I. INTRODUCTION

Multiagent coordination has been an active research area over the past few decades. Many aspects of multiagent coordination have been explored, and several centralized and decentralized multiagent control approaches already exist. In spite of vast amount of existing research on multiagent coordination, scalability, maneuverability, safety, resilience, and optimality of group coordination are still very important issues for exploration and study. The goal of this article is to address these important problems in a formal and algorithmic way through integrating the principles of continuum mechanics, A* search method, and classic optimal control approach.

A. Related Work

Consensus and containment control are two available decentralized multiagent coordination approaches. Multiagent consensus has found numerous applications, such as flight formation control [1], multiagent surveillance [2], and air traffic control [3]. Consensus control of homogeneous and heterogeneous multiagent systems [4] was studied in the past. Multiagent consensus under fixed [5] and switching [6], [7] communication topologies has been widely investigated by the researchers over the past two decades. Researchers have also investigated multiagent consensus in the presence of actuation failure [9], [10], sensor failure [11], and adversarial agents [12]. Sakurama [13] presents a unified solution for decentralized multiagent coordination problems integrating translation, rotation, reflection, and scale.

Containment control is a decentralized leader–follower multiagent coordination approach, in which the desired coordination is defined by leaders and acquired by followers through local communication. Early work studied stability and convergence of multiagent containment protocol in [14] and [15], under fixed [16] or switching [17] communication topologies, as well as multiagent containment in the presence of fixed [18] and time-varying [19] time delays. Resilient containment control is studied in the presence of actuation failure [20], sensor failure [21], and adversarial agents [22]. In addition, researchers investigated the problems of finite-time [23] and fixed-time [24] containment control of multiagent systems in the past.

B. Contributions

This article studies the problem of continuum deformation optimization of a multiquadcopter system (MQS) in an obstacle-laden environment, in which quadcopters are considered as a finite number of particles of a deformable body, and their desired coordination (transformation) is defined by a homogeneous transformation. Homogeneous transformation is an affine transformation with a Jacobian matrix that is nonsingular at all times $t$. Therefore, an $n$-dimensional ($n = 1, 2, 3$) continuum (deformable body) remains an $n$-dimensional deformable body under a homogeneous transformation coordination. Because homogeneous deformation is a linear transformation, an $n$-dimensional
(\(n = 1, 2, 3\)) homogeneous deformation coordination can be defined as a decentralized leader–follower coordination problem with \(n + 1\) leaders positioned at vertices of a virtual \(n\)-dimensional simplex and followers distributed inside the convex hull, defined by leaders.

This article integrates the principles of continuum mechanics with search and optimization methods to develop a three-layer planning approach for safe coordination of an MQS in geometrically constrained motion spaces (see Fig. 1). Without loss of generality, we treat quadcopters as particles of a virtual deformable triangle (2-D simplex) enclosing all follower quadcopters, where vertices of the triangle are defined by leaders’ desired positions at all times. In the first layer, the deformable triangle is contained by a rigid containment ball (sphere); the A* search is applied to determine the intermediate waypoints of the containment ball such that the MQS travel distance is minimized for given initial and final positions of the containment ball. The second layer uses eigendecomposition of the homogeneous transformation coordination to accomplish the following tasks: 1) assure interagent collision avoidance and quadcopter containment through assigning lower and upper bounds on the eigenvalues of the Jacobian matrix of the homogeneous transformation and 2) determine the intermediate configurations of the leading triangle by assigning the intermediate positions of the leader quadcopters. In the third layer, leaders’ desired trajectories, connecting consecutive configurations of the leader quadcopters, are assigned by solving a constrained optimal control problem.

The rest of this article is organized as follows. Preliminary notions, including graph theory definitions and position notations, are presented in Section II. Problem statement is presented in Section III and followed by continuum deformation coordination planning developed in Section IV. We review the existing approach for continuum deformation acquisition through local communication in Section V. Simulation results are presented in Section VI. Finally, Section VII concludes this article.

II. PRELIMINARIES

A. Graph Theory Notions

We consider the group coordination of a quadcopter team consisting of \(N\) quadcopters in an obstacle-laden environment. Communication among quadcopters is defined by graph \(G(\mathcal{V}, \mathcal{E})\) with node set \(\mathcal{V} = \{1, \ldots, N\}\), defining the index numbers of the quadcopters, and edge set \(\mathcal{E} \subset \mathcal{V} \times \mathcal{V}\). In-neighbors of quadcopter \(i \in \mathcal{V}\) are defined by set \(\mathcal{N}_i = \{j : (j, i) \in \mathcal{E}\}\). Quadcopters are treated as particles of a deformable triangle, which is called leading triangle. Deformation of the leading triangle is defined by three leaders identified by set \(\mathcal{V}_L = \{1, 2, 3\} \subset \mathcal{V}\). The remaining quadcopters are followers identified by set \(\mathcal{V}_F = \mathcal{V} \setminus \mathcal{V}_L = \{4, \ldots, N\}\) and all contained by the leading triangle. Note that leaders move independently; thus, \(\mathcal{N}_i = \emptyset, \text{if } i \in \mathcal{V}_L\). On the other hand, every follower quadcopter updates its own position through local communication, based on positions of three in-neighbor quadcopters; therefore, graph \(G(\mathcal{V}, \mathcal{E})\) is defined such that the following condition holds:

\[
\bigwedge_{i \in \mathcal{V}_F} (|\mathcal{N}_i| = 3),
\]

where “\(\bigwedge\)” means “include all,” i.e., cardinality of set \(\mathcal{N}_i\) (\(|\mathcal{N}_i| = 3\)) for every \(i \in \mathcal{V}_F\).

Fig. 2 shows an example configuration of the leading triangle containing eight quadcopters, with three leaders and five followers. Therefore, \(\mathcal{V}_L = \{1, 2, 3\}\) and \(\mathcal{V}_F = \{4, \ldots, 8\}\). As shown in Fig. 2, every follower quadcopter communicates with three in-neighbor quadcopters, where in-neighbors of every individual follower quadcopters are listed in Table I.

B. Homogeneous Transformation Coordination

For every quadcopter \(i \in \mathcal{V}\), we define the material position \(p_{0i} = \begin{bmatrix} x_{i,0} & y_{i,0} & 0 \end{bmatrix}^T\) and the global desired position \(p_{d}(t) = \begin{bmatrix} x_{i,HT} & y_{i,HT} & z_{i,HT} \end{bmatrix}^T\) that are both expressed with respect to an inertial coordinate system. Note that the material configuration of the quadcopter team is distributed in the plane \(z = 0\) [see Fig. 2(a)]. The global desired position of

---

**TABLE I**

| In-neighbors |  | \(\mathbf{\Omega}_2 = \begin{bmatrix} \alpha_{1,1} & \alpha_{2,2} & \alpha_{3,3} \end{bmatrix}^T\) | Communication weights |
|-------------|----------------|----------------------------------|------------------------|
| \(i\) | \(f_1\) | \(f_2\) | \(f_3\) | \(w_{f_1,i}\) | \(w_{f_2,i}\) | \(w_{f_3,i}\) |
| 4 | 1 | 7 | 8 | \([0.82 0.11 0.07]^T\) | 0.55 | 0.15 | 0.30 |
| 5 | 2 | 6 | 8 | \([0.17 0.69 0.14]^T\) | 0.60 | 0.15 | 0.25 |
| 6 | 3 | 5 | 7 | \([0.18 0.15 0.67]^T\) | 0.60 | 0.15 | 0.25 |
| 7 | 4 | 6 | 8 | \([0.60 0.19 0.22]^T\) | 0.40 | 0.20 | 0.40 |
| 8 | 4 | 5 | 7 | \([0.59 0.28 0.13]^T\) | 0.45 | 0.25 | 0.30 |
quadcopter \( i \in \mathcal{V} \) is defined by
\[
\mathbf{p}_i(t) = \mathbf{Q}(t)\mathbf{p}_{i,0} + \mathbf{d}(t), \quad t \in [t_s, t_a] \tag{2}
\]
where \( t_s \) and \( t_a \) denote the initial time and the final time, respectively, \( \mathbf{Q}(t) \in \mathbb{R}^{3 \times 3} \) is a nonsingular matrix at any time \( t \in [t_s, t_a] \), and \( \mathbf{d}(t) = [d_x(t) \; d_y(t) \; d_z(t)]^T \in \mathbb{R}^{3 \times 1} \) is the rigid-body displacement vector. The continuum deformation coordination defined by (2) is called homogeneous transformation in continuum mechanics [25].

Assumption 1 This article assumes that \( \mathbf{Q}(t_s) = \mathbf{I}_3 \). Therefore, initial and material positions of quadcopter \( i \in \mathcal{V} \) are related by
\[
\mathbf{p}_i(t_s) = \mathbf{p}_{i,0} + \mathbf{d}_s
\tag{3}
\]
where \( \mathbf{d}_s = \mathbf{d}(t_s) \).

For the MQS formation in Fig. 2, the initial configuration, shown in Fig. 2(b), is obtained by the rigid-body translation of the material configuration, shown in Fig. 2(a), with rigid-body displacement vector \( \mathbf{d}_s = [2250 \; 250 \; 50]^T \).

C. Leader–Follower Homogeneous Transformation Coordination

Because homogeneous transformation (2) is affine, the global desired position of quadcopter \( i \in \mathcal{V} \), denoted by \( \mathbf{p}_i(t) \), can be expressed as the convex combination of the leader quadcopters’ positions by
\[
\mathbf{p}_i(t) = \begin{bmatrix} \mathbf{p}_1(t) & \mathbf{p}_2(t) & \mathbf{p}_3(t) \end{bmatrix} \mathbf{\Omega}_2(\mathbf{p}_{1,0}, \mathbf{p}_{2,0}, \mathbf{p}_{3,0}, \mathbf{p}_{i,0}) = \sum_{j=1}^{3} \alpha_{i,j} \mathbf{p}_j(t)
\tag{4}
\]
where \( \mathbf{\Omega}_2(\mathbf{p}_{1,0}, \mathbf{p}_{2,0}, \mathbf{p}_{3,0}, \mathbf{p}_{i,0}) = \begin{bmatrix} \alpha_{i,1} & \alpha_{i,2} & \alpha_{i,3} \end{bmatrix} \mathbf{I}_3 \in \mathbb{R}^{3 \times 1} \) is obtained based on material positions of leaders 1–3, as well as material position of quadcopter \( i \in \mathcal{V} \) by
\[
\mathbf{\Omega}_2(\mathbf{p}_{1,0}, \mathbf{p}_{2,0}, \mathbf{p}_{3,0}, \mathbf{p}_{i,0}) = \begin{bmatrix} x_{1,0} & x_{2,0} & x_{3,0} \\ y_{1,0} & y_{2,0} & y_{3,0} \\ 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_{i,0} \\ y_{i,0} \\ 1 \end{bmatrix}.
\tag{5}
\]

