. This is done simply using the product rule: $\dot{q}_{ij} = \dot{(q_i^*)}q_j + q_i^*\dot{q}_j$. The only unfamiliar term is the derivative of the conjugate, which we find by deriving it from the following identity:

$$q_i^* q_i = 1$$

By differentiating both sides, one can obtain

$$\dot{(q_i^*)}q_i + q_i^*\dot{q}_i = 0$$

Substituting the control law (1) for $\dot{q}_i$, one finds

$$\dot{(q_i^*)}q_i = -q_i^* q_i \sum_{j\in N_i} f(q_i^* q_j)$$

Finally, by right-multiplying both sides by q_i^* and eliminating $q_i q_i^* = 1$, we have

$$\dot{(q_i^*)} = -\left(\sum_{j\in N_i} f(q_i^* q_j)\right) q_i^* \tag{5}$$

From here we will find an expression for $\dot{r}_{ij}$ in terms of r and $\mathbf{u}$ components. Let $k \in N_i$ and $h \in N_j$. Then

$$\begin{aligned}-f(q_i^* q_k)q_i^* q_j &= [0, r_{ik}\mathbf{u}_{ik}][r_{ij}, \mathbf{u}_{ij}] \\ &= [r_{ik}\langle \mathbf{u}_{ik}, \mathbf{u}_{ij}\rangle, -r_{ik}\mathbf{u}_{ik} \times \mathbf{u}_{ij} - r_{ij}r_{ik}\mathbf{u}_{ik}]\end{aligned} \tag{6}$$

and

$$\begin{aligned}q_i^* q_j f(q_j^* q_h) &= [r_{ij}, \mathbf{u}_{ij}][0, r_{jh}\mathbf{u}_{jh}] \\ &= [-r_{jh}\langle \mathbf{u}_{jh}, \mathbf{u}_{ij}\rangle, r_{jh}\mathbf{u}_{ij} \times \mathbf{u}_{jh} + r_{ij}r_{jh}\mathbf{u}_{jh}]\end{aligned} \tag{7}$$

We use this information to find $\dot{r}_{ij}$. Note that $r_{ij} = \text{Re }(q_{ij})$ and thus $\dot{r}_{ij} = \text{Re }(\dot{q}_{ij})$. Now, we are ready to find an expression for $\dot{r}_{ij}$ by substituting in our expressions for $\dot{q}_i^*$ from (5) and $\dot{q}_j$ from (1).

$$\begin{aligned}\dot{r}_{ij} &= \text{Re }(\dot{q}_{ij}) \\ &= \text{Re }\left(-(\sum_{k\in N_i} f(q_{ik}))q_{ij} + q_{ij}(\sum_{h\in N_j} f(q_{jh}))\right) \\ &= \text{Re }(\sum_{k\in N_i} -f(q_{ik})q_{ij}) + \text{Re }(\sum_{h\in N_j} q_{ij}f(q_{jh}))\end{aligned} \tag{8}$$

Finally, by substituting in our results from (6) and (7), we obtain

$$\dot{r}_{ij} = (\sum_{k\in N_i} r_{ik}\langle \mathbf{u}_{ik}, \mathbf{u}_{ij}\rangle) + (\sum_{h\in N_j} -r_{jh}\langle \mathbf{u}_{jh}, \mathbf{u}_{ij}\rangle) \tag{9}$$

2) Showing $\dot{V} \le 0$: Next, we substitute the expression from (9) into our formula for $\dot{V}$ from (4) and simplify as follows.

$$\begin{aligned}\dot{V} &= -\sum_{i=1}^{n}\sum_{j\in N_i} r_{ij}\Bigg((\sum_{k\in N_i} r_{ik}\langle \mathbf{u}_{ik}, \mathbf{u}_{ij}\rangle) \\ &\quad + (\sum_{h\in N_j} -r_{jh}\langle \mathbf{u}_{jh}, \mathbf{u}_{ij}\rangle)\Bigg) \\ &= -\sum_{i=1}^{n}\sum_{j\in N_i}\sum_{k\in N_i} r_{ij}r_{ik}\langle \mathbf{u}_{ik}, \mathbf{u}_{ij}\rangle \\ &\quad - \sum_{i=1}^{n}\sum_{j\in N_i}\sum_{h\in N_j} -r_{ij}r_{jh}\langle \mathbf{u}_{jh}, \mathbf{u}_{ij}\rangle \\ &= -\sum_{i=1}^{n}\sum_{j\in N_i}\sum_{k\in N_i} r_{ij}r_{ik}\langle \mathbf{u}_{ik}, \mathbf{u}_{ij}\rangle \\ &\quad - \sum_{i=1}^{n}\sum_{j\in N_i}\sum_{h\in N_j} r_{ji}r_{jh}\langle \mathbf{u}_{ji}, \mathbf{u}_{jh}\rangle\end{aligned} \tag{10}$$

