Scalable, Pairwise Collaborations in Heterogeneous Multi-Robot Teams

Alexander A. Nguyen, *Graduate Student Member, IEEE*, Luis Guerrero-Bonilla, *Member, IEEE*, Faryar Jabbari, *Senior Member, IEEE*, and Magnus Egerstedt, *Fellow, IEEE*

***Abstract*—This letter introduces a finite state machine (FSM) for encoding collaborative interactions among robots. The resulting novel architecture is particularly designed with heterogeneous multi-robot teams in mind, where pairwise collaborative arrangements can result in new capabilities for the participants. To ensure scalability, the proposed FSM's complexity does not depend on the overall team size for individuals' decisions. Additionally, we explore various selection strategies to facilitate the pairing of robots and demonstrate the framework's efficacy on a team of mobile robots with tasks requiring collaboration for their successful completion.**

***Index Terms*—Heterogeneous multi-robot teams, collaboration, barrier functions, finite state machines.**

I. Introduction

HETEROGENEOUS multi-robot systems can be found in a multitude of settings, including applications such as cooperative transportation [1], coverage control [2], environmental monitoring [3], or search and rescue [4]. Typically, heterogeneous teams are deployed to achieve complex objectives that require resilient and distributed solutions [5].

The standard approach to heterogeneous robot teams is to divide them into subteams based on their capabilities and then have these teams coordinate their activities with each other to try to complete the overall mission [6]. This approach works well in many scenarios, but the arrangement does not typically lead to truly new capabilities in that the robots do not obtain new functionalities through collaboration that did not already manifest individually. However, heterogeneous robots can acquire new capabilities when working together, as observed in [7], [8]; e.g., a strong robot could help launch a small robot through a window, or an amphibious vehicle could ferry a ground robot across a waterway.

One way to capture the capabilities of various types of robots is through control barrier functions (CBFs), whose purpose is to ensure that a dynamical system remains within a safe set [9]. Through collaborative arrangements, these sets can be made to expand as a function of the new capabilities [10]. If a robot finds itself in a situation where help is required, e.g., to traverse a region it could not by itself, it needs to expand its safe set through collaboration and must recruit a suitable teammate to make this happen. One way to approach this type of pairwise collaborative scenario is to enumerate all possible collaborations each robot might participate in. But that would result in a decision-making strategy whose complexity grows with the total team size, which should be avoided in large teams [11].

We approach this problem of scalable, pairwise interactions through a finite state machine (FSM) model [12]. Such FSMs have been extensively employed as decision-making formalisms to encode intricate robot behaviors [13], [14], [15]. To that end, we propose a novel and scalable architecture for pairwise collaborations in teams of heterogeneous robots, employing an FSM for decision-making together with a centralized queue to assign pairwise collaboration partners.

This letter builds upon the work developed in [10], where the idea of encoding pairwise collaborations through CBFs was introduced. The main contribution of this letter is in the proposed architecture – an FSM design that results in scalable decision-making, i.e., each individual's FSM is independent of the overall team's size, and a centralized queue, termed the "help queue", to form suitable collaborative arrangements between help-needing and help-providing robots based on the preferred selection prioritization strategy of a user.

The remainder of this letter is organized as follows: Section II shows how one can enumerate all pairwise collaborative interactions for robots in a team using FSMs, together with the relevant background material on multi-robot collaboration using barrier functions. Section III provides a scalable decision model that utilizes an FSM for encoding pairwise collaborations between heterogeneous robots. Section IV explores different selection prioritization strategies using a centralized queue to match robots for pairwise collaborations. Section V demonstrates the proposed collaborative control architecture on a team of mobile robots, and Section VI contains concluding remarks.

Manuscript received 8 March 2024; revised 30 April 2024; accepted 8 May 2024. Date of publication 13 May 2024; date of current version 28 May 2024. This work was supported in part by the U.S. Office of Naval Research under Grant N00014-22-1-2625, and in part by the Graduate Assistance in Areas of National Need Fellowship under Grant P200A210021. Recommended by Senior Editor M. Guay. *(Corresponding author: Alexander A. Nguyen.)*

Alexander A. Nguyen, Faryar Jabbari, and Magnus Egerstedt are with the Samueli School of Engineering, University of California at Irvine, Irvine, CA 92697 USA (e-mail: alexaan2@uci.edu; fjabbari@uci.edu; magnus@uci.edu).

Luis Guerrero-Bonilla is with the School of Engineering and Sciences, Tecnológico de Monterrey, Monterrey 64700, Mexico (e-mail: lguerrero@tec.mx).

Digital Object Identifier 10.1109/LCSYS.2024.3400702

II. Background

This section shows how FSMs can be applied to describe pairwise collaborations between robots, together with the relevant background on barrier functions for use in scenarios of multi-robot collaboration, as was done in [10].