Note that \( \mathbf{\Omega}_2(\mathbf{p}_{1,0}, \mathbf{p}_{2,0}, \mathbf{p}_{3,0}, \mathbf{p}_{i,0}) \) exists, if \( \mathbf{p}_{1,0}, \mathbf{p}_{2,0}, \) and \( \mathbf{p}_{3,0} \) form a triangle. Then,
\[
\begin{bmatrix} x_{1,0} & x_{2,0} & x_{3,0} \\ y_{1,0} & y_{2,0} & y_{3,0} \\ 1 & 1 & 1 \end{bmatrix}
\]
is invertible, and (5) can be converted to
\[
\begin{bmatrix} x_{i,0} \\ y_{i,0} \\ 1 \end{bmatrix} = \begin{bmatrix} x_{1,0} & x_{2,0} & x_{3,0} \\ y_{1,0} & y_{2,0} & y_{3,0} \\ 1 & 1 & 1 \end{bmatrix} \mathbf{\Omega}_2(\mathbf{p}_{1,0}, \mathbf{p}_{2,0}, \mathbf{p}_{3,0}, \mathbf{p}_{i,0}).
\tag{6}
\]

Per (6), sum of the entries of vector \( \mathbf{\Omega}_2(\mathbf{p}_{1,0}, \mathbf{p}_{2,0}, \mathbf{p}_{3,0}, \mathbf{p}_{i,0}) \) is 1 for an arbitrary distribution of vectors \( \mathbf{p}_{1,0}, \mathbf{p}_{2,0}, \mathbf{p}_{3,0}, \) and \( \mathbf{p}_{i,0} \) (i.e. \( \sum_{j=1}^{3} \alpha_{i,j} = 1 \)), when \( \mathbf{p}_{1,0}, \mathbf{p}_{2,0}, \mathbf{p}_{3,0}, \mathbf{p}_{i,0} \) form a triangle. Given the reference positions of the MQS shown in Fig. 2, \( \mathbf{\Omega}_2(\mathbf{p}_{1,0}, \mathbf{p}_{2,0}, \mathbf{p}_{3,0}, \mathbf{p}_{i,0}) \) is computed for every follower \( i \in \mathcal{V}_F = \{4, \ldots, 8\} \) and listed in Table I.

For the purpose of continuum deformation planning, (4) is rewritten as
\[
\mathbf{p}_i(t) = (\mathbf{I}_3 \otimes \mathbf{\Omega}_2^T(\mathbf{p}_{1,0}, \mathbf{p}_{2,0}, \mathbf{p}_{3,0}, \mathbf{p}_{i,0})) \mathbf{y}_{LT}(t).
\tag{7}
\]
Where
\[
\mathbf{y}_{LT}(t) = \text{vec}\left(\begin{bmatrix} \mathbf{p}_1(t) & \mathbf{p}_2(t) & \mathbf{p}_3(t) \end{bmatrix}^T\right) \in \mathbb{R}^{9 \times 1}
\tag{8}
\]
aggregates the components of the leaders’ global desired positions, subscripts “L” and “HT” refer to “Leaders” and “Homogeneous Transformation,” respectively, and “vec” is the matrix vectorization symbol.

D. Position Notations

This article uses the following position notations:
- \( x_{i,0}, y_{i,0} \) components of material position \( \mathbf{p}_{i,0} \);
- \( x_i, y_i, z_i \) components of actual position \( \mathbf{r}_i \);
- \( x_{i,HT}, y_{i,HT}, z_{i,HT} \) components of global desired position \( \mathbf{p}_i \);
- \( \mathbf{Y}_{LT} \) vector aggregating leaders’ global desired position components;
- \( \mathbf{Y}_{HT} \) vector aggregating components of the global desired positions of all quadcopters;
- \( \mathbf{Y}_L \) vector aggregating leaders’ actual position components;
- \( \mathbf{Y}_F \) vector aggregating followers’ actual position components;
- \( \mathbf{Y}_{FD} \) vector aggregating leaders’ local desired position components;
- \( L \) subscript stands for “leaders”;
- \( F \) subscript stands for “followers.”

III. PROBLEM STATEMENT

We treat the MQS as particles of a 2-D deformable body navigating in an obstacle-laden environment, where the global desired position of every quadcopter \( i \in \mathcal{V} \) can be uniquely related to \( \mathbf{y}_{LT}(t) \) over \([t_s, t_a]\) by using Eq. (7). Given the initial time \( t_s \), the main goal of this article is to determine \( \mathbf{y}_{LT}(t) \) and ultimate time \( t_a \) such that the MQS travel distances are minimized, and the following constraints are all satisfied.

1) The area of the leading triangle denoted by \( A_s \) remains constant at any time \( t \in [t_s, t_a] \).
2) Assuming that every quadcopter can be enclosed by a ball of radius \( e \), interagent collision avoidance is guaranteed at any time \( t \in [t_s, t_a] \).
3) While the MQS moves in a 3-D motion space, quadcopters are treated as particles of a 2-D deformable body deforming in the \( xy \) plane.
4) Quadcopters are all contained by rigid ball
\[ S(d(t), r_{\text{max}}) = \left\{ (x, y, z) : (x - d_x)^2 + (y - d_y)^2 + (z - d_z)^2 \leq r_{\text{max}}^2 \right\} \]  
with radius \( r_{\text{max}} \) that is centered at \( d(t) \) at any time \( t \in [t_s, t_a] \).

We can formally express the above four constraints by
\[ \mathbf{y}_{\text{T}}^T(t) \mathbf{P} \mathbf{y}_{\text{T}}(t) - A_i = 0 \quad \forall t \in [t_s, t_a] \]  
(10a) \[ \bigwedge_{i \in \mathcal{V}} \bigwedge_{j \in \mathcal{V}, j \neq i} \| \mathbf{r}_i(t) - \mathbf{r}_j(t) \| \geq 2 \varepsilon \quad \forall t \in [t_s, t_a] \]  
(10b) \[ \left( z_i, HT(t) = d_i(t) \right) \quad \forall t \in [t_s, t_a] \]  
(10c) \[ \left( \mathbf{y}_i, HT(t), y_i, HT(t), d_i(t) \in S(d(t), r_{\text{max}}) \right) \forall t \in [t_s, t_a] \]  
(10d)

where
\[ \mathbf{P} = \mathbf{O}^T \mathbf{Q} \mathbf{O} \]  
(11a) \[ \mathbf{O} = \begin{bmatrix} I_6 & 0_{6 \times 3} \end{bmatrix} \]  
(11b) \[ \mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \]  
(11c)

To accomplish the goal of this article, we integrate: 1) \( A^* \) search; 2) eigendecomposition; and 3) optimal control to assign leaders’ optimal trajectories ensuring safety requirements (10a)–(10d), by performing the following sequential steps.

Step 1: Assigning intermediate locations of the containment ball: Given initial and final positions of the center of the containment ball, denoted by \( \mathbf{d}_s = d(t_s) = \mathbf{d}_0 \) and \( \mathbf{d}_u = d(t_u) = \mathbf{d}_u \), and obstacle geometries, we apply the \( A^* \) search method to determine the intermediate positions of the center of the containment ball \( S \), denoted by \( \mathbf{d}_i, \ldots, \mathbf{d}_{n_i-1} \), such that: 1) the travel distance between the initial and final configurations of the MQS is minimized, and 2) the containment ball does not collide with the obstacles that are arbitrarily distributed in the motion space.

Step 2: Assigning leaders’ intermediate configurations: By knowing \( \mathbf{d}_1, \ldots, \mathbf{d}_{n_i-1} \), we define
\[ \beta_k = \frac{\sum_{j=0}^{k} \| \mathbf{d}_j - \mathbf{d}_0 \|}{\sum_{j=0}^{n_i} \| \mathbf{d}_j - \mathbf{d}_0 \|} , \quad k = 0, 1, \ldots, n_i \]  
(12) \[ t_k(t_u) = (1 - \beta_k) t_s + \beta_k t_u , \quad k = 0, 1, \ldots, n_i \]  
(13)

where \( t_k \) is the speed of the center of the containment ball \( S \) reaches the desired intermediate position \( \mathbf{d}_i \). Given \( \mathbf{y}_{\text{T}}(t_s) = \mathbf{y}_{L,b,0} \), \( \mathbf{y}_{\text{T}}(t_u) = \mathbf{y}_{L,b,n_i} \), Section IV-B decomposes the homogeneous deformation coordination to determine the intermediate configurations of the leaders that are denoted by \( \mathbf{y}_{\text{T},l,1}, \ldots, \mathbf{y}_{\text{T},l,n_i-1} \).

Step 3: Assigning leaders’ desired trajectories: By expressing \( \mathbf{d} = \begin{bmatrix} \mathbf{d}_{\mathcal{E},k} & \mathbf{d}_{\mathcal{E},k} \end{bmatrix}^T \) for \( k = 0, 1, \ldots, n_i \), \( z \) components of the leaders’ desired trajectories are the same at any time \( t \in [t_s, t_u] \) and defined by
\[ z_i, HT(t) = \mathbf{d}_{\mathcal{E},k} \left( 1 - \gamma(t, T_k) \right) + \mathbf{d}_{\mathcal{E},k+1} \gamma(t, T_k) , \quad \forall i \in \mathcal{V}_L \]  
(14)

at any time \( t \in [t_k, t_{k+1}] \) for \( k = 0, \ldots, n_i - 1 \), where \( T_k = t_{k+1} - t_k \), and
\[ \gamma(t, T_k) = 6 \left( \frac{t - t_k}{T_k} \right)^5 - 15 \left( \frac{t - t_k}{T_k} \right)^4 + 10 \left( \frac{t - t_k}{T_k} \right)^3 \]  
(15)
for \( t \in [t_k, t_{k+1}] \). Note that \( \gamma(t_k, T_k) = 0, \gamma(t_{k+1}, T_k) = 1, \gamma(t_k, T_k) = \gamma(t_{k+1}, T_k) = 0 \), and \( \gamma(t_k, T_k) = \gamma(t_{k+1}, T_k) = 0 \).

The \( x \) and \( y \) components of the desired trajectories of leaders are governed by dynamics
\[ \dot{x}_L = A_L x_L + B_L a_L \]  
(16)
where \( a_L \in \mathbb{R}^{9 \times 1} \) is the input vector,
\[ x_L(t) = (I_2 \otimes \mathbf{O}) \begin{bmatrix} \mathbf{y}_{\text{T},l,HT}(t) \\ \mathbf{y}_{\text{T},HT}(t) \end{bmatrix} \]  
(17a) \[ A_L = \begin{bmatrix} 0_{6 \times 6} & I_6 \\ 0_{6 \times 6} & 0_{6 \times 6} \end{bmatrix} \]  
(17b) \[ B_L = \begin{bmatrix} 0_{6 \times 6} & I_6 \end{bmatrix} \]  
(17c)
\( 0_{6 \times 6} \in \mathbb{R}^{6 \times 6} \) is a zero-entry matrix, and \( I_6 \in \mathbb{R}^{6 \times 6} \) is an identity matrix. Control input \( u_L \in \mathbb{R}^{6 \times 1} \) is optimized by minimizing the cost function
\[ \min J(u_L, t_u) = \frac{1}{2} \sum_{k=0}^{n_i-1} \int_{t_k}^{t_{k+1}} \left( r_{\text{rk}}(t_u) u_L(t) u_L(t) \right) dt \]  
(18)
subject to dynamics (16), safety conditions (10a)–(10d), and boundary conditions
\[ \left( x_L(t_k) = s_{L,k} \right) . \]  
(19)

A desired continuum deformation coordination, planned by the leader quadcopters, is acquired by followers in a decentralized fashion using the protocol developed in [26] and [27]. This protocol is discussed in Section V.

