IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 March 2024, accepted 14 April 2024, date of publication 23 April 2024, date of current version 2 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3392577

RESEARCH ARTICLE

Decentralized Asynchronous Formation Planning of Multirotor Aerial Vehicles in Dynamic Environments Using Flexible Formation Graphs and Tight Trajectory Hulls

FAHAD TANVEER[1] AND MUHAMMAD BILAL KADRI[2]
[1]Electronics Engineering Department, College of Engineering, Karachi Institute of Economics and Technology, Karachi 75190, Pakistan
[2]Computer Science Department, College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia

Corresponding author: Muhammad Bilal Kadri (mkadri@psu.edu.sa)

This work was supported by Prince Sultan University funded for Article Processing Charges (APC).

ABSTRACT Formation flight holds significant potential for various applications involving aerial robot swarms. However, current methodologies lack the capability to autonomously execute large-scale formation flights in densely populated environments. To bridge the gap, a decentralized and asynchronous formation flight planner is proposed based on a graph-based formation metric and tight representations of kino-dynamically feasible trajectories for collision avoidance. The planner handles formation path planning along with dynamic obstacles and intervehicle collision avoidance using minimum volume convex hulls for agent trajectories. The employed formation metric is invariant to rotation, translation, and scaling, granting greater flexibility in formation coordination. A decoupled and distributed trajectory optimization framework is proposed to enhance the computational feasibility of large-scale formation flights. Moreover, to mitigate issues relating to communication delays between agents, asynchronous execution of a finite horizon navigation framework with usage of sparse trajectory control points for trajectory segments is employed. Simulations with multiple agents, static and dynamic obstacles support the robustness of the planner to formation flights in real world. The planner demonstrates goal/waypoints achievement and formation adherence capabilities that are assessed and compared using a popular quantifiable formation similarity metric. Furthermore, the paper also serves as a guideline to build upon trajectory planning frameworks for tight formation control in cluttered, dynamic environments.

INDEX TERMS Autonomous aerial vehicles, collision avoidance, decentralized control, formation control, graph theory, path planning, swarm robotics, trajectory optimization.

I. INTRODUCTION

Formation flight has emerged as a pivotal capability for autonomous swarms, enabling them to execute coordinated aerial maneuvers. In intricate urban landscapes formation-based navigation holds immense potential for applications such as logistics of heavy loads, collaborative mapping [1], search and rescue [2], [39], package delivery [3], and similarly in large scale farming and agriculture activities [4], and more. However, effectively incorporating real-world constraints into aerial formations remains an unsolved challenge. This article seeks to investigate the possible solutions and empower aerial swarms by introducing a formation flight scheme that enables them to maintain cooperative formation behaviors in cluttered environments.

In the realm of aerial robot swarms, the primary focus lies on factors such as ensuring kino-dynamic feasibility, effective obstacle avoidance, swarm reciprocal avoidance and formation integrity within densely populated environments. In terms of formation coordination, methods can

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Alshabi.

be categorized into either those designed for formation flight in open space or for constrained environments (considering obstacle avoidance). For the former numerous techniques have been proposed for obstacle-free environments. These approaches encompass virtual structures [5], leader-follower [6], [42], navigation functions [7], [43], reactive behaviors [8], consensus-based control laws [9], and barycentric coordinate-based control [10]. Recent developments include the presentation of a VIO-swarm system [11], capable of executing collision-free formation flight in open space without inter-robot collisions. The scheme in [12], introduces a distributed formation control method, reducing the dependency on a common reference frame.

In the context of constrained environments, extensive research deals with formation navigation by formulating local feedback laws. Han et al. [13] introduce a formation controller based on a complex-valued graph Laplacian, where a leader regulates formation scale for specific swarm maneuvers, such as navigating through narrow corridors. Zhao [14] proposes a leader-follower control law allowing affine transformation of the formation in response to environmental changes. Zhou et al. [15] combine virtual structures with feedback potential fields to generate collision-free trajectories for formation flight, although their approach can be susceptible to deadlocks and overlooks trajectory optimality. In contrast, predictive optimization-based methods offer improved optimality for swarm movements [38]. Alonso-Mora et al. [16] control a formation of drones to avoid collisions by optimally rearranging the desired formation and subsequently planning local trajectories. However, this approach lacks inter-vehicle coordination within the distributed planners, thus overlooking formation maintenance during local planning. Parys et al. [17] employ DMPC to address formation preservation by enforcing relative position constraints on the swarm. In their framework, coordination among agents is disrupted passively when obstacles violate positional constraints.

As swarm size expands, it becomes apparent that maintaining formation solely through trajectory planning, especially in the presence of robot deadlocks, is challenging. Turpin et al. [18] address the issue of concurrent assignment and collision-free trajectory generation, offering both centralized and decentralized solutions, thus enabling large-scale flight formations. Similarly, in [19] model predictive control is leveraged to simultaneously solve task assignment and trajectory generation for achieving desired formation shapes. The concept is extended in [20] to consider formation alignment, optimizing parameters such as scale and location, reducing formation costs, and expediting convergence. However, these methods neglect the need for formations to adapt flexibly in constrained environments. In [36] formation maneuvering is handled by converting the system to an unconstrained optimization problem, using cost functions based on the MINCO representation [37] of the trajectories and a graph-based formation metric. However, for tightly constrained scenarios the MINCO representation might fail due to the underlying conservatism.

A. CHALLENGES FOR FORMATION FLIGHTS

While extensive research has been conducted on aerial swarm navigation in formation, achieving robust formation flights in obstacle-rich environments has remained elusive. Several fundamental challenges impede the practical application of formation flight:

a) The conflict between maintaining the formation (close proximity) and individual robots avoiding obstacles is inherently challenging.

b) Imperfect communication channels in a close proximity formation impede ideal planning in dense environments, requiring unconventional mitigation measures

c) The swarm system faces difficulties in rapidly recovering from unfavorable formation states resulting from unexpected obstacles or sudden changes in the desired formation shape. Predefined formations often lack the flexibility needed to adapt to unknown constrained environments.

B. RESEARCH AVENUES AND CONTRIBUTIONS

In light of the challenges, it can be concluded that an ideal formation flight planner should possess the capability to flexibly maintain formation shape while avoiding obstacles, and successfully achieve goal/ waypoints. It should continuously replan segments and minimize the planned trajectory safe zones to avoid conservatism in cluttered scenarios/tight formations. Use sparse representations of the planned trajectories/ safe zones instead of trajectory discretization to effectively utilize inter-agent communication bandwidth. Employ an effective planning/ execution strategy to handle communications delays/message dropouts. Nevertheless, at the same time generate kino-dynamically feasible trajectories and support scalability. To bridge the gap, the paper proposes the following contributions

a) A customized asynchronous flight planner based on the general definition in [41] is proposed and a graph-based formation metric is utilized as a heuristic that is invariant to rotation, translation, and scaling, granting greater flexibility in formation coordination.

b) For close proximity formations, path planning and intervehicle collision/dynamic obstacles avoidance is accomplished using clamped B-splines and use of minimum volume (MINVO basis) convex hulls is adapted from [28] for agent trajectory flight zones; incorporated within the planner.

c) An asynchronous execution framework, usage of sparse trajectory control points and time intervals-based trajectory segment are employed to mitigate issues relating to communication delays between agents.

d) A decoupled and distributed trajectory optimization framework is proposed to enhance the computational feasibility of large-scale formation flights.

The following sections start-off with the multi-copter system's differential flatness properties, the conventional

trajectory optimization problem and insights on the decoupling of the trajectory optimization with reference to formation flights. Then we present our choice of the trajectory parametrization and the tight outer polyhedral representations. Section VI provides the mathematical foundation of the formation metric followed by the trajectory planning and execution framework. Finally, the formation trajectory planner algorithm is discussed along with its implementation in a realistic simulation scenario with insights on the results that support the robustness of the planner to formation flights in real world scenarios.