A. Enumerating Pairwise Collaborations

Consider the setting of N robots placed into M distinct categories based on capability, where each robot can accomplish a specific task in its safe operating region. Safety, as will be further elaborated on in Section II-B, defines the part of the state space where a robot can operate without help, e.g., the regions that a robot can traverse effectively by itself. Moreover, by exploiting the capabilities of others, individual robots can expand the areas in which they can operate safely by collaborating. What collaboration entails is the ability to enhance an individual's capabilities by exploiting the functionalities of another robot type. For example, an aquatic robot can ferry a ground robot across water, whereas a ground robot can carry an aquatic robot over land.

Given this setup, the pairwise collaborations between heterogeneous robots can be encoded through an FSM – represented as a 5-tuple (E, Ξ, ξ_0, δ, Π) [12]; where E is the set of events (finite and non-empty), Ξ is the set of discrete states (finite and non-empty), $\xi_0 \in \Xi$ is the initial discrete state, $\delta : \Xi \times E \to \Xi$ is the discrete state transition function which provides a map from the current to a new discrete state within the FSM, and $\Pi \subseteq \Xi$ is the set of final (or terminal) discrete states (finite and possibly empty), where no further discrete state transitions are performed by the FSM once $\xi \in \Pi$.

Consider an example scenario with N robots placed into $M = 2$ capability categories. Specifically, we consider n ground robots (abbreviated by 'g') and m aquatic robots (abbreviated by 'a') such that $N = n + m$. In Fig. 1, we enumerate all possibilities for an aquatic robot, i.e., every individual or pairwise collaboration discrete state, where the discrete states that the robot i can operate in are:

- W_i, 'Wait': Robot i waits for a task to be assigned;
- I_i, 'Individual Task': Robot i attempts to complete its task by itself;
- S_i^j, 'Stuck': Robot i is unable to progress towards their task, so it awaits the help of robot j;
- R_i^j, 'Receiving Help': Robot i receives help from robot j;
- A_j^i, 'Approach': Robot i approaches robot j, who is unable to progress towards their task, to help it;
- P_j^i, 'Providing Help': Robot i provides help to robot j.

Here, we adopt the notational convention that superscripts and subscripts of the discrete states denote the robot providing and receiving help, respectively. Furthermore, for each robot's FSM, such as the one outlined in Fig. 1, we designate W_i as the initial discrete state and the only final discrete state, i.e., $\xi_0 = W_i$ and $\Pi = \{W_i\}$, respectively.

The FSM's discrete states can be related to the collaboration framework in [10], which proposed three high-level operating modes: 'Individual Tasks' (mode q_1; robots attempt to complete their tasks independently), 'Collaboration Setup' (mode q_2; robots prepare themselves for pairwise collaboration), and 'Collaborative Act' (mode q_3; robots work together such that a robot can expand its safe operating region). In particular, mode q_1 corresponds to the discrete state I_i, while mode q_2 corresponds to the discrete states S_i^j and A_j^i, and mode q_3 to the discrete states R_i^j and P_j^i for robot i.

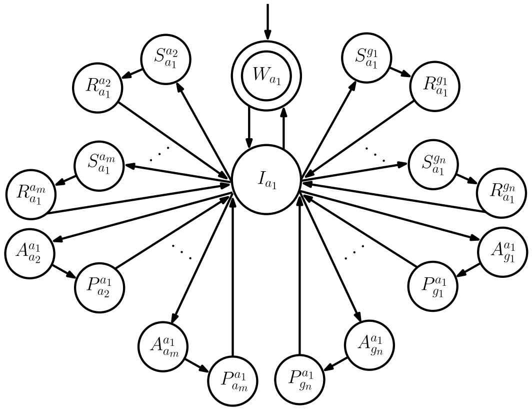


Fig. 1. Aquatic robot 1's FSM: Enumerates all possibilities, every individual and pairwise collaboration discrete state, for the scenario of N robots placed into aquatic or ground ($M = 2$) capability categories.

The FSM complexity for the aquatic robot in Fig. 1 and a ground robot, whose FSM can be defined similarly to Fig. 1, is taken as the total number (cardinality, $|\cdot|$) of discrete states [16], given by

$$\begin{aligned}|\Xi_{a_1}| &= 2 + 2n + 2n + 2(m-1) + 2(m-1), \\ &= 4N - 2,\end{aligned} \tag{1}$$

$$\begin{aligned}|\Xi_{g_1}| &= 2 + 2m + 2m + 2(n-1) + 2(n-1), \\ &= 4N - 2,\end{aligned} \tag{2}$$

respectively. As such, we have

$$|\Xi_{a_1}| = \mathcal{O}(N), \tag{3}$$

$$|\Xi_{g_1}| = \mathcal{O}(N), \tag{4}$$

i.e., linear complexity. In other words, each robot's FSM complexity is a function of the overall team size. Moreover, the individual robots must perform their selection computations locally, which is also a function of the overall team size. However, this is not a desirable quality in scalable algorithms [11]. To that end, what is needed is an alternative formulation to overcome this issue, which we will discuss in Section III.