IV. CONTINUUM DEFORMATION PLANNING

The desired configuration of the MQS is defined by (2), where nonsingular matrix \( Q = [Q_{ij}] \in \mathbb{R}^{3 \times 3} \) is given by
\[ Q(t) = \begin{bmatrix} Q_{xx}(t) & 0_{2 \times 1} \\ 0_{1 \times 2} & 1 \end{bmatrix} \quad \forall t \in [t_s, t_u] . \]  
(20)
Note that $Q_{3y}(t) \in \mathbb{R}^{2 \times 2}$ specifies the deformation of the leading triangle, defined by the three leaders. Because $Q_{31} = Q_{32} = Q_{13} = Q_{23} = 0$, the leading triangle lies in the horizontal plane at any time $t \in [t_s, t_u]$, if the z components of desired positions of the leaders are all identical at the initial time $t_s$.

**Remark 1** Equation (2) is used for the purpose of eigendecomposition of homogeneous transformation coordination, safety analysis, and planning of the desired continuum deformation coordination. On the other hand, (7) is used in Section V-A to define the MQS continuum deformation coordination as a decentralized leader–follower problem and ensure the stability of the desired trajectory tracking control.

**Theorem 1** Assume that three leader quadcopters 1–3 remain nonaligned at any time $t \in [t_s, t_u]$. Then, the desired configuration of the leaders at time $t \in [t_s, t_u]$, defined by $y_{LHT}(t)$, is related to the leaders’ material configuration, defined by $\tilde{y}_{LHT,0}$, by

$$\tilde{y}_{LHT,0} = \text{vec} \left( \begin{bmatrix} p_{1,0} & \cdots & p_{3,0} \end{bmatrix}^T \right) \in \mathbb{R}^{9 \times 1}$$

and the rigid body displacement vector $d(t)$ by

$$y_{LHT}(t) = D \left( I_3 \otimes Q(t) \right) \Delta y_{LHT,0} + D \left( I_{3 \times 1} \otimes d(t) \right)$$

where $\otimes$ is the Kronecker product symbol, and

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \end{bmatrix} \in \mathbb{R}^{9 \times 9}$$

is an involutory matrix.

In addition, elements of matrix $Q_{3y}(t)$ and rigid-body displacement vector $d(t)$ can be related to $y_{LHT}(t)$ by

$$Q_{11}(t) = E_1 \Gamma O_{LHT}(t)$$

$$Q_{12}(t) = E_2 \Gamma O_{LHT}(t)$$

$$Q_{21}(t) = E_3 \Gamma O_{LHT}(t)$$

$$Q_{22}(t) = E_4 \Gamma O_{LHT}(t)$$

$$d(t) = \begin{bmatrix} E_5 \Gamma O \\ E_6 \end{bmatrix} y_{LHT}(t)$$

at any time $t \in [t_s, t_u]$, where $E_1 = \begin{bmatrix} 1 & 0_{1 \times 5} \end{bmatrix}$, $E_2 = \begin{bmatrix} 0 & 1 & 0_{1 \times 4} \end{bmatrix}$, $E_3 = \begin{bmatrix} 0 & 1_{2 \times 2} & 1 & 0_{1 \times 3} \end{bmatrix}$, $E_4 = \begin{bmatrix} 0_{1 \times 3} & 1 & 0_{1 \times 2} \end{bmatrix}$, $E_5 = \begin{bmatrix} 0 & 2_{2 \times 4} & I_2 \end{bmatrix}$, $E_6 = \begin{bmatrix} 0 \end{bmatrix}$, and

$$\Gamma = \begin{bmatrix} x_{1,0} & y_{1,0} & 0 & 0 & 1 & 0 \\ x_{2,0} & y_{2,0} & 0 & 0 & 1 & 0 \\ x_{3,0} & y_{3,0} & 0 & 0 & 1 & 0 \\ 0 & 0 & x_{1,0} & y_{1,0} & 0 & 1 \\ 0 & 0 & x_{2,0} & y_{2,0} & 0 & 1 \\ 0 & 0 & x_{3,0} & y_{3,0} & 0 & 1 \end{bmatrix}$$

**Proof** Vectors $y_{LHT}(t)$ and $\tilde{y}_{LHT,0}$ can be expressed by $y_{LHT}(t) = D \begin{bmatrix} p_{1,0}^T & p_{2,0}^T & p_{3,0}^T \end{bmatrix}^T$ and $\tilde{y}_{LHT,0} = D \begin{bmatrix} p_{1,0}^T & p_{2,0}^T & p_{3,0}^T \end{bmatrix}^T$, respectively. By provoking (2), we can write

$$\begin{bmatrix} p_{1,0} \\ p_{2,0} \\ p_{3,0} \end{bmatrix} = (I_3 \otimes Q(t)) \begin{bmatrix} p_{1,0} \\ p_{2,0} \\ p_{3,0} \end{bmatrix} + I_{3 \times 1} \otimes d(t)$$

and (24) can be rewritten as follows:

$$Dy_{LHT}(t) = (I_3 \otimes Q(t)) \tilde{y}_{LHT,0} + I_{3 \times 1} \otimes d(t)$$

Because $D$ is involutory, $D = D^{-1}$ and (21) can be obtained by premultiplying $D$ on both the sides of (24). By replacing $p_{1,0}$ and $p_{2,0}$ by $\begin{bmatrix} x_{i,HT}(t) & y_{i,HT}(t) & z_{i,HT}(t) \end{bmatrix}^T$ and $\begin{bmatrix} x_{i,0} & y_{i,0}(t) & 0 \end{bmatrix}^T$ into (2) for every leader $i \in L$, elements of $Q_{3y}(t)$, denoted by $Q_{11}(t), Q_{12}(t), Q_{21}(t),$ and $Q_{22}(t)$, and $x$ and $y$ element of $d(t)$, denoted by $d_x(t)$ and $d_y(t)$, can be related to the $x$ and $y$ components of the leaders’ desired positions by

$$\begin{bmatrix} Q_{11}(t) & Q_{12}(t) & Q_{21}(t) & Q_{22}(t) & d_x(t) & d_y(t) \end{bmatrix}^T = \Gamma O_{LHT}(t)$$

at any time $t \in [t_s, t_u]$, where

$$Oy_{LHT} = \begin{bmatrix} x_{1,HT} & x_{2,HT} & x_{3,HT} & y_{1,HT} & y_{2,HT} & y_{3,HT} \end{bmatrix}^T$$

Note that matrix $\Gamma$ is nonsingular, if leaders are nonaligned in the reference configuration [26].

Theorem 1 is used in Section IV-A to obtain the final location of the center of the containment ball, denoted by $\tilde{d}_u$, where $\tilde{d}_u$ is one of the inputs of the A* solver (see Algorithm 2). In particular, $\tilde{d}_u = d(t_u)$ is obtained by (23e), if $y_{LHT}(t)$ is substituted by $y_{LHT,n_i} = y_{LHT}(t_u)$ on the right-hand side of (23e). In addition, Section IV-B uses Theorem 1 to assign the intermediate formations of the leader team.

A. A* Search Planning

A* search method is used to safely plan the coordination of the containment ball $S$ by optimizing the intermediate locations of its center, denoted by $\tilde{d}_i$ through $\tilde{d}_i$, for given $\tilde{d}_i$ and $\tilde{d}_i$, where obstacles are known in the motion space. We first develop an algorithm for collision avoidance of the MQS with obstacles in Section IV-A1. This algorithm
is used by the A* optimizer to determine \( \tilde{d}_1 \) through \( \tilde{d}_{n_i-1} \), as described in Section IV-A2.

**DEFINITION 1** Let \( i - j - k - l \) be an arbitrary tetrahedron whose vertices are positioned as \( \mathbf{p}_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T \), \( \mathbf{p}_j = \begin{bmatrix} x_j & y_j & z_j \end{bmatrix}^T \), \( \mathbf{p}_k = \begin{bmatrix} x_k & y_k & z_k \end{bmatrix}^T \), and \( \mathbf{p}_l = \begin{bmatrix} x_l & y_l & z_l \end{bmatrix}^T \) is a 3-D motion space. In addition, \( \mathbf{p}_f = \begin{bmatrix} x_f & y_f & z_f \end{bmatrix}^T \) is the position of an arbitrary point \( f \) in the motion space. Then

\[
\Omega_3(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k, \mathbf{p}_l, \mathbf{p}_f) = \begin{bmatrix} \mathbf{p}_i & \mathbf{p}_j & \mathbf{p}_k & \mathbf{p}_l & \mathbf{p}_f \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{p}_f^T \end{bmatrix} \tag{26}
\]

is a finite vector with the entries summing up to 1 [26].

The vector function \( \Omega_3 \) is used in Section IV-A1 to obtain collision avoidance conditions.

1) **Obstacle Collision Avoidance**: We enclose obstacles by a finite number of polytopes and identify both the obstacle and the enclosing polytope by the same identification number defined by set \( \mathcal{H} = \{1, \ldots, M\} \), where \( \mathcal{P} = \bigcup_{j=1}^{M} \mathcal{P}_j \) defines vertices of polytopes enclosing obstacles, and \( \mathcal{P}_j \) is a finite set defining the identification numbers of vertices of polytope \( j \in \mathcal{H} \) containing the \( j \)th obstacle in the motion space. Polytope \( \mathcal{P}_j \) is made of \( m_j \) distinct tetrahedral cells, where \( \mathcal{T}_{j,l} \) defines the identification numbers of the nodes of the \( l \)th tetrahedral cell of polytope \( \mathcal{P}_j (l = 1, \ldots, m_j \) and \( j \in \mathcal{H}) \). Therefore, \( \mathcal{P} \) can be expressed as follows:

\[
\mathcal{P} = \bigwedge_{j=1}^{M} \bigwedge_{l=1}^{m_j} \mathcal{T}_{j,l}. \tag{27}
\]

**DEFINITION 2** We say \( \mathbf{d} \) is a valid position for the center of the containment ball \( S \) with radius \( r_{\text{max}} \), if the following two conditions are satisfied:

\[
\bigwedge_{j=1}^{M} \bigwedge_{l=1}^{m_j} \left( (x_{p_l}, y_{p_l}, z_{p_l}) \notin S(\mathbf{d}, r_{\text{max}}) \right) \tag{28a}
\]

\[
\forall \mathbf{r} \in \partial S, \bigwedge_{j=1}^{M} \bigwedge_{l=1}^{m_j} \left( \Omega_3(\mathbf{p}_{v_1}, \mathbf{p}_{v_2}, \mathbf{p}_{v_3}, \mathbf{p}_{v_4}, \mathbf{r}) \not\leq 0 \right) \tag{28b}
\]

where \( \partial S(\mathbf{d}, r_{\text{max}}) \) is the boundary of the containment ball. In (28a), \( p \in \mathcal{T}_{j,l} \) is the index number of one of the nodes of tetrahedron \( \mathcal{T}_{j,l} \) that is positioned at \( (x_{p_l}, y_{p_l}, z_{p_l}) \) for \( j \in \mathcal{H} \) and \( l = 1, \ldots, m_j \). In (28b), \( \mathbf{p}_{v_1}, \mathbf{p}_{v_2}, \mathbf{p}_{v_3}, \) and \( \mathbf{p}_{v_4} \) denote positions of vertices \( v_1, v_2, v_3, \) and \( v_4 \) of tetrahedron \( \mathcal{T}_{j,l} \) for \( j \in \mathcal{H} \) and \( l = 1, \ldots, m_j \). The constraint equation (28a) ensures that vertices of the containment polytopes are all outside the ball \( S \). In addition, condition (28b) requires that the center of the containment ball is outside of all polytopes defined by \( \mathcal{P} \).
Algorithm 1: A* Planning of the MQS Coordination.