II. MULTICOPTER DYNAMICS & FLAT-OUTPUT SPACE

Multi-copter system's differential flatness properties [21], [22], [23] have shown that the overlapping of the flat-output space and the concerned configuration space is physically meaningful. Typical multi-copters with underlying underactuated dynamics can be characterized with the 04 flat output states: the 3-dminsional cartesian position vector of its Center of Gravity $(p_x, p_y, p_z)^T$ and the Euler-yaw angle ψ:

$$z = (p_x, p_y, p_z, \psi)^T \tag{1}$$

This flat output z, has complex spatial constraints on the translational part but offers the needed convenience for a multi-copter's motion planning in the Euclidean space. For Kino-dynamically feasible motion, the trajectory $z(t) : [0, T] \longrightarrow \mathbb{R}^m$ is optimized such that maximum spatial constraints are enforced directly in the flat output space.

For smoothness of the trajectory, quadratic function for control effort [24] is usually employed as a cost function over $z(t)$ with time regularization. The constraints on the system consists of the spatial constraints (from the configuration space) and the dynamics constraints, such as limits on the actuators or task-specific restrictions. Generally, for the collision-free motion

$$z(t) \in \mathcal{F}, \ \forall t \in [0, \ T], \tag{2}$$

where $\mathcal{F}$ is the chalked-out obstacle-free sectors in configuration space. The dynamic and other user specific constraints are usually given as an inequality

$$\mathcal{G}_D(x(t), u(t)) \preccurlyeq \mathbf{0}, \quad \forall t \in [0, \ T] \tag{3}$$

Since the original states $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ can be parameterized by finite derivatives of z, such that $x = \Psi_x(z(t), \dot{z}(t), \ldots z^{(s-1)}(t))$ and $\mathrm{u} = \Psi_u(z(t), \dot{z}(t), \ldots z^{(s)}(t))$, constraints on $z(t)$ are given as

$$\mathcal{G}_D\left(\Psi_x\left(z^{[s-1]}(t)\right), \Psi_u\left(z^{[s]}(t)\right)\right) \preccurlyeq \mathbf{0}, \tag{4}$$

Now due to the flatness property, the constraints on both x and u can be substituted by the equivalent output states and their finite derivatives

$$\mathcal{G}\left(z(t), \dot{z}(t), \ldots z^{(s)}(t)\right) \preccurlyeq \mathbf{0}, \quad \forall t \in [0, \ T], \tag{5}$$

where $\mathcal{G}$ encompasses of n_g corresponding constraints.

III. TRAJECTORY OPTIMIZATION PROBLEM

The flat output space along with the prescribed constraints gives us the following mathematical problem:

$$\begin{aligned} &\min_{z(t),T} \int_0^T v(t)^T \mathbf{W} v(t) dt + \rho(T), \\ &s.t. \ \ v(t) = z^{(s)}(t), \\ &\qquad \mathcal{G}\left(z(t), \dot{z}(t), \ldots, z^{(s)}(t)\right) \preccurlyeq \mathbf{0}, \\ &\qquad \forall \, t \ \in \ [0, \ T], z(t) \in \mathcal{F}, \\ &\qquad z^{[s-1]}(0) = \overline{z_0}, \\ &\qquad z^{[s-1]}(T) = \overline{z_f}, \end{aligned} \tag{6}$$

where the positive diagonal matrix $\mathbf{W} \in \mathbb{R}^{m \times m}$, the time regularization term $\rho : [0, \infty) \mapsto [0, \infty]$, the initial and terminal conditions $\overline{z_0} \in \mathbb{R}^{ms}$ and $\overline{z_f} \in \mathbb{R}^{ms}$ respectively. In addition, the control problem is commonly implemented in a finite number of discontinuous time instants. Moreover, due to the nonlinear continuous-time constraints $\mathcal{G}$ and the nonconvex (in-real) set constraints in $\mathcal{F}$ reasonable conditions are applied to make the optimization problem well defined and trivial.

The time regularization function ρ, makes it possible to trades off between the expected total time and control effort,

$$\rho_s(T) = \sum\nolimits_{i=0}^{M_T} b_i T^i \tag{7}$$

where $b_i is$ is strictly positive. Some choices are $\rho_s(T) = k_\rho T$ and $\rho_s(T) = k_\rho (T - T_\Sigma)^2$ with T_Σ as the expected time and we can also define ρ to fix the total time $\rho_s(T) = 0, \ T = T_\Sigma \ and \ \rho_s(T) = \infty, \ T \neq T_\Sigma$. Also, the nonlinear constraints $\mathcal{G}$, should be twice continuously differentiable and are required to be C^2. As for the free space $\mathcal{F}$, we approximate it by the union of closed convex sets $M_\mathcal{P}$ as

$$\mathcal{F} \simeq \tilde{\mathcal{F}} = \bigcup\nolimits_{i=1}^{M_\mathcal{P}} \mathcal{P}_i \tag{8}$$

For completeness, the convex sets are assumed to be locally sequential connected:

$$\begin{cases} \mathcal{P}_i \cap \mathcal{P}_j = \emptyset, & if \ \ |i-j| = 2, \\ Int \ \left(\mathcal{P}_i \cap \mathcal{P}_j\right) \neq \emptyset, & if \ \ |i-j| \leq 1, \end{cases} \tag{9}$$

where $Int \ (\cdot)$ stands for the interior of a set with $\overline{z_0}$ inscribed in $\mathcal{P}_1$ and $\overline{z_f}$ in $\mathcal{P}_{M_\mathcal{P}}$ and that each $\mathcal{P}_i$ is bounded within a convex polytope defined by its H-representation [25]:

$$\mathcal{P}_i^H = \{ x \ \in \ \mathbb{R}^m | A_i x \preccurlyeq b_i \} \tag{10}$$

Moreover, for formation adherence, additional formation constraints need to be enforced. This can be applied using undirected graph $\mathcal{G} = (\mathcal{V}, \ \mathcal{E})$ of N robots, where the set of vertices is specified by $\mathcal{V} := \{1, \ 2, \ldots, N\}$, and the edges are given by $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. For the graph $\mathcal{G}$, each vertex i denotes the i^{th} agent with position vector $p_i = [x_i, \ y_i, z_i] \in \mathbb{R}^3$. Here $e_{ij} \in \mathcal{E}$ is the edge that links agent $i \in \mathcal{V}$ and agent $j \in \mathcal{V}$.

This makes the optimization problem complex and computationally challenging without special measures. The

objective is to conceive an effective solver that can accomplish real time computing for trajectory solution at a high frequency. However, considering the subproblem where functional constraints $\mathcal{G}$ and spatial constraints $\mathcal{F}$ have been eliminated/already addressed and the intermediate points against a knot (time) vector are prior computed, we are left with a simpler M-stage minimization problem for the control effort for a chain of s-integrators as follows,

$$\begin{aligned} &\min_{z(t)} \int_{t_0}^{t_M} v(t)^T \mathbf{W} v(t) dt \\ &s.t.\ \ v(t) = z^{(s)}(t), t \in [t_0,\ t_M], \\ &\qquad z^{[s-1]}(t_0) = \overline{z_0}, \\ &\qquad z^{[s-1]}(t_M) = \overline{z_f}, \\ &\qquad z^{[d_i-1]}(t_i,) = \overline{z_i}, \\ &\qquad 1 \le i < M, t_{i-1} < t_i, \end{aligned} \tag{11}$$

Here, the interval $[t_0,\ t_M]$ is fragmented into M stages, corresponding M + 1 fixed time stamps, and specified boundary conditions. Also, $(d_i - 1) < s$ is the highest order of assumed derivatives of the flat output, with some derivatives fixed at end of each stage t_i, as $\overline{z_i} \in \mathbb{R}^{md_i}$.

IV. AGENT AND OBSTACLE TRAJECTORIES

Our environment consists of agents and obstacles. We define the agent as an element within the environment capable of exchanging information and making decisions accordingly. In other words, an agent can modify its trajectory based on the information received from the environment. Contrary the obstacle is an element within the environment that moves independently without considering the trajectories of other elements in the environment. Obstacles can either be static or dynamic.