B. Modes of Operation

We have just seen how the decision-making mechanism of the individual robots can be captured by an FSM, albeit, currently, one that exhibits a too great complexity. However, before introducing the new, scalable FSM design, we must first formally define the robots' discrete states, $\xi \in \Xi$, and events, $e \in E$, using barrier functions, as was similarly done in [10].

Assume, as before, that there are N robots, with $\mathcal{N} = \{1, \ldots, N\}$ (index set of robots), placed into M different capability categories. We suppose that the dynamics of each robot $i \in \mathcal{N}$ is given by

$$\dot{x}_i = f_{i,q_v}(x_i) + g_{i,q_v}(x_i)u_i, \tag{5}$$

where $x_i \in \mathcal{X}_i \subset \mathbb{R}^{d_i}$ is the (continuous) state and $u_i \in \mathcal{U}_i \subset \mathbb{R}^{p_i}$ is the control input. Additionally, $q_v \in \{q_1, q_2, q_3\}$ are the collaboration framework's high-level operating modes, and $f_{i,q_v}(x_i)$, $g_{i,q_v}(x_i)$ are locally Lipschitz vector fields that can exhibit a different structure within each operating mode; e.g., the dynamics of a ground robot and an aquatic robot working independently may differ from when a ground robot carries an aquatic robot.

Moreover, when viewed in isolation, robot i can operate effectively in its safe region, $\mathcal{S}_i$; the zero-superlevel set to a continuously differentiable function, $h_i : \mathcal{X}_i \rightarrow \mathbb{R}$, known as a control barrier function (CBF) [9]. For our problem setting, CBFs are used to capture the safe regions that robot i can operate in by itself. Then, as long as robot i is initialized within its safe set, i.e., $x_i(t_0) \in \mathcal{S}_i$, forward invariance of the safe set can be ensured, provided that

$$\dot{h}_i(x_i, u_i) \geq -\alpha(h_i(x_i)) \tag{6}$$

is satisfied for all time, where $\alpha(\cdot)$ is an extended class $\mathcal{K}_\infty$ function [17].

However, the safe operating region of robot i can expand due to the presence of other robots. By assuming robot interactions are pairwise, the influence that robot j at x_j has on robot i at x_j can be captured through the pairwise barrier function $h_{ij}(x_i, x_j)$. The use of pairwise CBFs is not new, e.g., collision avoidance [18], [19], but they are defined differently here. In particular, $h_{ij}(x_i, x_j)$ is positive if the collaboration between robots is potentially helpful in the states x_i and x_j when robot i belongs to its unsafe region, i.e., $h_i(x_i) < 0$.

The individual and pairwise barrier functions can then be composed to form a new barrier function, where robot i can be made safe due to the help of robot j, given by

$$H_i\big(x_i, x_j\big) = h_i(x_i) + h_{ij}\big(x_i, x_j\big), \tag{7}$$

as done in [10].

The new safe set for robot i, which includes the regions that a particular robot j can help robot i, is defined as

$$\mathcal{S}_{ij} = \{x_i \in \mathcal{X}_i \mid \exists x_j \in \mathcal{X}_j \ s.t. \ H_i\big(x_i, x_j\big) \geq 0\} \subseteq \mathcal{X}_i. \tag{8}$$

The existential quantifier, $\exists$, encodes that if a particular robot j is in the state x_j, the corresponding state x_i for robot i would be rendered safe thanks to robot j's help. The interpretation is that a region can be made safe if a suitable collaborative arrangement can be identified.

Next, we introduce the nominal dynamics of robot i, i.e., the behavior it would exhibit when attempting to accomplish its objective within mode q_1 ('Individual Tasks') while ignoring safety constraints, given by

$$\dot{x}_{i,\text{nom}} = f_{i,q_1}(x_i) + g_{i,q_1}(x_i)\hat{u}_i, \tag{9}$$

where $\hat{u}_i$ is robot i's nominal controller.

For each high-level operating mode, it is assumed that the individual robots, either help-providing or help-needing, have a nominal controller, $\hat{u}_i$, designed to have them progress towards an objective without considering any safety constraints. For example, $\hat{u}_i$ when $\xi_i = P_j^i$ ('Providing Help') may differ from $\hat{u}_i$ when $\xi_i = R_i^j$ ('Receiving Help') in mode q_3.

Now, we define the event-triggered collaboration signal condition, given by

$$\langle \nabla h_i(x_i), \dot{x}_{i,\text{nom}} \rangle < 0 \wedge h_i(x_i) = 0, \tag{10}$$

which enables robot i to discern if it requires assistance. Namely, (10) provides a robot with the ability to check if it would enter an unsafe region when following its nominal dynamics, $\dot{x}_{i,\text{nom}}$, while on the boundary of its safe set, $\partial\mathcal{S}_i$.

