A Sequential Composition Framework for Coordinating Multi-Robot Behaviors

Pietro Pierpaoli, Anqi Li, Mohit Srinivasan, Xiaoyi Cai, Samuel Coogan, and Magnus Egerstedt

***Abstract*—A number of coordinated behaviors have been proposed for achieving specific tasks for multi-robot systems. However, since most applications require more than one such behavior, one needs to be able to compose together sequences of behaviors while respecting local information flow constraints. Specifically, when the inter-agent communication depends on inter-robot distances, these constraints translate into particular configurations that must be reached in finite time in order for the system to be able to transition between the behaviors. To this end, we develop a framework based on finite-time convergence control barrier functions that drives the robots to the required configurations. In order to demonstrate the proposed framework, we consider a scenario where a team of eight planar robots explore an urban environment in order to localize and rescue a subject. The results are presented in the form of a case study, which is implemented on a multi-agent robotic test-bed.**

***Index Terms*—Multi-robot systems, control barrier functions, networked robots**

I. Introduction

As our understanding of how to structure control and coordination protocols for teams of robots increases, a number of application domains have been identified, such as entertainment [1] [2], surveillance [3] [4], manipulation [5], and search-and-rescue [6]. Along with a decrease in the production and manufacturing costs associated with the platforms themselves, these applications have been enabled by a number of theoretical results that have emerged at the intersection of different disciplines such as robotics, controls, computer science, and graph theory [7].

From a controls perspective, one notable requirement is given by the need to define actions that are *valuable* for the entire team, on the basis of locally available information. For instance, in the context of motion control, different extensions of the consensus equation have been used to arrive at locally defined controllers with provable, global properties [8]. In this context, individual controllers can be defined as weighted sums of neighboring robots' relative states, which are usually easy to obtain. By following this methodology, it is possible to construct coordinated controllers for the solution of many motion control problems, such as rendezvous [9] [10], cyclic pursuit [11], formation control [12] [13], area coverage [14] [3], leader-based control [15], and flocking [16]. Particular instantiations of some of these behaviors are shown in Fig. 1 on a group of six simulated differential drive robots.

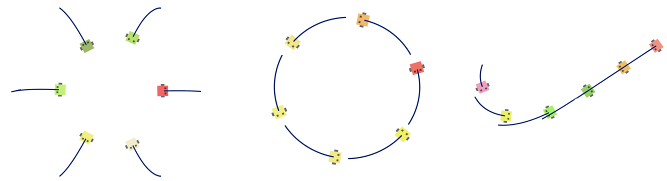


Fig. 1: Three examples of distributed multi-agent behaviors simulated on a group of differential drive robots. Starting from the left: rendezvous, cyclic-pursuit, and leader-follower. The lines indicate the past trajectories of the robots.

Even though these coordinated behaviors can address a number of different tasks, they have limited utility in the context of complex, real-world missions. However, the utility of these behaviors can be greatly expanded if they are sequenced together, which is the primary consideration in this paper. But, for a construction like this to work, it is necessary that the required information is available to the agents as they transition from one behavior to the next.

The available information can be encoded through *graphs*, whose vertices and edges are respectively represented by the robots and the sharing of information between them. For instance, considering the coordinated behaviors mentioned above, the rendezvous problem requires a connected graph [15], cyclic pursuit a directed cycle [11], formation control a rigid graph [15], and most coverage controllers require a Delaunay graph [14]. As such, the problem of composing different behaviors, can be recast in terms of the ability of the robots to enable the appropriate interactions needed at each stage of a mission.

When the communication between agents depends on their relative configurations (e.g. relative distance or orientation), realizing a certain communication structure directly affects the state of the system, which in turn, affects the execution of the mission itself. In order to overcome this coupling, we

This work was sponsored by DARPA Grant No. N66001-17-2-4059.

Pietro Pierpaoli is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA. (email: pietro.pierpaoli@gatech.edu)

Anqi Li is with the School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA. (email: anqi.li@gatech.edu)

Mohit Srinivasan is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA. (email: mohit.srinivasan@gatech.edu)

Xiaoyi Cai is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA. (email: xcai41@gatech.edu)

Samuel Coogan is with the School of Electrical and Computer Engineering and School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA. (email: sam.coogan@gatech.edu)

Magnus Egerstedt is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA. (email: magnus@gatech.edu)

separate the problem of generating a sequence of behaviors that corresponds to the solution of a mission objective from their composition. In fact, although generating an appropriate sequences of behaviors from given mission specifications is an interesting problem (e.g. [17]), in this work we focus on the problem of designing a provably correct composition framework given a sequence of coordinated behaviors.

The contribution of this paper is twofold. Firstly, we introduce a composition framework based on the results in [18], which guarantees correct composition of coordinated behaviors in the sense that their communication constraints are satisfied. The results in this paper extend those presented in [18] by introducing a decentralized formulation of the composition problem. Secondly, responding to the lack of established large-scale scenarios for the testing of multi-agent techniques, we describe in detail a particular scenario. This case study, called *Securing a Building* is rich and complex to capture many challenges and objectives of real-world implementation of multi-agent robotics, thus representing a standalone contribution of this paper.

The remainder of this paper is organized as follows. In Section III we review the definition of finite-time convergence barrier functions, while in Section IV we present the multi-robot composition framework. In Section V we describe the implementation of the composition framework in a distributed setting, appropriate for the deployment on a team of robots. In Section VI, we discuss the *Securing a Building* case study in detail, which is executed on the Robotarium [19]. Finally, motivated by the lack in literature of well-established scenarios for testing and comparing multi-agent robotics techniques, in Appendix A we include a discussion about open questions posed by the Securing a Building case study.

II. Related Work

The idea of partitioning complex objectives into simpler tasks can be approached by either sequentially composing *primitives*, e.g., [20], or by blending them simultaneously in a *hierarchical* fashion. An example of hierarchical composition for single robot motion control is navigation between points, e.g., [21].

The problem of composing different controllers has been approached as a hybrid systems problem, and in [22], controllers are constructed using motion description language. Symbolic methods are used in [23] for the solution of the motion control problem, where high-level instructions are considered in the form of *human-like* language. Because of the complexity emerging from the composition of distinct controllers, guarantees on the safety and correctness of the final results need to be established [24]. Provable correct composition of control laws is investigated in the formal methods literature. Recently, compositional strategies inspired from formal methods have been used for the development of control strategies for multi-robot systems [25] [26] [27]. In particular, [25] uses tools from linear temporal logic (LTL) for the specification of behaviors to be executed by the system. The authors use a sequence of constrained reachability problems, each consisting of a target set to be reached in finite time and a safety set within which the system must stay at all times. The authors in [26] discuss a hierarchical decomposition method for controller synthesis given LTL specifications.

In [28], authors introduce a framework for the composition of controllers in distributed robotic systems using Petri Nets. In [29], behaviors from the Null-Space-Behaviors framework are combined in order to solve ad-hoc tasks, such as perimeter patrol. A supervisor, represented as a finite state automata, selects high-level behaviors which are in turn composed of low-level ones. In [17], a revised version of the A^* algorithm is used to generate an optimal path of behaviors, such that the overall cost of the mission is minimized. Similarly, in [30] the motion planning problem for a team of quadcopters is solved by defining higher level motion primitives obtained by a spatial partition of the working environment. However, none of these approaches specifically address the problem of correct composition between primitives, which is the focus of this paper.

As discussed in the previous section, coordination between agents is possible only if particular interactions exist between the robots. In multi-robot systems, interaction requirements are commonly investigated in terms of connectivity maintenance, i.e., a certain graph or node-connectivity needs to be guaranteed at all times. Methods employed in the solution to this problem include edge weight functions [31], control rules based on estimate of algebraic connectivity [32], hybrid control [33], passivity [34], and barrier functions [35]. If connectivity between agents needs to be guaranteed in non-nominal circumstances, resilient solutions must be in place as well, e.g., [36], [37], and [38]. Notably, a technique based on graph process specifications for the sequential composition of different multi-agent controllers is discussed in [39]. Similar to our work, the authors in [39] bridge the gap between composition of controllers and the topology requirements by encoding requisites for each controller in terms of graphs. However, while in [39] *incompatible* controllers are combined through the introduction of a *bridging* controller, in our approach controllers are minimally modified by the robots in order to satisfy upcoming requirements. Our approach significantly reduces the complexity of the composition process and can accommodate additional constraints, such as inter-robot collisions and obstacles avoidance.