For our agents, we use clamped uniform B-Splines [26], which are defined by $n + 1$ control points $\{q_0, \ldots q_n\}$ and $m+1$ knots $\{t_0, t_1, \ldots t_m\}$, satisfying the following conditions for the degree of polynomial p:

$$\underbrace{t_0 = \ldots = t_p}_{\text{clamped } p+1 \text{ knots}} < \underbrace{t_{p+1} < \ldots < t_{m-p-1}}_{\text{internal knots}} < \underbrace{t_{m-p} = \ldots = t_m}_{\text{clamped } p+1 \text{ knots}}$$

In the context of clamped uniform B-Splines, the internal knots are evenly spaced by Δt(i.e., $\Delta t = t_{k+1} - t_k\ \forall k = \{p, \ldots m-p-1\}$). The relationship $m = n + p + 1$ holds, resulting in a total of $m - 2p = n + p + 1$ intervals denoted by $j \in J$ defined in the range $t \in [t_{p+j},\ t_{p+j+1}]$. For the purposes of this paper, we adopt $p = 3$, meaning cubic B-Splines are employed. Consequently, each interval represents a polynomial of degree 3, and it is guaranteed to lie within the convex hull of its four control points $\{q_j, q_{j+1}, q_{j+2}, q_{j+3}\}$. Moreover, clamped B-Splines are ensured to pass through the first and last control points (q_0 and q_n). The velocity and acceleration of a B-Spline are also B-Splines of degrees $p-1$ and $p-2$, respectively, with their control points determined as specified in [27].

$$\begin{aligned} v_l &= \frac{p(q_{l+1} - q_l)}{t_{l+p+1} - t_{l+1}}\ \forall l \in L \backslash \{n\} \\ a_l &= \frac{(p-1)(v_{l+1} - v_l)}{t_{l+p+1} - t_{l+2}}\ \forall l \in L \backslash \{n-1, n\} \end{aligned} \tag{12}$$

We define State vector: $\boldsymbol{x} := [\,\boldsymbol{p^T},\ \boldsymbol{v^T},\ \boldsymbol{a^T}\,]^T \in \mathbb{R}^3$, where $\mathbf{p}$, $\mathbf{v}$, $\mathbf{a}$, $\mathbf{j} \in \mathbb{R}^3$ are Position, Velocity, Acceleration and Jerk. Also u := Number of agents + Number of obstacles. The set that contains the indexes of all the intervals of a B-Spline $J := \{0, 1, \ldots m - 2p - 1\}$. $L = \{0, 1, \ldots, n\}$ and index of the control point $l \in L$ for position, $l \in L \backslash \{n\}$ for velocity and $l \in L \setminus \{n-1, n\}$ for acceleration. We assume s as the index of the planning agent and predictable obstacle trajectories for dynamic obstacles. The set $I = \{0,\ 1, \ldots u\} \backslash s$ contains the indexes all the obstacles/agents, except the agent s.

For an obstacle $i \in I$, (where i is index of the obstacle/agent I) with actual future trajectory $p_i^{real}(t)$, we let $p_i(t)$ be predicted trajectory using some tracking and estimation algorithm. For safe operations we represent the obstacle as an outer polyhedral approximation, such that it is the Minkowski sum of an axis-aligned bounding box D_{ij} and the trajectory of the obstacle. Then $2\left(\alpha_{ij} + \beta_{ij}\right) \in \mathbb{R}^3$ represents minimum safe dimensions of the box for which

$$p_i^{real}(t) \in conv\left(D_{ij} \oplus p_i\left(\tau_j\right)\right), \quad \forall t \in [t_{p+j},\ t_{p+j+1}], \tag{13}$$

where α_{ij} and β_{ij} represents the prediction and trajectory discretization errors respectively and are assumed to be known. Also $p_i\left(\tau_j\right) = \left\{p_i(t) | t \in \tau_j\right\}$ with τ_j Uniform discretization of $[t_{p+j}\ ,\ t_{p+j+1}]$(timespan of interval j of the trajectory of agent s) with step size γ_j and such that $t_{p+j}\ ,\ t_{p+j+1} \in \tau_j$. Also $\oplus$ depicts the Minkowski sum and $conv(\cdot)$ the convex hull.

V. POLYHEDRAL REPRESENTATIONS

In order to circumvent the computational complexity associated with imposing an infinite number of constraints to distinguish between two trajectories, it becomes essential to calculate a precise polyhedral outer depiction for each interval within the optimized trajectory, specifically the trajectory being pursued by agent s as well as the trajectories of other agents and obstacles. For this we first define D_i as a 3D axis-aligned bounding box of the agent/obstacle i and the length of each side of the planning agent (i.e. agent whose index is s), each entry $\eta_s \in \mathbb{R}^3$. Also $\boldsymbol{C_{ij}}$ is set of vertexes of the polyhedron that completely encloses the trajectory of the obstacle/agent i during the initial and final times of the interval j of the agent s and $\boldsymbol{c} \in \mathbb{R}^3$ the**v**ertex of a polyhedron.

A. POLYHEDRAL REPRESENTATION OF TRAJECTORY OF THE PLANNING AGENT

When B-Splines or Bernstein basis are employed, a common approach to establish an outer polyhedral representation for each interval involves utilizing the polyhedron defined by the control points of that interval. This choice ensures that the

interval is entirely encompassed by this polyhedron. However, it is important to note that this approximation introduces a significant level of conservatism both in terms of position and velocity dimensions. In contrast, we adopt the MINVO basis [28], both in position and velocity spaces. This basis, constructed as a polynomial basis, is designed to minimize the volume of the simplex that encloses a given polynomial curve. The MINVO basis achieves a significantly reduced volume, being 2.36 and 254.9 times smaller (in position space) and 1.29 and 5.19 times smaller (in velocity space) compared to the Bernstein and B-Spline bases [28], respectively. For each interval denoted as j, the relationships among the vertexes of the MINVO control points (Q_j^{MV} and V_j^{MV} for position and velocity, respectively) and the B-Spline control points (Q_j^{BS}, V_j^{BS}) are as follows:

$$\begin{aligned} \mathrm{Q}_j^{MV} &= \mathrm{Q}_j^{BS}\mathrm{A}_{pos}^{BS}(\mathrm{A}_{pos}^{MV})^{-1} \\ \mathrm{V}_j^{MV} &= \mathrm{V}_j^{BS}\mathrm{A}_{vel}^{BS}(\mathrm{A}_{vel}^{MV})^{-1} \end{aligned} \tag{14}$$

where the matrices A are available in (for the MINVO basis) and in [26] (for the Bernstein and B-Spline bases). Also $\boldsymbol{Q}_j^{\boldsymbol{b}} := [\boldsymbol{q}_j, \ldots, \boldsymbol{q}_{j+3}]^{\boldsymbol{b}}$ is set of control points for position of the trajectory of the agent s for the interval j using the basis b and equivalently for velocity control points: $\boldsymbol{V}_{\boldsymbol{j}}^{\boldsymbol{b}} := [\boldsymbol{v}_{\boldsymbol{j}}, \ldots, \boldsymbol{v}_{\boldsymbol{j+2}}]^{\boldsymbol{b}}$. Where Notation for the basis used: Bernstein ($b = Be$), B-Spline ($b = BS$) or MINVO ($b = MV$).

B. POLYHEDRAL OF OTHER AGENTS' TRAJECTORIES

For the agent $i \neq s$ the trajectory is similarly a B-Spline, however the initial and terminal times are different from agent s trajectory optimization times i.e t_{in} and t_f. Therefore to obtain the outer polyhedral representation in the intervals $[t_{in},\ t_{in} + \Delta t],\ [t_{in} + \Delta t,\ t_{in} + 2\Delta t], \ldots, [t_f - \Delta t,\ t_f]$ of the agent i, the MINVO control points are computed for each interval of agent $i's$ trajectory in these intervals. Next the inflated bounding box B_i' obtained by adding a safety η_s to the dimensions of B_i, is placed each of the control points to obtain an outer polyhedral representation of the agent i trajectory for the interval j. The vertexes of the combined polyhedral is taken as C_{ij}.