At last, we can formally describe robot i's discrete states, $\xi_i \in \Xi_i$, from the FSM discussed in Section II-A, to be

$$W_i : \tau_i = 0 \wedge h_i(x_i) \geq 0; \tag{11}$$

$$I_i : \tau_i = 1 \wedge h_i(x_i) \geq 0; \tag{12}$$

$$S_i^j : \tau_i = 1 \wedge \tau_j = 1 \wedge h_i(x_i) = 0 \wedge h_j\big(x_j\big) \geq 0 \ \wedge \ h_{ij}\big(x_i, x_j\big) = 0 \ \wedge \langle \nabla h_i(x_i), \dot{x}_{i,\text{nom}} \rangle < 0; \tag{13}$$

$$R_i^j : \tau_i = 1 \wedge \tau_j = 1 \wedge h_i(x_i) < 0 \wedge h_j\big(x_j\big) \geq 0 \ \wedge \ h_{ij}\big(x_i, x_j\big) > 0 \wedge H_i\big(x_i, x_j\big) \geq 0; \tag{14}$$

$$A_j^i : \tau_i = 1 \wedge \tau_j = 1 \wedge h_i(x_i) \geq 0 \wedge h_j\big(x_j\big) = 0 \ \wedge \ h_{ji}\big(x_j, x_i\big) = 0 \ \wedge \langle \nabla h_j\big(x_j\big), \dot{x}_{j,\text{nom}} \rangle < 0; \tag{15}$$

$$P_j^i : \tau_i = 1 \wedge \tau_j = 1 \wedge h_i(x_i) \geq 0 \wedge h_j\big(x_j\big) < 0 \ \wedge \ h_{ji}\big(x_j, x_i\big) > 0 \wedge H_j\big(x_j, x_i\big) \geq 0; \tag{16}$$

where $\tau_i \in \{0, 1\}$ is a logical variable representing whether or not robot i has an assigned task. Moreover, for the discrete states A_j^i and P_j^i (or S_i^j and R_i^j), we let $\tau_i = 1$ (or $\tau_j = 1$) if there is an unaccomplished objective or a pairwise collaboration assignment for robot i (or robot j).

As previously mentioned, the discrete states are constituents of the collaboration framework's high-level operating modes: $q_v \in \{q_1, q_2, q_3\}$. That is, $\xi_i \in \{I_i\}$ for mode q_1 ('Individual Tasks'); $\xi_i \in \{S_i^j, A_j^i\}$ for mode q_2 ('Collaboration Setup'); and $\xi_i \in \{R_i^j, P_j^i\}$ for mode q_3 ('Collaborative Act'). For instance, robot i will transition from mode q_1 to q_2, i.e., from I_i to S_i^j or from I_i to A_j^i, when either the conditions in (13) or (15) hold; robot i will transition from mode q_2 to q_3, i.e., from S_i^j to R_i^j or from A_j^i to P_j^i, when either the conditions in (14) or (16) hold; and robot i will transition from mode q_3 to q_1, i.e., from R_i^j to I_i or from P_j^i to I_i, when the conditions in (12) hold.

It is important to note that the pairwise partners undertaking a collaborative endeavor must first be identified before robots can operate in either mode q_2 or q_3. Section IV provides details on how such robot pairings for collaboration can be made.

To guarantee the robots remain safe for all time, we impose a certificate of safety, which, for robot i, is given by

$$\dot{H}_i(x, u) \ \geq -\alpha(H_i(x)), \tag{17}$$

where $\dot{H}_i(x, u) = L_f H_i(x) + L_g H_i(x) u$ with $L_f H_i(x) = \nabla H_i(x) \cdot f_{q_v}(x)$ and $L_g H_i(x) = \nabla H_i(x) \cdot g_{q_v}(x)$ using Lie derivative notation. Here, $x = [x_1^\top, \ldots, x_N^\top]^\top$ and $u = [u_1^\top, \ldots, u_N^\top]^\top$ are the stacked states and inputs, respectively.

A safety-critical controller, which guarantees each robot remains forward invariant within its safe set while attempting to track their nominal controller, $\hat{u}_i$, as closely as possible, is formulated as a Quadratic Program (QP) [17], given by

$$u^* = \arg\min_u \ \frac{1}{2} \sum_{i=1}^{N} \|u_i - \hat{u}_i\|_2^2 \quad \text{s.t.} \ \ a_i(x) u \leq b_i(x), \quad \forall i \in \mathcal{N} \tag{18}$$

where we have a linear constraint, enabling collaborative interactions to take place between robots, which defines the