From this step, we use the fact that $j \in N_i \Leftrightarrow i \in N_j$, and we rename the subscript h in the second triple sum to k.

$$\begin{aligned}\dot{V} &= -\sum_{i=1}^{n}\sum_{j\in N_i}\sum_{k\in N_i} r_{ij}r_{ik}\langle \mathbf{u}_{ik}, \mathbf{u}_{ij}\rangle \\ &\quad - \sum_{j=1}^{n}\sum_{i\in N_j}\sum_{k\in N_j} r_{ji}r_{jk}\langle \mathbf{u}_{ji}, \mathbf{u}_{jk}\rangle\end{aligned} \tag{11}$$

By further swapping the names of the subscripts i and j in the second triple sum term, we find

$$\dot{V} = -2\sum_{i=1}^{n}\sum_{j\in N_i}\sum_{k\in N_i} r_{ij}r_{ik}\langle \mathbf{u}_{ik}, \mathbf{u}_{ij}\rangle \tag{12}$$

Using the associative and distributive properties of the inner product, we can also write this as

$$\begin{aligned}\dot{V} &= -2\sum_{i=1}^{n}\sum_{j\in N_i}\sum_{k\in N_i} \langle r_{ij}\mathbf{u}_{ij}, r_{ik}\mathbf{u}_{ik}\rangle \\ &= -2\sum_{i=1}^{n}\sum_{j\in N_i} \langle r_{ij}\mathbf{u}_{ij}, \sum_{k\in N_i} r_{ik}\mathbf{u}_{ik}\rangle \\ &= -2\sum_{i=1}^{n}\left\langle \sum_{j\in N_i} r_{ij}\mathbf{u}_{ij}, \sum_{k\in N_i} r_{ik}\mathbf{u}_{ik}\right\rangle\end{aligned} \tag{13}$$

By noticing that j and k are mere choices of subscript, we obtain

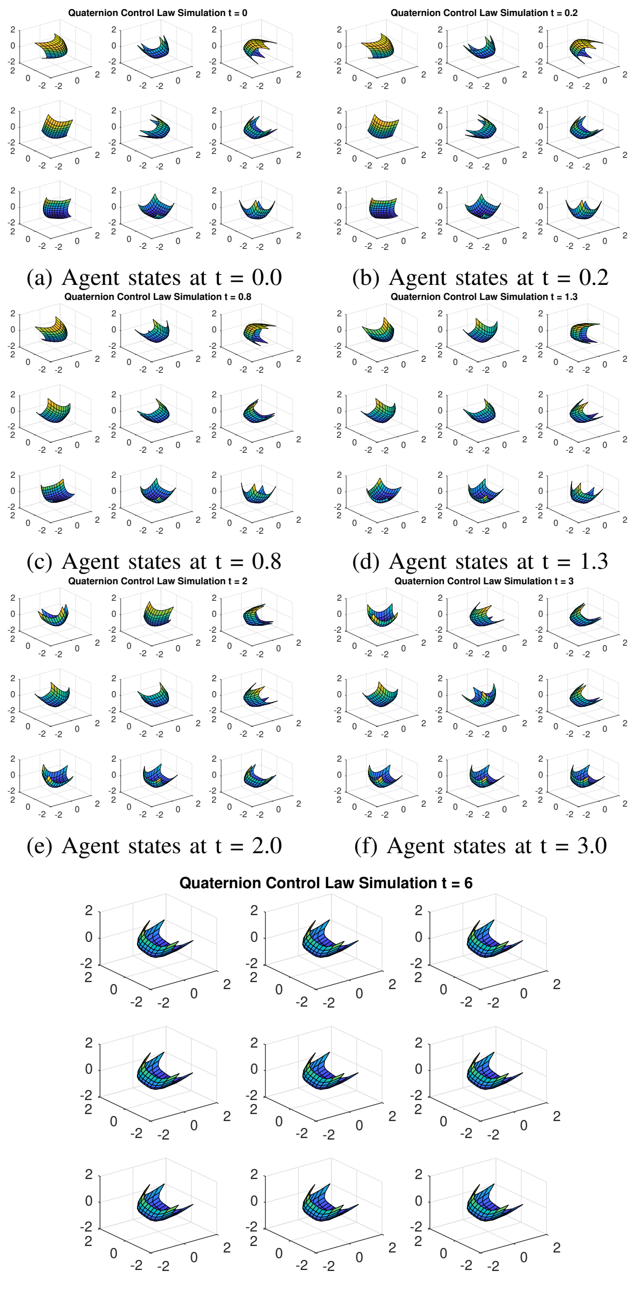
$$\dot{V} = -2\sum_{i=1}^{n} || \sum_{j\in N_i} r_{ij}\mathbf{u}_{ij}||^2 \tag{14}$$

