2024 International Conference on Unmanned Aircraft Systems (ICUAS)
June 4-7, 2024. Chania, Crete, Greece

Fluid Flow Modeling and Experimental Evaluation of Unscrewed Aerial System Coordination

Harshvardhan Uppaluru, Mohammad Ghufran, and Hossein Rastgoftar

Abstract— **Reliability is a critical aspect of multi-agent system coordination as it guarantees the system's accurate and consistent functionality. If one agent in the system fails or behaves unexpectedly, it can negatively impact the performance and effectiveness of the entire system. Therefore, it is important to design and implement multi-agent systems with a high level of reliability to ensure that they can operate safely and move smoothly in the presence of unforeseen agent failure or lack of communication with some agent teams moving in a shared motion space. This paper presents a novel navigation model that, in an ideal fluid-flow, divides agents into cooperative (non-singular) and non-cooperative (singular) agents, with cooperative agents sliding along streamlines safely enclosing non-cooperative agents in a shared motion space. A series of flight experiments utilizing crazyflie quadcopters will experimentally validate the suggested model.**

I. Introduction

Robotics research has long drawn inspiration from nature. Researchers have examined how animals move, communicate, and interact with their environments in order to build robots capable of performing similar tasks. Biomimicry, the idea of designing and building technology inspired by nature, has resulted in the development of efficient robots that can adapt in response to their environment. Our inspiration stems from the flow of a fluid around a rock offering a glimpse into how robots can navigate around obstacles in their environments. For example, the principles of fluid dynamics can be applied to the design of a robot's movement, allowing it to move smoothly and efficiently around obstacles.

A. Related Work

Multi-Agent Systems (MAS) have been deployed in a plethora of robotics applications such as search and rescue missions [1], forest robotics [2], and surveillance [3], due to their significant advantages when compared to a single agent. Such MAS must be equipped with robust algorithms that can safely navigate around both static and dynamic obstacles in order for all agents to successfully complete the cooperative job. Various collision-free path planning works have been previously published such as collision cone [4], navigation functions [5], velocity obstacle concept [6], [7], flocking [8] and sampling based methods [9]. Flow-based control strategies for marine robots in gyre-like flows have been previously studied [10]. Artificial Potential Fields (APF) [11]–[13] is a simple and mathematically elegant technique originally proposed for manipulators and mobile robots in an operational space. Combining a positive potential around goal location and a negative potential around obstacles, this method guides the robot toward its goal by following the gradients of potential field while steering away from obstacles. A well-known issue of such an approach is getting trapped in local minima and in a real-world dynamic environment it has been shown that APF is inefficient [14].

Control Barrier Functions (CBFs) have emerged as a potential mathematical tool for safety assurances [15]. Using system dynamics, CBFs can be used to define a admissible region in the robot's workspace, and the robot's control inputs are then calculated to ensure that the robot's state remains within this region at all times. CBFs for a safe behavior in multi-agent robotics was studied previously [16] and a decentralized supervisory controller based on CBFs has been presented [17]. A combination of CBFs with Control Lyapunov Functions (CLFs) via quadratic programming was studied for cruise control applications [18]. We have recently developed an advanced physics-based automation system for the safe and efficient coordination of large-scale multi-agent systems, even in the face of disturbances and unexpected failures [19]–[22]. This innovative approach is composed of two operation modes: Homogeneous Deformation Mode (HDM) and Failure Resilient Mode (FRM). By applying the principles of continuum mechanics, we have successfully formalized the transitions between these two modes, enabling a robust response to varying operating conditions.

B. Contributions

This work presents a novel approach to ensuring the safe and resilient coordination of multiple agent teams moving collectively in a shared motion environment. Drawing inspiration from ideal fluid-flow models, each team treats its agents as cooperative particles within an ideal fluid-flow field while considering other teams' agents as singular points in the field. To ensure inter-agent collision avoidance and safely wrap the non-cooperative agents, the cooperative agents slide along the streamlines of an ideal fluid-flow field. The proposed approach will be experimentally validated using Crazyflie quadcopters in an indoor flight space. Compared to existing literature and the authors' previous work, this paper offers the following key contributions:

Contribution 1: The work extends the experimental evaluation of navigation presented in [22], which investigated a single failed quadcopter, by modeling and experimentally validating navigation in multi-agent systems in the presence of multiple non-concurrent failures and obstacles with arbitrary sizes and geometries.

Contribution 2: The proposed navigation approach establishes a novel paradigm for collision avoidance, wherein: (i)

Authors are with the Department of Aerospace and Mechanical Engineering, University of Arizona, Tucson, AZ, 85721 USA e-mail: {huppaluru, ghufran1942, hrastgoftar}@arizona.edu.

2024 International Conference on Unmanned Aircraft Systems (ICUAS) | 979-8-3503-5788-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICUAS60882.2024.10557029

each obstacle is treated as a rigid body whose boundary is determined by a streamline enclosing it; and (ii) collision avoidance is ensured by defining agents desired trajectories along the streamlines that safely wrap obstacles.

Contribution 3: The work models and experimentally validates navigation for multiple agent teams simultaneously coordinating within a shared motion space. In particular, this work presents algorithmic approaches for navigation in the presence of stationary and dynamic obstacles, encompassing various situations such as Stationary Non-Concurrent Failures (SNCF), Time-Varying Non-Cooperative (TVNC), Time-Varying Cooperative (TVC), and Stationary Obstacle-Laden Environments (SOLE) containing many obstacles with arbitrary sizes and geometries that are randomly distributed.

Contribution 4: The proposed SOLE fluid-flow navigation approach applies the existing mesh generation techniques [23], mainly used in computational fluid dynamics, to convert a highly-constrained motion space, populated with a random number of obstacles of arbitrary size and geometry into an obstacle-free planning space, and ensure collision avoidance by planning the agent coordination in the planning space. To the best of authors' knowledge, this is the first work that leverages computational fluid dynamics (CFD) mesh generation principles to ensure collision-free multi-agent coordination within a highly-constrained motion environment.

C. Organization

The remaining sections of the paper are organized as follows: A detailed description of our proposed methods is presented in Section II. The proposed model will be used in Section III to present five different operation modes under different communication and constraint protocols. Section IV outlines the experimental setup and presents the results of the experiments. We finally conclude the paper in Section V with thoughts about future directions.

II. Methodology

We consider a MAS represented by the set $\mathcal{V}=\{1,\cdots,N\}$, which is subsequently clustered into m distinct groups. These groups are identified by the set $\mathcal{M}=\{1,\cdots,m\}$. Let $\mathcal{V}_l$ be a set defining agents of cluster $l\in\mathcal{M}$; consequently, the set $\bar{\mathcal{V}}_l=\mathcal{V}\setminus\mathcal{V}_l$ defines agents that do not belong to cluster $l\in\mathcal{M}$ Although, $\mathcal{V}$ remains time-invariant, the number of agents in $\mathcal{V}_l$ can vary with time, suggesting that $\mathcal{V}_l$ may lose or absorb agents at any given time t.