1: Get: \( \bar{d}_s \) and \( \bar{d}_u \)
2: Define: Open set \( O = \{ \bar{d}_s \} \), Closed set \( C = \emptyset \), and \( \bar{d}_{\text{best}} = \bar{d}_s \)
3: while \( \bar{d}_{\text{best}} \neq \bar{d}_u \) or \( O \neq \emptyset \) do
4: \( \bar{d}_{\text{best}} \leftarrow \arg\min_{\bar{d} \in O} (g(\bar{d}) + C_H(\bar{d}, \bar{d}_s)) \)
5: Update \( O: O \leftarrow O \setminus \{\bar{d}_{\text{best}}\} \)
6: Update \( C: C \leftarrow C \cup \{\bar{d}_{\text{best}}\} \)
7: Assign \( A(\bar{d}_{\text{best}}) \)
8: \( \mathcal{R}(\bar{d}_{\text{best}}) \leftarrow A(\bar{d}_{\text{best}}) \setminus (A(\bar{d}_{\text{best}}) \cap C) \)
9: for \( \forall \bar{d} \in \mathcal{R}(\bar{d}_{\text{best}}) \) do
10: if \( \bar{d} \in O \) then
11: if \( g(\bar{d}_{\text{best}}) + C_O(\bar{d}_{\text{best}}, \bar{d}) < g(\bar{d}) \) then
12: \( g(\bar{d}) \leftarrow g(\bar{d}_{\text{best}}) + C_O(\bar{d}_{\text{best}}, \bar{d}) \)
13: \( \bar{b}(\bar{d}) \leftarrow \bar{d}_{\text{best}} \)
14: end if
15: end if
16: end for
17: \( O \leftarrow \mathcal{R}(\bar{d}_{\text{best}}) \cup O \)
18: end while

Algorithm 2: Assignment of Optimal Waypoints \( \bar{d}_1, \ldots, \bar{d}_{n-1} \).

1: Get: \( \bar{b}_s = \bar{d}_s, \ldots, \bar{b}_m = \bar{d}_u \)
2: Set: \( i = 0 \)
3: for \( k < i - 1 \) to \( m - 1 \) do
4: if \( \bar{b}_k - \bar{b}_{k-1} \neq \bar{b}_{k+1} - \bar{b}_k \) then
5: \( i \leftarrow i + 1 \)
6: \( \bar{d}_i = \bar{b}_k \)
7: end if
8: end for

Given the initial and final locations of the center of the containment ball \( S \), denoted by \( \bar{d}_s \) and \( \bar{d}_u \), the A* path planning Algorithm 1 is applied to determine optimal intermediate positions \( \bar{b}_s, \ldots, \bar{b}_m \) along the optimal path of the containment ball \( S \) from \( \bar{d}_s \) to \( \bar{d}_u \) in an obstacle-laden environment. More specifically, the A* optimizer generates \( \bar{b}_s, \ldots, \bar{b}_m \), by searching over set \( \mathcal{F} \), where

\[
\bar{b}_k = \bar{d}_s \quad (34a)
\]
\[
\bar{b}_m = \bar{d}_u \quad (34b)
\]
\[
(\bar{b}_k, \bar{b}_{k+1}) \in A(\bar{b}_k) \quad k = 0, \ldots, m - 1. \quad (34c)
\]

The center of the containment ball \( S \) moves along the straight paths obtained by connecting \( \bar{b}_k \) to \( \bar{b}_{k+1} \) for \( k = 0, \ldots, m - 1 \), where \( \bar{b}_0 = \bar{b}_s \). Therefore, \( n_t \) serially connected line segments define the optimal path of the containment ball, where \( n_t \leq m, \bar{d}_0 = \bar{d}_s, \bar{d}_n = \bar{d}_u = \bar{d}_m = \bar{d}_u \), and the \( j \)-th line segment connects \( \bar{d}_{j-1} \) to \( \bar{d}_j \) for \( j = 0, 1, \ldots, n_t - 1 \). Given \( \bar{b}_s, \ldots, \bar{b}_m \), Algorithm 2 is used to determine \( \bar{d}_1, \ldots, \bar{d}_{n-1} \).

B. Intermediate Configuration of the Leading Triangle

Matrix \( Q_{xy}(t) \) can be expressed by

\[
Q_{xy}(t) = R_{xy}(t)U_{xy}(t) \quad (35)
\]

with rotation matrix \( R_{xy}(t) \) and pure deformation matrix \( U_{xy}(t) \) that are defined as follows:

\[
R_{xy}(t) = \begin{bmatrix}
\cos \theta_r & -\sin \theta_r \\
\sin \theta_r & \cos \theta_r
\end{bmatrix} \quad (36a)
\]
\[
U_{xy}(t) = R_D(t)A(t)R_D^T(t) \quad (36b)
\]

where

\[
A(t) = \begin{bmatrix}
\sigma_1(t) & 0 \\
0 & \sigma_2(t)
\end{bmatrix} \quad (37a)
\]
\[
R_D(t) = \begin{bmatrix}
\cos \theta_d & -\sin \theta_d \\
\sin \theta_d & \cos \theta_d
\end{bmatrix} \quad (37b)
\]

Note that \( \theta_r(t) > 0 \) and \( \theta_d(t) > 0 \) are the rotation and shear deformation angles, and \( \sigma_1(t) \) and \( \sigma_2(t) \) are the first and second deformation eigenvalues. Because \( A(t) \) is positive definite and diagonal, matrix \( U_{xy}(t) \) is positive definite at any time \( t \in [t_s, t_u] \) [26].
PROPOSITION 1 Matrix $U_{xy}^m (U_{xy} = (U_{xy})^m)$ can be expressed as

$$U_{xy}^m(t) = \begin{bmatrix} a_m(t) & b_m(t) \\ b_m(t) & a_m(t) \end{bmatrix}$$

with

$$a_m(t) = \sigma_1(t) \cos^2 \theta_d(t) + \sigma_2(t) \sin^2 \theta_d(t)$$
$$b_m(t) = (\sigma_1(t) - \sigma_2(t)) \sin \theta_d(t) \cos \theta_d(t)$$
$$c_m(t) = \sigma_3(t) \sin^2 \theta_d(t) + \sigma_4(t) \cos^2 \theta_d(t).$$

In addition, $\sigma_1$, $\sigma_2$, and $\theta_d$ can be related to $a_m$, $b_m$, and $c_m$ by

$$\sigma_1(t) = \frac{a_m(t) + c_m(t)}{2} + \sqrt{\left(\frac{1}{2} (a_m(t) - c_m(t))\right)^2 + b_m^2(t)}$$
$$\sigma_2(t) = \frac{a_m(t) + c_m(t)}{2} - \sqrt{\left(\frac{1}{2} (a_m(t) - c_m(t))\right)^2 + b_m^2(t)}$$
$$\theta_d(t) = \frac{1}{2} \tan^{-1} \left( \frac{2b_m(t)}{a_m(t) - c_m(t)} \right).$$

PROOF Because $R_D(t)$ is orthogonal at time $t$, $R_D(t)R_D(t) = I_2$. If matrix $U_{xy}^m$ is expressed as

$$U_{xy}^m(t) = R_D(t)A^mR_D(t)$$

for $m = 1, 2, \ldots$, then

$$U_{xy}^{m+1}(t) = R_D(t)AR_D^T(t)R_D(t)A^mR_D(t) = R_D(t)A^{m+1}R_D(t).$$

Since (41) is valid for $m = 0$, (42) ensures that (41) is valid for any $m > 0$. By replacing (37a) and (37b) into (41), elements of matrix $U_{xy}^m(t)$ ($a_m(t)$, $b_m(t)$, $c_m(t)$) are obtained by (39a)–(39c), respectively.

By provoking Proposition 1, matrix $U_{xy}^m = Q_{xy}^mQ_{xy}$ [26] can be expressed in the form of (38) where $m = 2$ and

$$a_2(t) = y_{L,HT}^T(t)O^T \Gamma^T (E_1^T E_1 + E_3^T E_3) \Gamma O_{L,HT}(t)$$
$$b_2(t) = y_{L,HT}^T(t)O^T \Gamma^T (E_1^T E_2 + E_3^T E_4) \Gamma O_{L,HT}(t)$$
$$c_2(t) = y_{L,HT}^T(t)O^T \Gamma^T (E_2^T E_2 + E_4^T E_4) \Gamma O_{L,HT}(t).$$

Therefore, we can determine $\sigma_1(t)$, $\sigma_2(t)$, and $\theta_d(t)$ by replacing $m = 2$, $a_m(t) = a_2(t)$, $b_m(t) = b_2(t)$, and $c_m(t) = c_2(t)$ into (40a), (40b), and (40c) at time $t \in [t_s, t_a]$, respectively. Furthermore, matrix $R_{xy}(t) = QU_{xy}^{-1}$ is related to $y_{L,HT}(t)$ by

$$R_{xy}(t) = \begin{bmatrix} E_1 \Gamma O_{L,HT}(t) & E_2 \Gamma O_{L,HT}(t) \\ E_3 \Gamma O_{L,HT}(t) & E_4 \Gamma O_{L,HT}(t) \end{bmatrix} \times \begin{bmatrix} a_2(t) & b_2(t) \\ b_2(t) & c_2(t) \end{bmatrix}^{\frac{1}{2}}.$$
Intermediate configurations of leaders: We offer a procedure with the following five main steps to determine the intermediate waypoints of the leaders.

Step 1: Given \( y_{L,HT,n} = y_{L,HT}(t_k) \), \( \sigma_{1,n} = \sigma_1(t_k) \), \( \theta_{d,n} = \theta_d(t_k) \), and \( \tau_{n} = \theta_n(t_k) \) are computed using (40a), (40c), and (44), respectively.

Step 2: We compute
\[
\begin{align*}
\sigma_{1,k} &= \beta_k \sigma_{1,0} + (1 - \beta_k) \sigma_{1,n}, \\
\theta_{d,k} &= (1 - \beta_k) \theta_{d,n}, \\
\tau_{k} &= (1 - \beta_k) \tau_{n}
\end{align*}
\]
for \( k = 1, \ldots, n_r - 1 \), where \( \beta_k \) is computed using (12).