III. Finite-Time Barrier Functions

In this section we review the general definition of Finite-time Convergence Control Barrier Function (FCBF) which was first introduced in [18] and inspired by the finite-time stability analysis for autonomous system introduced in [40]. Given a dynamical system operating in an open set $\mathcal{D} \subseteq \mathbb{R}^n$ and a set $\mathcal{C} \subset \mathcal{D}$, barrier functions [41] are Lyapunov-like functions that guarantee forward invariance of $\mathcal{C}$ with respect to the state of the system. In other words, if an appropriate barrier function exists, it can be used to show that if the state of a system is in $\mathcal{C}$ at some time, it will be in $\mathcal{C}$ thereafter. The concept of barrier functions was extended to Zeroing Control Barrier Functions (ZCBF) in [41], where asymptotic convergence of the state to the set $\mathcal{C}$ was discussed. Thus, provided that an appropriate ZCBF exists, if the state of the system is not in $\mathcal{C}$ at some initial time, it will asymptotically converge to $\mathcal{C}$.

As discussed in the introduction, before execution of a coordinated behavior, robots need to satisfy certain spatial configurations, which are imposed by the behavior itself. In this context, it is worth to note that asymptotic convergence to the correct configuration is not sufficient. In fact, if we consider $\mathcal{C}$ as the joint set of all initial configurations required for a particular behavior, the state must strictly belong to $\mathcal{C}$ for the behavior to work properly. Following this observation, the need for a finite-time convergence extension of the previous concepts becomes clear. In particular, we are interested in verifying the following conditions:

- if $x(t_0) \in \mathcal{C}$, then $x(t) \in \mathcal{C}$ for all $t > t_0$
- if $x(t_0) \notin \mathcal{C}$, then $x(t) \in \mathcal{C}$ for some $t_0 < t < \infty$.

In order to do this, we encode the set $\mathcal{C} \subset \mathcal{D} \subseteq \mathbb{R}^n$, through the superzero-level set of a continuous differentiable function $h : \mathcal{D} \to \mathbb{R}$, i.e.,

$$\mathcal{C} = \{x \in \mathcal{D} \,|\, h(x) \geq 0\}. \tag{1}$$

Definition 3.1: We introduce the following class-$\mathcal{K}$ function

$$\bar{\alpha}_{\rho,\gamma}(h(x)) = \gamma \cdot \text{ sign}(h(x)) \cdot |h(x)|^{\rho}, \tag{2}$$

with $\rho \in [0, 1)$ and $\gamma > 0$, which is continuous everywhere and locally Lipschitz everywhere except at the origin [40].

Definition 3.2: [18] For a dynamical system

$$\dot{x} = f(x) + g(x)u \tag{3}$$

with $x \in \mathcal{D}$, $u \in U \subset \mathbb{R}^m$, and for a set $\mathcal{C}$ induced by h, if there exists a function $\bar{\alpha}_{\rho,\gamma}(h(x))$ of the form (2) such that

$$\sup_{u \in U} \left\{ L_f h(x) + L_g h(x) u + \bar{\alpha}_{\rho,\gamma}(h(x)) \right\} \geq 0 \quad \forall x \in \mathcal{D}, \tag{4}$$

then, the function h is a *Finite-time Convergence Barrier Function* (FCBF) defined on $\mathcal{D}$.

Following from the definition above, we define the set of admissible control inputs as

$$K(x) = \{u \in U \,|\, L_f h(x) + L_g h(x) u + \bar{\alpha}_{\rho,\gamma}(h(x)) \geq 0\}. \tag{5}$$

For completeness, we state and prove the following theorem which was first introduced in [18].

Theorem 3.3: Given a set $\mathcal{C} \subset \mathbb{R}^n$, any Lipschitz continuous controller $\mathcal{U} : \mathcal{D} \mapsto U$ such that

$$\mathcal{U}(x) \in K(x) \qquad \forall x \in \mathcal{D}, \tag{6}$$

renders $\mathcal{C}$ forward invariant for the system (3). Moreover, given an initial state $x_0 \in \mathcal{D} \backslash \mathcal{C}$, the same controller $\mathcal{U}$ results in $x(T) \in \mathcal{C}$, where

$$T \leq \frac{1}{\gamma(1-\rho)} |h(x_0)|^{1-\rho}. \tag{7}$$

Proof: Let's consider the following Lyapunov function $V(x) = \max\{0, -h(x)\}$. It can be verified that

$$V(x) > 0 \qquad x \in \mathcal{D} \backslash \mathcal{C} \tag{8}$$

$$V(x) = 0 \qquad x \in \mathcal{C} \tag{9}$$

In addition, since

$$\frac{\partial V(x)}{\partial h(x)} = \begin{cases} -1 & x \in \mathcal{D} \backslash \mathcal{C} \\ 0 & x \in \mathcal{C} \end{cases} \tag{10}$$

it follows that $\dot{V}(x(t)) \leq -\gamma V^{\rho}(x(t))$, for all t.

Consider $x_0 = x(t_0) \in \mathcal{C}$. Since $V(x_0) = 0$ and $\dot{V}(t) = 0$, we have $x(t) \in \mathcal{C}$ for all $t > t_0$, from which forward invariance of $\mathcal{C}$ under $\mathcal{U}$ follows. Now, consider $x_0 \in \mathcal{D} \backslash \mathcal{C}$. As shown in [40], the dynamics $\dot{h}_1(t) = -\gamma \operatorname{sign}(h_1(t)) |h_1(t)|^{\rho}$, with $\rho \in [0, 1)$ and $\gamma > 0$ drives h_1 to the origin, and the minimum time T_1 for which h_1 reaches 0 (denoted finite-settling time) is

$$T_1 = \frac{1}{\gamma(1-\rho)} |h_1(t_0)|^{1-\rho}. \tag{11}$$

By applying the comparison lemma [42], if $h(0) \geq h_1(0)$ and $\dot{h}(t) \geq \dot{h}_1(t)$, then $h(t) \geq h_1(t)$ for all $t \geq 0$ and consequently under the effect of $\mathcal{U}$ we have

$$h(x(T)) = 0, \quad T \leq T_1. \tag{12}$$

■

In conclusion, by selecting controllers in the form of (6), both forward invariance and finite-time convergence to the desired set are guaranteed.

IV. Composition of Coordinated Behaviors

In this section we define the framework for sequential composition of coordinated behaviors. We start by introducing the model of our multi-robot system.

A. Multi-Robot System

We denote the state of a team of n homogeneous mobile robots operating in a planar and connected domain $\mathcal{D}$ as $x(t) = [x_1(t)^T, \ldots, x_n(t)^T]^T \in \mathcal{D} \subset \mathbb{R}^{2n}$ where $x_i(t)$ is the position of robot i at time t. As part of the coordinated nature of the behaviors being performed by the robots, each robot executes a control protocol which depends on the state of the subset of robots with which it interacts. We assume omnidirectional Δ-disk interactions between agents, i.e., agents can share information or cooperate if the distance between them is less or equal to a sensing threshold $\Delta \in \mathbb{R}_{>0}$. Thus, the list of possible interactions between agents are described by a time-varying proximity graph $\mathcal{G}(t) = (V, E(t))$, where $V = \{1, \ldots, n\}$ is the set of nodes representing the robots and $E(t)$ is the set of interacting pairs at time t, where

$$E(t) = \{(i, j) \in V \times V \,|\, \|x_i(t) - x_j(t)\| \leq \Delta\}. \tag{13}$$

For each robot $i = 1, \ldots, n$, we denote the set of available neighbors at time t as $\mathcal{N}_i(t) = \{j \in V \,|\, (i, j) \in E(t)\}$, which depends on the position of the robots at time t.