Thus, since for any i, $||\sum_{j\in N_i} r_{ij}\mathbf{u}_{ij}||^2 \ge 0$ we obtain $\dot{V} \le 0$ with equality if and only if for all $i = 1, \cdots, n$ we have $||\sum_{j\in N_i} r_{ij}\mathbf{u}_{ij}||^2 = 0$. We can also express this equality condition as follows: $\dot{V} = 0$ if and only if for all $i = 1, \cdots, n$ we have $\sum_{j\in N_i} r_{ij}\mathbf{u}_{ij} = \mathbf{0}$. For the desired equilibrium at which $r_{ij}^2 = 1$ and $\mathbf{u}_{ij} = 0$, or equivalently $q_{ij} = \pm q_I$, for $\forall(i, j) \in E$, it is locally stable as the global minimum of the Lyapunov function (at which V takes value zero). Therefore, the proposed control law guarantees the local convergence of the attitude synchronization control for the multi-agent systems.

Currently we are unable to characterize the general properties of other undesired equilibrium points (i.e., those which do not correspond to $q_{ij} = \pm q_I$), and the region of attraction of the desired equilibrium points. We have performed several numerical simulations which indicated a large region of attraction for attitude synchronization. For connected and undirected communication graphs (not necessarily complete graphs), and randomly generated initial attitude for each agent, attitude synchronization can be observed at least in simulations. The next section will show such a numerical example demonstrating the controller performance. A special case on a 2-agent attitude control system and the equilibrium analysis will be discussed in Section V for more understandings. The general case of equilibrium analysis will be in our future research.

IV. Simulation Results

To demonstrate the control law proposed, some numerical simulations were performed using MATLAB. Nine agents were constructed with a randomly generated unit quaternion used as the description of their initial orientation. Following this, a random connection graph (shown in Fig. 2) was made between the agents, and the control law was simulated using MATLAB's solver ode45. To demonstrate how the control law works, a number of agents' attitude states at several different times are captured in Fig. 1. As shown in Fig. 1(g), eventually all agents reach attitude synchronization. To gain a deeper understanding of the Lyapunov function associated with the control law and its proof, Fig. 3 and Fig. 4 show the value and derivative of the Lyapunov function of the system over time.



(a) Agent states at t = 0.0

(b) Agent states at t = 0.2

(c) Agent states at t = 0.8

(d) Agent states at t = 1.3

(e) Agent states at t = 2.0

(f) Agent states at t = 3.0

(g) Agent states at t = 6.0

Fig. 1: The attitude states of a nine-agent system at times $t = 0, 0.2, 0.8, 1.3, 2.0, 3.0, 6.0$ after being randomly generated and having the control law proposed in this paper applied.

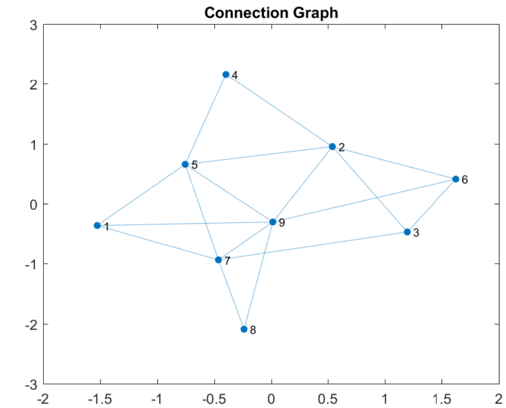


Fig. 2: The undirected interaction graph for the nine-agent system evolving as shown in Fig. 1.