Step 3: We compute \( \sigma_{2,k} = \frac{1}{\sigma_{1,k}} \) for \( k = 1, \ldots, n_r - 1 \).

Step 4: Given \( \sigma_{1,k}, \sigma_{2,k}, \) and \( \theta_{d,k} \), matrix \( U_{\sigma,k} = U_{\sigma}(t_k) \) is obtained by (36b) for \( k = 1, \ldots, n_r - 1 \). In addition, matrix \( R_{\sigma,k} = R_{\sigma}(t_k) \) is obtained using (36a) by knowing the rotation angle \( \theta_{d,k} \) for \( k = 1, \ldots, n_r - 1 \).

Step 5: By knowing \( R_{\sigma,k} = R_{\sigma}(t_k) \) and \( U_{\sigma,k} = U_{\sigma}(t_k) \), the Jacobian matrix \( Q_{\sigma,k} = Q_{\sigma}(t_k) \) is obtained using (35). Then, we can use (2) and (8) to obtain \( \hat{y}_{L,HT,k} \) by replacing \( Q_{\sigma,k} = Q_{\sigma}(t_k) \) and \( \hat{a}_k \) for \( k = 1, \ldots, n_r - 1 \).

C. Optimal Control Planning

This section offers an optimal control solution to determine the leaders’ desired trajectories connecting every two consecutive waypoinits \( \hat{y}_{L,HT,k} \) and \( \hat{y}_{L,HT,k+1} \) for \( k = 0, 1, \ldots, n_r - 1 \), where \( z \) components of the leaders are defined by (14), and \( x \) and \( y \) components of the leaders’ desired trajectories are updated by (16).

**Coordination constraint:** Per equality constraint (10a), the area of the leading triangle, given by
\[
A(t) = y_{L,HT}^T(t) \Psi_{y_{L,HT}}(t)
\]
must be equal to constant value \( A_1 \), at any time \( t \in [t_k, t_{k+1}] \). This equality constraint is satisfied, if: 1) \( y_{L,HT}(t) \) is updated by dynamics (16); 2) \( c(x_L, a_L) = \tilde{A}(t) = 0 \) at any time \( t \in [t_k, t_{k+1}] \) for \( k = 1, 0, \ldots, n_r - 1 \); and 3) the following boundary conditions are satisfied:
\[
\begin{align*}
\hat{y}_{L,HT}^T(t_k) \Psi_{y_{L,HT}}(t_k) - A_s &= 0, \\
\hat{y}_{L,HT}^T(t_{k+1}) \Psi_{y_{L,HT}}(t_{k+1}) &= 0
\end{align*}
\]

By taking the second time derivative of \( A(t) \), \( c(x_L, a_L) \) is obtained as follows:
\[
c(x_L, a_L) = x_L^T \Gamma_xx_L + 2x_L^T \Gamma_xa_L = 0
\]
where
\[
\begin{align*}
\Gamma_xx &= 2 \begin{bmatrix} 0_{6 \times 6} & 0_{6 \times 6} \\ 0_{6 \times 6} & P \end{bmatrix}, \\
\Gamma_x &= \begin{bmatrix} P & 0_{6 \times 6} \\ 0_{6 \times 6} & 0_{6 \times 6} \end{bmatrix}.
\end{align*}
\]

The objective of the optimal control planning is to determine the desired trajectories of the leaders by minimization of cost function
\[
J = \frac{1}{2} \int_{t_0}^{t_{k+1}} a_L^T(t) a_L(t) dt, \quad k = 0, 1, \ldots, n_r - 1,
\]
subject to boundary conditions
\[
\begin{align*}
x_L(t_k) &= \bar{x}_{L,k}, \\
x_L(t_{k+1}) &= \bar{x}_{L,k+1}
\end{align*}
\]
and equality constraint (53) at any time \( t \in [t_k(t_k), t_{k+1}(t_k)] \) for \( k = 0, 1, \ldots, n_r - 1 \), where \( t_k(t_k) \) is obtained by (13).

**Theorem 3** Suppose that leaders’ desired trajectories are updated by dynamics (16) such that equality constraint (53) is satisfied at any time \( t \in [t_k(t_k), t_{k+1}(t_k)] \) given the boundary conditions in (56). Assuming that the ultimate time \( t_o \) is given and \( t_k \) and \( t_{k+1} \) obtained by (13) are fixed. Then, the optimal desired trajectories of leaders minimizing the cost function (55) are governed by dynamics
\[
\begin{bmatrix} x_L \\ \lambda \end{bmatrix} = A_{x_L} \left( y(t) \right) \begin{bmatrix} x_L \\ \lambda \end{bmatrix}
\]
where
\[
A_{x_L} \left( y(t) \right) = \\
\begin{bmatrix}
-A_L - 2 y(t) B_2 \Gamma_x^{-1} & -B_2 \Gamma_x^{-1} \\
-2 y(t) \Gamma_x + 4 y(t)^2 \Gamma_x \Gamma_x^T & -A_L + 2 y(t) \Gamma_x \Gamma_x^T
\end{bmatrix}
\]
and \( \lambda \in \mathbb{R}^{12 \times 1} \) is the costate vector. In addition, the state vector \( x_L(t) \) and costate vector \( \lambda(t) \) are obtained by
\[
\begin{align*}
x_L(t) &= \left( \Phi_{11} \left( t, t_k \right) - \Phi_{12} \left( t, t_{k+1} \right) \Phi_{11} \left( t_{k+1}, t_k \right) \right) \bar{x}_{L,k} + \Phi_{12} \left( t, t_{k+1} \right) x_L(t_{k+1}) \\
\lambda(t) &= \left( \Phi_{21} \left( t, t_k \right) - \Phi_{22} \left( t, t_{k+1} \right) \Phi_{11} \left( t_{k+1}, t_k \right) \right) \bar{x}_{L,k} + \Phi_{22} \left( t, t_{k+1} \right) x_L(t_{k+1})
\end{align*}
\]
at time \( t \in [t_k, t_{k+1}] \), where
\[
\Phi = \begin{bmatrix} \Phi_{11} \left( t, t_k \right) & \Phi_{12} \left( t, t_k \right) \\ \Phi_{21} \left( t, t_k \right) & \Phi_{22} \left( t, t_k \right) \end{bmatrix}
\]
is the state transition matrix with partitions \( \Phi_{11}(t, t_k) \in \mathbb{R}^{12 \times 12}, \Phi_{12}(t, t_k) \in \mathbb{R}^{12 \times 12}, \Phi_{21}(t, t_k) \in \mathbb{R}^{12 \times 12}, \) and \( \Phi_{22}(t, t_k) \in \mathbb{R}^{12 \times 12} \).

**Proof** The optimal leaders’ trajectories are determined by minimization of the augmented cost function
\[
J_a = \int_{t_k}^{t_{k+1}} \left( \frac{1}{2} a_L^T a_L + \lambda^T \left( A_L x_L + B_L a_L - \bar{x}_L \right) + \gamma c \left( x_L, a_L \right) \right) dt
\]
where \( \lambda \in \mathbb{R}^{12 \times 1} \) is the costate vector and \( y(t) \) is the Lagrange multiplier. By taking variation from the augmented
cost function (61), we can write
\[
\delta I_a = \int_{t_k}^{t_{k+1}} \left[ \delta x_L^T \left( a_L + B_L^T \lambda + \gamma \frac{\partial c}{\partial a_L} \right) + \delta x_L^T \lambda \right] dt = 0
\]
and \( a_L \) is obtained as follows:
\[
a_L = -B_L^T \lambda - \gamma \frac{\partial c}{\partial a_L} = -2 \gamma(t) \Gamma_{xu} x_L.
\]
(64)

By substituting \( a_L = -B_L^T \lambda - 2 \gamma(t) \Gamma_{xu} x_L \), the equality constraint (53) is converted to
\[
c (x_L, a_L) = (4x_L^T \Gamma_{xu} \Gamma_{xu}^T x_L) \gamma(t) + x_L^T \Gamma_{xu} x_L - 2x_L^T \Gamma_{xu} B_L^T \lambda = 0.
\]
(65)

Therefore, Eq. (58b) is obtained from Eq. (65). By substituting \( a_L = -B_L^T \lambda - 2 \gamma(t) \Gamma_{xu} x_L \) into (16), we also obtain the leaders’ desired trajectories solving dynamics (57). The solution of dynamics (57) is given by
\[
\begin{bmatrix} x_L(t) \\ \lambda(t) \end{bmatrix} = \left[ \begin{bmatrix} \Phi_{11} (t, t_k) & \Phi_{12} (t, t_k) \\ \Phi_{21} (t, t_k) & \Phi_{22} (t, t_k) \end{bmatrix} \right] \begin{bmatrix} x_L(k) \\ \lambda_k \end{bmatrix} \quad \forall t \in [t_k, t_{k+1})
\]
(66)

at time \( t \in [t_k, t_{k+1}) \), where \( \lambda_k = \lambda(t_k) \). By imposing boundary condition (56b)
\[
\lambda_k = \Phi_{12} (t_k, t_{k+1}) (x_L (t_{k+1}) - \Phi_{11} (t_{k+1}, t_k) x_L (t_k))
\]
(67)
is obtained from (66). By substituting \( \lambda_k \) into (66), \( x_L(t) \) is obtained by (59a) at any time \( t \in [t_k, t_{k+1}) \).

By using (66) and (67), we can obtain the leaders’ global desired trajectories, defined by \( x_L(t) \) over the time interval \( [t_k, t_{k+1}) \), if \( \gamma(t) \), obtained based on \( x_L(t) \) and \( \lambda(t) \) in (58b), is known over \([t_k, t_{k+1}) \). Algorithm 3 proposes an iterative approach to determine \( \gamma(t) \) by combining (58b), (66), and (67). Note that the main functionality of Algorithm 3 is to determine ultimate time \( t_u \).

V. CONTINUUM DEFORMATION ACQUISITION

This article considers collective motion of a quadcopter team consisting of \( N \) quadcopters, where quadcopters are modeled by the dynamics presented in Appendix A. The dynamics of leader and follower quadcopter subteams are then presented by
\[
\begin{align*}
\dot{x}_L &= F_L (x_L) + G_L (x_L) u_L \\
y_L &= C_L x_L
\end{align*}
\]
(68a)

Algorithm 3: Assignment of Travel Time \( t_u \), \( \gamma \), and Desired Trajectory \( x_L(t) \) over \([t_0, t_u) \).