The ensemble dynamics of the multi-agent system is described by

$$\dot{x} = f(x) + g(x)\, u \tag{14}$$

where f and g are continuous locally Lipschitz functions and $u = [u_1^T, \ldots, u_n^T]^T \in U \subset \mathbb{R}^m$ is the vector containing agents' control inputs, which depends on the particular behavior being executed. At all times, the control input u in (14) is given by a controller $\mathcal{U}$, which can be defined as a state feedback law $\mathcal{U} : \mathcal{D} \mapsto U$ or by a combination of both external parameters and state feedback law $\mathcal{U} : \mathcal{D} \times \Theta \mapsto U$, where Θ is a space of parameters appropriate for the behavior. For instance, the controller corresponding to a *weighted consensus* protocol belongs to the first case. On the other side, a protocol where a leader moves to a given point while followers maintain prescribed inter-agent distances is described by a controller that depends on both state feedback (followers' control) and exogenous parameters (leader's goal).

B. Behaviors Sequencing Framework

We compactly represent an entire mission by an ordered sequence of M coordinated behaviors

$$\pi = \{\mathcal{B}_1, \ldots, \mathcal{B}_M\}. \tag{15}$$

The k^{th} behavior in π is defined by the pair

$$\mathcal{B}_k = \{\mathcal{U}_k, \mathcal{G}_k\}, \tag{16}$$

where $\mathcal{U}_k$ represents the coordinated controller described above and $\mathcal{G}_k$ is the interaction graph required by behavior $\mathcal{B}_k$ to function properly. We assume the list of behaviors π to be made available to all robots before the start of the mission.

As discussed in Section I, each coordinated behavior requires a certain interaction structure between the robots (i.e., pairs of robots that need to be neighbors). With reference to (16), we describe the required interaction structure via the graph $\mathcal{G}_k = (V, E_k)$. Thus, denoting by $t_k^{\vdash}$ and $t_k^{\dashv}$ the starting and ending times for behavior k respectively, the robots' configuration needs to satisfy $\mathcal{G}_k \subseteq \mathcal{G}(t)$ for all $t \in [t_k^{\vdash}, t_k^{\dashv}]$. In other words, as shown in Fig.2, the interaction structure required by each behavior needs to be a spanning graph of the graph induced by the state of the agents during the interval of time the behavior is executed. During the transition time between two behaviors, no particular requirement is imposed on the interaction structure.

In addition to the list π, transition between behaviors need to be synchronized, i.e., agents must start a new behavior at the same time, after all required edges are formed. To this end, we assume the existence of a discrete counter $\sigma \in [1, \ldots, M]$ which indicates what behavior is currently active and a binary signal

$$\eta(\sigma) = \begin{cases} 1 & \text{if} \quad \mathcal{G}_k \subseteq \mathcal{G}(t) \\ 0 & \text{o.w.} \end{cases} \tag{17}$$

which describes whether the interaction structure required by behavior $\mathcal{B}_\sigma$ is available. We assume both signals to be controlled by a supervisor and made available to the robots, e.g., through a dedicated static communication network. We note that the role of such a supervisor could be replaced by a decentralized estimation scheme of condition $\mathcal{G}_k \subseteq \mathcal{G}(t)$. As such implementation goes beyond the scope of this work, it will not be discussed here.

Because of the sensing modality assumed for the team of robots, communication constraints can be expressed in terms of relative positions between the agents. In other words, behavior $\mathcal{B}_k$ can be correctly executed if, for all $t \in [t_k^{\vdash}, t_k^{\dashv}]$, all the distances between pairs in E_k are below the proximity threshold Δ. To this end, a convenient pair-wise connectivity FCBF can be defined as

$$h_{ij}^c(x) = \Delta^2 - \|x_i - x_j\|^2, \tag{18}$$

and we note that if $\|x_i - x_j\| \leq \Delta$, then $h_{ij}^c(x) \geq 0$. In addition, the edge-level and ensemble-level connectivity constraint sets for behavior $\mathcal{B}_k$ are

$$\mathcal{C}_{ij}^c = \{x \in \mathcal{D} \,|\, h_{ij}^c(x) \geq 0\} \tag{19}$$

$$\mathcal{C}_k^c = \{x \in \mathcal{D} \,|\, h_{ij}^c(x) \geq 0, \, \forall (i,j) \in E_k\}. \tag{20}$$

Following the definition given in (5), the admissible set of control inputs that guarantees finite-time convergence to $\mathcal{C}_k^c$ is:

$$K_k^c(x) = \{u \in U \,|\, \dot{h}_{ij}^c(x) + \bar{\alpha}_{\rho,\gamma}(h_{ij}^c(x)) \geq 0, \; \forall (i,j) \in E_k\} \tag{21}$$

Theorem 4.1: Denoting with x_0 the initial state of a multi-agent system with dynamics described as in (14), any controller $\mathcal{U} : \mathcal{D} \mapsto U$ such that $\mathcal{U}(x_0) \in K_k^c(x_0)$ for all $x_o \in \mathcal{D}$, will drive the system to $\mathcal{C}_k^c$ within time

$$T_k = \max_{(i,j) \in E_k | h_{ij}^c(x_0) < 0} \left\{ \frac{1}{\gamma(1-\rho)} |h_{ij}^c(x_0)|^{1-\rho} \right\}. \tag{22}$$

Proof: Consider all pairs of agents i and j, such that $(i,j) \in E_k$. If $h_{ij}^c(x_0) \geq 0$, i.e., agents i and j are within communication distance, the forward invariance property of $\mathcal{U}$, guarantees that i and j will stay connected. In this case, the state will reach $\mathcal{C}_{ij}^c$, within time $T_{ij} = 0$. On the other

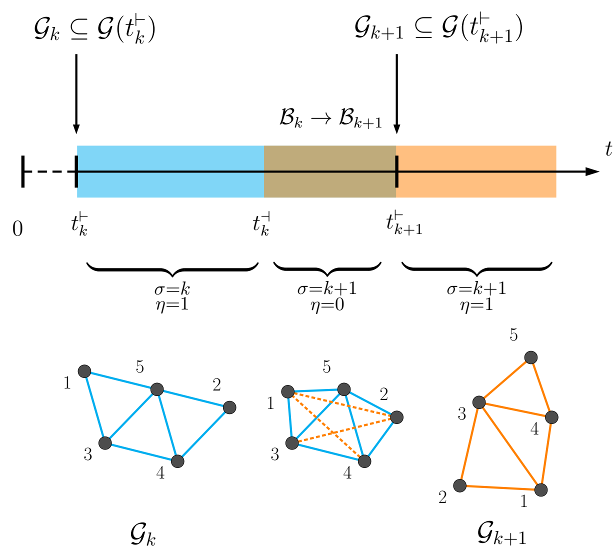


Fig. 2: Schematic representation of the behaviors sequencing framework. Behavior $\mathcal{B}_k$ is executed during the blue portion of the timeline and $\mathcal{B}_{k+1}$ is executed during the orange portion. Sequential execution of behaviors requires each agent to reach a spatial configuration such that the desired graph is a spanning graph of the communication graph, i.e., $\mathcal{G}_k \subseteq \mathcal{G}(t_k^{\vdash})$ and $\mathcal{G}_{k+1} \subseteq \mathcal{G}(t_{k+1}^{\vdash})$ respectively.

side, consider $h^c_{ij}(x_0) < 0$. Any $\mathcal{U}(x_0) \in K^c_k(x_0)$ satisfies the finite-time convergence barrier certificates, and because of Theorem 3.3, if $x_0 \notin \mathcal{C}^c_{ij}$, then $x(T_{ij}) \in \mathcal{C}^c_{ij}$, with

$$T_{ij} \leq \frac{1}{\gamma(1-\rho)} |h^c_{ij}(x_0)|^{1-\rho}. \tag{23}$$

Since every communication constraint $\mathcal{C}^c_{ij}$ will be reached within time T_{ij}, the total time required to drive $x(t)$ to $\mathcal{C}^c_k$ will be equal to the maximum of all these times, i.e.,

$$T_k = \max_{(i,j)\in E_k | h^c_{ij}(x_0)<0} T_{ij}. \tag{24}$$

■

Thus, by selecting control inputs from set (21), the system (14) will satisfy requirements for behavior $\mathcal{B}_k$, within time T_k.