C. POLYHEDRAL OF OBSTACLES TRAJECTORIES

For dynamic obstacles, if for the said times τ_j, $p_i(\tau_j)$ represents the set of locations of the obstacle i we compute the outer approximation of the 3D space the obstacle occupies. The obstacle size B_i is enlarged by the size of the agent s (η_s) and the obstacle prediction and trajectory discretization errors α_{ij}, β_{ij} for each interval j respectively to obtain the inflated box B_i'. B_i' is then placed at every $p_i(\tau_j)$ to obtain the convex hull of the entire set. For static obstacles the same method is adopted with no obstacle prediction and trajectory discretization errors and $p_i(t) =$ constant.

VI. FORMATION SIMILARITY METRIC

In order to specify the fixed formation model, an undirected graph $\mathcal{G} = (\mathcal{V},\ \mathcal{E})$ of N robots is employed, where the set

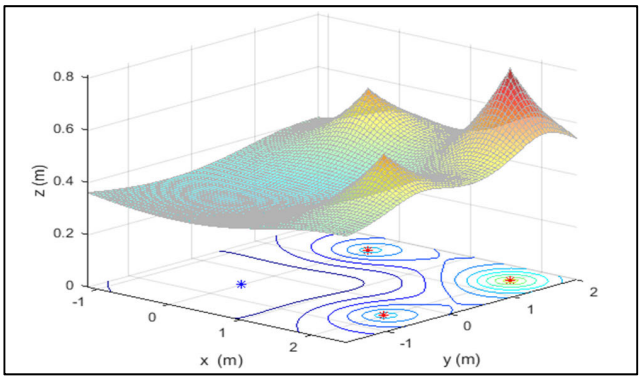

FIGURE 1. For the desired square formation in a 2D plane, the formation similarity metric is calculated for the blue agent, if it desires to move in any direction. The metric is minimum at the desired square formation position, increases as the agent moves out of this position and maximizes as it moves closer to other agents.

of vertices is specified by $\mathcal{V} := \{1,\ 2, \ldots, N\}$, and the edges is given by $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$.

For the graph $\mathcal{G}$, each vertex i denotes the i^{th} agent with position vector $p_i = [x_i,\ y_i, z_i] \in \mathbb{R}^3$ defined in their local formation frame. Here $e_{ij} \in \mathcal{E}$ is the edge that links agent $i \in \mathcal{V}$ and agent $j \in \mathcal{V}$ means the agents i and j can measure the Euclidean distance between them. We take on with the assumption, that each agent can communicate with other agents, hence we have a complete formation graph $\mathcal{G}$. Also, the weight of the edges of the formation graph $\mathcal{G}$ are calculated from the geometric distance, are all non-negative, and for edge e_{ij} can be given as

$$w = \|p_i - p_j\|^2,\ (i,j) \in \mathcal{V} \tag{15}$$

where $\|\cdot\|$ is the Euclidean norm. Now for the formation $\mathcal{G}$, the degree matrix $\mathbf{D} \in \mathbb{R}^{N\times N}$ and adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ is determined, for which, the equivalent Laplacian matrix is specified by

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \tag{16}$$

Thus, for the graph $\mathcal{G}$, we can determine the symmetric normalized Laplacian matrix as

$$\hat{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \tag{17}$$

where $\mathbf{I} \in \mathbb{R}^{N\times N}$ represents the identity matrix. The Laplacian matrix is a representation of the formation graph and contains information about the formation structure [32]. This can be employed to obtain a formation similarity distance as

$$\begin{aligned} f_s &= \left\|\hat{\mathbf{L}} - \hat{\mathbf{L}}_{des}\right\|_F^2 \\ &= tr\left\{\left(\hat{\mathbf{L}} - \hat{\mathbf{L}}_{des}\right)^T\left(\hat{\mathbf{L}} - \hat{\mathbf{L}}_{des}\right)\right\} \end{aligned} \tag{18}$$

This distance metric can be used to attain the swarm formation we desire and is based on the Frobenius norm $\|\cdot\|_F$. Here $tr\{\cdot\}$ represents the trace of a matrix, $\hat{\mathbf{L}}_{des}$ is the symmetric normalized Laplacian of desired formation

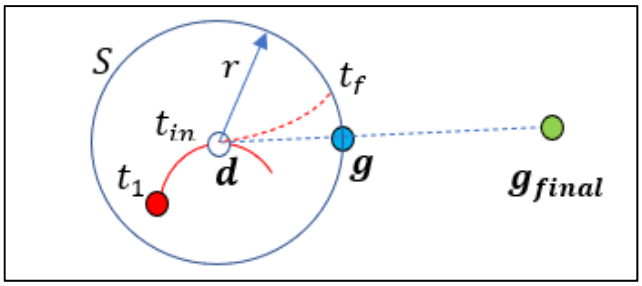

FIGURE 2. Here g_{final} is the goal, and ● the current position, ——— represents current executing trajectory, whereas - - - - the trajectory under planning (starts at $t = t_{in}$ and ends at $t = t_f$) with d (O)is used as the initial position, S is a sphere of radius r around d containing the planned trajectory *and* g (●) is the planned intermediate goal and is a projection of g_{final} on S.

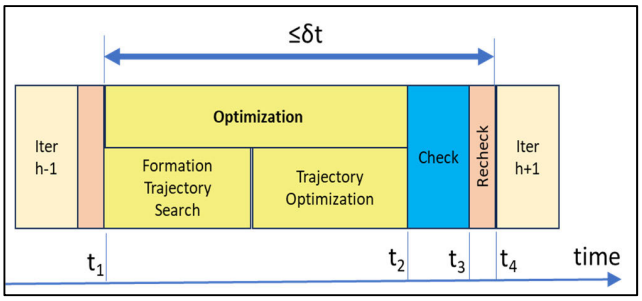

FIGURE 3. Each trajectory planning segment is divided into Formation Trajectory Search, Trajectory Optimization, collision Check and new trajectories available Re-check.

and $\hat{\mathbf{L}}$ is counterpart of the current swarm formation. Since the formation graph is based on the distance between agent positions, the distance metric f_s is inherently invariant to rotation and translation of the agent's formation. Moreover, by normalizing the graph Laplacian with the degree matrix in (3), invariance to scaling is also achieved.

This distance metric is differentiable analytically, with respect to the position of the agents. For agent i, we form the weight vector $w_i = [w_{i1}, w_{i1}, \ldots, w_{in}]^T$ of n adjacent edges $\{e_{i1}, e_{i1}, \ldots, e_{in}\}$. Then the gradient of f with respect to position $\boldsymbol{p_i}$ can be obtained by the chain rule as

$$\frac{\partial f}{\partial \boldsymbol{p_i}} = \frac{\partial \boldsymbol{w_i}}{\partial \boldsymbol{p_i}} \frac{\partial f}{\partial \boldsymbol{w_i}} \tag{19}$$

Resulting from the choice of the distance metric, its gradient respective of weight $\boldsymbol{w_{ij}}$ can be calculated as below:

$$\frac{\partial f}{\partial w_{ij}} = tr\left\{ \left(\frac{\partial f}{\partial \hat{\mathbf{L}}} \right)^T \left(\frac{\partial \hat{\mathbf{L}}}{\partial w_{ij}} \right) \right\}, \tag{20}$$

where,

$$\begin{aligned} \frac{\partial f}{\partial \hat{\mathbf{L}}} &= \frac{\partial \left\| \hat{\mathbf{L}} - \hat{\mathbf{L}}_{des} \right\|_F^2}{\partial \hat{\mathbf{L}}} = 2\left(\hat{\mathbf{L}} - \hat{\mathbf{L}}_{des} \right) \\ \frac{\partial \hat{\mathbf{L}}}{\partial w_{ij}} &= -\frac{\partial (\mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}})}{\partial w_{ij}} \end{aligned} \tag{21}$$

Then the for each agent i, $\partial f / \partial w_i$ can be given as $[\partial f / \partial w_{i1}, \partial f / \partial w_{i2} \ldots, \partial f / \partial w_{in}]^T$. Also $\partial \boldsymbol{w_i} / \partial \boldsymbol{p_i}$, is the Jacobian of the weight function (15) which is easily differentiable. Fig.1 shows the metric profile for a formation of four agents in a square formation in view from one agent, while others are stationary on desired positions.