1: Get: \( s_{L,0}, \ldots, s_{L,n} \), small \( \epsilon_T > 0 \), small \( \epsilon_F > 0 \), small \( T_{\min} \) and large \( T_{\max} \).
2: Obtain \( \beta_k \) using (12).
3: Set: small \( T_{\min} \), large \( T_{\max} \), \( t_0 = 0, t_1 = 0, \ldots, t_{n+1} = 0 \).
4: \( t_u = T_{\max} + T_{\min} \).
5: while \( |t_u - T_{\max}| \geq \epsilon_T \) do
6: for \( k < n \) do
7: \( t_k \leftarrow \beta_k t_u \)
8: \( t_{k+1} \leftarrow t_k + \epsilon_F \)
9: \( \gamma' (t) = 0 \) at every time \( t \in [t_k, t_{k+1}] \)
10: \( \gamma (t) = 0 \) at every time \( t \in [t_k, t_{k+1}] \)
11: \( e_F = 2 \epsilon_F \).
12: while \( e_F \geq \epsilon_F \) do
13: Compute \( A_{\lambda_k} (\gamma(t)) \) using (58a).
14: Compute \( (x_L (t), x_L (t)) \) using (60).
15: Obtain \( x_L(t) \) by (59a) for \( t \in [t_k, t_{k+1}] \).
16: Obtain \( \lambda(t) \) by (59b) for \( t \in [t_k, t_{k+1}] \).
17: Compute \( \gamma' (t) \) for \( t \in [t_k, t_{k+1}] \),
18: \( \gamma' (t) = -x_L^T \Gamma_{xu} x_L + 2x_L^T \Gamma_{xu} B_L^T \lambda \).
19: \( e_F = \max_{t \in [t_k, t_{k+1}]} |\gamma(t) - \gamma' (t)| \).
20: \( \gamma (t) \leftarrow \gamma' (t) \).
21: while \( \epsilon_T \leq \delta \) then
22: \( T_{\max} \leftarrow t_u \).
23: \( T_{\min} \leftarrow t_u \).
24: \( t_u = T_{\min} + T_{\max} / 2 \).
25: \( \epsilon_T = \max_{t \in [t_k, t_k+1]} \| r(t) - p(t) \| \).
26: \( \epsilon_T > \delta \) then
27: \( \epsilon_T > \delta \) then
28: \( T_{\max} \leftarrow t_u \).
29: \( t_u = T_{\min} + T_{\max} / 2 \).
30: while \( \epsilon_T \leq \delta \) then
31: end while

where \( C_L \) and \( C_F \) are constant, \( x_L = [x_1^T \cdots x_n^T] \) and \( x_F = [x_1^T \cdots x_n^T] \) are the state vectors of leaders and followers, \( u_L = [u_1^T \cdots u_n^T] \) and \( u_F = [u_1^T \cdots u_n^T] \) are the input vectors of leaders and followers, \( y_L = [r_1^T \cdots r_n^T] \) and \( y_F = [r_1^T \cdots r_n^T] \) are the output vectors of leaders and followers, and \( F_L (x_L) = [f_{1}^T (x_1) \cdots f_{n}^T (x_n)] \). \( F_F (x_F) = [f_{1}^T (x_1) \cdots f_{n}^T (x_n)] \). \( G_L (x_L) = [g_1^T (x_1) \cdots g_n^T (x_n)] \). \( G_F (x_F) = [g_1^T (x_1) \cdots g_n^T (x_n)] \) are smooth functions.

The continuum deformation, defined by (2) and planned by leaders 1–3, are acquired by followers in a decentralized fashion through local communication [26]. Communication among the quadcopters is defined by graph \( \mathcal{G}(\mathcal{V}, \mathcal{E}) \) with the properties presented in Section II-A. Here, we review the
existing communication-based guidance protocol and the trajectory control design [26] in Sections V-A and V-B.

A. Communication-Based Guidance Protocol

Given followers’ communication weights, we define matrix
\[
W = \begin{bmatrix}
0_{3 \times 3} & 0_{3 \times (N-3)} \\
B_{MQS} & A_{MQS}
\end{bmatrix} \in \mathbb{R}^{(N-3) \times N}
\]
with \( B_{MQS} \in \mathbb{R}^{(N-3) \times 3} \) and \( A_{MQS} \in \mathbb{R}^{(N-3) \times (N-3)} \). The \((i, j)\) entry of matrix \( W = [W_{ij}] \in \mathbb{R}^{N \times N}\) is defined as follows [26]:
\[
W_{ij} = \begin{cases} 
  w_{i,j}, & i \in \mathcal{V}_F, j \in \mathcal{N}_i \\
  -1, & j = i \\
  0, & \text{otherwise}
\end{cases}
\]  
(69)

where \( w_{i,j} \) is the communication weight between follower \( i \in \mathcal{V}_F \) and in-neighbor \( j \in \mathcal{N}_i \) that remains constant at any time \( t \in [t_0, t_f] \). In a continuum deformation coordination, communication weights of follower \( i \in \mathcal{V}_F \) are obtained based on the material positions of quadcopter \( i \in \mathcal{V} \) and its in-neighbors, defined by \( \mathcal{N}_i \), and satisfy the following relations:
\[
\bigwedge_{i \in \mathcal{V}_F} \left( \sum_{j \in \mathcal{N}_i} w_{i,j} (p_{j,0} - p_{i,0}) = 0 \right).
\]  
(70)

In [26], we show that
\[
y_{HT} = \text{vec} \left( \begin{bmatrix} p_1(t) & \cdots & p_N(t) \end{bmatrix} \right)^T \in \mathbb{R}^{3N \times 1}
\]
aggregating \( x, y, \) and \( z \) components of the global desired positions of all quadcopters, can be defined based on \( y_{LT}(t) \) by
\[
y_{HT}(t) = (I_3 \otimes W_L) y_{LT}(t)
\]  
(71)

where
\[
W_L = \begin{bmatrix}
\Omega_1^T (p_{1,0}, p_{2,0}, p_{3,0}, p_{1,0}) \\
\vdots \\
\Omega_3^T (p_{1,0}, p_{2,0}, p_{3,0}, p_{N,0})
\end{bmatrix} \in \mathbb{R}^{N \times 3}
\]
is defined based on \( W \) by
\[
W_L = (-I_N + W)^{-1} \begin{bmatrix} I_3 & 0_{3 \times (N-3)} \end{bmatrix}^T.
\]  
(73)

Given the output vectors of the leaders’ dynamics (68a), denoted by \( y_L \), and followers’ dynamics (68b), denoted by \( y_F \), we define the MQS output vector
\[
y(t) = R_L y_L(t) + R_F y_F(t)
\]
to measure deviation of the MQS from the desired continuum deformation coordination by checking constraint (46), where \( R_L = [R_{L_i}] \in \mathbb{R}^{3N \times 9} \) and \( R_F = [R_{F_i}] \in \mathbb{R}^{3N \times (3N-3)} \).

B. Trajectory Control Design

The objective of control design is to determine \( u_i = \begin{bmatrix} u_i^T & u_i^T & u_i^T \end{bmatrix} \in \mathbb{R}^{12 \times 1} \) and \( u_F = \begin{bmatrix} u_F^T & \cdots & u_F^T \end{bmatrix} \) such that (46) is satisfied at any time \( t \in [t_0, t_f] \). We can rewrite the safety condition (46) as
\[
\bigwedge_{i \in \mathcal{V}} \left( (y(t) - y_{HT}(t))^T S_i (y(t) - y_{HT}(t)) \leq \delta^2 \right) \forall t
\]  
(75)

where \( S_i = [S_{ij}] \in \mathbb{R}^{3 \times 3, N} \) is defined as follows:
\[
S_{ij} = \begin{cases} 
  1 & \bigwedge_{j=1}^3 ((p = j) \land (q = N(j - 1) + i)) \\
  0 & \text{otherwise}
\end{cases}
\]  
(76)

We use the feedback linearization approach presented in Appendix C to obtain the control input vector \( u_i(t) \) for every quadcopter \( i \in \mathcal{V} \) such that inequality constraint (75) is satisfied.

VI. SIMULATION RESULTS

We consider an MQS consisting of \( N = 8 \) quadcopters with the material (reference) and initial formations shown...
in Figs. 2(a) and (b). The MQS is initially distributed over horizontal plane $z = 50$ m, where $\mathbf{d}_u = [2250 \ 250 \ 50]^T$ is the position of the center of the containment ball $S$ at the initial time $t_i = 0$ s. It is desired that the MQS finally reaches the final formation shown in Fig. 5 in an obstacle-laden environment shown in Fig. 6. The final formation of the MQS is obtained by homogeneous transformation of the MQS initial formation and specified by choosing $\sigma_{1,m} = 1.2$, $\sigma_{2,n} = \frac{1}{\sigma_{1,m}} = 0.83$, $\theta_{d,n} = -\frac{\pi}{4}$, and $\mathbf{d}_u = [850 \ 2250 \ 50]^T$.

**Interagent communication:** Given quadcopters’ initial positions, followers’ in-neighbors and communication weights are computed using the approach presented in Section V-A and are listed in Table I. Note that quadcopters’ identification numbers are defined by set $\mathcal{V} = \{1, \ldots, 8\}$, where $\mathcal{V}_L = \{1, 2, 3\}$ and $\mathcal{V}_F = \{4, \ldots, 8\}$ define the identification numbers of the leader and follower quadcopters, respectively.

**Safety specification:** We assume that every quadcopter can be enclosed by a ball of radius $\epsilon = 0.465$ m. For the initial formation shown in Fig. 2(a), $d_{min} = 4$ m is the minimum separation distance between every two quadcopters. Furthermore, $\sigma_{min} = \frac{1}{\sigma_{1,m}} = 0.83$ is the lower bound for the eigenvalues of matrix $\bar{U}_{xy}$. Per (49)

$$\delta = \frac{1}{2} (d_{min} - 2\epsilon) = 1.2$$

is the upper bound for deviation of every quadcopter from its global desired position at any time $t \in [t_i, t_f]$.

**MQS planning:** It is desired that the MQS remains inside a ball of radius $r_{max} = 50$ m at any time $t \in [t_i, t_f]$. By using the A* search method, the optimal intermediate waypoints of the center of the containment ball are obtained. Fig. 6 shows the optimal paths of the leaders between their initial positions and their target destinations. Given the intermediate waypoints of the center of containment ball, the desired trajectories of the leaders are determined by solving the constrained optimal control problem given in Section IV-C. Given $t_i = 0$ s and $\delta = 1.2$ m, $t_f \simeq 900$ s ($t_f \leq 900$) is assigned by using Algorithm 3. Components of the optimal control input vector $\mathbf{a}_1^*(t)$, $\bar{x}_{1,HT}^*(t)$, $\bar{x}_{2,HT}^*(t)$, $\bar{x}_{3,HT}^*(t)$, $\bar{y}_{1,HT}^*(t)$, $\bar{y}_{2,HT}^*(t)$, and $\bar{y}_{3,HT}^*(t)$ are plotted versus time $t$ in Fig. 7.

**MQS continuum deformation acquisition:** MQS continuum deformation coordination is acquired in a decentralized fashion using the communication graph shown in Fig. 2 with the communication weights listed in Table I. Note that followers’ communication weights are consistent with the material positions shown in Fig. 2. Figs. 8–10 show $x$, $y$, and $z$ components of position of every quadcopter $i \in \mathcal{V}$ versus
time $t$, respectively. Deviation of every quadcopter from the global desired position is plotted in Fig. 11. It is seen that deviation of no quadcopter exceeds $\delta = 1.2$ m at any time $t \in [0, 900]$ s. Furthermore, thrust force $p_i$, roll angle $\phi_i$, and pitch angle $\theta_i$ are plotted versus time in Figs. 12–14, respectively.