C. Finite-Time Convergence Control Barrier Functions

In accordance with the composition framework described above, once behavior $\mathcal{B}_{k-1}$ is completed, robots are required to converge to the set $\mathcal{C}^c_k$ before behavior $\mathcal{B}_k$ can start. Following the synchronization model introduced in Section IV-B, the change of behavior is communicated to the robots through the signal σ, which transitions from value $k-1$ to k once $\mathcal{B}_{k-1}$ is completed. Now, although finite-time convergence to $\mathcal{C}^c_k$ can be achieved by selecting any control input in $K^c_k(x)$, we seek to minimally perturb the execution of the behavior just concluded, namely $\mathcal{B}_{k-1}$. This can be accomplished by solving a problem similar to the one proposed in [43], which we adapt to our framework. Denoting with $\hat{u}_{k-1} = \mathcal{U}_{k-1}(x)$ the nominal control input resulting from the concluded behavior $\mathcal{B}_{k-1}$, we define the actual control input to the robots u^* as

$$u^* = \arg\min_{u\in U} \|\hat{u}_{k-1} - u\|^2 \tag{25}$$

subject to

$$L_f\, h^c_{ij} + L_g\, h^c_{ij}\, u + \bar{\alpha}_{\rho,\gamma}(h^c_{ij}) \geq 0, \qquad \forall (i,j) \in E_{k-1} \cup E_k, \tag{26}$$

Once all required edges E_k are established (i.e., $\eta = 1$), edges in E_{k-1} are no longer needed. At this point, under the effect of the controller $\mathcal{U}_k$, the list of constraints in (26) is substituted with

$$L_f\, h^c_{ij} + L_g\, h^c_{ij}\, u + \bar{\alpha}_{\rho,\gamma}(h^c_{ij}) \geq 0, \qquad \forall (i,j) \in E_k. \tag{27}$$

Since the cost function is convex and the inequality constraints (26) and (27) are affine in the control input, the problem can be solved in real-time. In conclusion, because of the finite-time convergence and forward invariance properties of the above formulation, solution of (25) guarantees that robots will converge to the configuration required by $\mathcal{B}_k$, and maintain it throughout its execution.

D. Initial Constraints

In addition to the communication constraints considered above, certain missions might require additional conditions to be met before each behavior can start. For example, during the exploration of an area it might be desirable for one robot to always stay within range of communication with a human-operator, or to maintain a minimum distance from an unsafe area.

Assuming $\mathcal{B}_k$ requires a number of distinct s_k of such constraints, we encode the entire set of initial conditions through a list of barrier functions $h^s_\ell(x)$, with $\ell = 1,\ldots,s_k$:

$$\mathcal{C}^s_k = \{x \in \mathcal{D} \,|\, h^s_\ell(x) \geq 0,\ \forall \ell = 1,\ldots,s_k\}. \tag{28}$$

Following this definition, we define a set of admissible control inputs similar to the one in (21) that will drive the state of the system to the desired set within finite time:

$$K^s_k(x) = \{u \in U \,|\, \dot{h}^s_\ell(x) + \bar{\alpha}_{\rho,\gamma}(h^s_\ell(x)) \geq 0, \qquad \forall \ell = 1,\ldots,s_k\}. \tag{29}$$

The set of controls satisfying both communication and initial conditions constraints can thus be obtained by intersection of set (29) and (21):

$$K_k(x) = K^c_k(x) \bigcap K^s_k(x). \tag{30}$$

We note that the results in Theorem 4.1 and the formulation of minimally invasive controller in (25) still hold valid by considering the set $K_k(x)$ instead of $K^c_k(x)$ as the set of admissible control inputs.

V. Multi-Robot Implementation

The composition framework discussed in the previous section reduces to the minimum norm controller (25), which is not directly solvable by individual robots. In this section, we formulate an extension of the same problem, which is implementable in a distributed fashion and accounts for additional constraints necessary for the safe operations of the robots, e.g., inter-agent collisions and obstacles avoidance [35]. The formulation is derived following the approach described in [44], which we adapt here to our framework.

A. Distributed Composition of Coordinated Behaviors

The limitation in solving problem (25) in a distributed fashion is represented by the fact that knowledge of dynamics, input $\hat{u}$, and state x for the entire team need to be available to all robots. In addition, by solving problem (25), each agent would compute the control input for the entire team, which is clearly unnecessary.

In order to develop the correct decentralized formulation of (25), we first need an agent level decomposition of the dynamics (14). We denote by $\mathcal{D}_i \subset \mathbb{R}^2$ and $U_i \subset \mathbb{R}^2$ configuration space and set of feasible controls for agent i respectively, and we note that $\mathcal{D} = \mathcal{D}^n_i$. In addition, by denoting with $\bar{f}, \bar{g} : \mathcal{D}_i \mapsto \mathbb{R}^2$ the node-level terms of the control affine dynamics of agent i, the ensemble dynamics can be written as:

$$\dot{x} = \bar{f}(x_i) \otimes \mathbf{1}_n + (\bar{g}(x_i) \otimes I_n) \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}, \tag{31}$$

where $u_i \in U_i$ is the i^{th} agent's control input, $\otimes$ is the Kronecker product, and $\mathbf{1}_n$ and I_n are vector of ones and identity matrix of size n respectively.

Let's now consider two sequential behaviors $\mathcal{B}_{k-1}$ and $\mathcal{B}_k$. Upon completion of $\mathcal{B}_{k-1}$, for all edges $(i, j) \in E_k$ robots' configuration must satisfy

$$\dot{h}^c_{ij}(x_i, x_j) + \bar{\alpha}_{\rho,\gamma}(h^c_{ij}(x_i, x_j)) \geq 0. \tag{32}$$

From the i^{th} agent's point of view, the set of constraints that need to be satisfied are

$$\dot{h}^c_{ij}(x_i, x_j) + \bar{\alpha}_{\rho,\gamma}(h^c_{ij}(x_i, x_j)) \geq 0 \quad \forall j \in \mathcal{N}^i_k, \tag{33}$$

where we recall that $\mathcal{N}^i_k$ is the set of neighbors agent i will be connected to in the upcoming behavior $\mathcal{B}_k$. However, as it will be proved in the following theorem, since the same constraint (33) will appear exactly twice across the entire network of robots, it can be relaxed by considering an admissible set of control input of the form

$$K^{c,i}_k = \bigcap_{j \in \mathcal{N}^i_k} K^{c,i}_{k,ij} \tag{34}$$

with

$$K^{c,i}_{k,ij} = \{u_i \in U_i \,|\, L_{\bar{f}} h^c_{ij} + L_{\bar{g}} h^c_{ij} u_i + \frac{\bar{\alpha}_{\rho,\gamma}(h^c_{ij})}{2} \geq 0\}, \tag{35}$$

where explicit dependence from the states x_i and x_j is omitted for clarity.