To safely plan coordination of $\mathcal{V}_l$'s agents, in the presence of agents belonging to $\bar{\mathcal{V}}_l$, we consider $\mathcal{V}_l$'s agents as finite number of particles of an ideal fluid-flow field while $\bar{\mathcal{V}}_l$'s agents are either considered as "singularity points" or "rigid bodies" that are safely wrapped by the streamlines. For the ideal fluid-flow field, used for planning of coordination of $\mathcal{V}_l$'s agents, we define potential filed $\phi_l(x,y,\theta_l,t)$ and stream field $\psi_l(x,y,\theta_l(t),t)$, where x and y are position components, t denotes time, and $\theta_l(t)$ is the bulk motion direction of cluster $l\in\mathcal{M}$. Note that both potential and stream functions satisfy the Laplace equation:

$$\phi_{l_{xx}}+\phi_{l_{yy}}=0,\qquad \forall l\in\mathcal{M} \tag{1a}$$

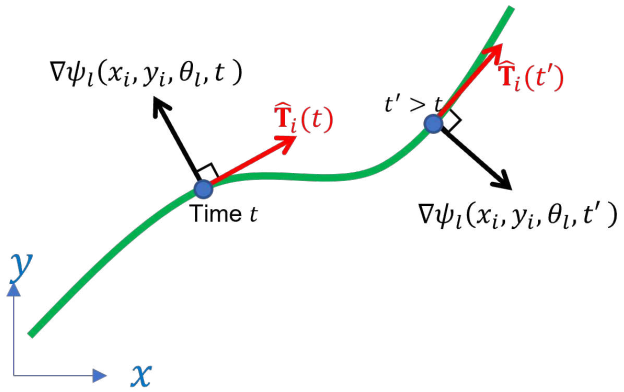

Fig. 1: Schematic of the desired path of agent $i\in\mathcal{V}_l$.

$$\psi_{l_{xx}}+\psi_{l_{yy}}=0,\qquad \forall l\in\mathcal{M} \tag{1b}$$

For the sake of simplicity, we use $\phi_{il}(t)$ and $\psi_{il}(t)$ to denote $\phi_{il}(t)=\phi_l(x_i,y_i,\theta_l,t)$ and $\psi_{il}(t)=\psi_l(x_i,y_i,\theta_l,t)$ to specify the corresponding potential and stream coordinates of agent $i\in\mathcal{V}_l$ positioned at (x_i,y_i) at time t.

Path Planning Strategy: Every agent $i\in\mathcal{V}_l$ can avoid inter-agent collision and hitting $\bar{\mathcal{V}}_l$'s agents when it slides along level curve $\psi_l(x_i,y_i,\theta_l(t),t)=\bar{\psi}_{i,l}(t)$ constant [21]. Therefore, the tangent vector to the desired sliding path of agent $i\in\mathcal{V}_l$ is obtained by

$$\hat{\mathbf{T}}_i(x,y,t)=\left[\frac{\partial\psi_l(x_i,y_i,\theta_l,t)}{\partial y}\quad -\frac{\partial\psi_l(x_i,y_i,\theta_l,t)}{\partial x}\right]^T, \tag{2}$$

for every $i\in\mathcal{V}_l$ and $l\in\mathcal{M}$ (See Fig. 6). for every $i\in\mathcal{V}_l$ and $l\in\mathcal{M}$ (See Fig. 6). The desired velocity of agent $i\in\mathcal{V}_l$ is given by

$$\mathbf{V}_i=v_l\hat{\mathbf{T}}_i,\qquad \forall i\in\mathcal{V}_l,\ l\in\mathcal{M}, \tag{3}$$

and we use the Algorithm 1 to update $\mathbf{z}_i$ at any time t.

Theorem 1. *Let $(x_{i,0},y_{i,0})$ denote the position of agent $i\in\mathcal{V}_l$ in the $x-y$ plane and $\theta_l(t_0)=\theta_{l0}$ denote the bulk motion direction of agents in $\mathcal{V}_l$ at initial (reference) time t_0 for every $l\in\mathcal{M}$. Define $\phi_{il}(t_0)=\phi_{il,0}=\phi_l(x_{i,0},y_{i,0},\theta_{l_0},t_0)$ and $\psi_{il}(t=t_0)=\psi_{il,0}=\psi_l(x_{i,0},y_{i,0},\theta_{l0},t_0)$ as the initial potential and stream coordinates of agent $i\in\mathcal{V}_l$ for every $l\in\mathcal{M}$, and*

$$r_{min,0}=\min_{\substack{i,j\\ i\neq j}}\sqrt{(\phi_{il,0}-\phi_{jl,0})^2+(\psi_{il,0}-\psi_{jl,0})^2} \tag{4}$$

as the minimum separation distance between agents in the $\phi_l-\psi_l$ plane, and

$$\zeta_{max}=\max_{x,y}\left(\left(\frac{\partial\phi_l}{\partial x}\right)^2+\left(\frac{\partial\phi_l}{\partial y}\right)^2\right). \tag{5}$$

Assume that the trajectory tracking control error of each individual agent does not exceed η and every agent can be enclosed by a ball of radius μ. Then, inter-agent collision avoidance is guaranteed, if the sliding speed $\dot{\phi}_{il}=v_l$ is the same for every agent in $\mathcal{V}_l$, and

$$\frac{r^2_{min,0}}{\zeta_{max}}\geq 4(\eta+\mu)^2 \tag{6}$$

Proof. According to equation (14), a one-to-one mapping between infinitesimal elements in $(d\phi_l-d\psi_l)$ and $(dx-dy)$ planes exists; they can be related by

$$\begin{bmatrix} d\phi_l \\ d\psi_l \end{bmatrix} = \mathbf{J} \begin{bmatrix} dx \\ dy \end{bmatrix} \tag{7}$$

where $\mathbf{J}$ is the Jacobian matrix defined as follows:

$$\mathbf{J}(x,y) = \begin{bmatrix} \frac{\partial \phi_l}{\partial x} & \frac{\partial \phi_l}{\partial y} \\ \frac{\partial \psi_l}{\partial x} & \frac{\partial \psi_l}{\partial y} \end{bmatrix} \tag{8}$$

Under Cauchy-Reimann conditions, we can write

$$\mathbf{J}^T\mathbf{J} = \left(\left(\frac{\partial \phi_l}{\partial x} \right)^2 + \left(\frac{\partial \phi_l}{\partial y} \right)^2 \right) \mathbf{I}_2,$$

where $\mathbf{I}_2 \in \mathbb{R}^{2\times 2}$ is the identity matrix. We can therefore write:

$$d\phi_l^2 + d\psi_l^2 = \begin{bmatrix} dx & dy \end{bmatrix} \mathbf{J}^T\mathbf{J} \begin{bmatrix} dx \\ dy \end{bmatrix} = \left(\phi_{lx}^2 + \phi_{ly}^2 \right) \left(dx^2 + dy^2 \right). \tag{9}$$

If the x and y coordinates along the stream line of every agent $i \in \mathcal{V}_l$ satisfies

$$\left(dx^2 + dy^2 \right) \geq \frac{d\phi_l^2 + d\psi_l^2}{\zeta_{max}}, \tag{10}$$

then,

$$(d_{min}(t))^2 \geq \frac{(r_{min}(t))^2}{\zeta_{max}}, \qquad \forall t \tag{11}$$

where

$$r_{min}(t) = \min_{\substack{\text{i,j} \\ \text{i}\neq\text{j}}} \sqrt{\left(\phi_{il}(t) - \phi_{jl}(t)\right)^2 + \left(\psi_{il}(t) - \psi_{jl}(t)\right)^2}, \qquad \forall t, \tag{12a}$$

$$d_{min}(t) = \min_{\substack{\text{i,j} \\ \text{i}\neq\text{j}}} \sqrt{\left(x_i(t) - x_j(t)\right)^2 + \left(y_i(t) - y_j(t)\right)^2}, \qquad \forall t. \tag{12b}$$