VII. TRAJECTORY COMMITMENT AND EXECUTION

Establishment of reliable and safe communication channels between agents and their operators for effective operations is continuously evolving [40]. It is assumed that all agents possess the capability to establish communication directly with their counterparts, and to guarantee safety at least over a safety distance of 4 times than their planning horizon distance r. Moreover, despite sharing a common reference time, these agents initiate planning iterations in an asynchronous manner.

Due to asynchronous planning, it is required that two agents never simultaneously execute a new trajectory, for which a deconfliction strategy for check and re-check is proposed as in [33] and [35]. Each planning iteration is divided into 03 parts. The trajectory is planned in the interval $t \in (t_1, t_2]$ followed by check in the interval $t \in (t_2, t_3]$ where collision check is performed and a simple re-check at $t \in (t_3, t_4)$ which accounts for if any new trajectory has been committed during the check interval. This assumption is rooted in the objective of ensuring the safety of a UAV when it commits to a trajectory at $t = t_4$, subsequent to assessing all previously committed trajectories of other agents at times preceding t_4.

The agents keep on the computed trajectory previously found unless either the trajectory computed at the end of the optimization has collision indication with any agent trajectories conveyed during the optimization or during the Check period. Moreover, for real-time implementation the same trajectory is also executed if no feasible solution has been found in the optimization or the current iteration takes longer than δt seconds.

VIII. FORMATION TRAJECTORY PLANNER

To ensure collision-free constraints when static obstacles are present, a widely-used method involves initially identifying convex decompositions of free space. Subsequently, in the optimization problem, the outer polyhedral representation of each interval is constrained to remain within these identified convex decompositions [29], [30], [31]. However, this approach may become overly conservative, particularly in cluttered environments where the convex decomposition algorithm might not accurately capture the tight representation of the free space. When dealing with dynamic obstacles, these convex decompositions become even more challenging and potentially intractable due to the additional time dimension.

To ensure collision-free constraints in the presence of dynamic obstacles or agents, we utilize planes to separate the polyhedral representations of each trajectory. To achieve this, we employ a search-based algorithm specifically designed

to handle dynamic environments. This algorithm effectively determines the control points of the trajectory and identifies the planes that facilitate its separation from other obstacles or agents. Where the planes $\boldsymbol{\pi_{ij}}\left(\boldsymbol{n_{ij}},\boldsymbol{d_{ij}}\right) := \boldsymbol{n_{ij}^T x} + \boldsymbol{d_{ij}} = 0$ that separates $\boldsymbol{C_{ij}}$ from $\boldsymbol{Q_j^b}$.

To obtain the trajectory we compute the control points $\{q_0, \ldots q_n\}^{BS}$ and the separation planes π_{ij}, for the formation using a variant of the Octopus Search algorithm [33] that takes inspiration from A* [34] algorithm. The Octopus search computes trajectory using B-Splines and for collision avoidance uses the MINVO basis to handle dynamic obstacles/agents. As in A* the nodes are defined by the computed control point in the search. A priority queue Q, is initiated will all the available nodes. These nodes are ordered in ascending order of $f = g + \epsilon_h h + \epsilon_f f_s$, which comprises of the accumulative distance g between consecutive control points to the current node from the starting node q_0, the distance from the goal to the current node, the formation similarity metric and the bias terms ϵ_h and ϵ_f.

The trajectory is initialized by determining the control points $\boldsymbol{q_0}$, $\boldsymbol{q_1}, \boldsymbol{q_2}$, from $\mathbf{p_{in}}$, $\mathbf{v_{in}}$ **and** $\mathbf{a_{in}}$. We than insert $\boldsymbol{q_2}$ in the queue Q (line 3). The loop is run until as follows unless the queue Q is empty:

a) An empty set of control points $\boldsymbol{q_l}$ is initialized by inserting the first element of Q, and the same is removed from Q (lines 5-6).
b) Then the velocity samples for $\boldsymbol{v_l}$ satisfying both v_{max} and a_{max} are computed and stored in the set M.
c) We than check for the fitness of the current $\boldsymbol{q_l}$ and discard if either of the following conditions are true (l.s. in this text means linearly separable):
   i. Q_{l-3}^{MV} is not l.s. from $C_{i,\ l-3}$ for some $i \in I$.
   ii. $l = (n-2)$ and Q_{n-4}^{MV} is not l.s. from $C_{i,\ n-4}$ for some $i \in I$.
   iii. $l = (n-2)$ and Q_{n-3}^{MV} is not l.s. from $C_{i,\ n-3}$ for some $i \in I$.
   iv. $\left\|\boldsymbol{q_l} - \boldsymbol{d}\right\|_2 > r$.
   v. $\left\|\boldsymbol{q_l} - \boldsymbol{q_k}\right\|_\infty \leq \varepsilon'$ for some $\boldsymbol{q_k}$ already added to Q.
   vi. No feasible velocity samples in M.

In condition 1 it is made sure that no collision occurs between the (convex) hull of Q_{l-3}^{MV} and any other agent/obstacle interval at $l-3$. The linear separability is confirmed for interval $j = l-3$ of every agent/obstacle $i \in I$ by solving the linear problem and checking its feasibility:

$$\begin{aligned} n_{ij}^T c + d_{ij} > 0, \quad & \forall c \in C_{ij} \\ n_{ij}^T \boldsymbol{q} + d_{ij} < 0, \quad & \forall q \in Q_j^{MV} \end{aligned} \tag{22}$$

where the n_{ij} and d_{ij} define the decision variables planes π_{ij}. Equivalent MINVO control points are employed.

Due to the clamped B-spline, choice of $\boldsymbol{q_{n-1}}$ and $\boldsymbol{q_n}$ is restricted as $\boldsymbol{q_{n-2}} = \boldsymbol{q_{n-1}} = \boldsymbol{q_n}$ hence conditions 2 and 3 need to be checked.

Due to the restricted planning horizon, we confine the trajectory to be within a radius r, therefore we discard any $\boldsymbol{q_l}$ not satisfying condition 4. Moreover, any $\boldsymbol{q_l}$ is discard if it is very close to any already added $\boldsymbol{q_k}$ to Q (condition 5) which is realized by assuming a voxel grid of $2\varepsilon'$ voxel size, and searching if the new control point has un-occupied voxel. Finally, any $\boldsymbol{q_l}$ is also discard if M is an empty set such that no feasible $\boldsymbol{v_l}$ samples exist (condition 6).

At each iteration of the loop, we check if $\boldsymbol{q_{n-2}}$ has reached at less than ε'' of the goal g. Otherwise, velocity samples M are used and $\boldsymbol{q_{l+1}}$ is generated and added to Q. If goal is reached and all control points are determined, $\boldsymbol{q_{n-1}}$ and $\boldsymbol{q_n}$ are added satisfying clamped b-spline. In case all samples in Q are exhausted before reaching the goal, the nearest trajectory control points are returned. All separating planes are accompanied as well i.e. $\pi_{ij} \forall i \in I,\ \forall j \in J$.

TABLE 1. Initial and final conditions.

Initial Conditions	$\mathbf{p_{in}}$ (m)			$\mathbf{v_{in}}$	$\mathbf{a_{in}}$
	X	y	z		
Agent 01	-4.5	0.5	1	**0**	**0**
Agent 02	-4.5	-1	2.5	**0**	**0**
Agent 03	-4.5	0.5	2.5	**0**	**0**
Agent 04	-4.5	-1	1	**0**	**0**
Final Conditions	$\mathbf{p_g}$ (m)			$\mathbf{v_g}$	$\mathbf{a_g}$
	x	y	z		
Agent 01	3	-0.4	0	**0**	**0**
Agent 02	3	-1.9	1.5	**0**	**0**
Agent 03	3	-0.4	1.5	**0**	**0**
Agent 04	3	-1.9	0	**0**	**0**

IX. SIMULATION RESULTS

A simulation framework has been developed and simulations for 04 flying agents in a dynamic environment were performed. The initial and final conditions are specified for a square based formation, with enroute static and dynamic obstacles. The input and goal positions and conditions are specified as in Table 1, with $\mathbf{v_{max}} = 7\ m/s$ in all axis and $\mathbf{a_{max}} = [40,\ 40,\ 9.8]\ m/s^2$. The bounding box considered for the agent $\eta_s = 0.4$m all 03 sides. Two static obstacles and one dynamic obstacle with predictable trefoil knot trajectory are considered, with dimensions same as η_s. Static obstacles are centered at $[-0.3, -1.7, 0.3]$ and $[-0.3, -3.3, 0.3]$ and the initial position for dynamic obstacle is [0,0,0.2] and parametric equations as in [44]. The goal reaching time is tasked as $t_{max} = t_{min} + \left\|\mathbf{p_{in}} - \mathbf{p_g}\right\|_2 / (t_{margin} \cdot \max(\mathbf{v_{max}}))$ where $t_{min} = 0$ and $t_{margin} = 0.65$ is assumed. The arena size is constrained from -10m to $+10$m in all axis, with $r = 10m$ and $\varepsilon' = 0.01m$. The priority queue bias is set at $\epsilon_h = 6$ and $\epsilon_f = 3$.