Theorem 5.1: Denoting with $x_0 = [x^T_{0,1}, \ldots, x^T_{0,n}]^T$ the initial state of a multi-agent system with dynamics described as in (31), any controller $\mathcal{U}_i : \mathcal{D}_i^{|\mathcal{N}^i_k|} \mapsto U_i$ such that $\mathcal{U}_i(x_0) \in K^{c,i}_k$ for all $x_0 \in \mathcal{D}_i^{|\mathcal{N}^i_k|}$, will drive the ensemble state to $\mathcal{C}^c_k$ within time

$$T_k = \max_{\substack{(i,j)\in E_k \\ \text{s.t. } h^c_{ij}(x_{0,i}, x_{0,j})<0}} \left\{ \frac{1}{\gamma(1-\rho)} |h^c_{ij}(x_{0,i}, x_{0,j})|^{1-\rho} \right\}. \tag{36}$$

Proof: From Theorem 3.3, agents i and j, with $(i, j) \in E_k$, will satisfy $h^c_{ij} \geq 0$ in finite time if

$$\dot{h}^c_{ij} + \bar{\alpha}_{\rho,\gamma}(h^c_{ij}) \geq 0. \tag{37}$$

Considering the node level dynamics in (31), the constraint (37) reduces to

$$\begin{aligned} &\frac{\partial h^c_{ij}}{\partial x_i} \left(\bar{f} + \bar{g} u_i\right) + \frac{\partial h^c_{ij}}{\partial x_j} \left(\bar{f} + \bar{g} u_j\right) + \bar{\alpha}_{\rho,\gamma}(h^c_{ij}) \geq 0 \\ &2\, L_{\bar{f}} h^c_{ij} + L_{\bar{g}} h^c_{ij}\, u_i + L_{\bar{g}} h^c_{ij} u_j + \bar{\alpha}_{\rho,\gamma}(h^c_{ij}) \geq 0 \end{aligned} \tag{38}$$

which will be satisfied if both agents i and j satisfy the constraint

$$\dot{h}_{ij}(x_i, x_j) + \frac{\bar{\alpha}_{\rho,\gamma}(h_{ij}(x_i, x_j))}{2} \geq 0. \tag{39}$$

In addition, as discussed in Theorem 4.1, constraint (38) will still be satisfied at time

$$T_{ij} \leq \frac{1}{\gamma(1-\rho)} |h^c_{ij}(x_{0,i}, x_{0,j})|^{1-\rho}. \tag{40}$$

The same argument can be repeated for all pairs $(i, j) \in E_k$, and condition $\mathcal{G}_k \subseteq \mathcal{G}(t)$ will be satisfied within time

$$T_k = \max_{\substack{(i,j)\in E_k \\ \text{s.t. } h^c_{ij}(x_{0,i}, x_{0,j})<0}} \{T_{ij}\}. \tag{41}$$

■

Following this result, we apply the same design principle described in Section IV-C and the minimally invasive control action can be computed by each agent as

$$u^*_i = \arg\min_{u_i \in U_i} \|\hat{u}_{k-1,i} - u_i\|^2 \tag{42}$$

subject to

$$L_{\bar{f}}\, h^c_{ij} + L_{\bar{g}}\, h^c_{ij}\, u_i + \frac{\bar{\alpha}_{\rho,\gamma}(h^c_{ij})}{2} \geq 0, \quad \forall j \in \mathcal{N}^i_{k-1} \cup \mathcal{N}^i_k. \tag{43}$$

Similarly to what we discussed for constraint (26), once all required edges are created, constraint (43) will be relaxed and substituted with

$$L_{\bar{f}}\, h^c_{ij} + L_{\bar{g}}\, h^c_{ij}\, u_i + \frac{\bar{\alpha}_{\rho,\gamma}(h^c_{ij})}{2} \geq 0, \quad \forall j \in \mathcal{N}^i_k. \tag{44}$$

Thus, once behavior $\mathcal{B}_{k-1}$ is completed, agent i computes the control input necessary to create the interaction structure required by $\mathcal{B}_k$ by solving (42). We note that, in order for agent i to respect (44) (and similarly (43)), the only information needed is the state of all future neighbors, i.e. x_j for all $j \in \mathcal{N}^i_k$. This requirement can be satisfied through an estimate scheme (e.g. EKF [45]) on the state of the system, which in turn requires knowledge of all the agents' dynamics (satisfied by having assumed homogeneous agents). Alternatively, the state of future neighbors could be made available through high-power communication sensors that, for efficient energy management, can only be active for limited periods of time. It is worth to note that the ability of each agent to have access to an estimate of their neighbors' state does not automatically eliminate the necessity for agents to become actual neighbors. In fact, proximity might be justified by some desired performance requirements on the coordinated behavior or cooperation in the environment that requires psychical proximity, e.g. collaborative manipulation or sharing of resources [36].

B. Additional Constraints

In addition to the proximity constraints discussed above, additional constraints can be imposed on the robots for safe operation. The requirement on collision free trajectories is satisfied when all pair-wise distances between the agents are greater or equal to a safety distance $D_a > 0$. Following the approach described in [35], we can conveniently encode each pair-wise separation condition through the following barrier certificate

$$h^a_{ij}(x) = \|x_i - x_j\|^2 - D^2_a \tag{45}$$

and the minimum separation conditions are satisfied if $h^a_{ij}(x) \geq 0$, for all physical neighbors $j \in N^i(t)$, where recalling from the previous section, we denoted the complete set of neighbors available to agent i as N^i, which depends on the state of the system at time t.

In addition to the inter-agent collision avoidance, we also require that agents do not collide with fixed objects in the environment. We consider M obstacles modeled as two dimensional ellipses and we denote by $o_m \in \mathcal{D}$ and $o = [o_1^T, \ldots, o_M^T]^T$ the center of the m^{th} obstacle and the ensemble of the centers respectively. For every agent-obstacle pair (i, m) we define a pairwise barrier function as

$$h^o_{im}(x) = (x_i - o_m)^T \, P_m \, (x_i - o_m) - 1 \tag{46}$$

$$P_m = \begin{bmatrix} a_m & 0 \\ 0 & b_m \end{bmatrix} \quad a_m, b_m > 0. \tag{47}$$

The object avoidance constraints are satisfied if $h^o_{im}(x) \geq 0$, for all $i \in V$ and $m \in \{1, \ldots, M\} = \mathcal{I}_M$.

C. Complete Problem

Collecting all the constraints discussed in this section, we expand the problem formulation in (25) as follows:

$$\begin{aligned} & u_i^* = \arg\min_{u_i \in U_i} \|\hat{u}_{k-1,i} - u_i\|^2 \\ & L_f\, h^c_{ij} + L_g\, h^c_{ij}\, u_i + \frac{\bar{\alpha}_{\rho,\gamma}(h^c_{ij})}{2} \geq 0, \qquad \forall j \in \mathcal{N}^i_k \\ & L_f\, h^s_{ij} + L_g\, h^s_{ij}\, u_i + \alpha(h^s_{ij}) \geq 0, \qquad \forall j \in \mathcal{N}^i(t) \\ & L_f\, h^o_{im} + L_g\, h^o_{im}\, u_i + \alpha(h^s_{ij}) \geq 0, \qquad \forall m \in \mathcal{I}_M \end{aligned} \tag{48}$$

where α is a locally Lipschitz extended class-$\mathcal{K}$ function. We note that during transitions between behaviors, the first set of constraints is replaced by (43). In conclusion, the local control input that simultaneously satisfies all safety constraints and minimally perturbs the execution of each behavior is obtained by solving the quadratic program in (48).

VI. Case Study: Securing a Building

The objective of this section is to define the *Securing a Building* mission, which will be used as testing scenario for the composition framework. We describe now the main structure and objective of the mission, while we deconstruct it into coordinated behaviors in the next subsection.

A. Mission Overview

In the Securing a Building mission, a group of autonomous robots are deployed in an urban environment to identify a previously unknown target building in order to rescue a subject located inside. Based on [46], we decompose this mission into the following four phases:

FIND - During the find phase, the robots are tasked with identifying the target building by means of surveillance of the perimeters of all the buildings in the environment. For efficient exploration, robots can be broken into different teams and investigate preassigned subsets of buildings. Each team must then report back to the human operators at the base station in order to share the collected information. Once the target building has been identified, the robots which are still exploring the environment should reunite with the other robots and prepare for the next phase.