When the sliding speed $\dot{\phi_{il}} = v_l$ is the same for every agent $i \in \mathcal{V}_l$, it's agents move as particles of a rigid-body in the $\phi_l - \psi_l$ plane, and thus, inter-agent distances in the $\phi_l - \psi_l$ plane are time-invariant. As a result, the minimum separation distance of the desired formation in the $\phi_l - \psi_l$ plane can be assigned at reference time t_0, when the failed agent no-fly zone first appears. Therefore, $p_{min,0} = p_{min}(t)$ and Eq. (13) simplifies to

$$(d_{min}(t))^2 \geq \frac{(r_{min,0})^2}{\lambda_{max}}, \qquad \forall t \tag{13}$$

Since $d_{min}(t) \geq 2(\eta + \mu)$ is the collision avoidance condition at any time t, the inter-agent collision avoidance is assured, if Eq. (6) is satisfied. □

Remark 1. We note that the ideal fluid-flow coordination is defined over a 2-D plane, which is called $x-y$ plane in this paper. However, every agent $i \in \mathcal{V}$ is free to move along a direction that is normal to the $x-y$ plane, while x and y components of its desired position is restricted to slide along a streamline determined by Eq. (2). For better clarification, Fig. 2 shows how a multi-agent system, moving in a 3-D space, can apply the ideal fluid flow model to safely wrap obstacles specified as vertical cylinders [24].

Solutions: Given above problem setting, we will develop *analytic* and *numerical* solutions with the details provided in Sections II-A and II-B to define the potential fiction ϕ_l and stream function ψ_l for group coordination of $\mathcal{V}_l$'s agents. The analytical approach will be used when $\bar{\mathcal{V}}_l$'s agents are dynamic, and thus, potential and stream curves are time-varying. The analytic method considers $\bar{\mathcal{V}}_l$'s agents as singularity points that are excluded by combining irrational fluid flow patterns. On the other hand, the numerical solution will be applied to safely plan coordination $\mathcal{V}_l$ in the presence of many static obstacles, with arbitrary size and geometry, that are randomly distributed in the motion space, to maximize the motion space usability while ensuring collision avoidance.

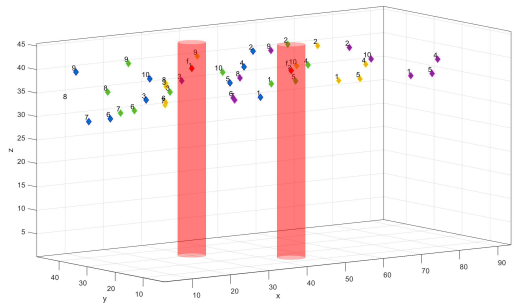

Fig. 2: Two-dimensional fluid flow coordination in a 3-dimensional motion space where obstacles are wrapped by vertical cylinders [24].

Navigation Modes: By applying the proposed fluid flow guidance, this paper implements and experimentally evaluates collision-free navigation of multiple groups of agents under (i) Stationary Non-Concurrent Failures (SNCF), (ii) Time-Varying Non-Cooperative (TVNC), (iii) Time-Varying Cooperative (TVC), and Stationary Obstacle-Laden Environment (SOLE) scenarios, with the properties listed in Table I.

A. Analytic Approach

Assuming the agents in $\mathcal{V}$ operate in the $x-y$ plane, we represent the position of agent $i \in \mathcal{V}$ by complex variable $\mathbf{z}_i = x_i + \mathbf{j}y_i$. In order to safely plan a collision-free coordination for agents in $\mathcal{V}_l$, in the presence of agents in $\bar{\mathcal{V}}_l$, we treat agents in $\mathcal{V}_l$ as a finite number of particles in a time-varying ideal fluid-flow field. This ideal flow field is defined by combining uniform flow and doublet flow in the $x-y$ plane. Therefore, agents in $\mathcal{V}_l$ perceive agents in $\bar{\mathcal{V}}_l$ as a collection of singularities in the $x-y$ plane and exclude them by employing vertical cylinders. These cylinders are derived by defining the following complex function:

$$\begin{aligned} \mathbf{f}\left(\mathbf{z}_i e^{-\mathbf{j}\theta_l(t)}, t\right) &= \phi_l(x_i, y_i, \theta_l(t), t) + \mathbf{j}\psi_l(x_i, y_i, \theta_l(t), t) \\ &= (1-\beta_l)\mathbf{z}_i e^{-\mathbf{j}\theta_l} + \beta_l \sum_{h\in\bar{\mathcal{V}}_l} \left(\left(\mathbf{z}_i e^{-\mathbf{j}\theta_l} - \mathbf{z}_h(t) \right) + \frac{\Delta_h^2}{\mathbf{z}_i e^{-\mathbf{j}\theta_l} - \mathbf{z}_h(t)} \right), \end{aligned} \tag{14}$$

This equation applies to every cluster $l \in \mathcal{M}$ and agent $i \in \mathcal{V}_l$, where $\beta_l \in \{0,1\}$ is a binary variable, $\theta_l(t)$ is a time dependent angle that determines the bulk motion direction of cluster $l \in \mathcal{M}$, and $\Delta_h \in \mathbb{R}_+$ is the chosen exclusion radius such that the size of agent $h \in \bar{\mathcal{V}}_l$ is properly incorporated. Because Eq. (14) establishes a nonsingular transformation

TABLE I: Properties of the investigated fluid-flow navigation problems.

Scenarios	θ_l $(l \in \mathcal{M})$	$\mathcal{V}_l$ $(l \in \mathcal{M})$	β_l $(l \in \mathcal{M})$	m
SNCF	Time-Invariant	Time-Varying	$\beta_l = 1$	$m = 2$
TVNC	Time-Varying	Time-Invariant	$\beta_l = 1$	$m = 2$
TVC	Time-Varying	Time-Invariant	$\beta_l \in \{0, 1\}$	$m > 1$
SOLE	Time-Invariant	Time-Invariant	$\beta_l = 1$	$m = 1$

between $\mathbf{z}_i = x_i + \mathbf{j}y_i$ and $\phi_{il} + \mathbf{j}\psi_{il}$ for every $i \in \mathcal{V}_l$ and $l \in \mathcal{M}$, (x_i, y_i) can be uniquely obtained based on $(\phi_{il}(t), \psi_{il}(t))$ at any time t by

$$x_i = g_1(\phi_{il}, \psi_{il}, \theta_l, t) \tag{15a}$$

$$y_i = g_2(\phi_{il}, \psi_{il}, \theta_l, t) \tag{15b}$$

We use g_1 and g_2 in Algorithm 1 for presenting the position update law.

Algorithm 1 Position Update Algorithm for Every Cluster $l \in \mathcal{M}$ under Fluid-Flow Navigation Strategy.