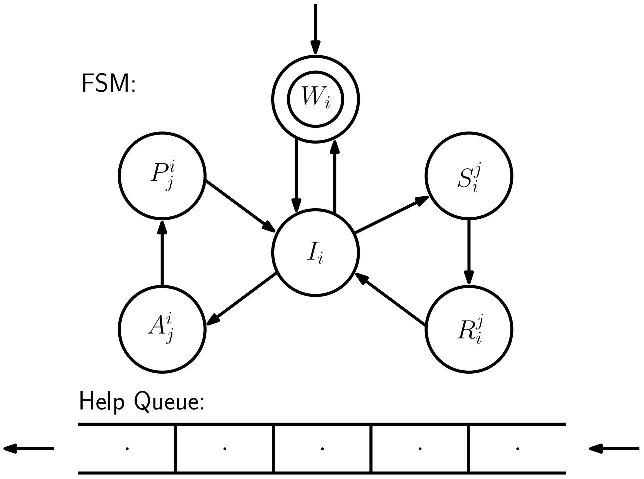


Fig. 2. Dispatch Model: Utilizes a finite state machine for decision-making and a centralized queue for matching.

half-space $a_i(x) = -L_g H_i(x)$ and $b_i(x) = L_f H_i(x) + \alpha(H_i(x))$. In addition, the actual and nominal control signals are denoted as u_i (decision variable) and $\hat{u}_i$, respectively.

III. Scalable Decision Model

This section describes the scalable decision model proposed to encode pairwise collaborations using a centralized queue.

In light of complexity being $\mathcal{O}(N)$ for the decision-making architecture presented in Fig. 1, we propose the "dispatch model" – a FSM design that encodes pairwise collaborations in a scalable fashion by establishing a centralized queue, termed the "help queue", as portrayed in Fig. 2.

The dispatch model's FSM contains six discrete states, rather than $4N - 2$ discrete states when all possibilities are enumerated, in which a robot can operate. Namely, we have $\Xi_i = \{W_i, I_i, S_i^j, R_i^j, A_j^i, P_j^i\}$, along with $\delta_0 = W_i$ and $\Pi \in \{W_i\}$, defined as in Section II-A. Then, as before, the discrete state transition function, $\delta(\xi_i, e_i)$, maps to new discrete states, $\xi_i \in \Xi_i$, whenever a particular event, $e_i \in E_i$, triggers a transition that is dependent on the conditions in (11)–(16), as discussed in Section II-B.

To ensure the FSM design is scalable, we employ a help queue – a key component of the dispatch model – enabling the individual robots' discrete states to avoid growing with the overall team's size. Since the help queue is centralized, each robot can add and remove information stored on it and maintain continual communication with it, which facilitates the dispatch of help to robots requesting assistance.

If a robot requires assistance, it is paired with a suitable partner by the help queue, which performs much of the computational and communication overheads, where we assume the help queue has the resources, i.e., processing power and bandwidth, needed to perform such overheads. Thus, with the help queue established, the individual robots will have their decision-making and selection computations, together with communication, decoupled from the overall team's size. Whereas, without the help queue established, each individual requires continual communication with all other robots, as in Fig. 1, and would need to perform its decision-making and selection computations locally.

The FSM complexity for robot i in the dispatch model (Fig. 2), can be obtained by computing the total number of discrete states (cardinality, $|\cdot|$) [16], given by

$$|\Xi_i| = 6, \tag{19}$$

Then, we have

$$|\Xi_i| = \mathcal{O}(1), \tag{20}$$

i.e., constant complexity – no matter a robot's capability type. Thus, the cardinality of each robot's discrete states is independent of the team's size.

As such, the dispatch model results in scalable decision-making, even as the number of robots increases, due to having a centralized queue to decouple the team's size from computational and communication overheads. Thus, the dispatch model lends itself well to large-sized teams implementing pairwise control strategies. However, now arises the issue of how to employ the help queue to pair robots together for collaboration. To that end, we provide potential selection prioritization strategies that leverage the help queue to form collaborative partnerships.