ISOLATE - Having identified the target building's location, the robots must isolate the building by patrolling its perimeter. To achieve this, the robots are divided into different subgroups - the *security agents* and the *maneuvering agents*. The security agents are responsible for achieving boundary protection, while the maneuvering agents look for the building entrance and prepare to enter.

RESCUE - During the rescue phase, the security agents keep patrolling around the building. The robots not tasked with boundary protection, the maneuvering agents, will enter the building in formation and subsequently clear rooms and seize positions as they maneuver through the building to find the subject to be rescued. Once the subject has been located, the robots should transport it to the safe zone.

FOLLOW-THROUGH - As the interior of the building is being cleared, individual robots are left inside as beacons to signal that the area has been cleared, while the remaining robots from the maneuvering agents leave the building, gather on the outside with the security agents, and report back to the base station.

A number of arguments support the choice of the Securing a Building mission as an ideal test-bed for testing multi-agent techniques and algorithms developed by the control and hybrid system communities. First, the requirement of spatially diverse functionalities that cannot be provided by single robots naturally requires the use of multi-robot systems. Second, the final goal of the mission, namely rescuing the subjects of interest, requires a number of different behaviors that necessarily involve the switching between different controllers. Lastly, thanks to the modularity of the mission, novel techniques focusing on specific aspects of the mission can be integrated and tested without influencing the overall structure of the mission (see the Appendix for more details).

B. Securing a Building Through Composition of Behaviors

We describe now the main contribution of this section, that is, the deconstruction of the Securing a Building mission described above, through ordered sequences of coordinated behaviors. The process is summarized in Fig. 3. We refer to behaviors in terms of their main objectives, acknowledging that many actual different implementations can be used to achieve the same results. We highlight these behaviors in parenthesis.

a) FIND: Robots initially coordination with human operators at the base station (*rendezvous*). After that, robots are divided into different search teams, each assigned with a list of buildings to investigate (*task allocation*). Subsequently, all the teams investigate their own lists of buildings. First robots travel to the vicinity of a building (*leader-follower*), then start to survey the exterior of the building (*perimeter patrol*), and after having collected the necessary information, return to the base (*leader-follower*). This process repeats until the target building is discovered.

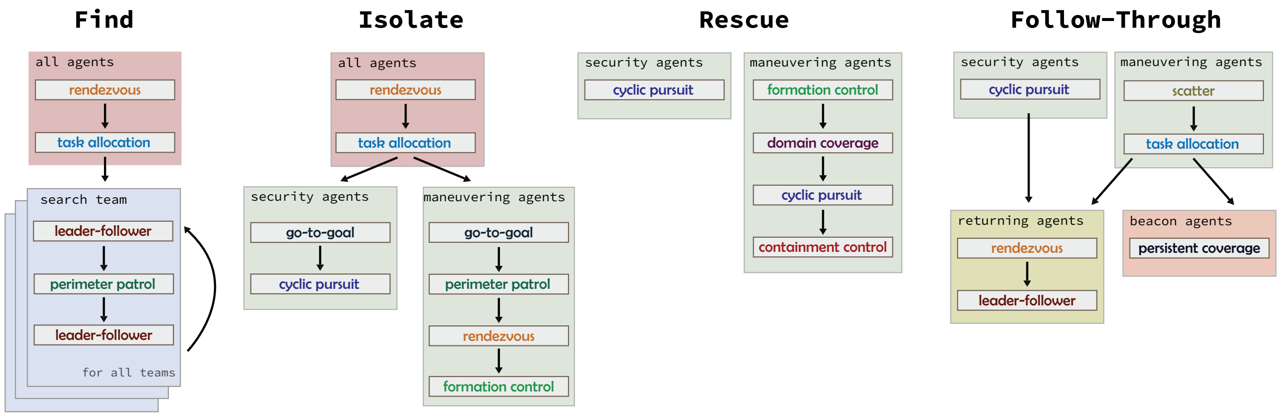


Fig. 3: Mission design chart showing how distributed multi-agent behaviors are composed together to tackle the Securing a Building mission. The four bold titles are the mission phases and the large boxes below them indicate specific agent roles and associated behaviors. The arrows in the chart indicate the transitions between different behaviors. Note that different implementations can be used to execute these behaviors, and we only insist that they produce the same results as described in the chart.

b) ISOLATE: Robots gather near the base station (*rendezvous*), then are divided into *security* and *maneuvering* agents (*task allocation*). After traveling from the base to the vicinity of the target building (*go-to-goal*), security agents protect the building's perimeter (*cyclic pursuit*). This behavior will continue until the end of the RESCUE phase. Meanwhile, the maneuvering agents locate the building's entrance, by following its perimeter (*perimeter patrol*). Once the entrance has been found, the maneuvering agents gather at the entrance (*rendezvous*) and create a compact formation (*formation control*) and prepare to enter.

c) RESCUE: The maneuvering agents enter the building in formation (*formation control*) and distribute themselves in order to cover the interior area (*area coverage*). Once the location of the subject to rescue is identified, all the robots form a circular closure around the subject (*cyclic pursuit*). Then, the robots transport the subject to the safety zone, while maintaining the circular closure around the subject (*containment control*).

d) FOLLOW-THROUGH: The maneuvering agents spread (*scatter*) uniformly over the interior of the building. To signify that the area has been cleared, certain members of the maneuvering agents stay inside the building as beacons (*persistent coverage*). The rest of the maneuvering agents and the security agents reunite at the entrance outside the building (*rendezvous*). At last, they return to the base station (*leader-follower*).

C. Results

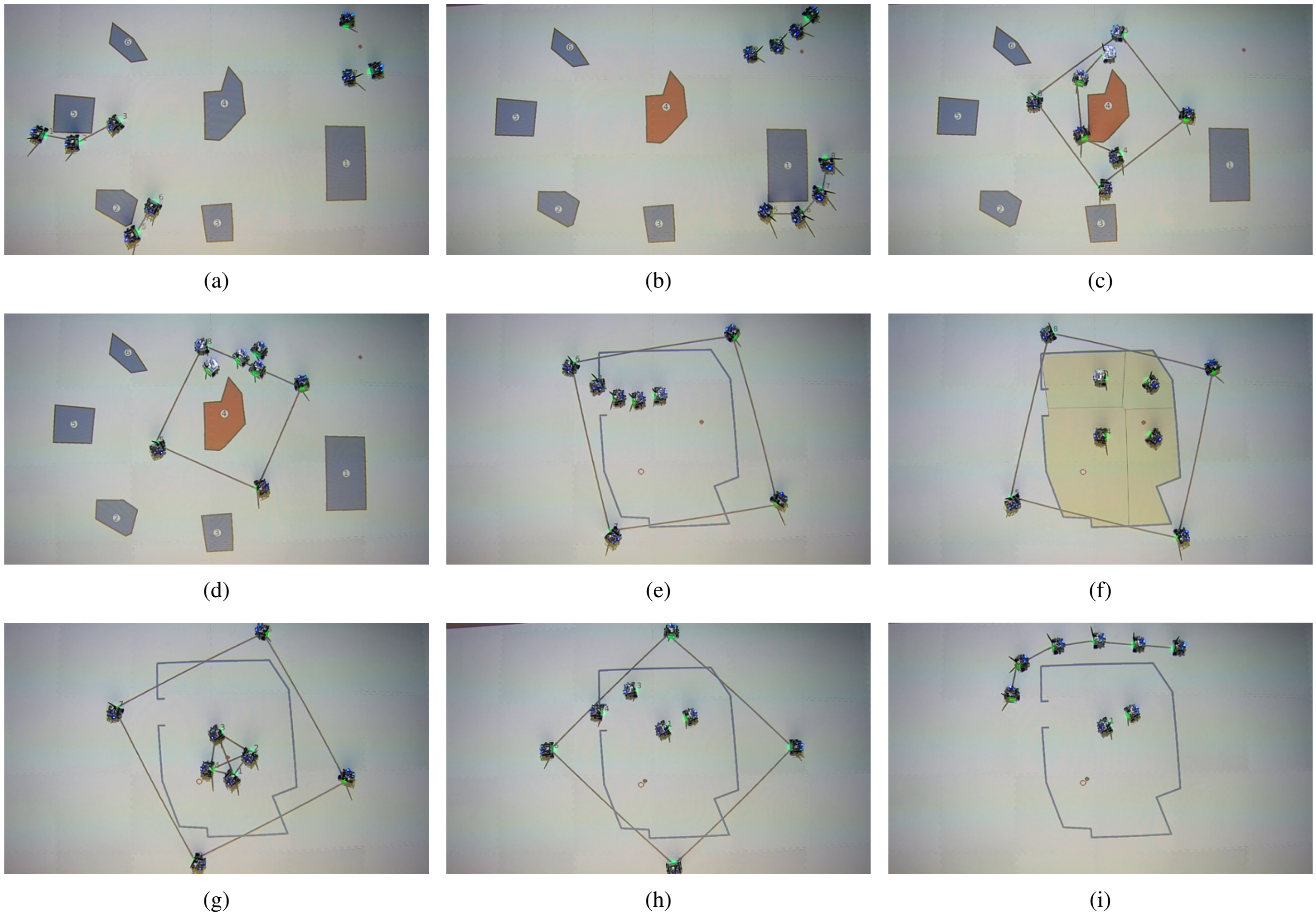
We tested the behavior composition framework on the Securing a Building mission, which was executed on the Robotarium [19], a remotely accessible multi-robot platform. In Fig. 4, we display selected snapshots of the mission obtained by a camera mounted on the ceiling. In the experiment, 8 differential-drive robots, indexed $1, \ldots, 8$ are deployed in a simulated urban environment composed of 6 buildings, blue polygons indexed $1, \ldots, 6$. In this experiment, we simulate a maximum sensor range $\Delta = 0.5m$. Because of the different spatial scales between FIND/ISOLATE phases and RESCUE/FOLLOW-THROUGH phases, the entire mission is divided in two parts. In the first part (Fig. 4a to Fig. 4d) the experiment is performed at a *neighborhood*-level scale. The remaining two phases are executed in a zoomed-in environment, which focuses on the one building of interest (Fig. 4e to Fig. 4i).