In Fig. 4, the trajectory for 1^{st} agent is shown with the goal marker shown behind the dynamic obstacle. In the top figure, the convex hull traced by the dynamic obstacle is shown for first-three segments, along with the agent's trajectory on

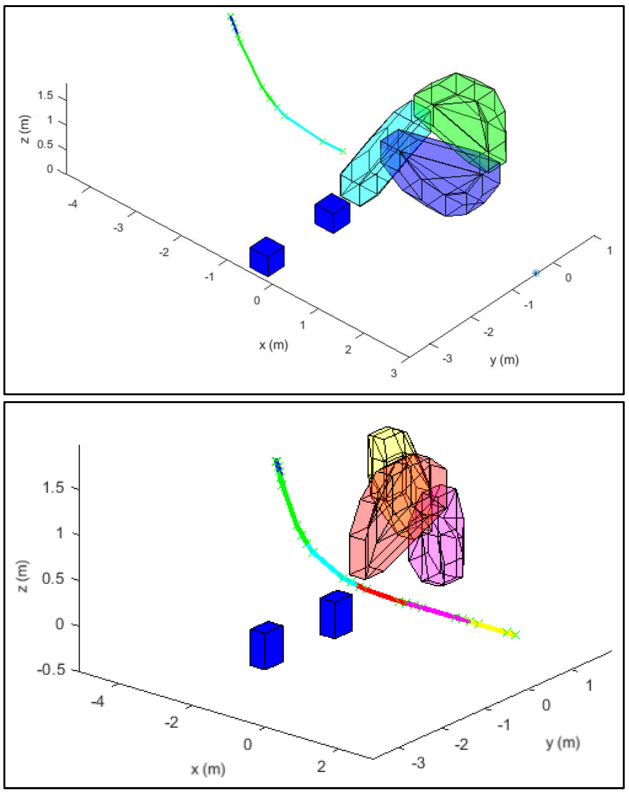

FIGURE 4. Trajectory for single agent, with 02 static and 01 dynamic obstacle. Colors showing convex hulls and trajectory segments in different time segments (Blue, Green, Cyan, Red, Purple and Yellow in chronological order). Top figure shows approach at the end of 3rd time segment, towards goal with dynamic obstacle in path. In bottom figure, the red and purple segments depict the 4th and 5th time segments respectively and are planned in a way that dynamic obstacle is avoided.

approach towards the goal. In the bottom figure the 1^{st} agent plans its time segments in a manner that the trajectory is avoided with the corresponding convex hulls traced by the dynamic obstacle in the respective time segments.

In Fig. 5, the trajectory for 04 agents are shown with the goal markers shown behind the static and dynamic obstacles from different viewpoints, that depicts that the goal is reached by the agents and formation is minimum disturbed in all time segments. In the bottom figure, agent 4 trajectory planning is zoomed-in to reveal the iterations taken by the planner to overcome the static obstacle.

A popular formation similarity transformation $Sim(3)$ [12], [36], can be used as a benchmark to assess the deviation between the desired formation F_d and current formation F_c, in terms of the overall position error e_{dist}. If $\mathbf{p_i^d}$ and $\mathbf{p_i^c}$ denote the position of i^{th} robot in formation F_d and F_c, respectively, related by a translation $\mathbf{t} \in \mathbb{R}^3$, rotation $\mathbf{R} \in SO(3)$, and a scaling factor $s \in \mathbb{R}_+$, then overall position error can be given as

$$e_{dist} = \min_{\mathbf{R},\, \mathbf{t},\, s} \sum_{i=1}^{N} \left\| \mathbf{p_i^d} - (s\mathbf{R}\mathbf{p_i^c} + \mathbf{t}) \right\|^2 \qquad (23)$$

By finding the optimal solutions to the transformations $\mathbf{R}$, $\mathbf{t}$ **and** $\boldsymbol{s}$ in (23), the influence to these can be alleviated

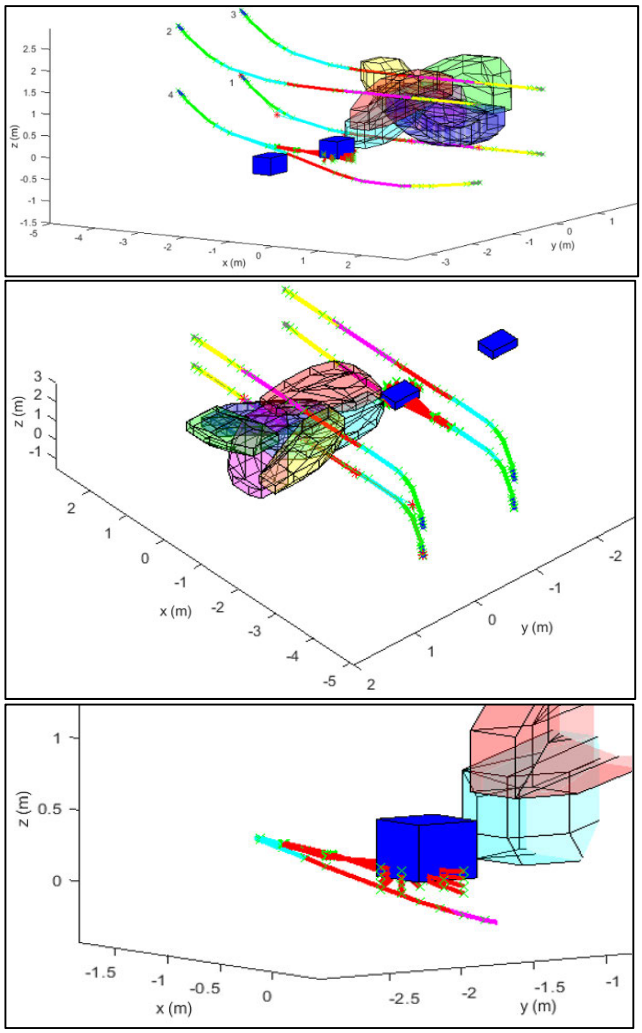

FIGURE 5. The trajectories for the 04 agents are given. The results show that the formation is maintained by the planner while avoiding obstacles and maintaining smooth kino-dynamic trajectories. In the last figure the planner finds the alternative route for the 4^{th} agent while avoiding the static obstacle and generates control points satisfying 3^{rd} order clamped Beizier curves.

and the formation positions can be fairly compared by calculating the position error with respect to $\boldsymbol{F_d}$. The more the two formations deviate the greater is the value of $\boldsymbol{e_{dist}}$.