Get: Time increment Δt, Δ_h and $\mathbf{z}_h$ for every $h \in \bar{\mathcal{V}}_l$, $\beta_l \in \{0, 1\}$, θ_l sliding speed v_l, and current position $\mathbf{z}_i$ of every agent $i \in \mathcal{V}_l$.
Obtain: Next position $\mathbf{z}'_i = x'_i + \mathbf{j}y'_i$.
for $i \in \mathcal{V}_l$ **do**
    Compute current $\phi_{il}(t)$ and $\psi_{il}(t)$ using Eq. (14).
    Compute next potential ϕ'_{il}: $\phi_{il} = \phi_{il} + v_l \Delta t$.
    Compute next stream ψ'_{il}: $\psi_{il} = \psi_{il}$.
    Compute next x'_i: $x'_i = g_1(\phi'_{il}, \psi'_{il}, \theta_l, t)$.
    Compute next y'_i: $y'_i = g_2(\phi'_{il}, \psi'_{il}, \theta_l, t)$.
end for

B. Numerical Approach

For the numerical solution, we propose to establish a non-singular mapping between an obstacle-laden "motion space", specified by position components $X = x\cos\theta_l + y\sin\theta_l$ and $Y = y\cos\theta_l - x\sin\theta_l$, and an obstacle-free "planning space" that is defined by coordinates ϕ_l and ψ_l, where $X + \mathbf{j}Y = \mathbf{z}e^{-\mathbf{j}\theta_l}$ and θ_l is constant. By using the method presented in [23], $X(\phi_l, \psi_l)$ and $Y(\phi_l, \psi_l)$, defined over the planning space, are obtained by solving

$$aX_{\phi_l\phi_l} - 2bX_{\phi_l\psi_l} + cX_{\psi_l\psi_l} = 0, \qquad l \in \mathcal{M}, \tag{16a}$$

$$aY_{\phi_l\phi_l} - 2bY_{\phi_l\psi_l} + cY_{\psi_l\psi_l} = 0, \qquad l \in \mathcal{M}, \tag{16b}$$

where $a = X^2_{\psi_l} + Y^2_{\psi_l}$, $b = X_{\phi_l}X_{\psi_l} + Y_{\phi_l}Y_{\psi_l}$, and $c = X^2_{\phi_l} + Y^2_{\phi_l}$.

For better clarification, Fig. 5 shows an obstacle-laden environment with eight obstacles. The streamlines shown by the black curves and the potential lines shown by the red curves are both obtained numerically by solving partial differential equations (16) that are defined over the "planning" space. As shown in Fig. 5, an agent following the streamline shown by green can safely avoid an obstacle in the motion space, thus, the path planning strategy presented in Section II can be used by every agent to safely warp obstacles by sliding along a streamline.

We implement the proposed numerical approach over a rectangular motion space defined by

$$\mathcal{P} = \{(X, Y) : X \in [X_{min}, X_{max}], Y \in [Y_{min}, Y_{max}]\} \tag{17}$$

where (X_{min}, Y_{min}), (X_{min}, Y_{max}), (X_{max}, Y_{min}), and (X_{max}, Y_{max}) are positions of the $\mathcal{P}$'s corners. Obstacles are defined by subset $\mathcal{O} \subset \mathcal{P}$, and divided into n_o subsets $\mathcal{O}_1$ through $\mathcal{O}_{n_o}$ $(\mathcal{O} = \mathcal{O}_1 \bigcup \cdots \mathcal{O}_{n_o})$. Obstacle subset $\mathcal{O}_j$ consists of finite number of compact obstacle zones whose Y components of their center of mass are the same and equal to $\bar{Y}_j$. Given $\mathcal{P}$ and $\mathcal{O}$, $\mathcal{N} = \mathcal{P} \setminus \mathcal{O}$ define the navigable space. The navigable space is divided into $n_o + 1$ navigable channels $\mathcal{N}_0$, $\cdots$, $\mathcal{N}_{n_o}$, where

$$\mathcal{N}_j = [X_{min}, X_{max}] \times \left[\bar{Y}_j, \bar{Y}_{j+1}\right) - \mathcal{O}, \qquad j = 0, \cdots, n_o, \tag{18}$$

$\times$ is Cartesian product symbol, $\bar{Y}_0 = Y_{min}$, and $\bar{Y}_{n_o+1} = Y_{max}$.

For better clarification, we consider the available floor area of the SMART lab as a rectangular motion space with $X_{min} = Y_{min} = -2.5$ and $X_{max} = Y_{max} = 2.5$ (See Fig. 3). For the flight experiments, we consider 8 cylinders, based by rectangles and diamonds, as static obstacles as shown in Fig. 9. The obstacles are divided into three group $\mathcal{O}_1$ with $\bar{Y}_1 = -1.15$, $\mathcal{O}_2$ with $\bar{Y}_2 = -0.20$, and $\mathcal{O}_3$ with $\bar{Y}_3 = 1.10$. The four navigable channels $\mathcal{N}_0$, $\mathcal{N}_1$, $\mathcal{N}_2$, and $\mathcal{N}_3$, obtained Eq. (18), are colored in purple, yellow, red, and blue, respectively.

Solution: To obtain ϕ_l and ψ_l values over $\mathcal{N}$, we first define the boundary of navigable channel $\mathcal{N}_j$, that is denoted by $\partial\mathcal{N}_j$, as follows:

$$\partial\mathcal{N}_j = \partial\mathcal{N}_{1,j} \bigcup \partial\mathcal{N}_{2,j} \bigcup \partial\mathcal{N}_{3,j} \bigcup \partial\mathcal{N}_{4,j}, \qquad j = 1, \cdots, n_o, \tag{19}$$

where $\partial\mathcal{N}_{1,j}$, $\partial\mathcal{N}_{2,j}$, $\partial\mathcal{N}_{3,j}$, and $\partial\mathcal{N}_{4,j}$ define the bottom, right, top, and left boundaries of $\mathcal{N}_j$, respectively. We uniformly distribute n_j nodes along boundaries $\partial\mathcal{N}_{2,j}$ and $\partial\mathcal{N}_{4,j}$ while p nodes distributed over the boundaries $\partial\mathcal{N}_{1,j}$ and $\partial\mathcal{N}_{3,j}$ are the same for every navigable channel (for every $j \in \{0, \cdots, n_o\}$). As a result, the planning space is also divided into $n_o + 1$ rectangles denoted by $\mathcal{S}_0$ through $\mathcal{S}_{n_o}$ where $\partial\mathcal{S}_j$ denotes the boundary of the j-th rectangle in the planning space.

We generate a uniform grid of size $p \times n_j$ over $\mathcal{S}_j \bigcup \partial\mathcal{S}_j$, for every $j \in \{0, \cdots, n_o\}$, as shown in Fig. 4. The boundary conditions of the planning space are then defined as follows:

$$X(\phi_l, \psi_l) = \begin{cases} \phi_l & (\phi_l, \psi_l) \in \partial\mathcal{N}_{1,j} \bigcup \partial\mathcal{N}_{3,j} \\ \phi_{min} & (\phi_l, \psi_l) \in \partial\mathcal{N}_{4,j} \\ \phi_{max} & (\phi_l, \psi_l) \in \partial\mathcal{N}_{2,j} \end{cases} \tag{20a}$$

$$Y(\phi_l, \psi_l) = \begin{cases} \psi_l & (\phi_l, \psi_l) \in \partial\mathcal{N}_{2,j} \bigcup \partial\mathcal{N}_{4,j} \\ \psi_{min} & (\phi_l, \psi_l) \in \partial\mathcal{N}_{1,j} \\ \psi_{max} & (\phi_l, \psi_l) \in \mathcal{N}_{3,j} \end{cases} \tag{20b}$$

for $j = 0, \cdots, n_o$, where $\bar{Y}_j \le \psi_l \le \bar{Y}_{j+1}$, $\phi_{min} = X_{min}$, $\phi_{max} = X_{max}$, $\psi_{min} = Y_{min}$, and $\psi_{max} = X_{max}$.