VII. CONCLUSION

This article developed an algorithmic and formal approach for the continuum deformation planning of an MQS coordinating in a geometrically constrained environment. By using the principles of Lagrangian continuum mechanics, we obtained safety conditions for interagent collision avoidance and follower containment through constraining the eigenvalues of the Jacobian matrix of the continuum deformation coordination. To obtain safe and optimal transport of the MQS, we contained the MQS by a rigid ball and determined the intermediate waypoints of the containment ball by using the $A^*$ search method. Given the intermediate configurations of the containment ball, we first determined the leaders’ intermediate configurations by decomposing the homogeneous deformation coordination. Then, we assigned the optimal desired trajectories of the leader quadcopters by solving a constrained optimal control problem.

Because this article uses the double integrator dynamics (16) to plan the desired trajectories of the leaders between consecutive waypoints, the desired positions and velocities of the leaders are continuous, but the desired accelerations of the leaders are discontinuous at times $t_1, \ldots, t_n$. If we choose quadruple integrator dynamics to plan the desired trajectories of the leaders, the desired positions, velocities, accelerations, and jerks of all leaders will be continuous at any time $t$, and thus, the continuum deformation planning and acquisition will be more smooth. This will be considered as one of the future plans. One other future direction is to integrate the mid-level optimal control planning with other high-level search planning such as RRT* or stable sparse RRT to advance the adaptability of the proposed approach.

APPENDIX A
QUADCOPTER DYNAMICS

Dynamics of every $i \in \mathcal{V}$ is given by

$$\begin{aligned}
\dot{x}_i &= f_i(x_i) + g_i(x_i)u_i \\
y_i &= \begin{bmatrix} x_i & y_i & z_i & \psi_i \end{bmatrix}^T
\end{aligned}
$$

(77)

where $y_i$ is the output vector, and

$$\begin{aligned}
x_i &= \begin{bmatrix} x_i & y_i & z_i & \dot{x}_i & \dot{y}_i & \dot{z}_i & \dot{\psi}_i \end{bmatrix}^T \\
u_i &= \begin{bmatrix} u_{1,i} & u_{2,i} & u_{3,i} & u_{4,i} \end{bmatrix}^T
\end{aligned}
$$

(78a)

(78b)
are state vector and control input of quadcopter \( i \in \mathcal{V} \), respectively. In (77)

\[
\mathbf{f}(\mathbf{x}_i) = \begin{bmatrix}
\dot{x}_i \\
\dot{y}_i \\
\dot{z}_i \\
\frac{\rho}{m} (\sin \phi_i \sin \psi_i + \cos \phi_i \cos \psi_i \sin \theta_i) \\
\frac{\rho}{m} (\cos \phi_i \sin \psi_i \sin \theta_i - \sin \phi_i \cos \psi_i) \\
\frac{\rho}{m} \cos \phi_i \cos \theta_i - g \\
\end{bmatrix}
\]

(79a)

\[
\mathbf{g}(\mathbf{x}_i) = \begin{bmatrix} \mathbf{g}_1 \mathbf{g}_2 \mathbf{g}_3 \mathbf{g}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 1} & \mathbf{0}_{9 \times 3} \\
\mathbf{0}_{3 \times 1} & \mathbf{I}_3 \\
0 & \mathbf{0}_{1 \times 3} \\
1 & \mathbf{0}_{1 \times 3} \\
\end{bmatrix}
\]

(79b)

are smooth functions obtained in [28], where \( m \) and \( \mathbf{J} \) are the mass and mass moment of inertia of quadcopter \( i \in \mathcal{V} \), respectively, \( \mathbf{0}_{3 \times 1} \in \mathbb{R}^{3 \times 1}, \mathbf{0}_{3 \times 3} \in \mathbb{R}^{3 \times 3} \), and \( \mathbf{0}_{3 \times 9} \in \mathbb{R}^{3 \times 9} \) are the zero-entry matrices, \( \mathbf{I}_3 \in \mathbb{R}^{3 \times 3} \) is the identity matrix, \( g = 9.81 \text{ m/s}^2 \) is the gravity, \( \mathbf{e}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \), and

\[
\mathbf{k}_{\theta,i} = \begin{bmatrix} S_\phi S_\theta + C_\phi C_\psi S_\theta & C_\phi S_\psi S_\theta - S_\phi C_\theta & C_\phi C_\psi \end{bmatrix}^T
\]

(80)

\[
\mathbf{\Gamma}_i (\phi_i, \theta_i, \psi_i) = \begin{bmatrix} 1 & 0 & -\sin \theta_i \\
0 & \cos \phi_i & \cos \theta_i \sin \phi_i \\
0 & -\sin \phi_i & \cos \phi_i \cos \theta_i \end{bmatrix}
\]

(81)

where \( S(\cdot) \) and \( C(\cdot) \) abbreviate \( \cos(\cdot) \) and \( \sin(\cdot) \), respectively.

APPENDIX B
QUADCOPTER ROTATION AND ANGULAR VELOCITY

We use 3-2-1 standard to characterize rotation of quadcopter \( i \) by defining body axes \( (\mathbf{\hat{e}}_{b,i}, \mathbf{\hat{r}}_{b,i}, \mathbf{\hat{k}}_{b,i}) \) with respect to the initial coordinate system with base vectors \( (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) \), where \( \mathbf{e}_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \) and \( \mathbf{e}_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \). Given roll angle \( \phi_i \), pitch angle \( \theta_i \), and yaw angle \( \psi_i \), we use the \( 3 - 2 - 1 \) Euler angle standard and obtain the following transformations:

\[
\begin{align*}
\mathbf{\hat{r}}_{b,i} &= \cos \psi_i \mathbf{\hat{e}}_1 + \sin \psi_i \mathbf{\hat{e}}_2 \\
\mathbf{\hat{r}}_{b,i} &= -\sin \psi_i \mathbf{\hat{e}}_1 + \cos \psi_i \mathbf{\hat{e}}_2 \\
\mathbf{\hat{k}}_{b,i} &= \mathbf{\hat{e}}_3
\end{align*}
\]

(82a)

\[
\begin{align*}
\mathbf{\hat{r}}_{b,i} &= \cos \psi_i \mathbf{\hat{e}}_1 + \sin \psi_i \mathbf{\hat{e}}_2 \\
\mathbf{\hat{r}}_{b,i} &= \sin \psi_i \mathbf{\hat{e}}_1 + \cos \psi_i \mathbf{\hat{e}}_2 \\
\mathbf{\hat{k}}_{b,i} &= \mathbf{\hat{e}}_3
\end{align*}
\]

(82b)

\[
\begin{align*}
\mathbf{\hat{r}}_{b,i} &= \cos \psi_i \mathbf{\hat{e}}_1 + \sin \psi_i \mathbf{\hat{e}}_2 \\
\mathbf{\hat{r}}_{b,i} &= \sin \psi_i \mathbf{\hat{e}}_1 + \cos \psi_i \mathbf{\hat{e}}_2 \\
\mathbf{\hat{k}}_{b,i} &= \mathbf{\hat{e}}_3
\end{align*}
\]

(82c)

The angular velocity of quadcopter \( i \in \mathcal{V} \) is then given by

\[
\mathbf{\omega}_i = \dot{\psi}_i \mathbf{\hat{k}}_{1,i} + \dot{\theta}_i \mathbf{\hat{k}}_{3,i} + \dot{\phi}_i \mathbf{\hat{k}}_{b,i}.
\]

(83)

APPENDIX C
QUADCOPTER TRAJECTORY CONTROL DESIGN

We use the feedback linearization method to design quadcopter control input \( \mathbf{u}_i \) for every \( i \in \mathcal{V} \). To this end, we define state transformation

\[
\mathbf{z}_i (\mathbf{x}_i) = \begin{bmatrix} r_i^T & r_i^T & r_i^T & \psi_i & \psi_i \end{bmatrix}^T
\]

(84)

and update \( \mathbf{z}_i \) by the following dynamics:

\[
\dot{\mathbf{z}}_i = \mathbf{A}_{FL} \mathbf{z}_i + \mathbf{B}_{FL} \mathbf{v}_i
\]

(85)

where \( \mathbf{v}_i = \begin{bmatrix} \dot{r}_i^T & \dot{r}_i^T & \dot{r}_i^T & \dot{\psi}_i & \dot{\psi}_i \end{bmatrix}^T \) is the control input of dynamics (85), and

\[
\mathbf{A}_{FL} = \begin{bmatrix}
\mathbf{0}_{3 \times 3} & \mathbf{0}_{9 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{9 \times 1} \\
\mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 9} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\
\mathbf{0}_{3 \times 1} & \mathbf{0}_{1 \times 9} & 0 & 1 \\
\mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 9} & 0 & 0 \\
\end{bmatrix}
\]

\[
\mathbf{B}_{FL} = \begin{bmatrix}
\mathbf{0}_{9 \times 3} \\
\mathbf{I}_3 \\
\mathbf{0}_{3 \times 1} \\
\mathbf{0}_{1 \times 3} \\
\end{bmatrix}
\]

Note that \( \mathbf{v}_i \) is related to the control input of quadcopter \( i \in \mathcal{V} \), denoted by \( \mathbf{u}_i \), by

\[
\mathbf{v}_i = \mathbf{M}_{1,i} \mathbf{u}_i + \mathbf{M}_{2,i}
\]

(86)

where

\[
\mathbf{M}_{1,i} = \begin{bmatrix}
L_\phi^i L_\psi^i L_\theta^i x_i & L_\phi^i L_\psi^i L_\theta^i y_i & L_\phi^i L_\psi^i L_\theta^i z_i \\
L_\phi^i L_\psi^i L_\theta^i y_i & L_\phi^i L_\psi^i L_\theta^i \dot{y}_i & L_\phi^i L_\psi^i L_\theta^i \dot{z}_i \\
L_\phi^i L_\psi^i L_\theta^i z_i & L_\phi^i L_\psi^i L_\theta^i \dot{z}_i & L_\phi^i L_\psi^i L_\theta^i \dot{\psi}_i \\
\end{bmatrix} \in \mathbb{R}^{4 \times 4}
\]

(87a)

\[
\mathbf{M}_{2,i} = \begin{bmatrix}
L_\phi^i \dot{x}_i & L_\phi^i \dot{y}_i & L_\phi^i \dot{z}_i & L_\phi^i \dot{\psi}_i \end{bmatrix}^T \in \mathbb{R}^{4 \times 1}.
\]

(87b)

PROPOSITION 3 Matrix \( \mathbf{M}_{1,i} \) is nonsingular if roll angle and thrust force of quadcopter \( i \in \mathcal{V} \) satisfies the following conditions: \( \phi_i \neq \frac{\pi}{2} \) and \( p_i \neq 0 \).
Proof By taking the first and second time derivatives of the acceleration of quadcopter $i \in V$, we can write

$$\ddot{\mathbf{r}}_i = \frac{p_i}{m_i} \mathbf{k}_{b,i} + \frac{1}{m_i} \sum_{j \in N_i} \omega_i \mathbf{k}_{b,i} \mathbf{u}_i + \dot{\mathbf{b}}_i \dot{\mathbf{r}}_i + \dot{\mathbf{b}}_i \mathbf{r}_i,$$  