During FIND phase (Fig. 4a and 4b), two groups of robots $\text{TEAM1} : \{1, 2, 3, 4\}$ and $\text{TEAM2} : \{5, 6, 7, 8\}$ investigates preassigned lists of buildings, leaving some agents near the base station (the purple filled dot in the top right corner) if destination building cannot be reached without breaking the connectivity constraints. The red polygon in Fig. 4b to Fig. 4d is the target building after being inspected by TEAM1. Once the target building has been identified, the ISOLATE phase (Fig. 4c and 4d) starts. The maneuvering agents look for the entrance, while the security agents secure the outer perimeter.

During the RESCUE phase (Fig. 4e to 4g), the agents inside the building, i.e. TEAM1, localize the target (red dot) using and escort it to the safe area (red circle). In order, maneuvering agents enter the building (Fig. 4e), perform Voronoi coverage (Fig. 4f), and move the target to the safe area (Fig. 4g). Finally, during FOLLOW-THROUGH phase, agents 1 and 2 are left as beacon inside the building, while 3 and 4 reunite with agents outside the building (Fig. 4h). In Fig. 4i all except the beacon agents return to the base.

VII. Conclusion

Sequential execution of coordinated behaviors can be employed by a team of robots to solve real-world complex missions. However, the sequence of behaviors can be executed only if the robots can create all the required interactions in finite time. In this paper, we described a provably correct framework for sequential composition of coordinated behaviors designed on finite-time convergence control barrier functions. The resulting composition framework is formulated in the form of a quadratic program based controller, which is



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 4: Overhead screen-shots from experiments on the Robotarium multi-robot test-bed. A team of eight robots is divided in TEAM1 : $\{1, 2, 3, 4\}$ and TEAM2 : $\{5, 6, 7, 8\}$. Because of the different spatial scales between FIND/ISOLATE phases and RESCUE/FOLLOW-THROUGH phases the mission is executed on two different environments. Each team is assigned with a list of three buildings to inspect sequentially. FIND: (a) perimeter patrol of buildings 2 and 5; (b) building 4 is identified as the target building, while TEAM1 waits for TEAM2 to return to base. ISOLATE: (c) TEAM2 secures perimeter of building, while TEAM1 inspects exterior of building, searching for the entrance. RESCUE: TEAM1 (d) gathers in proximity of building's entrance, (e) enters the building, (f) performs domain coverage of the building until target (red dot) is identified; after this, (g) robots escort target to safe location (red circle). FOLLOW-THROUGH: (h) two robots are left as beacons inside the building while the remaining robots reunite with TEAM2 outside; (i) all robots except beacons proceed back to base.

solved locally by individual robots. The proposed framework was performed on a multi-robot test-bed for the solution of a large-scale complex scenario, which is presented as a stand-alone contribution. To this end, because of its modularity and multi-tasking requirements, the "Securing a Building" mission is proposed as an ideal testing environment for multi-robot control techniques.

Appendix A
Open Questions in the Securing a Building Scenario

Testing the performance of techniques and algorithms for the control of multi-agent systems in real-world scenarios is a challenging task. This is particularly true when addressing novel approaches, where the focus is generally on a specific aspect of the problem considered in an isolated scenario. Testing isolated scenarios, however, may not account for the challenges unique to more complex missions, whose sub-problems are closely intertwined. To this end, the "Securing a Building" mission presents itself as a promising testing framework, because of the possibility to assess specific methodologies and techniques, in a mission structure that captures the intimate interplay between the agents and the environment across different mission phases (Section 2). In this section, we discuss a number of open issues, for which this mission serves as an effective testing framework when aiming to evaluate performance and relevance of new techniques in the context of a complex mission.

a) Agent Recruitment: Considerable efforts have been devoted to the development of team composition techniques in the presence of heterogeneous robots [47], [48]. Based on the skill set required to solve a particular task, e.g., certain actuation, sensing, locomotion, or communication capabilities, the question is to find a recruitment rule that produces a team capable of delivering the best performance. For instance, in the rescue phase of our mission, robots capable of opening doors may be required in the *maneuvering* agents for entering the target building. However, these skills may not be required in the find phase, where fast and agile aerial robots have more performance advantages.

b) Communication: In the context of autonomous networked systems, central roles are played by the flow of information between agents, and the infrastructure required for it [49]. A number of questions can be posed in relation to the distribution of agents over a domain, given the constraints of communication systems, such as limited range, power requirements, and privacy of the information. In our mission, these questions may concern how to propagate the mission information back to human operators when agents are far away, how to communicate with minimum energy consumption, how to ensure the communication scheme does not expose mission plans in the presence of adversarial components, etc.

c) Unknown Environment: The amount of prior knowledge about the environment plays an important role in the definition of both low-level robot controllers and high-level mission plans. Active currents of research target the design of distributed solutions to the localization and mapping problems [50], and the Securing a Building mission can be effectively used to test the performance of these techniques in real-world scenarios. For instance, new techniques may focus on balancing the trade-off between exploitation of the known and exploration of the unknown. Given incomplete information about the buildings in the environment, agents have to strategically plan the investigation of buildings with more prior information, an expedient process, and the investigation of buildings with less or no prior information, a time-consuming process.

d) Resilience: In real-world scenarios, failure of the mission can be attributed to factors such as damaged parts, sensing errors, communication dropouts, delays, control disturbances, reduction of functionalities due to adversarial attacks, etc. A number of different research thrusts focus on the problem of detecting and responding to faults and malicious attacks in multi-agent and cyber-physical systems [29], [27], [9]. Therefore, the solution to the mission can focus on making controller more resilient to these failures, by explicitly modeling them in the mission.

e) Adaptivity: Dynamic planning and autonomous adjustment to changing environments are emerging research currents in the multi-agent system communities [17]. A number of different aspects of the Securing a Building mission can be extended to test this family of techniques. For instance, agents may discover that the original routes planned for searching the target building are no longer feasible, the subjects of interest are not present in the target building, different skills are required from what agents have expected, etc. In these scenarios, agents should have the ability to reason about the situation and adjust the mission plan.

f) Composability of Behaviors: The possibility to encode real-world missions through sequences of behaviors requires the employment of techniques that ensure correct switching between these behaviors. For example, in order to perform coordinated behaviors in a distributed fashion, agents need to maintain or establish the necessary communication links with each other. To this end, the correct flow of information between the agents not only is necessary for the execution of the single behaviors but also for the switching between them.