III. Operation Modes

We use the foundations provided in Section II to develop algorithms for implementations SNCF, TVNC, TVC, and SOLE operation modes in this section.

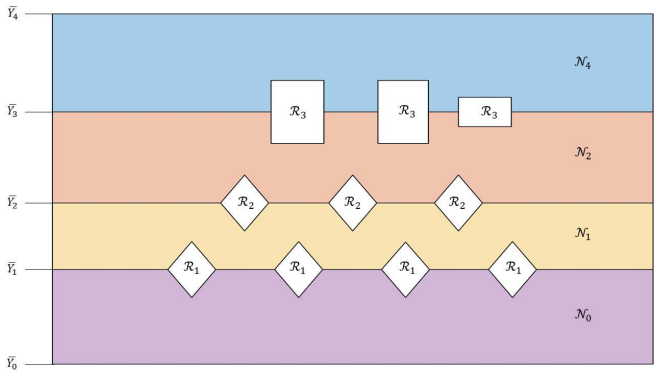

Fig. 3: The SMART lab floor is defined as the motion space $\mathcal{P}$ and divided into four navigable channels using Eq. (18).

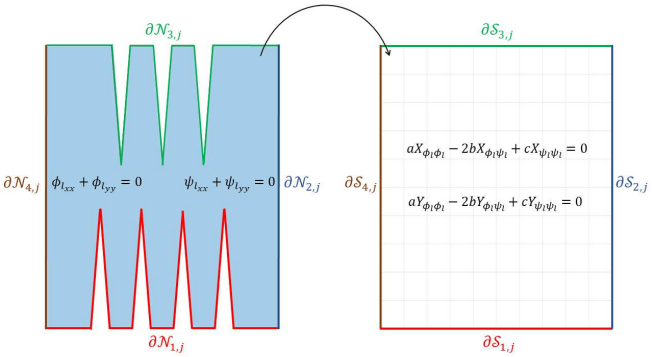

Fig. 4: Transformation between $\mathcal{N}_j$ (the j-th navigable channel) and the $\mathcal{S}_j$.

A. SNCF Navigation

For the SNCF navigation model, set $\mathcal{V}$ is divided into $\mathcal{V}_1$ and $\mathcal{V}_2 = \bar{\mathcal{V}}_1$, where $\mathcal{V}_1(t)$ and $\mathcal{V}_2(t)$ are disjoint subsets of $\mathcal{V}$ defining the "healthy" and "faulty" agents, respectively, at time t.

Definition 1. We define $t_{\text{fail}} \geq t_0$ as the **most recent time** when the status of failure of the agents has changed.

For the SNCF coordination, we make the following assumptions:

Assumption 1. *Angle $\theta_1(t)$ is constant at any time $t \geq t_0$, where t_0 is the start time of the SNCF coordination.*

Assumption 2. *We assume that either the sliding speed v_l, used in Eq.* (3)*, is sufficiently large, or the geometry of the domain enclosing $h \in \bar{\mathcal{V}}_l$ is spacious enough, such that every faulty agent $h \in \mathcal{V}_2$ remains inside a stationary vertical cylinder after its failure is detected.*

For the SNCF, we define potential and stream functions only for the healthy agents. Thus, potential function ϕ_1, stream function ψ_1, β_1, and θ_1 are denoted by ϕ, ψ, β, and θ, respectively, and substituted in Eq. (14) to compute the potential and stream functions under SNCF. By imposing Assumptions 1 and 2, potential function $\phi(x, y, t_{\text{fail}})$ and stream $\psi(x, y, t_{\text{fail}})$ are piece-wise time-invariant and remains spatially-varying at any time $t \geq t_{\text{fail}}$ until the status of agents' failures change. We apply Algorithm 2 to safely plan coordination of healthy agents under the SNCF strategy.

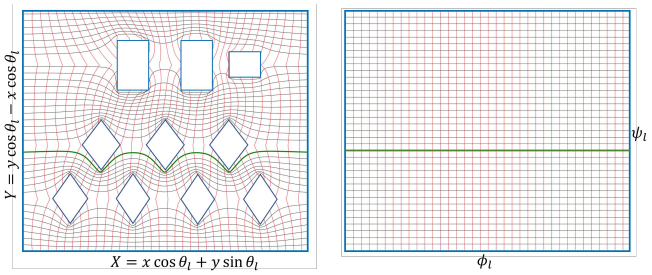

Fig. 5: *Left:* Motion space with streamlines shown by black and potential lines shown by red. *Right:* Planning space. The green curve in the motion space is an streamline used by an agent $i \in \mathcal{V}_l$ to avoid collision with obstacles (the projection of the agent i's path is a horizontal line in the planning space).

Algorithm 2 Algorithm for SNCF Fluid-Flow Navigation.

Get: Initial time t_0, number of time steps denoted by n, time increment Δt, healthy agent set $\mathcal{V}_1(t_0)$, faulty agent set $\mathcal{V}_2(t_0)$, time increment Δt, $\Delta_h = \Delta$ for every $h \in \mathcal{V}_2$, θ_l, v_l, initial position $\mathbf{z}_i(t_0)$ of every healthy agent $i \in \mathcal{V}_1$, and initial position $\mathbf{z}_h(t_0)$ of every faulty agent $h \in \mathcal{V}_2$
Set: $\beta = 1$, $k = 1$, $t_h = t_0$.
while $k < n$ **do**
  if $\mathcal{V}_2(t_k) \neq \mathcal{V}_2(t_{k-1})$ **then**
    $t_{\text{fail}} \leftarrow t_k$
    Update $\phi(x, y, t_{\text{fail}})$.
    Update $\psi(x, y, t_{\text{fail}})$.
  end if
  $\phi(x, y, t_k) \leftarrow \phi(x, y, t_{\text{fail}})$.
  $\psi(x, y, t_k) \leftarrow \psi(x, y, t_{\text{fail}})$.
  Obtain $\mathbf{z}_i(t_{k+1})$ for every $i \in \mathcal{V}_1(t_k)$ by Algorithm 1.
  Return $\mathbf{z}_i(t_{k+1})$.
  $\mathbf{z}_i(t_k) \leftarrow \mathbf{z}_i(t_{k+1})$ for every $i \in \mathcal{V}_1(t_k)$.
  $k \leftarrow k + 1$.
end while

B. TVNC Navigation

For the TVNC navigation model, set $\mathcal{V}$ is divided into time-invariant subsets $\mathcal{V}_1$ and $\mathcal{V}_2 = \bar{\mathcal{V}}_1$, where $\mathcal{V}_1$ and $\mathcal{V}_2$ define "cooperative" and "non-cooperative" agents, respectively. The noncooperative agents have a predefined trajectories in the motion space whereas the cooperative agents uses the navigation model, presented in Algorithm 1, to safely update their positions and reach their target positions.