IV. Selection Prioritization Strategies

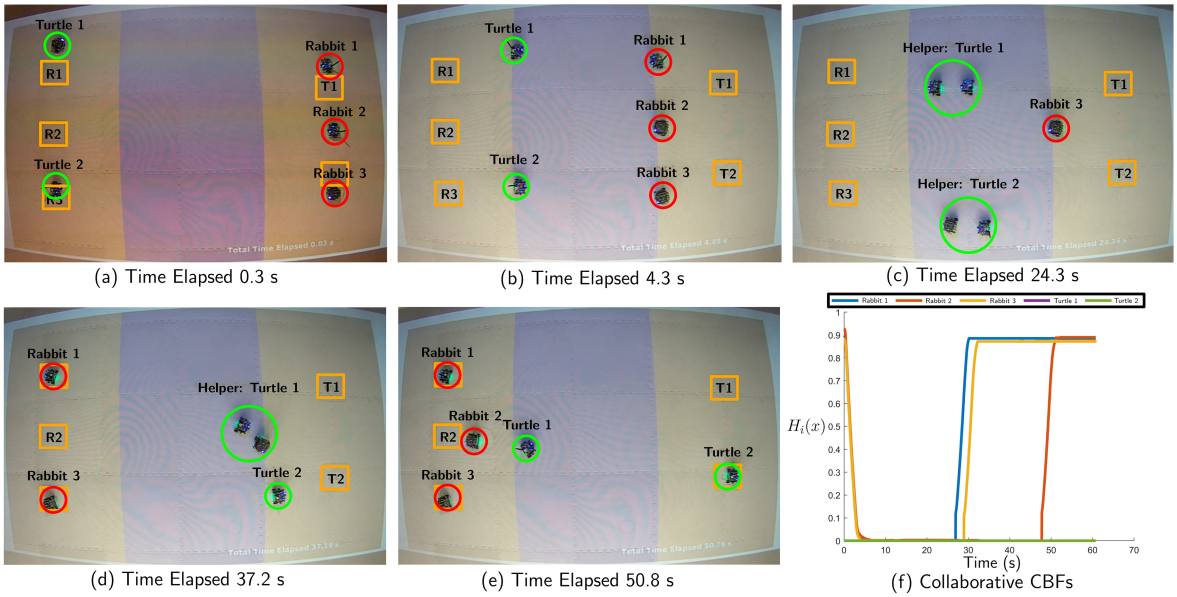
This section explores potential selection prioritization strategies that robots can employ in the dispatch model, shown in Fig. 2, to determine their partner for pairwise collaboration. In particular, we discuss: A) first in, first out; B) importance functions; and C) matching algorithms, which leverage the help queue to act as the matchmaking mechanism that dispatches robots to help any individuals requesting assistance. Moreover, we discuss the algorithmic complexity [20] – both the run-time complexity, i.e., the execution time of an algorithm, and space complexity, i.e., the memory usage for an algorithm – for pairing c help-providing robots and k help-needing robots with the mentioned selection strategies.

A. First In, First Out

A first in, first out queue is a simple, low-complexity selection prioritization strategy that does not require sorting to pair the help-needing robots and the help-providing robots. Particularly, the help-providing robots will have a pre-assigned priority for being dispatched to help-needing individuals. Thus, the space and run-time complexity are both constant, i.e., $\mathcal{O}(1)$. The disadvantage of this method is that a feasible selection can exist, but it may be poor; e.g., paired robots at distant locations would spend excessive energy and time during collaboration compared to paired robots at nearby locations.

B. Importance Functions

Importance functions is a medium-complexity selection prioritization strategy that sorts help-needing and help-providing robots based on relative importance. Particularly, for both individuals requesting and providing help, one can define a function to capture some notion of importance, e.g., mass, energy, time, and task credit score. The help-needing and help-providing robots are sorted, relative to each other, from highest to lowest priority by their importance functions. Then, the help-providing robot with the highest priority is paired with



(a) Time Elapsed 0.3 s

(b) Time Elapsed 4.3 s

(c) Time Elapsed 24.3 s

(d) Time Elapsed 37.2 s

(e) Time Elapsed 50.8 s

(f) Collaborative CBFs

Fig. 3. Fig. 3(a)-(e) provides snapshots of an experiment conducted in the Robotarium in which three ground robots ("rabbits") and two amphibious robots ("turtles") are tasked with safely reaching their respective goal points within an environment comprised of land terrain ($\mathcal{D}_{\text{land}}$; brown pixels) and water terrain ($\mathcal{D}_{\text{water}}$; brown pixels). [Supplemental Video: https://youtu.be/myaprkpNT6M]. Fig. 3(f) shows each robot's composed collaborative barrier function throughout the experiment.

the help-needing robot with the highest priority. However, it is possible that sorting is needed to properly compare the queue of the robots providing help to the queue of the robots receiving help, depending on the choice of importance measure. Thus, the space and run-time complexity are linear, i.e., $\mathcal{O}(c + k)$, and quasilinear, i.e., $\mathcal{O}(c\log(c) + k\log(k))$, respectively.

C. Matching Algorithms

Matching algorithms is a high-complexity selection prioritization strategy that pairs robots receiving and providing help based on the solution of the so-called matching problem in optimization [21]. This method is particularly suited for when one desires to compute the best, i.e., the (global) optimal, match between help-needing and help-providing while considering constraints such as the robots' barrier functions and selection preferences. However, as the number of constraints grows, so does the complexity, which makes obtaining an optimal solution substantially more difficult. Thus, the space and run-time complexity are quadratic, i.e., $\mathcal{O}((c + k)^2)$, and polynomial, i.e., $\mathcal{O}((c + k)^3)$, respectively, when employing the Hungarian algorithm [22], for example.

V. Experimental Results

This section presents an experiment conducted in the Robotarium [23], where teams of differential-drive robots can perform coordinated control strategies within a 3.6 m × 2.4 m testbed.