For our scenario the maximum value for $e_{dist} = 0.795$. This value is comparable to the methods compared in [36], for a sparse scenario. The scenario for a cluttered environment with 05 agents in a pyramidal configuration is given in Fig. 6, with "Agent 05" in lead of 0.5m at the center of formation. All agents are now extended to travel a further 6m in x-direction with another set of dynamic and static obstacles in the path. The scenario depicts trajectory corrections for agents 01, 04 and 05. The agent 05 replans the trajectory for the first dynamic obstacle moving downward to avoid it in the red interval instance. In the same interval agent 04 also replans to avoid the static obstacle. The color patterns repeat again after the yellow interval. The agent 01 avoids the static obstacle in the second set of obstacles. Here again

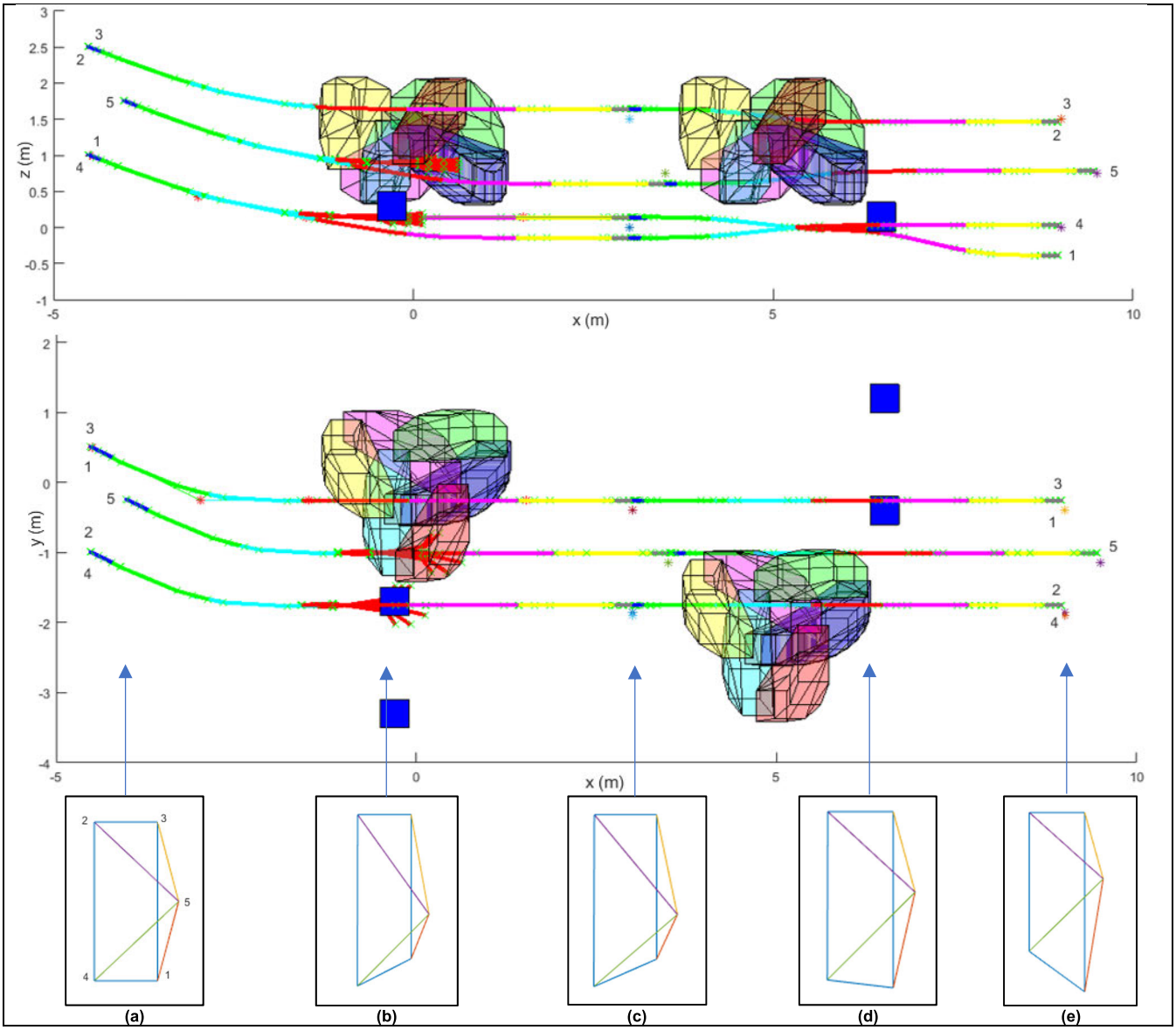

FIGURE 6. The trajectories for 05 agents in a pyramidal configuration. Top and middle figures depict the side and top views respectively. The bottom figures labeled a-e depict the formation shape at the marked intervals.

TABLE 2. Formation navigation schemes comparison.

Method	Success rate (%)		e_{dist}	Dynamic Obstacles	Execut -ion
	Sparse Scenario	Cluttered Scenario			
Zhou [15]	65	<15	High	No	Sync
Quan [36]	100	>95	Low	No	Sync
Ours	100	>95	Low	Yes	Async

the configuration is only marginally deviated and $e_{dist} < 1.0$ throughout.

However, our method additionally considers dynamic obstacles and demonstrates collision avoidance maneuvers in $\mathbb{R}^3$. The following table summarizes the comparison with recent methods reported in [36].

X. CONCLUSION

The Formation Planning of Multirotor Aerial Vehicles in Dynamic Environments is a challenging problem that has been addressed by a Decentralized and Asynchronous Formation Planner. We have tried to incorporate the frontiers of research in UAV cooperative control and formation planning in a Kino-dynamic trajectory (path) planning framework. The robustness of the planner lies within the Tight Trajectory Hulls representations, that avoids conservatism and preserve safety; when dealing with sparse control points information for planned trajectories, shared over limited communication bandwidths. These are essential to handle close proximity maneuvering, required by formation flying in cluttered dynamic environments. Moreover, using a Formation Graph Metric that is invariant to rotation, translation and

scaling provides the additional flexibility. The planner also intrinsically computes Kino-dynamically feasible trajectory control points, minimizing the convergence challenges in the conventional trajectory optimization problem. Finally, issues due to communication delays/constraints can be managed by adopting to the proposed/similar trajectory commitment and execution framework. The simulation results reveal the efficacy of the proposed planner and are benchmarked against recent published works. However, the planner may still be prone to deadlocks in a more constrained environment. In future work, we propose to extend the framework to incorporate strategies to timely identify and replan for potential deadlocks and add navigation and tracking components for real-world deployment.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of Prince Sultan University for paying Article Processing Charges (APC) of this publication.

REFERENCES

[1] N. Mahdoui, V. Frémont, and E. Natalizio, "Communicating multi-UAV system for cooperative SLAM-based exploration," *J. Intell. Robotic Syst.*, vol. 98, no. 2, pp. 325–343, May 2020.

[2] H. Khalil, S. U. Rahman, I. Ullah, I. Khan, A. J. Alghadhban, M. H. Al-Adhaileh, G. Ali, and M. ElAffendi, "A UAV-swarm-communication model using a machine-learning approach for search-and-rescue applications," *Drones*, vol. 6, no. 12, p. 372, Nov. 2022.

[3] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 47, no. 1, pp. 70–85, Jan. 2017.

[4] A. Jahn, R. J. Alitappeh, D. Saldaña, L. C. A. Pimenta, A. G. Santos, and M. F. M. Campos, "Distributed multi-robot coordination for dynamic perimeter surveillance in uncertain environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 273–278.

[5] F. Tanveer and M. B. Kadri, "A simulation framework for decentralized formation control of non-holonomic differential drive robots," in *Proc. SICE Int. Symp. Control Syst. (SICE ISCS)*, Tokyo, Japan, Mar. 2018, pp. 232–238, doi: 10.23919/siceiscs.2018.8330181.

[6] D. Panagou and V. Kumar, "Cooperative visibility maintenance for Leader–Follower formations in obstacle environments," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 831–844, Aug. 2014.

[7] M. C. De Gennaro and A. Jadbabaie, "Formation control for a cooperative multi-agent system using decentralized navigation functions," in *Proc. Amer. Control Conf.*, Jun. 2006, p. 6.

[8] H. Rezaee and F. Abdollahi, "A decentralized cooperative control scheme with obstacle avoidance for a team of mobile robots," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 347–354, Jan. 2014.

[9] A. Najm, I. Ibraheem, A. Azar, and A. Humaidi, "Genetic optimization-based consensus control of multi-agent 6-DoF UAV system," *Sensors*, vol. 20, no. 12, p. 3576, Jun. 2020.