Similar the SNCF coordination model, we denote potential function ϕ_1, stream function ψ_1, β_1, and θ_1 by ϕ, ψ, β, and θ, respectively, and substitute them in Eq. (14) to compute the potential and stream functions. Let $\mathbf{z}_{i,f} = x_{i,f} + \mathbf{j}y_{i,f}$ be the known target position of cooperative agent $i \in \mathcal{V}_1$, then, then angle θ, assigning the bulk motion direction of $\mathcal{V}_1$ is obtained by

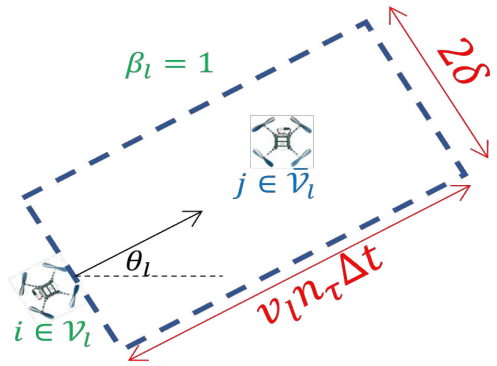

Fig. 6: The virtual box $\mathcal{B}_i$ with side lengths 2δ and $v_l n_\tau \Delta t$ used by agent $i \in \mathcal{V}_l$ to estimate the possibility of colliding an agent $j \in \bar{\mathcal{V}}_l$.

$$\theta(t) = \theta_l = \arg\left(\sum_{i \in \mathcal{V}_1} \left(\mathbf{z}_{i,f} - \mathbf{z}_i(t)\right)\right), \qquad \forall t \geq t_0,\ l \in \mathcal{M}, \quad (21)$$

where t_0 is the initial time. We use Algorithm 3 to safely plan coordination of every agent $i \in \mathcal{V}$ in a shared motion space.

Algorithm 3 Algorithm for TVNC Fluid-Flow Navigation.

Get: Initial time t_0, time increment Δt, $\epsilon > 0$, Δ_h for every $h \in \bar{\mathcal{V}}_l$, θ_l, v_l, initial position $\mathbf{z}_i(t_0)$ of every cooperative agent $i \in \mathcal{V}_1$.
Set: $\beta = 1$, $k = 0$.
while $\sum_{i \in \mathcal{V}_1} \left|\mathbf{z}_i(t_k) - \mathbf{z}_{i,f}\right| > |\mathcal{V}_1|\,\epsilon$ for every $i \in \mathcal{V}_1$ **do**
    Get $\mathbf{z}_h(t_k)$ for every $h \in \mathcal{V}_2$.
    Compute $\theta(t_k)$ by Eq. (21).
    Update $\phi(x, y, \theta(t_k), t_k)$ and $\psi(x, y, \theta(t_k), t_k)$.
    Obtain $\mathbf{z}_i(t_{k+1})$ for every $i \in \mathcal{V}_1$ by Algorithm 1.
    Return $\mathbf{z}_i(t_{k+1})$.
    $\mathbf{z}_i(t_k) \leftarrow \mathbf{z}_i(t_{k+1})$ for every $i \in \mathcal{V}_1$.
    $k \leftarrow k + 1$.
end while

C. TVC Navigation

For the TVC navigation, we enable every agent $i \in \mathcal{V}_l$ to check if there is a possibility of colliding with an agent $h \in \bar{\mathcal{V}}_l$ within the next n_τ time steps. To this end, we define virtual box $\mathcal{B}_i(t) \subset \mathbb{C}$ for every agent $i \in \mathcal{V}_l$, with side lengths 2δ and $v_l n_\tau \Delta t$, to check possibility of collision with an agent $j \in \bar{\mathcal{V}}_l$ within the next $n_\tau \Delta t$ seconds. To formally specify collision avoidance condition, we define condition ζ as follows:

$$\bigvee_{l \in \mathcal{M}} \bigvee_{i \in \mathcal{V}_l} \bigvee_{j \in \bar{\mathcal{V}}_l} \left(\mathbf{z}_j \in \mathcal{B}_i\right), \qquad (\zeta)$$

where "$\bigvee$" is used to specify "at least one". Note that ζ is satisfied, if there exists at least one agent $j \in \bar{\mathcal{V}}_l$ that is inside one of the safety boxes of $\mathcal{V}_l$'s agents.

Therefore, β_l is specified as follows:

$$\zeta \implies \bigwedge_{l \in \mathcal{M}} (\beta_l = 1), \qquad (22a)$$

$$\neg\zeta \implies \bigwedge_{l \in \mathcal{M}} (\beta_l = 0), \qquad (22b)$$

where "$\bigwedge$" is used to specify "include all"; "$\implies$" means "implies that"; and "$\neg$" is the "negation" symbol. For the TVC, we use Algorithm 4 to safely plan coordination of every agent $i \in \mathcal{V}$ in a shared motion space.

Algorithm 4 Algorithm for TVC Fluid-Flow Navigation.

1: *Get:* Initial time t_0, number of sample times denoted by n, time increment Δt, Δ_h for every $h \in \bar{\mathcal{V}}_l$, v_l, δ, n_τ, initial position $\mathbf{z}_i(t_0)$ of every agent $i \in \mathcal{V}_l$ and every cluster $l \in \mathcal{M}$.
2: *Set:* $k = 0$.
3: *Set:* $\beta_l(t_0) = 0, \cdots, \beta_l(t_n) = 0$ for every $l \in \mathcal{M}$.
4: **for** $k \in \{0, \cdots, n\}$ **do**
5: **for** $l \in \mathcal{M}$ **do**
6: **if** ζ is satisfied **then**
7: $\beta_l(t_k) = 1$.
8: Get $\mathbf{z}_h(t_k)$ for every $h \in \bar{\mathcal{V}}_l$.
9: **end if**
10: Compute $\theta_l(t_k)$ by Eq. (21).
11: Update $\phi_l(x, y, \theta(t_k), t_k)$ and $\psi_l(x, y, \theta(t_k), t_k)$.
12: Obtain $\mathbf{z}_i(t_{k+1})$ for every $i \in \mathcal{V}_l$ by Algorithm 1.
13: Return $\mathbf{z}_i(t_{k+1})$.
14: $\mathbf{z}_i(t_k) \leftarrow \mathbf{z}_i(t_{k+1})$ for every $i \in \mathcal{V}_l$.
15: **end for**
16: $k \leftarrow k + 1$.
17: **end for**

D. SOLE Navigation

For the SOLE navigation, we consider operation of a single agent team in an obstacle-laden environment, thus, $m = 1$ and $\mathcal{V} = \mathcal{V}_1 = \{1, \cdots, N\}$ defines the identification numbers of the agents, $\phi(x, y) = \phi_1(x, y)$ and $\psi(x, y) = \psi_1(x, y)$ denote the potential and stream function, and $\theta = \theta_1$ is constant. We propose the Algorithm 5 to obtain safe trajectories of every $i \in \mathcal{V}$ by following the motion strategy presented in Section II. Note that "Cθ" and "Sθ" in line 15 of Algorithm 5 stand for "$\cos\theta$" and "$\sin\theta$", respectively.

IV. Experiments and Discussion

We experimentally evaluate the performance of the proposed algorithms and validate the results on a group of tiny quadcopters (The multimedia of our experiments is available at YouTube (Link)).