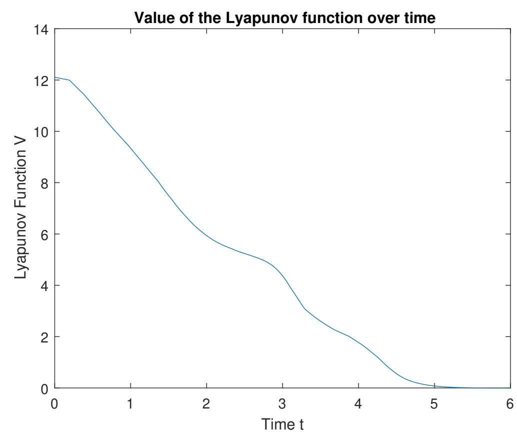


Fig. 3: The value of the Lyapunov function over time for the multi-agent system shown evolving in Fig. 1.

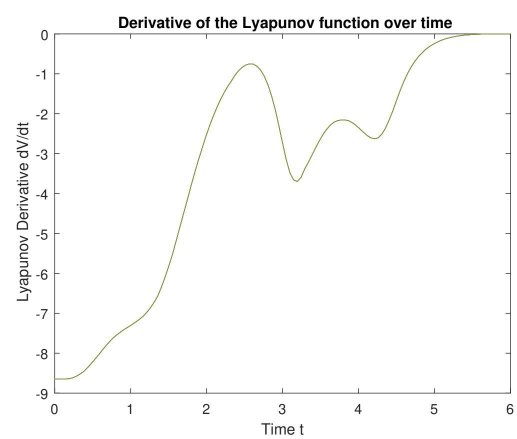


Fig. 4: The derivative of the Lyapunov function over time for the multi-agent system shown evolving in Fig. 1.

V. The Two-Agent Case

A. Equilibrium Points

To give an intuition for how the control law acts on a system of agents, we analyze the case of a two-agent system. Then, following our analysis from Section III.B, we have the Lyapunov function

$$\begin{aligned} V &= \frac{1}{2}\sum_{i=1}^{2}\sum_{j\neq i}(1-r_{ij}^2) \\ &= \frac{1}{2}\left((1-r_{12}^2)+(1-r_{21}^2)\right) \\ &= 1-r_{12}^2 \end{aligned} \tag{15}$$

and its derivative

$$\begin{aligned} \dot{V} &= -2\sum_{i=1}^{2}||\sum_{j\neq i} r_{ij}\mathbf{u}_{ij}||^2 \\ &= -2\left(||r_{12}\mathbf{u}_{12}||^2+||r_{21}\mathbf{u}_{21}||^2\right) \\ &= -4||r_{12}\mathbf{u}_{12}||^2 \leq 0 \end{aligned} \tag{16}$$

Thus we know that at every equilibrium of the system $||r_{12}\mathbf{u}_{12}||^2 = 0$, and so $r_{12}||\mathbf{u}_{12}|| = 0$. At the desired equilibrium, $r_{12}^2 = 1$ and $||\mathbf{u}_{12}|| = 0$. The only other equilibrium possible given the derivative of the Lyapunov function occurs where $||\mathbf{u}_{12}|| = 1$ and $r_{12} = 0$. We can understand what this means in terms of the states of agents $1, 2$ by recalling that $r_{12} = cos(\theta_{12}/2)$ where θ_{12} is the angle of the rotation of agent 2 with respect to agent 1. Thus, at the undesired equilibrium, we have

$$\begin{aligned} cos(\theta_{12}/2) &= 0 \\ \theta_{12}/2 &= \pi/2 + n\pi \text{ for some } n\in\mathbb{Z} \\ \theta_{12} &= \pi + n(2\pi) \\ \theta_{12} &= \pm\pi \end{aligned} \tag{17}$$

As shown in (17), the undesired equilibrium points of the system occur exclusively where the relative angle of rotation between the two agents is $\pm\pi$.