References

[1] E. Ackerman, "Flying lampshadebots come alive in cirque du soleil performance," *iEEE Spectrum*, 2014.

[2] X. Du, C. E. Luis, M. Vukosavljev, and A. P. Schoellig, "Fast and in sync: Periodic swarm patterns for quadrotors," *arXiv preprint arXiv:1810.03572*, 2018.

[3] M. Santos, Y. Diaz-Mercado, and M. Egerstedt, "Coverage control for multirobot teams with heterogeneous sensing capabilities," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 919–925, 2018.

[4] D. Shishika and V. Kumar, "Local-game decomposition for multiplayer perimeter-defense problem," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 2093–2100.

[5] H. Han and R. G. Sanfelice, "A hybrid control algorithm for object grasping using multiple agents," in *2018 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2018, pp. 652–657.

[6] J. Suarez and R. Murphy, "A survey of animal foraging for directed, persistent search by rescue robotics," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*. IEEE, 2011, pp. 314–320.

[7] D. Zelazo, M. Mesbahi, and M.-A. Belabbas, "Graph theory in systems and controls," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 6168–6179.

[8] J. Cortés and M. Egerstedt, "Coordinated control of multi-robot systems: A survey," *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 6, pp. 495–503, 2017.

[9] J. Lin, A. S. Morse, and B. D. Anderson, "The multi-agent rendezvous problem," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, vol. 2. IEEE, 2003, pp. 1508–1513.

[10] W. Ren, R. W. Beard, and T. W. McLain, "Coordination variables and consensus building in multiple vehicle systems," in *Cooperative control*. Springer, 2005, pp. 171–188.

[11] J. L. Ramirez, M. Pavone, and E. Frazzoli, "Cyclic pursuit for spacecraft formation control," in *Proceedings of the American Control Conference*, 2009, pp. 4811–4817.

[12] J. R. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation maneuvers," *IEEE transactions on robotics and automation*, vol. 19, no. 6, pp. 933–941, 2003.

[13] I. Buckley and M. Egerstedt, "Infinitesimally shape-similar motions using relative angle measurements," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1077–1082.

[14] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[15] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010, vol. 33.

[16] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Transactions on Automatic control*, vol. 52, no. 5, pp. 863–868, 2007.

[17] S. Nagavalli, N. Chakraborty, and K. Sycara, "Automated sequencing of swarm behaviors for supervisory control of robotic swarms," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2674–2681.

[18] A. Li, L. Wang, P. Pierpaoli, and M. Egerstedt, "Formally correct composition of coordinated behaviors using control barrier certificates," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3723–3729.

[19] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The robotarium: A remotely accessible swarm robotics research testbed," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1699–1706.

[20] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*. Springer Science & Business Media, 2009.

[21] R. C. Arkin, R. C. Arkin *et al.*, *Behavior-based robotics*. MIT press, 1998.

[22] P. Martin and M. B. Egerstedt, "Hybrid systems tools for compiling controllers for cyber-physical systems," *Discrete Event Dynamic Systems*, vol. 22, no. 1, pp. 101–119, 2012.
[23] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas, "Symbolic planning and control of robot motion [grand challenges of robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 61–70, 2007.
[24] H. Kress-Gazit, M. Lahijanian, and V. Raman, "Synthesis for robots: Guarantees and feedback for robot behavior," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 211–236, 2018.
[25] M. Srinivasan, S. Coogan, and M. Egerstedt, "Control of multi-agent systems with finite time control barrier certificates and temporal logic," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 1991–1996.
[26] P.-J. Meyer and D. V. Dimarogonas, "Hierarchical decomposition of ltl synthesis problem for nonlinear control systems," *IEEE Transactions on Automatic Control*, 2019.
[27] J. Chen, S. Moarref, and H. Kress-Gazit, "Verifiable control of robotic swarm from high-level specifications," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 568–576.
[28] E. Klavins and D. E. Koditschek, "A formalism for the composition of concurrent robot behaviors," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 4. IEEE, 2000, pp. 3395–3402.
[29] A. Marino, L. Parker, G. Antonelli, and F. Caccavale, "Behavioral control for multi-robot perimeter patrol: A finite state automata approach," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 831–836.
[30] M. Vukosavljev, A. P. Schoellig, and M. E. Broucke, "Hierarchically consistent motion primitives for quadrotor coordination," *arXiv preprint arXiv:1905.00500*, 2019.
[31] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.
[32] L. Sabattini, C. Secchi, N. Chopra, and A. Gasparri, "Distributed control of multirobot systems with global connectivity maintenance," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1326–1332, 2013.
[33] M. M. Zavlanos, H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Hybrid control for connectivity preserving flocking," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2869–2875, 2009.
[34] Y. Igarashi, T. Hatanaka, M. Fujita, and M. W. Spong, "Passivity-based attitude synchronization in $se(3)$," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1119–1134, 2009.
[35] L. Wang, A. D. Ames, and M. Egerstedt, "Multi-objective compositions for collision-free connectivity maintenance in teams of mobile robots," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 2659–2664.
[36] R. K. Ramachandran, J. A. Preiss, and G. S. Sukhatme, "Resilience by reconfiguration: Exploiting heterogeneity in robot teams," *arXiv preprint arXiv:1903.04856*, 2019.
[37] J. Panerati, M. Minelli, C. Ghedini, L. Meyer, M. Kaufmann, L. Sabattini, and G. Beltrame, "Robust connectivity maintenance for fallible robots," *Autonomous Robots*, vol. 43, no. 3, pp. 769–787, 2019.
[38] V. S. Varadharajan, B. Adams, and G. Beltrame, "The unbroken telephone game: Keeping swarms connected," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 2241–2243.
[39] P. Twu, P. Martin, and M. Egerstedt, "Graph process specifications for hybrid networked systems," *IFAC Proceedings Volumes*, vol. 43, no. 12, pp. 65–70, 2010.
[40] S. P. Bhat and D. S. Bernstein, "Finite-time stability of continuous autonomous systems," *SIAM Journal on Control and Optimization*, vol. 38, no. 3, pp. 751–766, 2000.
[41] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.
[42] B. G. Pachpatte, *Inequalities for differential and integral equations*. Elsevier, 1997.
[43] 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.
[44] E. Squires, P. Pierpaoli, R. Konda, S. Coogan, and M. Egerstedt, "Composition of safety constraints with applications to decentralized fixed-wing collision avoidance," *arXiv preprint arXiv:1906.03771*, 2019.
[45] R. K. Williams and G. S. Sukhatme, "Observability in topology-constrained multi-robot target tracking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1795–1801.
[46] "Military operations in urbanized terrain," *US Army Field Manual 90-10*, 1975.
[47] A. Prorok, M. A. Hsieh, and V. Kumar, "Formalizing the impact of diversity on performance in a heterogeneous swarm of robots," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5364–5371.
[48] M. Koes, I. Nourbakhsh, K. Sycara, M. Koes, K. Sycara, I. Nourbakhsh, M. Koes, I. Nourbakhsh, K. Sycara, S. D. Ramchurn *et al.*, "Heterogeneous multirobot coordination with spatial and temporal constraints," in *AAAI*, vol. 5, 2005, pp. 1292–1297.
[49] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in uav communication networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2016.
[50] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, "Collaborative monocular slam with multiple micro aerial vehicles," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3962–3970.