The experimental setup includes 4 major components shown in Fig. 7: (i) Motion Capture System (MCS), (ii) Ground Control Station (GCS), (iii) Crazyradio PA, and (iv) Crazyflie 2.1. The MCS captures the position of the each crazyflie in 3-D space and sends the information to the GCS through an ethernet cable at 100Hz. The GCS is an Intel i7 11-th gen desktop, with 16 GB of RAM running Ubuntu 20.04 and ROS Noetic. GCS is also installed with the Crazyswarm [25] ROS stack built by USC-ACT Lab and acts as a centralized planner for the system. The GCS uses the information from MCS to compute the desired states for each crazyflie and then transmits the data to the onboard

Algorithm 5 Algorithm for SOLE Fluid-Flow Navigation

Get: Initial time t_0, number of sample times denoted by n, time increment Δt, O_1 through O_{n_o}, initial positions $\mathbf{z}_{i,0} = x_{i,0} + \mathbf{j}y_{i,0}$ of every $i \in \mathcal{V}$, ϕ_{min}, ϕ_{max}, p, and θ.
Set: $k = 0$.
$\Delta\phi = (\phi_{max} - \phi_{min})/p$.
Determine navigable channels by Eq. (18).
Specify boundary conditions using Eq. (20).
Obtain $X(\phi,\psi)$ and $Y(\phi,\psi)$ over $\mathcal{P}$ numerically, by solving Eq. (16).
Obtain $X_{i,0} = x_{i,0}\cos\theta + x_{i,0}\sin\theta$ for every $i \in \mathcal{V}$.
Obtain $Y_{i,0} = y_{i,0}\cos\theta - x_{i,0}\sin\theta$ for every $i \in \mathcal{V}$.
Compute associated $(\phi_i(t_0), \psi_i(t_0)) \in \mathcal{S}$ for every $i \in \mathcal{V}$.
for $k \in \{0, \cdots, n\}$ **do**
  for $i \in \mathcal{V}$ **do**
    $\phi_i(t_{k+1}) \leftarrow \phi_i(t_k) + \Delta\phi$.
    $\psi_i(t_{k+1}) \leftarrow \psi(t_k)$.
    Obtain (X_i, Y_i) associated with (ϕ_i, ψ_i).
    $\mathbf{z}_i(t_{k+1}) \leftarrow (X_i C\theta - Y_i S\theta) + \mathbf{j}(X_i S\theta + Y_i C\theta)$.
    Return $\mathbf{z}_i(t_{k+1})$.
    $\mathbf{z}_i(t_k) \leftarrow \mathbf{z}_i(t_{k+1})$ for every $i \in \mathcal{V}$.
  end for
  $k \leftarrow k + 1$.
end for

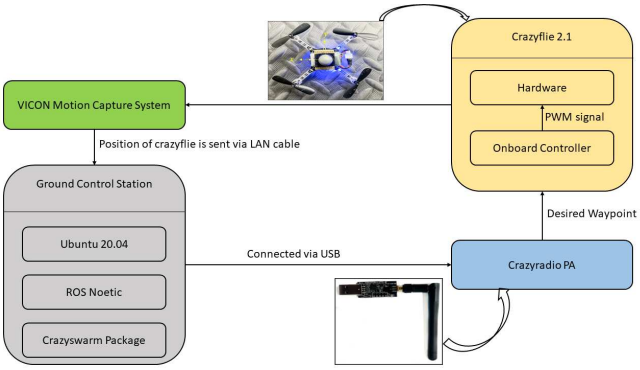

Fig. 7: An overview of the experiment setup.

controller through Crazyradio PA. We conducted flight tests at the University of Arizona's Scalable Move and Resilient Transversability (SMART) lab's indoor flying area with a volume of 5m × 5m × 2m equipped with 8 VICON motion capture cameras. We assume that all crazyflies are flying at the altitude of 1 m.

A. Stationary Non-Concurrent Failures (SNCF) Experiment

For this experiment, we follow the approach presented in Section III-A where Δ_h = 0.4m (Eq. (14)). The CFs are uniquely identified using the set $\mathcal{V} = \{1, \cdots, 6\}$. As indicated in Table I, we have $m = 2$. At time t_0, $\mathcal{M} = \{1, 2\}$, $\mathcal{V}$ is divided into $\mathcal{V}_1 = \{1, \cdots, 6\}$ and $\mathcal{V}_2 = \emptyset$. Until the first failure, the CFs all move together represented as solid lines (See Fig. 8(a)). At t_{fail_1} = 2s, CF4, chosen randomly, is subjected to failure and is wrapped by a green cylinder representing the unsafe zone (See Fig. 8(a)). At this instant, $\mathcal{V}_1 = \{1, 2, 3, 5, 6\}$ and $\mathcal{V}_2 = \mathcal{V} \setminus \mathcal{V}_1 = \{4\}$. The desired paths for all CFs belonging to $\mathcal{V}_1$ are computed based on algorithm 2. We deploy another failure, CF5, in the system at t_{fail_2} = 12s (See Fig. 8(b)). The sets $\mathcal{V}_1$ and $\mathcal{V}_2$ are updated: $\mathcal{V}_1 = \{1, 2, 3, 6\}$ and $\mathcal{V}_2 = \{4, 5\}$. The desired paths for the healthy CFs are again computed until $\mathcal{V}_1$'s CFs have completely passed the unsafe-zones (See Fig. 8(b)).

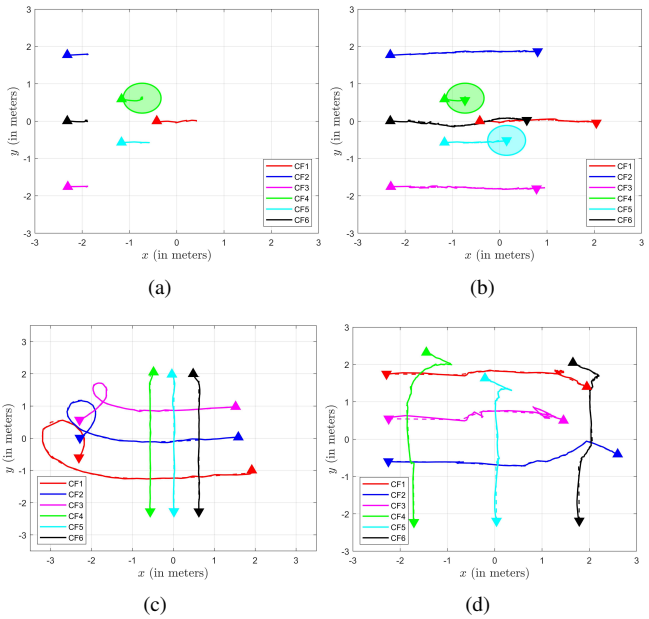

Fig. 8: (a) Location of healthy CFs at the time of first failure. The green circle corresponds to the unsafe-zone of CF4. (b) Desired (dashed line) versus actual (solid line) paths tracked by CF to avoid unsafe-zones. (c) Desired (dashed line) vs actual (solid line) paths undertaken by CFs. We can see $\mathcal{V}_1$ taking advantage of the recovery algorithm in order to avoid any collision with non-cooperative CFs in set $\mathcal{V}_2$. (d) Desired (dashed line) vs actual (solid line) paths tracked by CFs using algorithm 4 when $N = 6$.