B. Favoring $q_{12} = \pm q_I$

Using the two-agent case, we can also gain intuition for how the q_1 converges to q_2 or $-q_2$ depending on which state is closer. Considered as vectors, the angle between two unit quaternions q_1, q_2 can be found by $\langle q_1, q_2\rangle$. We will look at how the control law acts on the agents' quaternion states in two cases. The first case we consider is where $\langle q_1, q_2\rangle > \langle q_1, -q_2\rangle$. That is, q_1 is closer to q_2 than to $-q_2$ in terms of the angle between the quaternions. We observe that $r_{12} = \langle q_1, q_2\rangle$ and so we are describing here a situation where $r_{12} > 0$. This equivalently means that the Lyapunov function $V < 1$, and thus $\dot{V} < 0$ for all trajectories until $r_{ij}^2 = 1$. This tells us that $r_{12} > 0$ remains true until equilibrium is attained, and so we would have $r_{ij} = 1 \Leftrightarrow q_i = q_j$ at the equilibrium state. Likewise, in the case where q_1 is closer to $-q_2$ than to q_2 in terms of the angle, we would have $r_{12} < 0$. Following the same line of reasoning as before, we find that the equilibrium achieved by the system under these conditions occurs where $r_{ij} = -1 \Leftrightarrow q_i = -q_j$.

VI. Conclusion

This paper has proposed a control law for reaching multi-agent's attitude synchronization with agents' states represented using unit quaternions. It has been shown that the attitude control system with the proposed control law is Lyapunov stable with respect to the desired equilibrium point where each relative agent state reaches agreement. Finally, simulation results on a nine-agent system have been provided to demonstrate and justify the proposed control law, and the two-agent system is analyzed to gain an intuition behind the control law governing the system. Further extensions of this work could include investigating the effect of switching connection graph topology, directed graphs, and how the control law could be adapted to be used in formation control.

References

[1] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems*, vol. 27, no. 2, pp. 71–82, 2007.

[2] S. Knorn, Z. Chen, and R. H. Middleton, "Overview: Collective control of multiagent systems," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 4, pp. 334–347, 2016.

[3] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[4] J. Stuelpnagel, "On the parametrization of the three-dimensional rotation group," *SIAM review*, vol. 6, no. 4, pp. 422–430, 1964.

[5] S. P. Bhat and D. S. Bernstein, "A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon," *Systems & Control Letters*, vol. 39, no. 1, pp. 63–70, 2000.

[6] J. Thunberg, W. Song, E. Montijano, Y. Hong, and X. Hu, "Distributed attitude synchronization control of multi-agent systems with switching topologies," *Automatica*, vol. 50, no. 3, pp. 832–840, 2014.

[7] L. Scardovi, A. Sarlette, and R. Sepulchre, "Synchronization and balancing on the n-torus," *Systems & Control Letters*, vol. 56, no. 5, pp. 335–341, 2007.

[8] P. O. Pereira and D. V. Dimarogonas, "Family of controllers for attitude synchronization in s 2," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pp. 6761–6766, IEEE, 2015.

[9] C. Lageman and Z. Sun, "Consensus on spheres: Convergence analysis and perturbation theory," in *Proc. of the 2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 19–24, IEEE, 2016.

[10] A. Tayebi, "Unit quaternion-based output feedback for the attitude tracking problem," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1516–1520, 2008.

[11] C. G. Mayhew, R. G. Sanfelice, and A. R. Teel, "On quaternion-based attitude control and the unwinding phenomenon," in *Proc. of the 2011 American Control Conference (ACC)*, pp. 299–304, IEEE, 2011.

[12] C. Godsil and G. F. Royle, *Algebraic graph theory*, vol. 207. Springer Science & Business Media, 2013.

[13] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[14] B. Noble, J. W. Daniel, *et al.*, *Applied linear algebra*, vol. 3. Prentice-Hall New Jersey, 1988.

[15] J. B. Kuipers *et al.*, *Quaternions and rotation sequences*, vol. 66. Princeton university press Princeton, 1999.

[16] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on automatic control*, vol. 48, no. 6, pp. 988–1001, 2003.

[17] M.-J. Kim, M.-S. Kim, and S. Y. Shin, "A compact differential formula for the first derivative of a unit quaternion curve," *Journal of Visualization and Computer Animation*, vol. 7, no. 1, pp. 43–57, 1996.