For this experiment, the robots attempt to reach goal points within a workspace comprised of water regions ($\mathcal{D}_{\text{water}}$) and land regions ($\mathcal{D}_{\text{land}}$), illustrated in Fig. 3(a), while employing the dispatch model (Fig. 2) in Section II-A. We consider a team of $N = 5$ mobile robots: three ground robots, referred to as "rabbits", and two amphibious robots, referred to as "turtles". The turtles can safely traverse $\mathcal{D}_{\text{land}}$ and $\mathcal{D}_{\text{water}}$ alone, whereas the rabbits can safely traverse $\mathcal{D}_{\text{land}}$ alone. Therefore, each rabbit must receive help from a turtle to traverse $\mathcal{D}_{\text{water}}$. As such, we assume the turtles can reach the rabbits from anywhere, i.e., their individual safe sets are path-connected.

Each robot's position states and control inputs are defined as $x_i = [p_{i,x}, p_{i,y}]^\top \in \mathcal{X}_i \subset \mathbb{R}^2$ and $u_i \in \mathcal{U}_i \subset \mathbb{R}^2$, respectively, for all $i \in \mathcal{N}$. The rabbits are indexed as $i \in \{1, 2, 3\}$, whereas turtles are indexed as $i \in \{4, 5\}$. In addition, we assume that each robot has single-integrator dynamics, i.e., $\dot{x}_i = u_i$, since we can abstract differential-drive robots as single integrators by using a near-identity diffeomorphism [24], and tasked with driving to a goal location, $x_i^g \in \mathcal{X}_i$, in the workspace.

The nominal controller – designed with this particular experiment in mind – that robot i executes to progress towards its objective, whether it collaborates or not, is given by

$$\hat{u}_i = \begin{cases} 0_{2\times 1}, & \text{if } \xi_i \in \{W_i\} \\ K_i\big(x_i^g - x_i\big), & \text{if } \xi_i \in \{I_i\} \\ K_i\big(x_i^g - x_i\big), & \text{if } \xi_i \in \{S_i^j, R_i^j\} \\ K_i\big(x_j - x_i\big), & \text{if } \xi_i \in \{A_j^i, P_j^i\} \end{cases}, \tag{21}$$

where $K_i \succ 0$ is a proportional gain and $\xi_i \in \Xi_i$ is a discrete state of the FSM. Moreover, it is assumed that each robot has a maximum speed threshold, $\bar{u}_i$, where its nominal controller, $\hat{u}_i$, is scaled down appropriately when $\|\hat{u}_i\|_2 > \bar{u}_i$.

The barrier functions used to encode the safe operating regions of each rabbit and turtle are given by

$$h_r(x_r) = p_{r,x}^2 - (\ell/2)^2, \tag{22}$$

$$h_{rt}(x_r, x_t) = \begin{cases} -p_{r,x}^2 + (\ell/2)^2, & \text{if } x_r = x_t \in \mathcal{D}_{\text{water}} \\ 0, & \text{else} \end{cases}, \tag{23}$$

and

$$h_t(x_t) = 0, \tag{24}$$

$$h_{tr}(x_t, x_r) = 0, \tag{25}$$

respectively, where ℓ is the width of $\mathcal{D}_{\text{water}}$.

It is important to mention that, due to the collision avoidance safety constraints of the Robotarium [23], the turtle cannot physically carry the rabbit across its unsafe region. Therefore, we depict "carrying" as when the turtle and rabbit move together in close proximity. In particular, the nominal controller, given by (21), was designed such that the turtle (help-provider) trails behind the rabbit (help-receiver) as they progress toward the rabbit's goal location during pairwise collaboration.

For this experiment, we utilized importance functions as the selection strategy. Each rabbit requesting help is associated with an importance function defined as the distance between its current and goal location. The rabbits' importance functions are sorted from highest priority (smallest distances) to lowest priority (largest distances). Similarly, each turtle is associated with an importance function defined as the distance between their current position and the highest-priority rabbit, i.e., closest to its goal location. We then pair the rabbit and turtle with the highest priority together. If more than one rabbit requests help, the selection process continues until matching robots becomes infeasible.

The experimental parameters are $\alpha(r) = 100r^3$ (extended class $\mathcal{K}_\infty$ function), $K_i = I_{2\times 2}$ $\forall i \in \mathcal{N}$ (proportional gain), $\ell = 1.07$ m (width of $\mathcal{D}_{\text{water}}$), $\bar{u}_r = 0.18$ m/s and $\bar{u}_r = 0.14$ m/s (max velocity threshold of rabbits and turtles without collaboration), and $\bar{u}_r = \bar{u}_t = 0.105$ m/s (max velocity threshold of rabbits and turtles with collaboration).