B. Time-Varying Non-Cooperative (TVNC) Experiment

In this experiment, we evaluate the performance of the algorithm 3 proposed in Section III-B using $m = 2$ groups of crazyflies. Specifically, at time t_0, the set $\mathcal{V} = \{1, \cdots, 6\}$ is divided into time-invariant subsets $\mathcal{V}_1 = \{1, 2, 3\}$ and $\mathcal{V}_2 = \{4, 5, 6\}$. The aim of agents in $\mathcal{V}_2$, known as non-cooperative agents, is to reach their goal locations quickly. Therefore, trajectories of $\mathcal{V}_2$'s agents are predefined, as indicated in Fig. 8(c) (See the green, cyan, and black paths). However, the agents belonging to $\mathcal{V}_1$, termed as cooperative agents, use the fluid-flow navigation function to safely plan paths in the shared motion space. As shown in Fig. 8(c), cooperative CFs 1, 2, and 3 reach their target locations by following the red, blue, and pink paths.

C. Time-Varying Cooperative (TVC)

According to Section III-C and 4, we define the set $\mathcal{V} = \{1, \cdots, 6\}$ to uniquely identify all CFs. CFs are divided into two groups identified by $\mathcal{V}_1 = \{1, 2, 3\}$ and $\mathcal{V}_2 = \{4, 5, 6\}$. For this experiment, we choose v_l = 0.3m/s, $\delta = 0.15$, and

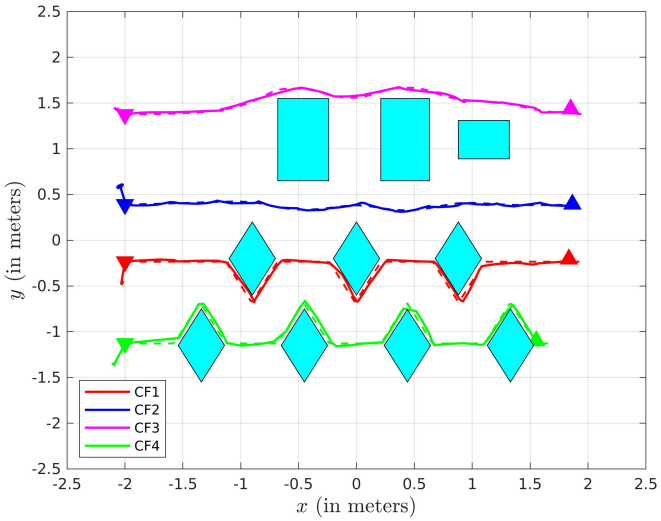

Fig. 9: Desired (dashed line) vs Actual (solid line) paths tracked by crazyflies using Algorithm when $N = 4$.

$n_\tau = 3$. We have experimentally evaluated the scenario when θ is varying in time but constant for each individual group of agents. This approach ensures that each agent in a group move parallel to other agents in the group. Figure 8(d) plots the result of our experiment.

D. Stationary Obstacle-Laden Environment (SOLE) Experiment

In this experiment, we have an obstacle-laden environment as shown in Figure 9. We follow the approach presented in Section III-D and define the set $\mathcal{V} = \{1,2,3,4\}$. By implementing Algorithm 5, CF quadcopters 1 through 4 follow the paths shown in Fig. 9 to safely pass through obstacles.

V. CONCLUSION

In this work, we proposed multiple recovery algorithms based on ideal fluid-flow for collision-free coordination between multiple groups of agents. Our algorithms were able to handle different scenarios including stationary non-concurrent failures, time-varying non-cooperative failures, and time-varying cooperative failures. Experimental results using teams of crazyflies displayed the advantage of our proposed algorithms in handling different situations. Future work on this direction include incorporating reinforcement learning techniques.

REFERENCES

[1] D. S. Drew, "Multi-agent systems for search and rescue applications," *Current Robotics Reports*, vol. 2, no. 2, pp. 189–200, 2021.

[2] L. F. Oliveira, A. P. Moreira, and M. F. Silva, "Advances in forest robotics: A state-of-the-art survey," *Robotics*, vol. 10, no. 2, p. 53, 2021.

[3] J. J. Acevedo, B. C. Arrue, I. Maza, and A. Ollero, "A decentralized algorithm for area surveillance missions using a team of aerial robots with different sensing capabilities," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 4735–4740.

[4] V. Sunkara, A. Chakravarthy, and D. Ghose, "Collision avoidance of arbitrarily shaped deforming objects using collision cones," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2156–2163, 2019.

[5] H. G. Tanner and A. Boddu, "Multiagent navigation functions revisited," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1346–1359, 2012.

[6] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE international conference on robotics and automation*. Ieee, 2008, pp. 1928–1935.

[7] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Optimal reciprocal collision avoidance for multi-agent navigation," in *Proc. of the IEEE International Conference on Robotics and Automation, Anchorage (AK), USA*, 2010.

[8] F. R. Inácio, D. G. Macharet, and L. Chaimowicz, "United we move: Decentralized segregated robotic swarm navigation," in *Distributed Autonomous Robotic Systems: The 13th International Symposium*. Springer, 2018, pp. 313–326.

[9] B. Ichter, J. Harrison, and M. Pavone, "Learning sampling distributions for robot motion planning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7087–7094.

[10] G. Knizhnik, P. Li, M. Yim, and M. A. Hsieh, "Flow-based rendezvous and docking for marine modular robots in gyre-like environments," 2023.

[11] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.

[12] M.-H. Kim, J.-H. Heo, Y. Wei, and M.-C. Lee, "A path planning algorithm using artificial potential field based on probability map," in *2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. IEEE, 2011, pp. 41–43.

[13] F. Chen, P. Di, J. Huang, H. Sasaki, and T. Fukuda, "Evolutionary artificial potential field method based manipulator path planning for safe robotic assembly," in *2009 International Symposium on Micro-NanoMechatronics and Human Science*. IEEE, 2009, pp. 92–97.

[14] O. Montiel, U. Orozco-Rosas, and R. Sepúlveda, "Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles," *Expert Systems with Applications*, vol. 42, no. 12, pp. 5177–5191, 2015.

[15] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, 2018.

[16] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, "Control barrier certificates for safe swarm behavior," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015.

[17] Y. Chen, A. Singletary, and A. D. Ames, "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 127–132, 2020.

[18] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 6271–6278.

[19] H. Rastgoftar and E. Atkins, "Physics-based freely scalable continuum deformation for uas traffic coordination," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 2, pp. 532–544, 2019.

[20] H. Rastgoftar, "Fault-resilient continuum deformation coordination," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 1, pp. 423–436, 2020.

[21] H. Uppaluru, H. Emadi, and H. Rastgoftar, "Resilient multi-uas coordination using cooperative localization," *Aerospace Science and Technology*, vol. 131, p. 107960, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963822006344

[22] M. Romano, H. Uppaluru, H. Rastgoftar, and E. Atkins, "Quadrotor formation flying resilient to abrupt vehicle failures via a fluid flow navigation function," *arXiv preprint arXiv:2203.01807*, 2022.

[23] K. A. Hoffmann and S. T. Chiang, "Computational fluid dynamics for engineers, vol. i," *Wichita, Engineering Education System*, pp. 124–137, 1993.

[24] H. Emadi, H. Uppaluru, and H. Rastgoftar, "A physics-based safety recovery approach for fault-resilient multi-quadcopter coordination," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 2527–2532.

[25] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3299–3304.