(88a)

$$\dddot{\mathbf{r}}_i = \frac{1}{m_i} \left[ \mathbf{k}_{b,i} - p_i \mathbf{j}_{b,i} \mathbf{p}_i \mathbf{k}_{b,i} \right] + \dot{\phi}_i \left[ \mathbf{V}_i \phi_1 \mathbf{j}_{b,i} \right] \mathbf{u}_i + \ddot{\mathbf{b}}_i \ddot{\mathbf{r}}_i + \dot{\mathbf{b}}_i \dot{\mathbf{r}}_i + \dot{\mathbf{b}}_i \mathbf{r}_i,$$  

(88b)

Per (82c), $\mathbf{j}_{b,i} = \cos \phi_i \mathbf{k}_{b,i} - \sin \phi_i \mathbf{k}_{b,i}$ and $\mathbf{j}_{b,i} \times \mathbf{k}_{b,i} = \cos \phi_i \mathbf{i}_{b,i}$ can be substituted into (88b). Therefore, $\mathbf{M}_{i,i}$ becomes

$$\mathbf{M}_{i,i} = \begin{bmatrix} \frac{1}{m_i} \mathbf{k}_{b,i} - p_i \mathbf{j}_{b,i} \mathbf{p}_i \mathbf{k}_{b,i} & p_i \cos \phi_i \mathbf{i}_{b,i} & p_i \sin \phi_i \mathbf{i}_{b,i} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (89a)$$

If $\phi_i \neq \frac{\pi}{2}$ and $p_i \neq 0$, partition $\begin{bmatrix} \frac{1}{m_i} \mathbf{k}_{b,i} - p_i \mathbf{j}_{b,i} \mathbf{p}_i \cos \phi_i \mathbf{i}_{b,i} \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{3 \times 3}$ is nonsingular, which, in turn, implies that $\mathbf{M}_{i,i}$ is nonsingular, as well.

The control design objective that deviation of actual output $y_i$ from the desired output $y_{i,d} = \left[ \mathbf{r}_{i,d}^T \, \psi_{i,d} \right]^T$ remains bounded for every quadcopter $i \in V$ at any time $t \in [t_s, t_f]$ and safety condition (46) is satisfied, where

$$\mathbf{r}_{i,d}(t) = \begin{cases} \mathbf{p}_i(t), & i \in V_L \\ \sum_{j \in N_i} \omega_j \mathbf{r}_j(t), & i \in V_F \end{cases} \quad (90)$$

is called local desired position and $\psi_{i,d}(t)$ is the desired yaw angle of quadcopter $i \in V$. Note that the local desired position of follower $i \in V_F$ is defined based on the actual positions of its in-neighbor agents, but local and global desired positions of every leader $i \in V_L$ are the same. Without loss of generality, this article assumes that $\psi_{i,d}(t) = 0$ at any time $t$. To achieve the control objective, we define the desired state vector

$$\mathbf{z}_{i,d} = \left[ \mathbf{r}_{i,d}^T \, \dot{\mathbf{r}}_{i,d}^T \, \ddot{\mathbf{r}}_{i,d}^T \, \mathbf{r}_{i,d}^T \, \psi_{i,d}^T \, \dot{\psi}_{i,d}^T \right]^T, \quad \forall i \in V$$

and choose

$$\mathbf{v}_i = \mathbf{K}_i (\mathbf{z}_{i,d} - \mathbf{z}_i) \quad (91)$$

such that $\mathbf{A}_{FL} - \mathbf{B}_{FL} \mathbf{K}_i$ is Hurwitz. Then, the control input of quadcopter $i \in V$ is obtained by

$$\mathbf{u}_i = \mathbf{M}_{i,i}^{-1} (\mathbf{v}_i - \mathbf{M}_{2,i}). \quad (92)$$

Fig. 15 shows the block diagram of the quadcopter control design based on the feedback linearization method.

Acknowledgement

The author would like to thank Prof. E. Atkins.

References

[1] J. Zhang, J. Yan, P. Zhang, and X. Kong
Collision avoidance in fixed-wing UAV formation flight based on a consensus control algorithm
IEEE Access, vol. 6, pp. 43672–43682, 2018.

[2] S.-L. Du, X.-M. Sun, M. Cao, and W. Wang
Pursuing an evader through cooperative relaying in multi-agent surveillance networks
Automatica, vol. 83, pp. 155–161, 2017.

[3] A. Artiñano, R. M. Del Toro, and R. E. Haber
Consensus-based cooperative control based on pollution sensing and traffic information for urban traffic networks
Sensors, vol. 17, no. 5, 2017. Art. no. 953.

[4] B. Cheng, X. Wang, and Z. Li
Event-triggered consensus of homogeneous and heterogeneous multiagent systems with jointly connected switching topologies
IEEE Trans. Cybern., vol. 49, no. 12, pp. 4421–4430, Dec. 2019.

[5] Z. Tu, H. Yu, and X. Xia
Decentralized finite-time adaptive consensus of multiagent systems with fixed and switching network topologies
Neurocomputing, vol. 219, pp. 59–67, 2017.

[6] F. Muñoz, E. S. Espinosa Quesada, H. M. La, S. Salazar, S. Commuri, and L. R. García Carrillo
Adaptive consensus algorithms for real-time operation of multi-agent systems affected by switching network events
Int. J. Robust Nonlinear Control, vol. 27, no. 9, pp. 1566–1588, 2017.

[7] T. Liu and J. Huang
Leader-following attitude consensus of multiple rigid body systems subject to jointly connected switching networks
Automatica, vol. 92, pp. 63–71, 2018.

[8] Q. Ma and S. Xu
Consensus switching of second-order multiagent systems with time delay
IEEE Trans. Cybern., early access, doi: 10.1109/TCYB.2020.3011448.

[9] X. Wang and G.-H. Yang
Fault-tolerant consensus tracking control for linear multiagent systems under switching directed network
IEEE Trans. Cybern., vol. 50, no. 5, pp. 1921–1930, May 2019.

[10] M. A. Shahab, B. Mozafari, S. Soleymani, N. M. Dehkordi, H. M. Shourkaei, and J. M. Guerrero
Distributed consensus-based fault tolerant control of islanded microgrids
IEEE Trans. Smart Grid, vol. 11, no. 1, pp. 37–47, Jan. 2020.

[11] Q. Liu, Z. Wang, X. He, and D. Zhou
On Kalman-consensus filtering with random link failures over sensor networks
IEEE Trans. Autom. Control, vol. 63, no. 8, pp. 2701–2708, Aug. 2018.

[12] H. J. LeBlanc and X. D. Koutsoukos
Consensus in networked multi-agent systems with adversaries
In Proc. 14th Int. Conf. Hybrid Syst.: Comput. Control, 2011, pp. 281–290.

[13] K. Sakurama
Unified formulation of multi-agent coordination with relative measurements
IEEE Trans. Autom. Control, vol. 66, no. 9, pp. 4101–4116, Sep. 2021.

[14] M. Ji, G. Ferrari-Trecate, M. Egerstedt, and A. Buffa
Containment control in mobile networks
IEEE Trans. Autom. Control, vol. 53, no. 8, pp. 1972–1975, Sep. 2008.

[15] H. Liu, G. Xie, and L. Wang
Necessary and sufficient conditions for containment control of networked multi-agent systems
Automatica, vol. 48, no. 7, pp. 1415–1422, 2012.
B. Li, Z.-Q. Chen, Z.-X. Liu, C.-Y. Zhang, and Q. Zhang
Containment control of multi-agent systems with fixed time-delays in fixed directed networks
*Neurocomputing*, vol. 173, pp. 2069–2075, 2016.

H. Su and M. Z. Chen
Multi-agent containment control with input saturation on switching topologies
*IET Control Theory Appl.*, vol. 9, no. 3, pp. 399–409, 2015.

M. Asgari and H. Atrianfar
Necessary and sufficient conditions for containment control of heterogeneous linear multi-agent systems with fixed time delay
*IET Control Theory Appl.*, vol. 13, no. 13, pp. 2065–2074, 2019.

H. Atrianfar
Sampled-time containment control of high-order continuous-time mass under heterogeneous time-varying delays and switching topologies: A scrambling matrix approach
*Neurocomputing*, vol. 395, pp. 24–38, 2020.

G. Cui, S. Xu, Q. Ma, Z. Li, and Y. Chu
Command-filter-based distributed containment control of nonlinear multi-agent systems with actuator failures
*Int. J. Control*, vol. 91, no. 7, pp. 1708–1719, 2018.

D. Ye, M. Chen, and K. Li
Observer-based distributed adaptive fault-tolerant containment control of multi-agent systems with general linear dynamics
*ISA Trans.*, vol. 71, pp. 32–39, 2017.

S. Zhao, F. L. Lewis, and A. Davoudi
Resilient output containment of heterogeneous cooperative and adversarial multigroup systems
*IEEE Trans. Autom. Control*, vol. 65, no. 7, pp. 3104–3111, Jul. 2020.

H. Qin, H. Chen, Y. Sun, and L. Chen
Distributed finite-time fault-tolerant containment control for multiple ocean bottom flying node systems with error constraints
*Ocean Eng.*, vol. 189, 2019, Art. no. 106341.

T. Xu, G. Lv, Z. Duan, Z. Sun, and J. Yu
Distributed fixed-time triggering-based containment control for networked nonlinear agents under directed graphs
*IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 10, pp. 3541–3552, Oct. 2020.

J. N. Reddy
*An Introduction to Continuum Mechanics*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

H. Rastgoftar, E. M. Atkins, and I. Kolmanovsky
Scalable vehicle team continuum deformation coordination with eigen decomposition 2020, arXiv:2002.03036.

H. Rastgoftar
Fault-resilient continuum deformation coordination
*IEEE Trans. Control Netw. Syst.*, vol. 8, no. 1, pp. 423–436, Mar. 2021.

H. Rastgoftar and I. V. Kolmanovsky
Safe affine transformation-based guidance of a large-scale multi-quadcopter system (MQS)
*IEEE Trans. Control Netw. Syst.*, vol. 8, no. 2, pp. 640–653, Jun. 2021.

**Hossein Rastgoftar** received the B.Sc. degree in mechanical engineering (thermofluids) from Shiraz University, Shiraz, Iran, the M.S. degree in solid mechanics from Shiraz University, the M.S. degree in mechanical systems from the University of Central Florida, Orlando, FL, USA, and the Ph.D. degree in mechanical engineering from Drexel University, Philadelphia, PA, USA, in 2015.

He is currently an Assistant Professor with the University of Arizona, Tucson, AZ, USA. From 2020 to 2021, he was an adjunct Assistant Professor with the University of Michigan, Ann Arbor, MI, USA, where he was also a Postdoctoral Researcher from 2015 to 2017 and an Assistant Research Scientist from 2017 to 2020 with the Department of Aerospace Engineering.

His current research interests include dynamics and control, multiagent systems, cyber-physical systems, and optimization and Markov decision processes.