[10] K. Fathian, S. Safaoui, T. H. Summers, and N. R. Gans, "Robust distributed planar formation control for higher order holonomic and nonholonomic agents," *IEEE Trans. Robot.*, vol. 37, no. 1, pp. 185–205, Feb. 2021.

[11] A. Weinstein, A. Cho, G. Loianno, and V. Kumar, "Visual inertial odometry swarm: An autonomous swarm of vision-based quadrotors," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1801–1807, Jul. 2018.

[12] P. C. Lusk, X. Cai, S. Wadhwania, A. Paris, K. Fathian, and J. P. How, "A distributed pipeline for scalable, deconflicted formation flying," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5213–5220, Oct. 2020.

[13] Z. Han, L. Wang, and Z. Lin, "Local formation control strategies with undetermined and determined formation scales for co-leader vehicle networks," in *Proc. 52nd IEEE Conf. Decis. Control*, Dec. 2013, pp. 7339–7344.

[14] S. Zhao, "Affine formation maneuver control of multiagent systems," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4140–4155, Dec. 2018.

[15] D. Zhou, Z. Wang, and M. Schwager, "Agile coordination and assistive collision avoidance for quadrotor swarms using virtual structures," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 916–923, Aug. 2018.

[16] J. Alonso-Mora, E. Montijano, M. Schwager, and D. Rus, "Distributed multi-robot formation control among obstacles: A geometric and optimization approach with consensus," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 5356–5363.

[17] R. Van Parys and G. Pipeleers, "Distributed model predictive formation control with inter-vehicle collision avoidance," in *Proc. 11th Asian Control Conf. (ASCC)*, Dec. 2017, pp. 2399–2404.

[18] M. Turpin, N. Michael, and V. Kumar, "Capt: Concurrent assignment and planning of trajectories for multiple robots," *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 98–112, Jan. 2014.

[19] D. Morgan, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, "Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1261–1285, Sep. 2016.

[20] S. Agarwal and S. Akella, "Simultaneous optimization of assignments and goal formations for multiple robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 6708–6715.

[21] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 2520–2525.

[22] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 620–626, Apr. 2018.

[23] B. Mu and P. Chirarattananon, "Trajectory generation for underactuated multirotor vehicles with tilted propellers via a flatness-based method," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2019, pp. 1365–1370.

[24] E. Verriest and F. Lewis, "On the linear quadratic minimum time problem," *IEEE Trans. Automat. Control*, vol. 36, no. 7, pp. 859–863, Jul. 1991.

[25] C. D. Toth, J. O'Rourke, and J. E. Goodman, *Handbook of Discrete and Computational Geometry*. Boca Raton, FL, USA: CRC Press, 2017.

[26] K. Qin, "General matrix representations for B-splines," *Vis. Comput.*, vol. 16, nos. 3–4, pp. 177–186, May 2000.

[27] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3529–3536, Oct. 2019.

[28] J. Tordesillas and J. P. How, "MINVO basis: Finding simplexes with minimum volume enclosing polynomial curves," 2020, *arXiv:2010.10726*.

[29] J. Tordesillas, B. T. Lopez, and J. P. How, "FASTER: Fast and safe trajectory planner for flights in unknown environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 1934–1940.

[30] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-D complex environments," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1688–1695, Jul. 2017.

[31] R. Deits and R. Tedrake, "Efficient mixed-integer planning for UAVs in cluttered environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 42–49.

[32] M. H. Bashir, M. Ahmad, D. R. Rizvi, and A. A. A. El-Latif, "Efficient CNN-based disaster events classification using UAV-aided images for emergency response application," *Neural Comput. Appl.*, Mar. 2024, doi: 10.1007/s00521-024-09610-4.

[33] J. Tordesillas and J. P. How, "MADER: Trajectory planner in multiagent and dynamic environments," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 463–476, Feb. 2022, doi: 10.1109/TRO.2021.3080235.

[34] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.

[35] K. Kondo, J. Tordesillas, R. Figueroa, J. Rached, J. Merkel, P. C. Lusk, and J. P. How, "Robust MADER: Decentralized and asynchronous multiagent trajectory planner robust to communication delay," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, London, U.K., May 2023, pp. 1687–1693, doi: 10.1109/ICRA48891.2023.10161244.

[36] L. Quan, L. Yin, C. Xu, and F. Gao, "Distributed swarm trajectory optimization for formation flight in dense environments," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Philadelphia, PA, USA, May 2022, pp. 4979–4985, doi: 10.1109/ICRA46639.2022.9812050.

[37] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Trans. Robot.*, vol. 38, no. 5, pp. 3259–3278, Oct. 2022, doi: 10.1109/TRO.2022.3160022.
[38] J. Tang, X. Chen, X. Zhu, and F. Zhu, "Dynamic reallocation model of multiple unmanned aerial vehicle tasks in emergent adjustment scenarios," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 2, pp. 1139–1155, Apr. 2023, doi: 10.1109/TAES.2022.3195478.
[39] E. T. Alotaibi, S. S. Alqefari, and A. Koubaa, "LSAR: Multi-UAV collaboration for search and rescue missions," *IEEE Access*, vol. 7, pp. 55817–55832, 2019, doi: 10.1109/ACCESS.2019.2912306.
[40] M. A. Khan, N. Kumar, S. A. H. Mohsan, W. U. Khan, M. M. Nasralla, M. H. Alsharif, J. Zywiolek, and I. Ullah, "Swarm of UAVs for network management in 6G: A technical review," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 1, pp. 741–761, Mar. 2023, doi: 10.1109/TNSM.2022.3213370.
[41] A. T. Azar, F. E. Serrano, N. A. Kamal, A. Koubaa, and A. Ammar, "Robust decentralized asynchronous control of unmanned aerial vehicles swarm with fast convergence switching topology," in *Advanced Machine Learning Technologies and Applications*. Cham, Switzerland: Springer, 2021.
[42] A. M. Parrany and A. Alasty, "Decentralized aggregation and leader-following control of a swarm of quadcopters with nonlinear under-actuated dynamics," *Aerosp. Sci. Technol.*, vol. 107, Dec. 2020, Art. no. 106317.
[43] J. Tang, J. Sun, C. Lu, and S. Lao, "Optimized artificial potential field algorithm to multi-unmanned aerial vehicle coordinated trajectory planning and collision avoidance in three-dimensional environment," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 233, no. 16, pp. 6032–6043, Dec. 2019.
[44] E. W. Weisstein. *Trefoil Knot*. Accessed: Apr. 18, 2024. [Online]. Available: https://mathworld.wolfram.com/

FAHAD TANVEER received the B.E. degree in electrical engineering from NED UET, Karachi, Pakistan, in 2007, and the M.S. degree in electrical controls from the National University of Science and Technology (NUST), Karachi, in 2015. He is currently pursuing the Ph.D. degree in electronics engineering with Karachi Institute of Economics and Technology (KIET), Karachi. His area of research is formation control and robust reactive trajectory planning for multi-agent aerial systems. He has over 13 years' experience of working in aerospace research and development sector. His research interests include mobile and aerial robotics, path and trajectory planning algorithms, satellite attitude and orbit control, and motion estimation.

MUHAMMAD BILAL KADRI received the B.S. degree in computer system engineering from the Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan, in May 2002, and the D.Phil. degree from the University of Oxford, U.K., in 2009, with research work focused on neuro-fuzzy control. He was an Assistant Professor with the Department of Electronics and Power Engineering, PN Engineering College, National University of Science and Technology (NUST), Pakistan, from October 2009 to August 2014. From September 2014 to February 2023, he was an Associate Professor and then a Professor with the Mechatronics Department, College of Engineering, KIET, Karachi, Pakistan. Currently, he is an Associate Professor with the College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia. He has over 20 years of teaching and research experience. He was the Director of the Intelligent Mobile Robotics (IMR) Laboratory, KIET, where he won several research grants in areas related to unmanned ground vehicles (UGV), unmanned air vehicles (UAV), and developing localization schemes for GPS denied environment. He has authored/coauthored more than 25 research articles in reputable international journals and 40 research papers in international conferences. His research interests include AI, robotics, ROS, sensor fusion, and fuzzy control systems.