Fig. 3(a)–(e) portrays snapshots of the mobile robots during the experiment. Fig. 3(a) shows the initial configuration of the rabbits and turtles. Fig. 3(b) shows turtles 1 and 2 approaching rabbits 1 and 3, respectively, while rabbit 2 is stuck waiting for help. Fig. 3(c) shows turtles 1 and 2 helping rabbits 1 and 3, respectively. Fig. 3(d) shows rabbits 1 and 3 at their respective goal points, whereas turtle 2 navigates towards its goal point, while turtle 1 helps rabbit 2. Fig. 3(e) shows rabbit 2 and turtle 1 navigating towards their respective goal points while the others have already reached their respective goal points. Fig. 3(f) portrays each robot's composed collaborative barrier function, highlighting that the robots reached their respective goal points while remaining safe throughout the experiment, i.e., $H_i(x(t)) \geq 0$ $\forall t$, even during pairwise collaborations.

VI. Conclusion

In this letter, we proposed a scalable, pairwise collaboration architecture, termed the "dispatch model", that employs a finite state machine (FSM) to encode the decision-making of individual robots and establishes a centralized queue, termed the "help queue", to pair robots for collaboration. Furthermore, we showed that the FSM complexity of individual robots employing the dispatch model is decoupled from the overall team's size when using the help queue. In addition, we discussed potential selection prioritization strategies with varying levels of algorithmic complexity, i.e., run-time and space, when pairing the help-providing and help-needing robots to collaborate. Lastly, we validated the effectiveness of the dispatch model on a team of mobile robots in the Robotarium.

References

[1] E. Tuci, M. Alkilabi, and O. Akanyeti, "Cooperative object transport in multi-robot systems: A review of the state-of-the-art," *Front. Robot. AI*, vol. 5, p. 59, May 2018.

[2] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.

[3] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robot. Autom. Mag.*, vol. 19, no. 1, pp. 24–39, Mar. 2012.

[4] J. P. Queralta et al., "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191617–191643, 2020.

[5] Y. Rizk, M. Awad, and E. W. Tunstel, "Cooperative heterogeneous multi-robot systems: A survey," *ACM Comput. Surv.*, vol. 52, no. 2, pp. 1–31, Apr. 2020.

[6] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *Int. J. Adv. Robot. Syst.*, vol. 10, no. 12, p. 399, Jan. 2013.

[7] M. Krizmancic, B. Arbanas, T. Petrovic, F. Petric, and S. Bogdan, "Cooperative aerial-ground multi-robot system for automated construction tasks," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 798–805, Apr. 2020.

[8] I. D. Miller, F. Cladera, T. Smith, C. J. Taylor, and V. Kumar, "Stronger together: Air-ground robotic collaboration using semantics," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9643–9650, Oct. 2022.

[9] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. Eur. Control Conf.*, 2019, pp. 3420–3431.

[10] A. A. Nguyen, F. Jabbari, and M. Egerstedt, "Mutualistic interactions in heterogeneous multi-agent systems," in *Proc. IEEE Conf. Decision Control*, 2023, pp. 411–418.

[11] N. A. Lynch, *Distributed Algorithms*. Amsterdam, The Netherlands: Elsevier, 1996.

[12] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*. Boston, MA, USA: Addison-Wesley, 2001.

[13] M. Risler, "Behavior control for single and multiple autonomous agents based on hierarchical finite state machines," Ph.D. dissertation, Dept. Comput. Sci., Technische Universität Darmstadt, Darmstadt, Germany, 2010.

[14] S. Park, Y. D. Zhongeng, and N. E. Leonard, "Multi-robot task allocation games in dynamically changing environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 8678–8684.

[15] C. Ju and H. I. Son, "A hybrid systems-based hierarchical control architecture for heterogeneous field robot teams," *IEEE Trans. Cybern.*, vol. 53, no. 3, pp. 1802–1815, Mar. 2023.

[16] M. Sipser, *Introduction to the Theory of Computation*. Boston, MA, USA: Cengage Learn., 2012.

[17] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

[18] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, "Learning safe multiagent control with decentralized neural barrier certificates," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1724–1735.

[19] 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 Syst. Lett.*, vol. 5, no. 1, pp. 127–132, Jan. 2021.

[20] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to the Algorithms*. Cambridge, MA, USA: MIT, 2022.

[21] L. W. Wolsey, *Integer Programming*. Hoboken, NJ, USA: Wiley, 1998.

[22] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Log. Quart.*, vol. 2, nos. 1–2, pp. 83–97, Jan. 1955.

[23] D. Pickem et al., "The Robotarium: A remotely accessible swarm robotics research testbed," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 1699–1706.

[24] R. Olfati-Saber, "Near-identity diffeomorphisms and exponential /spl epsi/-tracking and /spl epsi/-stabilization of first-order non-holonomic SE(2) vehicles," in *Proc. Amer. Control Conf.*, 2002, pp. 4690–4695.